

# Intelligente Sehsysteme

## **11 Merkmale (Features)**

---

Feature Engineering

Handcrafted Features, Mid-level Features

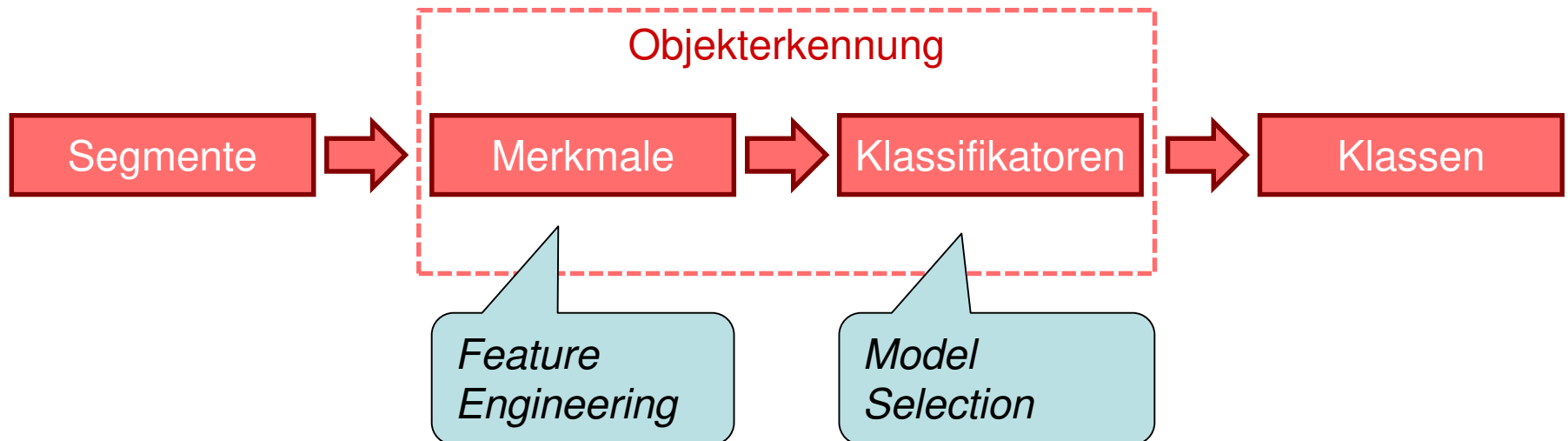
Hierarchical and trained Features

*Volker Steinhage*

# Entwurfsaspekte der Objekterkennung

Objekterkennung zeigt zwei Entwurfsaspekte

- (1) **Feature Engineering**: Welche Merkmale (engl. Features) sind geeignet?
- (2) **Model Selection**: Welches Modell des maschinellen Lernens\* ist geeignet?



\* Siehe Vorlesung „Grundlagen der KI“: Lernverfahren, Hyperparameter, etc. (SVM, LDA, Boosting, KNN, Clustering, ...)

# Inhalt: Ansätze des Feature Engineerings

## Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor



## Mainstream Modern Pattern Recognition: Unsupervised mid-level features



## Deep Learning: Representations are hierarchical and trained

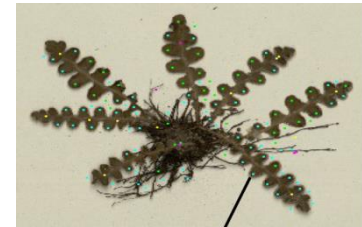


# Handcrafted Features

## Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor



- Lokale Merkmale
  - werden an verschiedenen Orten des Segments ermittelt
  - z. B. **SIFT-Deskriptoren von Keypoints**
- Globale Merkmale
  - beziehen sich auf das gesamte Segment
  - umfassen u.a. **Formmerkmale, Regionenmerkmale (Textur, Farben, Intensitäten, ...), Hu-Momente**, Elliptische Fourier-deskriptoren (EFD), Radiale Basisfunktionen (RBF), **Local Binary Patterns (LBP)**



0.142	0.001	0.712	0.000	0.134		0.121	0.258	0.307
1	2	3	4	5	...	126	127	128



Bildquellen: (1) <http://www.pamitc.org/cvpr15/files/lecun-20150610-cvpr-keynote.pdf> (26.11.2019)

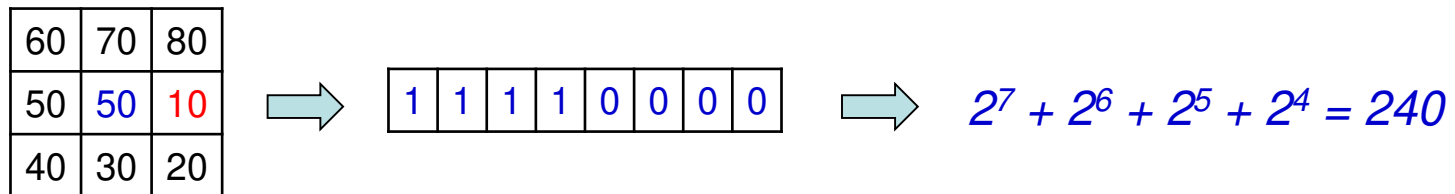
(2) J.Grimm, M. Hoffmann, B. Stöver, K. Müller, V. Steinhage: Image-based Identification of Plant Species using a Model-Free Approach and Active Learning. LNCS, Vol. 9904, Springer, KI 2016: Advances in Artificial Intelligence, 169-176, 2016.

(3) Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

# Local Binary Patterns

Local Binary Patterns (LBP) ist ein histogrammbasierter Deskriptor: <sup>(1),(2)</sup>

- Für jedes Segmentpixel wird dessen Intensität  $g_c$  mit den Intensitäten  $g_p$  ( $p = 0, \dots, 7$ ) aller Nachbapixel aus der Achtnachbarschaft im Uhrzeigersinn mit dem „Ostpixel“ startend verglichen: wenn  $g_p \geq g_c$ , dann resultiert der Wert 1, andernfalls 0
- So wird jedem Segmentpixel wird eine **achtstellige Binärzahl** zuordnet
- Der **LBP-Deskriptor** eines Segments  $s$  ist das 256-dimensionale Histogramm über den achtstelligen Binärzahlen (bzw. deren Dezimalwerten) seiner Pixel.



<sup>(1)</sup> T. Ojala, M. Pietikainen, D. Harwood. "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994), vol. 1, pp. 582 - 585.

<sup>(2)</sup> T. Ojala, M. Pietikainen, D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions," Pattern Recognition, Vol. 29, No. 1, 1996, pp. 51-59.

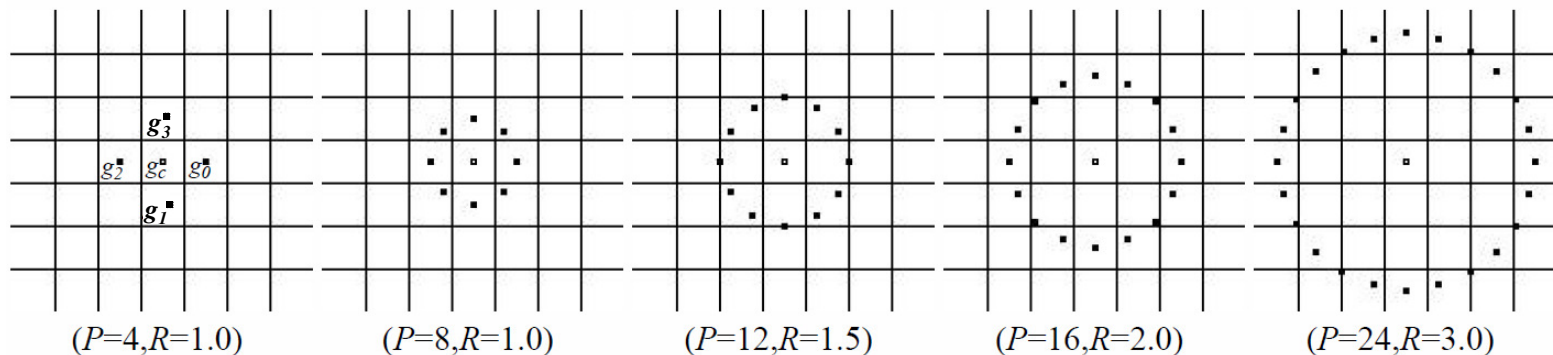
# Local Binary Patterns: Parameter P, R

Local Binary Patterns (LBP) sind über zwei Parameter generalisierbar:

Die Nachbarschaft wird verallgemeinert als kreisförmige und symmetrische Nachbarschaft, die  $P$  gleichmäßig verteilte Pixel auf einem Kreis mit Radius  $R$  ( $R > 0$ ) zeigt.

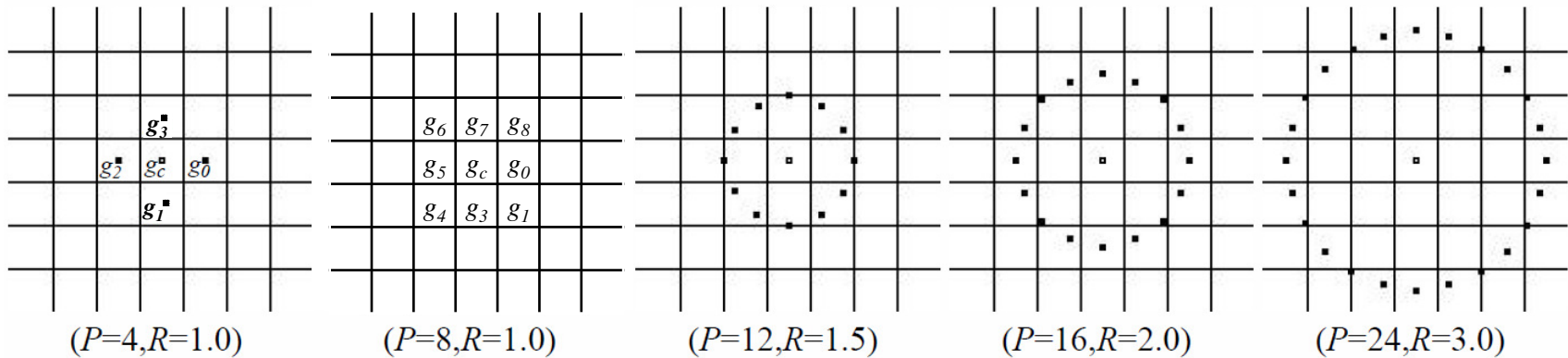
Mit  $g_c = (0,0)$  folgen die Koordinaten der  $g_p$  zu  $(-R \cdot \sin(2\pi p/P), R \cdot \cos(2\pi p/P))$  für  $p = 0, 1, \dots, P-1$ .

Intensität für Werte von  $g_p$ , die nicht exakt in Pixelzentren liegen, werden durch Interpolation ermittelt.



Circularly symmetric neighbor sets for different  $(P,R)$

# Local Binary Patterns: formalisiert mit Parametern P, R



$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p$$

where

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

# Rotation Invariant Local Binary Patterns

Rotation Invariant Local Binary Patterns ( $LBP^{ri}$ ) <sup>(3)</sup>

- zählt Rotationsvarianten eines Musters durch zirkuläre bitweise Rechtsshifts auf
- reduziert alle Rotationsvarianten auf die Variante mit der kleinsten Binärzahl, welche die gesetzten Bits auf den Positionen mit geringster Signifikanz hat:

$$LBP_{P,R}^{ri} = \min \{ ROR(LBP_{P,R}, i) \mid i = 0, 1, \dots, P-1 \}$$

where  $ROR(x, i)$  performs a circular bit-wise right shift on the  $P$ -bit number  $x$   $i$  times

Zwei von acht Rotationsvarianten eines Musters (schwarz = 0, weiß = 1):



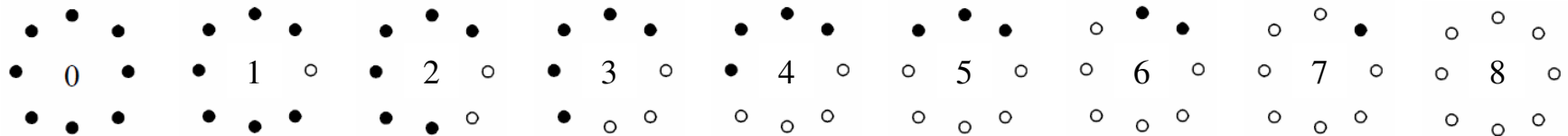
<sup>(3)</sup> Ojala, T., Pietikainen, M., Maenpaa, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7): 971–987, 2002.



# Uniform Local Binary Patterns

## Uniform Rotational Invariant Local Binary Patterns ( $LBP^{riu2}$ ) <sup>(3)</sup>

- basieren auf Untersuchungen, nach denen Muster mit maximal *zwei* Wechseln zwischen 0 und 1 die größten Beiträge zur Erkennung leisten
- kodieren Mikromerkmale wie Ecken- und Kantensegmente
- zeigen lediglich 9 Muster, so dass das resultierende Histogramm nur 10 Einträge zeigt (Eintrag 10 für alle anderen nicht uniformen Muster):



nine ‘uniform’ patterns, and the numbers  
inside them correspond to their unique  $LBP_{8,R}^{riu2}$  codes

<sup>(3)</sup> Ojala, T., Pietikainen, M., Maenpaa, T., “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7): 971–987, 2002.

# Momente (1)

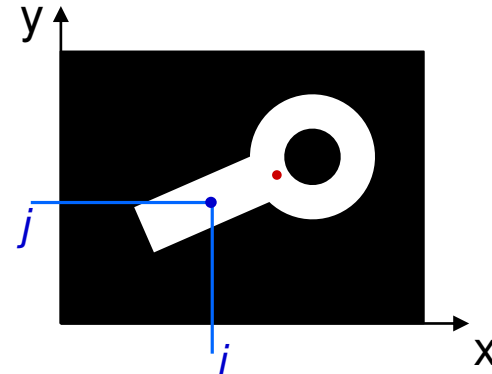
Formmerkmale von Bildsegmenten  $s$  wurden bereits eingeführt (Vorl. 10)

- Für binäres Bild  $B(x,y)$ :

mit

1 = weiß = Segmentpixel

0 = schwarz = Hintergrundpixel



- Einfache Formmerkmale sind z.B. folg. Statistiken:

– Fläche:

$$F = \sum_{(i,j) \in s} B(i,j)$$

– Schwerpunkt  $(i_\mu, j_\mu)$ :  $i_\mu = \frac{1}{F} \sum_{(i,j) \in s} i \cdot B(i,j)$ ,  $j_\mu = \frac{1}{F} \sum_{(i,j) \in s} j \cdot B(i,j)$

# Momente (2)

---

Ein allg. Formalismus sind Momente

$$m_{pq}(s) = \sum_{(i,j) \in s} i^p j^q$$

- Die Ordnung von Moment  $m_{pq}$  ist  $p + q$
- Die Fläche  $F$  eines Segments entspricht  $m_{00}$
- Der Schwerpunkt

$$(i_\mu, j_\mu) = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

---

Für Grauwertbilder mit Intensitätsfunktion  $g(x,y)$ :

$$m_{pq}(s) = \sum_{(i,j) \in s} i^p j^q g(i,j)$$

# Zentrale Momente (1)

---

Momente liefern nicht translationsinvariante Ergebnisse:

$$m_{pq}(s) = \sum_{(i,j) \in s} i^p j^q$$

- Lösung: zentrale Bildmomente:

$$\mu_{pq}(s) = \sum_{(i,j) \in s} (i - i_\mu)^p (j - j_\mu)^q$$

mit Schwerpunkt  $(i_\mu, j_\mu)$

# Zentrale Momente (2)

---

Die zentrale Bildmomente:

$$\mu_{pq}(s) = \sum_{(i,j) \in s} (i - i_{\mu})^p (j - j_{\mu})^q$$

- mit zentr. Moment 0-ter Ordnung  $\mu_{00}(s) = \text{Fläche}$
- mit zentr. Momenten 2-ter Ordnung  $\mu_{ij}(s) = \text{Trägheitsmomente mit } p + q = 2$   
aus Vorl. 10

# Normalisierte zentrale Momente (1)

---

Zentrale Momente liefern nicht skalierungsinvariante Ergebnisse:

$$\mu_{pq}(S) = \sum_{(i,j) \in S} (i - i_\mu)^p (j - j_\mu)^q$$

- Lösung: normalisierte zentrale Momente:

$$\eta_{pq}(S) = \frac{\mu_{pq}(S)}{[\mu_{00}(S)]^\gamma}$$

mit  $\gamma = \frac{p+q}{2} + 1$  für  $p+q \geq 2$

# Hu-Momente (1)

Auf der Basis der normalisierten Momente 2. und 3. Ordnung stellt Hu (1962)\* sieben Momente vor, die invariant bzgl. Translation, Skalierung, Spiegelung (bis auf Vorzeichen) und Rotation sind.

$$\phi_1 = \eta_{2,0} + \eta_{0,2}$$

$$\phi_2 = (\eta_{2,0} - \eta_{0,2})^2 + 4 \cdot \eta_{1,1}^2$$

$$\phi_3 = (\eta_{3,0} - 3 \cdot \eta_{1,2})^2 + (3 \cdot \eta_{2,1} - \eta_{0,3})^2$$

$$\phi_4 = (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{2,1} + \eta_{0,3})^2$$

$$\begin{aligned} \phi_5 = & (\eta_{3,0} - 3 \cdot \eta_{1,2})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3 \cdot (\eta_{2,1} - 3 \cdot \eta_{0,3})^2] \\ & + (3 \cdot \eta_{3,0} + \eta_{1,2})(\eta_{2,1} + \eta_{0,3})[3 \cdot (\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] \end{aligned}$$

$$\begin{aligned} \phi_6 = & (\eta_{2,0} - \eta_{0,2})[(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] \\ & + 4 \cdot \eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{2,1} + \eta_{0,3}) \end{aligned}$$

$$\begin{aligned} \phi_7 = & (3 \cdot \eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3 \cdot (\eta_{2,1} + \eta_{0,3})^2] \\ & + (3 \cdot \eta_{1,2} - \eta_{3,0})(\eta_{0,3} + \eta_{2,1})[3 \cdot (\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] \end{aligned}$$

\* Ming-Kuei Hu. Visual pattern recognition by moment invariants. IRE Transactions on Information Theory **8**(2), 1962.

# Hu-Momente (2)

---

- Die sieben invarianten Momente  $\phi_1, \dots, \phi_7$  werden als Hu-Momente bezeichnet.
- Da die Hu-Momente sehr klein werden können, wird i.a. eine Skalierung vorgenommen:\*

$$\phi_i' = \text{sign}(\phi_i) \log_{10}(|\phi_i|)$$

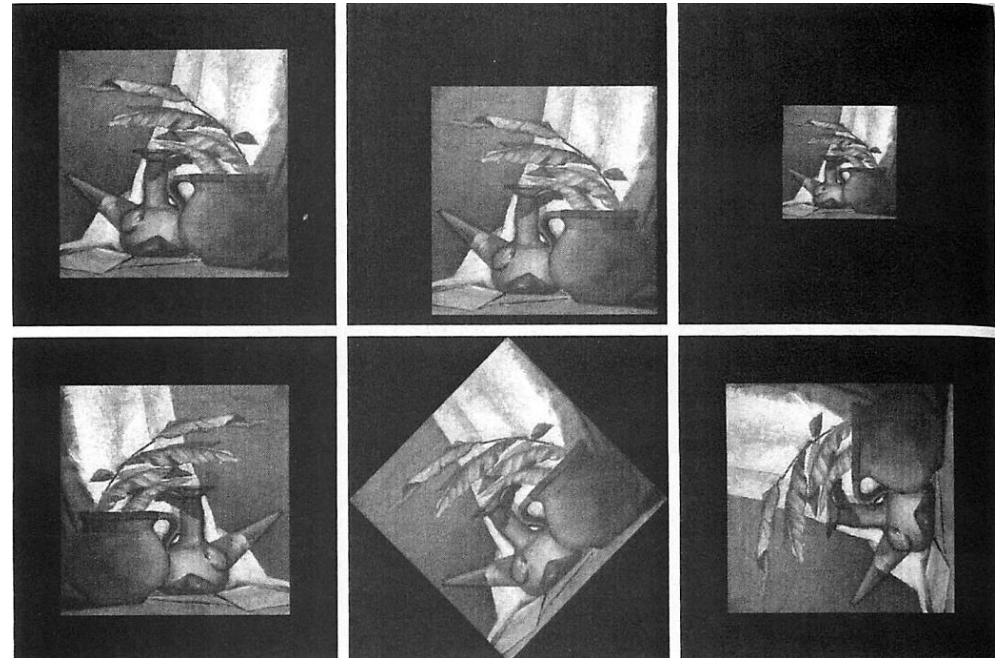
---

\* Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing, 3 Edition, Pearson, 2008, p. 841.



# Hu-Momente (3)

Beispiel für Invarianz  
der Hu-Momente



Moment Invariant	Original Image	Translated	Half Size	Mirrored	Rotated 45°	Rotated 90°
$\phi_1$	2.8662	2.8662	2.8664	2.8662	2.8661	2.8662
$\phi_2$	7.1265	7.1265	7.1257	7.1265	7.1266	7.1265
$\phi_3$	10.4109	10.4109	10.4047	10.4109	10.4115	10.4109
$\phi_4$	10.3742	10.3742	10.3719	10.3742	10.3742	10.3742
$\phi_5$	21.3674	21.3674	21.3924	21.3674	21.3663	21.3674
$\phi_6$	13.9417	13.9417	13.9383	13.9417	13.9417	13.9417
$\phi_7$	-20.7809	-20.7809	-20.7724	20.7809	-20.7813	-20.7809

Vorzeichen von  $\phi_7$   
wechselt für Spiegelung

# Mid-Level Features

## ■ Mainstream Modern Pattern Recognition: Unsupervised mid-level features

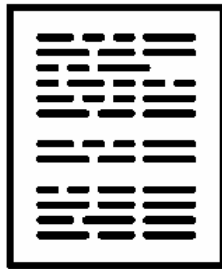


*Mid-level features* werden auf der Basis von *handcrafted Features* erzeugt:

- Unüberwachte Ansätze sind z.B.
  - Bag of Visual Words
  - Principal Component Analysis (PCA)
- Überwachte Ansätze sind z.B.
  - Multi-Layer Perceptrons (MLP)
  - Sparse Coding

# Bag of Words

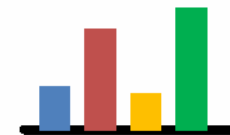
*Bag-of-Words* (BoW) ist ursprünglich ein Ansatz zur effizienten Repräsentation von Dokumenten für *Information Retrieval* und Sprachverarbeitung.



Document



Word Occurrences



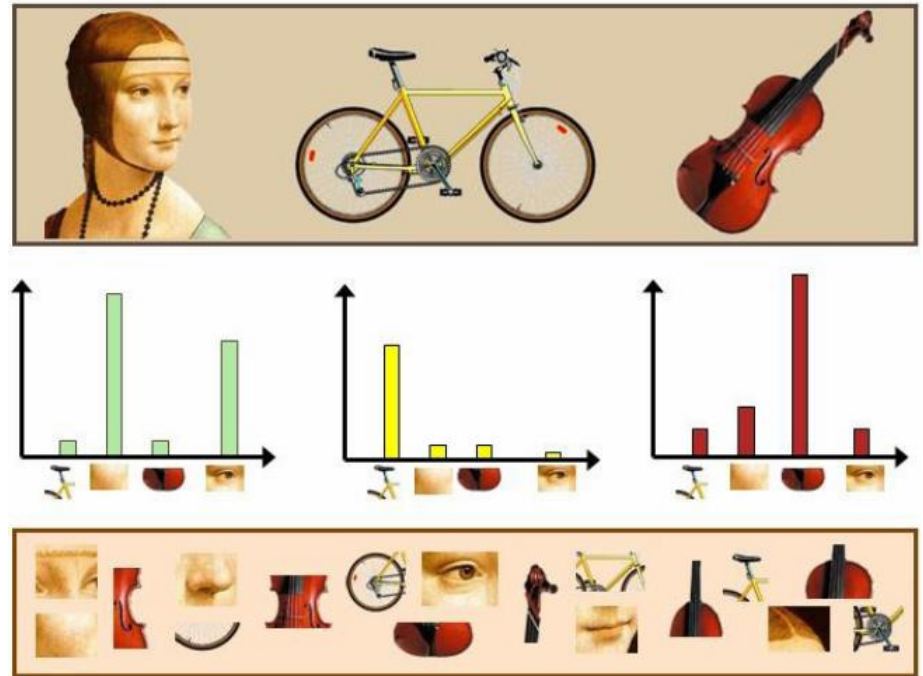
Histogram  
Bag-of-Words

- Vorgabe eines Wörterbuchs (Vokabulars)
- Jedes Dokument wird durch die Zahlen der auftretenden Wörter aus dem Vokabular repräsentiert:
  - Histogramm der Auftreten der Wörter aus dem Vokabular
  - Vektor mit Dimension = Umfang des Wörterbuchs
  - Klassifikation durch MAP, SVM, ...

# Bag of Visual Words

Sivic & Zissermann (2003)\* haben die BoW-Idee für die Klassifikation von Bildern adaptiert.

- Grundlage: Bildmerkmale wie z.B. SIFT-Deskriptoren
- Clustering von SIFT-Deskriptoren aus einer Trainingsmenge von Bildern/Videos z.B. durch  $k$ -Means in  $k$  Cluster s. Anhang
- Wörterbuch = Zentren der  $k$  Cluster =  $k$  Visual Words
- Zur Klassifikation eines Bildes s. Anhang
  - Nächste-Nachbar-Zuordnung abgeleiteter SIFT-Deskriptoren zu Wörterbuch
  - Histogramm der Auftreten der visuellen Wörter aus dem Wörterbuch
  - Vektor mit Dimension = Umfang des Wörterbuchs
  - Klassifikation durch MAP, SVM, ...



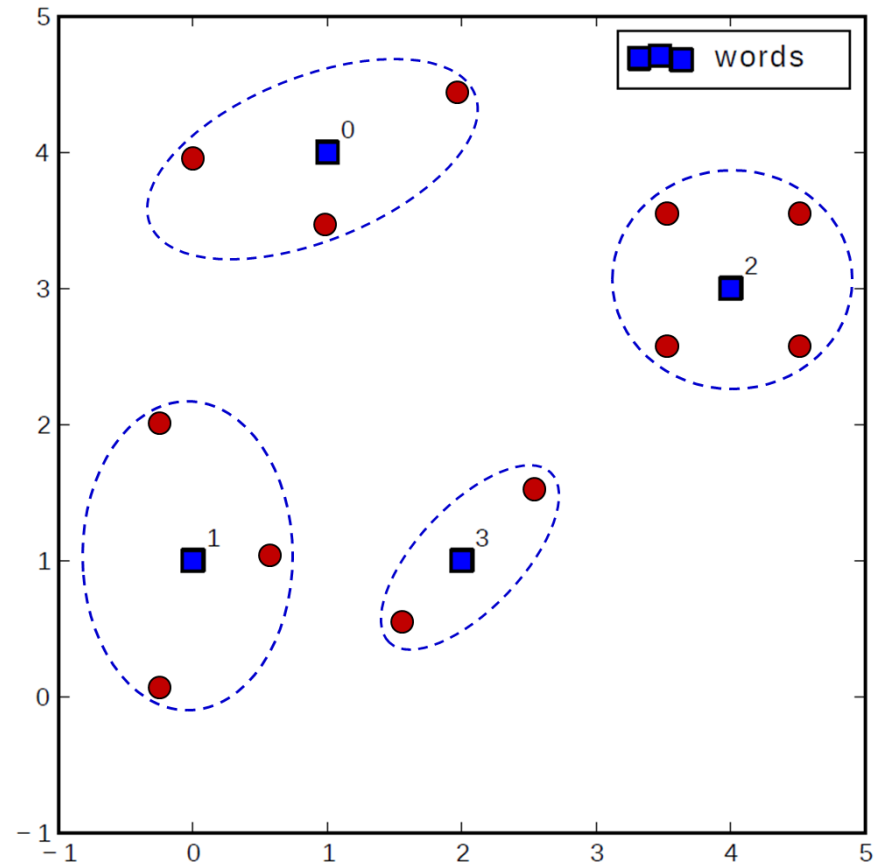
Bildquelle: Li Fei-Fei



# Bag of Visual Words: Wörterbuch

Beispiel:

- Geg.: 2-dim. Deskriptoren
- Clustering der Deskriptoren aus Trainingsmenge durch  $k$ -Means in  $k$  Cluster mit
  - $k = 4$
  - Euklid. Distanz
- Wörterbuch = *Dictionary*  $D \leftarrow$  Zentren der  $k$  Cluster =  $k$  Visual Words  
hier:  $D = \{(0,1), (1,4), (2,1), (4,3)\}$

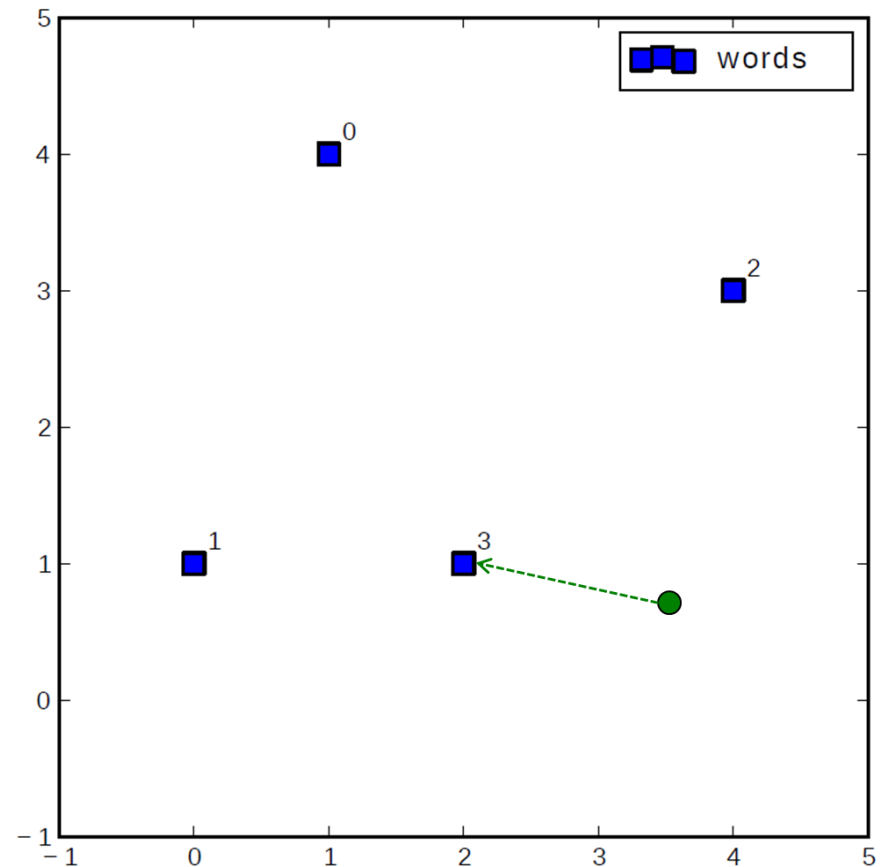


# Bag of Visual Words: Nächste-Nachbar-Zuordnung

Beispiel:

- Geg.: 2-dim. Deskriptoren
- Zuordnung eines Deskriptors eines ungesehenen Segments  $s$  durch
  - Nächste-Nachbar-Zuordnung
  - Euklid. Distanz

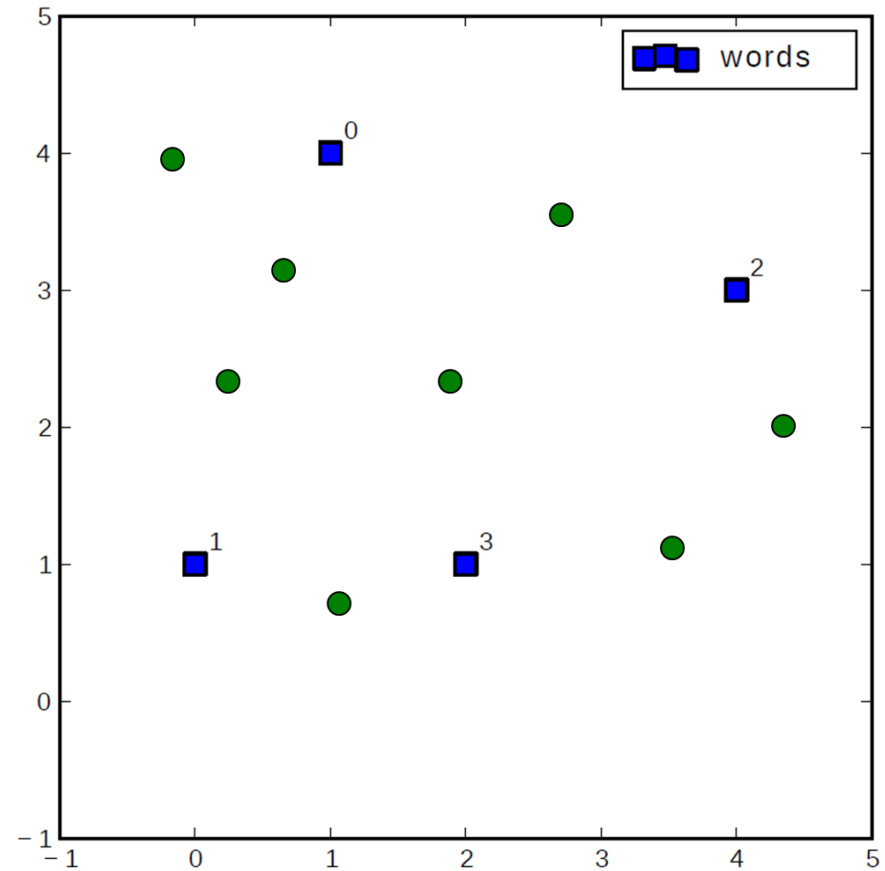
hier: Zuordnung von  
(3.5,0.75) zu Wort (2,1)



# Bag of Visual Words: BoW-Repräsentation eines Segments

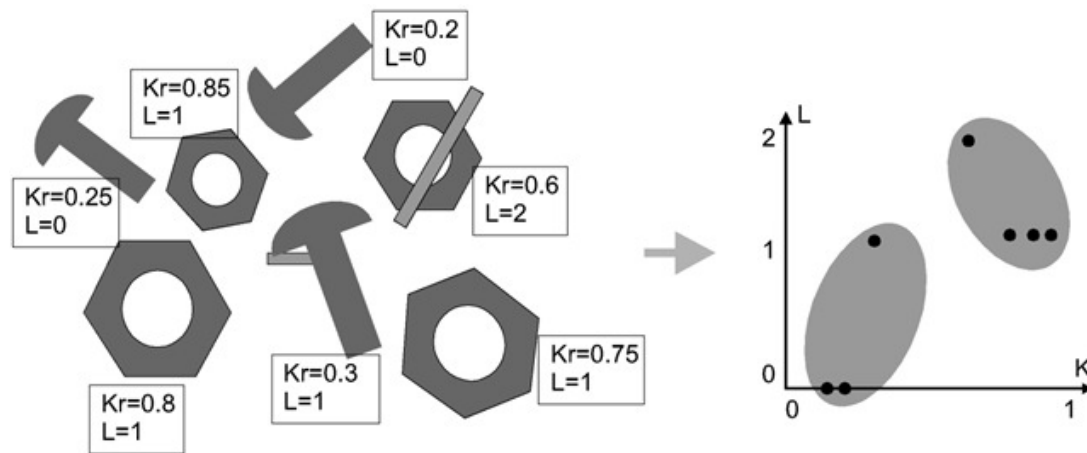
Beispiel:

- Geg.: 2-dim. Deskriptoren
- Zuordnung **aller Deskriptoren eines ungesehenen Segments  $s$** 
  - Durch Nächste-Nachbar-Zuordnung
  - über Euklid. Distanz
- BoW-Repräsentation  $BoW(s)$  ist das **normalisierte Histogramm der Auftreten der visuellen Wörter** aus dem Wörterbuch
  - ↪ s. Übungsaufgabe
  - ↪ Vektor mit Dimension  $k = 4$
- Klassifikation der Vektoren z.B. über MAP, das über die Trainingsmenge trainiert wurde.



# Principal Component Analysis (PCA)

- Geg.:  $n$  Segmente einer Trainingsmenge  $T$ , die jeweils über  $k$  skalare Merkmale beschrieben sind, die möglicherweise linear korreliert sind. Das heißt: die Kovarianz zw. Merkmalen  $z_j$  und  $z_l$  ist nicht Null. Dann gibt es Skalare  $s \neq 0$  und  $d$  mit  $z_l = s \cdot z_j + d$ .
- Die  $n$  Segmente entsprechen  $n$  Punkten im  $k$ -dim. Merkmalsraum
- Ziel der Hauptkomponentenanalyse\* ist die Transformation der  $n$  Punkte in einen  $m$ -dim. Merkmalsraum mit  $m < k$ , indem die Zahl der Merkmale durch Dekorrelation reduziert wird.



Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

\* Die **Hauptkomponentenanalyse** ist auch bekannt als **Hauptachsentransformation** (HAT) oder im Englischen als **Principal Component Analysis** (PCA). In der Bildverarbeitung wird die HAT auch als Karhunen-Loève-Transformation bezeichnet.



# PCA: Hauptkomponenten

---

- Die Transformation ist eine orthogonale Transformation, welche die  $k$  linear korrelierten Merkmale  $(z_1, \dots, z_k)$  in  $m$  linear dekorrelierte Merkmale  $(w_1, \dots, w_m)$  überführt. Die dekorrelierten Merkmale  $w_j$  werden als **Hauptkomponenten** bezeichnet.\*
- Die erste Hauptkomponente zeigt die größte Varianz, also die größte Variabilität der Merkmalswertebelegung. Alle folgenden Hauptkomponenten zeigen schrittweise abnehmende Varianz.
- Wegen der linearen Korrelation zwischen den Ursprungsmerkmalen  $z_i$  werden die Varianzen der letzten Hauptkomponenten  $w_j$  so gering sein, dass diese zur Charakterisierung der Segmente nur wenig beitragen. Daher kann auf sie verzichtet werden.

---

\* Zwei Verteilungen  $f, g$  sind linear dekorreliert, wenn die Kovarianz zw. ihren Merkmalen Null ist. Dann gibt es keine Skalare  $s \neq 0$  und  $d$  mit  $g = s \cdot f + d$ .

# PCA: Normierung

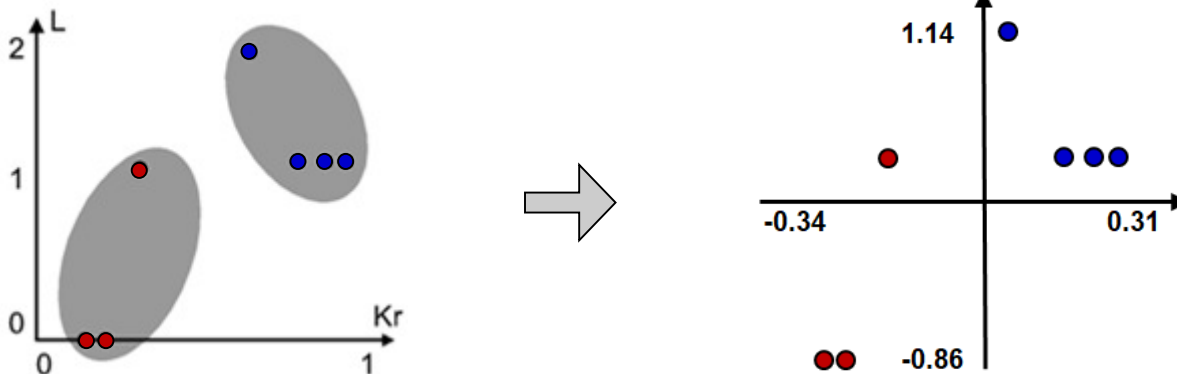
- Für die Transformation sind die Daten in ihrem Schwerpunkt zu zentrieren:

Für alle  $n$  Segmente  $s_i$  ( $i = 1, \dots, n$ ) mit  $k$  Merkmalen

Für jedes Merkmal  $z_j$  ( $j = 1, \dots, k$ )

1) Berechnung von Mittelwert  $\mu_j = \frac{1}{n} \sum_{i=1}^n z_j(s_i)$

2) Ersetzung von  $z_j(s_i)$  durch  $z_j(s_i) - \mu_j \rightarrow E(\mathbf{z}) = 0$



# PCA: Kovarianzmatrix

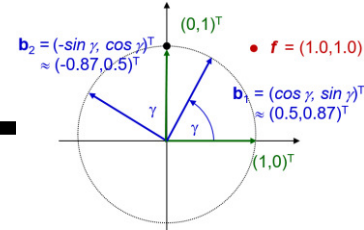
- Zur Herleitung der Hauptkomponenten in Reihenfolge der Varianzen wird aus den  $n$  Merkmalsvektoren  $\mathbf{z}_i = (z_{i1}, \dots, z_{ik})$  die **Kovarianzmatrix**  $\mathbf{C}(\mathbf{z})$  aufgestellt:

$$\mathbf{C}(\mathbf{z}) = \begin{pmatrix} \sigma_1^2 & \cdots & \sigma_1 \sigma_k \\ \vdots & \ddots & \vdots \\ \sigma_k \sigma_1 & \cdots & \sigma_k^2 \end{pmatrix}$$

- Die Diagonale von  $\mathbf{C}(\mathbf{z})$  zeigt die Varianzen  $\sigma_j^2$  der Merkmale  $z_j$  und die restlichen Einträge die Kovarianzen  $\sigma_j \sigma_{j'}$  zwischen Merkmalen  $z_j$  und  $z_{j'}$ :

$$\sigma_j \sigma_{j'} = \frac{1}{n} \sum_{i=1}^n (z_{ij} - \mu_j)(z_{ij'} - \mu_{j'}) \text{ mit } \mu_l = \frac{1}{n} \sum_{i=1}^n (z_{il})$$

# PCA: Dekorrelation



- Die Basis des Merkmalsraums wird so um den Ursprung rotiert, so dass die neuen Basisvektoren an den Richtungen der größten Varianzen im Merkmalsraum ausgerichtet werden.
- Diese Richtungen ergeben sich aus den Eigenvektoren der Kovarianzmatrix  $\mathbf{C}(\mathbf{z})$ . Die Transformationsmatrix  $\mathbf{E}$  setzt diese Rotation um und hat als Spalten die Eigenvektoren  $\mathbf{e}_1, \dots, \mathbf{e}_k$  der Kovarianzmatrix  $\mathbf{C}(\mathbf{z})$ . Die zugehörigen Eigenwerte  $\lambda_1, \dots, \lambda_k$  beschreiben die Varianz der Werte entlang der jeweiligen Achse.
- Die Dekorrelation der korrelierten Merkmale  $z_i$  erfolgt dann durch Transformation in den Vektorraum mit neuer Basis, so dass die Merkmale  $w_j$  dort linear unkorreliert sind, d.h.,  $\mathbf{C}(\mathbf{w})$  ist eine Diagonalmatrix. Die Varianzen von  $\mathbf{C}(\mathbf{w})$  sind die Eigenwerte  $\lambda_1, \dots, \lambda_k$  und es gibt keine Kovarianzen mehr:

$$(z_{i1}, \dots, z_{ik}) \begin{pmatrix} e_{11} & \cdots & e_{k1} \\ \vdots & \ddots & \vdots \\ e_{1k} & \cdots & e_{kk} \end{pmatrix} = (w_{i1}, \dots, w_{ik}) \rightsquigarrow \mathbf{C}(\mathbf{w}) = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_k \end{pmatrix}$$

# PCA: Transformation & Rücktransformation

---

- Charakt. Polynom:

Für Eigenvektoren  $\mathbf{e}_k$  und Eigenwerte  $\lambda_k$  einer quadr. Matrix  $\mathbf{M}$  gilt:

$$\mathbf{M} \times \mathbf{e}_k = \lambda_k \mathbf{e}_k \Leftrightarrow (\mathbf{M} - \lambda_k \mathbf{Id}) \times \mathbf{e}_k = \mathbf{0} \Rightarrow \det(\mathbf{M} - \lambda_k \mathbf{Id}) = 0.$$

- Die ursprünglichen korrelierten Merkmale  $z_j$  sind aus den neuen unkorrelierten Merkmalen  $w_j$  rekonstruierbar nach \*

$$\mathbf{z} = \mathbf{E}(\mathbf{z}) + \mathbf{w} \times \mathbf{E}^T$$

mit dem Erwartungswert  $\mathbf{E}(\mathbf{z})$  über alle Vektoren  $\mathbf{z}$

---

\* Matrix  $\mathbf{E}$  ist orthonormal  $\rightarrow$  transponierte Matrix = inverse Matrix:  $\mathbf{E}^{-1} = \mathbf{E}^T$

# PCA: Rücktransformation

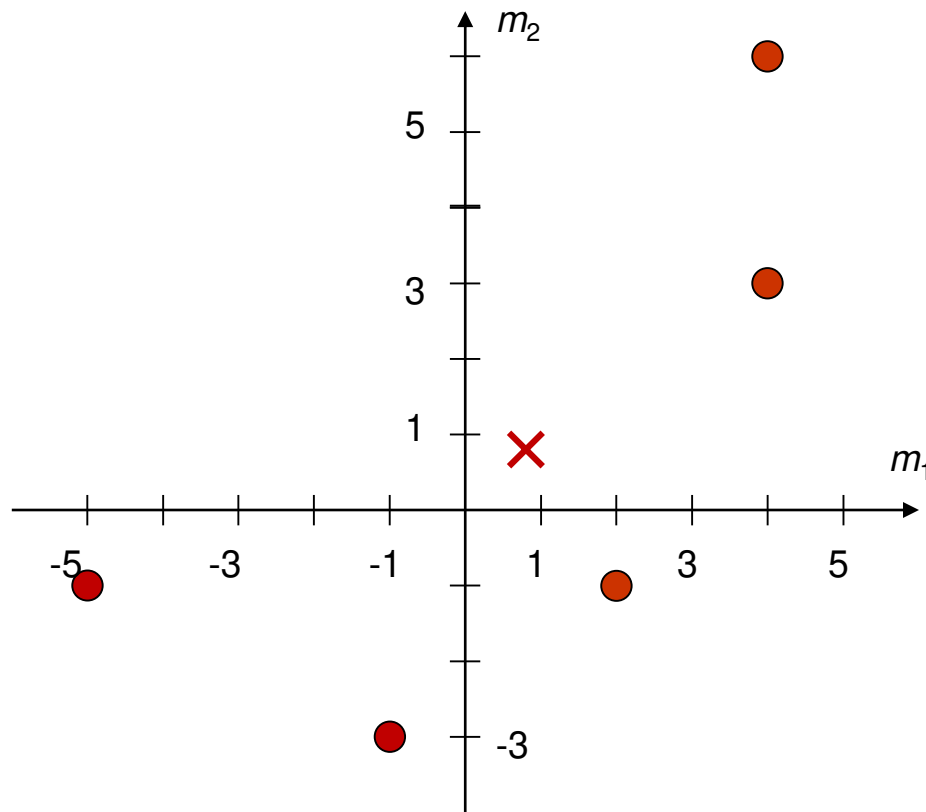
---

- Die **ursprünglichen korrelierten Merkmale**  $z_i$  müssen aber nicht aus allen **neuen unkorrelierten Merkmalen**  $w_j$  nach  $\mathbf{z} = \mathbf{E}(\mathbf{z}) + \mathbf{w} \times \mathbf{E}^T$  rekonstruiert werden.
- Der Grund: wegen der ursprünglichen Kohärenz zwischen den Ursprungsmerkmalen  $z_i$  werden die Varianzen etlicher neuer Merkmale  $w_j$  so gering sein, dass bei entsprechender Umordnung der Indizes die folg. Näherung ausreicht:

$$\mathbf{z} = (z_1, \dots, z_k) = \mathbf{E}(\mathbf{z}) + (w_1, \dots, w_m) \begin{pmatrix} e_{11} & \cdots & e_{1k} \\ \vdots & \ddots & \vdots \\ e_{m1} & \cdots & e_{mk} \end{pmatrix}$$

- Parameter  $m \leq k$  steht für die Zahl der Merkmale  $w_j$  mit *hinreichend großen Varianzen* bzw. Eigenwerten. Diese  $m$  Komponenten heißen **Variationsmodi** (*Modes of Variation*).

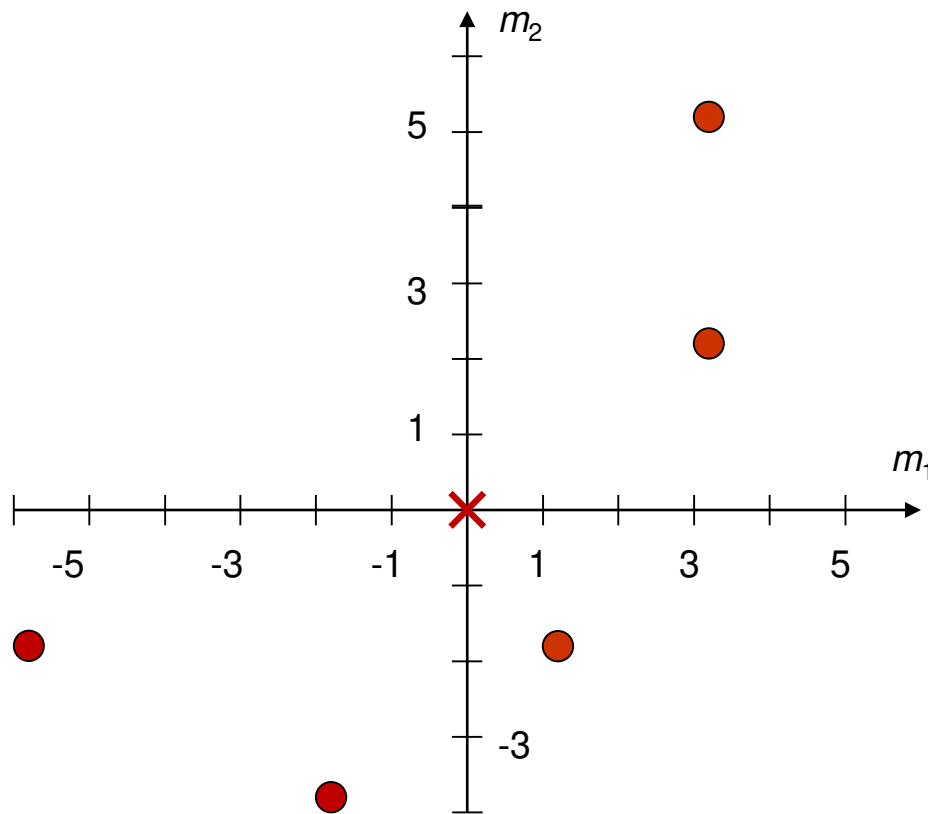
# PCA: Beispiel 1 (Daten)



Punkte  $\mathbf{z}_j$  in 2D-Merkmalsraum:  
 $(-5,-1), (-1,-3), (2,-1), (4,3), (4,6)$

→ Erwartungswert  $E(\mathbf{z}) = (0.8, 0.8)$

# PCA: Beispiel 1 (Zentrierung)



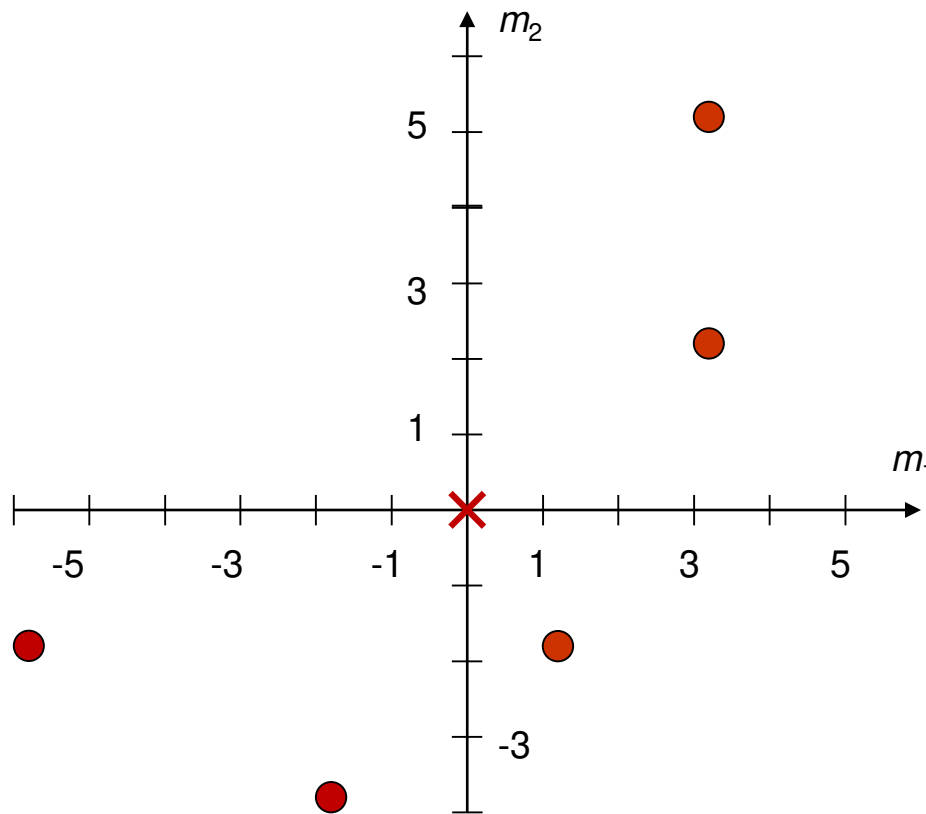
Punkte  $\mathbf{z}_j$  in 2D-Merkmalsraum:  
 $(-5,-1), (-1,-3), (2,-1), (4,3), (4,6)$

→ Erwartungswert  $E(\mathbf{z}) = (0.8, 0.8)$

→ Zentriert in  $E(\mathbf{z})$ :  
 $(-5.8, -1.8), (-1.8, -3.8), (1.2, -1.8),$   
 $(3.2, 2.2), (3.2, 5.2)$



# PCA: Beispiel 1 (Kovarianzmatrix)



Punkte  $\mathbf{z}_j$  in 2D-Merkmalsraum:  
 $(-5,-1), (-1,-3), (2,-1), (4,3), (4,6)$

→ Erwartungswert  $E(\mathbf{z}) = (0.8, 0.8)$

→ Zentriert in  $E(\mathbf{z})$ :  
 $(-5.8, -1.8), (-1.8, -3.8), (1.2, -1.8),$   
 $(3.2, 2.2), (3.2, 5.2)$

→ Kovarianzmatrix

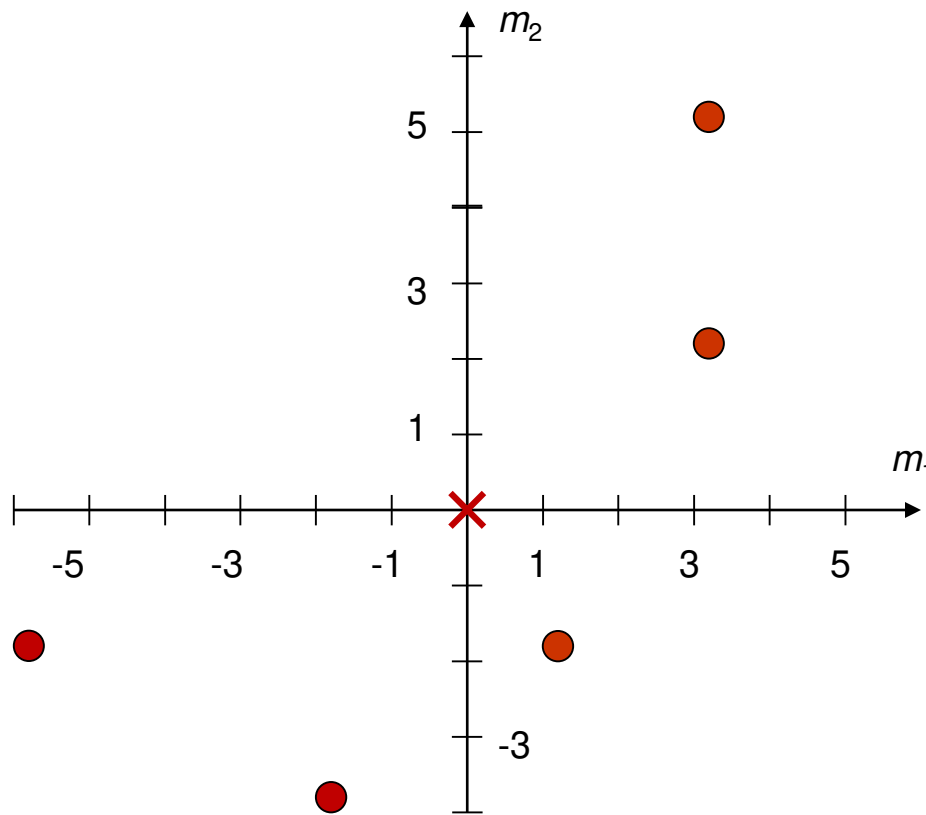
s. Folie 27

$$C(\mathbf{z}) = \begin{pmatrix} 11.76 & 7.76 \\ 7.76 & 10.56 \end{pmatrix}$$

→  $\sigma_{12} = 7.76$

korrelierte Merkmale

# PCA: Beispiel 1 (Eigenwerte)



Punkte  $\mathbf{z}_j$  in 2D-Merkmalsraum:

$(-5, -1), (-1, -3), (2, -1), (4, 3), (4, 6)$

→ Erwartungswert  $E(\mathbf{z}) = (0.8, 0.8)$

→ Zentriert in  $E(\mathbf{z})$ :

$(-5.8, -1.8), (-1.8, -3.8), (1.2, -1.8),$   
 $(3.2, 2.2), (3.2, 5.2)$

→ Kovarianzmatrix

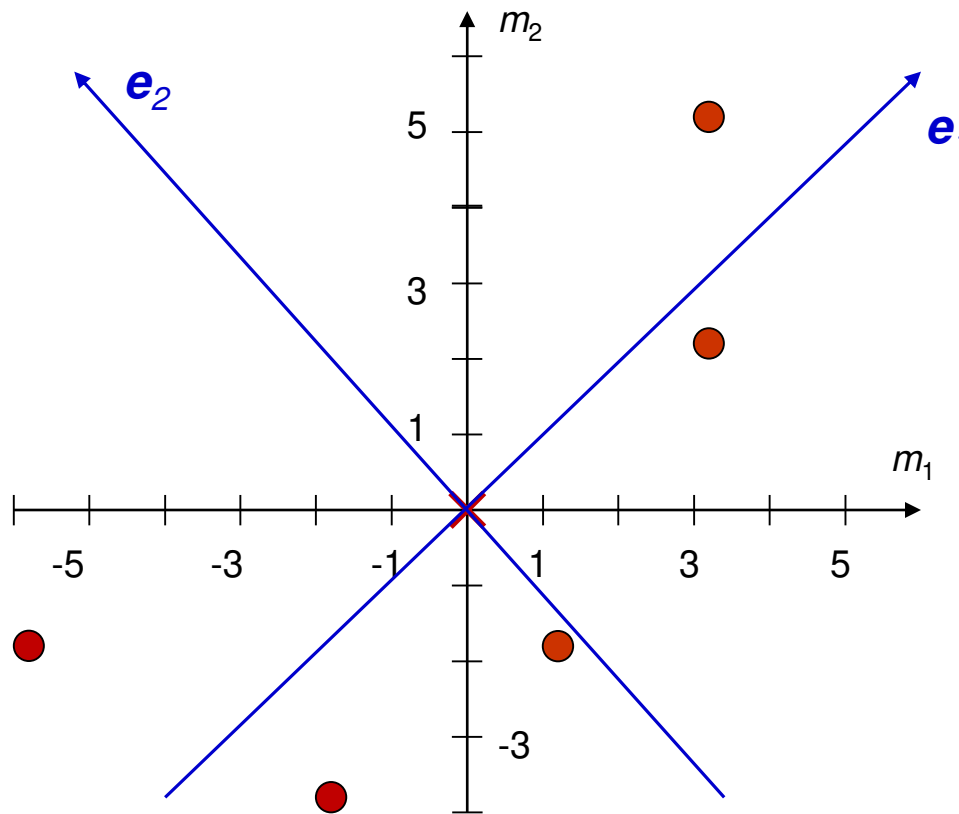
$$C(\mathbf{z}) = \begin{pmatrix} 11.76 & 7.76 \\ 7.76 & 10.56 \end{pmatrix}$$

s. Folie 29

→ Eigenwerte über char. Polynom:  $\begin{vmatrix} 11.76 - \lambda & 7.76 \\ 7.76 & 10.56 - \lambda \end{vmatrix} = 0$

→ Eigenwerte  $\lambda_1 = 18.943, \lambda_2 = 3.377$

# PCA: Beispiel 1 (Eigenvektoren)



Punkte  $\mathbf{z}_j$  in 2D-Merkmalsraum:

$(-5,-1), (-1,-3), (2,-1), (4,3), (4,6)$

→ Erwartungswert  $E(\mathbf{z}) = (0.8, 0.8)$

→ Zentriert in  $E(\mathbf{z})$ :

$(-5.8, -1.8), (-1.8, -3.8), (1.2, -1.8),$   
 $(3.2, 2.2), (3.2, 5.2)$

→ Kovarianzmatrix

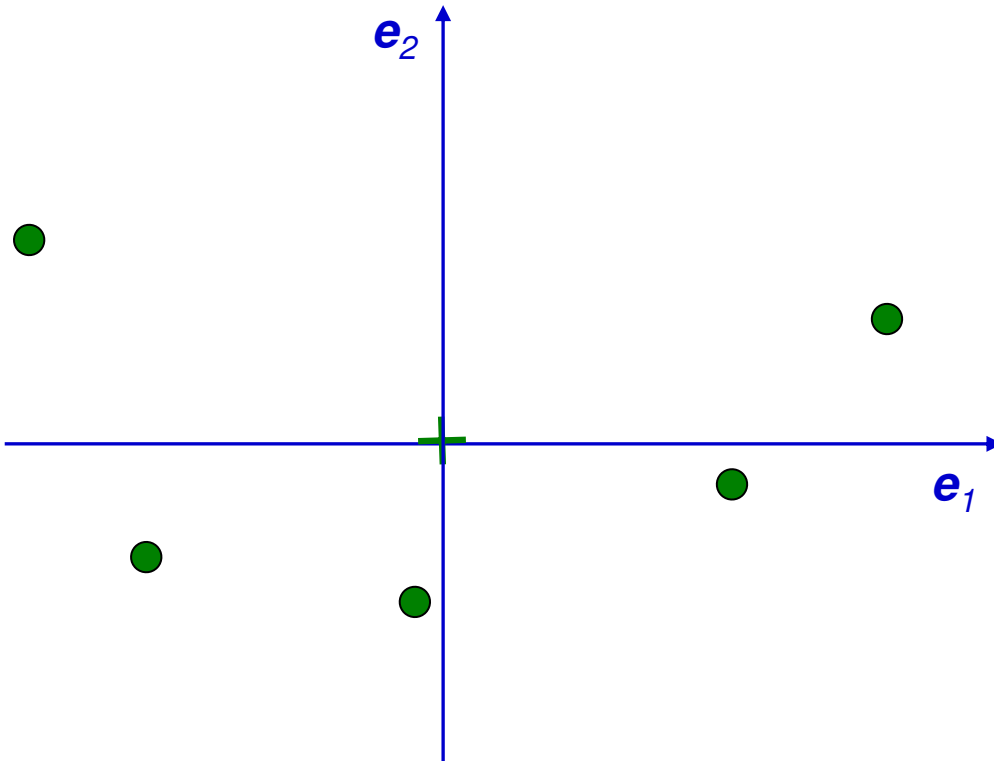
$$\mathbf{C}(\mathbf{z}) = \begin{pmatrix} 11.76 & 7.76 \\ 7.76 & 10.56 \end{pmatrix}$$

→ Eigenwerte  $\lambda_1 = 18.943, \lambda_2 = 3.377$

→ Eigenvektoren über  $\mathbf{C} \times \mathbf{e}_i = \lambda_i \cdot \mathbf{e}_i$ :  $\mathbf{e}_1 = \begin{pmatrix} 0.734 \\ 0.679 \end{pmatrix}, \mathbf{e}_2 = \begin{pmatrix} -0.679 \\ 0.734 \end{pmatrix}$

s. Folie 29

# PCA: Beispiel 1 (Transformation)



Punkte  $\mathbf{z}_j$  zentriert in  $E(\mathbf{z})$ :

$(-5.8, -1.8), (-1.8, -3.8), (1.2, -1.8),$   
 $(3.2, 2.2), (3.2, 5.2)$

Transformationsmatrix

$$\mathbf{E} = \begin{pmatrix} 0.734 & -0.679 \\ 0.679 & 0.734 \end{pmatrix}$$

→ Transformation:

$$\mathbf{z}_j \times \mathbf{E} = \mathbf{w}_j$$

s. Folie 28

→ Punkte  $\mathbf{w}_j$ :

→ s. Übungsaufgabe

# PCA: Beispiel 1 (Rücktransformation)

Punkte  $\mathbf{w}_j$ :

→ s. Übungsaufgabe

Inverse Transformationsmatrix

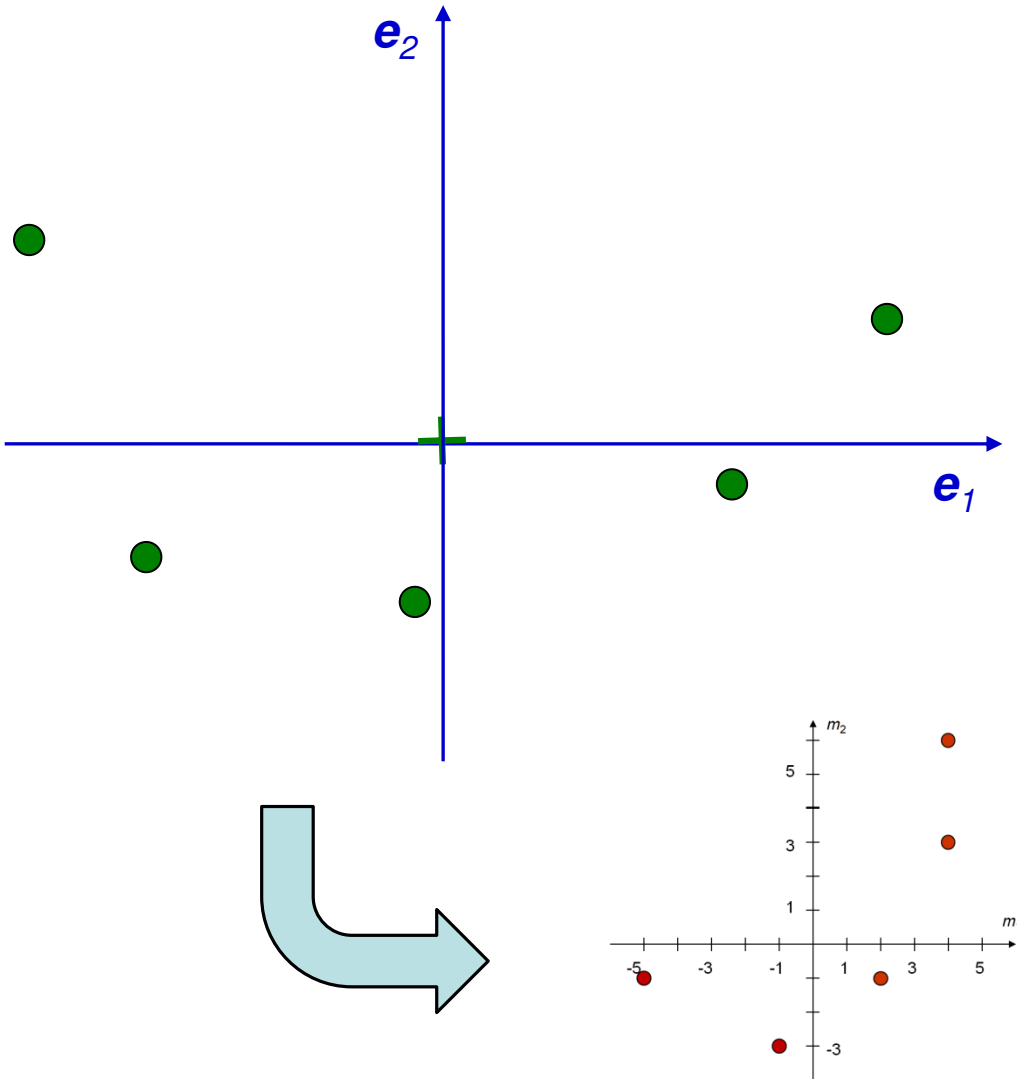
$$\mathbf{E}^{-1} = \mathbf{E}^T = \begin{pmatrix} 0.734 & 0.679 \\ -0.679 & 0.734 \end{pmatrix}$$

→ Rücktransformation:

$$\mathbf{z} = \mathbf{E}(\mathbf{z}) + \mathbf{w} \times \mathbf{E}^T$$

mit  $\mathbf{E}(\mathbf{z}) = (0.8, 0.8)$

→ Punkte  $\mathbf{z}_j$  in 2D-Merkmalsraum:  
 $(-5, -1), (-1, -3), (2, -1), (4, 3), (4, 6)$



# PCA: Beispiel 1 (Rücktransformation)

Punkte  $\mathbf{w}_j$ :

→ s. Übungsaufgabe

Inverse Transformationsmatrix

$$\mathbf{E}_m^{-1} = \mathbf{E}_m^T = \begin{pmatrix} 0.734 & 0.679 \end{pmatrix}$$

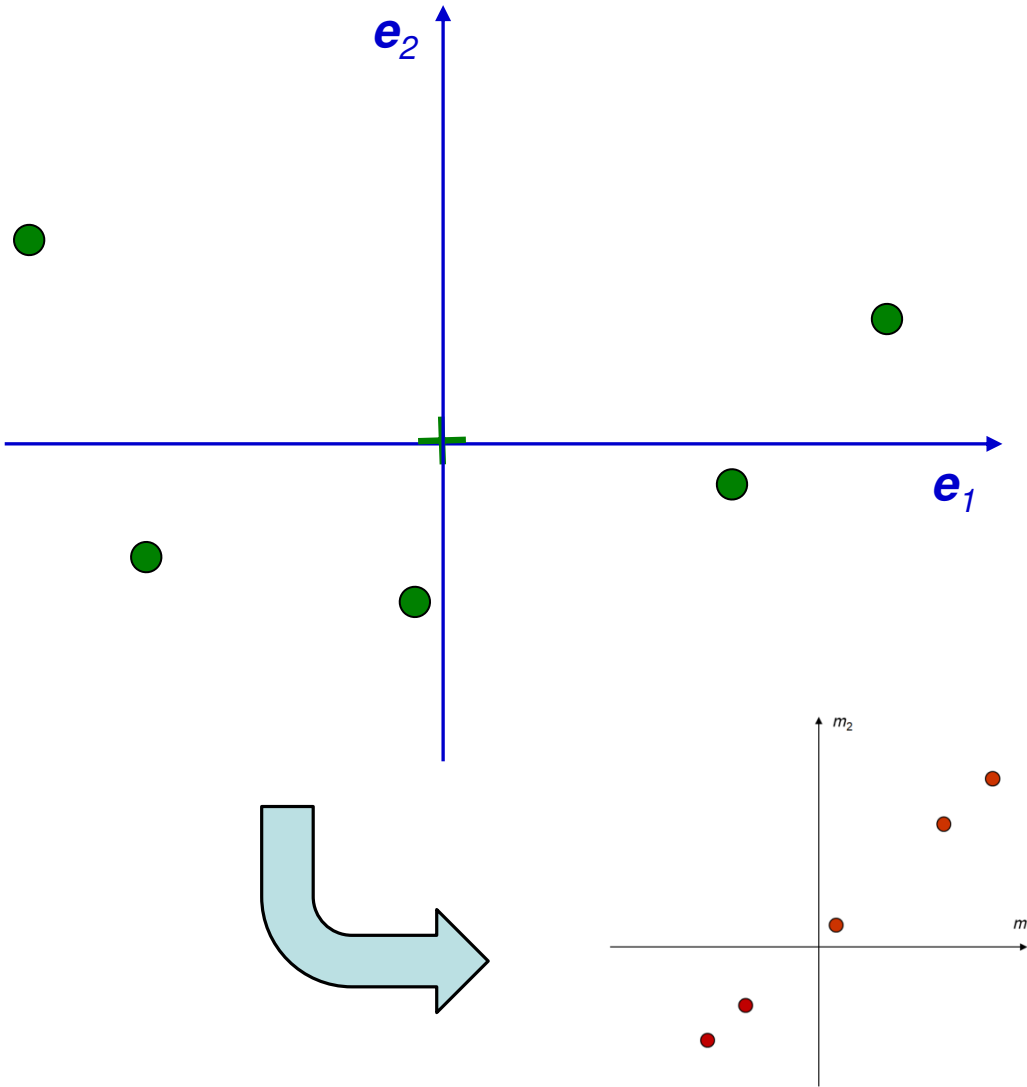
→ Rücktransformation:

$$\mathbf{z} = \mathbf{E}(\mathbf{z}) + \mathbf{w}_m \times \mathbf{E}_m^T$$

mit  $\mathbf{E}(\mathbf{z}) = (0.8, 0.8)$

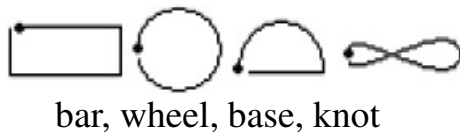
→ Punkte  $\mathbf{z}_j$  in 2D-Merkmalsraum:

→ s. Übungsaufgabe



# PCA: Beispiel 2 (skizzenbasierte Biometrie)

- Das Beispiel behandelt eine skizzenbasierte biometrische Erkennung
- Aufgaben (s. Abbildungen):



task	description	objects
1	Draw three connected wheels of different sizes	3
2	Draw 3 connected bars one bar is bigger than the others Connect the bars to 3 knots	6
3	Draw 2 connected wheels one wheel is bigger than the other Connect the wheels to a small bar Connect bar to a big base	4
4	draw Task 2 and task 3 connect them with a knot	11

	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9	user 10
task 1										
task 2										
task 3										
task 4										

# PCA: Beispiel 2 (Experimente)

---

## Durchführung und Ergebnisse:

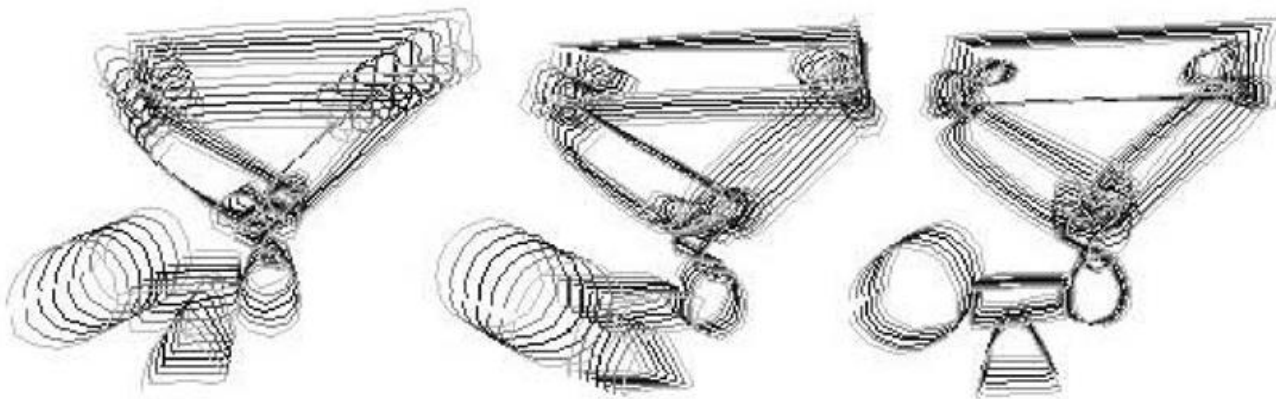
- Each user was given 4 tasks ( $t_1, \dots, t_4$ ) of increasing complexity to complete in his way [...]. Each stroke was sampled by 16 points.
- For every sketch, the number of principal components was set to explain 95% of the samples. The number of principal components ranges between 10 for task 1 and 15 for task 4.
- The experiments were conducted on 10 users ( $u_1, \dots, u_{10}$ ). Each user sketched each task 30 times. For every user task, 20 randomly selected samples were used for training and the remaining 10 were used for testing. The tests were cross validated 10 times and averaged. [...] the average recognition error decreases as the complexity of the structures increases.
- Task 4 consisting of 11 objects had no error within this laboratory test setup.



# PCA: Beispiel 2 (Variationsmodi)

---

- Die ersten drei Variationsmodi die skizzenbasierte biometrische Erkennung:



Beispiel und Abbildungen: S. Al-Zubi, A. Brömme, K. Tönnies. Using an Active Shape Structural Model for Biometric Sketch Recognition. In *Proc. DAGM*, 187–195, 2003.

# PCA: Beispiel 2 (Wahl der Variationsmodi)

---

- Die Zahl  $M$  der zu berücksichtigenden Variationsmodi wird i.A. abhängig von der durch die entspr.  $M$  Merkmale repräsentierten Varianz in den Daten gewählt.
- Beispiel:
  - 100 gesampelte Punkte, also 200 Merkmale  $z_i$  (den x-, y-Koordinaten entsprechend)
  - 30% der transformierten Merkmale  $w_i$  erklären 95% der Datenvariation bei:

$$\sum_{j=0}^m \lambda_j \geq 0.95 \cdot \sum_{j=0}^k \lambda_j, k = 200, m = 30$$

# Hierarchical and trained Features

## ■ Deep Learning: Representations are hierarchical and trained



- *Deep Learning* reduziert *Feature Engineering* auf das Lernen von hierarchischen Merkmalsrepräsentationen
- Das Lernen der Merkmalsrepräsentationen erfolgt alleine durch Trainingsmengen von unbearbeiteten Bilddaten und ihren Zuordnungen zu Objektklassen.
- Im optimalen Fall ergibt sich ein vollständiges *End-to-End Learning*: durch annotierte Bilddaten werden sowohl hierarchische Merkmalsrepräsentationen als auch die Klassifikation erlernt.

→ Folgevorlesung

# Zusammenfassung (1)

---

- Handcrafted Features
  - Lokale Merkmale (z.B. SIFT) werden an verschiedenen Orten des Segments ermittelt
  - Globale Merkmale beziehen sich auf das gesamte Segment
    - Local Binary Patterns (LBP) ist ein histogrammbasierter Deskriptor, der lokale Muster durch Vergleich der Intensität eines Pixels mit denen seiner Nachbarpixel kodiert.
    - Momente sind ein allg. Ansatz um bestimmte gewichtete Mittelwerte aus den Intensitäten der Pixel eines Segments zu rechnen. Eigenschaften von Segmenten, die durch Momente berechnet werden können, sind u.a. Fläche, Schwerpunkt, Ausrichtung, Varianz, Schiefe und Wölbung.  
Normalisierte zentrale Momente sind translations- und skalierungs-invariant. Die Hu-Momente sind zudem rotationsinvariant.

# Zusammenfassung (2)

---

- *Mid-level features* werden auf der Basis von *handcrafted Features* erzeugt
  - *Bag of Visual Words* beschreibt Segmente durch Histogramme der Auftreten von visuellen Wörtern aus dem Wörterbuch
  - Die Hauptachsentransformation ist eine orthogonale Transformation, welche  $k$  linear korrelierte Merkmale  $(z_1, \dots, z_k)$  in  $m$  linear dekorrelierte Merkmale  $(w_1, \dots, w_m)$  überführt.

Wegen der ursprünglichen Kohärenz zwischen den Merkmalen  $z_i$  werden Varianzen vieler Merkmale  $w_j$  so gering sein, dass bei entsprechender Umordnung der Indizes eine Beschreibung der Daten durch Merkmale  $w_j$  mit großen Varianzen ausreicht, also  $m < k$ .
- *Hierarchical and trained Features* werden durch *Deep Learning* alleine durch Trainingsmengen von unbearbeiteten Bilddaten und ihren Zuordnungen zu Objektklassen erzeugt.

# Anhang: *k*-Means (1)

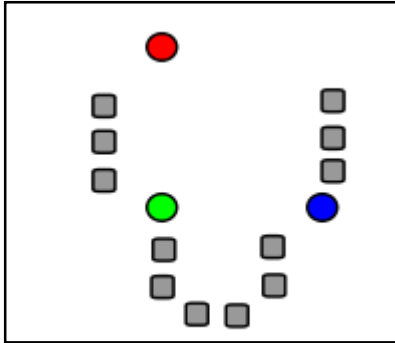
---

## Der *k*-Means-Algorithmus

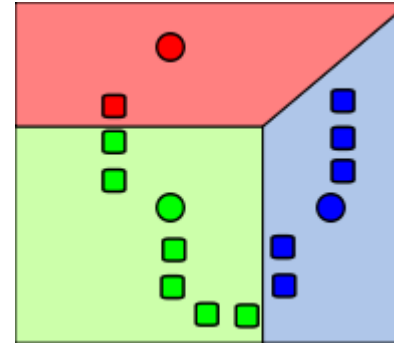
- erzeugt eine *a priori* vorgegebene Anzahl von *k* Clustern aus einer Menge von Datensätzen
- ist eine der meist verwendeten Techniken zur Clusteranalyse, da er die Zentren der Cluster schnell findet
- zeichnet sich durch große Einfachheit aus

# Anhang: *k*-Means (2)

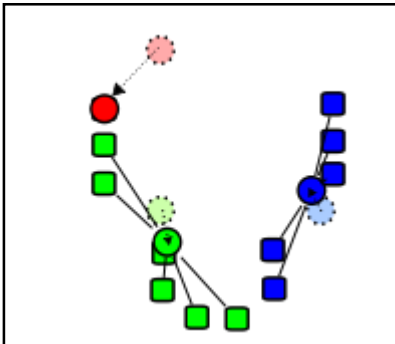
Beispiel für Datensätze mit zwei beschreibenden Attributen:



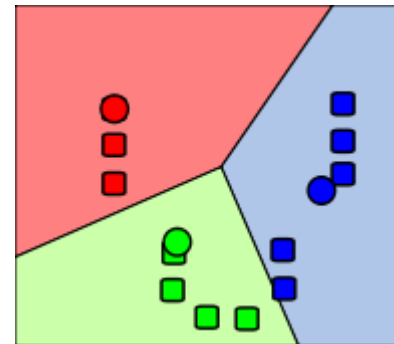
*k* = 3 initiale Zentren  
zufällig positioniert



*k* = 3 Cluster mit Zuordnung der Datenpunkte zu den nächsten Zentren



Neuberechnung der Zentren



Wiederholte Neuberechnungen von Zentren und Cluster-Zuordnungen



## Anhang: *k*-Means (3)

---

Die Schritte vom *k*-Means-Algorithmus:

- (0) Vor Ausführung ist die Anzahl *k* der zu ermittelnden Cluster festzulegen
- (1) Die *k Cluster-Schwerpunkte* werden zufällig im Datenraum verteilt
- (2) **Datenzuordnung**: Jeder Datensatz wird demjenigen Cluster zugeordnet, dessen Schwerpunkt ihm am nächsten liegt \*
- (3) **Schwerpunktberechnung**: Nach der Neuordnung der Datensätze werden die Schwerpunkte aller Cluster neu berechnet \*
- (4) Gehe zu Schritt (2), bis
  - die Positionen der Schwerpunkte stabil bleiben (d.h. keine Neuverteilung der Datensätze erfolgt) oder
  - eine festgelegte maximale Zahl von Iterationen erreicht wird

---

\* Unter Verwendung einer Distanzfunktion wie z.B. der Euklid. Distanz.



## Anhang: *k*-Means (4)

---

Die zentralen Schritte des ***k*-Means-Algorithmus** konkreter:

(1) **Initialisierung**: zufällige Vorgabe von  $k$  Schwerpunkten  $\mathbf{m}_1^{(1)}, \dots, \mathbf{m}_k^{(1)}$

(2) **Neuzuordnung der Datensätze**  $\mathbf{x}_j$  zu dem Cluster  $\mathbf{S}_i^{(t)}$  in Iteration  $t$ :

$$\mathbf{S}_i^{(t)} = \{ \mathbf{x}_j \text{ mit } \|\mathbf{x}_j - \mathbf{m}_i^{(t)}\| \leq \|\mathbf{x}_j - \mathbf{m}_{i'}^{(t)}\| \forall i' \in \{1, \dots, k\} \setminus \{i\} \}$$

(3) **Neuberechnung der Schwerpunkte**:

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|\mathbf{S}_i^{(t)}|} \cdot \sum_{\mathbf{x}_j \in \mathbf{S}_i^{(t)}} \mathbf{x}_j$$

(4) **Terminierung**, wenn keine Neuzuordnungen in (2)

Der Algorithmus versucht also, die **Kompaktheit** aller Cluster  $\mathbf{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$  zu **maximieren**:

$$\arg \min_{\mathbf{S}} = \sum_{i=1}^k \sum_{\mathbf{x}_j \in \mathbf{S}_i} \|\mathbf{x}_j - \mathbf{m}_i\|$$

# Anhang: Klassifikation mit $k$ -Nearest Neighbor

---

$k$ -NN: eine der einfachsten Methoden des Maschinellen Lernens

- Training: einfach Speicherung der Merkmalsvektoren aller annotierten Trainingbeispiele
- Klassifikation: ein ungesehener Merkmalsvektor (*Query*) wird der Klasse zugewiesen, welche die größte Anzahl der Objekte unter den  $k$  nächsten Nachbarn hat.
- Verschiedene Abstandsmaße sind möglich: Euklidische Distanz, Manhattan-Distanz, etc.

