Computer Vision (ZDO) - Features and Description

Introduction

Zdeněk Krňoul

Department of Cybernetics
FACULTY OF APPLIED SCIENCES
University of West Bohemia

DEPARTMENT OF CYBERNETICS

Contents:

Object description aims to get:

- numeric - feature vector
- non-numeric - syntactic description

which characterizes both the **shape** and **other properties** of the described object.

- Such features are the input to the **classifier** and the following step - object detection/recognition.
- In most cases, the features are determined only **in 2D** (ie without 3D reconstruction).

Algorithms:

- The description technique is, in some cases, closely connected with the segmentation technique (as a conditioning operation for the description)
- Despite the existence of many practically applicable description methods, **a general methodology has not yet been defined;** current approaches have their pros and cons

Methods:

Type of representation:

- Boundary
- Area

Reconstruction:

- Can reconstruct the shape of the object
- Is not possible

Techniques/Algorithm:

- Mathematical
- Heuristic (for example, trial and error method)

Type of feature:

- Numeric vector
- Syntactic string

OBJECT IDENTIFICATION

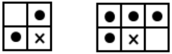
- Identification of the object(s) is a necessary condition for description;
- Provides the ability to refer uniquely to each region of the image

Connected Components Labeling:

- We provide each area with a non-repeating natural number
- Background has number 0, areas are assigned numbers from 1,
- - then the largest area identification number indicates the number of areas in the image;
- - this identification is called **labeling**

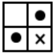
Labeling is a sequential process - first pass:

- We go through the image line by line
- We assign a value to each non-zero pixel according to the value of all its already labeled neighbours.
- if all are zero, we will assign a unassigned label
- if one is non-zero, or is more non-zero, but with the same label, we assign this label (color) to the pixel
- mask for 4-neighborhood and 8-neighborhood:



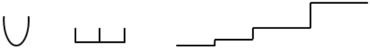
- if there are more non-zero ones (with different labels), we assign one of these labels to the pixel and record them in the so-called label equivalence table (*label collisions*)

0	0	1	0	1
0	0	1	0	1
0	0	1	0	1
0	1	1	1	1



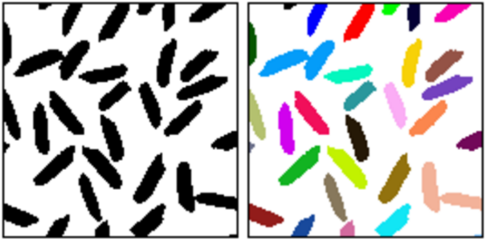
0	0	1	0	2
0	0	1	0	2
0	0	1	0	2
0	3	?	?	?

Note: color collisions occur in practice very often for objects that have a shape:

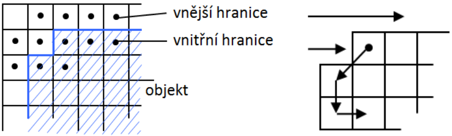


Second pass:

- We go through the whole image line by line again and label the pixels with the collision according to the label equivalence table;
- then each area corresponds to a single label, in another area a non-occurring label
- If we want to find out the number of objects at the same time, labels from the set of natural numbers must be assigned in ascending order so that none is omitted



SHAPE DESCRIPTION BASED ON BOUNDARIES



Algorithm:

- We search the image line by line until we find a pixel belonging to the new area;
- then we go through the points that are part of the border counterclockwise

Simple geometric boundary descriptions:

Length - is a basic property of the border
the simplest case - based on Freeman's code representation, even shifts (horizontal and vertical) are evaluated by length 1, odd (diagonal) by length $\sqrt{2}$

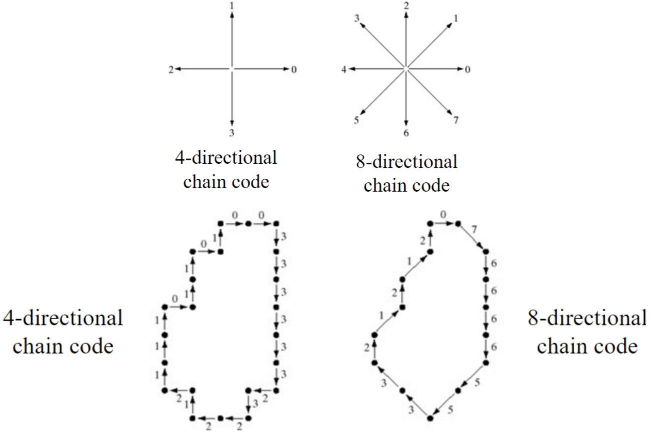
Area Perimeter - length of the closed boundary [1]

Boundary Straightness - The ratio between the total number of cells in the boundary and the number of cells in which the boundary changes direction.

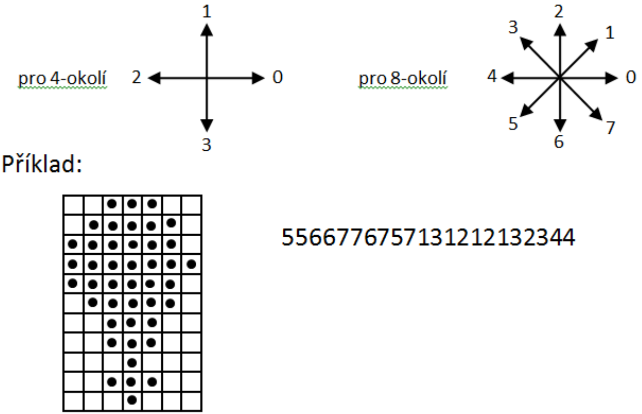
[1] Length will be greater in 4-neighborhood because diagonal shifts are rated 2

Freeman's chain codes

- The boundary is determined by the starting point and the sequence of symbols corresponding to the **unit length** directions.
- The assignment of symbols to individual directions is:



Example:


pro 4-okolí
1
2
0
3
pro 8-okolí
2
3
1
4
0
5
6
7
Příklad:
5566776757131212132344

Independent of starting point - one of the used methods is to determine the starting point of the description so that the string interpreted as a number in the octal system is the smallest number of all possible strings: eg 5566776757131212132344 $\rightarrow$ **1212132344556677675713**

Rotated code by k - multiple of 45° (90°) - added k to each symbol of the chain modulo operation 8 (4)

Rotation independent - derivative (1st difference modulo operation 8 (4)) can be used, which is a sequence of numbers that show changes in the direction of the boundary. e.g. 5566776757131212132344 $\rightarrow$ **0101071622261717271101**

Differential Independent - independent of starting point selection e.g. 01010716222617172711 01 $\rightarrow$ **0101010716222617172711**

Note: Freeman's chain code can also be used to describe a skeleton.
Note 2: This description is suitable for syntactic (structural) recognition methods.

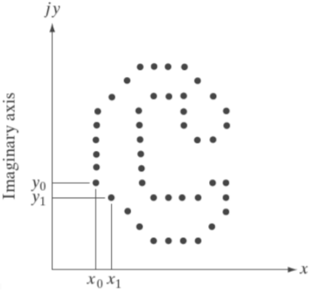
Fourier Descriptors

Frequency Shape Description:

- boundary point expressed as a complex number:

$$s(k) = x(k) + jy(k) \; k = 0, 1, 2 \ldots K - 1 \qquad (1)$$

x, y are the coordinates of the boundary point



DEPARTMENT OF CYBERNETICS

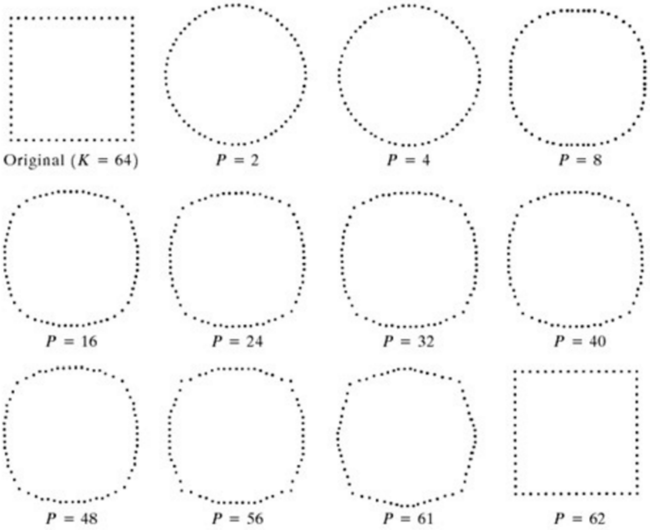
Fourier descriptor is expressed in the frequency domain as:

$$a(u) = \sum_{k=0}^{K-1} s(k) e^{\frac{-j2\pi uk}{K}} \;\; u = 0, 1, 2 \ldots K-1 \tag{2}$$

and the re-projection of the shape of the border into the image plane is then:

$$\hat{s}(k) = \frac{1}{K} \sum_{k=0}^{K-1} a(u) e^{\frac{j2\pi uk}{K}} \;\; k = 0, 1, 2 \ldots K-1 \tag{3}$$

Example:



- Fourier descriptors are not invariant to a transformation, but their relationships to these transformations are known:
- translation:

$$a_T(u) = a(u) + \Delta_{xy}\delta(u) \tag{4}$$

- rotation θ:

$$a_R(u) = a(u)e^{j\theta} \tag{5}$$

- scale α:

$$a_s(u) = \alpha a(u) \tag{6}$$

- starting point k_0:

$$a_p(u) = a(u)e^{\frac{-j2\pi k_0 u}{K}} \tag{7}$$

Sequence of segments:

The sequences of segments of the given properties, the boundary is described by *chain of segments*.

Polynomial segments Line segments and second-order polynomial approximation - **parts of circles, ellipses**, etc. The resulting description is a string of primitives (type of segments)[2];

Algorithm:

- Agglomerative: adjacent boundary points are added to a line segment until the segment loses its linear character. In this case, a new segment is created.
- Divisional: opposite approach - recursive splitting. We start from the endpoints and divide the boundary into smaller sections until all segments have a linear character (expressed by the criterion)

[2]suitable for syntactic recognition

DEPARTMENT OF CYBERNETICS

Pre-defined segments

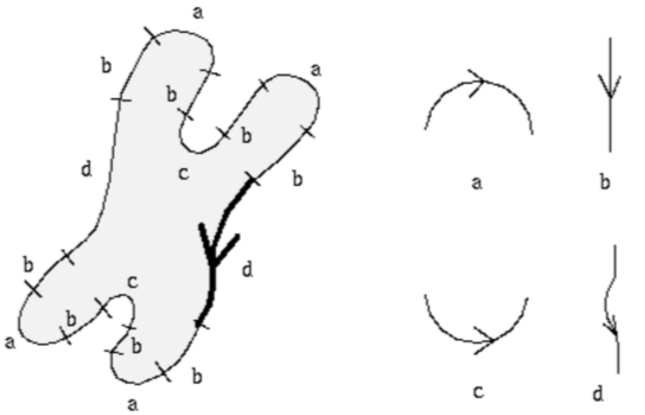


Figure: Example: Pre-defined segments of a chromosome object boundary; sequence of 4 types of segments, the obtained description is $(d, b, a, b, c, b, a, b, d, b, a, b, c, b, a, b)$

DESCRIPTION OF AREA

Simple, heuristic-motivated methods: eg size, squareness, elongation, etc.

- The features are **simple** and give good results for **simple shapes**, but for more complex shapes they fail and you need to choose methods that first divide the region into simpler parts that can be described separately.
- An complex object composed can be described:
  - by *graph*, whose nodes correspond to the parts created by the decomposition of the region.
  - algorithm - *skeleton* or *decomposition* (eg by obtaining convex subregions) $\rightarrow$ creating a graph with nodes connected by some neighborhood relation

Simple scalar descriptions of areas:

Size - Area

- the simplest and most natural feature
- is given by the number of pixels contained in the area
- if you know the pixel size, you can also find out the actual size of the area (the dot size may not be the same for all pixels - eg satellite image)
- calculate the size in the labeled image:

$$S_{area} = \sum_{i,j} g(i,j,p) \tag{8}$$

$$where \;\; g(i,j,p) = \begin{cases} 1 & \text{for } f(i,j) = p \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

p is ID of the region

DEPARTMENT OF CYBERNETICS

Euler's number

- the simplest and most natural feature

$$E = S - N \tag{10}$$

S - number of contiguous parts of the area N - number of holes

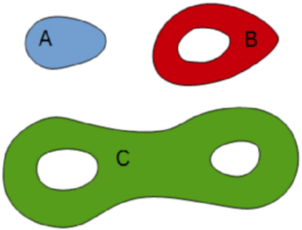


Figure: Example of Euler's numbers: A=1, A=0, A=-1[3]

[3] http://imagej.net

Area Projection

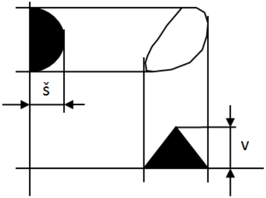
Horizontal projection: projection of a shape into the y-axis image

$$p_H(i) = \sum_j g(i, j, p) \tag{11}$$

Vertical projection: projects shape into x-axis image

$$p_V(j) = \sum_i g(i, j, p) \tag{12}$$

where p is ID of the region



Height:

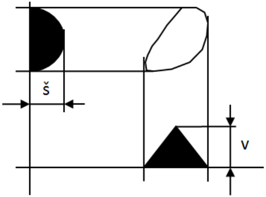
$$height = \max_j p_V(j) \tag{13}$$

Width:

$$width = \max_i p_H(i) \tag{14}$$

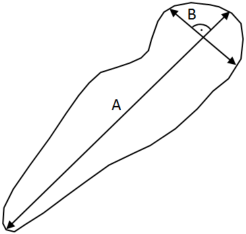
Feret's projections

- first rotates the object by that angle
- then the horizontal projection is calculated

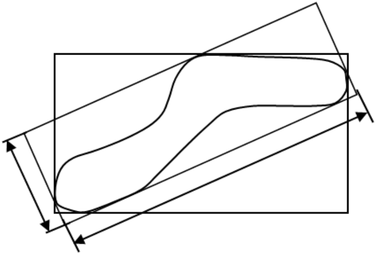


Area shape:

Eccentricity the ratio of the lengths of the longest line A and the longest line B perpendicular to A



Elongation is the ratio between the height and width of a rectangle that is surrounding the region and has the smallest area of all possible rectangles:



Rectangularity is the maximum of all ratios F_k between the size of the region area and the area of the surrounding rectangle in the given direction (rotation) k

We change k discreetly in the range of 0° to 90°

$$\mathit{rectangularity} = \max_k F_k \quad ; \in (0, 1\rangle$$

$\mathit{rectangularity} = 1$ ... describes a perfectly rectangular area

Direction
- only makes sense for elongated regions
- the direction of the longer side of the surrounding rectangle used by rectangularity algorithm

Incompatibility is a property given by the ratio of the perimeter of the area and its content

$$incompatibility = \frac{(o_{area})^2}{S_{area}}$$

Note: the most compact in Euclidean space is the circle

Statistical methods: Moment description

- We interpret the normalized brightness function of an image as the probability density of a two-dimensional random variable.
- Properties can be expressed using statistical moments. Can be used for binary and gray-scale images.

Raw moments:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx\, dy$$

in a discrete case (image)

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

note: it is not invariant to resizing, rotating, shifting, or gray-scale transformations

DEPARTMENT OF CYBERNETICS

Central moment:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_t)^p (y - y_t)^q f(x, y) dx\, dy$$

in a discrete case (image)

$$\mu_{pq} = \sum_{i,j} (i - i_t)^p (j - j_t)^q f(i, j)$$

where $i_t = \frac{m_{10}}{m_{00}}$ a $j_t = \frac{m_{01}}{m_{00}}$
Note: is invariant to shift

Standardized moments:

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

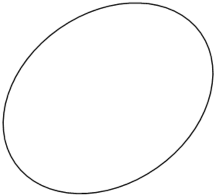
where

$$\gamma = \textit{whole partof}\left(\frac{p+q}{2}\right) + 1$$

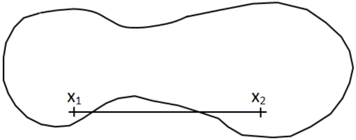
Note: in addition, it is invariant to scaling

Convex descriptions

The region R is *convex* just when for every two points $x_1, x_2 \in R$ it holds that all points of the line x_1, x_2 also belongs to R.

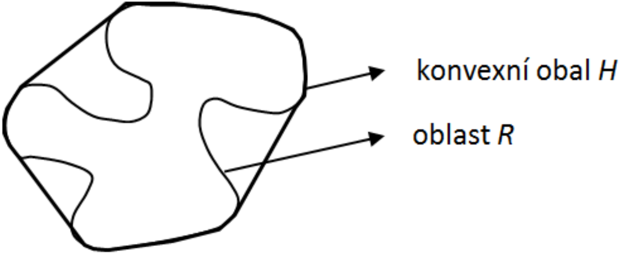


konvexní oblast



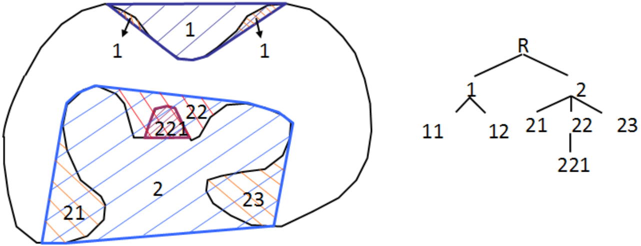
konkávní oblast

Convex Hull ▶ the smallest convex region H such that $R \subset H$



DEPARTMENT OF CYBERNETICS

Tree of concavity areas ▶ we create a convex hull of areas, convex hull of concave parts, hull of concave parts of these parts, etc.



Advantages of graph representation:

- independence of position and rotation, while both properties can be included in the description by the graph
- size independence (unless there is a collision with the image resolution)
- form representation close to man, important elements of the description can be easily determined

Note The complexity of obtaining a shape description also follows from the mentioned properties. If we want to get closer to real computer vision, there is probably no other way.

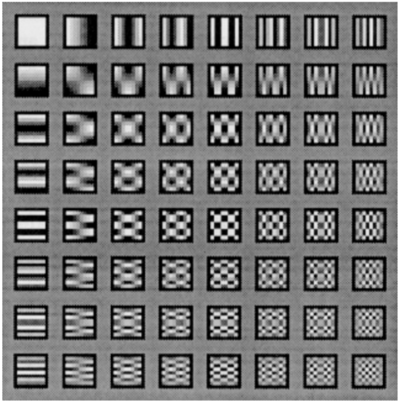
Discrete Cosine Transform (DCT)

- is basically based on the Fourier transform (there is a decomposition into sin and cos functions.)
- suitable description for textures
- basically we get a description of a part (block) of the area of functions expressed as the sum of cos functions oscillating at different frequencies and different amplitudes
- will only use real coefficients

$$F(u,v) = \sqrt{\frac{2}{N}}\sqrt{\frac{2}{M}}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\Lambda(i)\Lambda(j) cos[\frac{\pi u}{2N}(2i+1)] cos[\frac{\pi v}{2M}(2j+1)] f(i,j) \tag{15}$$

- where M, N is size of the block a $f(i,j)$ brightness function

- The principle is used, for example, in JPEG compression, where high frequencies are discarded by compression
- Area is often spread out eg for 8x8 px. blocks ($N = 8$) in which the DCT coefficients *Lambda* are calculated
- The equivalent approach is to calculate the convolution with the calculated 2D cos basis functions



DEPARTMENT OF CYBERNETICS

Gabor filter

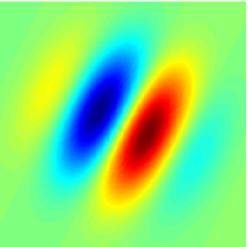
- Gabor filter is used to describe general texture analysis (representation or discrimination) or often for descriptions suitable for OCR, fingerprint, etc.
- 2D Gabor filters are Gaussian kernel functions modulated by a sine plane wave.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) exp(i(2\pi \frac{x'}{\lambda} + \psi))$$

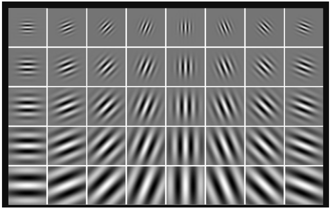
where

$$x' = x \cos\theta + y \sin\theta$$
$$y' = -x \cos\theta + y \sin\theta$$

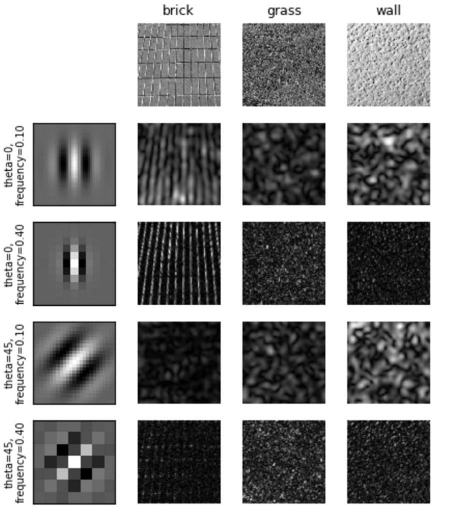
λ represents the wavelength, θ orientation, ψ phase shift, σ modulation - standard deviation and γ aspect ration



DEPARTMENT OF CYBERNETICS



- Images are filtered using the real part of various kernel functions;
- Gabor descriptions always use a set of filters;
- The mean and variance of the filtered images are used directly for classification (in the simplest case according to the smallest quadratic error).
- The frequency and orientation of the filters contained are similar to the procedures used by humans;



demo (http://matlabserver.cs.rug.nl/cgi-bin/matweb.exe)