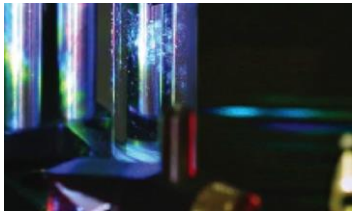
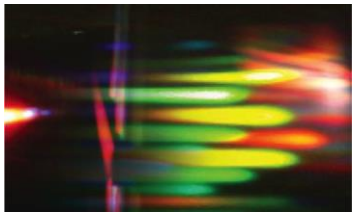
Imperial College London





COMP70058 Computer Vision

Lecture 7 – Shape Representation and Matching

Stamatia (Matina) Giannarou, PhD
The Hamlyn Centre for Robotic Surgery

stamatia.giannarou@imperial.ac.uk

The Hamlyn Centre
for Robotic Surgery

Contents

- Template matching
- Shape Discriminants
  - Area, Perimeter, Elongatedness
  - Moments, Direction
  - Energy, Entropy
  - Signatures
- Shape discriminants based on geometric features

Example Videos for Shape Recognition
http://www.youtube.com/watch?v=SLYgvHzAm2w
http://www.youtube.com/watch?v=ziliaA8K3Dg
http://metaio.com

Breast cancer cells dividing and moving
http://www.youtube.com/watch?v=Hm03rCUODqg

Red blood cells moving
http://www.youtube.com/watch?v=bNiXfFv0IGc

Manufacturing
http://www.youtube.com/watch?v=TTnho9-i6dI
http://www.youtube.com/watch?v=bPd7II_6ws0

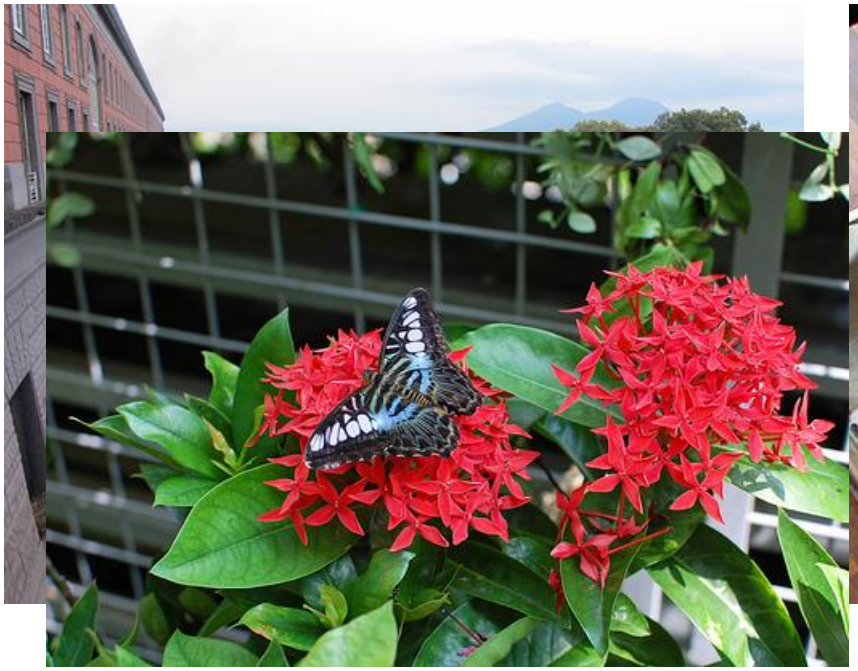
Shape Recognition Applications

- Object sorting
- Orienting objects
- Positioning for robot arm movements
- Security applications

Many vision techniques can be used in manufacturing for a variety of processes such as sorting objects, orienting objects or determining exact positions for robot arm movements. One dimensional systems are now commonplace and include bar code readers and counting machines. Two dimensional systems, using simply bi-level images can be used for a wide variety of tasks, and systems using grey level images are beginning to find application.

Template Matching

- The simplest solution to object recognition is to find the best match of the object with a number of characteristic templates
- However, to do this exhaustively, we need to check all possibilities
- This can be done with cross-correlation



Template Matching

- Cross-correlation defines a way of combining two functions. In the most general form it is defined as a continuous integral which in two dimensions is:

$$C(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(p,q)h(p+x,q+y)dpdq$$

The function h in the above equations is usually called a template

- For our purposes we are interested in discrete functions of finite size. For these, the cross-correlation integral simplifies to:

$$C(x,y) = \sum_{p=1}^{x_{res}}\sum_{q=1}^{y_{res}} f(p,q)h(p+x,q+y)$$

where f(x,y) is the discrete function defining the pixel values, h(u,v) is the template function, and [1..x_{res},1..y_{res}] is the range over which f(x,y) is non zero, i.e., the resolution of the picture. The normal restriction on the above equations is that the functions are symmetric about zero, i.e.,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(p,q)dpdq = 0 \; and \; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(p,q)dpdq = 0$$

For any image, this may be arranged by subtracting the average value from each image point.

Convolution and Cross-Correlation

- Convolution is defined as:

$$g[n] = f[n] * h[n] = \sum_{m=-\infty}^{\infty} f[m]h[n-m]$$
$$= \sum_{m=-\infty}^{\infty} f[n-m]h[m]$$
$$= \sum_{m=-\infty}^{\infty} f[n+m]h[-m]$$

When you perform convolution, you **flip** the kernel.

- Cross-correlation is defined as:

$$g[n] = f[n] \star h[n] = \sum_{m=-\infty}^{\infty} f[m]h[n+m]$$

$$h[n] \star f[n] = \sum_{m=-\infty}^{\infty} f[n+m]h[m]$$

When you perform cross-correlation, you **do not flip** the kernel.

Convolution and Cross-Correlation

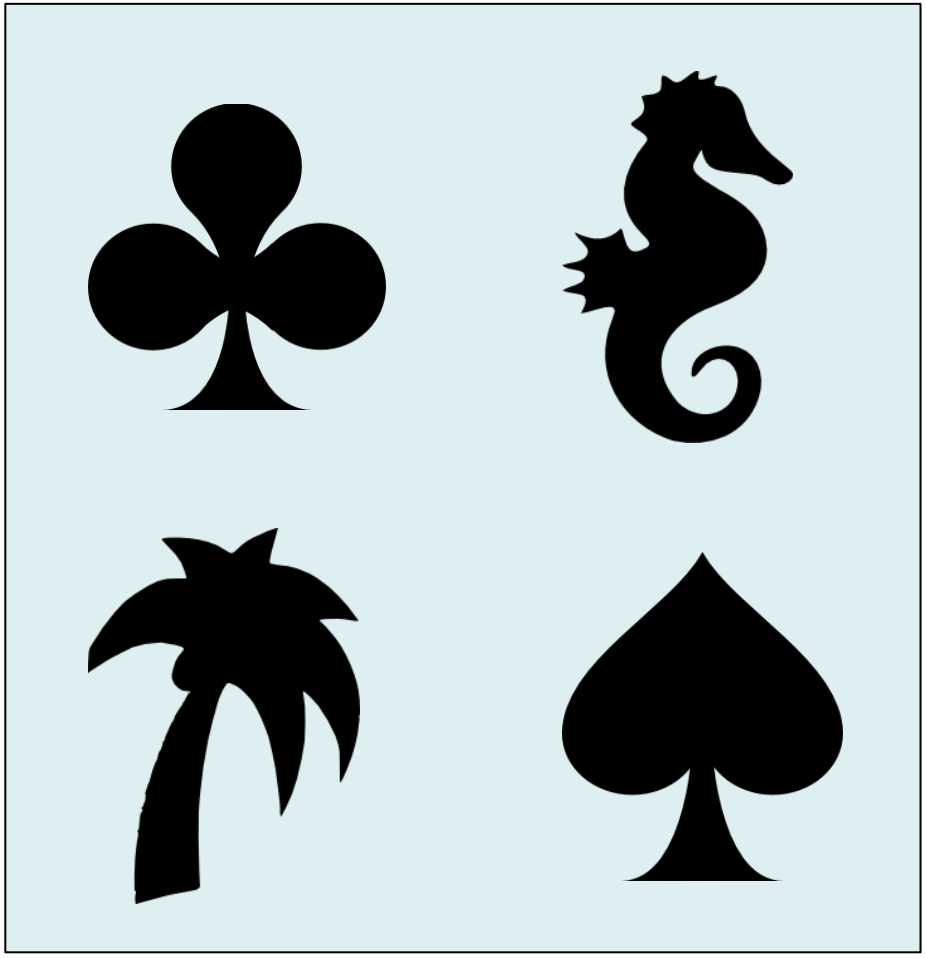
- If h is symmetric, i.e. $h[i,j] = h[-i,-j]$, convolution is equivalent to cross-correlation.

$$f[m,n] * h[m,n] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[m+i, n+j] h[-i,-j]$$

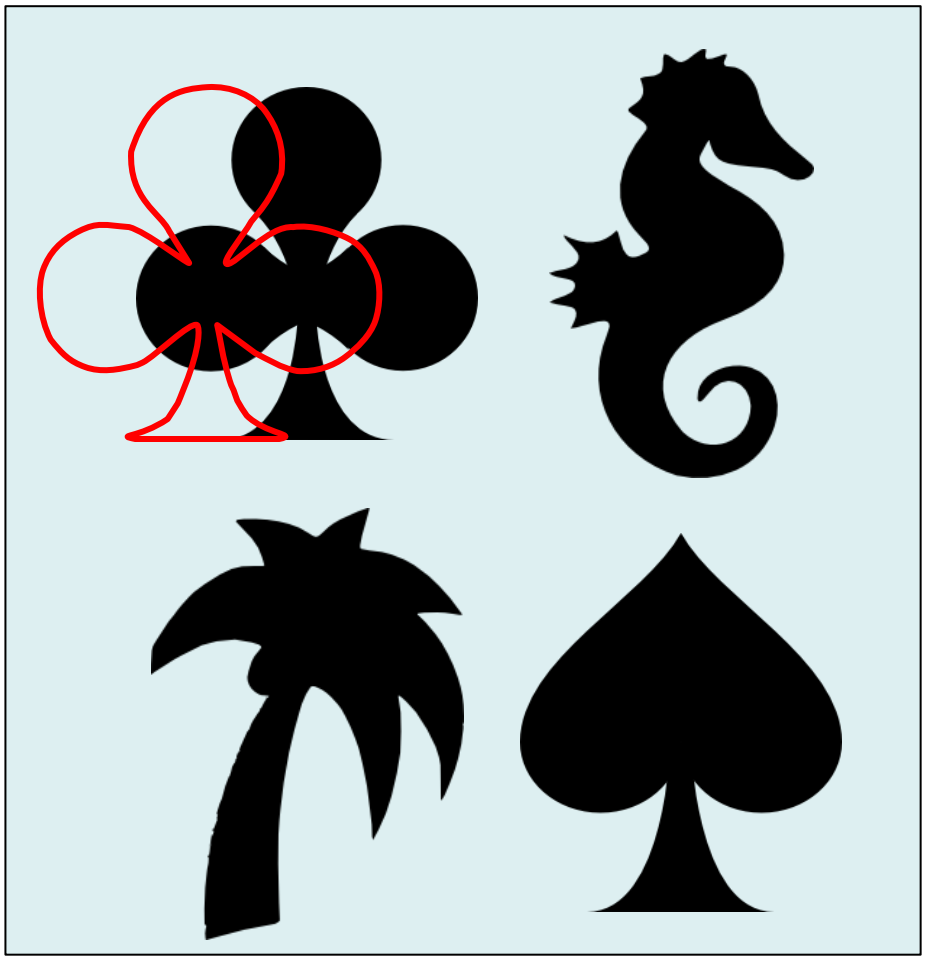
$$= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[m+i, n+j] h[i,j]$$

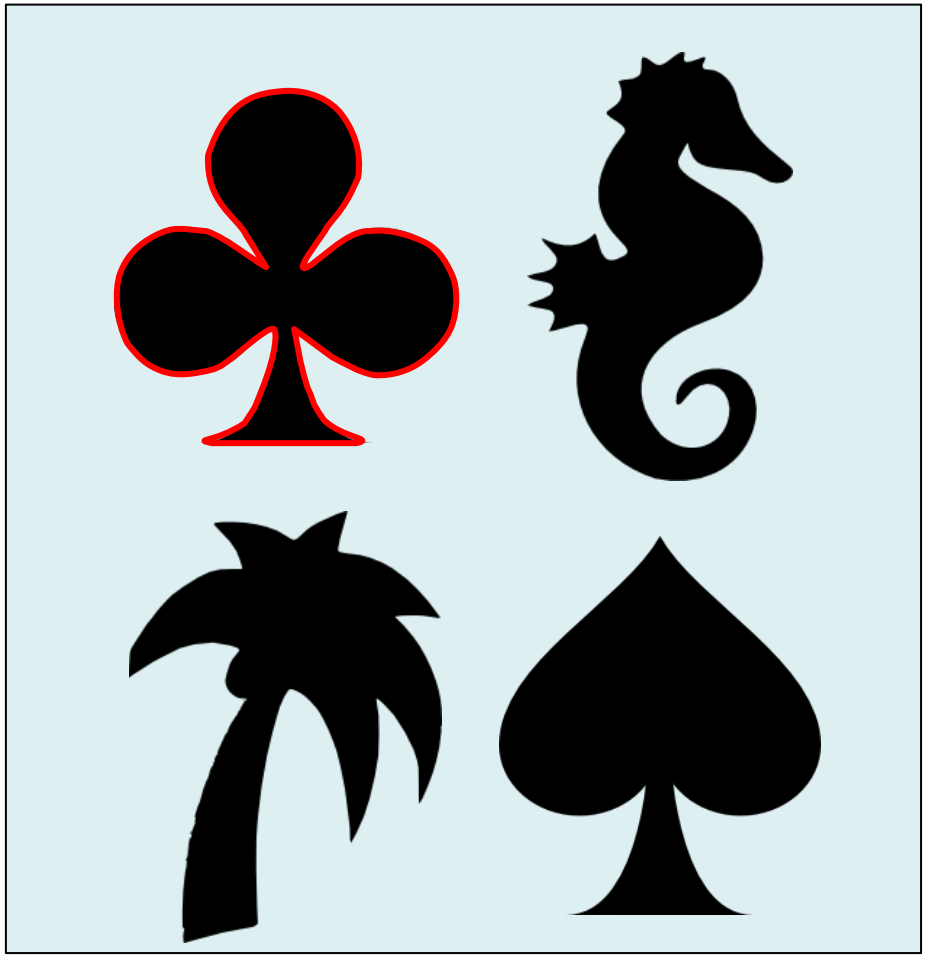
Template Matching

- In the simplest case, we need to determine the best fit of the template to the image. This is the point where C(x,y) is maximum, and the (x,y) determines the position in the image where the template is recognised.

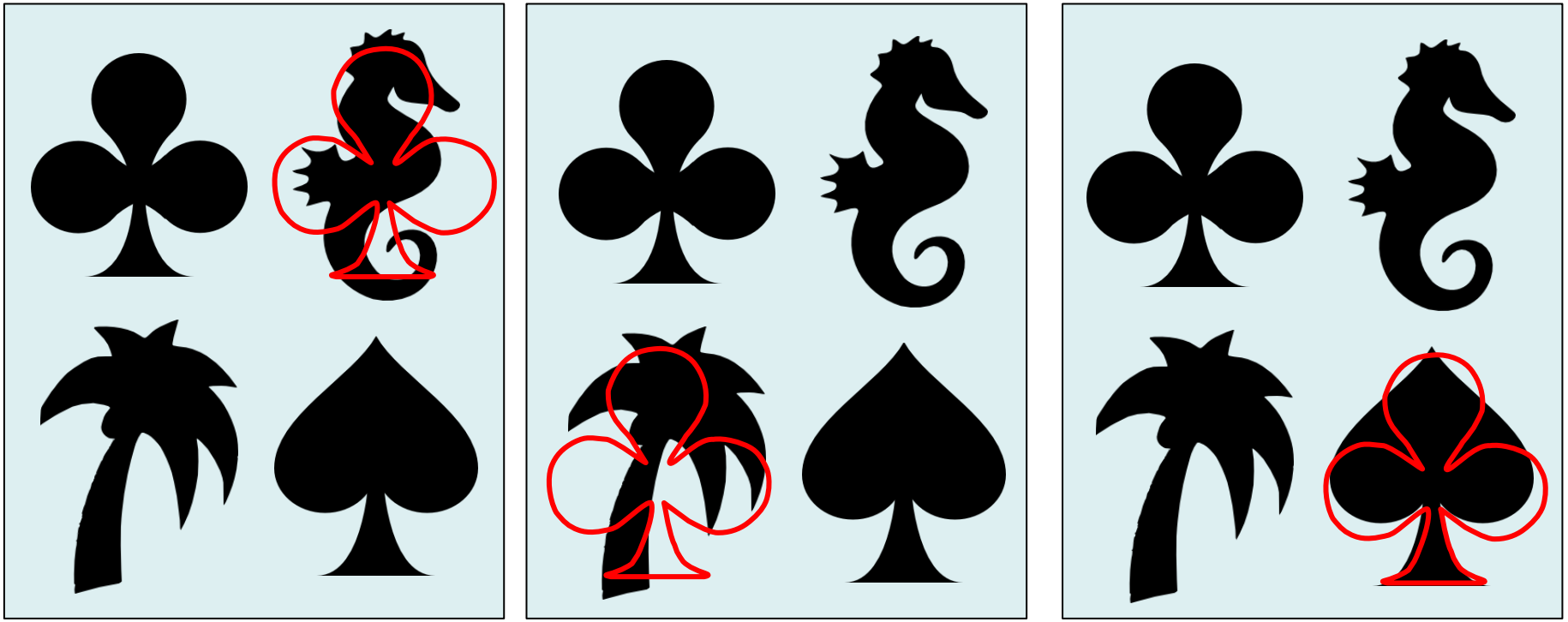


Template Matching





Template Matching



Template Matching

Fourier Transform for Template Matching

- A theorem which should be noted:

$$F(u)G(u) \Rightarrow \text{FT of the convolution of } f \text{ and } g$$

$$F(u)G(u)^* \Rightarrow \text{FT of the correlation of } f \text{ and } g$$

- where G(u)* is the complex conjugate of G(u)

If we can compute the transforms quickly, then we can also do template matching by simply carrying out pointwise multiplies of the spectra, and taking the inverse transform. However, this technique is rarely used, since the result for continuous space does not generalise to discrete space very well.

Fourier Transform for Template Matching

128 pixels

128 pixels

Image to be searched

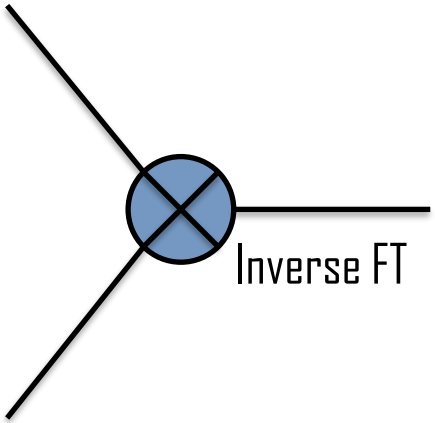
32 pixels

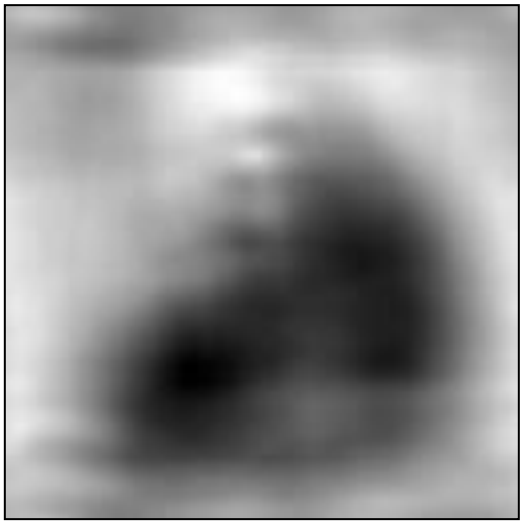
32 pixels

Target

Fourier Transform for Template Matching

FT





FT complex conjugate

Shape Discriminants

- The previous approach is very time consuming
- How can we hasten the process of separating the different objects that we wish to identify or sort?
- Use shape discriminants
  - Area
  - Perimeter
  - Elongatedness
  - Moments
  - Direction
  - Energy, entropy
  - Chord distribution

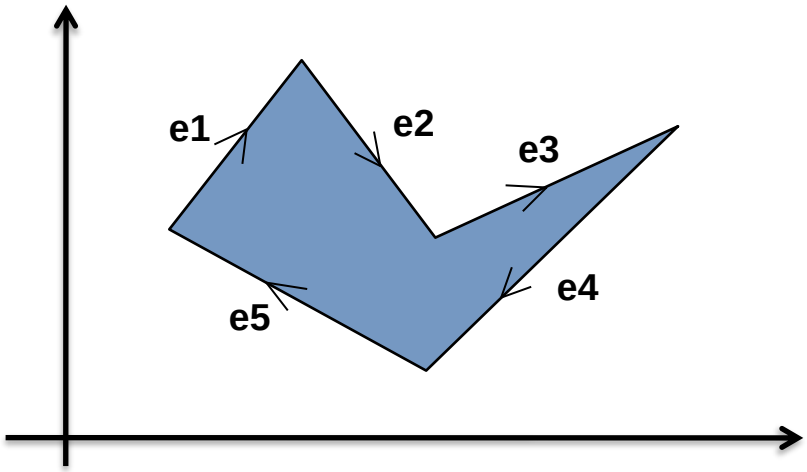
Although template matching can provide a solution for cases where the object is aligned with the axes, to check all possible positions and orientations of the templates for the best match is clearly very time consuming. Hence, one approach to define a set of discriminants which will separate out the different objects that we wish to identify or to sort.
Discriminants are any simple object property that can be computed from the image, but they must have one important property, that they are **independent of the object position**. The simpler ones depend only on the boundary of the objects, but more complex ones can involve the grey levels and the shape of the boundaries.

Area and Perimeter

- Object area and perimeter are the simplest features that can be extracted from the pixel image

Providing that the image is bi level, and that a single object is present in the image, we can of course compute the area by **summing all the pixels** of the camera image. This method is, however, inefficient since it requires every pixel in the image to be processed. Early systems **traced round the boundary** at the pixel level, but this was rather inaccurate, being prone to alias effects, particularly when trying to determine perimeters.

- In cases where a **piece wise boundary** has been determined using say a **Hough transform**, the area can be measured by processing the **edge vectors** of a polygon in order, adding the area under the edge vectors when moving to the right, and subtracting it when moving to the left. In this case, the perimeter can be computed accurately from the coordinates of the vertices.

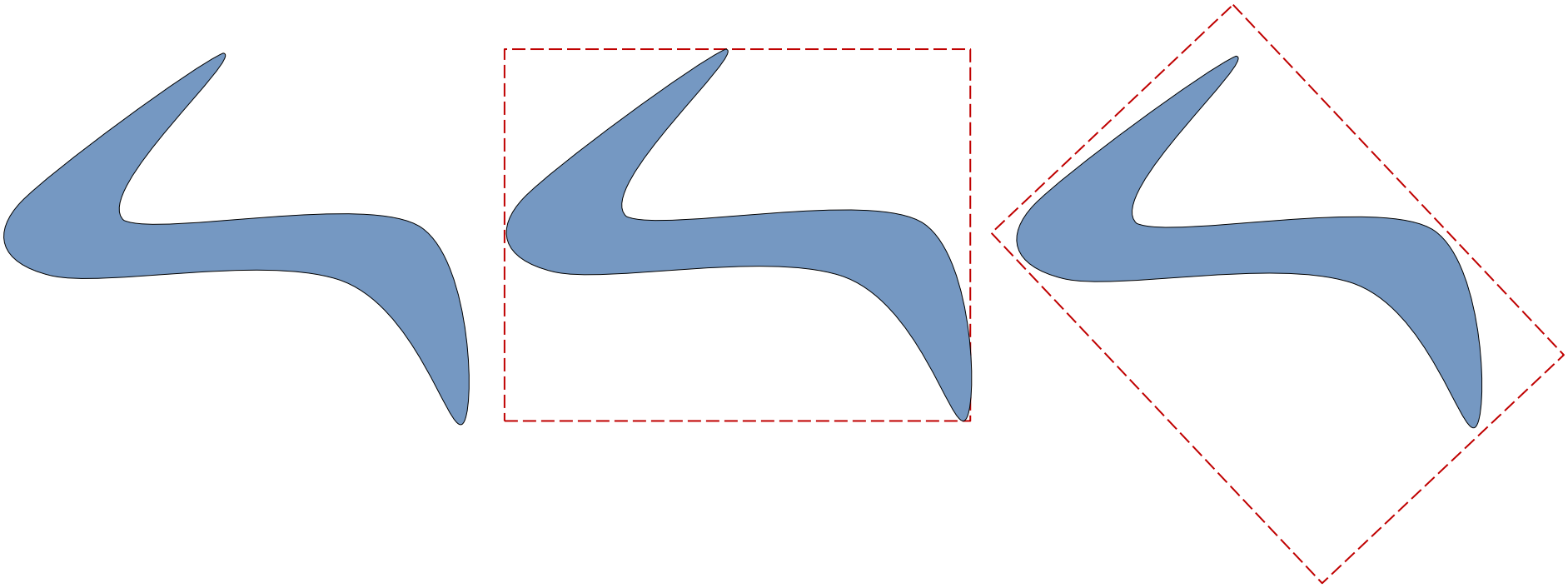


Area:
Add area under e1, e2 and e3
Subtract area under e4 and e5

Perimeter:
|e1| + |e2| + |e3| + |e4| + |e5|

Elongatedness

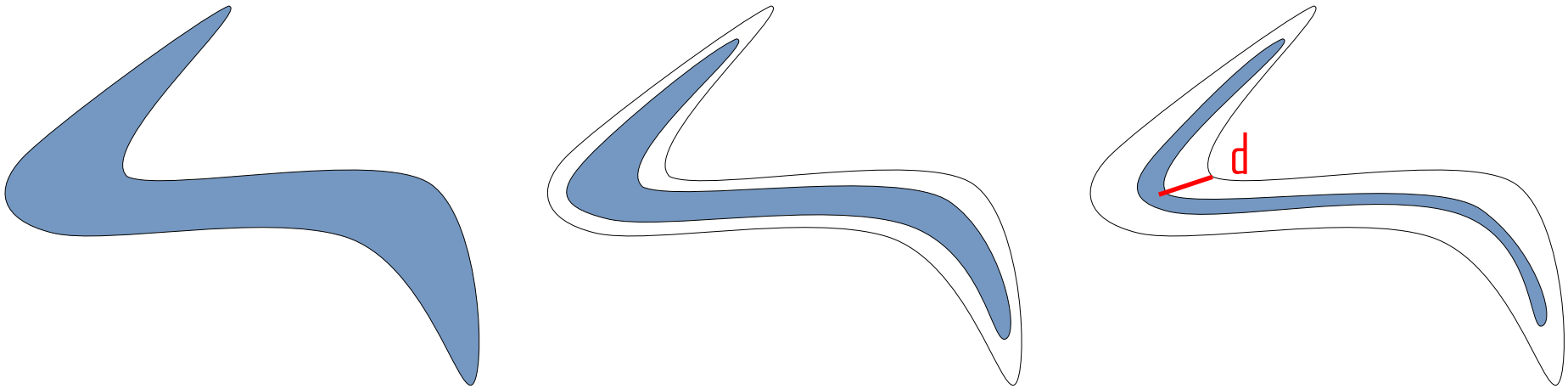
- Elongatedness is a ratio between the length and width of the region bounding rectangle - the bounding rectangle is turned in discrete steps, and a minimum located. The rectangle considered is that of **minimum area**.
- This criterion cannot succeed in curved regions, for which the evaluation of elongatedness must be based on maximum region thickness.



Elongatedness

- Elongateness can be evaluated as a ratio of the region area and the square of its thickness.
- The maximum region thickness can be determined as the number of mathematical **erosion steps** that may be applied before the region totally disappears.
- Elongatedness is independent of linear transformations - translation, rotation, and scaling.

$$Elongatedness = area/(2d)^2$$



Moments

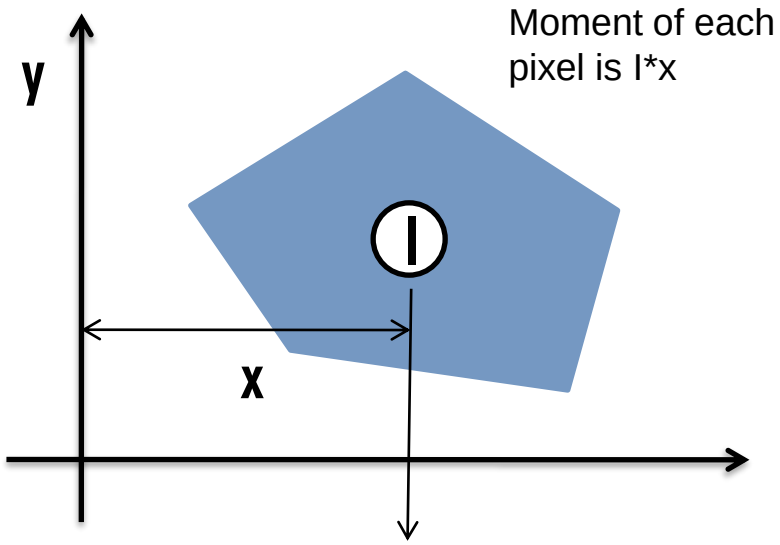
- For better recognition, moments of area are frequently used
- Moments are defined equivalently to mechanical moments treating the grey level at a pixel to be equivalent to a weight

- Formally, central moments are defined as an integral over the object

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

- In digitized images we evaluate sums

$$m_{pq} = \sum_{i=1}^{x_{res}} \sum_{j=1}^{y_{res}} i^p j^q f(i,j)$$



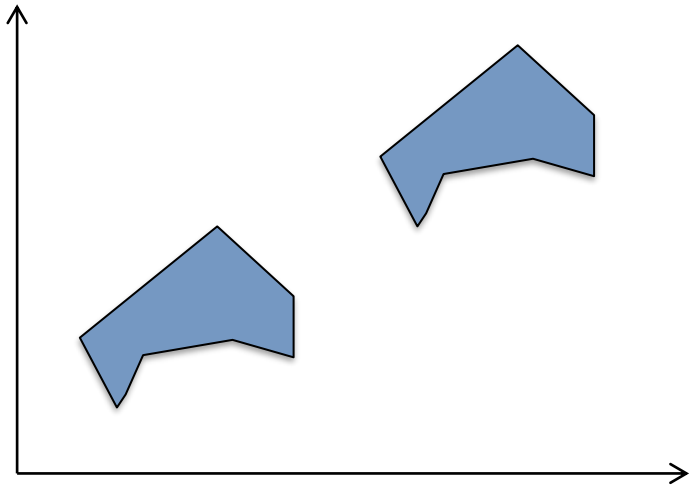
First moments of area about y

Moments – Translation Invariance

- Translation invariance can be achieved if we use central moments

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

- where x_c and y_c are the coordinates of the region's centre of gravity (**centroid**) which can be obtained using the following relationships:



$$x_c = \frac{m_{10}}{m_{00}}, \qquad y_c = \frac{m_{01}}{m_{00}}$$

Area

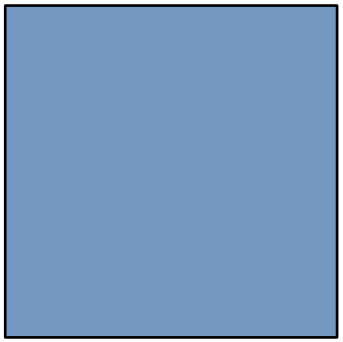
Second Moments

- The second moment gives us a measure of how **spread** out the object is
- We can look at this in terms of two components, a second moment about the x direction (p=0,q=2) and a second moment about the y direction (p=2,q=0)

$$\mu_{20} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_c)^2 f(x,y) dx dy$$

$$\mu_{02} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (y - y_c)^2 f(x,y) dx dy$$

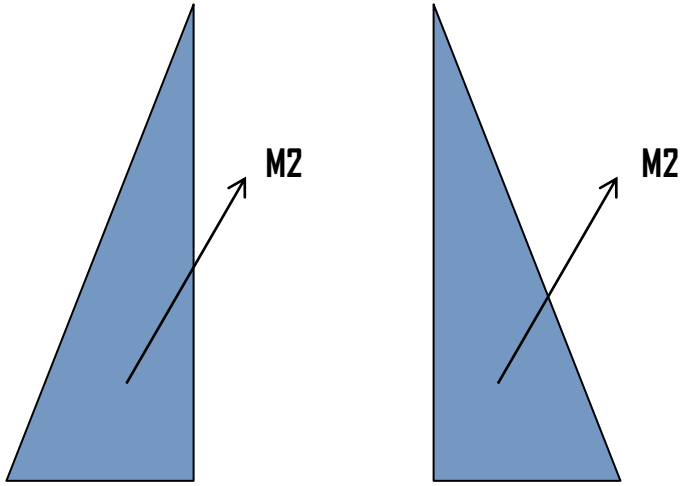
High 2nd Moment (about y)



Low 2nd Moment (about y)

Second Moments

The second moment gives us a measure of how spread out the object is. If we consider the vector formed by the two components $\mu_2=[\mu_{20}, \mu_{02}]$, its magnitude is position independent, and so can be used as a discriminant, and its direction can be used to measure the orientation of the object. This is a useful property for control of a robot gripper. Note though, that the second moment is always positive, and hence the second moment cannot detect reflected objects (see below). However, the same fact means that $\mu_{20}+ \mu_{02}$ may be used in place of the normal Euclidian distance for a magnitude discriminant. Rarely however do higher order moments yield any more discriminant power than the second.

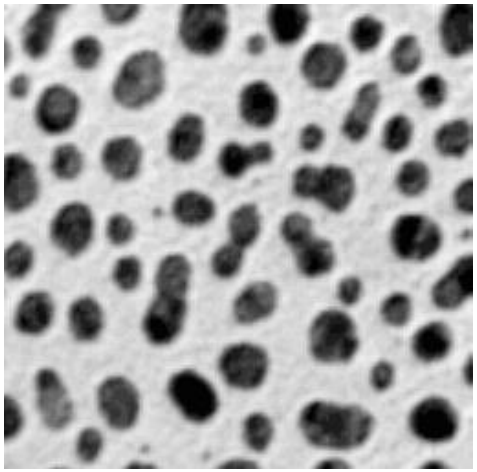


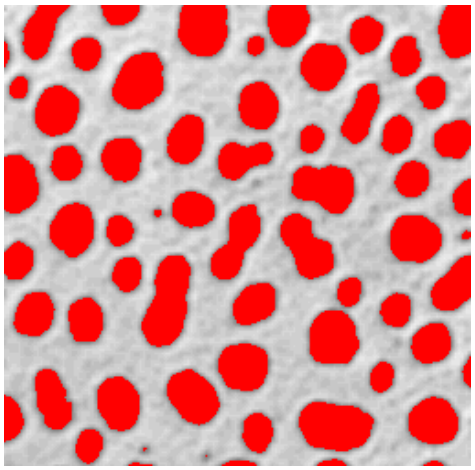
However, the second moment cannot distinguish reflected objects (e.g. upside down)

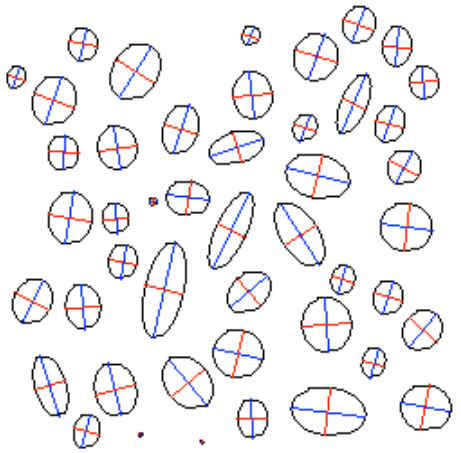
Direction

- Direction is a property which makes sense in elongated shapes only
- If the region is elongated, direction is the direction of the longer side of minimum bounding rectangle
- If the shape moments are known, the direction θ can be computed as

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$$

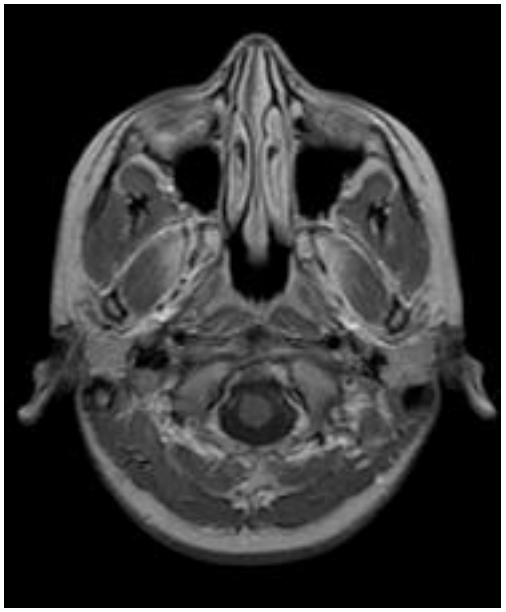


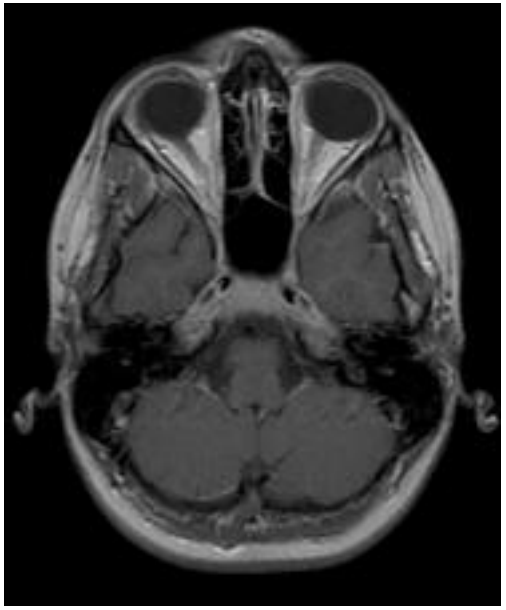




Energy and Entropy

- Energy and entropy can be applied to objects where the shade or texture may have given characteristics.
- The image intensities need to be normalised before the discriminants are computed to account for varying lighting conditions





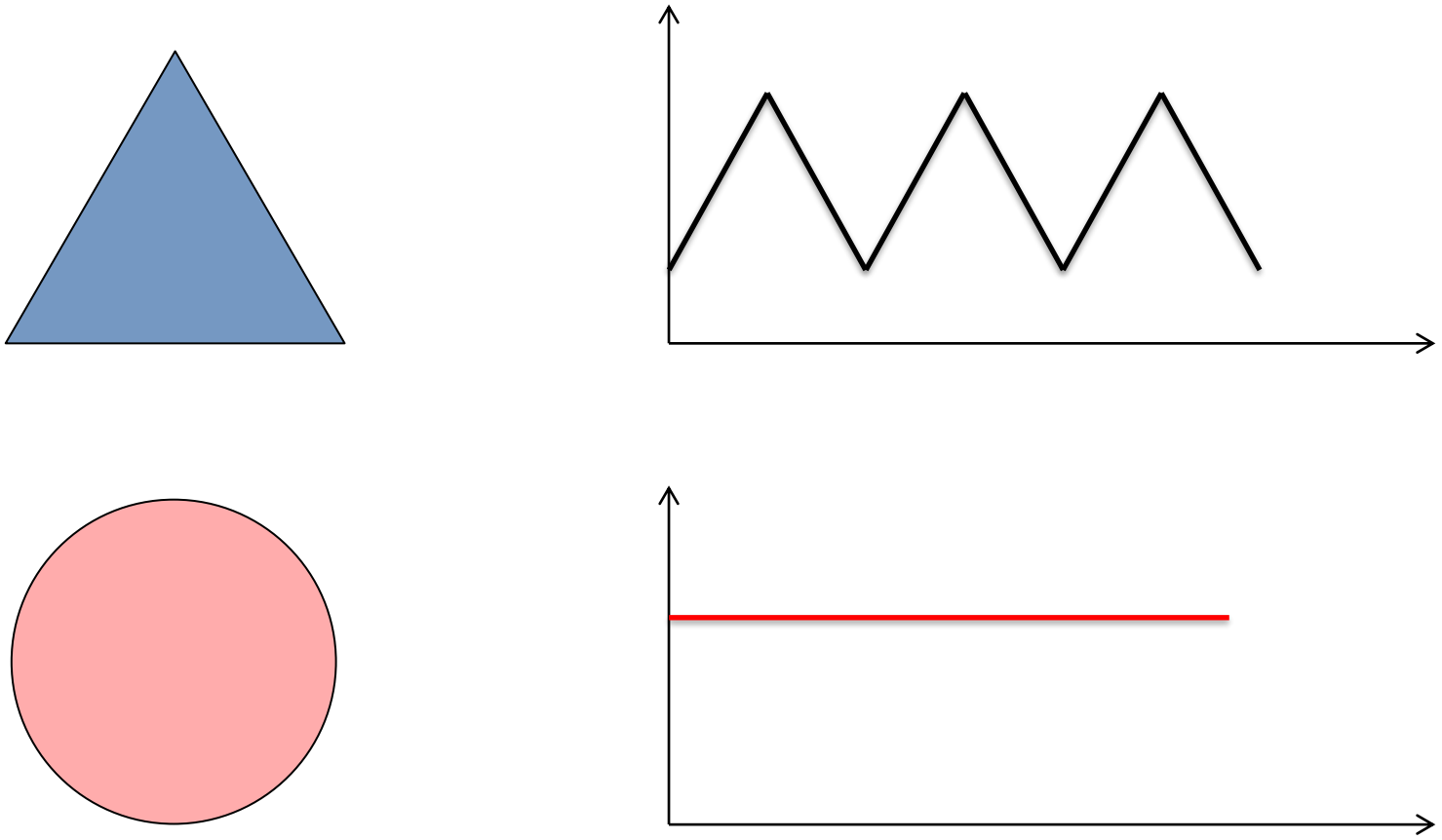
$$Energy = \sum_{K} P_k^2$$

$$Entropy = -\sum_{K} P_k \log(P_k)$$

Probability of grey level k

Signature

- The signature represents a shape using a one dimensional function derived from the shape's boundary points
- Many signatures exist; the ones shown below are the centroid distance



Discriminants Based on Geometric Features

- Area, perimeter, moments energy and entropy do not rely on any particular feature of the objects being recognised, and this makes them widely applicable for **sorting** and **orienting** objects.
- However, if the number of objects is great they may not be sufficient to discriminate all possibilities. In these cases it becomes necessary to extract **features**, and these are **object specific**.
- Possible features for discrimination, for objects with polygonal outline are:
  - Number of vertices
  - Distance of each vertex from the centre of gravity
  - Angle subtended by the vertices at the centre of gravity

Conclusions

- Template matching
- Shape Discriminants
  - Area, Perimeter, Elongatedness
  - Moments, Direction
  - Energy, Entropy
  - Signatures
- Shape discriminants based on geometric features