ifis
Institut für Informationssysteme
Technische Universität Braunschweig

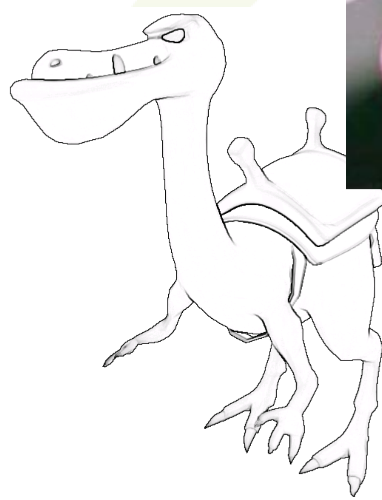
Multimedia Databases

Wolf-Tilo Balke
Silviu Homoceanu
Institut für Informationssysteme
Technische Universität Braunschweig
http://www.ifis.cs.tu-bs.de

Previous Lecture

Summary

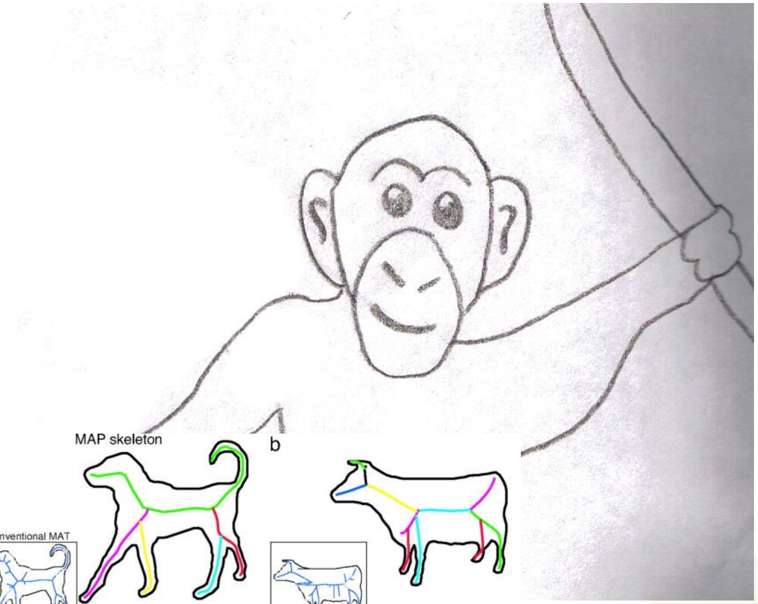
- Multiresolution Analysis
- Shape-based Features
  - Thresholding
  - Edge detection
  - Morphological Operators

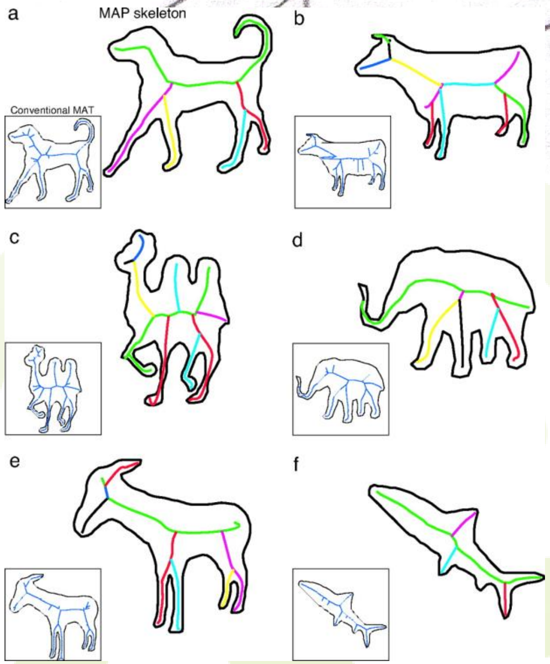


5 Shape-based Features

5 Shape-based Features

- 5.1 Chain Codes
- 5.2 Area-based Retrieval
- 5.3 Moment Invariants
- 5.4 Query by Example





5.1 Shape Representation

- Segmentation provides multiple different possibilities for the **representation** of individual objects or of the entire image
  - **Individual objects**
    - Description of the contours by characteristics of a closed curve
    - Description of the area that is enclosed by the curve
    - Hybrid representation (curve and surface)
  - **Entire image**
    - Description of the dominant edges in the image (e.g., edge histograms)

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

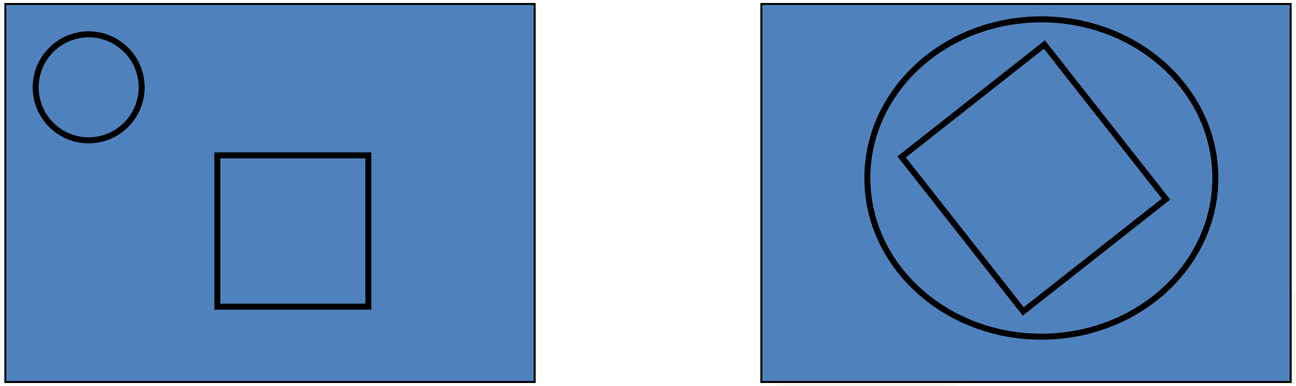
5.1 Contour-based Comparison

- 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 Contour-based Comparison



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

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 Low Level Features

- 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

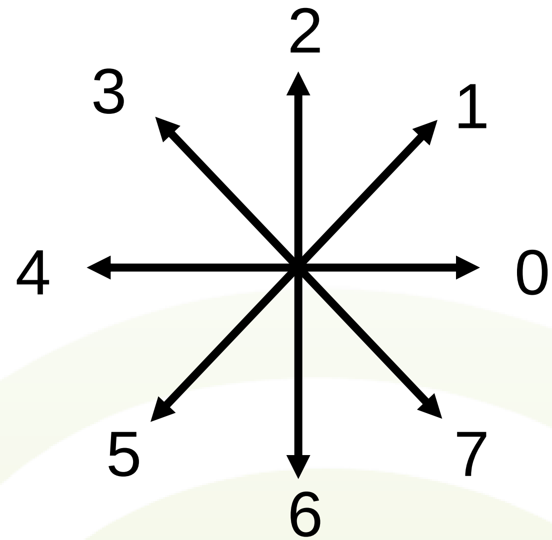
5.1 Chain Codes

- **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

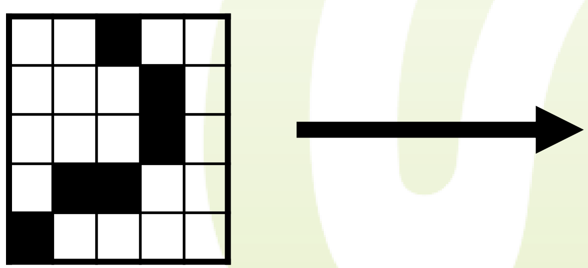
5.1 Chain Codes

- **Direction codes**

3	2	1
4	X	0
5	6	7



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

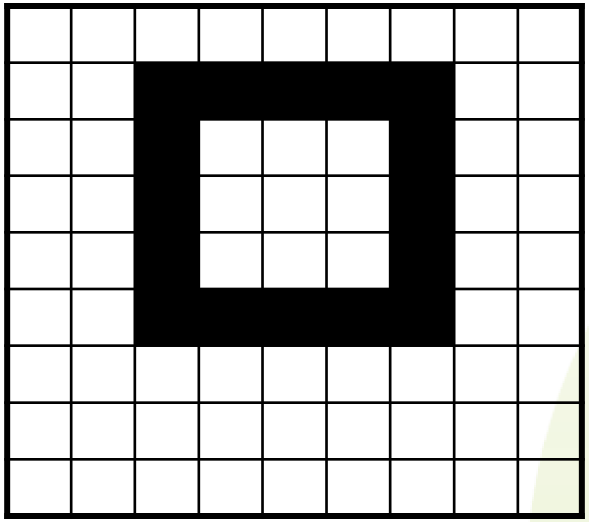


… x 1 0 1 2 3 …

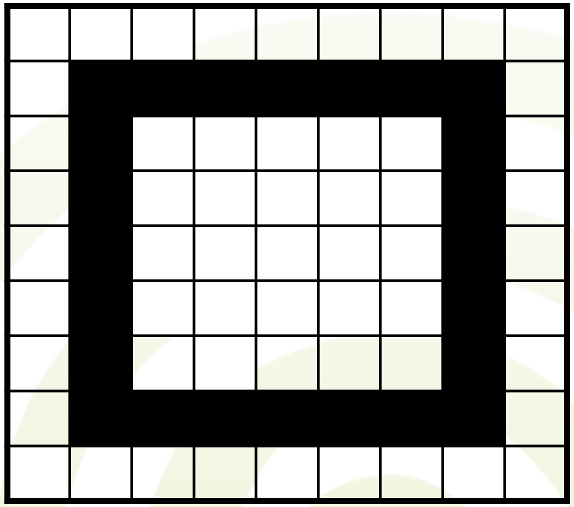
(Chain Code of the image)

5.1 Chain Codes

- **For scale invariance:**
  - Remove equal consecutive numbers (works poorly with complex shapes)



$00000666664444422222 \rightarrow 0642$



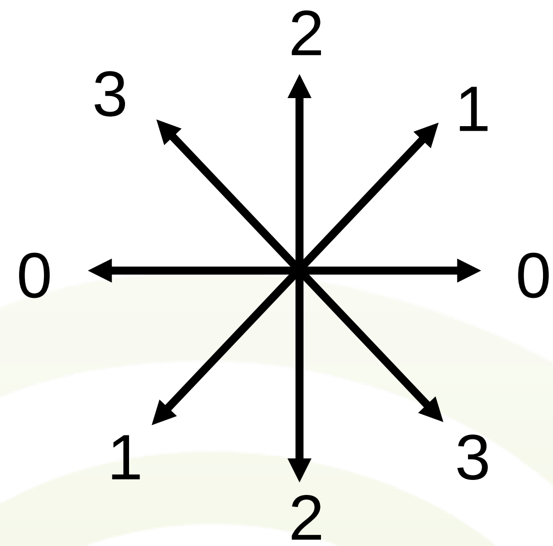
$0000000666666644444442222222 \rightarrow 0642$

  - Rectangles have the same code as squares

5.1 Chain Codes

- **Reduced Chain code**

3	2	1
0	X	0
1	2	3



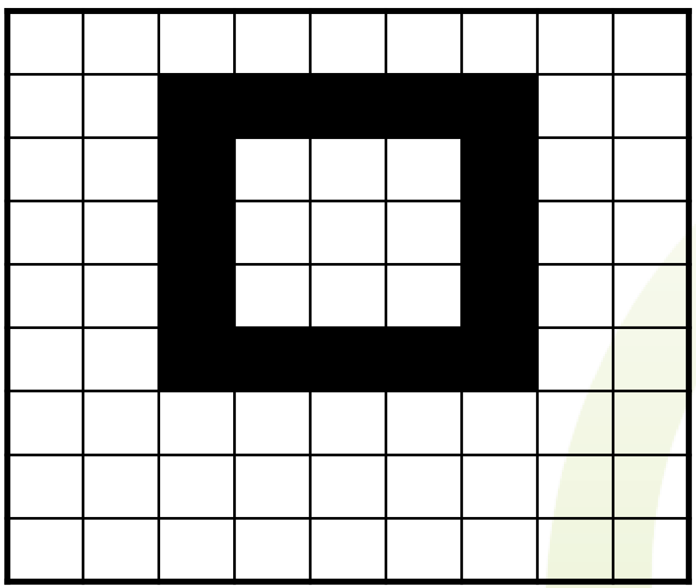
- Opposite directions receive the same encoding

5.1 Chain Codes

- **Reduced Difference Chain Code** (RDC) (Freeman, 1961)
  - Each two consecutive points are summarized by their difference
  - Advantage: compression
  - $(0\ 0\ 0\ 0\ 0\ 2\ 2\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 2\ 2\ 2\ 2\ 2)$
    $\rightarrow (0\ 0\ 0\ 0\ {-2}\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ {-2}\ 0\ 0\ 0\ 0\ 2)$
    $\rightarrow (0\ {-2}\ 0\ 2\ 0\ {-2}\ 0\ 2)$

5.1 Chain Codes

- Reduced Difference Chain Code bring a conditional **rotational invariance**



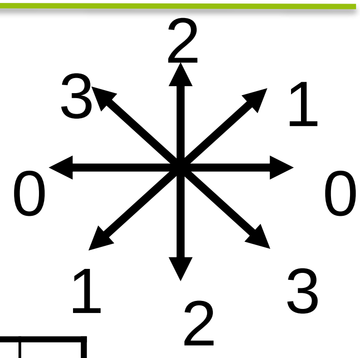
Chain Code:

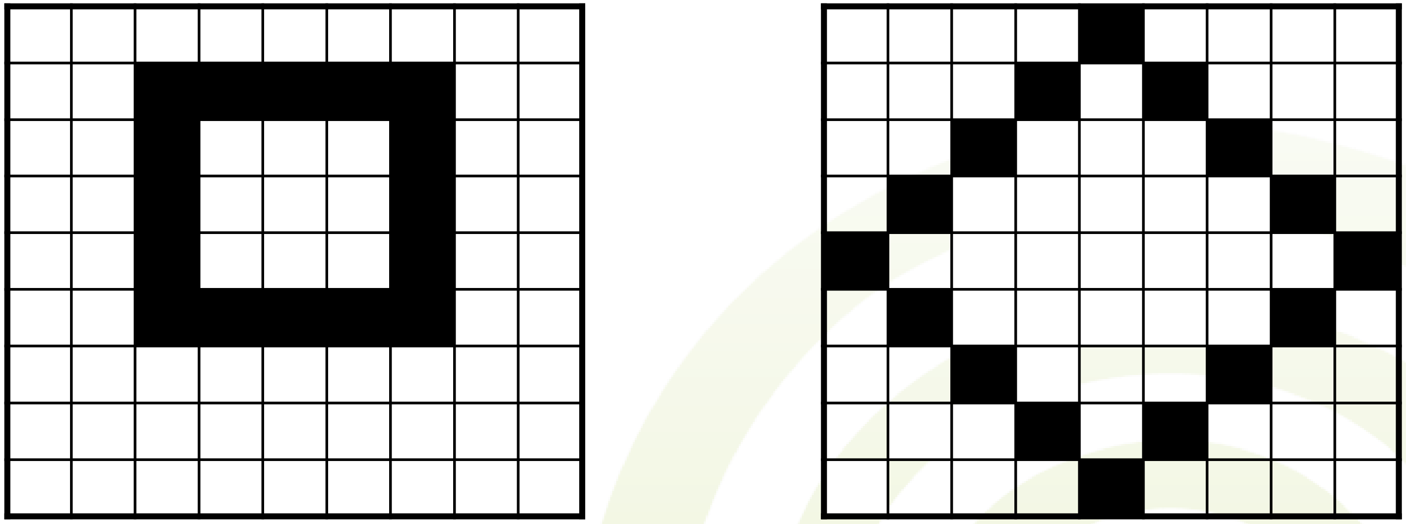
00000666664444422222

Reduced Chain Code:

00000222220000022222

- Example: rotational invariance

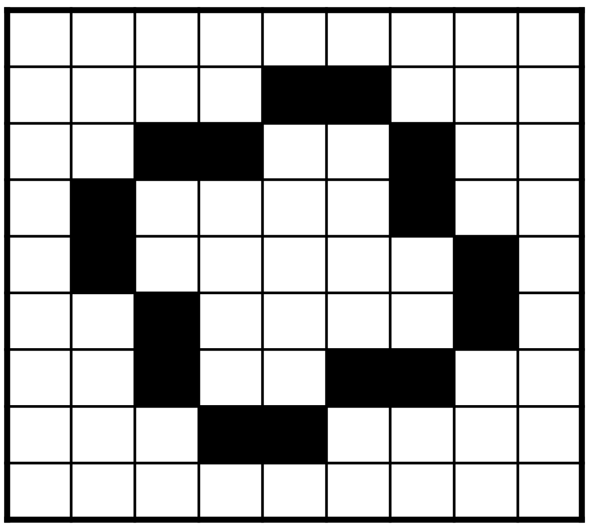




000002222200000022222	111113333311111133333
→ 0 -2 0 2 0 -2 0 2	→ 0 -2 0 2 0 -2 0 2

5.1 Chain Codes

- Works only with rotations by multiples of 45°



$0\ 1\ 0\ 1\ 2\ 3\ 2\ 3\ 0\ 1\ 0\ 1\ 2\ 3\ 2\ 3\ 0$

$\rightarrow -1\ \ 1\ \ -1\ \ -1\ \ -1\ \ 1\ \ -1\ \ 3$

$-1\ \ 1\ \ -1\ \ -1\ \ -1\ \ 1\ \ -1\ \ 3$

$\rightarrow -1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 3$

$-1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 3$

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

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

5.1 Chain Codes

- Matching of two chain codes by comparing the two generated strings

  $A = (a_1,\dots, a_m)$ and $B = (b_1,\dots, \mathrm{b}_n)$

- Often is **edit distance** used for comparison:
  - Levensthein-distance
  - Advanced Levensthein-distance
  - …

5.1 Chain Codes

- **Weighted Levenstein distance**
  - Idea: string A can be converted through a sequence of
    - Substitutions of single characters (a → b),
    - Insertions (ε → a) and
    - Deletions (a → ε)

    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

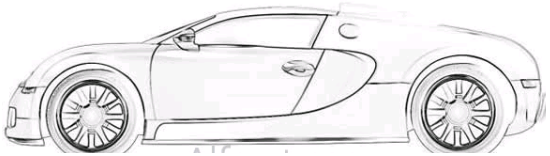
- **Advanced Levenstein-distance**
  - Generalization of Levenstein-distance
  - Additional, operations:
    - aab $\longrightarrow$ abb
    - abb $\longrightarrow$ aab
    - a $\longrightarrow$ aa
    - aa $\longrightarrow$ a
  - This will also be assigned cost values
  - Distance is again the minimum total value of all the transformations from A to 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

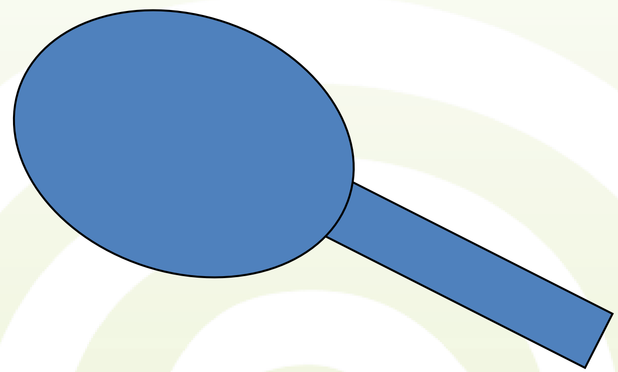
- **Transformation**
  - Hough, Walsh, Wavelet transforms
- **Structural representation**
  - Primitive shapes which cover an area (rectangles, circles, …)
- **Geometric representation**
  - Shape area, number of holes, compactness, symmetry, moments, moment invariants, ...

5.2 Low Level Features

- **Shape area**
  - Number of set pixels
- **Roundness**
  - Perimeter^2/surface area (minimum) for circles
- **Euler number**
  - Difference:
    - Number of connected components
    - Number of holes in the components

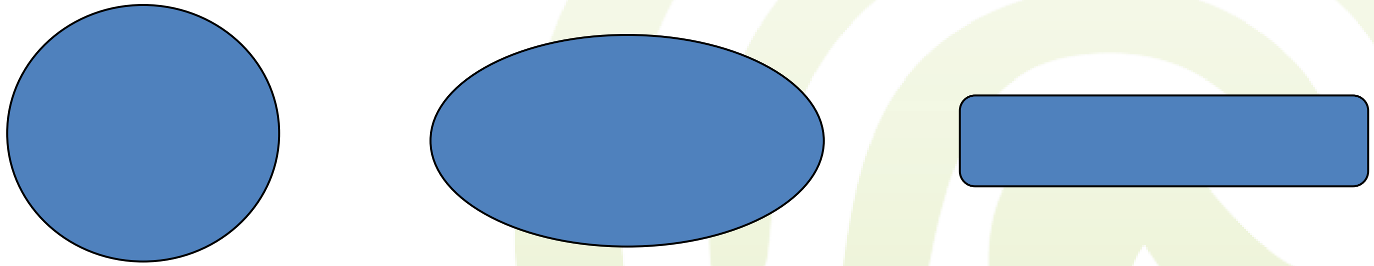
5.2 Structural Representation

- 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, ...

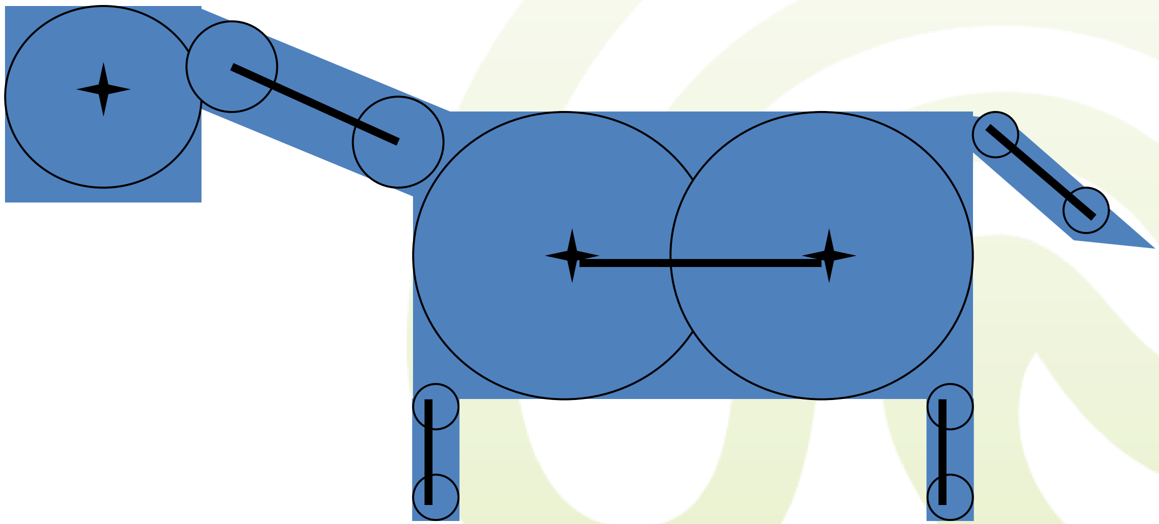


5.2 Structural Representation

- 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

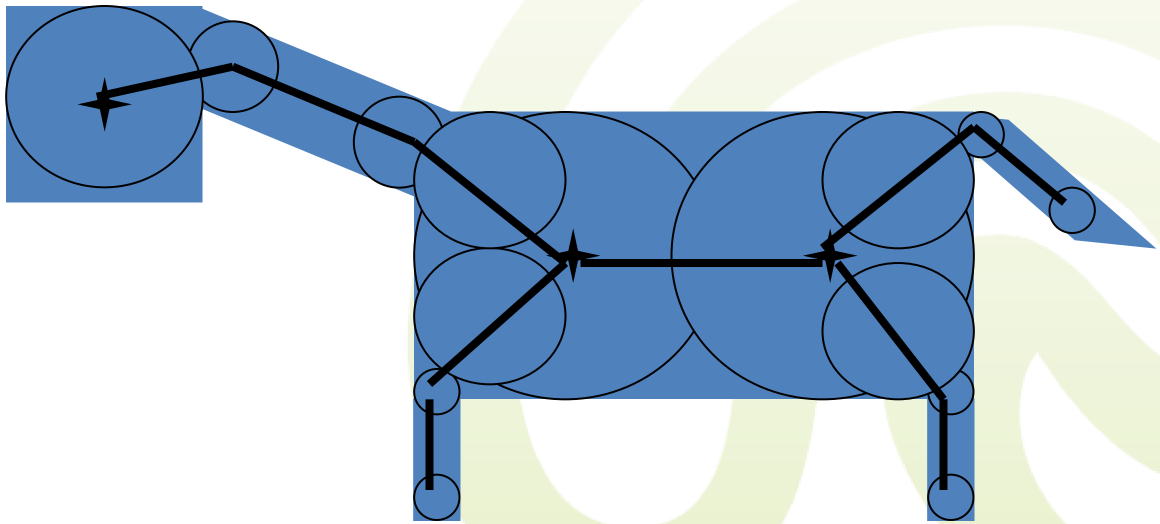
5.2 Skeleton

- 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



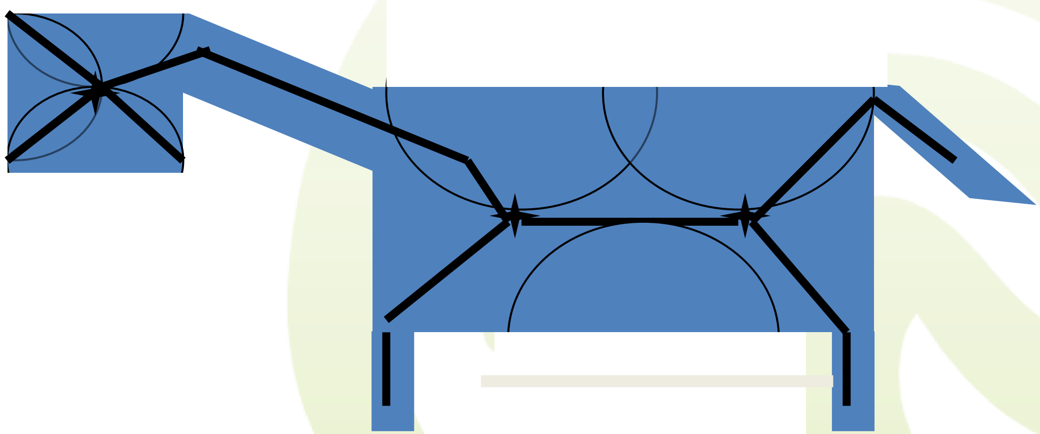
5.2 Skeleton

- **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



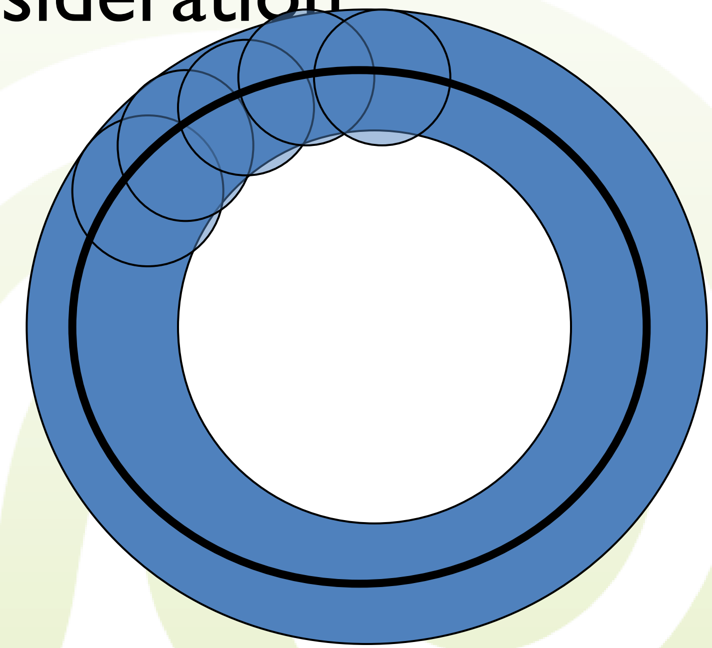
5.2 Skeleton

- 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 were the wave fronts meet (like wildfire)



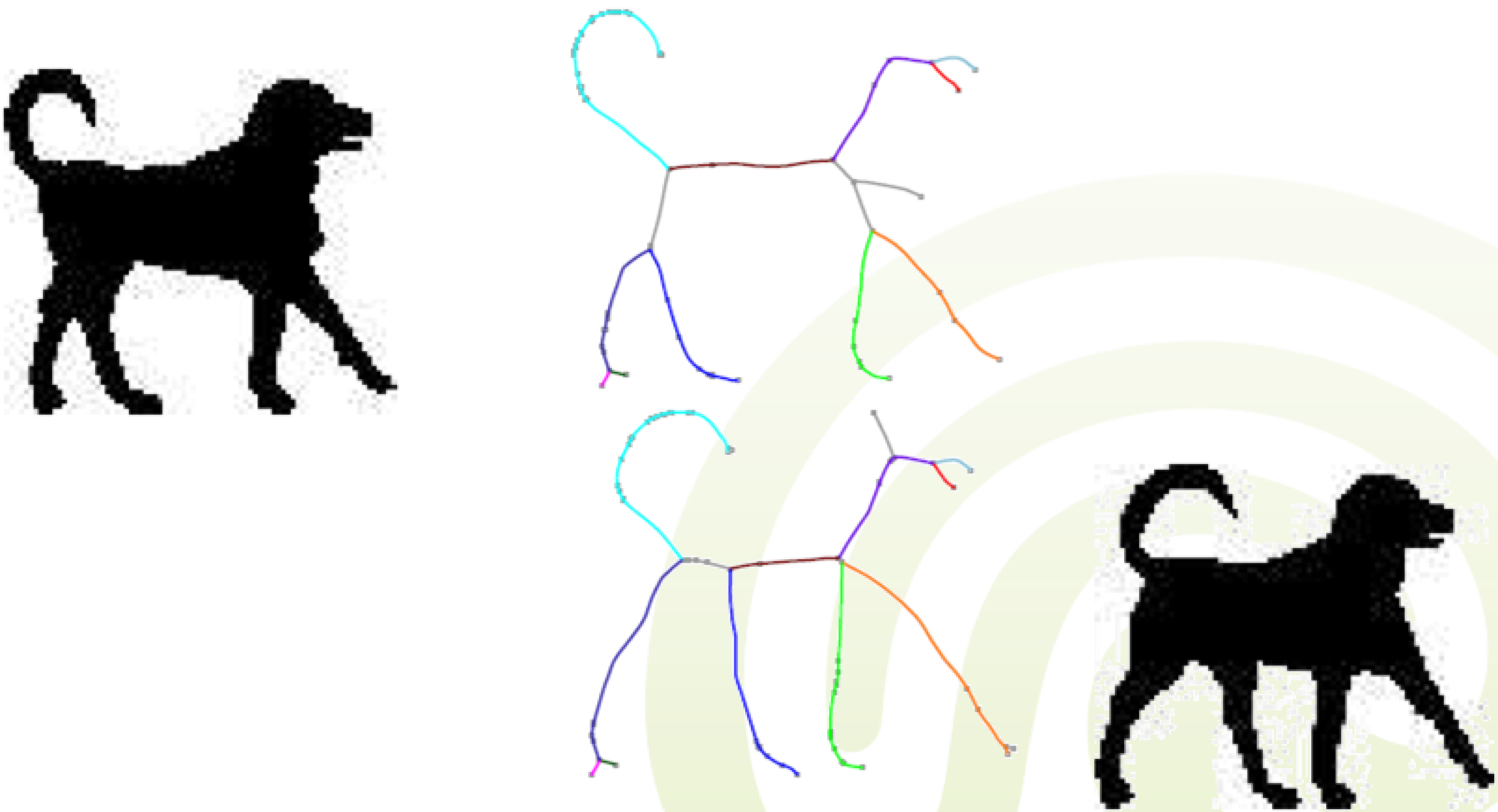
5.2 Skeleton

- 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



5.2 Skeleton

- Example: (Sebastian and Kimia, 2005)

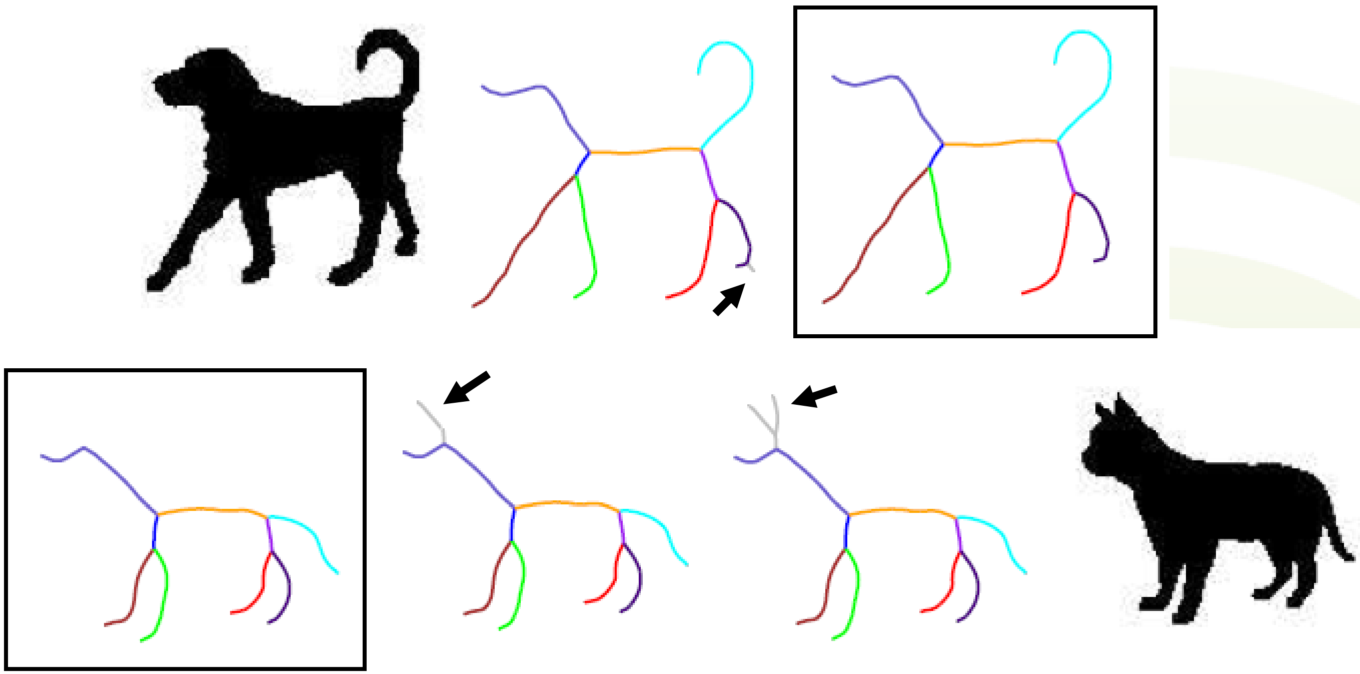


5.2 Skeleton

- 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 is bright
- The **statistical properties** of I can be used as shape features

5.3 A little Stochastics

- 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 A little Stochastics

- 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**

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 Stochastics

- The **i-th central moment** of X is

$$\mu_i := \sum_{x \in A} (x - \bar{x})^i \cdot f(x)$$

  where $\overline{x}$ 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**

5.3 2-D Moments

- 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

5.3 2-D Moments

- 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

$$\mu_{i,j} := \sum_{(x,y)\in A\times B} (x - m_{1,0})^i \cdot (y - m_{0,1})^j \cdot f(x,y)$$

- Known: $\mu_{1,1}$, is the **covariance** of X and Y
- The **uniqueness theorem** applies also here, as before

5.3 Image as Distribution

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

$b = 10$

$h = 6$

$I(0, 0) = 0, \quad I(2, 0) = 1, \quad I(1, 1) = 1, \quad \ldots$

- By normalizing I, we obtain a two-dimensional **discrete probability distribution f:**

$$f(x, y) := \frac{I(x, y)}{\sum_{(u,v) \in A \times B} I(u, v)}$$

$$A := \{0, 1, \ldots, b-1\}$$
$$B := \{0, 1, \ldots, h-1\}$$

5.3 Image Moments as Features

- 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 shifts!**
- But: how do we obtain invariance against scaling and rotation?

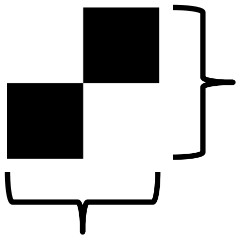
5.3 Scaling Invariant Moments

- 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**

5.3 Example (Scaling Invariant)

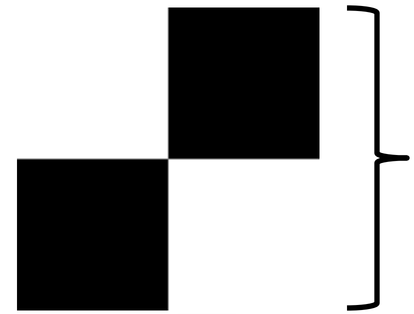


292

292

- $m_{0,1} = 146$
- $m_{1,0} = 146$
- $m_{0,2} = 28349$
- $m_{1,1} = 16060$
- $m_{2,0} = 28349$

- $\mu_{0,1} = 0$
- $\mu_{1,0} = 0$
- $\mu_{0,2} = 7032$
- $\mu_{1,1} = -5256$
- $\mu_{2,0} = 7032$



584

- $m_{0,1} = 291{,}5$
- $m_{1,0} = 291{,}5$
- $m_{0,2} = 113100$
- $m_{1,1} = 63947$
- $m_{2,0} = 113100$

- $\mu_{0,1} = 0$
- $\mu_{1,0} = 0$
- $\mu_{0,2} = 28131$
- $\mu_{1,1} = -21025$
- $\mu_{2,0} = 28131$

- $\eta_{0,1} = 0$
- $\eta_{1,0} = 0$
- $\eta_{0,2} = 0{,}165$
- $\eta_{1,1} = -0{,}1233$
- $\eta_{2,0} = 0{,}165$

5.3 Linear Transformation

- We still lack the **rotational invariance**
- Rotations (and scaling) in the $\Re^2$ can be described through **linear transformations**
  - These are functions t: $\Re^2 \rightarrow \Re$, described by a (2×2) matrix A, thus $t(x, y) = A \cdot (x, y)$
- Rotation with angle α (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

		Linear transformation	
Shape:	$I(x, y)$	$\longmapsto$	$I'(x, y)$
	$\downarrow$		$\downarrow$
Distribution:	$f(x, y)$		$f'(x, y)$
	$\downarrow$		$\downarrow$
Moments:	$\eta_{0,0}, \eta_{0,1}, \eta_{1,0}, \ldots$		$\eta'_{0,0}, \eta'_{0,1}, \eta'_{1,0}, \ldots$

- Goal: **invariant functions g** with

$$g(\eta_{0,0}, \eta_{0,1}, \eta_{1,0}, \ldots) = g(\eta'_{0,0}, \eta'_{0,1}, \eta'_{1,0}, \ldots)$$

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)

5.3 Algebraic Invariants

- How do we find moment invariants?
  - Algebra: **algebraic invariants**
  - A function $g: \Re^n \to \Re$ is called **relative invariant** with weight $w \in \Re$, if for all $(n \times n)$ matrices with full rank and all $x \in \Re^n$ we have:
    - $g(A \cdot x) = [\det(A)]^w \cdot g(x)$
  - Thus invariant under linear transformation expressed by A
- For $w = 0$, g is called **absolute invariant**

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

$$h(x) := \frac{\left(g_1(x)\right)^{w_2}}{\left(g_2(x)\right)^{w_1}}$$

    is an **absolute invariant**

- Proof:

$$h(A \cdot x) = \frac{\left(g_1(A \cdot x)\right)^{w_2}}{\left(g_2(A \cdot x)\right)^{w_1}} = \frac{\left(\det(A)\right)^{w_1 \cdot w_2} \cdot \left(g_1(x)\right)^{w_2}}{\left(\det(A)\right)^{w_1 \cdot w_2} \cdot \left(g_2(x)\right)^{w_1}} = h(x)$$

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)
  - $g_1 (\dots) = \eta_{2,0} + \eta_{0,2}(\dots)$
  - $g_2 (\dots) = (\eta_{2,0} - \eta_{0,2})^2 + 4\,\eta_{1,1}{}^2$
  - $g_3 (\dots) = (\eta_{3,0} - \eta_{0,2})^2 + (\eta_{2,1} - \eta_{0,3})^2$
  - $g_4 (\dots) = (\eta_{3,0} - \eta_{1,2})^2 + (3\,\eta_{2,1} + \eta_{0,3})^2$

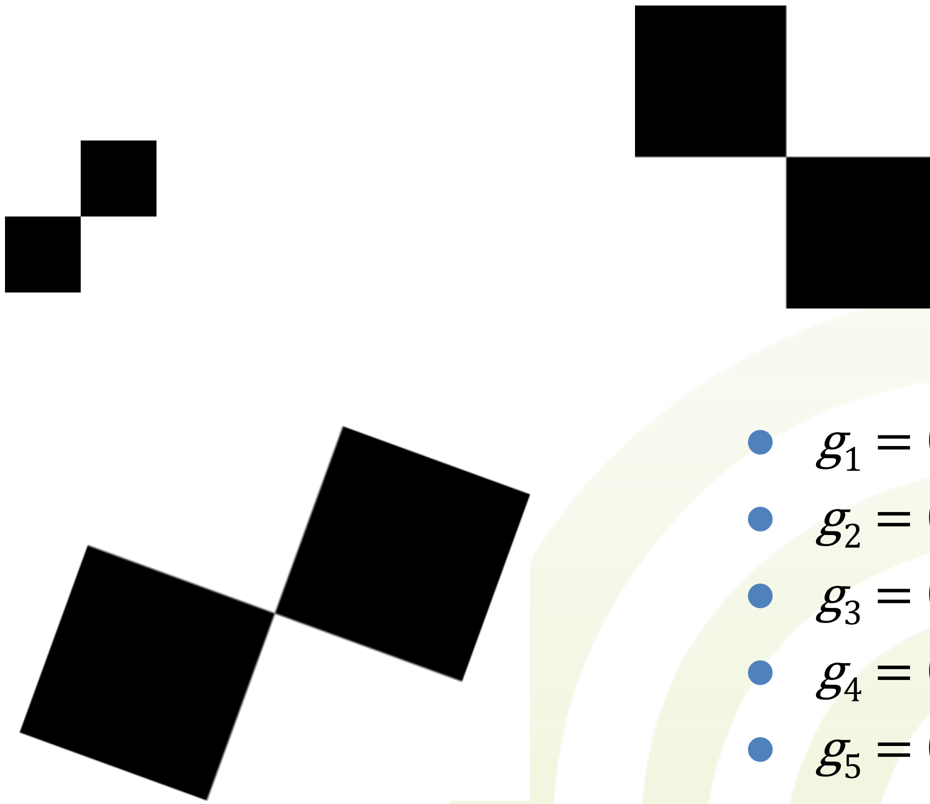
5.3 Algebraic Invariants

- $g_5(\ldots) = (\eta_{3,0} - 3\,\eta_{1,2})(\eta_{3,0} + \eta_{1,2})$
  $[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2]$
  $+ (3\,\eta_{2,1} - \eta_{0,3})\,(\eta_{2,1} + \eta_{0,3})$
  $[3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2]$

- $g_6(\ldots) = (\eta_{2,0} - \eta_{0,2})\,[(\eta_{3,0} + \eta_{1,2})^2$
  $- (\eta_{2,1} + \eta_{0,3})^2]$
  $+ 4\,\eta_{1,1}\,(\eta_{3,0} + \eta_{1,2})\,(\eta_{2,1} + \eta_{0,3})$

- $g_7(\ldots) = (3\,\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})$
  $[(\eta_{3,0} + \eta_{1,2})^2 - 3\,(\eta_{2,1} - \eta_{3,0})^2]$
  $+ (3\,\eta_{1,2} - \eta_{3,0})\,(\eta_{2,1} + \eta_{0,3})$
  $[3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2]$

5.3 Example



- $g_1 = 0{,}3299$
- $g_2 = 0{,}1697$
- $g_3 = 0$
- $g_4 = 0$
- $g_5 = 0$
- $g_6 = 0$
- $g_7 = 0$

5.3 Moment Invariants

- 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 Separability Property

- 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 Moment Invariants

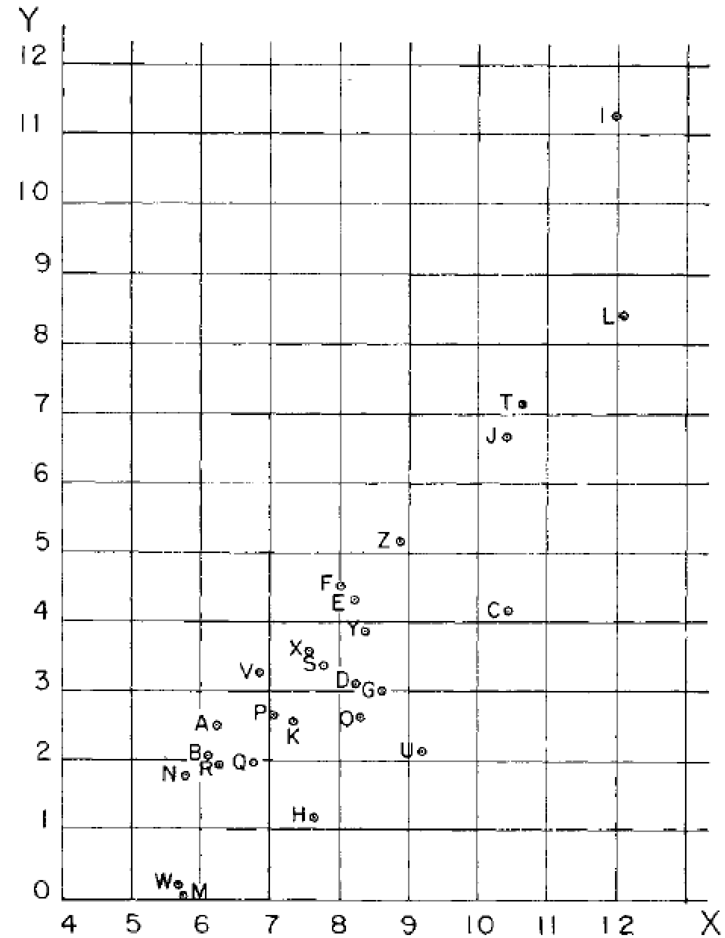
- The quality of the representation of shapes by moment invariants can be increased, by using other types of moments
- Examples:
  - Zernike moments
  - Tschebyschew moments
  - Fourier moments

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

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



5.3 Moment Invariants

- 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

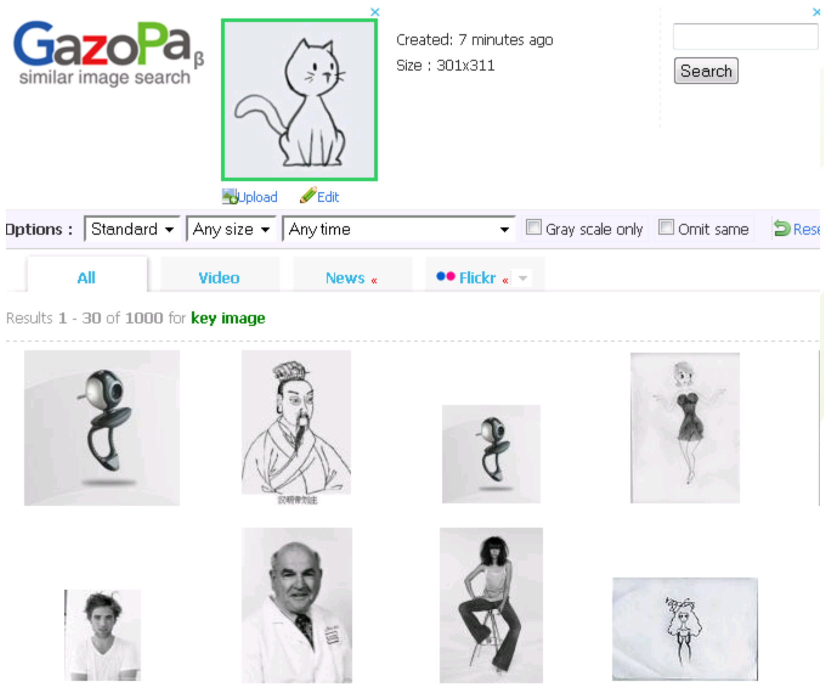
Detour

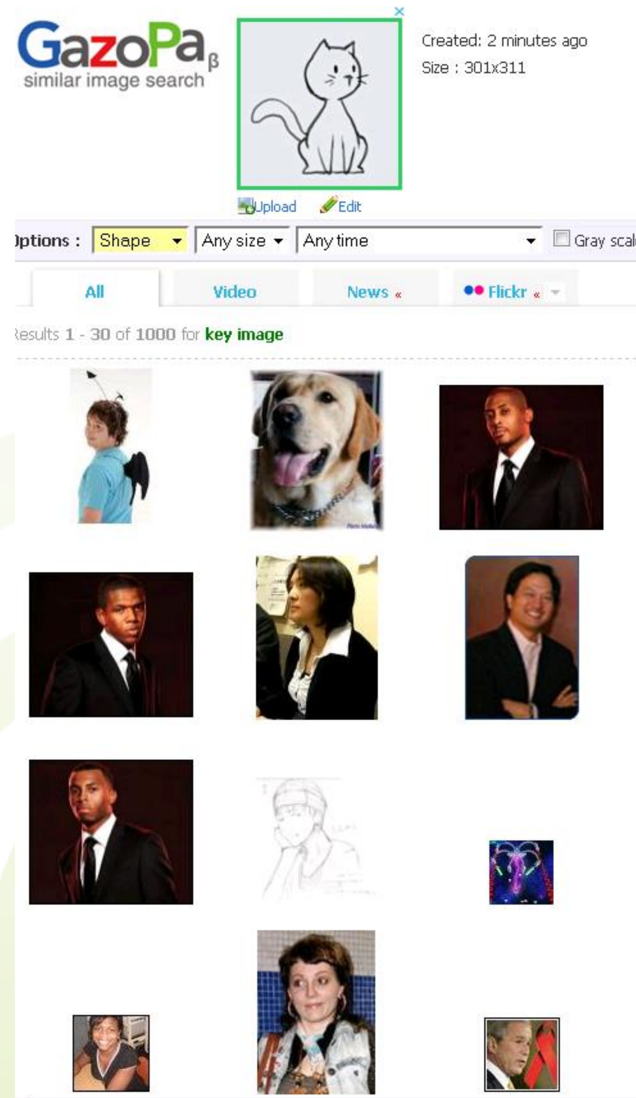
- No description of individual shapes, but of the **overall impression created by** the shapes in the picture
- Images are considered perceptionally similar if shapes occur in **similar correlations**
- Simple queries:
  - **Query by visual example**
  - **Query by sketch**

5.4 Whole Image Description

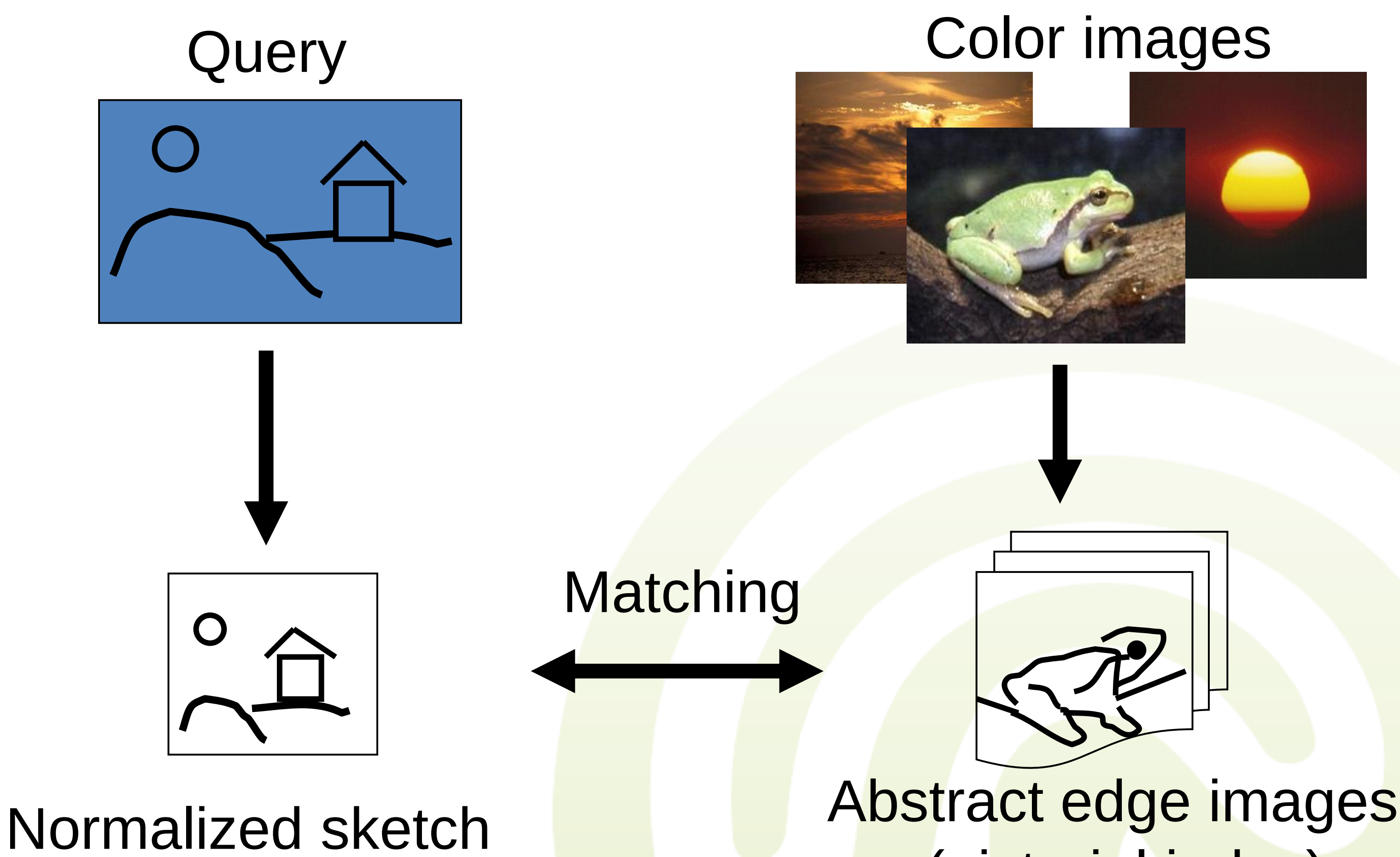
Detour

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





- 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
Color images
Matching
Normalized sketch
Abstract edge images
(pictorial index)

5.4 Query by Visual Example

Detour

- **Image abstraction** for the pictorial index
  - **Reduce the image size** to, e.g., 64 × 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

5.4 Query by Visual Example

Detour

- Compute edges with **strong local significance:**
  - All points p belonging to global edges, which also have been recognized as global edges in a 7×7 **resolution sample** around p
- Remove all global edges, which are not local
- Thinning provides the final edge image

5.4 Query by Visual Example

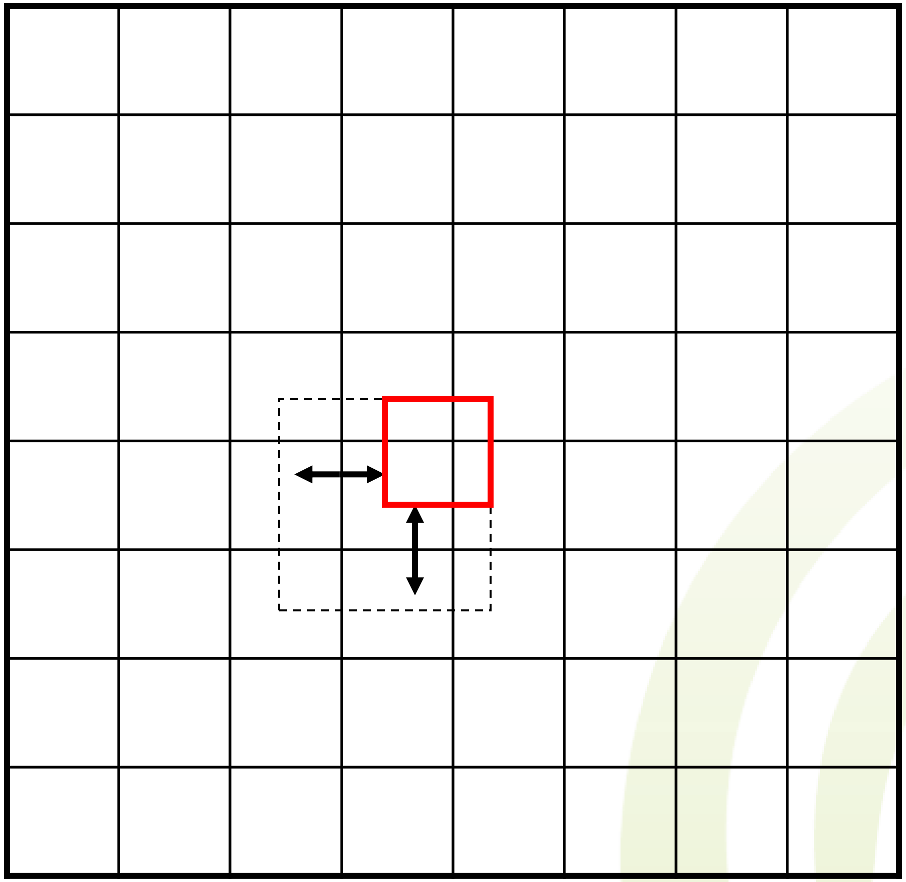
Detour

- **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

5.4 Query by Visual Example

Detour

- Calculating the **local correlation:**
  - Divide the edge image and the sketch in 8×8 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



Edge image 8 × 8 Blocks

Move the equivalent block of the sketches against the edge image and count intensity matching pixels

Shifting -4 to 4 pixels in both directions results in 64 shifts

Detour

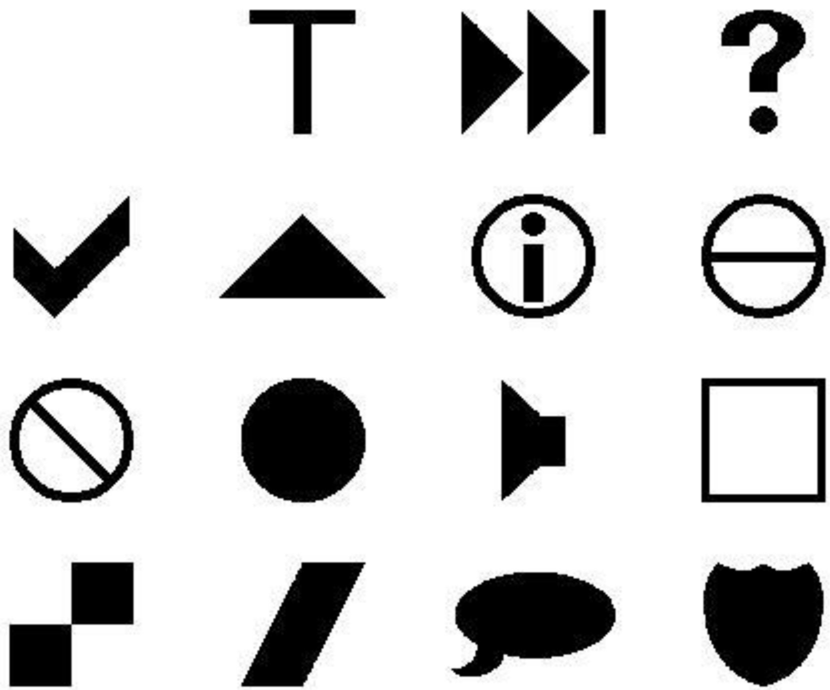
- Calculation of the **global correlation**
  - The global correlation is simply the **sum of all local correlations**
  - After calculating the global correlation for each image in the database, sort the database by correlation size

5.4 Query by Visual Example

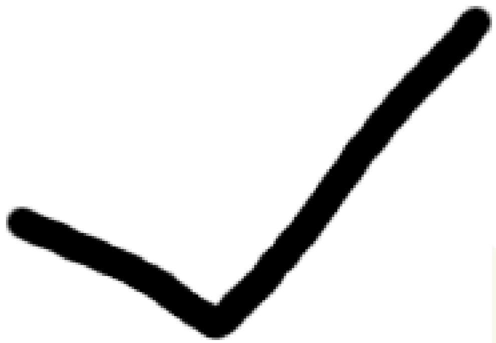
- **Advantages**
  - Good retrieval results with respect to the overall visual impression
  - Imprecision in the sketch is adjusted in matching
- **Disadvantages**
  - The calculation of similarities is **very expensive** and **can not be calculated in advance**

- Matlab example

Image database

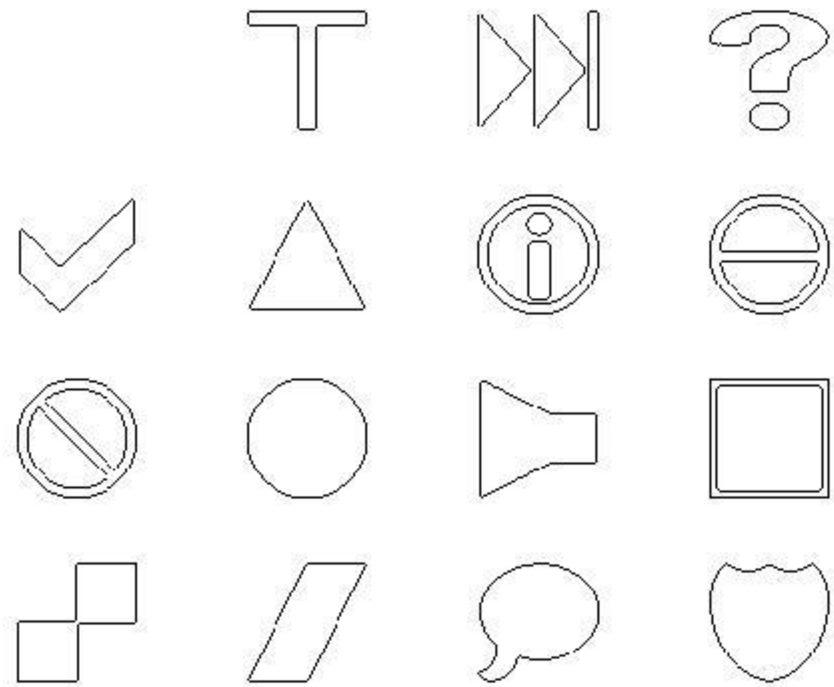


Query image

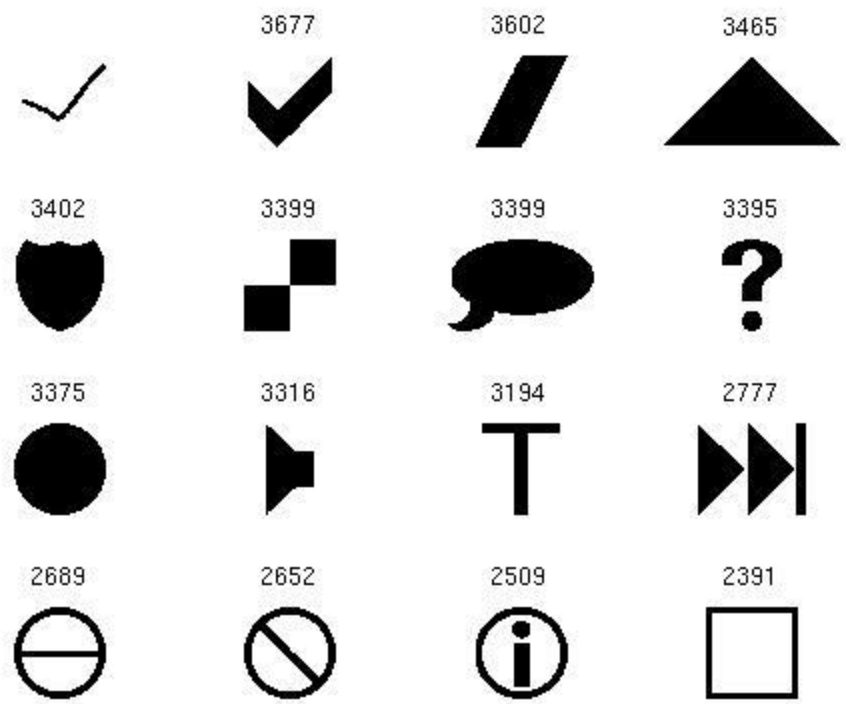


5.4 Query by Visual Example

- Extract edges



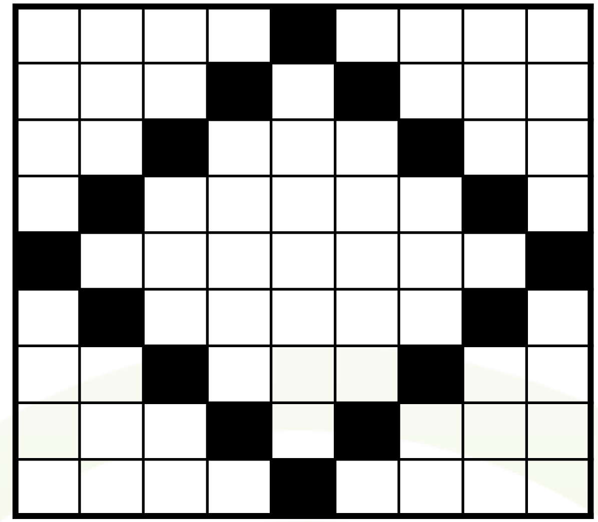
- Result

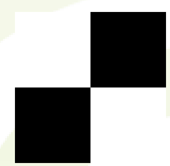

3677
3602
3465
3402
3399
3399
3395
3375
3316
3194
2777
2689
2652
2509
2391

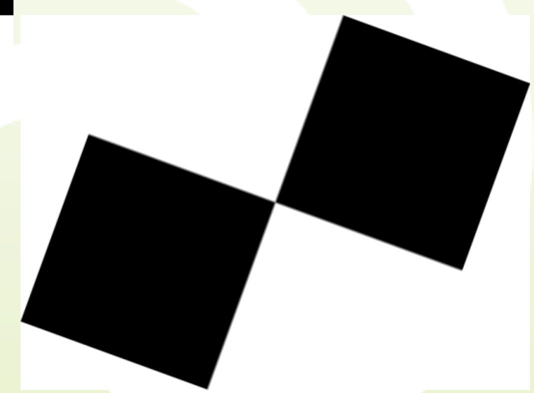
This Lecture

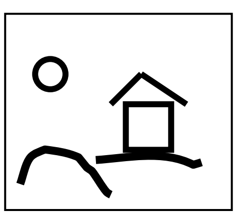
Summary

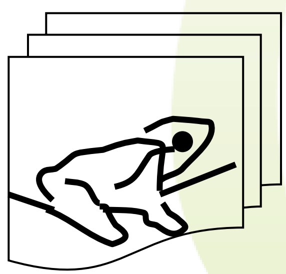
- Shape-based Features
  - Chain Codes
  - Area-based Retrieval
  - Moment Invariants
  - Query by Example











Next lecture

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