Pre-processing

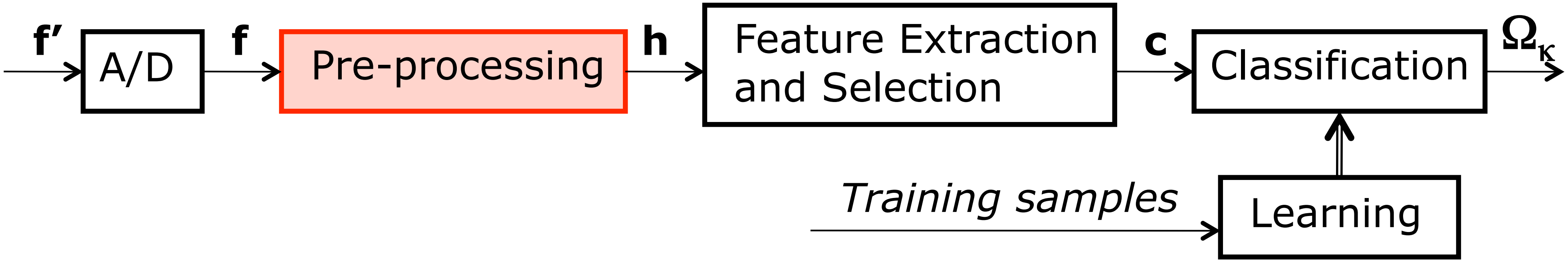
Pattern Normalization

Dr. Elli Angelopoulou

Lehrstuhl für Mustererkennung (Informatik 5)

Friedrich-Alexander-Universität Erlangen-Nürnberg

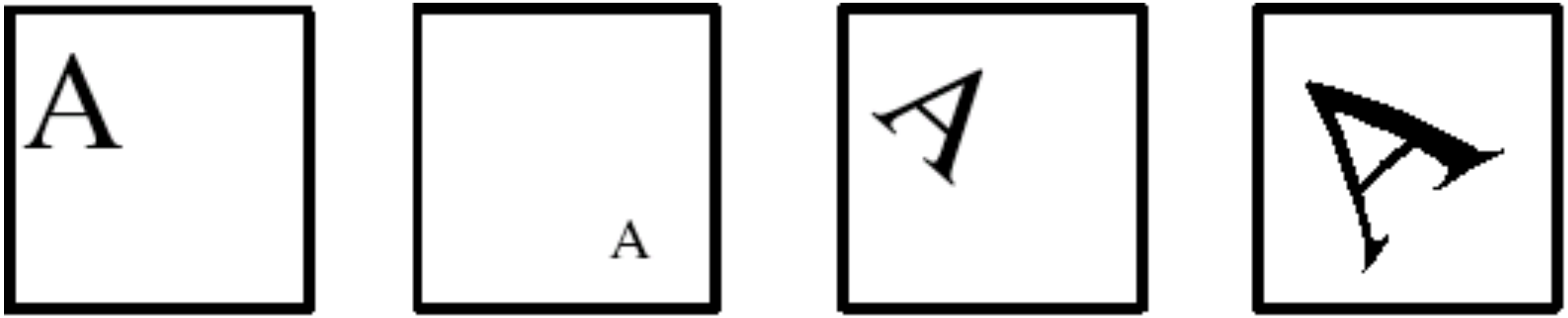
Pattern Recognition Pipeline



- Preprocessing methods we have already covered include:
  - Histogram equalization
  - Thresholding
  - Linear Shift-Invariant Filtering (low-pass and high-pass filtering)
  - Non-Linear Filtering (homomorphic transformations, cepstrum, morphological operations, rank operations)

Variations in Pattern

- Patterns (images) can vary in various parameters and still convey the same information (show the same object).



- An object can vary in:
  - Size
  - Position (translated version)
  - Pose (rotated version)
  - Non-rigid transformation
  - Any combination of the above

Normalization

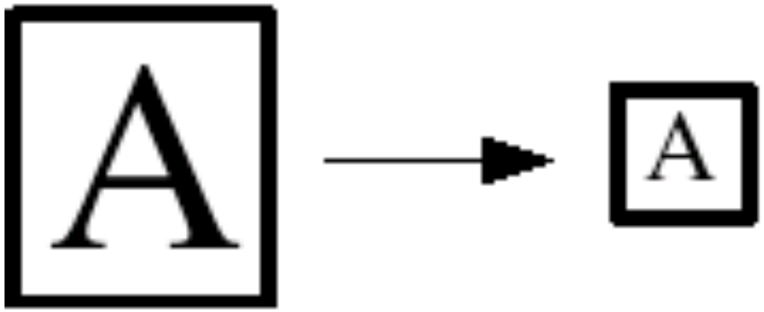
- In the context of speech recognition, a signal can vary in:
  - energy level,
  - intonation/prosody
  - Length (duration)
- Can we normalize the signal so that the recognition task is much simpler afterwards?
- The goal of normalization is to map the signal to some **normalized representation** so that the classification task is:
  - simpler (reduced complexity in storage or time),
  - more reliable (lower probability of error),
  - or both.

Normalization Parameters

- Which parameters can one normalize?
  - Size
  - Duration
  - Pose
  - Position
  - Energy
  - Illumination
  - …
- Beware: Do not normalize the parameters that are important to the classification task.

Size Normalization

- Goal: Transform the pattern (object) to a standardized size.



- Straightforward method:

1. Find the bounding box of the current size and the standardized size.
2. Compute the ratio between the two bounding boxes.
3. Use this ratio as the scaling factor.

Pose Normalization

- Pose normalization typically involves both centering the object and changing its orientation.
- What does this involve?

1. Moving the entire object so that its center coincides with the origin.
2. Rotating the object so that its axis of elongation is aligned with the vertical axis (or the horizontal axis, or any other application-specific standard orientation).

Geometric Moments

- How do I compute the location of the center and the angle of rotation?
- Use moments.
- Can be computed on either binary or gray-scale images.
- Computation on binary images is more intuitive. Object pixels are assumed to have a value 1.
- Given an image f(x,y) a geometric moment is defined as:

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

Geometric Moments - continued

- Assume that f(x,y) is a binary image.
- Geometric moments:

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

- You most probably have already used moments, but you just haven't used their formal name.
- Interpretation:
  - What is m_{00}?
  - For a binary image m_{00} computes the area of the object.
  - For a grayscale image m_{00} computes the mass (assuming that higher values map to higher density)

Center of Mass

- Assume that f(x,y) is a binary image.
- Geometric moments:

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

- Interpretation:
  - m_{00} is the area of the shape.
  - Defining the center of the shape using moments:
    - $[x_c, y_c] = [m_{10}/m_{00}, m_{01}/m_{00}]$.
    - Sometimes also denoted as $[x_s, y_s]$

Translation

- Recall that, one goal of normalization is to position an object to a standardized location.
- Typical standard location: Position the center of the object at position (0,0),
- Once we know the position of the center of mass we have sufficient information to normalize its position.
- The center of mass should be changed from (x_s, y_s) to (0,0).

$$(x_s, y_s) \rightarrow (0,0)$$

Translation - continued

- Goal: shift the center of mass from (x_s, y_s) to (0,0).
- Take every object pixel and translate it by

$$t=(-x_s, -y_s)$$

- For every object pixel (x,y) in f(x,y):

$$x' = x - x_s$$

$$y' = y - y_s$$

- If we want to have a unit mass object, we also replace f(x,y) with:

$$h(x',y') = f(x,y)/m_{00}$$

Translation – unit-mass object

- The image h(x,y) that results by setting:

$$x' = x - x_s$$

$$y' = y - y_s$$

$$h(x', y') = f(x, y) / m_{00}$$

- has its center at (0,0), i.e. ${}^{h}m_{10} = {}^{h}m_{01} = 0$
- and a center of mass set to 1, i.e. ${}^{h}m_{00} = 1$
- This is another form of standardizing an image position and mass.

Central Moments

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

- Like raw moments, but computed with respect to the *centroid* of the shape.
- What is μ_{00}?

Central Moments

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

- Like raw moments, but computed with respect to the *centroid* of the shape.
- What is μ_{00}?
  - μ_{00} = the area of the shape.
- μ_{10} = ?

Central Moments

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

- Like raw moments, but computed with respect to the *centroid* of the shape.
- What is mu_{00}?
  - μ_{00} = the area of the shape. What is μ_{10}?
- μ_{10} is the x coordinate of the centroid of the shape, *in a coordinate system whose origin is that centroid.* So, $\mu_{10} = \mu_{01} = 0$.

Central Moments

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Image 1

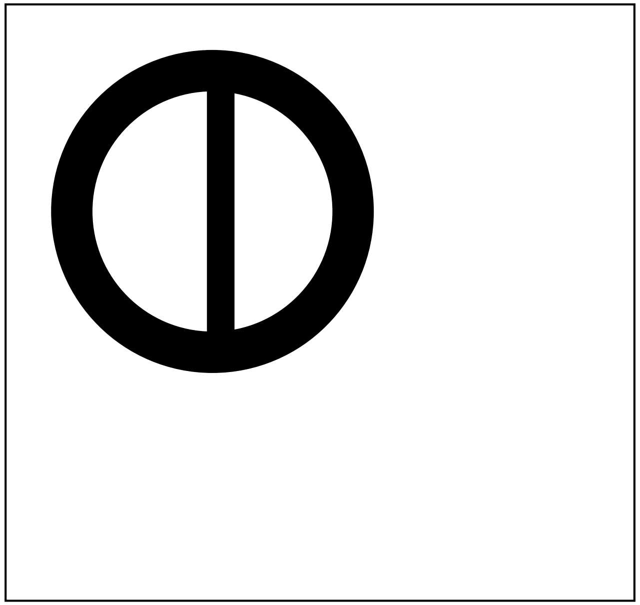
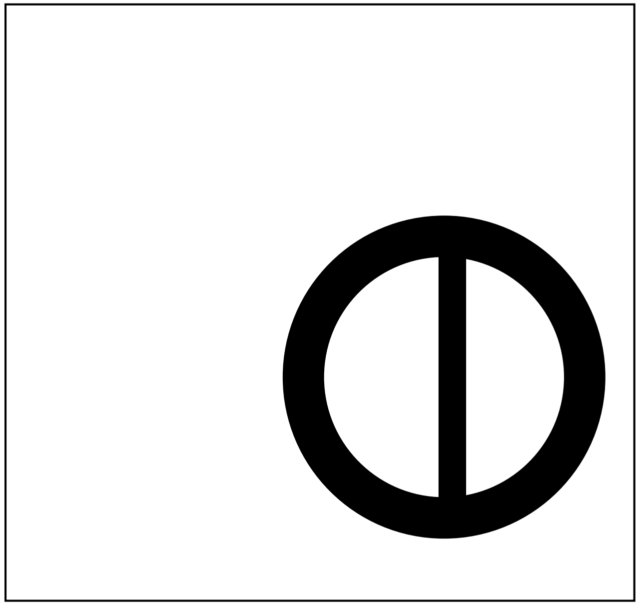


Image 2



- Will the raw moments be equal?
- Will the central moments be equal?

Central Moments

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Image 1

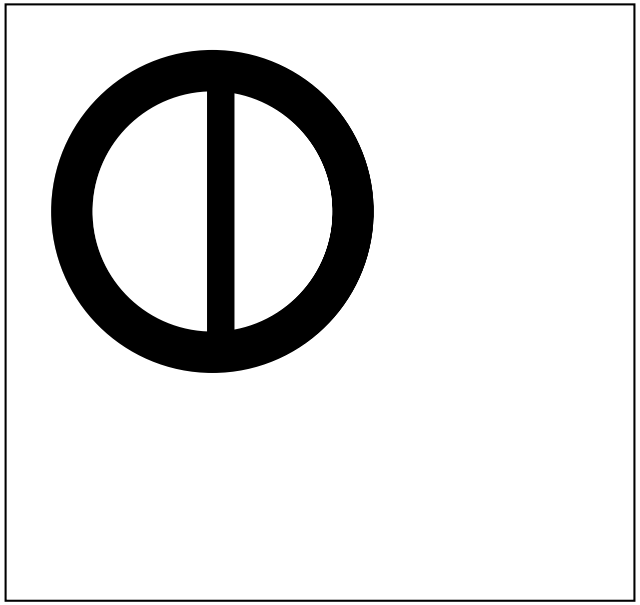
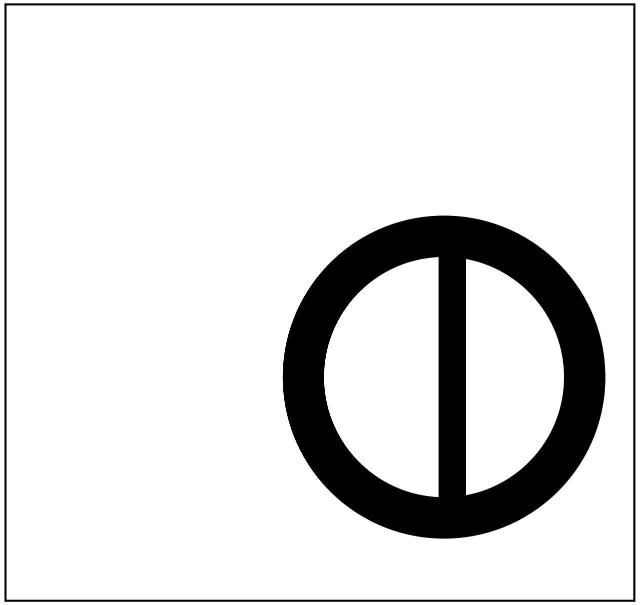


Image 2



- Will the raw moments be equal? No.
- Will the central moments be equal? Yes.

Central Moments

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x,y)$$

Image 1

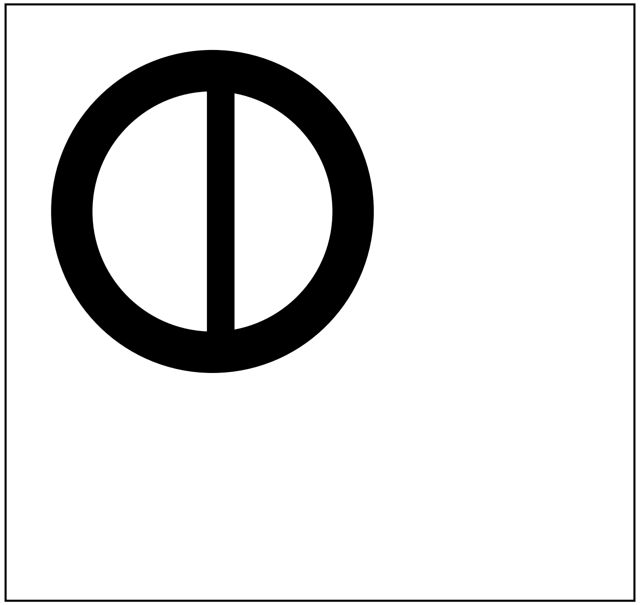
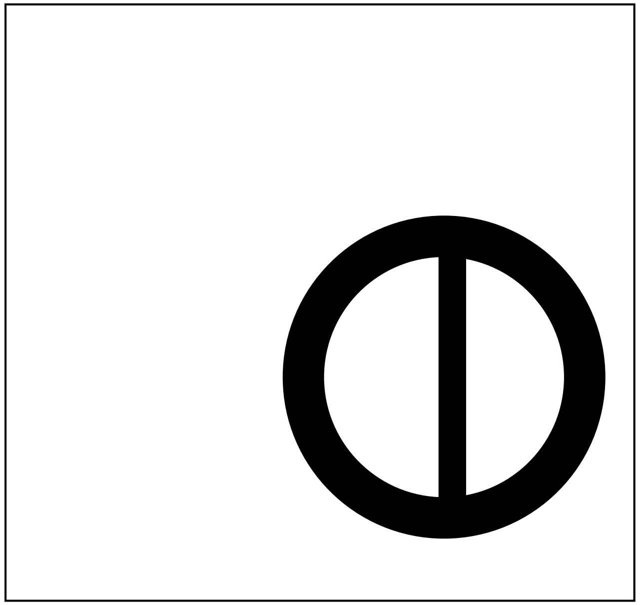


Image 2



- Central moments are *translation invariant.*

Central Moments

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Image 1

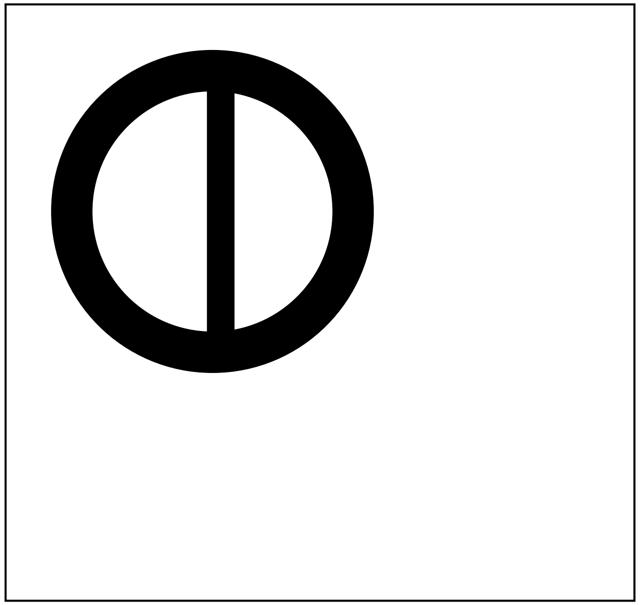
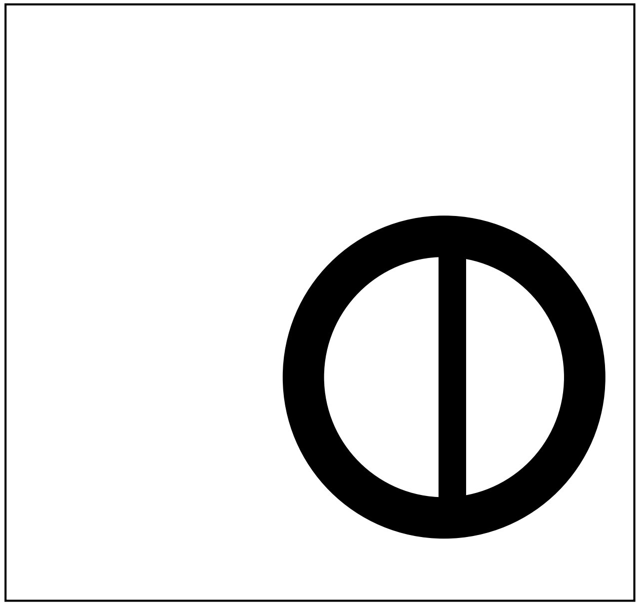


Image 2



- How can we make these moments *translation and scale invariant*?

Normalized Central Moments

For i+j >= 2: $$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}$$

For i = 1, j = 0,
or i = 0, j = 1:
Just divide by shape area.

Image 1

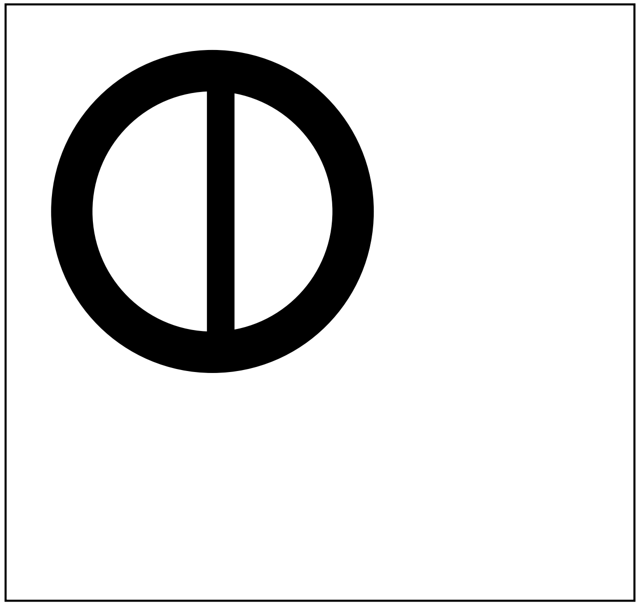
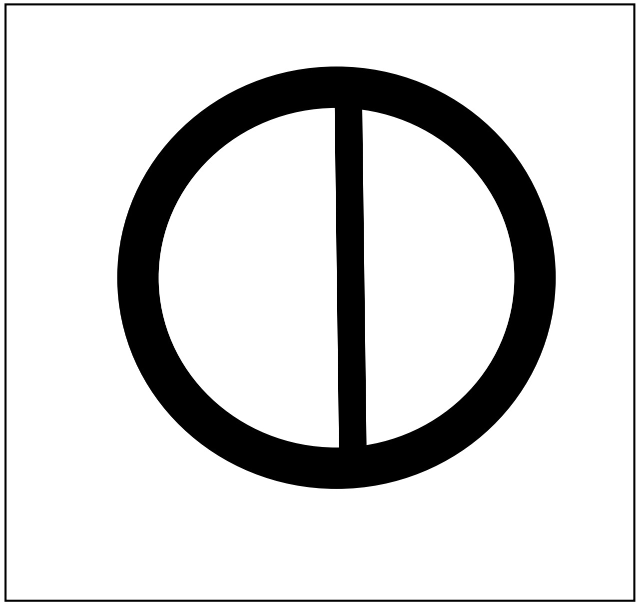


Image 2



- Normalized central moments are *translation and scale invariant.*

Pose Normalization

- Pose normalization typically involves both centering the object and changing its orientation.
- What does this involve?

1. Moving the entire object so that its center coincides with the origin.

2. Rotating the object so that its axis of elongation is aligned with the vertical axis (or the horizontal axis, or any other application-specific standard orientation).

Pose Normalization

- Pose normalization typically involves both centering the object and changing its orientation.
- What does this involve?

1. Moving the entire object so that its center coincides with the origin.
2. Rotating the object so that its axis of elongation is aligned with the vertical axis (or the horizontal axis, or any other application-specific standard orientation).

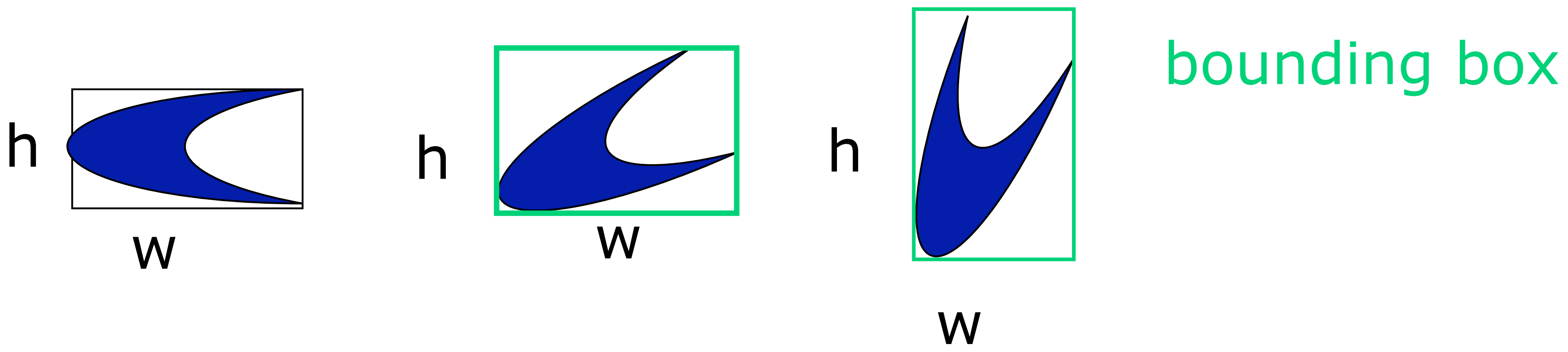
Axis Alignment

- In order to align the axis of elongation of the object with a standardized direction, we need to:
- Find the axis of elongation, also known as the principal axis.
- Find the angle θ between the axis of elongation and the desired direction.
- Once θ is known, rotate every pixel to the new desired position as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

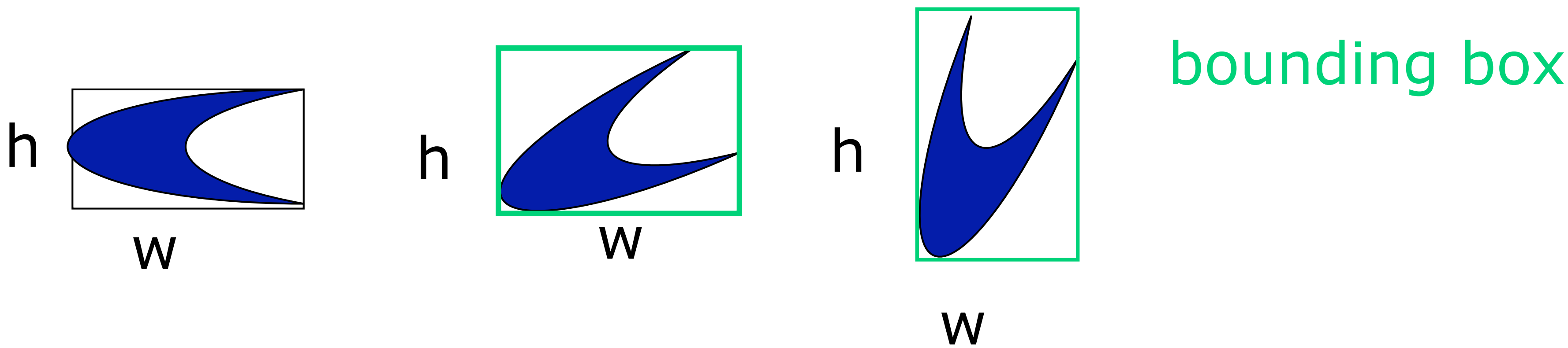
Covariance Ellipse

- The height and width of an object depend on the orientation of the object

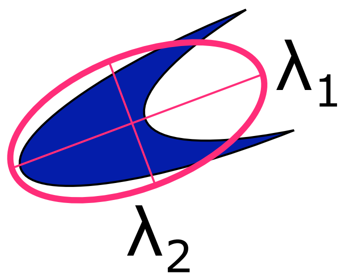


Covariance Ellipse - continued

- Height and width of an object depend on the orientation of the object



- whereas the eigen values of the covariance C_P are invariant



Computing the Axis of Elongation

- The covariances define an ellipse. The direction and length of the major axis of the ellipse are computed by principle component analysis.
- Find a rotation Φ, such that

$$\Phi C_P \Phi^T = \Lambda,\ \text{such that } \Lambda \text{ is diagonal}$$

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

$$\Phi = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}, \Phi^T \Phi = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Phi C_P \Phi^T \Phi = \Phi C_P = \Lambda \Phi$$

Φ eigen vectors, Λ eigen values

Moments - summarized

- Input: a binary image.
- Let S be the sum of all white pixels.
- 1st moment (μ_i,μ_j) (center of gravity)
- 2nd moments (covariance)

$$S = \sum_i \sum_j I(i,j)$$

$$\mu_i = \tfrac{1}{S} \sum_i \sum_j I(i,j) \cdot i, \mu_j = \tfrac{1}{S} \sum_i \sum_j I(i,j) \cdot j$$

$$\sigma_{ii}^2 = \tfrac{1}{S} \sum_i \sum_j I(i,j) \cdot (i - \mu_i)^2$$

$$\sigma_{ij}^2 = \sigma_{ji}^2 = \tfrac{1}{S} \sum_i \sum_j I(i,j) \cdot (i - \mu_i)(j - \mu_j)$$

$$\sigma_{jj}^2 = \tfrac{1}{S} \sum_i \sum_j I(i,j) \cdot (j - \mu_j)^2$$

$$C_p = \begin{pmatrix} \sigma_{ii}^2 & \sigma_{ij}^2 \\ \sigma_{ji}^2 & \sigma_{jj}^2 \end{pmatrix}$$

Sources

1. A number of slides is based on the material of V. Athitsos http://vlm1.uta.edu/~athitsos/courses/cse6367_spring2010/lectures.html and D. Hall http://www-prima.imag.fr/perso/Hall/Courses/FAI05/Session4.ppt