Institute of Informatics
Eötvös Loránd University
Budapest, Hungary

Basic Algorithms for Digital Image Analysis: a course

Dmitrij Csetverikov

with help of Attila Lerch, Judit Verestóy, Zoltán Megyesi, Zsolt Jankó and Levente Hajder

http://visual.ipan.sztaki.hu

Lecture 13: 2D shape analysis

- Entities and goals of shape analysis
- Criteria for selection of shape analysis techniques
  - Area-based and contour-based methods
- Data structures for 2D shapes
  - Chain code
  - Contour slope sequence (CSS)
  - Radial function
- Shape moments
- Fourier analysis of CSS
- Circularity, or shape factor
- Shape dimensions

Entities and goals of 2D shape analysis

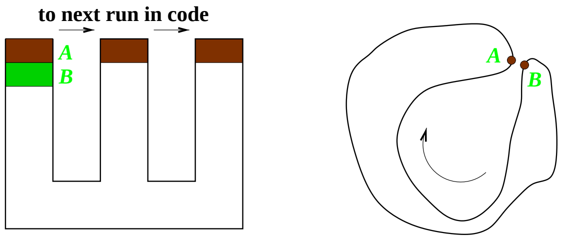
- **Entities of shape analysis**:
  - flat objects
  - projections of 3D objects
  - segmented binary images
  - planar shapes
  - curves, closed contours
- **Goals of shape analysis**:
  - shape description
  - shape decomposition
  - matching
  - recognition
  - determination of position and orientation

Criteria for selection of shape analysis methods

1. Scalar transform versus space domain methods
   - **Scalar transform**: Output is set of scalar features (feature vector).
     - for statistical pattern recognition algorithms
   - **Space domain**: Output is another picture. (Example: MAT.)
     - for structural pattern recognition algorithms
2. Information preserving versus non-preserving methods
   - **Information preserving**: Loss of information is controllable.
   - **Information non-preserving**: Loss of information is **not** controllable.
3. **Area-based versus contour-based methods**
   - Suitable for different kinds of analysis
   - Need different representations
     - area-based: points ordered in 2D
     - contour-based: points ordered along contours

Different representations support different operations

- Area-based representations support computation of **integral features**
  - does not support local contour analysis
- Contour-based representations support **local contour analysis** and computation of some integral features
  - does not support proximity analysis



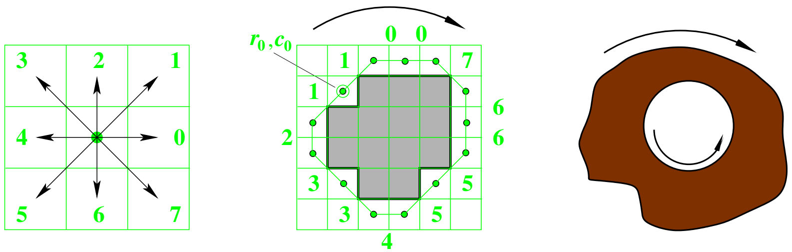
Left: Points A and B are adjacent on contour, but separated in RLC.
Right: Points A and B are close in space, but far away along contour.

Data structures for 2D shapes

- Area-based data structures (already discussed):
  - Binary **image matrix**
    - ∗ supports image processing and computation of features
    - ∗ no data compression
    - ∗ rotating binary object is possible, but care should be taken
  - **Run-length code**
    - ∗ supports computation of area-based features
    - ∗ significant data compression in most cases
    - ∗ RLC of rotated object is re-computed from scratch
- Contour-based data structures (discussed below):
  - **Chain code**
  - **Contour slope sequence**
  - **Radial function**

Chain code

Chain code is obtained by **contour following** and consists of 2 parts: (1) coordinates of the starting point (r_0, c_0); (2) a sequence of codes $\{c_1, c_2, \ldots\}$ pointing at the next contour pixel.



Chain coding. Left: 8 chain codes pointing to 8 neighbours of a pixel.
Center: Example of chain coding of 4-connected object.
Right: Outer and inner contours are traced in opposite directions.

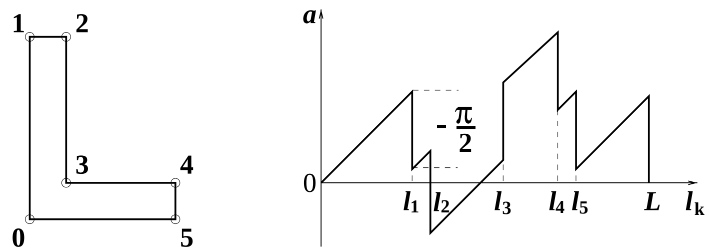
Chain code is **regenerative**: shape is restored from its chain code. If (r_0, c_0) is given, original position is restored; otherwise, shape is restored up to shift.

Contour slope sequence

Contour slope sequence (CSS), or saw-tooth function

$$a(l_k) = \alpha_k + \frac{2\pi}{L} \cdot l_k \tag{1}$$

Here α_k is slope (tangent angle) of contour in current position k relative to slope in starting position $k = 0$. L is total contour length, $l_k \leq L$ is current arc length.

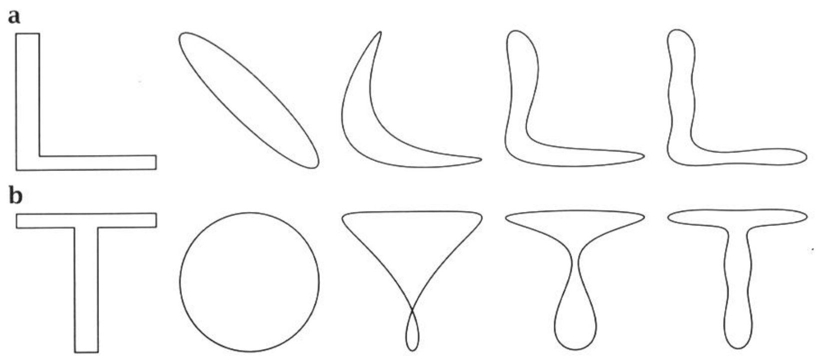


A simple shape and its CSS.

CSS is **regenerative** up to shift, rotation and **reflection**.

Fourier analysis of contour slope sequence

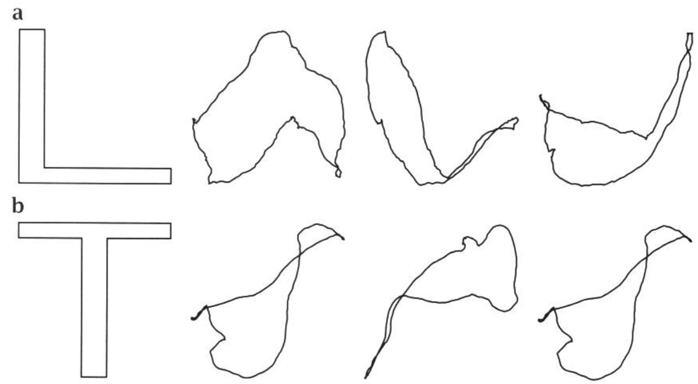
Fourier coefficients of the periodic CSS are used to describe shape.



Reconstruction of shape of letter 'L' (a) and letter 'T' (b) with 2, 3, 4 and 8 pairs of Fourier coefficients.

Importance of phase in DFT

Although only the **magnitude** of DFT is usually used, the **phase** also carries important structural information.

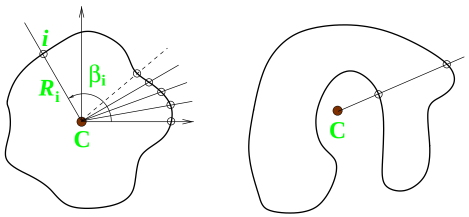


Importance of phase for shape description with Fourier transform. The letters are shown restored with unchanged magnitudes but random modifications of phase.

Radial function

Radial function $R(\beta)$ is a contour-based polar shape representation. It shows the distance R between an interior point (usually, centroid C) and the contour points, as a function of the polar angle β.

- $R(\beta)$ is often derived from the chain code.
- Usually, it is only applied to **star-shaped objects**, when $R(\beta)$ is single-valued.
- $R(\beta)$ is **regenerative** up to shift, rotation and reflection.



Radial function $R(\beta)$. C is centroid of object. Left: Parameters of radial function. Right: Multi-valued radial function of non-star-shaped object.

Obtaining the contour-based representations

- Chain code is obtained from binary image or from labelled run-length code by **contour following**.
  - The algorithms are relatively complex. **Not considered** in this course.
- CSS and radial function are usually obtained from chain code.
- Both need accurate **resampling** of arc length.
  - Reason: chain code measures arc length in 1's and $\sqrt{2}$'s
  - Warning: angular resolution in $R(\beta)$ must be fine enough not to loose narrow spikes on contour.
- CSS also needs an accurate estimate of **slope**. (Can be done with splines.)
- Radial function needs **centroid** (centre of mass) (x_c, y_c) of shape Q with area S:

$$x_c = \frac{1}{S} \sum_{x,y \in Q} x, \quad y_c = \frac{1}{S} \sum_{x,y \in Q} y \tag{2}$$

Summary of contour-based representations

All contour-based data representations usually result in substantial **data compression** compared to image matrix.

- **Chain code**
  - supports computation of contour features and some area-based features
  - is used to obtain other contour-based representations
  - must be computed from scratch when object is rotated
- **Contour slope sequence**
  - supports more precise estimation of contour features
  - undergoes circular shift when object is rotated
- **Radial function** $R(\beta)$
  - supports computation of contour features and some area-based features
  - undergoes circular shift when object rotates around centroid

Shape moments

The **central moment** of order pq for object (region) Q is defined as

$$\mu_{pq} = \frac{1}{S} \sum_{x,y \in Q} (x - x_c)^p \cdot (y - y_c)^q \,, \qquad (3)$$

where $S = \mu_{00}$ is area of Q (number of pixels in Q), (x_c, y_c) centroid of Q calculated by (2), and $p, q = 0, 1, \ldots$.

μ_{pq} are called **central** because they are defined relative to centre of mass.

Preservation of information:

- In theory, moments are information preserving: shape can be restored from **all** μ_{pq}.
- In practice, lower-order moments $p + q \leq 4$ are only used as **information non-preserving** features.
  - Reason: Higher order moments are noise-sensitive since x^p with large p amplifies noise in x.

Second order moments: $p + q = 2$

Most frequently used in practice. Two features can be defined that are **shift, rotation- and scale invariant** in continuous case:

$$M_{cmp} = \frac{1}{2\pi} \cdot \frac{S}{\mu_{20} + \mu_{02}}$$

$$M_{ect} = \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{\mu_{20} + \mu_{02}}$$

Here μ_{20}, μ_{02} and μ_{11} are the central moments defined in (3).

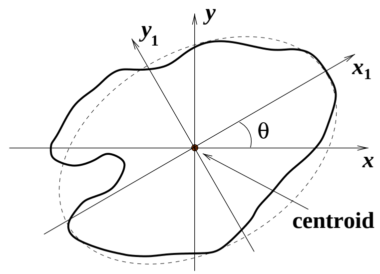
- $0 \leq M_{cmp} \leq 1$ is normalised feature of **compactness**
  - Shows radial distribution of points: for disc, $M_{cmp} = 1$
  - Robust: insensitive to noise and to rotation of **discrete** shape
- $0 \leq M_{ect} \leq 1$ is normalised feature of **eccentricity**
  - Shows elongation: for disc, $M_{ect} = 0$; for line, $M_{ect} = 1$
  - Less robust than M_{cmp}

Second order moments and object orientation

μ_{20}, μ_{02} and μ_{11} form components of **inertia tensor** for rotation of object around axes passing through its centre of mass.

Orientation of object can be defined as angle between x axis and **principal axis**: axis around which the object can be rotated with **minimum inertia**

$$\theta = \frac{1}{2} \arctan \frac{\mu_{02} - \mu_{20}}{2\mu_{11}} + (\operatorname{sign} \mu_{11}) \cdot \frac{\pi}{4} + \pi n, \quad n = 0, 1 \qquad (4)$$



Orientation θ obtained by (4) is

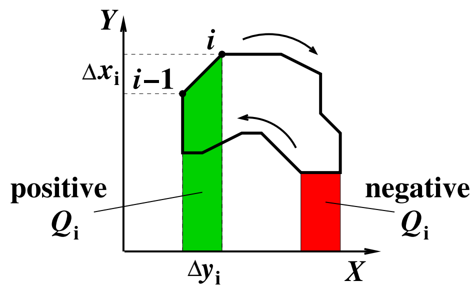
- axial data defined modulo π
- accurate for **elongated** shapes, but inaccurate for **compact**, close-to-circular shapes
  - reason: numerical instability when both $\mu_{02} - \mu_{20}$ and μ_{11} are small
- undefined for shapes with more than 2 axes of symmetry

Shape description by second order moments models the object by an **ellipse**.

- Principal (major) axis is longer axis of ellipse, giving minimum inertia I_{min}.
- Shorter (minor) axis gives maxumum inertia I_{max}.
- Eccentricity (or elongation) M_{ect} is normalised difference $I_{max} - I_{min}$
  - The greater the difference the longer the ellipse
  - The longer the ellipse the more accurate the orientation estimate (4)

Computing moments of polygon and chain coded shape

$$\triangle M_{lm}^{(i)} \doteq \int_{Q_i} x^l y^m dxdy = \int_{x_{i-1}}^{x_{i-1}+\triangle x_i} x^l dx \int_{0}^{y_{i-1}+\frac{\triangle y_i}{\triangle x_i}(x-x_{i-1})} y^m dy$$



Computing shape moments of polygon traced clockwise.

For chain code, $\triangle x_i = \triangle y_i = 1$. Introduce

$$a_{ix} = (0, 1, 1, 1, 0, -1, -1, -1) \qquad x_i = x_{i-1} + a_{ix}$$
$$a_{iy} = (1, 1, 0, -1, -1, -1, 0, 1) \qquad y_i = y_{i-1} + a_{iy}$$

Then one can show that area is

$$S = M_{00} = \sum_{i=1}^{n} a_{ix} \left\{ y_{i-1} + \frac{1}{2} a_{iy} \right\},$$

first order sums are

$$M_{10} = -\frac{1}{2} \sum_{i=1}^{n} a_{iy} \left\{ x_{i-1}^2 + a_{ix} \left(x_{i-1} + \frac{a_{ix}}{3} \right) \right\}$$

$$M_{01} = \frac{1}{2} \sum_{i=1}^{n} a_{ix} \left\{ y_{i-1}^2 + a_{iy} \left(y_{i-1} + \frac{a_{iy}}{3} \right) \right\},$$

and second order sums are

$$M_{20} = -\frac{1}{12} \sum_{i=1}^{n} a_{iy} \left(x_i^2 + x_{i-1}^2 \right) \left(x_i + x_{i-1} \right)$$

$$M_{02} = \frac{1}{12} \sum_{i=1}^{n} a_{ix} \left(y_i^2 + y_{i-1}^2 \right) \left(y_i + y_{i-1} \right)$$

$$M_{11} = \frac{1}{24} \sum_{i=1}^{n} a_{ix} \left\{ a_{ix} \left[(y_i + y_{i-1})^2 + 2y_i^2 \right] + 4x_{i-1} \left[(y_i + y_{i-1})^2 - y_{i-1} y_i \right] \right\}$$

Computing centroid

$$x_c = \frac{M_{10}}{S}, \quad y_c = \frac{M_{01}}{S},$$

and substituting x_c, y_c and M_{20},M_{02},M_{11} into equation (3), one obtains central moments μ_{20},μ_{02},μ_{11}.

Summary of shape moments

- When $p + q = 3$, five invariant features can be defined which reflect **asymmetry** of shape.
  - These features are less frequently used than M_{cmp} and M_{ect}
- **Greyscale, colour** and **affine-invariant** moments also exist.
- Moment features are used
  - for recognition of small set (10–20 pcs) of distinct objects
  - for pre-classification before more precise comparison (contour matching), in order to reduce number of candidates
  - in combination with other features
- Typical applications: robot vision, character recognition.

Advantages of shape moments:

- Simple operations involved.
- Can be computed from image matrix, run-length code and chain code
  - Formulae for image matrix are very simple
  - Formulae for RLC are simple
  - Formulae for chain code are more complicated
- M_{cmp} is robust to noise, rotation and distortions.
- Relatively good discriminating power.

Drawbacks of shape moments:

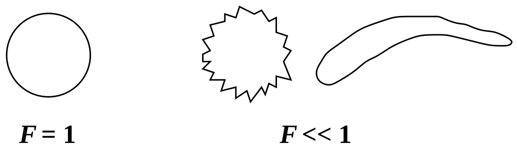
- Do not reflect local contour features.
- M_{ect} and higher order moment features are less robust.
- Orientation θ may be inaccurate or even undefined.

Circularity, or shape factor

The feature

$$F = \frac{4\pi S}{P^2}$$

is a measure of **circularity**, or compactness of shape. L is the perimeter, S the area. Sometimes called shape factor, F also reflects the **smoothness of contour**. $0 \leq F \leq 1$; for circle, $F = 1$.



Examples of objects with different shape factors.

F is **rotation- and scale-invariant**, but depends on resolution. It is easy to compute and relatively robust. Sensitive to elongated local contour features. In case of noise, contour smoothing or approximation is used.

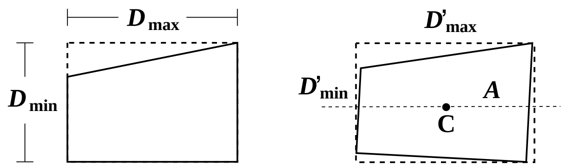
Shape dimensions

Ratio of shape dimensions:

$$T = \frac{D_{min}}{D_{max}}$$

where D_{min}, D_{max} are dimensions of **minimum enclosing rectangle**. T is rotation- and scale invariant measure of **shape elongation**.

It is easier to compute the smallest rectangle aligned with the **principal axis** of inertia. Then T is defined in a similar way for dimensions of this rectangle.



Shape dimensions. Left: Minimum enclosing rectangle. Right: Smallest rectangle aligned with major axis of inertia. C is the centroid, A the major axis.

Both solutions may be computationally unstable:

- The minimum enclosing rectangle is sensitive to local contour features.
- The other solution is sensitive to orientation of the major axis which may be inaccurate.

Usage: Compute the diameters of simple shapes when the dimensions themselves are of interest.

- Sorting by size (for example, fruits)
- Part gripping by robot