

# **5. Objekterkennung in Bildern und Videos**

## **Videoanalyse**

Stephan Kopf

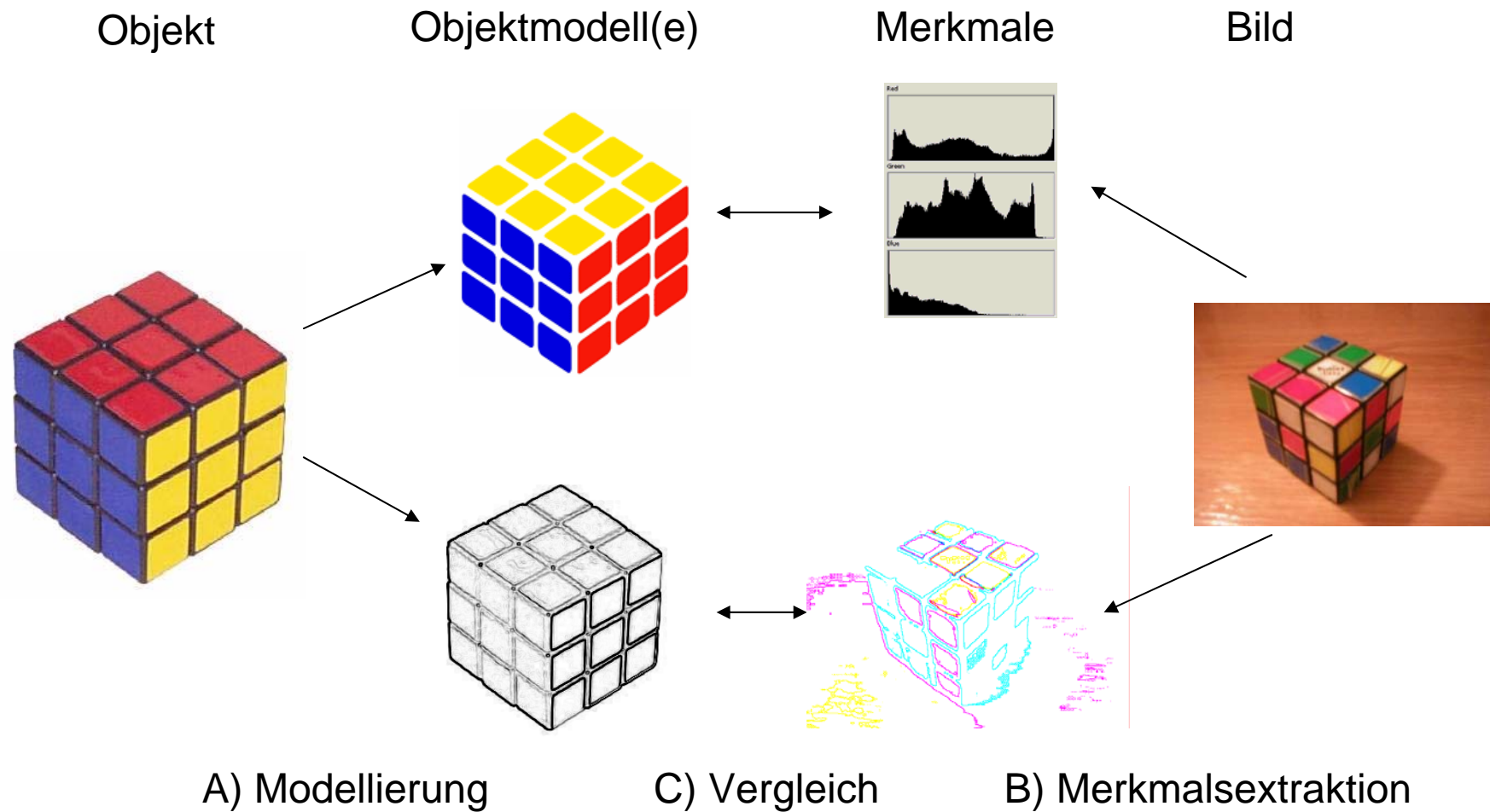
# Übersicht

- Motivation
- Anforderungen an Objekterkennungsalgorithmen
- Objekterkennung mittels
  - Konturen
  - Farben
  - Texturen
  - Bewegungen
- Beispielanwendungen: Objekterkennung in Videos
  - Video-OCR
  - Analyse der Bewegung einer Person

# Warum Objekterkennung ?

- **Schrifterkennung**
  - OCR-Software zur Texterkennung
  - Automatische Postleitzahlenerkennung
- **Video-Überwachung:**
  - Identifikation verdächtiger Personen (Diebstahl, Bedrohung)
  - Erkennung verdächtiger Gegenstände (unbeaufsichtigte Koffer auf Flughäfen oder Bahnhöfen)
- **Inhaltsbasierte Bildsuche**
  - Internetsuche (z.B. [Google Bilder](#))
  - Bild- und Videoarchive (z.B. [gettyimages](#))

# Modell-basierte Objekterkennung



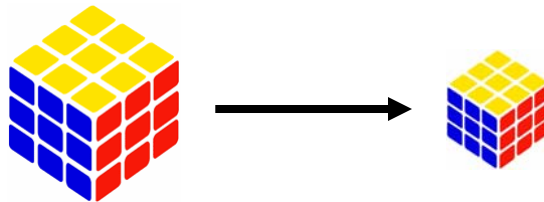
# Was ist ein gutes Objektmodell ?

- ***sensitivity***
  - Das Modell muss alle relevanten Merkmale darstellen und zwischen diesen unterscheiden können.
- ***uniqueness***
  - Das Modell soll in der Lage sein, Objekte möglichst eindeutig zu beschreiben; gleichartige Objekte sollen die gleiche Beschreibung haben.
- ***stability***
  - Kleine Veränderungen am Objekt sollen kleine Auswirkungen auf das Modell haben.
- ***efficiency***
  - Es muss möglich sein, Merkmale effizient aus Daten zu berechnen und Merkmale effizient zu vergleichen.

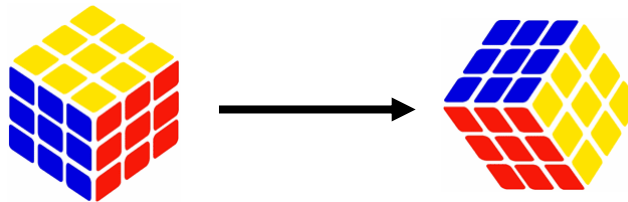
# Anforderungen (I)

**Das Modell sollte invariant sein gegenüber:**

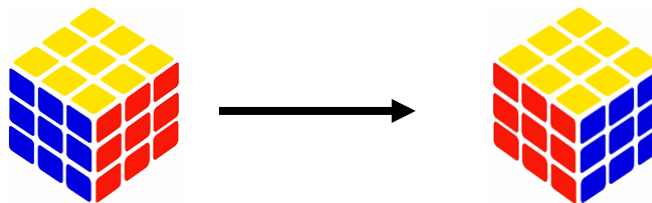
- Skalierungen



- Rotationen



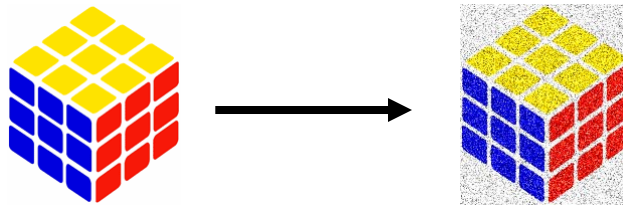
- Spiegelungen



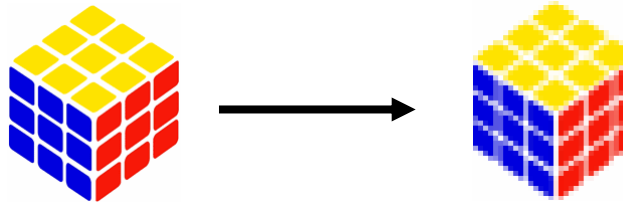
# Anforderungen (II)

**Das Modell sollte invariant sein gegenüber:**

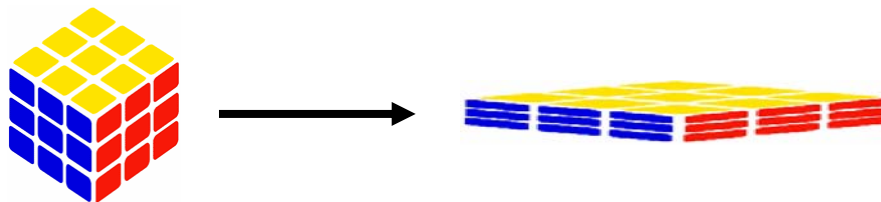
- Rauschen



- Unschärfe



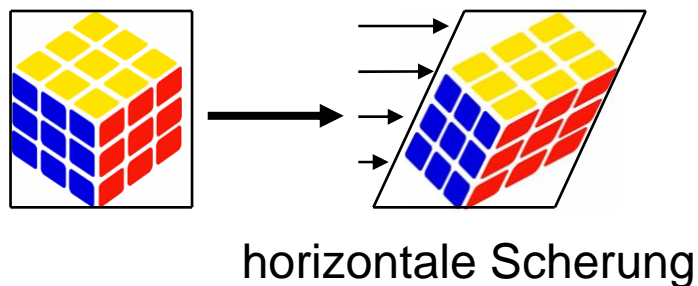
- Stauchungen



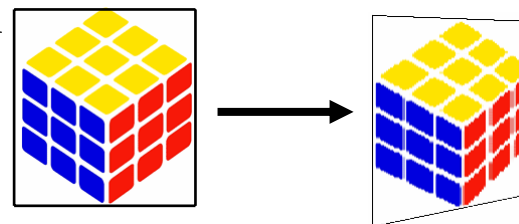
# Anforderungen (III)

**Das Modell sollte invariant sein gegenüber:**

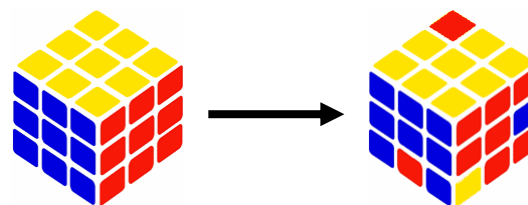
- Scherungen



- Perspektivischen Verzerrungen

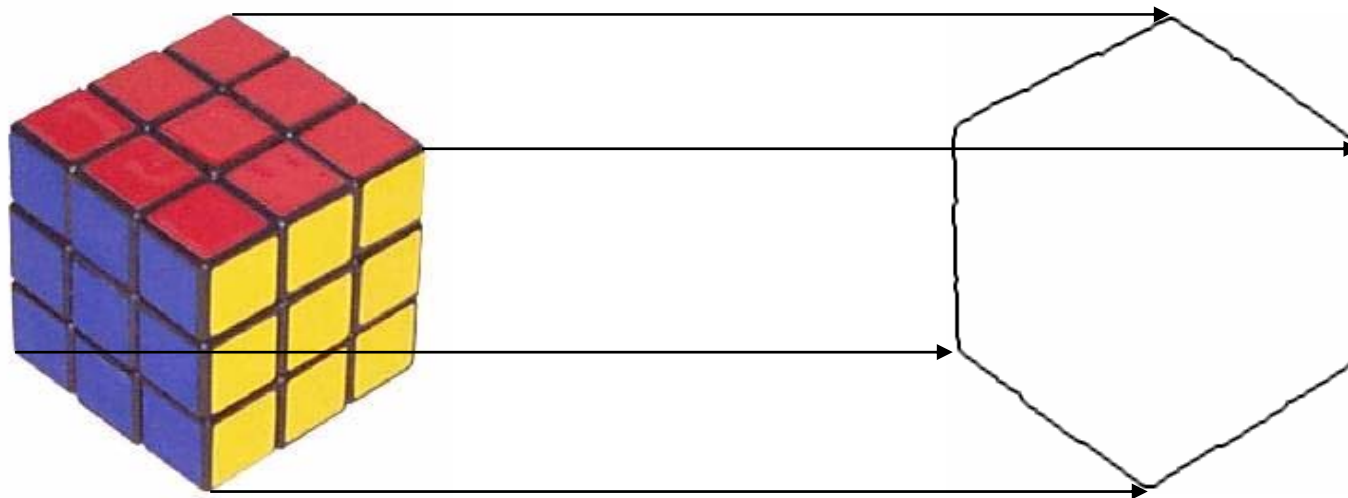


- Objektverformungen

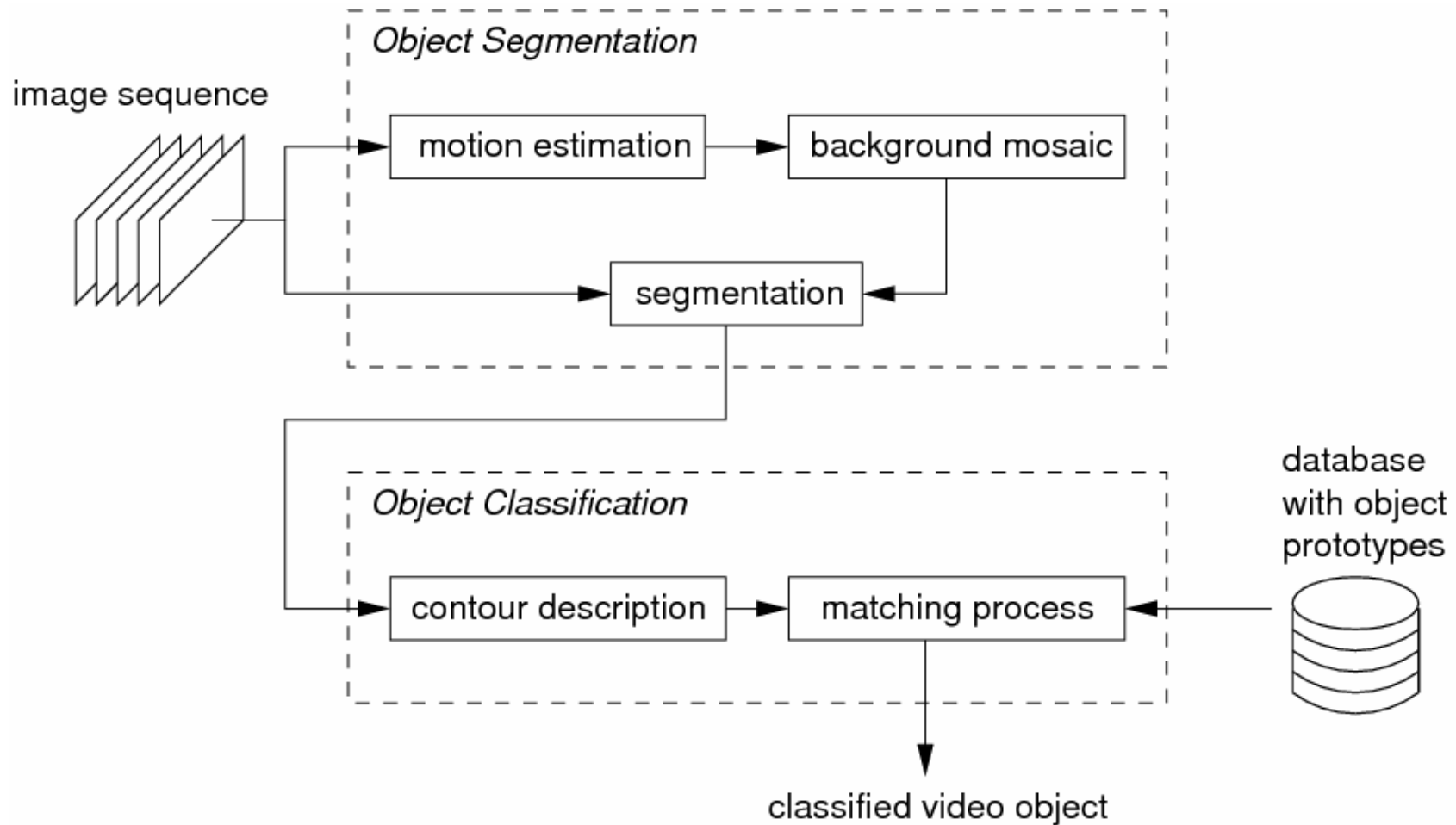




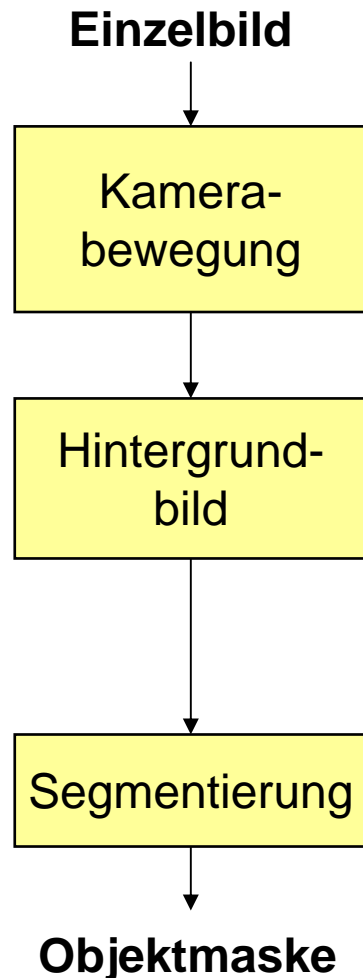
# Objekterkennung durch Vergleich von Konturen (I)



# Objekterkennung durch Vergleich von Konturen (II)



# Objektsegmentierung



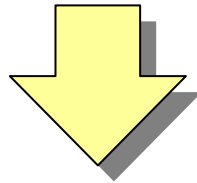
- Annahme: Die Hälfte aller Pixel eines Bildes sind Hintergrundpixel.
- Berechne Kamerabewegung zwischen zwei benachbarten Bildern des Videos.
- Richte alle Bilder einer Kameraeinstellung entsprechend des Kameramodells aus, so dass der Bildhintergrund deckungsgleich ist.
- Verwende einen Medianfilter, um aus den transformierten Bildern ein Hintergrundbild zu erzeugen.
- Vergleiche das Hintergrundbild mit den transformierten Einzelbildern.

# Ablauf der Objekterkennung

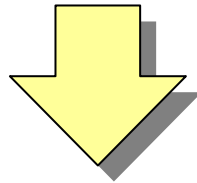
1. Parametrisierung (Abtastung) der Kontur mit einer vordefinierten Anzahl an Konturpunkten.
2. Identifiziere Merkmalspunkte zur Beschreibung der Kontur
  - *Kompaktheit*
  - *Exzentrizität*
  - *Curvature-Scale-Space-Verfahren*
3. Vergleiche die Merkmale mit Merkmalen von bekannten Objekten (*Shape Matching*)

# Parametrisierung einer Kontur

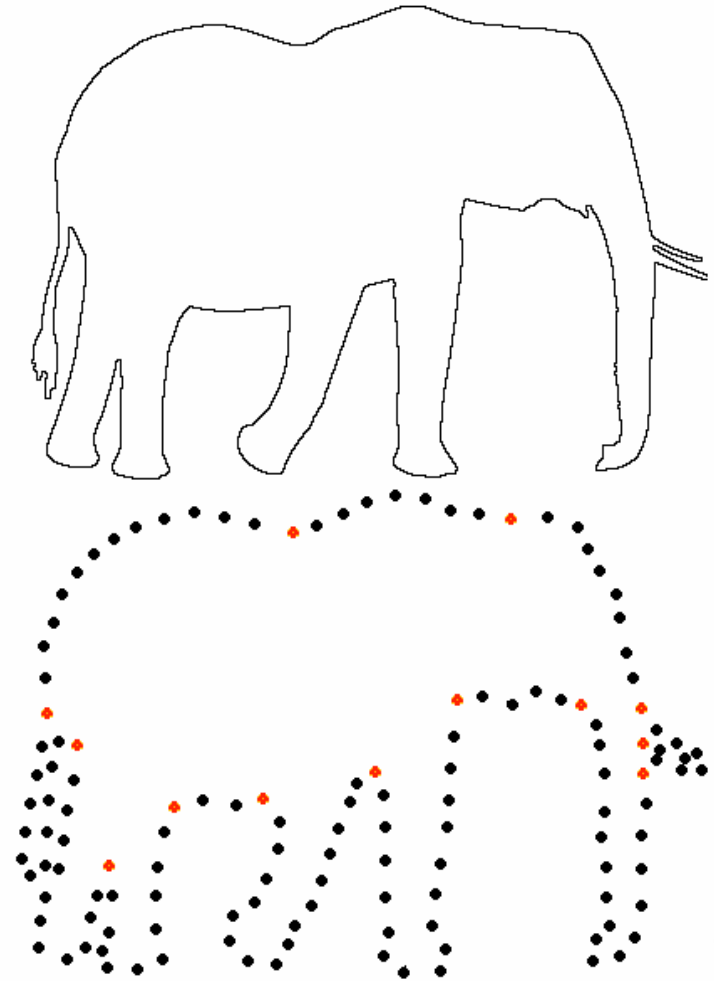
Segmentierung



Parametrisierung  
der Kontur



Berechnung der  
Konturmerkmale



# Vergleich von Konturen (I)

## Kompaktheit (*compactness*)

- Globaler Konturdeskriptor, der einen aggregierten Wert für die komplette Kontur liefert
- Eignet sich nur für eine grobe Abschätzung der Ähnlichkeit zweier Konturen
- Einfach zu berechnen
- beschreibt die Ähnlichkeit einer Kontur mit einem Kreis
- invariant gegenüber geometrischen Transformationen wie Rotation oder Skalierung
- Der Wert für die Kompaktheit wird bei einem Kreis minimal.

# Vergleich von Konturen (II)

## Kompaktheit (*compactness*)

- $c_i$ : Kompaktheit
- $i$ : segmentiertes Objekt
- $U$ : Länge der Kontur
- $F$ : Fläche des Objektes

$$c_i = \frac{U^2}{4 \cdot \pi \cdot F}$$

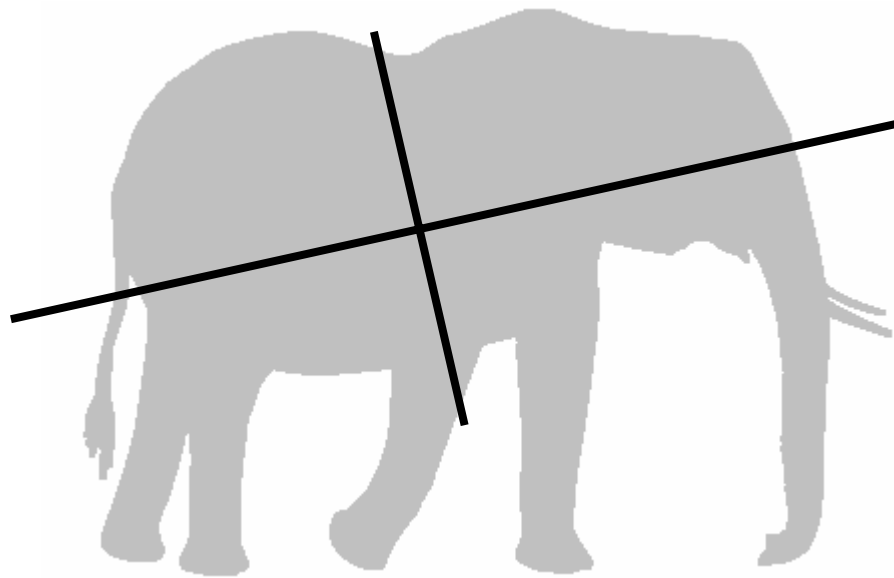
- Unterschiede zwischen zwei Konturen  $i$  und  $j$

$$\alpha_c(i, j) = \frac{|c_i - c_j|}{\max(c_i, c_j)}$$

# Vergleich von Konturen (III)

## Exzentrizität (*eccentricity*)

- Verhältnis der Längen der Hauptachsen bezogen auf die zentralen Momente der Konturpixel  
→ invariant gegenüber geometrischen Transformationen.





# Vergleich von Konturen (IV)

## Exzentrizität (*eccentricity*)

zentralen Momente der Konturpixel

$$M_{n,m} = \sum_{x,y} (\bar{x} - x(u))^n (\bar{y} - y(u))^m$$

Schwerpunkt der Konturpixel

$$\bar{x} = \frac{1}{N} \sum_{u=0}^{N-1} x(u) \quad \text{und} \quad \bar{y} = \frac{1}{N} \sum_{u=0}^{N-1} y(u)$$

# Vergleich von Konturen (V)

## Exzentrizität (*eccentricity*)

$e_i$ : Exzentrizität

$$e_i = \frac{(M_{2,0} - M_{0,2})^2 + 4 \cdot M_{1,1}}{F}$$

Vergleich zweier Objekte  $i$  und  $j$ :

$$\alpha_e(i, j) = \frac{|e_i - e_j|}{\max(e_i, e_j)}$$

# Vergleich von Konturen (VI)

## Skalenraumabbildung (*curvature scale space*)

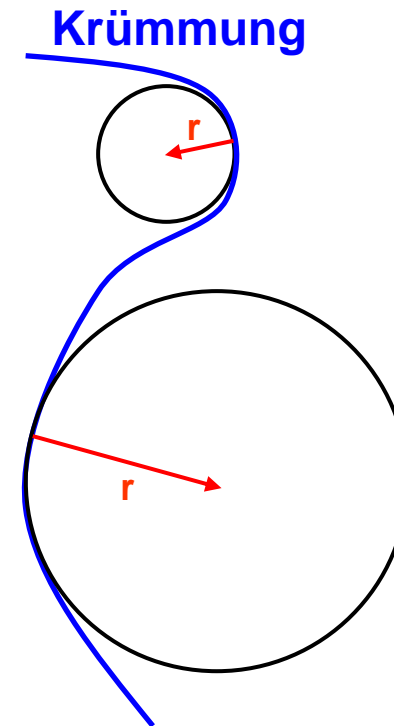
- Beschreibung der Krümmung der Kontur eines Objektes
  - Stark und lang gekrümmte Bereiche sollen durch hohe Merkmalswerte beschrieben werden
  - Die Größe eines Objektes soll keinen Einfluss auf die Merkmalswerte haben.
- Vergleich von Konturen mittels Skalenraumabbildungen

# Definition der Krümmung (I)

- Die **Stärke der Krümmung** in einem Punkt entspricht dem Kehrwert des Radius eines sich anschließenden Kreises (der Kreis berührt die Kurve):

$$K = \frac{1}{r}$$

- Die Krümmung ist ein Vektor, der in die Richtung der Kreismittelpunktes zeigt.
- Ein kleiner Kreis repräsentiert eine hohe Krümmung; eine Gerade hat eine Krümmung von Null.



# Definition der Krümmung (II)

- Gegeben ist eine planare Kurve  $u(t)$  innerhalb eines 2D Raums.  $u(t)$  wird durch die Bogenlänge  $t$  parametrisiert.
- Die Kurve  $u(t)$  wird durch die beiden Funktionen  $x(t)$  und  $y(t)$  definiert:

$$u(t) = (x(t), y(t)).$$

- Die Krümmung  $K$  für die Kurve  $u(t)$  ist definiert als:

$$K = \frac{\dot{x} \cdot \ddot{y} - \dot{y} \cdot \ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}$$

$\dot{x}$  und  $\dot{y}$  definieren die erste Ableitung (Gradient),

$\ddot{x}$  und  $\ddot{y}$  definieren die zweite Ableitung (Änderung des Gradienten).

# Definition der Krümmung (III)

- Herleitung einer nicht-allgemeinen Definition der Krümmung: verwende explizite planare Kurven:  $y = f(x)$ .
- Krümmung im Punkt  $(x, f(x))$ :

$$K = \frac{f''(x)}{\left(1 + (f'(x))^2\right)^{3/2}}$$

- Typische Berechnung der Krümmung in vielen Bereichen:
  - berechne die Bewegung von Gasen oder Flüssigkeiten entlang von Oberflächen,
  - schätze das Verhalten, falls Bauteile gebogen werden (Statik von Brücken).

# Definition der Krümmung (IV)

## Beispiel

- Beispielfunktion  $u(t) = (x(t), y(t)) = (t, t^2)$ .

Explizite Definition der Funktion:  $y = f(x) = x^2$ .

- Krümmung basierend auf der **parametrisierten Kurve**:

Erste und zweite Ableitung:  $\dot{x} = 1$ ,  $\ddot{x} = 0$ ,  $\dot{y} = 2t$ ,  $\ddot{y} = 2$ .

$$K(t) = \frac{\dot{x} \cdot \ddot{y} - \dot{y} \cdot \ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} = \frac{1 \cdot 2 - 2t \cdot 0}{(1^2 + (2t)^2)^{3/2}} = \frac{2}{(1 + 4t^2)^{3/2}}$$

- Krümmung basierend auf der **expliziten Definition**:

$$f'(x) = 2x, \quad f''(x) = 2 \quad K(x) = \frac{f''(x)}{(1 + (f'(x))^2)^{3/2}} = \frac{2}{(1 + 4x^2)^{3/2}}$$

# Definition der Krümmung (V)

- Approximation der Ableitung für diskrete Werte (parametrisierte Kurve):

$$\dot{x}(t) = \frac{x(t+1) - x(t-1)}{2 \cdot h_x}$$

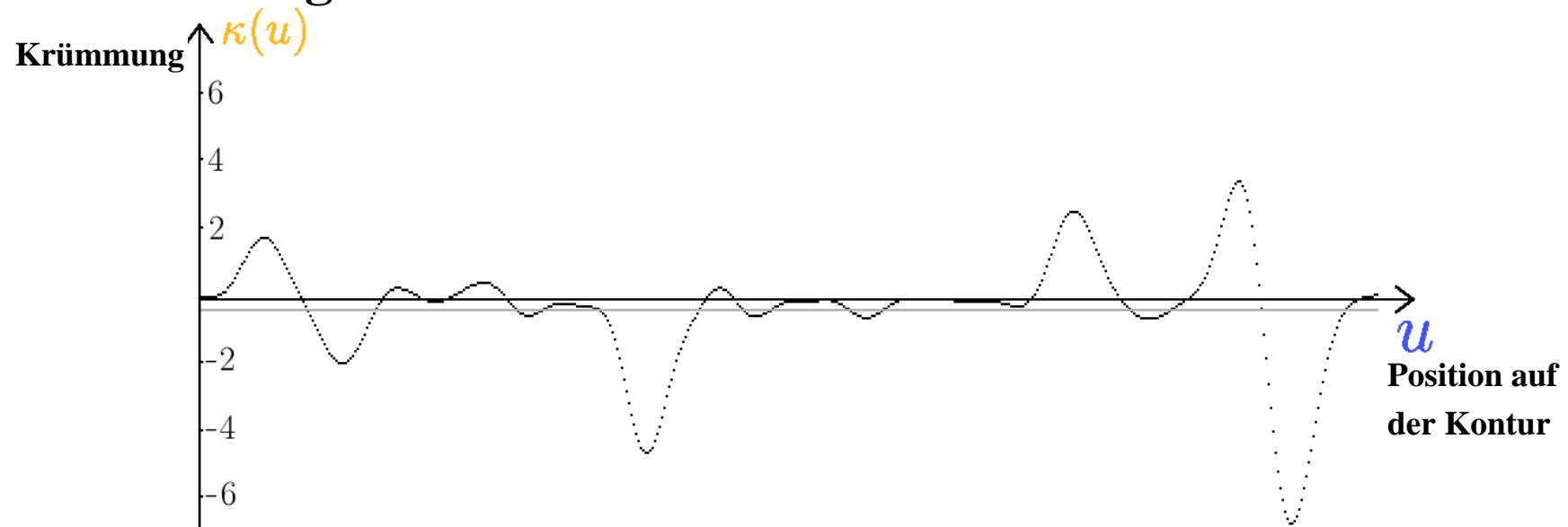
$$\dot{y}(t) = \frac{y(t+1) - y(t-1)}{2 \cdot h_y}$$

- Der Parameter  $t$  ist für ganze Zahlen definiert ( $t \in \mathbb{N}$ ).
- $h_x$  und  $h_y$  normalisieren die Ableitungen abhängig von der Entfernung der Abtastpunkte.



# Konturvergleich mit Hilfe von Krümmungen

## Krümmungsfunktion der Kontur

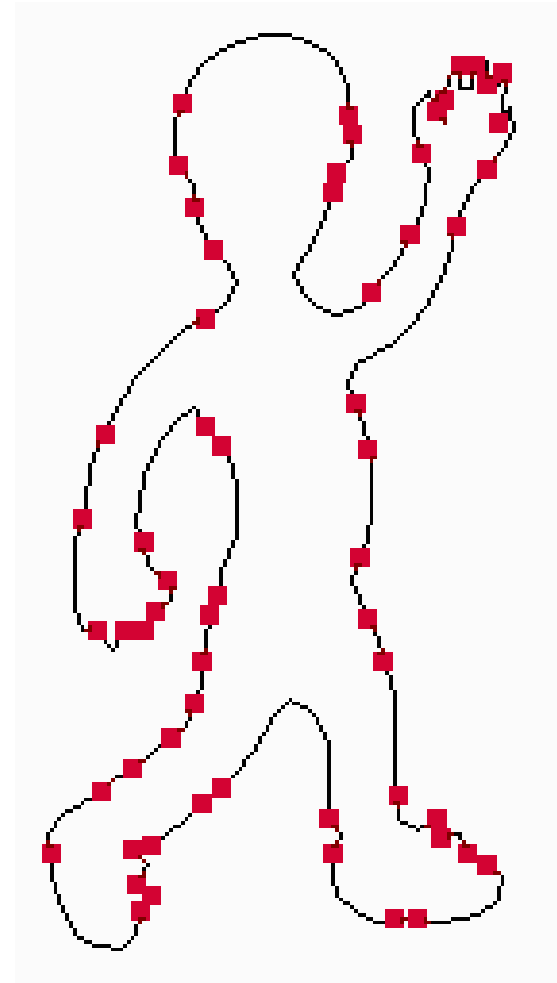


**Problem:** Zwei Krümmungsfunktionen können nur sehr schwer miteinander verglichen werden.

→ Identifiziere signifikante Punkte der Krümmungsfunktion (das Curvature-Scale-Space-Verfahren liefert eine gute Beschreibung)

# Curvature-Scale-Space-Verfahren (I)

- Analysiere äußere Kontur eines Objektes.
- Glätte Kontur mit einem Gaußfilter.
- Die **Wendepunkte der Kontur** werden als Merkmalspunkte gespeichert.



# Curvature-Scale-Space-Verfahren (II)

## Ablauf

1. Iterative Glättung der Kontur
2. Berechnung der Krümmung
3. Betrachte Nullstellen der Krümmungsfunktion  
(entspricht den Wendepunkten der Kontur) :

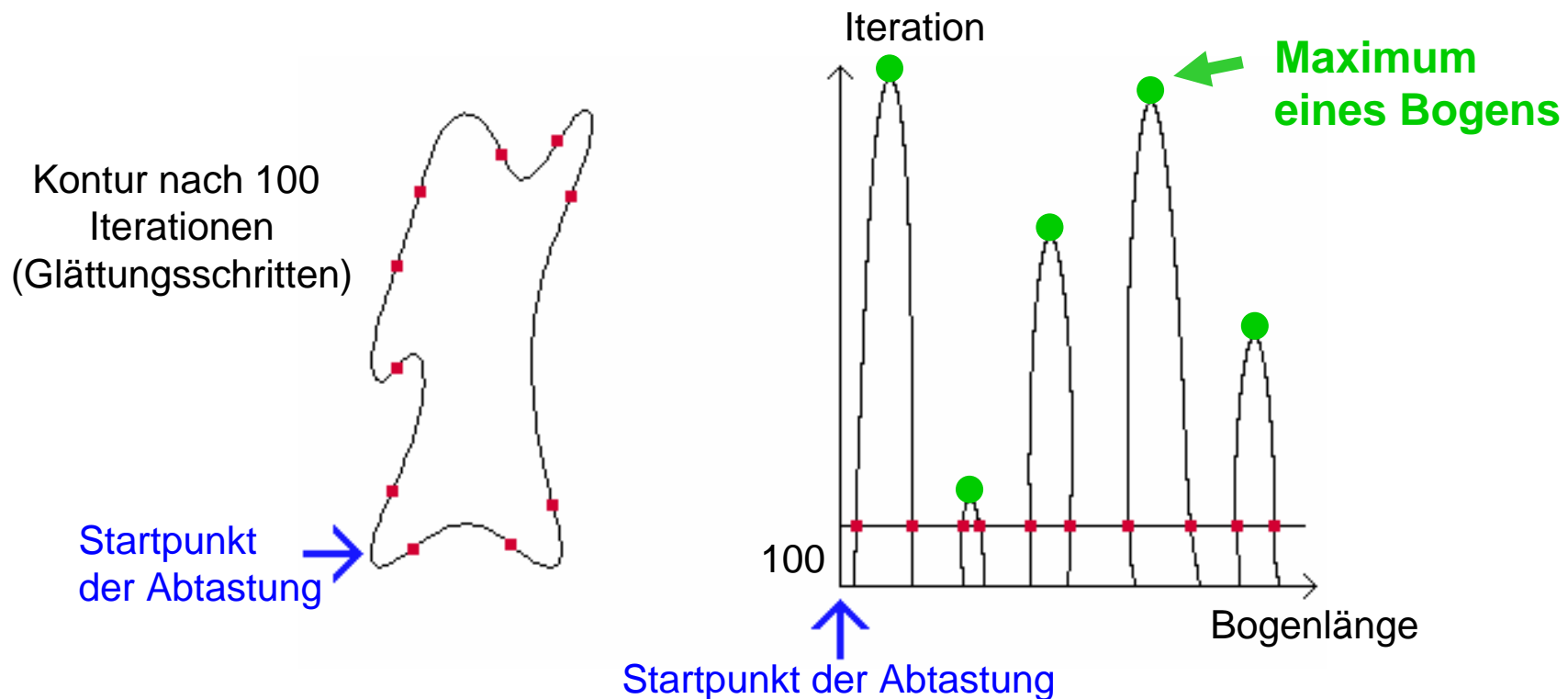
$$\kappa(u, n) = 0$$

**Definition: Skalenraumabbildung (Curvature-Scale-Space-Bild)**

$$I(u, n) = \{ (u, n) \mid \kappa(u, n) = 0 \}$$

# Curvature-Scale-Space-Verfahren (III)

- Ein **Curvature-Scale-Space-Bild** (CSS-Bild) stellt die Wendepunkte, die bei der Glättung entstehen, visuell dar.



Die **Bögen** werden als Merkmale zur Beschreibung des Objektes verwendet.

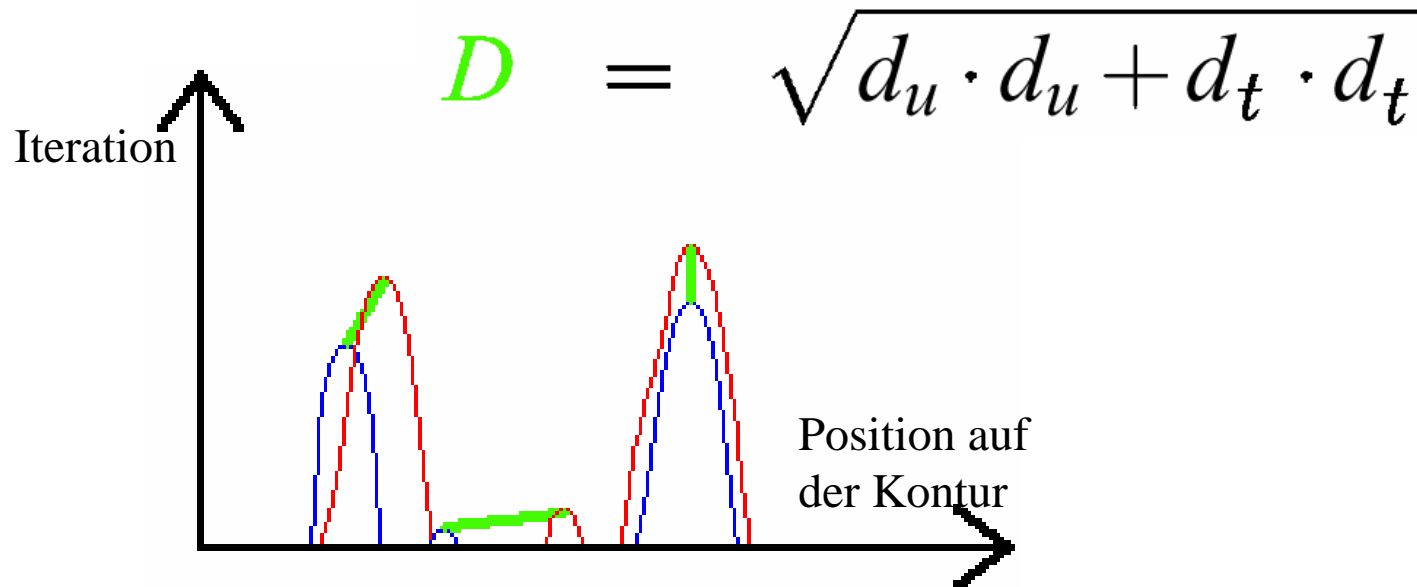
# Curvature-Scale-Space-Verfahren (IV)

## Merkmale von CSS-Bildern

- Die Bögen im CSS-Bild beschreiben **konkave Bereiche** einer Kontur.
- Die Bögen liefern die Merkmale zur Beschreibung einer Kontur.
- Jeder Bogen wird beschrieben durch
  - eine **Position**:  
relative Position im Vergleich zu den anderen Bögen,
  - eine **Höhe** (Anzahl der erforderlichen Glättungen):  
Stärke eines konkaven Segmentes.

# Vergleich zweier Konturen

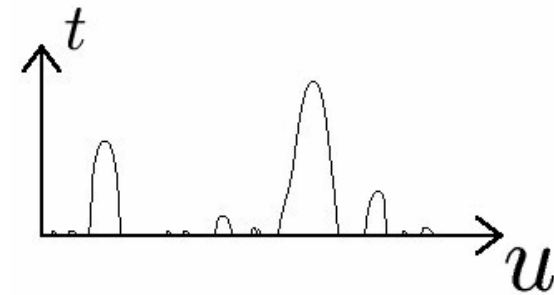
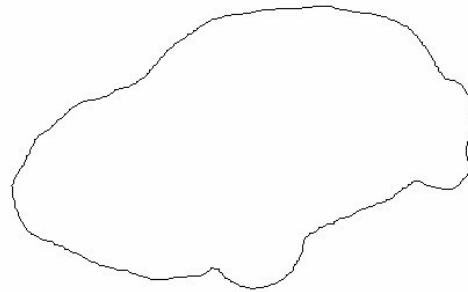
1. Verschiebe ein CSS-Bild horizontal bis die höchsten Bögen übereinstimmen (Verfahren wird invariant gegenüber Rotationen)
2. Berechne die Euklidische Distanz zwischen zwei Bögen



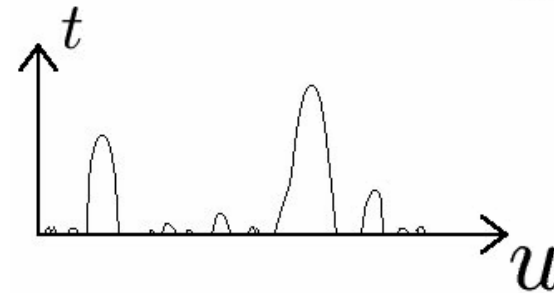
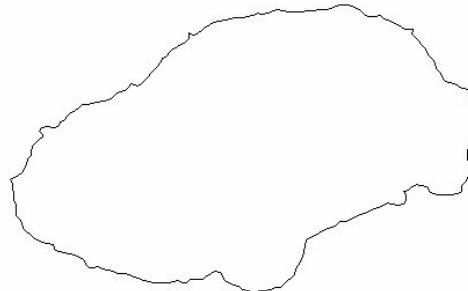
3. Summiere die Distanzen

# Merkmale von CSS-Bildern (I)

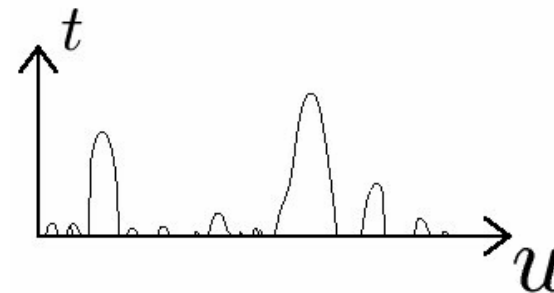
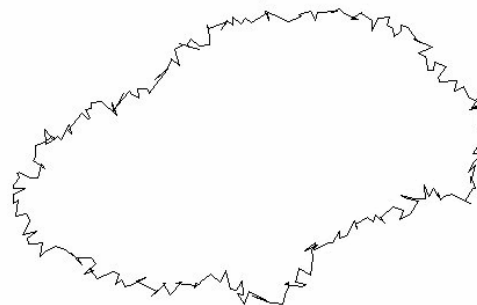
- Originalkontur



- Verrauschte Kontur



- Stark verrauschte Kontur



→ Die CSS-Bilder sind sehr ähnlich.

# Merkmale von CSS-Bildern (II)

## Vorteile

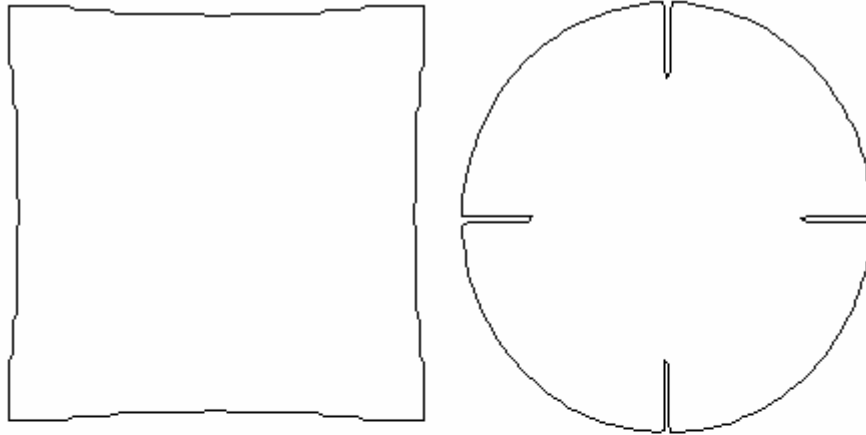
- Gute Klassifikationsergebnisse
- Nur wenige Werte werden zur Beschreibung komplexer Objekte benötigt.
- Der Ansatz ist invariant gegenüber Rotationen oder Skalierungen.
- Robust bei Rauschen und perspektivischen Verzerrungen
- Schnelle Berechnung der Merkmale und beim Vergleich zweier Konturen.

## Nachteile

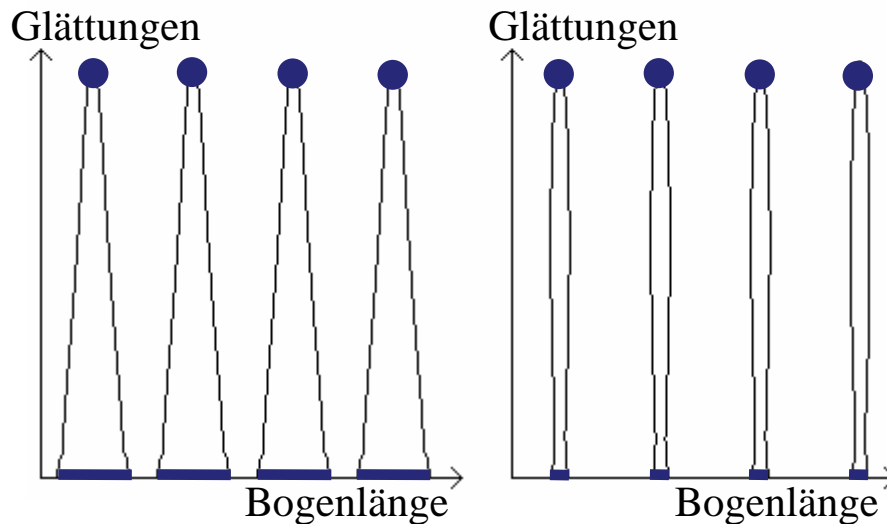
- Schlechte Klassifikationsergebnisse bei einzelnen Konturen:
  - Mehrdeutigkeiten
  - Konvexe Objektregionen werden nicht ausreichend berücksichtigt



# Mehrdeutigkeiten von CSS-Bildern (I)



- Die **Höhe eines Bogens** beschreibt die Länge der gekrümmten Region **und** die Stärke der Krümmung.



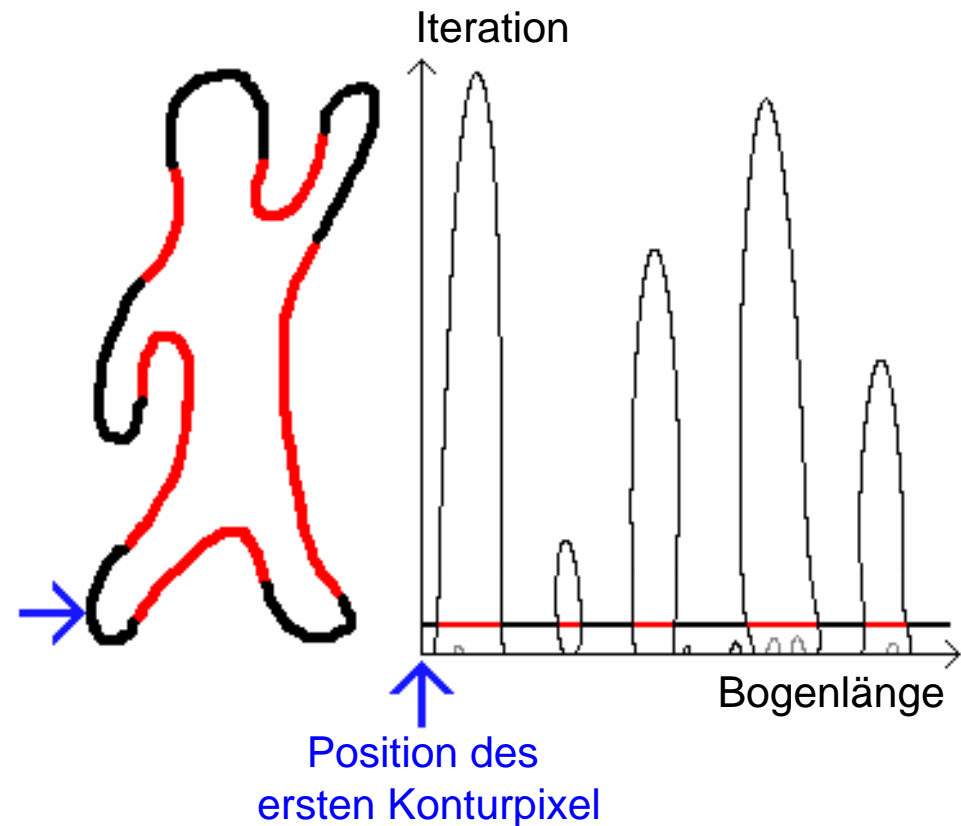
- Für jeden Bogen wird zusätzlich dessen **Breite** gespeichert.

# Mehrdeutigkeiten von CSS-Bildern (II)

## Konvexe Regionen

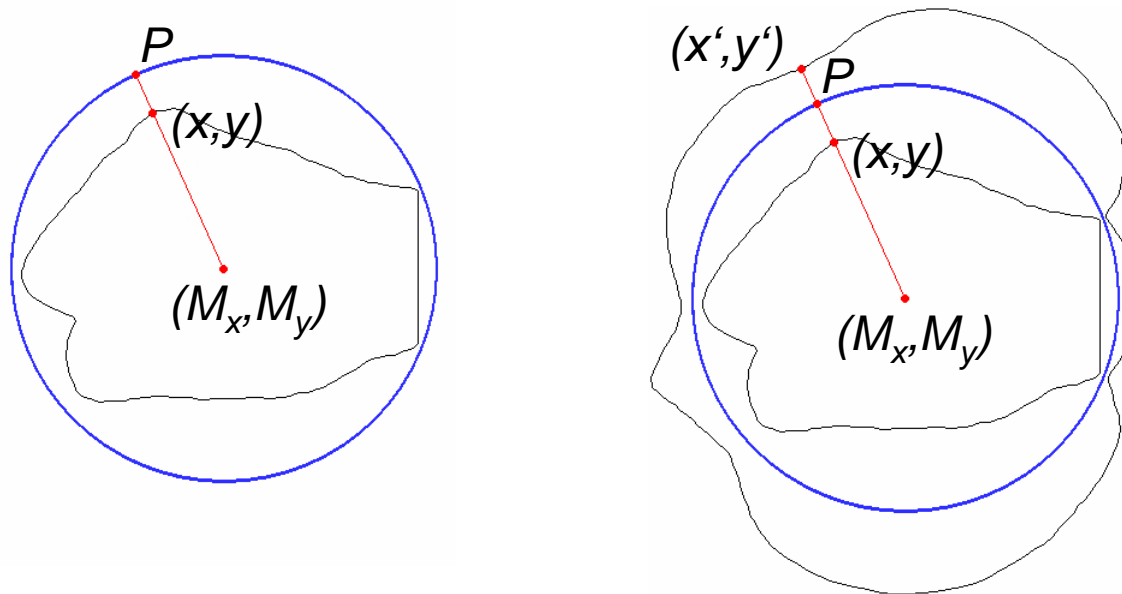
- Schlechte Repräsentation der konvexen Region einer Kontur.
- Konvexe Objekte können gar nicht unterschieden werden.

→ Lösung:  
Spiegelung der Kontur

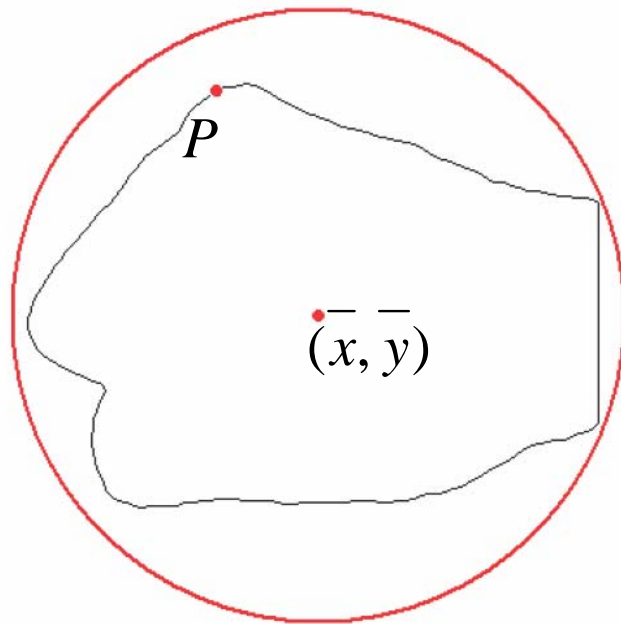


# Gespiegelte Konturen (I)

- **Idee:** Spiegele jedes Konturpixel an einem die Kontur umgebenden Kreis und erzeuge eine neue Kontur.
- Starke konvexe Regionen der ursprünglichen Kontur werden konkave Regionen der gespiegelten Kontur.



# Gespiegelte Konturen (II)

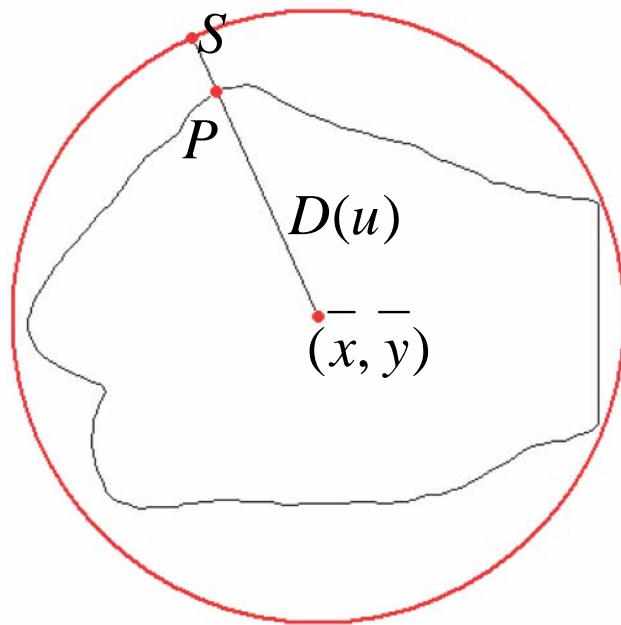


$(\bar{x}, \bar{y})$  Schwerpunkt der Konturpixel

$P$  Konturpixel

$$R = \max_u \{ \sqrt{(\bar{x} - x(u))^2 + (\bar{y} - y(u))^2} \}$$

# Gespiegelte Konturen (III)



$(\bar{x}, \bar{y})$  Schwerpunkt der Konturpixel

$P$  Konturpixel

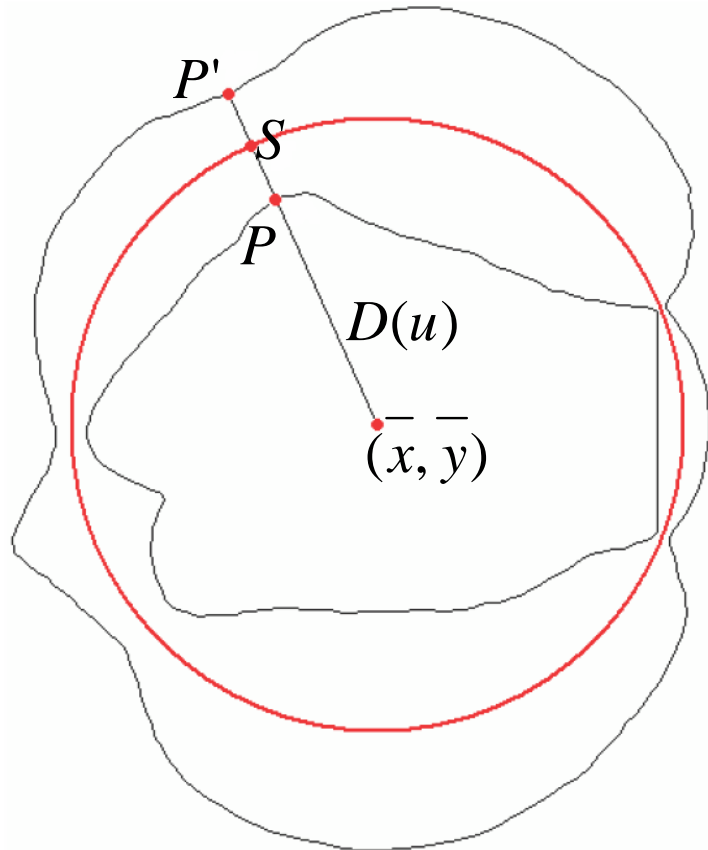
$D(u)$  Entfernung zum Mittelpunkt

$S$  Punkt auf Kreislinie

$$R = \max_u \{ \sqrt{(\bar{x} - x(u))^2 + (\bar{y} - y(u))^2} \}$$

$$D_u = \sqrt{(\bar{x} - x(u))^2 + (\bar{y} - y(u))^2}$$

# Gespiegelte Konturen (IV)



$(\bar{x}, \bar{y})$  Schwerpunkt der Konturpixel

$P$  Konturpixel

$D(u)$  Entfernung zum Mittelpunkt

$S$  Punkt auf Kreislinie

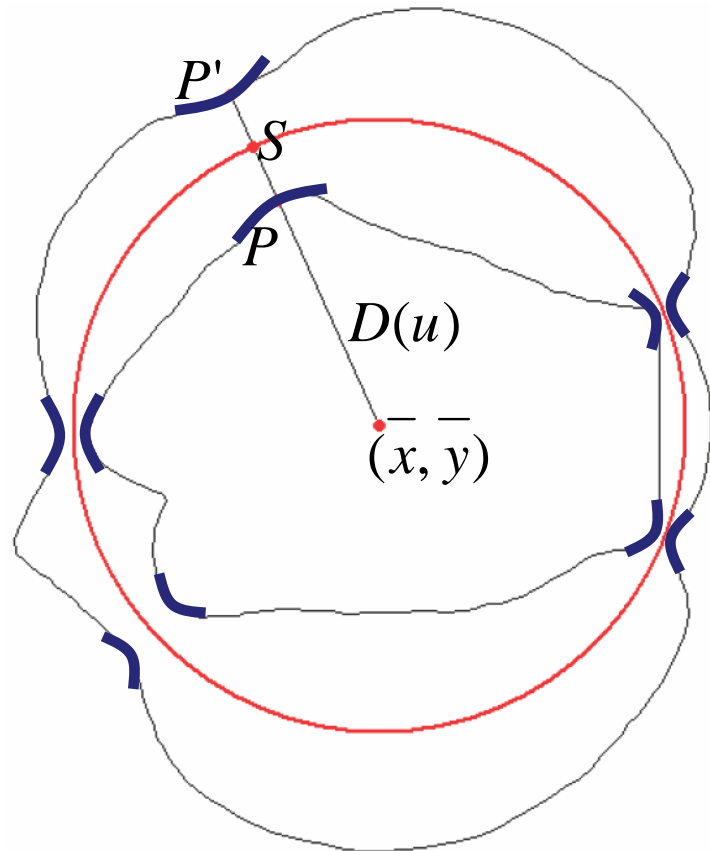
$P'$  Gespiegeltes Konturpixel

$$R = \max_u \{ \sqrt{(\bar{x} - x(u))^2 + (\bar{y} - y(u))^2} \}$$

$$D_u = \sqrt{(\bar{x} - x(u))^2 + (\bar{y} - y(u))^2}$$

$$x'(u) = \frac{2R - D(u)}{D(u)} \cdot (x(u) - \bar{x}) + \bar{x}$$

# Gespiegelte Konturen (V)



$(\bar{x}, \bar{y})$  Schwerpunkt der Konturpixel

$P$  Konturpixel

$D(u)$  Entfernung zum Mittelpunkt

$S$  Punkt auf Kreislinie

$P'$  Gespiegeltes Konturpixel

$$R = \max_u \{ \sqrt{(\bar{x} - x(u))^2 + (\bar{y} - y(u))^2} \}$$

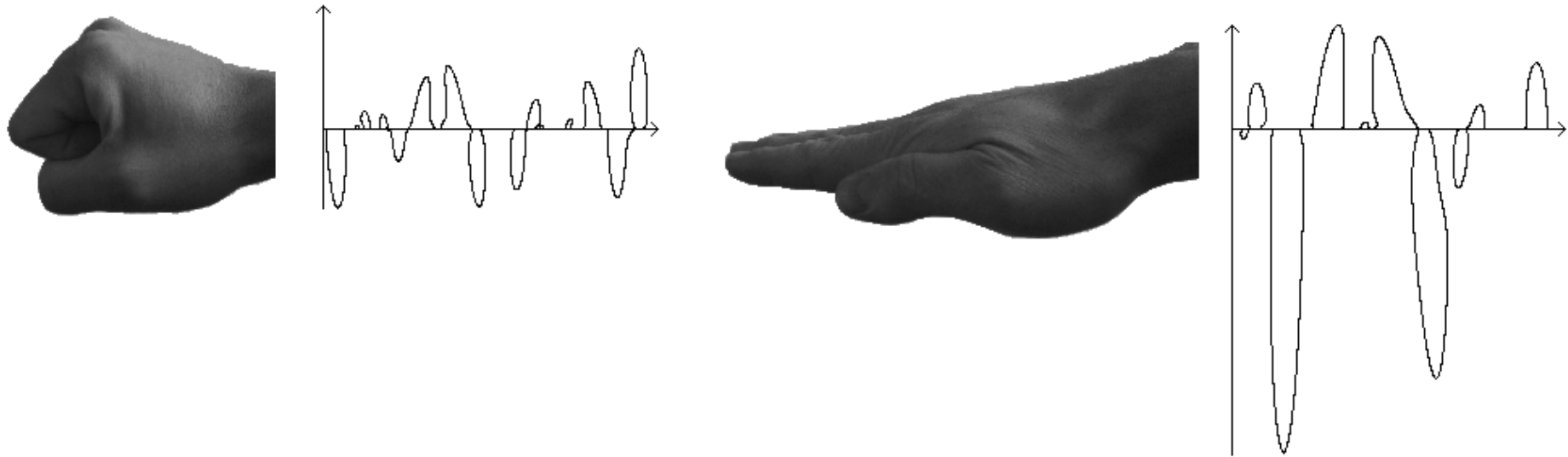
$$D_u = \sqrt{(\bar{x} - x(u))^2 + (\bar{y} - y(u))^2}$$

$$x'(u) = \frac{2R - D(u)}{D(u)} \cdot (x(u) - \bar{x}) + \bar{x}$$

# Gespiegelte Konturen (VI)

## Erweiterter Konturvergleich

- Berechne ursprüngliche Merkmale der CSS-Bilder.
- Berechne Merkmale für die gespiegelte Kontur.





# Objekterkennung in Videos (I)

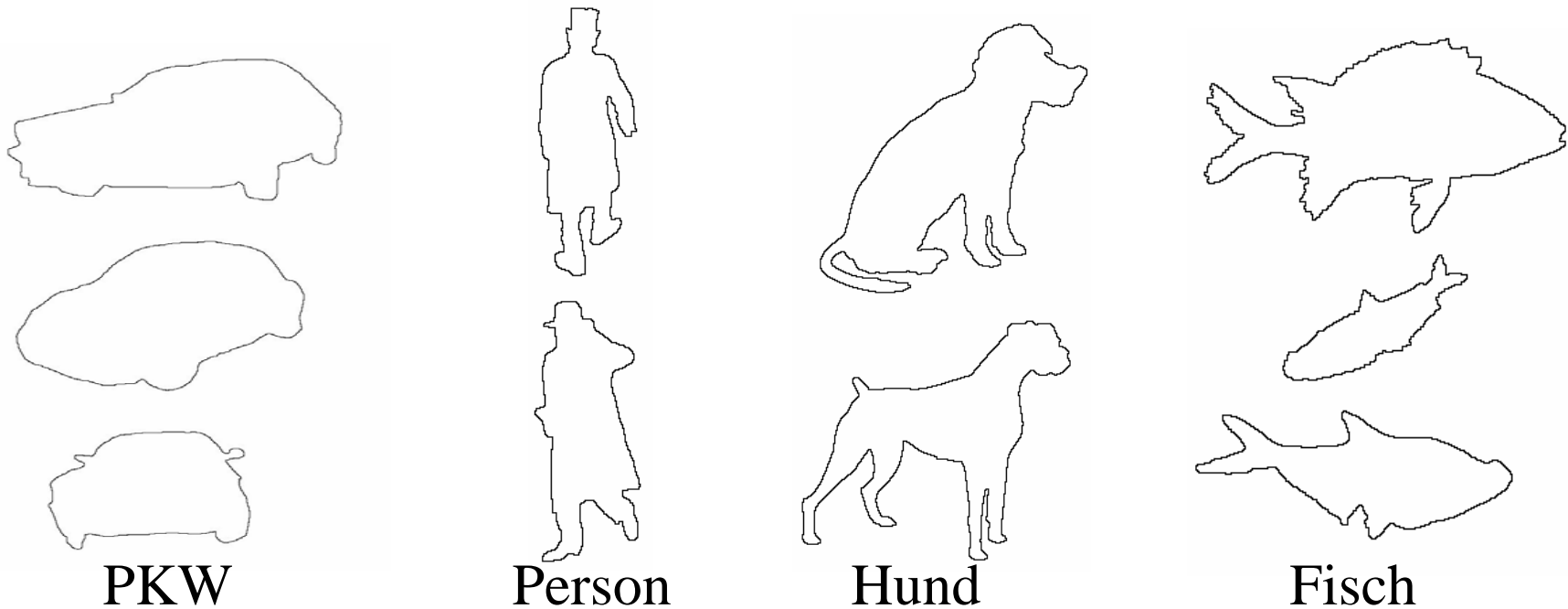
## Ansatz

- Vergleiche jedes Objekt in den einzelnen Bildern mit bekannten Objekten einer Datenbank.
- Berechne die durchschnittliche Differenz zwischen jedem Objekt und eine Objektklasse. Dabei werden ähnliche Objekte in einer Objektklasse zusammengefasst.
- Gebe die ähnlichsten Objekte aus.

# Objekterkennung in Videos (II)

## Objekte der Datenbank

- Datenbank enthält mehr als 300 Objekte
- 13 Objektklassen gruppieren ähnliche Objekte



# Objekterkennung in Videos (III)

## Kanonische Sichten

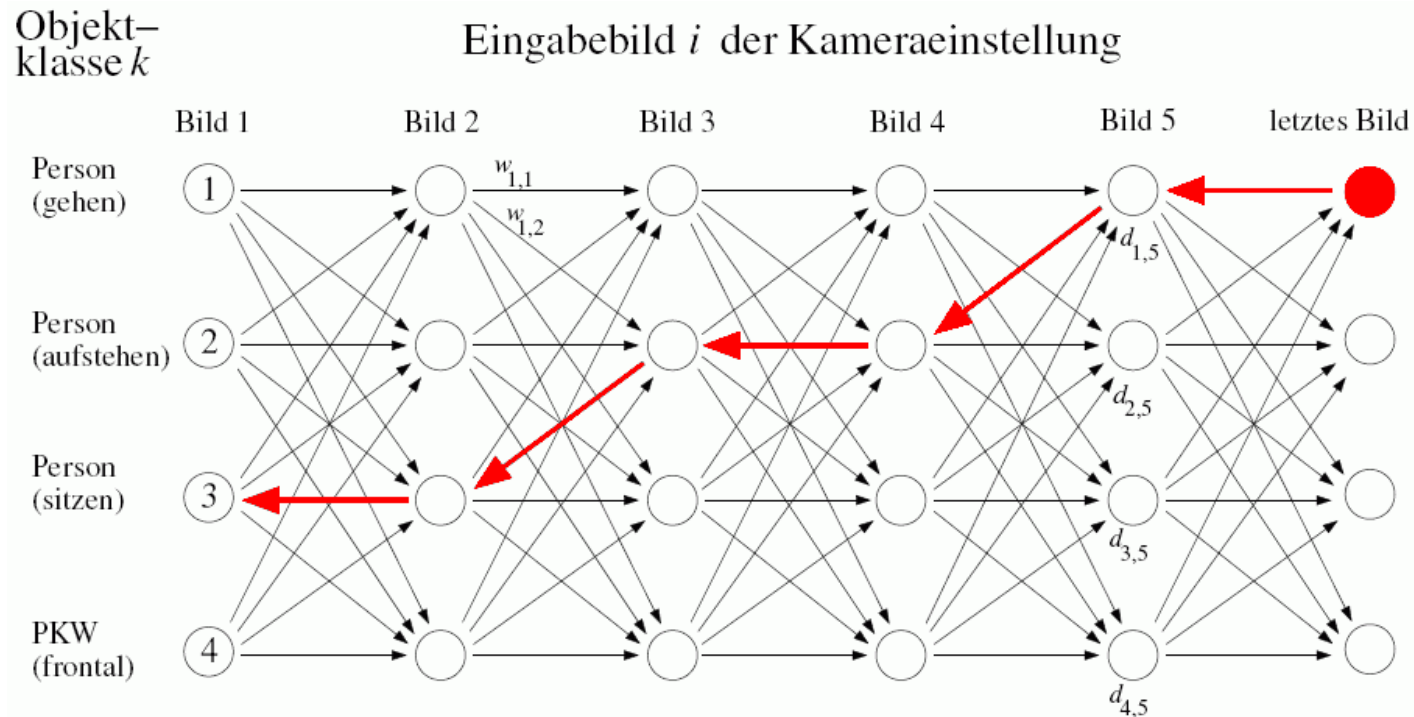
- Wie werden Objekte im menschlichen Gehirn repräsentiert?  
→ unbekannt
- aber: dreidimensionale Objekte scheinen als zweidimensionale Ansichten abgebildet zu werden
- Die Drehung eines Objektes zur Kamera hat starken Einfluss, ob und wie schnell ein Mensch ein Objekt erkennt.
- Einfach zu erkennende zweidimensionale Projektionen eines dreidimensionalen Objektes werden als *kanonische Sichten* (*canonical view*) bezeichnet.

# Objekterkennung in Videos (IV)

## Beispiele für kanonische Sichten

- Ansichten im Profil oder leicht erhöhte Ansichten von schräg vorne
- Vertraute Perspektiven, also Perspektiven, aus denen ein Objekt üblicherweise betrachtet oder im Fall von Gebrauchsgegenständen verwendet wird
- Bei der Auswahl von Objekten für die Referenzdatenbank wurden kanonische Sichten mit Vorrang berücksichtigt.

# Objekterkennung in Videos (V)



- Übergänge zwischen Objektklassen:  $W_{c_{i-1},i}$
- Knoten  $d_{c_i}$  speichert Differenz des Objektes  $i$  zur Objektklasse  $c_i$
- Pfad mit minimalen Gesamtkosten
- Letzter Knoten mit minimalen Kosten

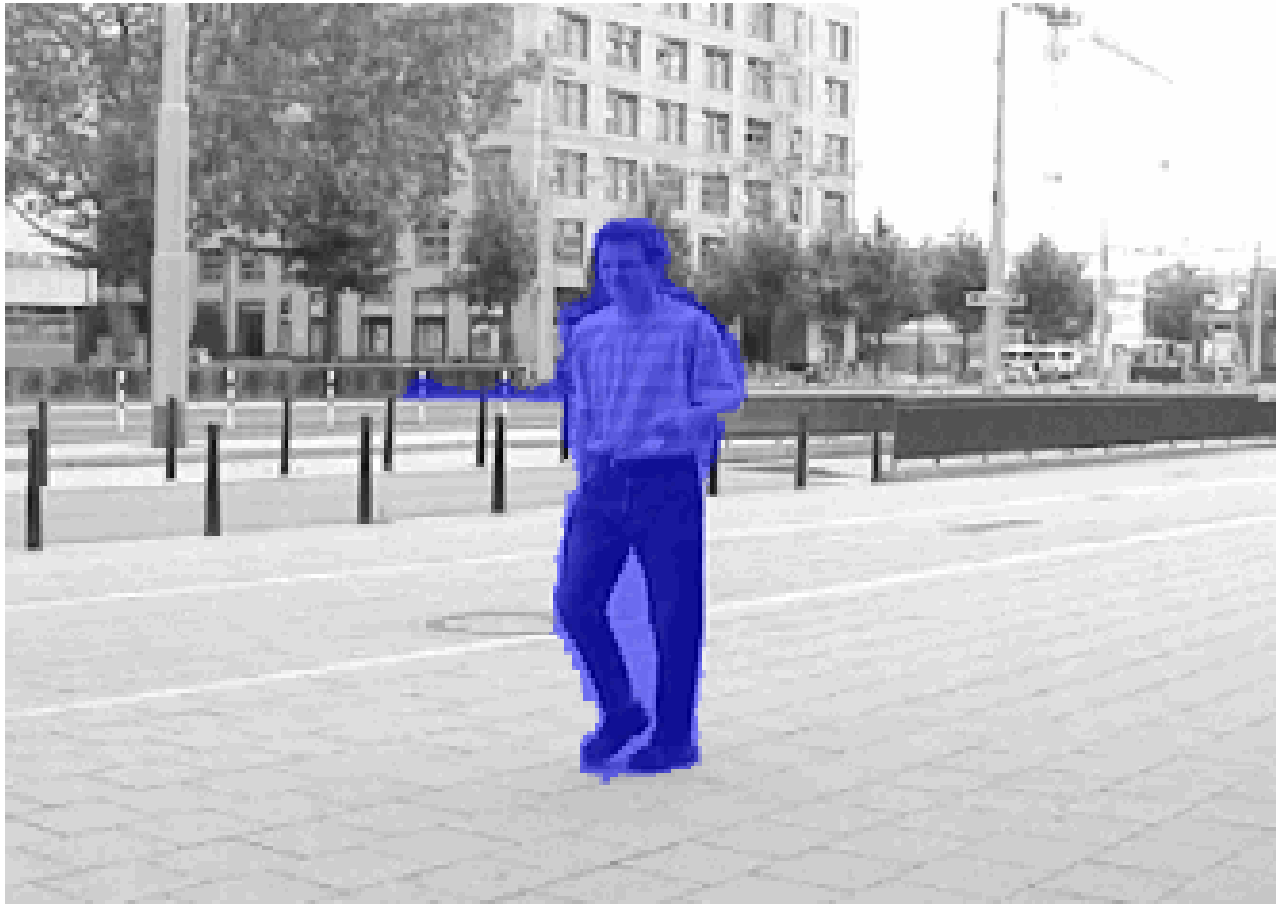
$$K = \min_c \sum_{i=1}^N d_{c_i,i} + w_{c_{i-1},c_i}$$

# Objekterkennung in Videos (VI)

## Erkennungsraten

- Die Erkennungsraten liegen zwischen 25-95 % (abhängig von der Komplexität des Objektes und der Anzahl an verfügbaren Objekten in der Datenbank).
- Starre Objekte (z.B Autos) werden deutlich zuverlässiger erkannt.
- Das CSS-Verfahren ist invariant gegenüber Skalierungen und Rotationen; es ist sehr robust bei Rauschen.
- Ein Vergleich von Konturen ist sehr effizient möglich (einmaliges Glätten, häufiger Vergleich mittels Euklidischer Distanz)

# Objekterkennung in Videos (VII)



**stehen**

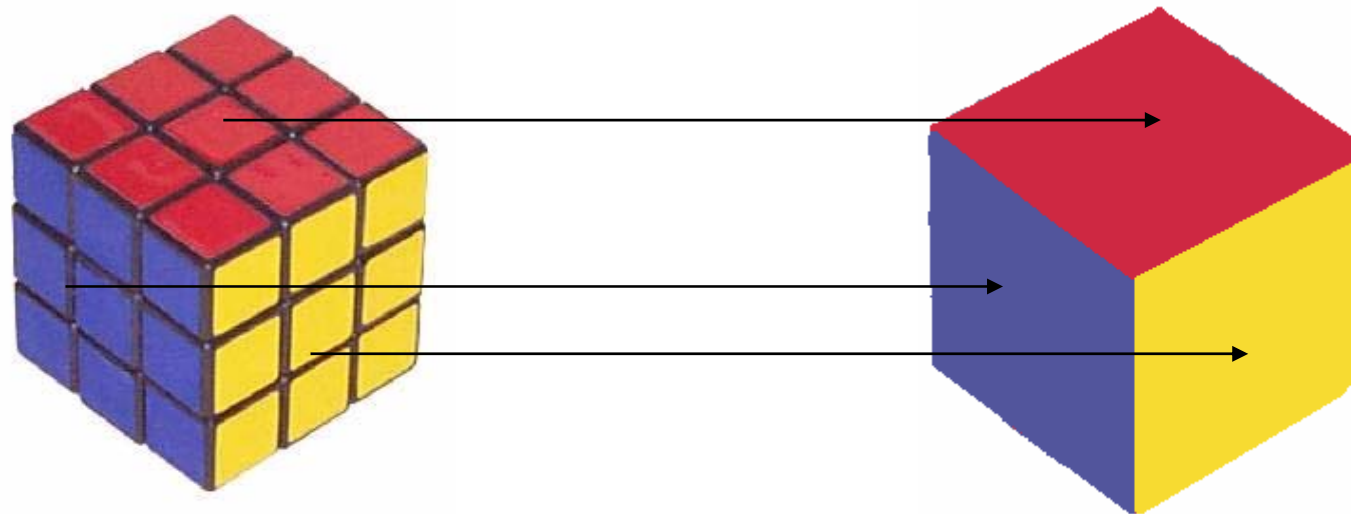
**gehen**

umdrehen

**hinsetzen**

**sitzen**

# Objekterkennung durch Vergleich von Farben





# Farbräume (I)

## Definitionen:

- **Farbe:** Sinnesempfindung (keine physikalische Eigenschaft), falls Licht einer bestimmten Wellenlänge auf die Netzhaut des Auges fällt. Die Sinneszellen im menschlichen Auge (**Zapfen** für Farben, **Stäbchen** für Helligkeitswerte) leiten Impulse an das Gehirn, das diese als Farben wahrnimmt. Mensch besitzt drei Arten von Zapfen (empfindlich für rot, grün und blau).
- **Sichtbares Licht:** zwischen 400 – 700 nm



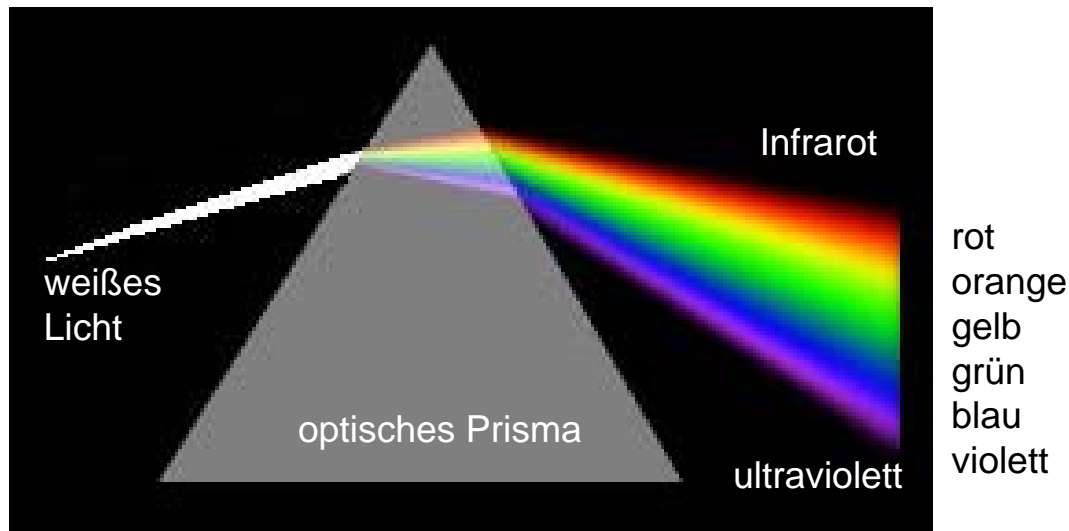
Quelle: Wikipedia.de

- **Farbraum:** Menge von Farben
- **Farbkörper:** Farbraum des menschlichen Auges
- **Farbmodell:** Beschreibt den Farbraum, den der Sehsinn oder ein Ein- oder Ausgabegerät (Display, Scanner, Drucker, Projektor, Foto, Kamera, Fernseher) darstellen kann.

# Farbräume (II)

## Grundlagen von Farben:

- Entdeckung 1666 von Newton: Sonnenlicht, das durch ein Prisma fällt, wird in ein kontinuierliches Farbspektrum aufgespalten.



# Farbräume (III)

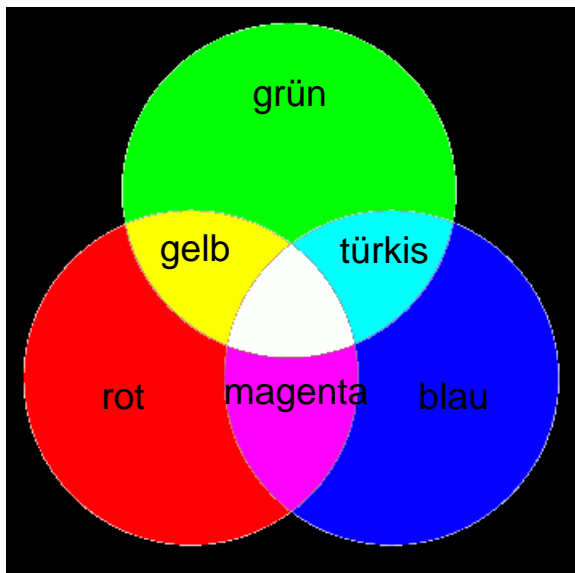
## Aufbau eines Farbraumes:

- Koordinatensystem, bei dem einzelne Farben die Achsen definieren (durch Aufbau des menschlichen Auges meistens 3 Dimensionen).
- Physikalische Farbmodelle (Mischung von Farben)
  - RGB, CMYK
  - Anordnung als Würfel
  - Bei Änderung einer Farbe ändern sich gleichzeitig Helligkeit, Sättigung, Farbton.
  - Im Jahr 1931 Definition der Primärfarben rot (=435,8 nm), grün (=546,1 nm) und blau (=700 nm) durch die CIE (International Commission on Illumination)
- Wahrnehmungsorientierte Farbmodelle (Beschreibung durch Helligkeit, Sättigung, Farbton)
  - HSV, HSI
  - Beschreibung durch Zylinderkoordinaten (Winkel definiert die Farbe)

# Farbräume (IV)

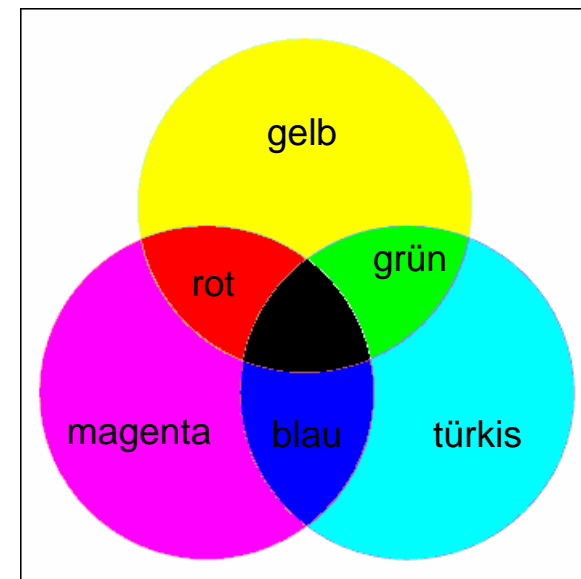
## Additiver Farbraum

- Grundfarben addieren sich zu Weiß
- Bsp: Displays oder Beamer nutzen die Grundfarben RGB



## Subtraktiver Farbraum

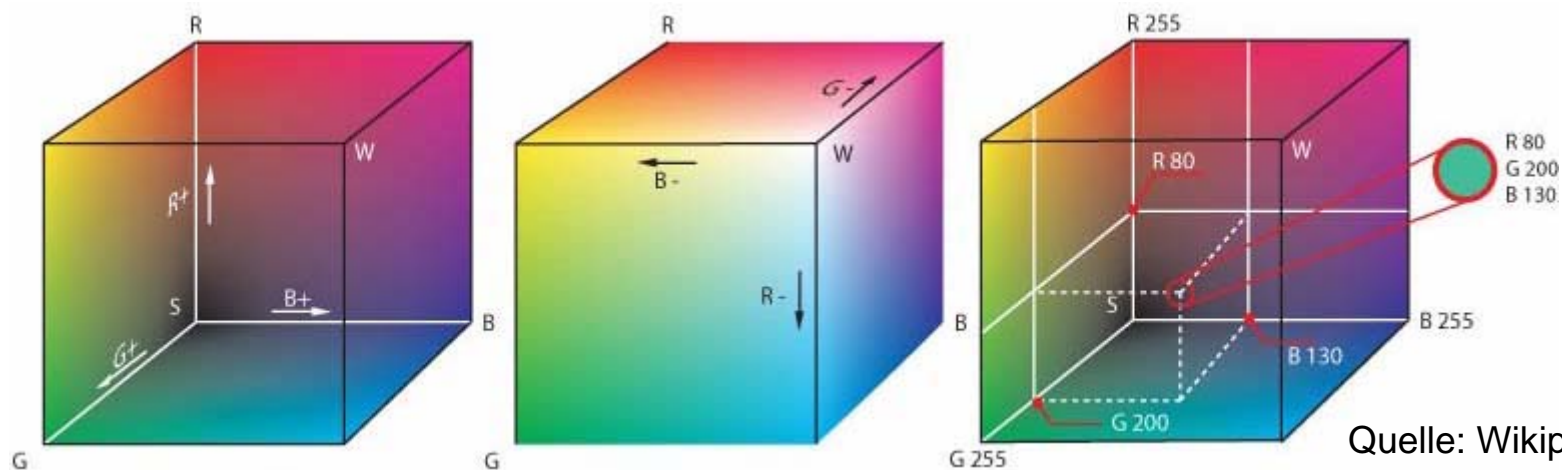
- Von Weiß werden Grundfarben subtrahiert
- Bsp: Tintenstrahldrucker (Pixel absorbieren weißes Licht), Folien vor weißer Lampe filtern einzelne Farbkomponenten



# Farbräume (V)

## RGB-Farbraum:

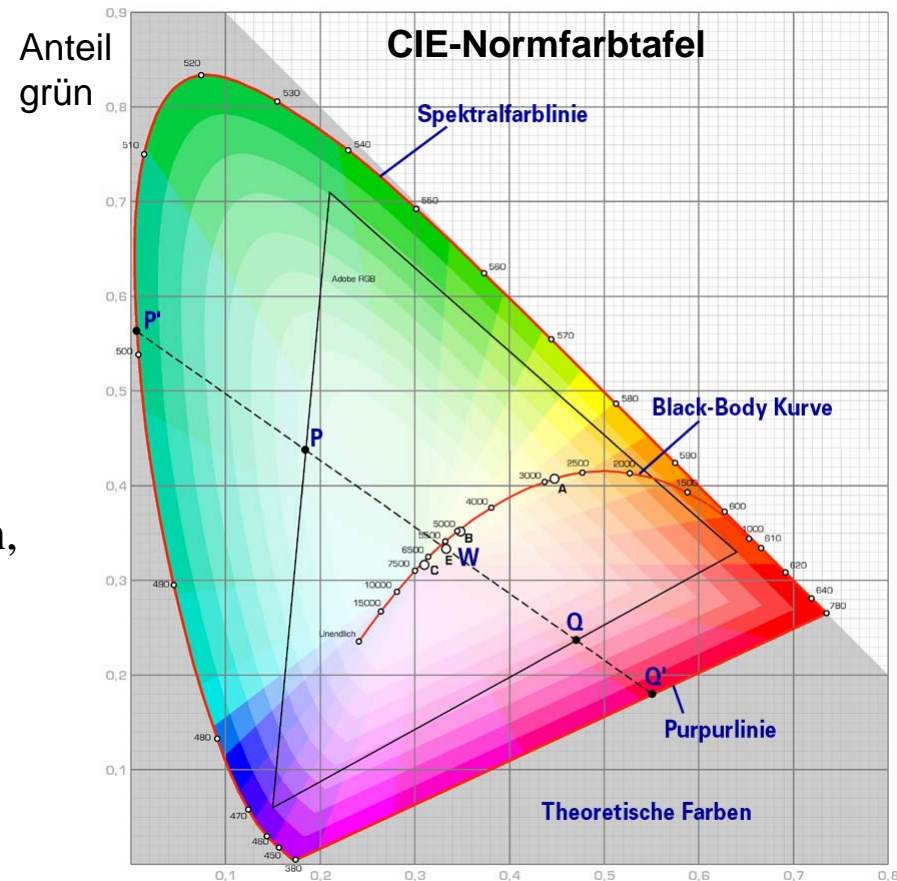
- Idee: aus farbigem Licht dreier Primärfarben kann man beliebige Farben mischen.
- Additiver Farbraum (Farben addieren sich zu weiß)
- 8 Bit / 16 Bit pro Farbkanal



# Farbräume (VI)

## RGB-Farbraum:

- Der RGB-Farbraum ist parabelförmig begrenzt (theoretische Farben).
- Reine (vollständig gesättigte) Farben liegen auf der Spektralfarblinie.
- Innerhalb des Diagramms liegen Mischfarben.
- Mittlerer Weißpunkt: W
- Alle Mischfarben, die durch Kombination zweier Farben entstehen, liegen auf einer Geraden definiert durch die beiden Farben.
- Drei Farben spannen ein Dreieck auf (RGB-Farbraum auf dem Computer wird durch schwarzes Dreieck beschrieben).
- Ecken des Farbraums werden durch Darstellungsmöglichkeiten von Monitoren definiert.



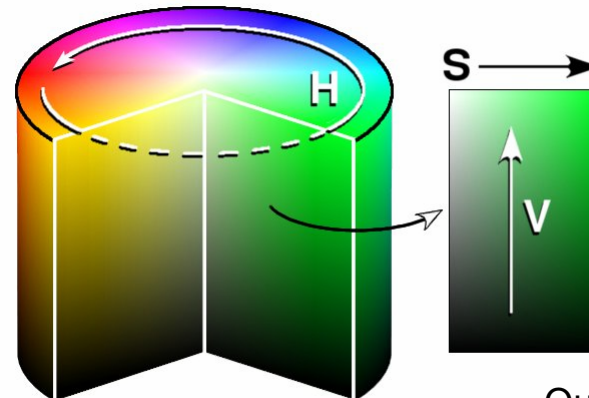
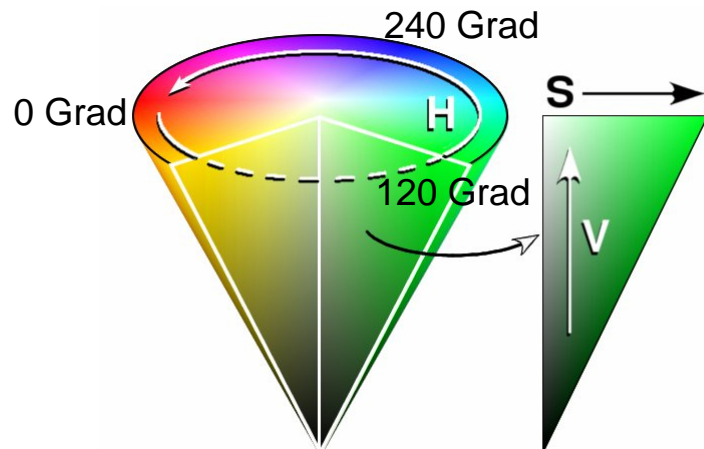
Anteil rot + Anteil grün + Anteil blau = 1    Anteil rot  
 → Anteil blau = 1 - Anteil rot - Anteil grün

Quelle: Wikipedia.de

# Farbräume (VII)

## HSV-Farbraum:

- Hue (Farbe): dominante Wellenlänge (dominante Farbe, mit der ein Mensch ein Objekt beschreibt)
- Saturation (Sättigung): Beschreibt wie kräftig eine Farbe ist, d.h. das Mischungsverhältnis einer Farbe mit Weiß (Welche Wellenlängen sind neben der dominanten Wellenlänge vorhanden?).
- Value (Intensität/Farbwert): Helligkeit einer Farbe



Quelle: Wikipedia.de

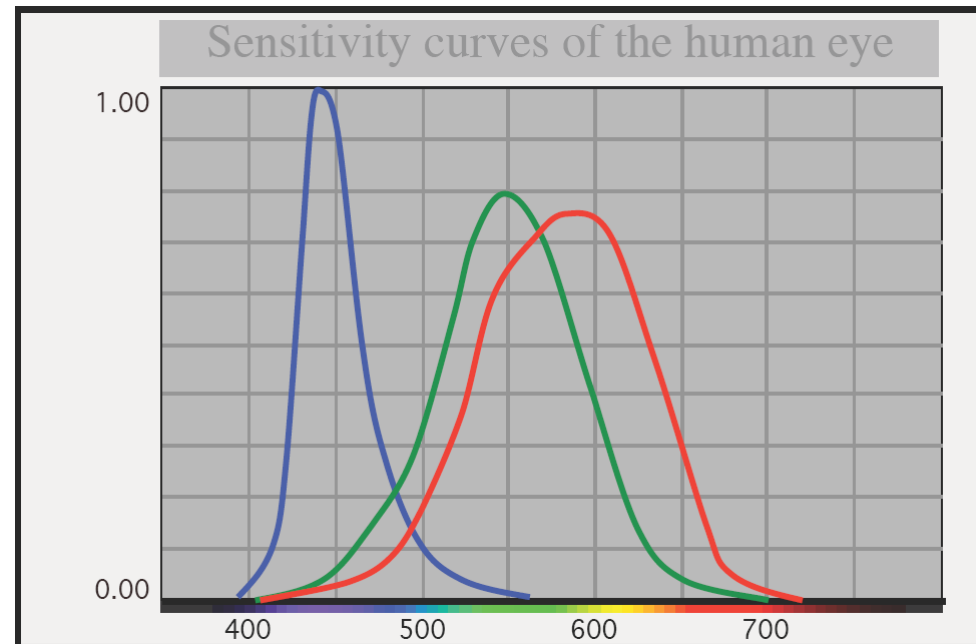
# Menschliche Wahrnehmung (I)

**Die Netzhaut des menschlichen Auges besteht aus:**

- Zapfen (cone cells): Erkennung von Farben
- Stäbchen (rod cells): Erkennung von Helligkeitswerten

**Zapfen enthalten drei Arten von Farbpigmenten, die sensitiv gegenüber bestimmten Farben sind:**

- niedrige Frequenzen: blaue Farbtöne
- mittlere Frequenzen: grüne/gelbe Farbtöne
- hohe Frequenzen: rote Farbtöne

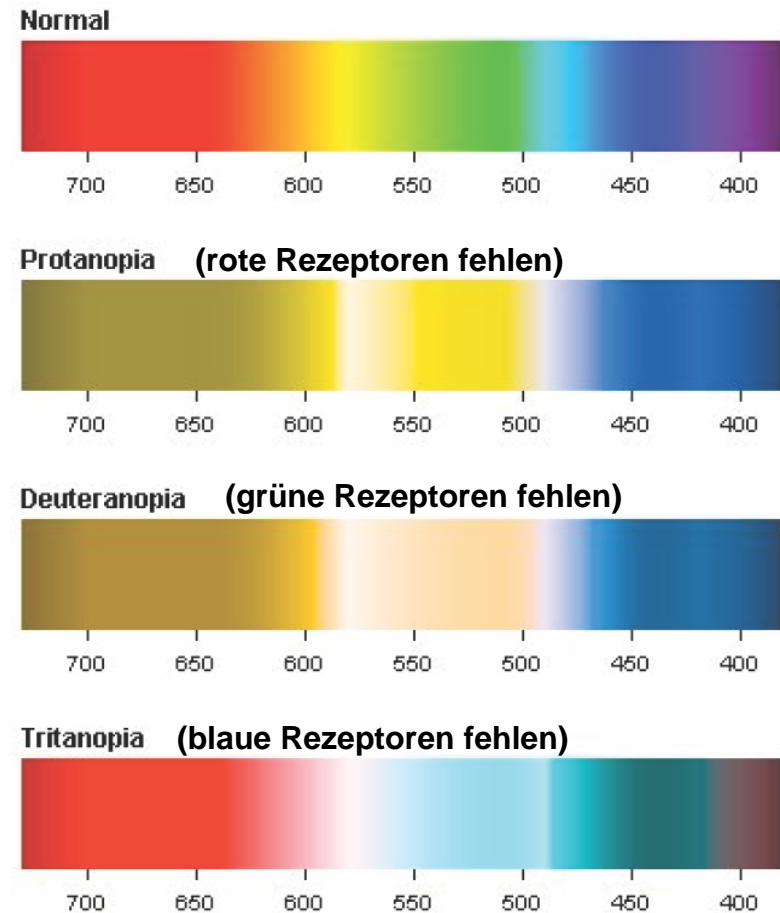


Quelle: Gabriel Marcu, Apple



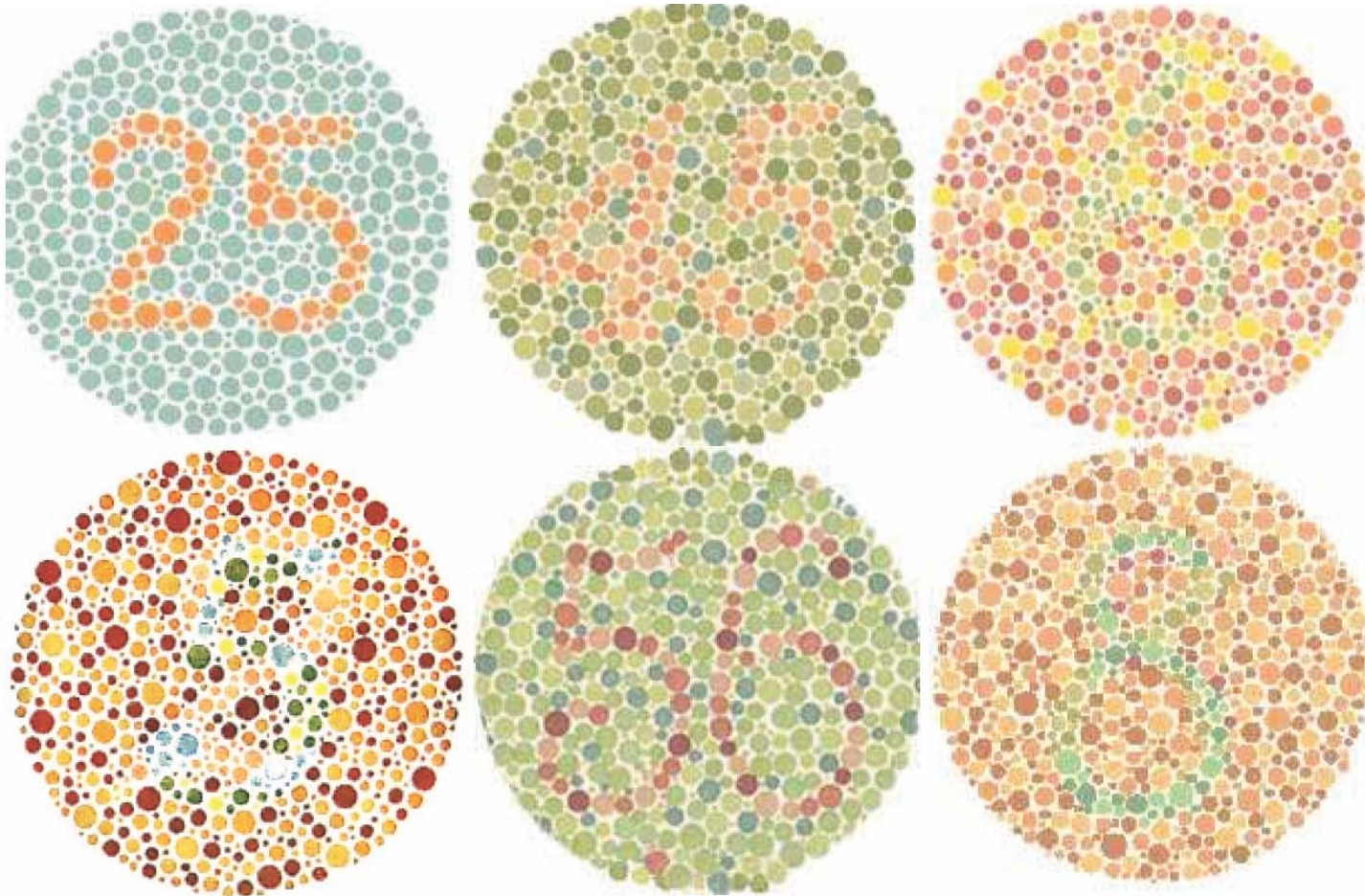
# Menschliche Wahrnehmung (II)

- Falls eins der Farbpigmente keine vollständigen Signale übermittelt, spricht man von Farbenblindheit
- 10 % der Bevölkerung können Farben nur eingeschränkt wahrnehmen



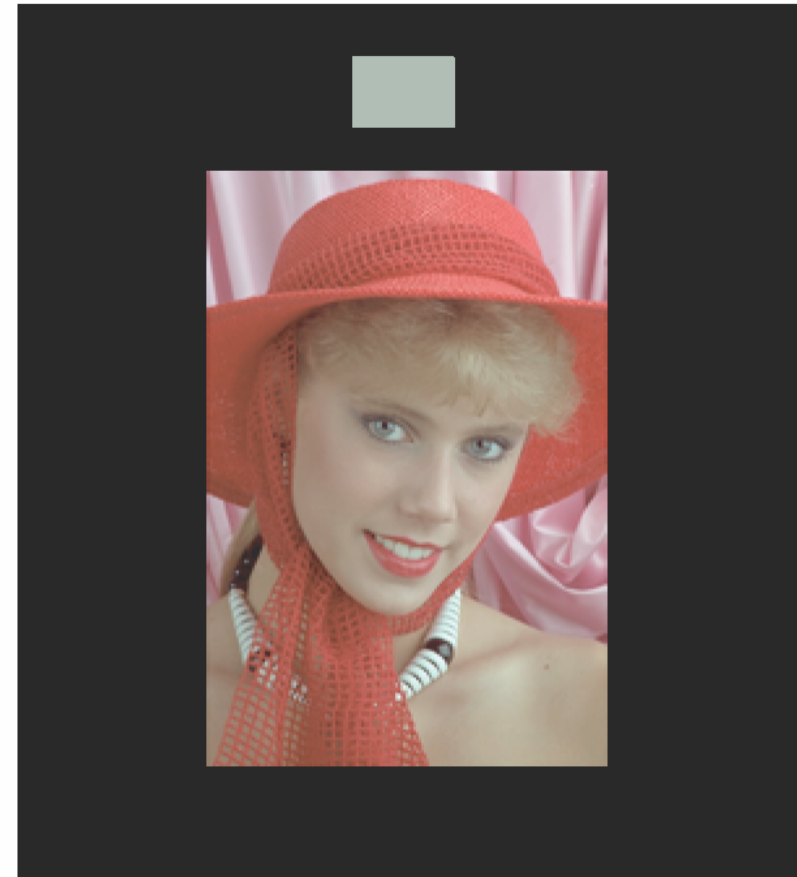
Quelle: Gabriel Marcu, Apple

# Menschliche Wahrnehmung (III)



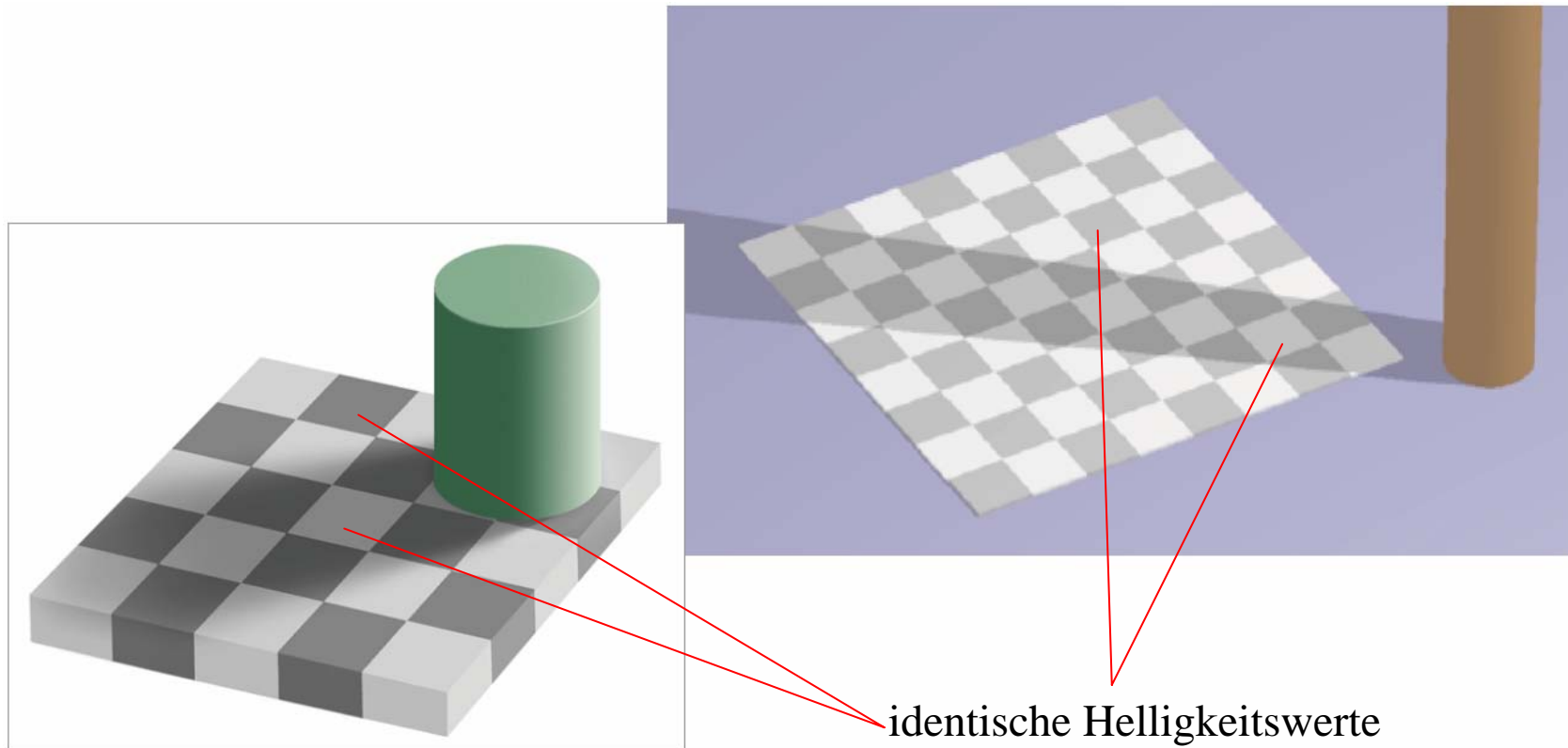
Ishihara-Test zur Erkennung von Farbenblindheit Quelle: Gabriel Marcu, Apple

# Menschliche Wahrnehmung (IV)



Farben werden durch ihre Umgebung beeinflusst Quelle: Gabriel Marcu, Apple

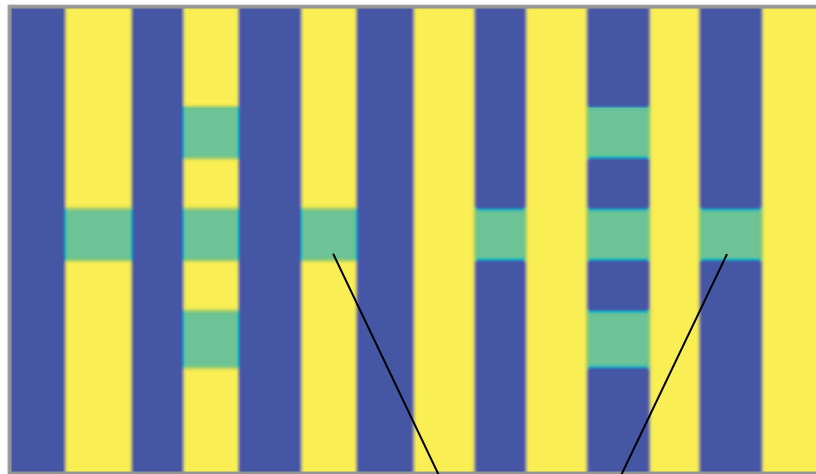
# Menschliche Wahrnehmung (V)



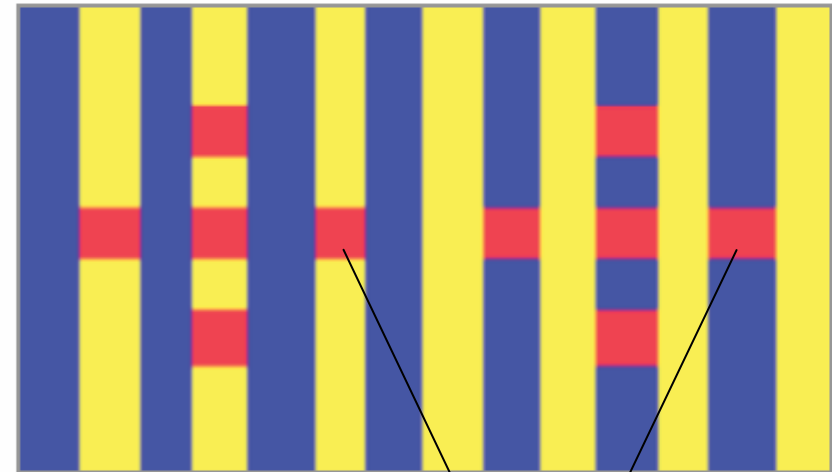
Helligkeiten werden durch Formen, Objekte und deren Schatten beeinflusst

Quelle: Gabriel Marcu, Apple

# Menschliche Wahrnehmung (VI)



identische Farbe



identische Farbe

Farben werden durch benachbarte Farbreionen beeinflusst

Quelle: Gabriel Marcu, Apple

# Objekterkennung mittels Farben (I)

## **Erkennung von Farben beim Menschen**

1. Signale der drei Farbpigmente der Netzhaut
2. Psycho-physikalische Einflussfaktoren (Umgebung, absolute Helligkeit, Textur, Form, Rauschen, Größe)
3. Psychologische Einflussfaktoren (Erinnerung an Farben, Farbpräferenzen)

→ menschliche Farbwahrnehmung

## **Erkennung von Farben durch den Rechner**

1. Sensoren (Scanner, Kamera, Messgeräte)
2. Farbkorrektur, Anpassung der Helligkeit
3. Analysealgorithmen

→ Computergestützte Farbwahrnehmung

# Objekterkennung mittels Farben (II)

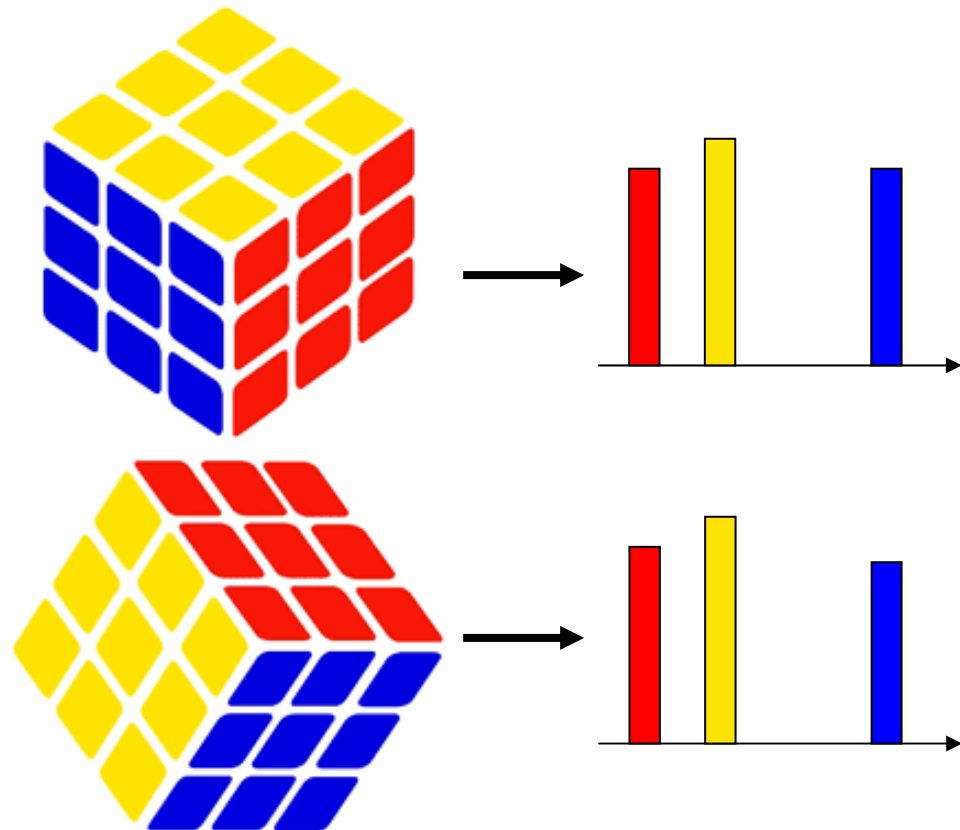
## Ablauf:

- Auswahl eines Farbraumes
- Abbildung der Pixel eines Objektes auf Merkmalswerte
  - Histogramme
  - Dominante Farbe
- Vergleich mit Merkmalswerten bekannter Objekte durch ein Entfernungsmaß

# Objekterkennung mittels Farben (III)

## Histogramme:

- Beschreibung eines Objektes durch Anzahl der Pixel der einzelnen Farben
- Vergleich zweier Histogramme mittels  $L_1$ - oder  $L_2$ -Norm
- Zuverlässige Ergebnisse bei affinen Transformationen des Objektes

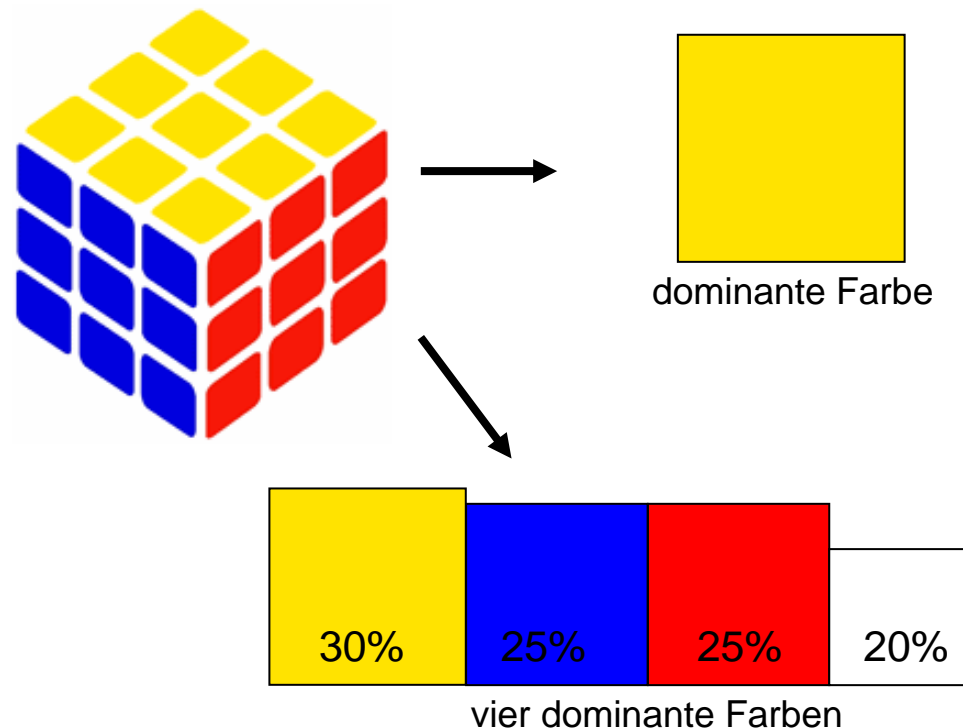




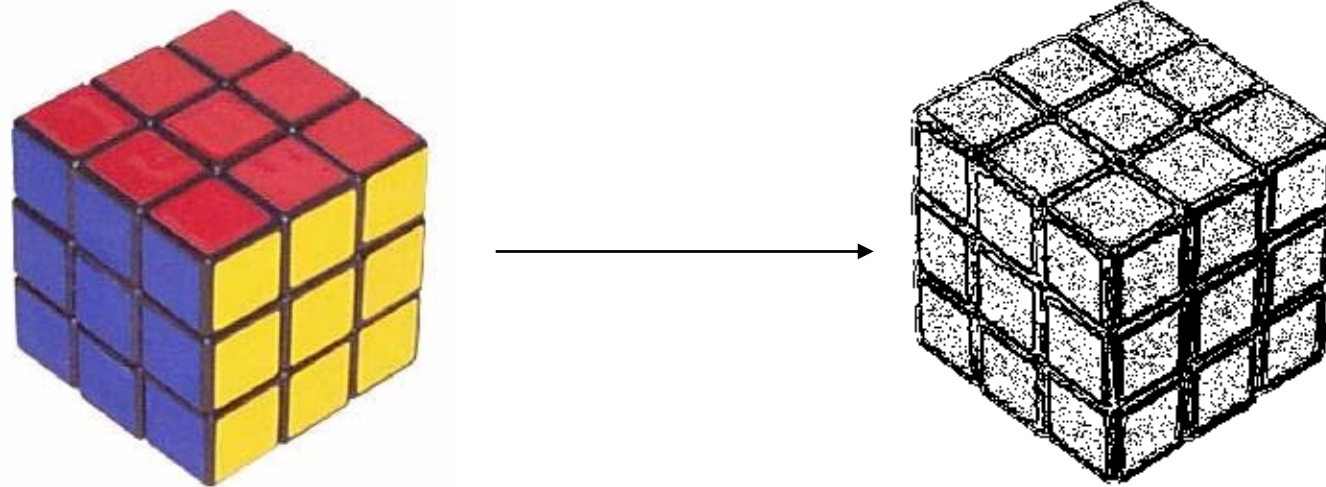
# Objekterkennung mittels Farben (IV)

## Dominante Farbe:

- Annahme: Die am häufigsten in einem Bild enthaltene Farbe liefert charakteristische Informationen.
- Problem, falls mehrere Farben ähnlich häufig auftreten
- → Beschreibung eines Objektes durch wenige dominante Farben und Anteil jeder Farbe



# Objekterkennung durch Vergleich von Texturen



# Objekterkennung mittels Texturen (I)

## Textur:

- Visuelle Muster, die durch Farben bzw. Helligkeiten in einem Bild entstehen.
- Entstehen durch Reflektion des Lichtes an einer Oberfläche (Gras, Holz, Metall, Stoff, aber auch Wolken)
- Enthalten Informationen über die Struktur der Oberfläche



# Objekterkennung mittels Texturen (II)

## Homogener Texturdeskriptor

- Beschreibt Richtung, Unebenheit und Regelmäßigkeit einer Textur
- Gut geeignet zur Beschreibung homogener Flächen
- Die im Bild enthaltenen Frequenzen werden ermittelt und deren Durchschnitt und Standardabweichung wird berechnet.
- Invariant gegenüber Skalierung und Rotation

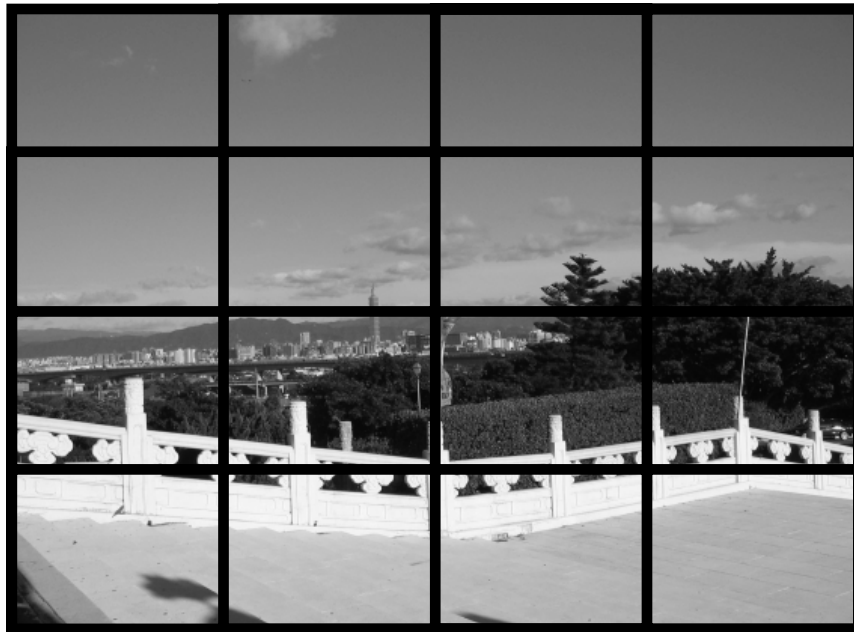
# Objekterkennung mittels Texturen (III)

## **Kantenhistogramm (edge histogram)**

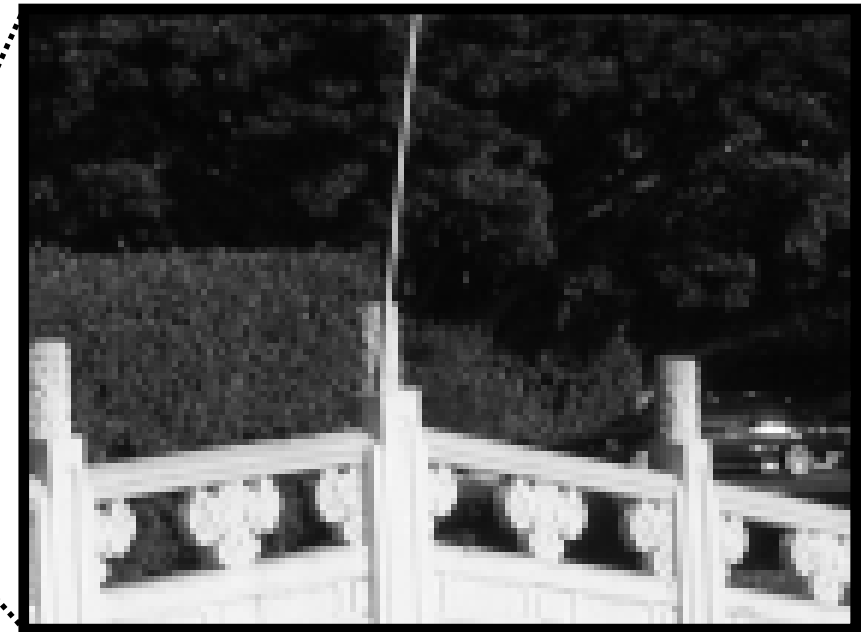
- Beschreibt nicht-homogene Texturen
- Beschreibt lokale (in einem Bildbereich) Verteilung von Kanten
- Einteilung des Bildes in 16 gleichgroße Blöcke
- Berechnung von Kanten für jeden Block
- 5 Typen von Kanten: vertikal, horizontal, 45 Grad, 135 Grad, ungerichtet
- Speicherung der Werte in einem Histogramm für jeden Typ und jeden Block ( $5 \times 16 = 80$  Elemente)
- Skalierungsinvariant
- Bei einem Vergleich kann (muss aber nicht) die Rotation berücksichtigt werden.
- In MPEG-7 werden nur 3 Bits zur Beschreibung eines Histogrammelementes verwendet (insgesamt 240 Bits zur Beschreibung der Textur).

# Objekterkennung mittels Texturen (IV)

## Berechnung des Kantenhistogramms



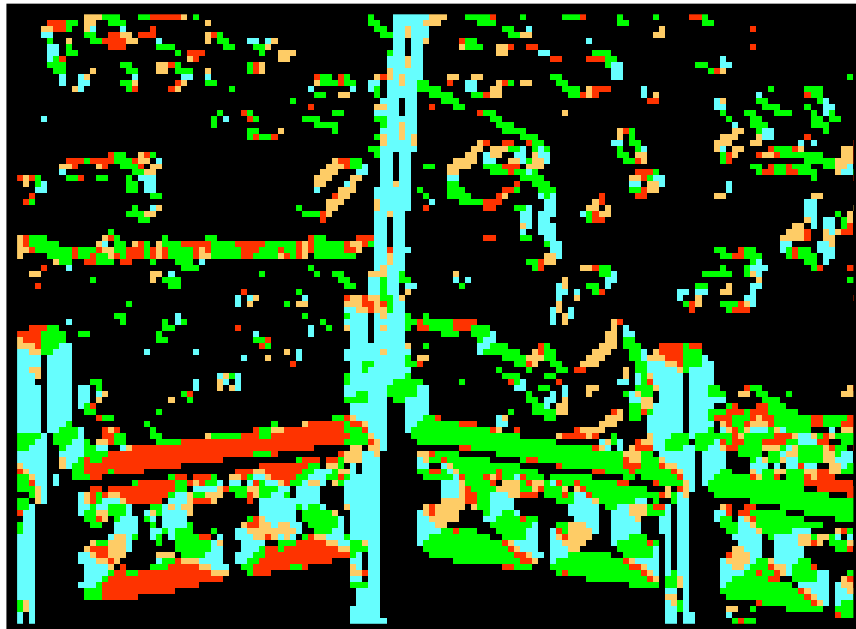
Einteilung des Bildes  
in 16 Regionen



Berechnung der Kanten  
für jede Region

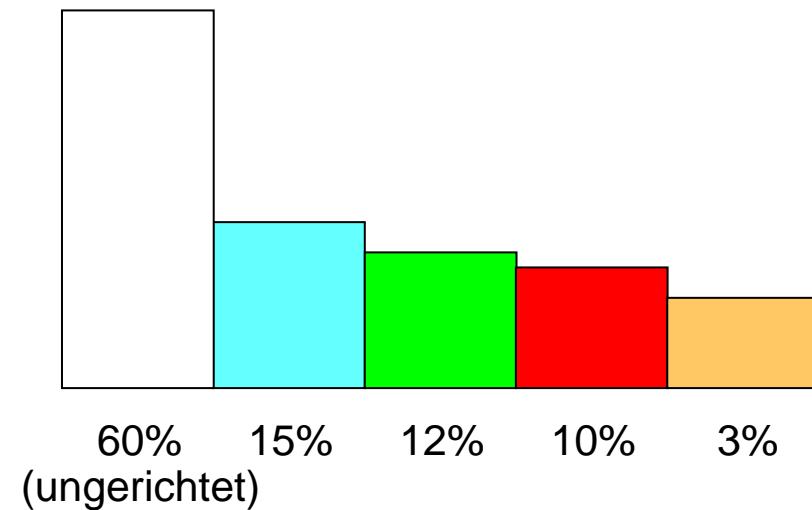
# Objekterkennung mittels Texturen (V)

## Berechnung des Kantenhistogramms



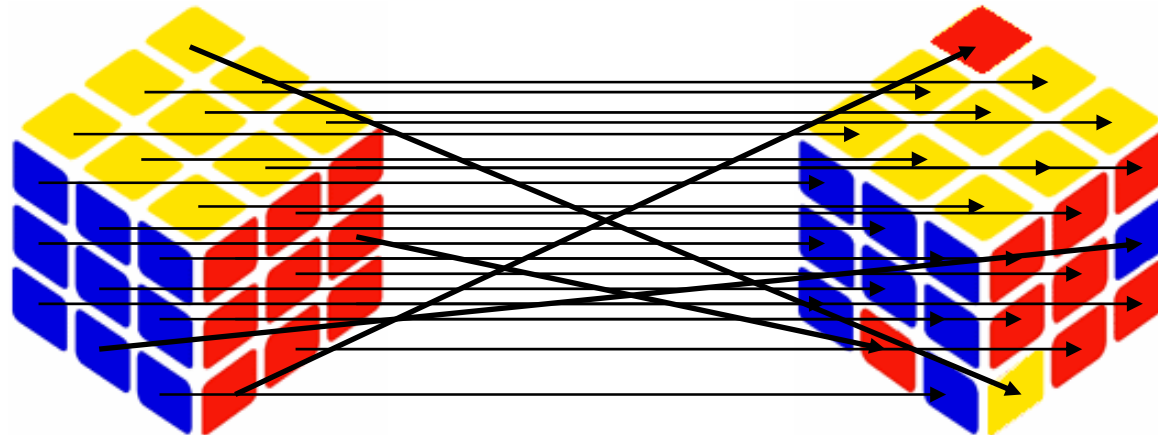
0 Grad, 45 Grad, 90 Grad, 135 Grad

Berechnung des  
Kantenanstiegs



Berechnung der  
Histogramme

# Objekterkennung durch Vergleich von Bewegungen





# Objekterkennung durch Bewegungsanalyse (I)

## **Bewegungsaktivität (motion activity)**

- Beschreibung der Bewegung in einem Videosegment
- Grobeinteilung in Kategorien:
  - langsam: Fernsprecher
  - schnell: Strassenszene
  - rasend: Fußball, Basketball
- Standardabweichung der Längen der Bewegungsvektoren
- Standardabweichung wird einer von 5 Kategorien zugeordnet
- Optionale Parameter:
  - Richtung der Bewegung
  - Bewegungsaktivität einer Bildregion
  - Zeitliche Dauer einer Bewegung

# Objekterkennung durch Bewegungsanalyse (II)

## **Bewegungsbahn (motion trajectory)**

- Beschreibung der Bewegung eines Bildbereiches in einem Videosegment
- Ähnlich wie Bewegungsaktivität, nur können mehrere Bewegungen beschrieben werden
- Beispiel: Verkehrsüberwachung
  - bei einer Verkehrsüberwachung wird für jedes einzelne Fahrzeug und jede Person im Bild eine Beschreibung der Bewegung gespeichert
  - Suchanfrage liefert Objekte in der Nähe
- Beispiel: Überwachung eines öffentlichen Platzes
  - Suchanfrage liefert Personen, die sich nach einem bestimmten Muster bewegen (z.B. sich langsam bewegen und plötzlich losrennen)

# Zusammenfassung

- Modellbasierte Objekterkennung basiert auf dem Vergleich aus dem Bild extrahierter Merkmale mit einem Modell des gesuchten Objektes
- Die Wahl des Objektmodells orientiert sich an dem zu erkennenden Objekt sowie den zur Verfügung stehenden Daten.
- Erkennung von Objekten durch Vergleich von:
  - Konturen (Kompaktheit, Exzentrizität, Krümmungen)
  - Farben (Farbraum, menschliche Wahrnehmung, Histogramm, dominante Farbe)
  - Texturen (homogener Texturdeskriptor, Kantenhistogramm)
  - Bewegungen (Bewegungsaktivität, Bewegungsbahn)

# Zusammenfassung

- Was wären gute Modelle für folgende Objekte?



# Fragen ?