5 Shape-based Features

5.1 Shape Representation

- Shape based image similarity allows for different interpretations:
  - Images with similar shaped objects
  - Images with similar dominant shapes
- Both are reasonable ideas and a “meaningful” definition is highly dependent on the particular application

5.1.1 Contour-based Comparison

- By comparing the contours we can determine which images contain similarly shaped objects
- The outline is usually viewed as closed contour
- This is more or less provided through segmentation
- The semantics of the objects here is better described than e.g., global edge images
5.1 Contour-based Comparison

- Shape matching requires **complex similarity measures**
- Requirements for the comparative measure:
  - Invariant regarding **shifts** (translation invariance)
  - Invariant regarding **scaling**
  - Invariant regarding **rotations** (rotational invariance)

5.1 Low Level Features

- Simple indicators of forms, which are characterized by their contour:
  - Number of vertices
  - Area
  - Enclosed area (holes are not included)
  - Eccentricity
  - ...

5.1 Chain Codes

- **Chain codes** (also known as Freeman codes)
  - Are very simple pixel-based descriptions of a form (Freeman, 1961)
- The contour is traversed either clockwise/inverse
- Changes of the edges direction are logged
- Each pixel receives a code depending on its predecessor

- Visual impression of the two images is different, but the emerging forms are identical

- These numbers only give an **absolute sense** of the shape
  - Scale invariance is not provided
  - The shape is not reconstructable
  - The similarity of shapes due to such numbers (e.g., shape area) is doubtful
  - In shape description, low level features are **only helpful** in combination with other features

- **Direction codes**

```
3 2 1
4 X 0
5 6 7
```

- Translation invariance is clear in this way
- E.g.:

```
... x 1 0 1 2 3 ...
```

(Chain Code of the image)
5.1 Chain Codes

• For scale invariance:
  – Remove equal consecutive numbers (works poorly with complex shapes)

- Rectangles have the same code as squares

  000066666444444222222222222222222 → 0642
  000000666664444442222222222222222 → 0642

• Reduced Chain code

  3 2 1
  0 X 0
  1 2 3

  • Opposite directions receive the same encoding

5.1 Chain Codes

• Reduced Difference Chain Code (RDC) (Freeman, 1961)
  – Each two consecutive points are summarized by their difference
  – Advantage: compression
  – Example: rotational invariance

- Works only with rotations by multiples of 45°

- Sequence of numbers in the code is not unique
5.1 Chain Codes

- Alternative coding describes this behavior with edges (Shape numbers) (Bribiesca / Guzman, 1978)

- convex corner edge concave corner
  - Code 1
  - Code 2
  - Code 3

- Matching of two chain codes by comparing the two generated strings
  \[ A = (a_1, a_2, \ldots, a_m) \] and \[ B = (b_1, b_2, \ldots, b_n) \]

- Often is edit distance used for comparison:
  - Levenstein-distance
  - Advanced Levenstein-distance
  - ...

5.1 Chain Codes

- Shape numbers
  - Generate all cyclic permutations of the chain code
  - Sort the list of these permutations lexicographically
  - Select as encoding of the shape first permutation of this list

- Weighted Levenstein distance
  - Idea: string A can be converted through a sequence of
    - Substitutions of single characters \((a \rightarrow b)\),
    - Insertions \((\epsilon \rightarrow a)\) and
    - Deletions \((a \rightarrow \epsilon)\) into string B
  - Each of these operations have associated costs (natural numbers)
  - Find a sequence of operations, which converts A to B, with minimal cost
  - These costs are the distance between A and B

5.1 Chain Codes

- Advantages:
  - Relatively easy to calculate

- Disadvantages:
  - Scaling and rotation invariance are not always given
  - Much information is reduced or lost
5.2 Area-based Retrieval

- Representation
  - Area based description doesn’t only use the contour, but also the interior of a shape
  - Representations are divided into
    - Information-preserving representations (Image transformations, etc.)
    - Non-information-preserving representations (Low-Level Features, descriptive moments, ...)

5.2 Representation

- Transformation
  - Hough, Walsh, Wavelet transforms
- Structural representation
  - Primitive shapes which cover an area (rectangles, circles, ...)
- Geometric representation
  - Shape area, number of holes, compactness, symmetry, moments, moment invariants, ...

5.2 Low Level Features

- Shape area
  - Number of set pixels
- Roundness
  - Perimeter²/surface area (minimum) for circles
- Euler number
  - Difference:
    - Number of connected components
    - Number of holes in the components

5.2 Structural Representation

- How well can shapes be covered with a minimal number of primitive shapes?

5.2 Structural Representation

- Primitive shapes are e.g., Superquadratics (Barr, 1981)
  - Distortion of circles (spheres), e.g., ellipsoids, hyperboloids, etc.
  - Distortions are twists, bends, ...
- We aim at obtaining a minimal coverage
- What does minimal mean?
  - The encoding of each shape requires a certain length (depending on complexity)
  - If only primitive shapes are used, then, representation is susceptible to flaws
  - If more shapes are used...
    - Then the total length of the coding is higher
    - But the error is smaller
  - Therefore: Minimize a weighted sum consisting of length and coding errors
Shapes can also be described by their **skeleton** (Blum, 1973)

- **Central axis**: the number of centers of all circles with maximum area, inscribed in the shape

**Symmetric boundary points**

- Set of centers of all inscribed, bitangent circles (bitangent = 2 points of contact)
- Slightly more accurate than the central axis, but very sensitive to small changes in the shape

**The shock set** approach

- Also results in a skeleton
- Wave fronts start from the edges with the same speed. The skeleton is provided by the points where the wave fronts meet (like wildfire)

**The graph of the skeleton** is stored and used for comparison

- Skeletons are indeed calculated from boundary points, but also take into consideration shapes, e.g., holes

**Example**: (Sebastian and Kimia, 2005)

- The matching of different skeletons is usually done by using the editing distance with different editing costs
- Four basic editing operations:
  - **Splice** removes a skeleton branch
  - **Contract** represents $n$ branches at a node with $n-1$ branches
  - **Merge** removes a node between exactly two skeleton branches
  - **Deform** deforms a branch
5.2 Skeleton

- Example: skeletons have the same topology after some splice operations

5.3 Moments

- A special type of shape features based on the image moments
- The intensity function $I(x, y)$ of the gray values of an image (after appropriate normalization) can be in addition interpreted as a probability distribution on the pixels of the image
  - If we take a random pixel of the image, considering this distribution, there is a high probability that the pixel is dark and a low probability that it is bright
- The statistical properties of $I$ can be used as shape features

5.3 A little Stochastic

- Let $f$ be a discrete probability distribution on a finite set $A$ of real numbers
- Then:
  - $f(x) \geq 0$ for all $x \in A$
  - $\sum_{x \in A} f(x) = 1$
- If $X$ is a random variable with distribution $f$, then $f(x)$ is the probability that $X$ takes the value $x$

5.3 Uniqueness Theorem

- Each distribution function can be uniquely described by its moments
- **Uniqueness Theorem:**
  - $f$ can uniquely be reconstructed from the sequence of moments $m_0, m_1, m_2, \ldots$
  - The only condition: all elements must exist, that is, be finite

5.3 A little Stochastic

- The $i$-th moment of $X$ is
  $$m_i := \sum_{x \in A} x^i \cdot f(x)$$
- Already known from the stochastic:
  The first moment of $X$ is the expected value

- The $i$-th central moment of $X$ is
  $$\mu_i := \sum_{x \in A} (x - \bar{x})^i \cdot f(x)$$
  where $\bar{x}$ denotes the expected value of $X$
- The second central moment of $X$ is the variance
- The first central moment is always 0
- Important property: central moments are invariant to shifts
Now let f be a two-dimensional discrete distribution function, e.g.:
- \( f: A \times B \rightarrow [0, 1] \)
- \( f(x, y) \geq 0 \) for all \( (x, y) \in A \times B \)
- \( \sum_{(x,y) \in A \times B} f(x, y) = 1 \)

Where \( (X, Y) \) is a random vector with distribution \( f \)

The \((i, j)\)-th moment of \( (X, Y) \)
- \( m_{i,j} := \sum_{(x,y) \in A \times B} x^i \cdot y^j \cdot f(x, y) \)

The \((i, j)\)-th central moment of \( (X, Y) \) is
- \( \mu_{i,j} := \sum_{(x,y) \in A \times B} (x - m_{0,1})^i \cdot (y - m_{1,0})^j \cdot f(x, y) \)

Known: \( \mu_{1,1} \) is the covariance of \( X \) and \( Y \)

The uniqueness theorem applies also here, as before

Example: an image of width \( b \) and height \( h \) with pixel intensities \( I(x, y) \):

By normalizing \( I \), we obtain a two-dimensional discrete probability distribution \( f \):

\[
f(x,y) := \frac{I(x,y)}{\sum_{(u,v) \in A \times B} I(u,v)} \quad A := \{0, 1, \ldots, b - 1\} \quad B := \{0, 1, \ldots, h - 1\}
\]

Considering the uniqueness theorem, the moments of \( f \) (the image moments) represent a complete description of the image
- Therefore: use the (first \( k \)) image moments as shape features
- By using the central moments we have features that are invariant towards scaling and rotation!

From the central moments, we can calculate the normalized central moments:
- \( \eta_{i,j} := \frac{\mu_{i,j}}{\left( \sum_{(x,y) \in A \times B} I(x,y) \right)^{(i+j)/2}} \)

It can be shown that:
- The normalized central moments \( \eta_{i,j} \) are invariant towards scaling

Example (Scaling Invariant)
5.3 Linear Transformation

- We still lack the **rotational invariance**
- Rotations (and scaling) in the $\mathbb{R}^2$ can be described through **linear transformations**
  - These are functions $t: \mathbb{R}^2 \to \mathbb{R}$, described by a $(2 \times 2)$ matrix $A$, thus $t(x, y) = A \cdot (x, y)$
- Rotation with angle $\alpha$ (followed by scaling with factor $s$):

\[
A = s \cdot \begin{pmatrix}
\cos(\alpha) & -\sin(\alpha) \\
\sin(\alpha) & \cos(\alpha)
\end{pmatrix}
\]

5.3 Moment Invariants

- We are looking for functions $g$, which transform the normalized central moments to **new characteristic values**, so that ...
- Rotations of the original shape do not change these metrics
- These metrics describe the form, regardless of their location and size
- Such functions are called **moment invariants** (Hu, 1962)

\[
g(\eta_0, \eta_1, \eta_2, \ldots) = g'(\eta'_0, \eta'_1, \eta'_2, \ldots)
\]

5.3 Algebraic Invariants

- Important property
  - If $g_1$ and $g_2$ (independent of one another) are relative invariants with weights $w_1$ and $w_2$, then
  \[
h(x) := \frac{g_1(x)^{w_2}}{g_2(x)^{w_1}}
\]
  is an absolute invariant
- Proof:
  \[
h(A \cdot x) = \frac{(g_1(A \cdot x))^{w_2}}{(g_2(A \cdot x))^{w_1}} = \frac{(det(A))^{w_1 \cdot w_2}}{(det(A))^{w_1 \cdot w_2}} \cdot h(x)
\]

5.3 Algebraic Invariants

- There are known methods in the **linear algebra** that can be used to find relative algebraic invariants for our special case
- A set of seven (absolute) moment invariants for moments of degree 2 and 3 is presented in (Hu, 1962)
  - $g_1(\cdot) = \eta_0 + \eta_2(\cdot)$
  - $g_2(\cdot) = (\eta_1 - \eta_0)_0^2 + 4 \eta_1^2$
  - $g_3(\cdot) = (\eta_1 - \eta_0)_0^2 + (\eta_2 - \eta_1)_0^2$
  - $g_4(\cdot) = (\eta_1 - \eta_0)_0^2 + (3 \eta_2 - 2 \eta_1)_0^2$
5.3 Algebraic Invariants

\[ g_0 = (\eta_{0,0} - 3 \eta_{1,2}) (\eta_{0,0} + \eta_{1,2}) \]
\[ \left[ (\eta_{2,0} + \eta_{1,2})^2 - 3 (\eta_{2,1} + \eta_{0,0})^2 \right] \]
\[ + (3 \eta_{2,1} - \eta_{0,0}) (\eta_{1,2} + \eta_{0,0}) \]
\[ 3 (\eta_{2,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,0})^2 \]
\[ g_1 = (\eta_{1,0} - \eta_{0,0}) (\eta_{0,0} + \eta_{1,2}) \]
\[ (\eta_{2,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,0}) \]
\[ + 4 \eta_{1,0} (\eta_{3,0} + \eta_{1,2}) (\eta_{2,1} + \eta_{0,0}) \]
\[ 3 (\eta_{2,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,0})^2 \]
\[ g_2 = (3 \eta_{1,1} - \eta_{0,0}) (\eta_{0,0} + \eta_{1,0}) \]
\[ (\eta_{2,0} + \eta_{1,2})^2 - 3 (\eta_{2,1} - \eta_{0,0})^2 \]
\[ + (3 \eta_{1,2} - \eta_{0,0}) (\eta_{1,2} + \eta_{0,0}) \]
\[ 3 (\eta_{2,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,0})^2 \]

5.3 Example

- If we found suitable moment invariants, we can characterize shapes by the vector of related characteristic values
- The comparison of shapes is then performed by measuring the distance of real vectors
- How many moment invariants do we need?

5.3 Moment Invariants

- Separability:
  - Two different shapes in the database must differentiate in at least one element of the feature vector
  - This requirement determines how many different moment invariants are necessary

5.3 Separability Property

- The quality of the representation of shapes by moment invariants can be increased, by using other types of moments
- Examples:
  - Zernike moments
  - Tschebyschev moments
  - Fourier moments

5.3 Moment Invariants

- The calculation of feature vectors can be simplified if the contour of the shapes have a special form
- Examples:
  - Splines (based) on polynomial functions
  - Polygons
  - Curves in parametric representation

5.3 Moment Invariants
5.3 Moment Invariants

- **Example** (Hu, 1962):
  - The shapes of the characters in the alphabet are represented each with a two-dimensional vector

- Experiments: Retrieval System (STAR Mehtre and others, 1995)
  - Test collection: company logos
  - Moment invariants show an average retrieval efficiency of 85-88%
  - Combined feature vectors:
    - In combination with other features even 89-94% is obtained
  - “Retrieval Effectiveness” is here a mix of precision and recall

5.4 Whole Image Description

- No description of individual shapes, but of the overall impression created by the shapes in the picture
- Images are considered perceptually similar if shapes occur in similar correlations
- Simple queries:
  - Query by visual example
  - Query by sketch

5.4 Query by Visual Example

- Procedure (Hirata and Kato, 1992)
  - Pre-process the images in the database
    - Segment the images from the database and extract the edges (resulting in a binary image)
    - For each image from the database, save a normalized representation of the dominant shapes (Pictorial Index)
  - Users provide a rough drawing (binary)
  - Compare the drawing with the Pictorial Index

- Query by visual example
  - GazoPa shape similarity
    - Doesn’t work that great
5.4 Query by Visual Example

- **Image abstraction** for the pictorial index
  - Reduce the image size to, e.g., 64 x 64 pixels
  - Gradient calculation in four directions using the brightness values of each pixel
  - Calculate the edges:
    - All points with gradient greater than the average gradient plus standard deviation

- **Matching** can not simply compare at pixel level
  - White spots in the sketch may mean that **nothing should be there**, or **it's not important**, what is at the point
  - Sketches could be simplified, deformed and/or moved
  - Therefore, calculate the **local correlation** between the edge image and the sketch

- **Calculating the local correlation**:  
  - Divide the edge image and the sketch in 8 x 8 blocks, and compare any two blocks at the same coordinates  
  - Move the sketch-block over the edge image (original image), in the x and y directions (-4 to +4 pixels) and sum over the number of each matching pixel values  
  - The maximum of these sums is the **local correlation**  
  - The aim of this step is to compensate **local inaccuracies** in the drawing and the pictorial index

- **Calculation of the global correlation**  
  - The global correlation is simply the sum of all local correlations  
  - After calculating the global correlation for each image in the database, sort the database by correlation size
5.4 Query by Visual Example

- **Advantages**
  - Good retrieval results with respect to the overall visual impression
  - Imprecision in the sketch is adjusted in matching

- **Disadvantages**
  - The calculation of similarities is very expensive and can not be calculated in advance

---

**Matlab example**

- **Query image**

---

**Extract edges**

---

**Result**

---

This Lecture

- **Shape-based Features**
  - Chain Codes
  - Area-based Retrieval
  - Moment Invariants
  - Query by Example

---

Next lecture

- **Introduction to Audio Retrieval**
  - Basics of audio
  - Audio information in databases
  - Basics of audio retrieval