

## Bildverarbeitung: Bildmerkmale

Was sind Bildmerkmale?

Wofür werden sie verwendet?

- Qualitative Aussagen über Bildausschnitte – Erkennung.
  - Wiederfinden ähnlicher Bildausschnitte – Matching.
- Beides bedeutet Bildmerkmale vergleichen zu können.

Probleme – hohe Dimension, sehr viele denkbare Merkmale  
⇒ Merkmalselektion (Auswahl)

Techniken – Hauptkomponentenanalyse, Clusterung, Momente.

---

I. Das einfachst mögliche – der Bildmerkmal ist der Bildausschnitt (als Vektor) selbst.

II. Die Werte als Merkmale zu verstehen,  
die bei der Ermittlung der besondere Punkte berechnet werden:

z.B. Eigenwerte (ggf. auch Eigenvektoren) bei Harris-Detektor,

bei MSER werden Regionen ermittelt,

Bildmerkmale – Farbcharakteristika (mittlere Farbe, Streuung, Stabilität ...),  
geometrische Charakteristika (Größe, Umfang, Krümmung, Ausdehnung, Konvexität ...)

- 1) Interessante Punkte werden gesucht
- 2) Bildfragmente werden normalisiert (meist affine, seltener projektive Verzerrung)  
z.B. mithilfe der Autokorrelationsfunktion
- 3) Der Deskriptor ist:

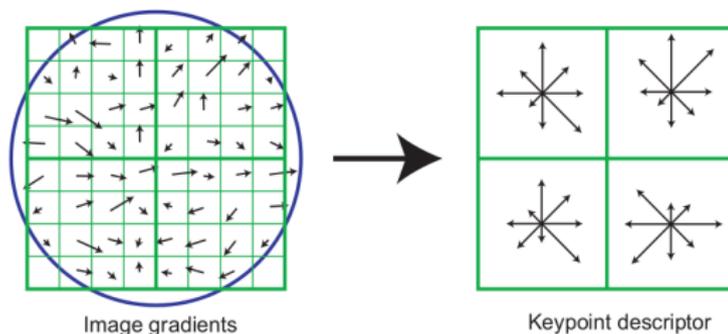


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

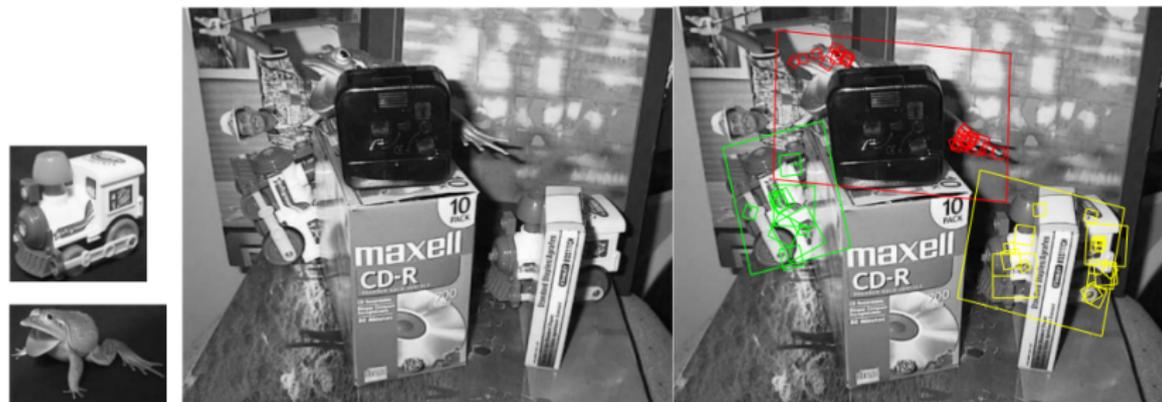
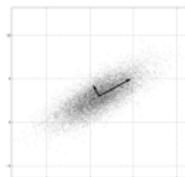


Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

Es ist bisher nicht klar, warum der SIFT in der Praxis so gut funktioniert.

# Hauptkomponentenanalyse (PCA)

Die Vektoren (Bildausschnitte) werden in einem anderen Basis dargestellt.



Annahme: die „Richtungen“ kleiner Varianzen entsprechen dem Rauschen und können vernachlässigt werden.

Die Idee: den Merkmalsraum auf einen linearen Unterraum projizieren so, dass die Varianzen im Unterraum so groß wie möglich sind.

Einfachheit halber – die Daten sind bereits zentriert, der Unterraum ist eindimensional, d.h. durch einen Vektor  $e$  mit  $\|e\|^2 = 1$  angegeben. Projektion eines  $x$  auf  $e$  ist  $\langle x, e \rangle$ .

$$\sum_l \langle x^l, e \rangle^2 \rightarrow \max_e \quad \text{s.t. } \|e\|^2 = 1$$

Lagrange Funktion:

$$\sum_l \langle x^l, e \rangle^2 + \lambda (\|e\|^2 - 1) \rightarrow \min_{\lambda} \max_e$$

Ableitung:

$$\sum_l 2 \langle x^l, e \rangle \cdot x^l + 2\lambda e = 2e \sum_l x^l \otimes x^l + 2\lambda e = 0$$
$$e \cdot \text{cov} = \lambda e$$

→  $e$  ist Eigenvektor der Kovarianzmatrix,  $\lambda$  ist der entsprechende Eigenwert.

Welchen Eigenvektor soll gewählt werden? Die mit einem  $\lambda$  erreichte Varianz ist

$$\sum_l \langle x^l, e \rangle^2 = e \cdot \sum_l x^l \otimes x^l \cdot e = e \cdot cov \cdot e = \|e\|^2 \cdot \lambda = \lambda$$

→ wähle den Eigenvektor zum größten Eigenwert.

Ähnliche Vorgehensweise: den Merkmalsraum auf einen Unterraum projizieren so, dass die summarische quadratische Abweichung der Datenpunkte von entsprechenden Projektionen so klein wie möglich ist → Approximation. Das Ergebnis ist dasselbe.

Zusammenfassend:

- 1) Berechne die Kovarianzmatrix der Daten  $cov = \sum_l x^l \otimes x^l$
- 2) Suche alle Eigenwerte und Eigenvektoren
- 3) Ordne sie (fallend) nach Eigenwerten
- 4) Wähle  $m$  Eigenvektoren zu  $m$  größten Eigenwerten
- 5) Die  $n \times m$  Projektionsmatrix besteht aus  $m$  Spalten, die jeweils die gewählten Eigenvektoren sind.

---

Ist die Projektion auf einen linearen Unterraum gut? Alternativen?

# Funktionsräume

Bilder sind Funktionen über einen kontinuierlichen Definitionsbereich  $I : D \subset \mathbb{R}^2 \rightarrow C$

	Vektor $v = (v_1, v_2 \dots v_n)$	Funktion $f(x), x \in \mathbb{R}$
Definitionsbereich	$\{1, 2 \dots n\}$	$\mathbb{R}$
Abbildung	$\{1, 2 \dots n\} \rightarrow \mathbb{R}$	$\mathbb{R} \rightarrow \mathbb{R}$
Raum	$\mathbb{R}^n$	$\mathbb{R}^\infty$
Skalarprodukt	$\langle u, v \rangle = \sum_i u_i v_i$	$\int f(x)g(x) dx$
Länge	$\sum_i v_i^2 = \langle v, v \rangle$	$\int f(x)^2 dx$

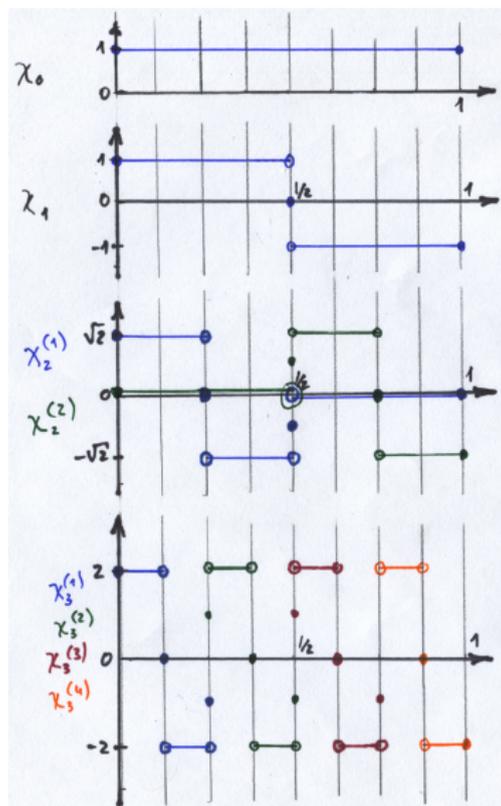
Die Aufgabe ist, eine gegebene Funktion  $f(x)$  auf Basisfunktionen zu zerlegen:

$$f(x) = \int_y v(x, y) \lambda(y) dy$$

Beispiel: Basisfunktionen  $\cos(yx)$  und  $\sin(yx) \rightarrow$  Fourier-Transformation

# Basis-Funktionen von Haar (1909)

Eindimensional:



$$\chi_0(s) = 1, 0 \leq s \leq 1$$

$$\chi_1(s) = \begin{cases} 1 & 0 \leq s < 1/2 \\ -1 & 1/2 < s \leq 1 \end{cases}$$

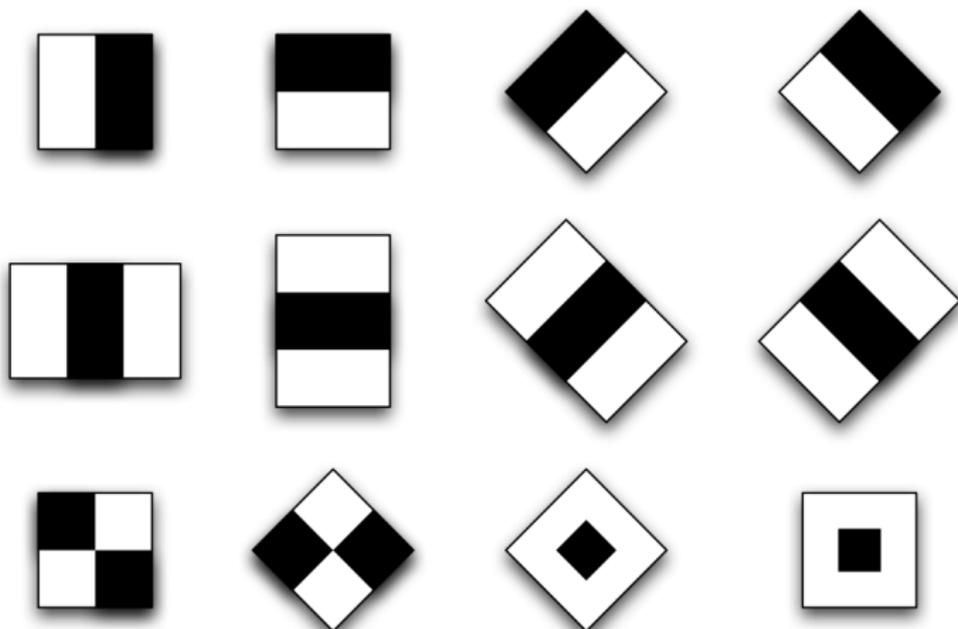
$$\chi_2^{(1)}(s) = \begin{cases} \sqrt{2} & 0 \leq s < 1/4 \\ -\sqrt{2} & 1/4 < s < 1/2 \\ 0 & \text{sonst} \end{cases}$$

$$\chi_2^{(2)}(s) = \begin{cases} \sqrt{2} & 1/2 < s < 3/4 \\ -\sqrt{2} & 3/4 < s \leq 1 \\ 0 & \text{sonst} \end{cases}$$

$$\chi_3^{(1)}, \chi_3^{(2)}, \chi_3^{(3)}, \chi_3^{(4)}$$

usw.

Filter:  $\pm 1$  (Schwarz/Weiß)



Können mithilfe des Integralbildes sehr effizient berechnet werden!!!

Wie sind zwei Bildfragmente (Bild  $I_1$  an der Stelle  $p_1 = (i_1, j_1)$  mit dem Bild  $I_2$  an der Stelle  $p_2 = (i_2, j_2)$ ) zu vergleichen? Zum Beispiel mit dem quadratischen Abstand

$$D(I_1(p_1), I_2(p_2)) = \sum_{p' \in W} (I_1(p_1 + p') - I_2(p_2 + p'))^2$$

Weitere Schritte – gewisse Farbtransformationen zu erlauben.

Zum Beispiel: Die Färbungen dürfen sich um eine additive Konstante unterscheiden (unterschiedliche Helligkeit) – nur der „Rest“ wird beurteilt:

$$D_a(I_1(p_1), I_2(p_2)) = \min_{C_a} \sum_{p' \in W} (I_1(p_1 + p') + C_a - I_2(p_2 + p'))^2$$

Ableitung nach  $C_a$ , auf Null setzen, auflösen  $\Rightarrow$

$$C_a^* = \frac{1}{|W|} \sum_{p' \in W} (I_2(p_2 + p') - I_1(p_1 + p'))$$

$$D_a(I_1(p_1), I_2(p_2)) = D(I_1(p_1), I_2(p_2)) - C_a^{*2} = D(\tilde{I}_1(p_1), \tilde{I}_2(p_2))$$

Mittelwertfreie Färbungen.

Weitere Invarianten:

Skalierung (Kontrast):  $I_1(p_1 + p')$  wird zu  $C_m \cdot I_1(p_1 + p') + C_a$

⇒ Korrelationskoeffizient – mittelwertfreie Färbungen, normiert auf ihre Varianzen.

Geometrische Transformationen:  $I_1(p_1 + p')$  wird zu  $I_1(Tr(p_1 + p'))$

usw.

---

Zurück zu Bilddeskriptoren:

Idee (Annahme) – Die Menge aller Bildfragmente besteht aus Teilmengen, jede Teilmenge hat einen Repräsentant – ein idealer Muster, alle Bildfragmente sind verrauschte Varianten der Repräsentanten.

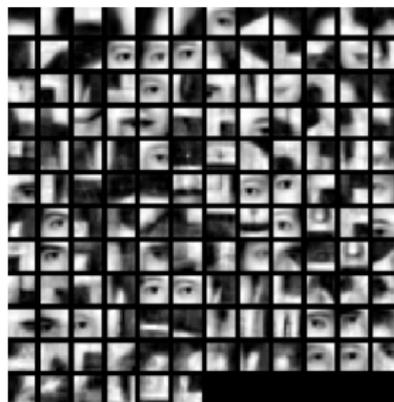
Der Merkmal eines Bildfragmentes ist der ähnlichste Repräsentant.

Man finde die Repräsentanten anhand einer Lernstichprobe von Bildfragmenten  
⇒ Clusterung

- Finde interessante Punkte in Bildern einer Datenbank
- Betrachte die entsprechenden Bildfragmente als Vektoren
- Clustere sie (Distanzmaß aus Block Matching)



Bilddatenbank



⇒

Visuelle Wörter

Jedem Bildausschnitt wird das Wort zugeordnet, zu welchem er am nächsten liegt.  
Bildmerkmal – Nummer des Wortes