Chapter 5
Object Recognition

Distributed Algorithms
for Image and Video Processing

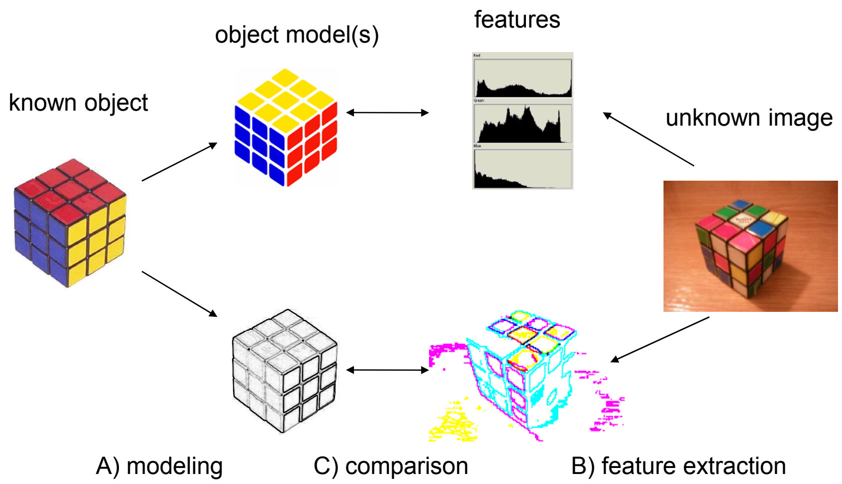
Overview

- Motivation
- Requirements
- Object recognition via
  - Colors
  - Shapes (outlines)
  - Textures
  - Movements
- Summary

Why object recognition?

- Character recognition
  - OCR-software for character recognition
  - Automatic zip code recognition
- Video surveillance
  - Identification suspicious persons (thievery, harassment)
  - Recognition suspicious objects (unattended suitcases at airports or railway stations)
- Content-based image search
  - internet search (e.g. Google images, MS Bing)
  - image- and video archive (e.g. Flickr, YouTube)

Model based object recognition



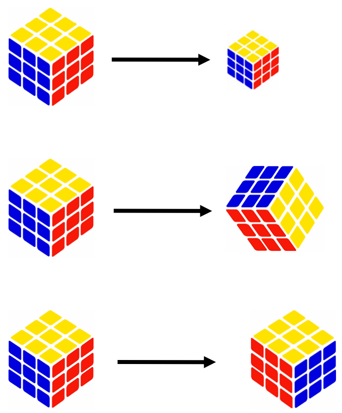
What is a good object model?

- *Sensitivity*: The model must illustrate all relevant features and be able to differentiate between them.
- *Uniqueness*: The model should be capable to describe objects as clearly as possible; similar objects are to have the same description.
- *Stability*: Slight modifications of the object are to have small effects on the model.
- *Efficiency*: It must be possible to efficiently calculate features out of data and to compare these.

Requirements (I)

The model should be invariant compared to:

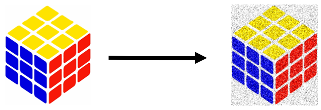
- **scaling**
- **rotation**
- **reflection**



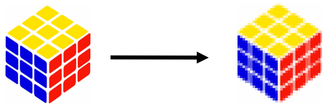
Requirements (II)

The model should be invariant compared to:

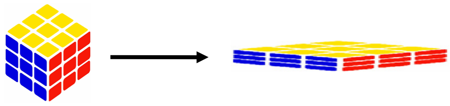
- noise



- blurring



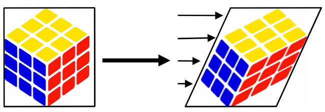
- anisotropic scaling (non-uniform)



Requirements (III)

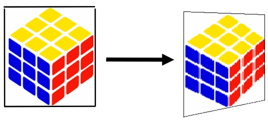
The model should be invariant compared to:

- shear

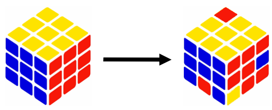


horizontal shear

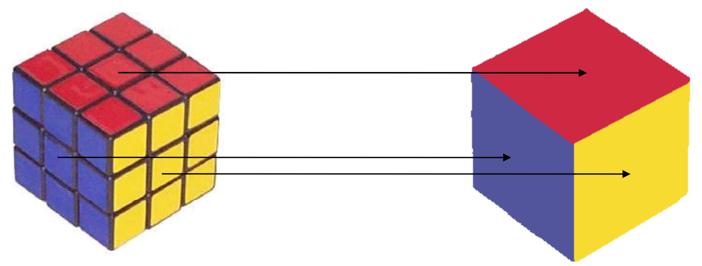
- perspective distortion



- object deformation



Object recognition by color



Color space (I)

Definitions:

- **Color:** Sense perception (no physical feature) in case light of a certain wavelength falls on the retina of the eye. The sensory cells in the human eye (cones for colors, rods for brightness value) release impulses to the brain, which perceives these as colors. Human possesses three types of cones (sensitive for red, green, and blue).
- **Visible light:** between 400 – 700 nm

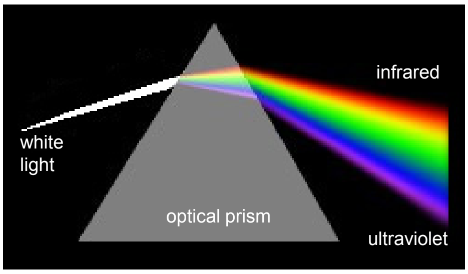
Source: Wikipedia.de

- **Color space:** amount of colors
- **Pigment:** color space of the human eye
- **Color model:** Describes the color space which the sense of light or an input or output device (display, scanner, printer, projector, photo, camera, television) is able to illustrate.

Color space (II)

Basis of colors:

- discovery 1666 by Newton: sunlight which falls through a prism is split up into a continuous range of colors.



red
orange
yellow
green
blue
violet

Color space (III)

Composition of a color space:

- Coordinate system in which single colors define the axes (due to the human eye's construction mostly 3 dimensions).
- Physical color models (mixture of colors): RGB, CMYK
  - arrangement as dice
  - modification of a color → simultaneous modification of brightness, chroma, and coloring
  - In 1931, definition of the elementary colors by the CIE (International Commission on Illumination): red (=435,8 nm), green (=546,1 nm), and blue (=700 nm)
- Perception oriented color models (description by brightness, chroma, and coloring): HSV, HSI
  - description by cylindrical coordinates (angle defines the color)

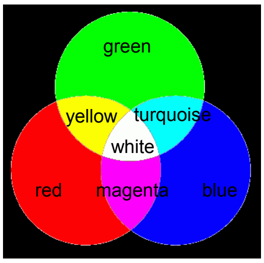
Color space (IV)

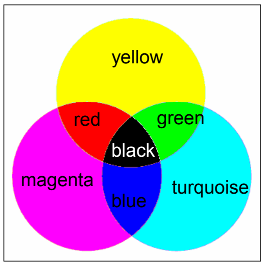
Additive color space

- Elementary colors add up to white
- E.g.: Displays or data projectors use the elementary colors RGB

Subtractive color space

- Elementary colors are subtracted from white
- E.g.: ink jet printer (pixel absorb white light), slides in front of white lamps filter single color components

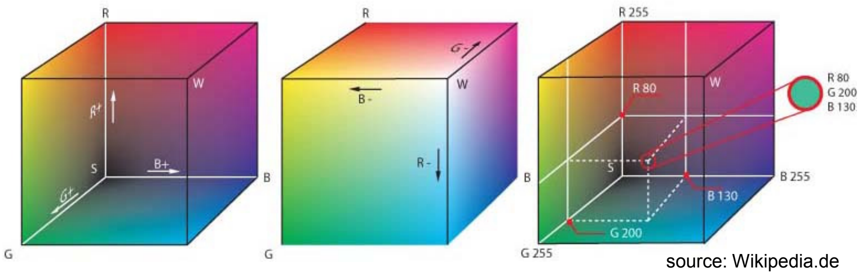




Color space (V)

RGB color space

- idea: out of the colored light of three elementary colors one can mix arbitrary colors
- additive color space (colors add up to white)
- 8 bit / 16 bit per color channel

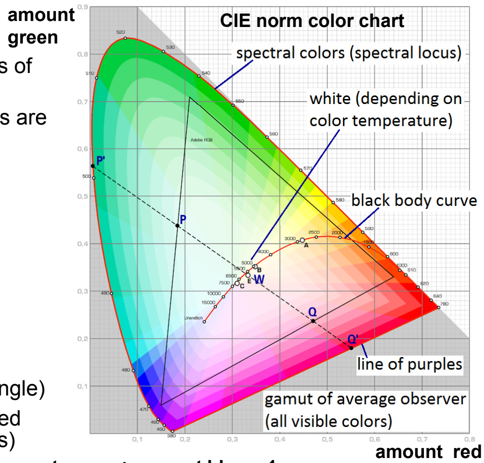


source: Wikipedia.de

Color space (VI)

RGB color space

- parabolically limited (visible colors of an average human observer)
- pure (completely saturated) colors are on the spectral color line (locus)
- mixed color: in the interior of the diagram
- white: several colors with similar intensities
- combination of two colors: all combinations on a straight line between these two colors can be created
- three colors span the RGB color space of the computer (black triangle)
- corners of the triangle are specified by different devices (e.g. monitors)



amount red + amount green + amount blue = 1
→ amount blue = 1 – amount red – amount green

source: Wikipedia.de

Color space (VII)

Color spaces on displays

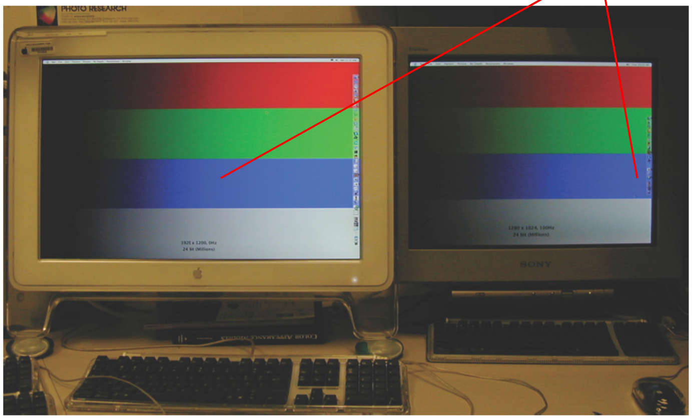
- The two displays do cover a very similar color space; but they display different maximum luminance values.
- The device with lesser luminance seems to cover a lesser color range.

source: Gabriel Marcu, Apple.

Color space (VIII)

Comparison of CRT and TFT displays

identical colors



source: Gabriel Marcu, Apple.

Color space (IX)

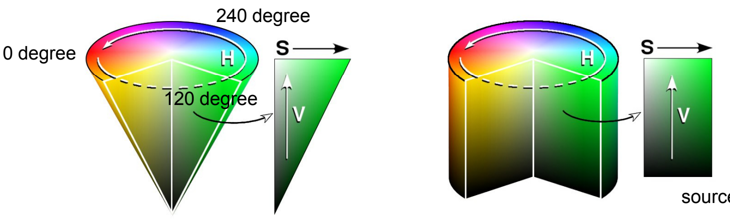
Comparison of CRT and TFT displays (same color ranges)

source: Gabriel Marcu, Apple.

Color space (X)

HSV color space

- Hue (color): dominant wavelength (dominant color by which a human describes an object)
- Saturation (chroma): describes how strong a color is, i.e. the mixture ratio of one color with another color (which wavelengths do exist among the dominant wavelength?).
- Value (intensity/color value): lightness of a color



source: Wikipedia.de

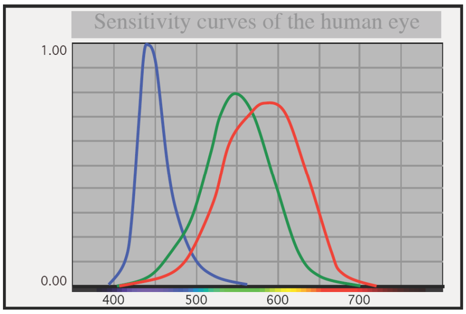
Human Perception (I)

The retina of the human eye does consist of:

- Cones (cone cells): Identification of colors
- Rods (rod cells): Identification of brightness values

Cones contain three types of pigments which are sensitive to specific colors:

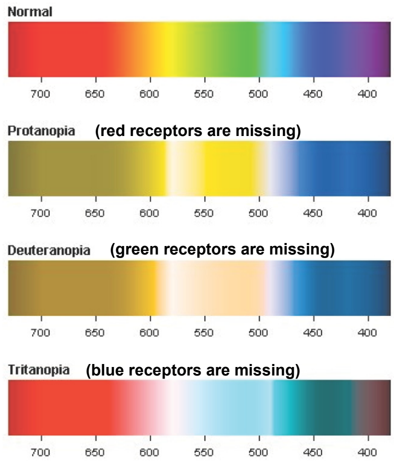
- Low frequencies: blue colorings
- Median frequencies: green/yellow colorings
- High frequencies: red colorings



source: Gabriel Marcu, Apple

Human Perception (II)

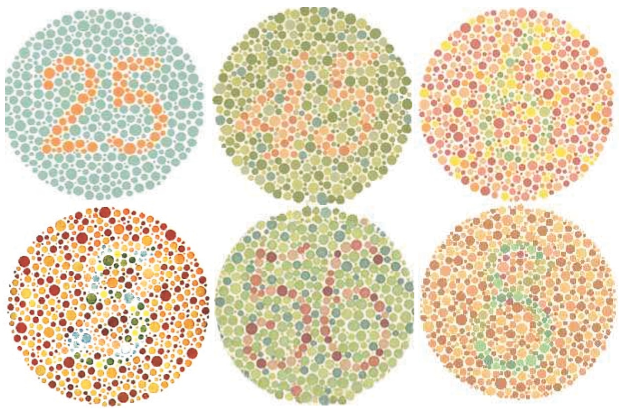
- In case that one of the pigments transfers no complete signals, one speaks of color blindness.
- 10 % of the population can perceive colors only restrictively



source: Gabriel Marcu, Apple

Human Perception (III)

Ishihara test for the detection of color blindness



source: Gabriel Marcu, Apple

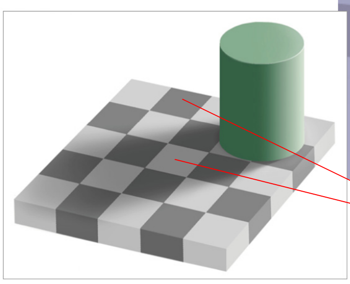
Human Perception (IV)

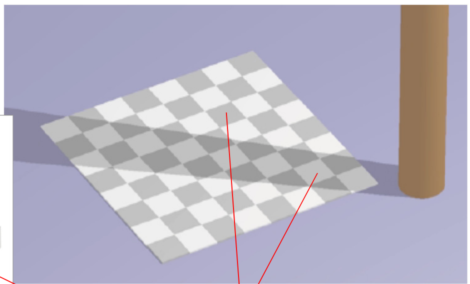
Perceived color is affected by its surroundings

source: Gabriel Marcu, Apple

Human Perception (V)

Luminance is affected by forms, objects, and their shades



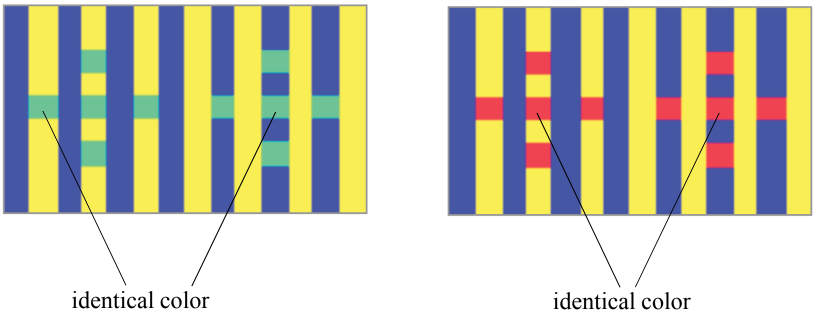


identical brightness values

source: Gabriel Marcu, Apple

Human Perception (VI)

Colors are affected by neighboring color regions



source: Gabriel Marcu, Apple

Object Recognition via colors (I)

Color recognition by humans

- Signals of the retina's three color pigments
- Psycho-physical factors of influence (environment, absolute brightness, texture, shape, noise, size)
- Psychological factors (correlations, feelings and preferences for colors)

Color recognition via computer

- Sensors (scanner, camera, measuring instruments)
- Color correction, adaptation of lightness
- Analysis algorithms

Object Recognition via colors (II)

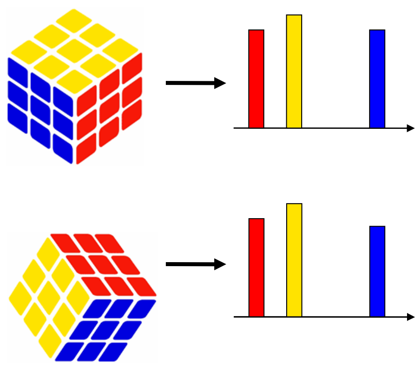
Process

1. Selection of a color space
2. Mapping of an object's pixels onto characteristic values
   – histogram
   – dominant color
3. Comparison with characteristic values of known objects by the use of a distance metric

Object Recognition via colors (III)

Histograms

- Description of an object through number of pixels of each color
- Comparison of two histograms via L1- or L2-norm
- Reliable results with affine transformations of the object

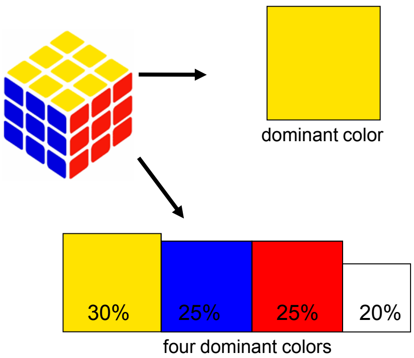


Object Recognition via colors (IV)

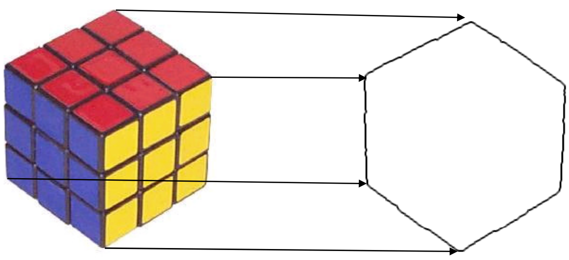
Dominant color

- **Assumption**: The color most present in a picture delivers characteristic information.
- Problems occur in case that multiple colors are present in similar frequencies

→ description of an object via few dominant colors and the amount of each color



Object Recognition via comparison of contours (shapes)



Process of Object Recognition

1. Parameterization (sampling) of contours with a predefined number of contour points
2. Identify characteristic points for the description of the contour
   – compactness
   – eccentricity
   – curvature scale space features
3. Compare the features with features of known objects (Shape Matching)

Parameterization of a contour

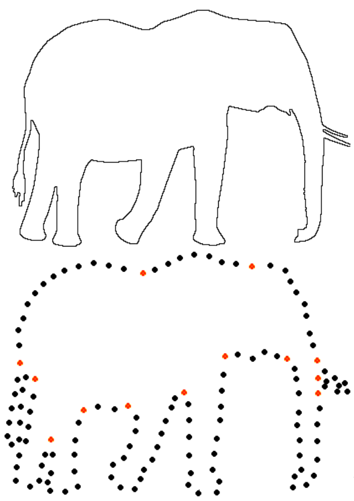
Segmentation

↓

Parameterization of contour

↓

Calculation of contour characteristics



Comparison of contours (I)

Compactness

- Global contour descriptor, which delivers an aggregated value for the complete contour (describes a contour's resemblance with a circle)
- Easy to calculate
- Only suitable to get a rough estimation of the resemblance of two contours
- Invariant against geometric transformations (rotation, scaling)
- The value for the compactness is minimal for a circle

Comparison of contours (II)

Compactness

- c_i: compactness
- i: segmented object
- U: contour's length
- F: object's area

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

- Varieties between two contours i and j

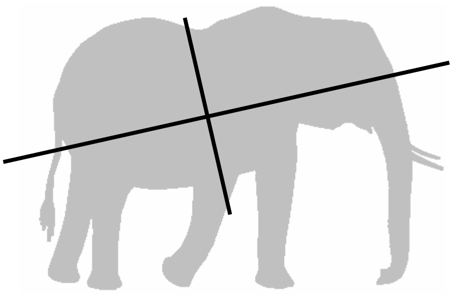
$$\alpha_c(i, j) = \frac{|c_i - c_j|}{\max(c_i, c_j)}$$

Comparison of contours (III)

Eccentricity

Relation of the two principal axis' lengths with reference to the central moments of the contour pixels

→ invariant against geometric transformations



Comparison of contours (IV)

Eccentricity

central moment of contour pixels

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

center of gravity of contour pixels

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

Comparison of contours (V)

Eccentricity

e_i: Eccentricity

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

Comparison of two objects *i* and *j*:

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

Comparison of contours (VI)

Curvature scale space

- Description of the curvature of an object's contour
- We get high peaks (characteristic values) in case of:
    - regions with high curvature
    - long curved regions
- An object's size shall not have an influence on the characteristic values

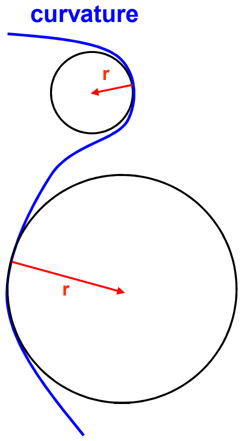
→ Comparison of contours via curvature scale space

Definition of the curvature (I)

- The **strength of a curvature** at one point corresponds to the reciprocal value of the radius of a clinging circle (the circle touches the curve):

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

- The curvature is a vector which points at the direction of the center of the circle
- A small circle represents a high curvature; a straight line has a curvature of zero



Definition of the curvature (II)

- A planar curve ***u(t)*** is given within a 2D space. ***u(t)*** is parameterized through the arc length ***t***.
- The curve ***u(t)*** is defined through the two functions ***x(t)*** and ***y(t)***:

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

- The curvature ***K*** for the curve ***u(t)*** is defined as:

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

$\dot{x}$ and $\dot{y}$ define the first derivative (gradient),
$\ddot{x}$ and $\ddot{y}$ define the second derivative (variation of the gradient)

Definition of the curvature (III)

Definition of the curvature for functions:

- Use explicit planar curves: ***y = f(x)***.
- Curvature in point ***(x, f(x))***:

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

Definition of the curvature (IV)

Example

- Exemplary function ***u(t) = (x(t), y(t)) = (t, t²)***
  Explicit definition of the function: ***y = f(x) = x²***
- Curvature based on the **parameterized curve**:
  First and second derivative: $\dot{x}$ = 1, $\ddot{x}$ = 0, $\dot{y}$ = 2t, $\ddot{y}$ = 2

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

- Curvature based on the **explicit definition**:

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

Definition of the curvature (V)

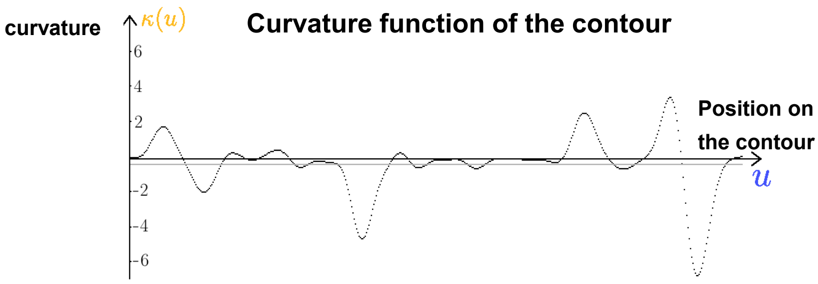
- Approximation of the derivative for discrete values (parameterized curve):

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

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

- The parameter *t* is defined for integral numbers t $(t \in \mathrm{N})$.
- *hx* and *hy* normalize the derivatives dependent on the distance of the scan points

Contour comparison by means of curvatures



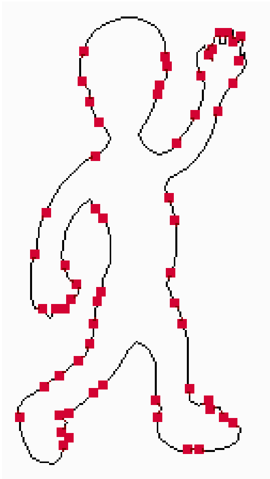
Problem: Two curvature functions can only be compared with each other with great difficulty

→ Identify significant points of the curvature function

→ the curvature-scale-space-method delivers good characteristic values

Curvature-Scale-Space-Method (I)

- Analyze outer contour of an object
- Smooth contour (Gaussian filter)
- Characteristic points: inflection points (turning points) of the contour



Curvature-Scale-Space-Method (II)

Process

1. Iterative smoothing of the contour
2. Calculation of the curvature
3. Consider zero crossings of the curvature function (corresponds to the contour's inflection points) :

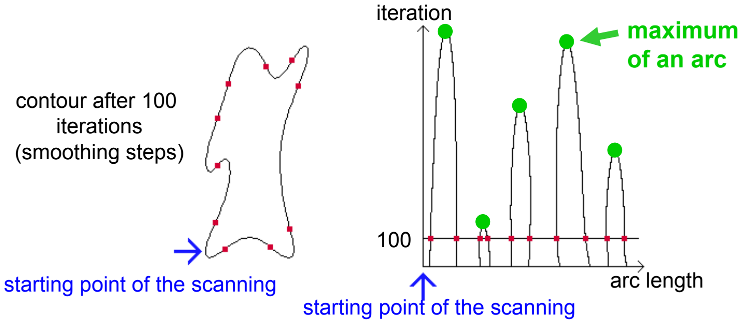
$$\kappa(u,n) = 0$$

4. Produce curvature-scale-space-mapping (curvature scale space image):

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

Curvature-Scale-Space-Method (III)

- A curvature scale space image (CSS image) visualizes the inflection points which result from the smoothing



The local maxima of the **arcs (peaks)** are used as characteristics for the object's description.

Curvature-Scale-Space-Method (IV)

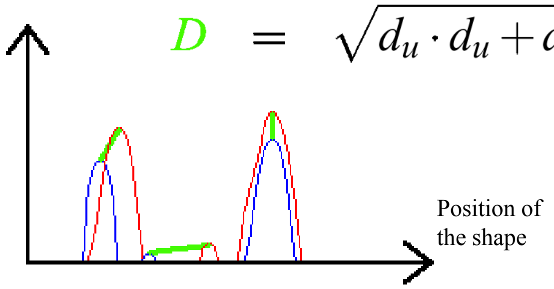
Characteristics of CSS images

- An arc (peak) in the CSS image describes a concave area of a contour.
- The arcs deliver the characteristics for a contour's description.
- Each arc is described by
  - a position:
    relative position in comparison to the other arcs
  - a height (number of the required iterations to smooth an image):
    strength of a concave segment

Comparison of two contours

- Horizontal shift of two CSS images until larges peaks match (makes technique invariant to rotations)
- Calculate Euclidian distance between two local maxima of the scale space images
- Summarize distances:

$$D = \sqrt{d_u \cdot d_u + d_t \cdot d_t}$$

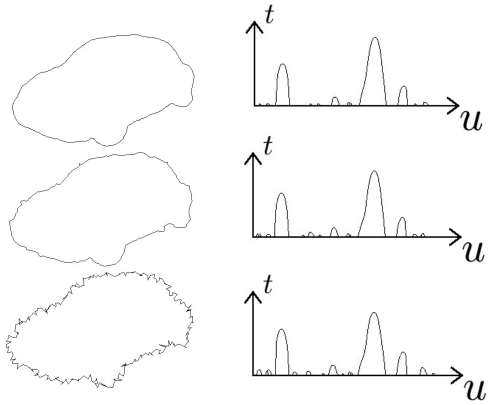


Features of CSS images (I)

Original contour

Noisy contour

Contour with significant noise



→ **Even in case of severe noise, both CSS images are very similar.**

Features of CSS images (II)

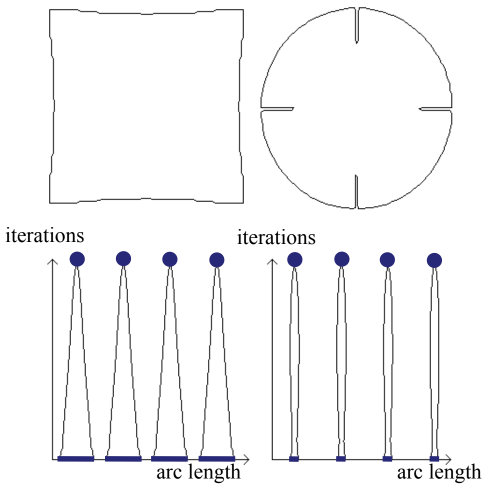
Advantages

- Good classification results
- Few values are required to describe complex objects
- Invariant to rotations and scaling
- Robust in case of noise or perspective transformations
- Fast calculation of features
- Very fast comparison of two shapes

Disadvantages

- Bad classification results for some specific contours (ambiguities, convex object regions)

Ambiguities of CSS images (I)



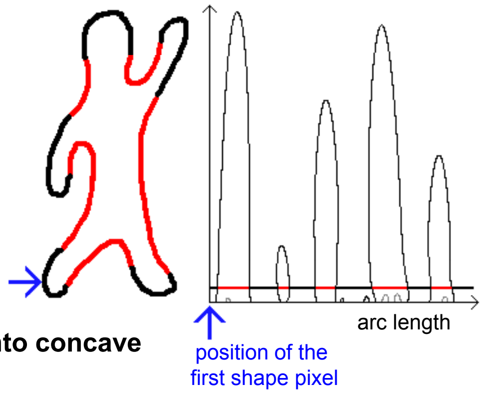
- The height of a peak describes the length of a curved region and the strength of its curvature.

→Therefore, we store the **width** of each peak.

Ambiguities of CSS images (II)

Convex regions

- Bad representation of convex regions of a contour
- Convex objects cannot be represented at all !

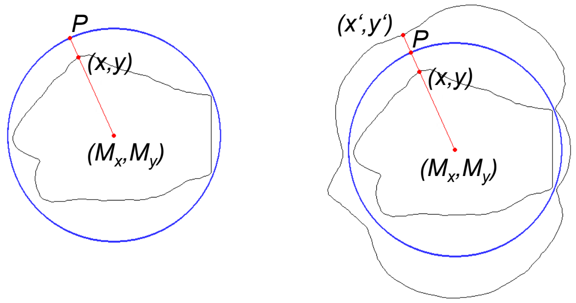


→ **Solution: Convert convex regions into concave regions and vice versa**

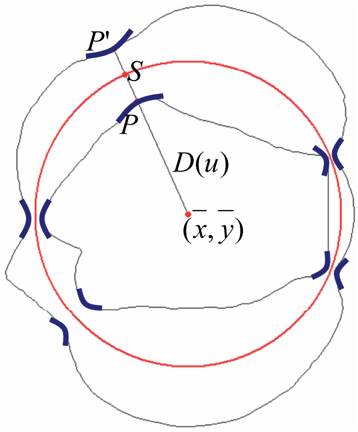
→ **Technique: Mirror the shape**

Reflected Contours (I)

- **Idea**: Create a new contour through the reflection of the contour pixels at a surrounding circle.
- Strongly convex areas of the original contour turn into concave areas in the reflected contour.



Reflected Contours (V)



$(\bar{x}, \bar{y})$ center of gravity of contour pixels

P contour pixel ($x(u)$,$y(u)$)

$D(u)$ distance to center

S point on circle line

P' reflected pixel (x'(u), y'(u))

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

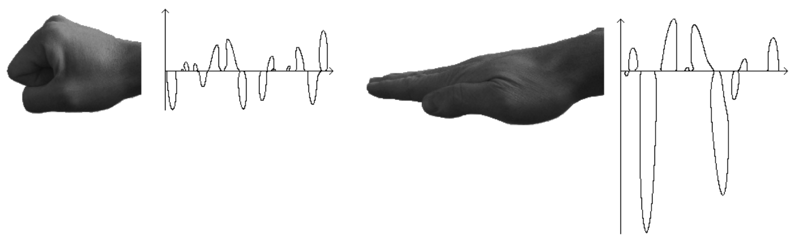
$$D(u) = \sqrt{(\bar{x} - x(u))^2 + (\bar{y} - y(u))^2}$$

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

Reflected Contours (VI)

Extended contour comparison

- Calculate original characteristics of CSS-images
- Calculate characteristics for reflected contour



Object Recognition in Images (I)

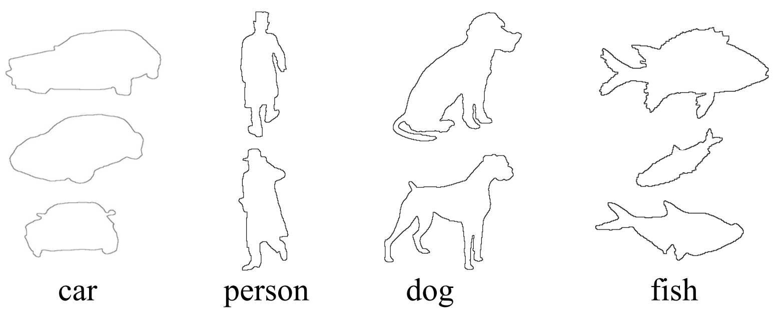
Approach

1. Compare unknown object with known objects of a data base
2. Calculate the minimum difference between unknown object and all objects of a *class* (aggregation of similar objects)
3. Return the name of the most similar object class

Object Recognition in Images (II)

Objects of the reference database

- Database includes more than 300 objects
- 13 object classes group similar objects



Object Recognition in Images (III)

Criteria for the selection of images for the database: Canonical views

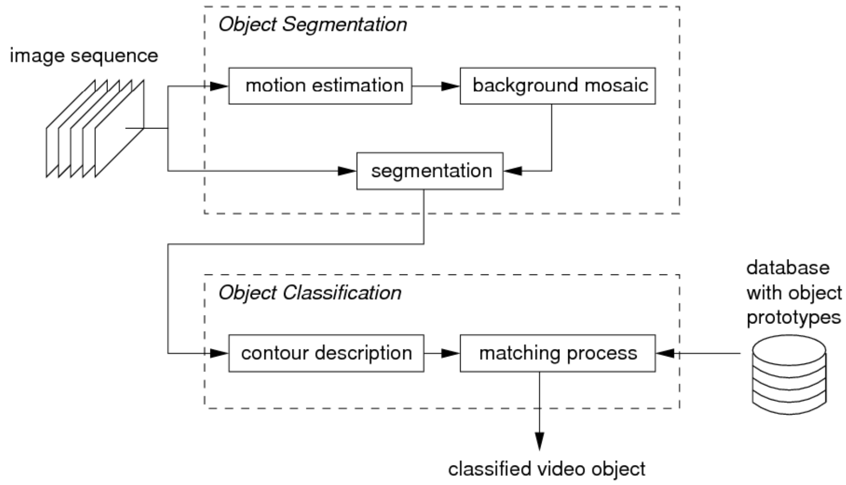
- How are objects represented in the human brain?
  → unknown
  → **but**: three-dimensional objects seem to be mapped as two-dimensional views
- The rotation of an object to the camera has a high influence on whether at all and how quickly a human being recognizes an object
- Easily recognizable two-dimensional projections of a three-dimensional object are called *canonical views*

Object Recognition in Images (IV)

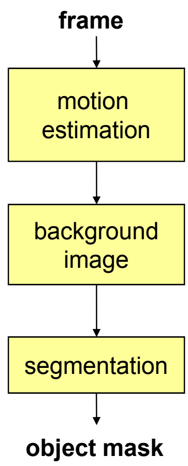
Examples for canonical views

- Views at profile or lightly raised views from the front
- Familiar perspectives: perspectives from which objects are usually observed or, in case of commodities, are utilized
- As objects of the reference database, objects are selected in canonical views

Object Recognition in Videos (I)

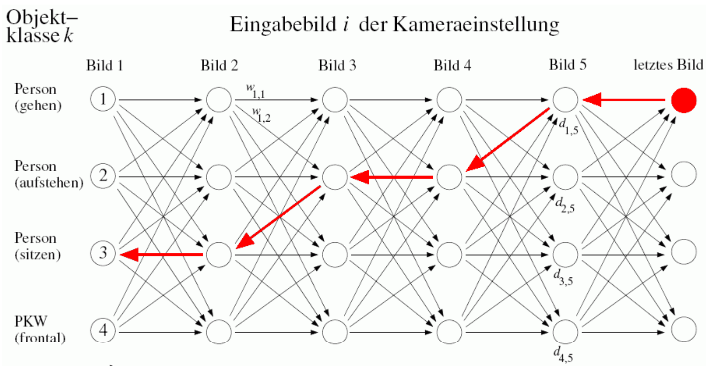


Object Recognition in Videos (II)



- **Assumption**: half of all pixels of an image are background pixels
- Calculate camera motion between two adjacent images of the video
- Align all pictures of a camera angle according to the camera model so that the image background is congruent
- Use median filter in order to create a background image out of the transformed pictures
- Compare the background image with the transformed frames and get segmented objects

Object Recognition in Videos (III)



transition between object classes: $w_{c_{i-1},\, c_i}$

node $d_{c_i, i}$ evaluates difference of the object i to the object class c_i

path with minimal total costs

last knot with minimal costs

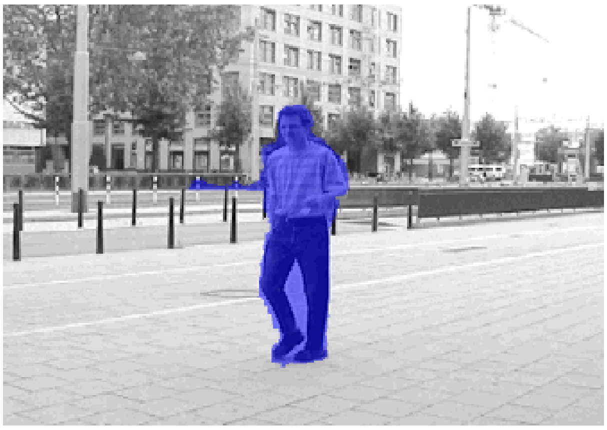
$$K = \min_{c} \sum_{i=1}^{N} d_{c_i, i} + w_{c_{i-1}, c_i}$$

Object Recognition in Videos (IV)

Recognition rates

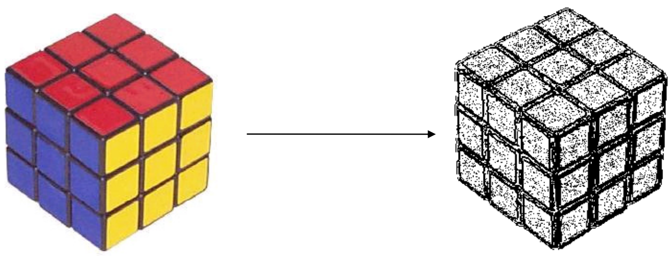
- The recognition rates are between 25-95 % (dependent on the object's complexity and the number of available objects in the database)
- Inflexible objects (e.g. cars) are recognized by far more reliably
- The CSS method is invariant to scaling and rotation; it is very robust to noise
- A comparison of contours is possible in a very efficient way (smooth only once; several comparison of peaks via Euclidian distance)

Object Recognition in Videos (V)



standing
walking
turn around
sit down
sitting

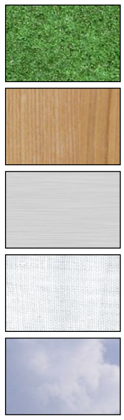
Object Recognition via Textures



Object Recognition via Textures (I)

Texture

- Visual *patterns* which emerge from colors respectively lightness in a picture
- Is caused by reflections of the light at a surface (grass, wood, metal, cloth, but also clouds)
- Contain information about the surface's structure



Object Recognition via Textures (II)

Homogeneous texture descriptor

- Describes direction, unevenness and regularity of a texture
- Suitable for the description of homogeneous areas
- The frequencies contained in the picture are determined and their intersection and standard deviation are calculated
- Invariant to scaling and rotation

Object Recognition via Textures (III)

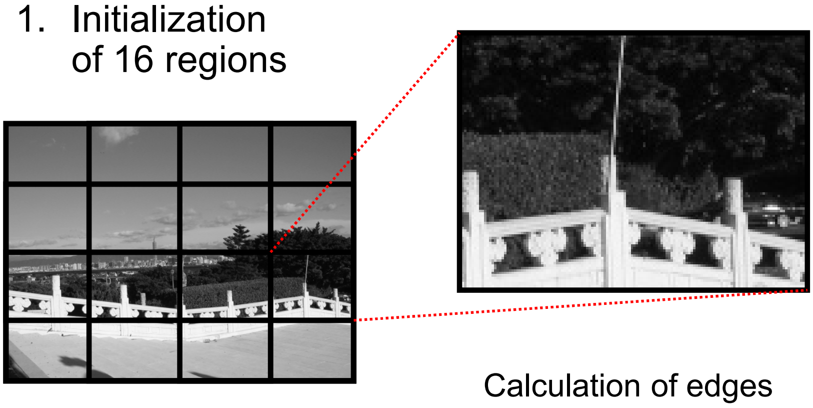
Edge histogram

- Describes local (in an image range) arrangement of edges in image area (textures do not have to be homogeneous)
- Division of the image into 16 equal blocks (invariant to scaling), calculation of edges for each block
- 5 types of edges: vertical, horizontal, 45 degree, 135 degree, undirected
- Storage of the values in a histogram for each type and each block (5 x 16 = 80 elements)
- In a comparison, the rotation can (but does not have to be) considered
- MPEG-7: Only 3 bits are used for the description of a histogram element (in total 240 bits for the texture's description of an image)

Object Recognition via Textures (IV)

Calculation of edge histograms

1. Initialization of 16 regions

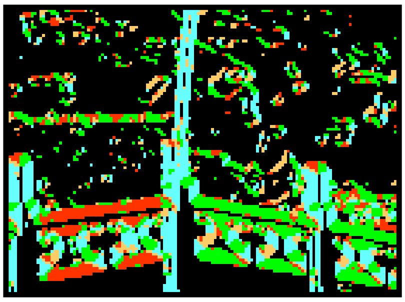


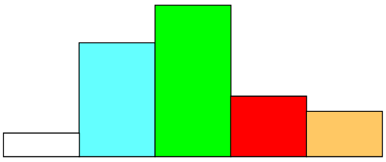
Calculation of edges

Object Recognition via Textures (V)

Calculation of edge histograms

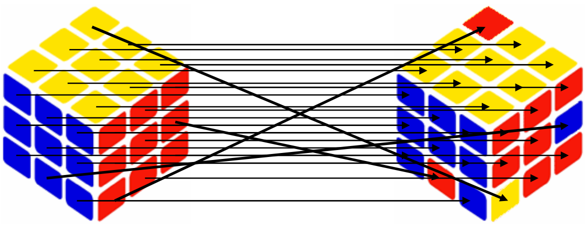
2. Calculation of edge direction and counting edge pixel in histogram





0 degrees 45 degrees 90 degrees 135 degrees

Object Recognition in Videos via Motion (I)



Object Recognition in Videos via Motion (II)

Motion activity

- Description of the motion of a video segment
- General categories:
  - slow: news speaker
  - fast: street scene
  - very fast: soccer, basketball
- Definition of category based on standard deviation of the length of motion vectors
- Optional parameters:
  - Direction of motion
  - Motion activity of regions
  - Duration of the motion

Object Recognition in Videos via Motion (III)

Motion trajectory

- Description of the motion of an image region in video segments
- Similar to motion activity, except several (different) moving regions can be classified
- Examples:
  - Traffic surveillance: Store description of motion for each vehicle and person. A search query may return objects in the local neighborhood.
  - Surveillance of public places: Search query returns people that move in a certain pattern (e.g., first moving slowly, and suddenly start running)

Summary (I)

- Model based object recognition: Extract features of an image and compare it with known object features
- The choice of the object model depends on the object to be detected and the available data
- Features like:
  - **colors** (color space, human color vision, histogram, dominant colors)
  - **contours** (compactness, eccentricity, curvature)
  - **texture** (homogeneous texture descriptor, edge histogram)
  - **motion** (motion activity, motion trajectory)

  can be used to classify objects.

Summary (II)

Name good object models to describe the following objects

Questions ?