

COMP3005: Computer Vision

# Shape description and modelling

Jonathon Hare  
[jsh2@ecs.soton.ac.uk](mailto:jsh2@ecs.soton.ac.uk)

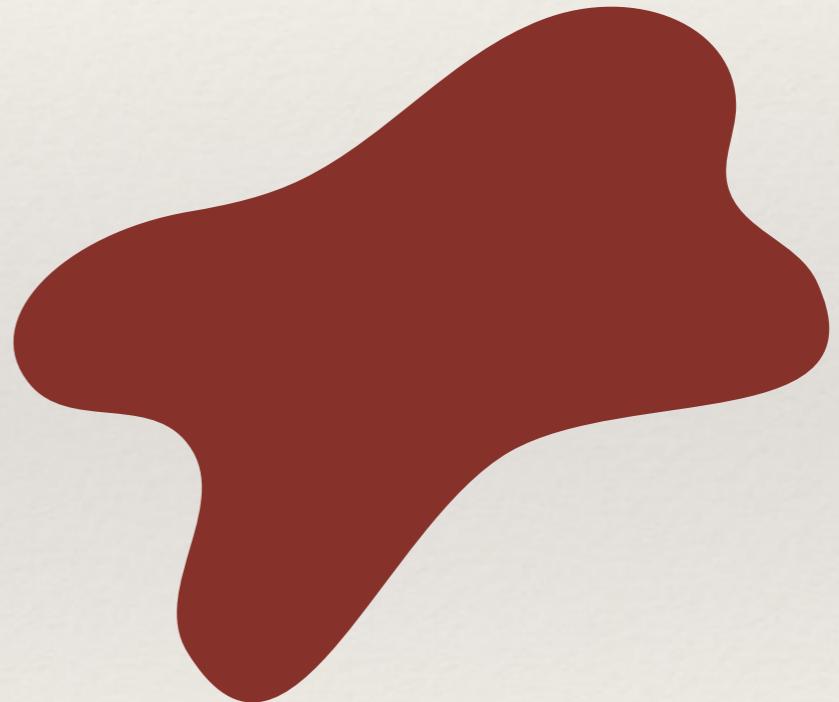
Extracting features from shapes  
represented by connected  
components

---

# Recap: Connected Component

---

*A connected component is a set of pixels in which every pixel is connected either directly or through any connected path of pixels from the set.*



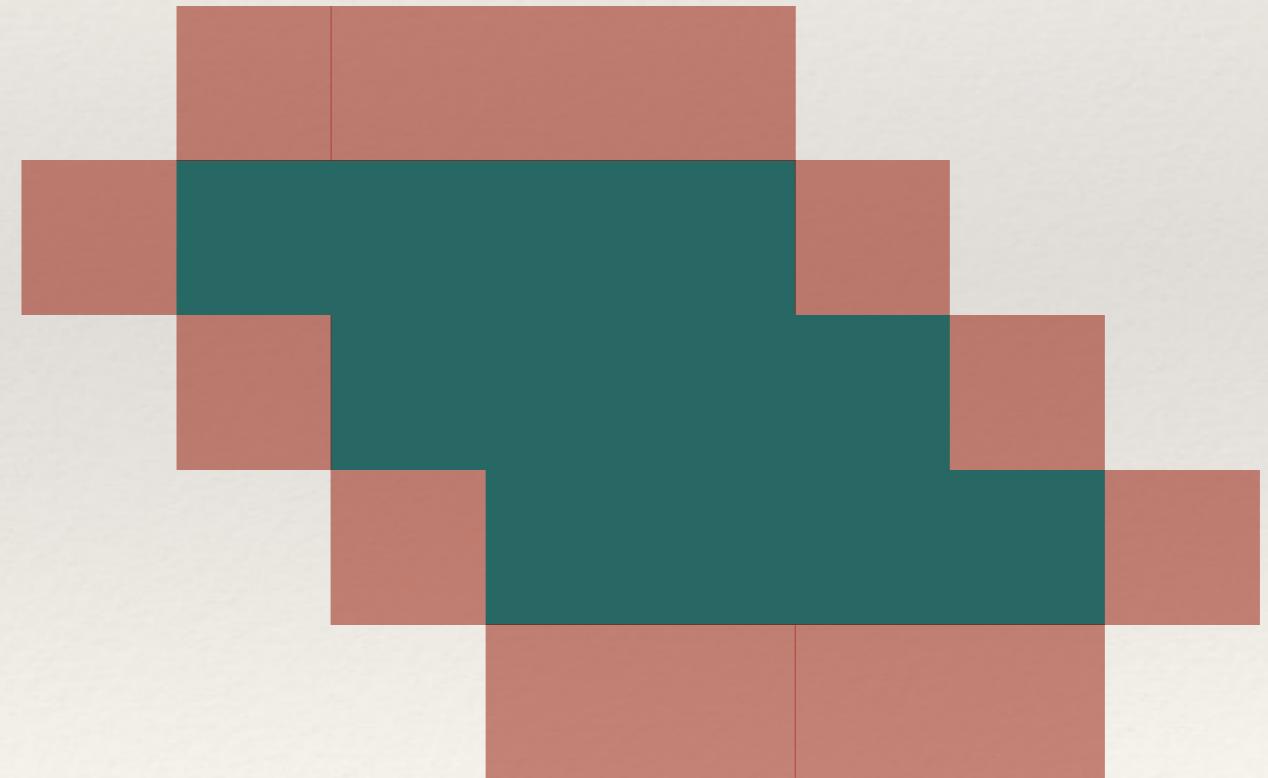
---

# Borders

---



*Inner Border*

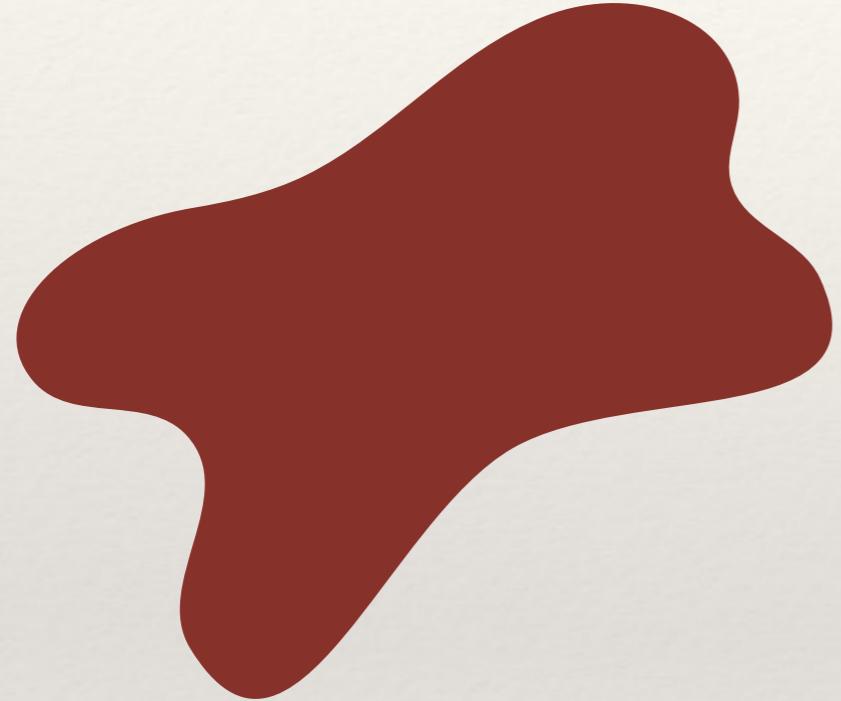


*Outer Border*

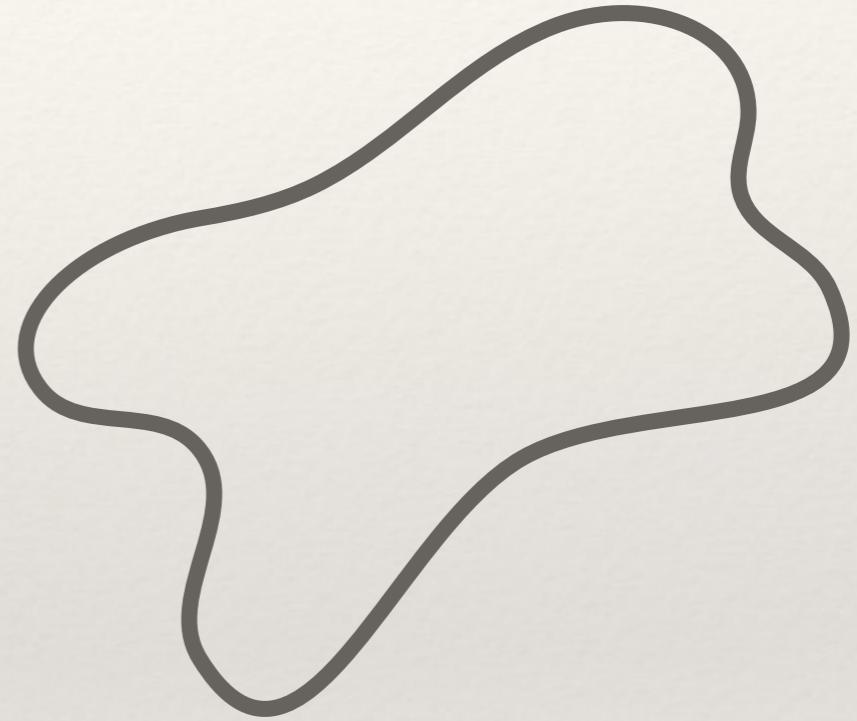
---

# Two ways to describe shape

---



Region Description

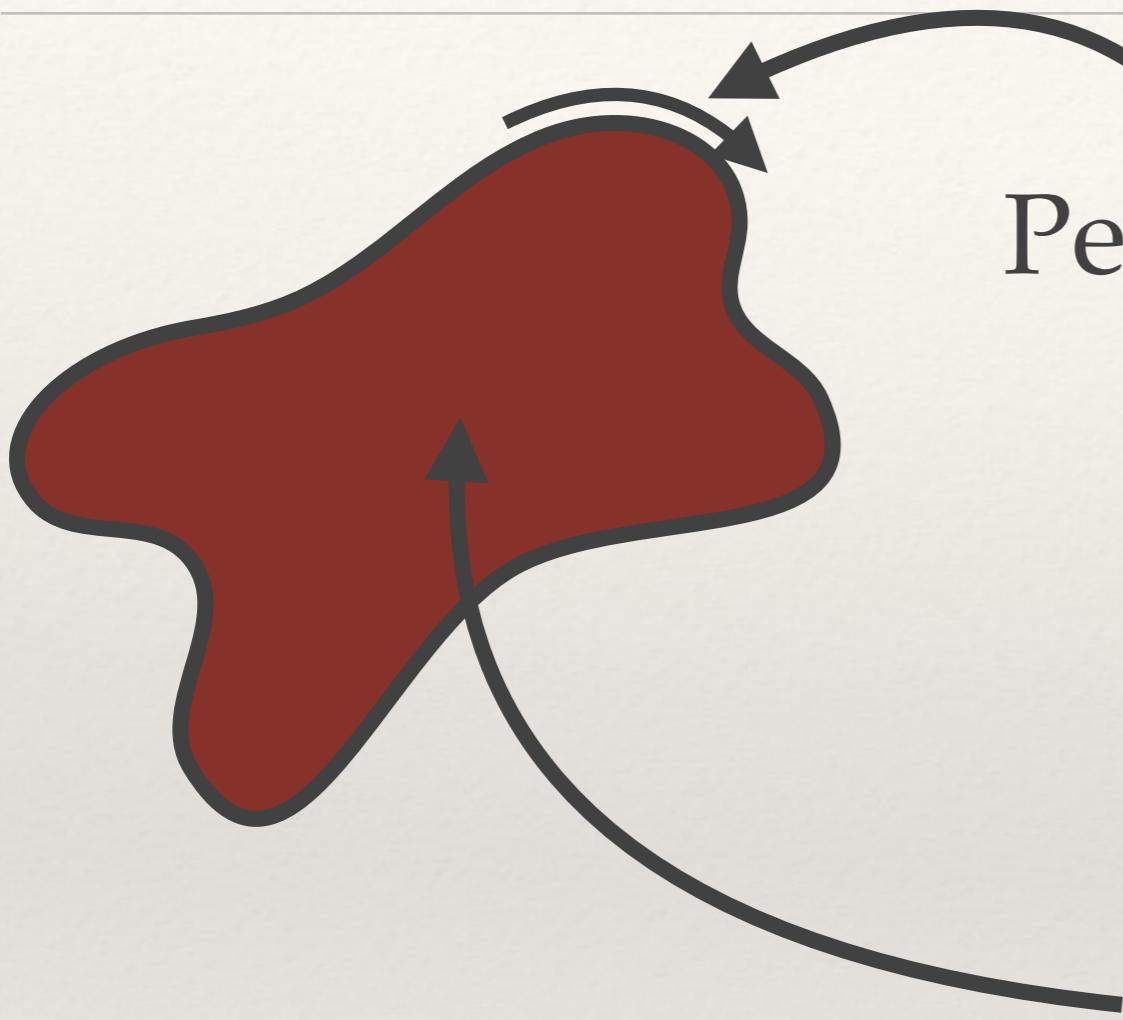


Boundary Description

# Region Description

# Simple Scalar Shape Features

# Area and Perimeter



Perimeter = length around the outside of component

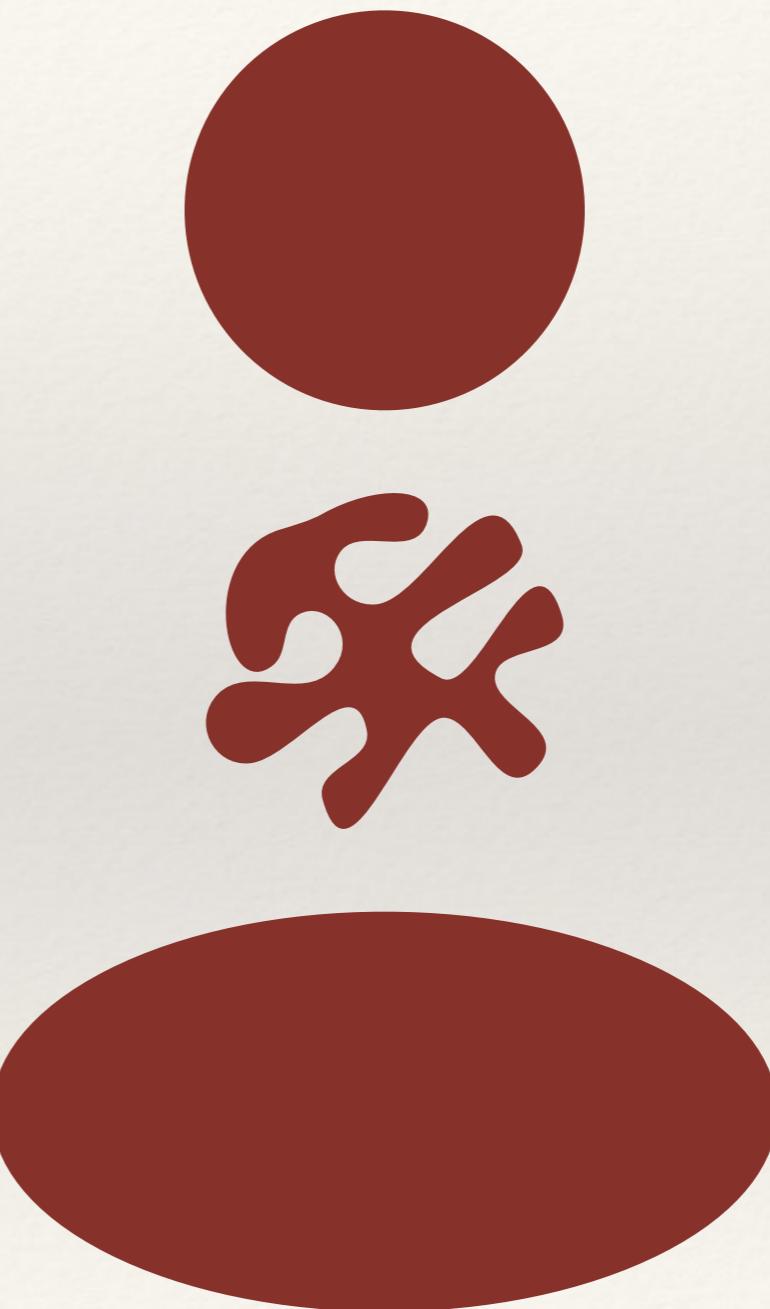
$$P(S) = \sum_j \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}$$

Area = number of pixels in component

# Compactness

- ❖ Compactness measures how tightly packed the pixels in the component are.
- ❖ It's often computed as the weighted ratio of area to perimeter squared:

$$C(s) = \frac{4\pi A(s)}{(P(s))^2}$$



$C \approx 1$

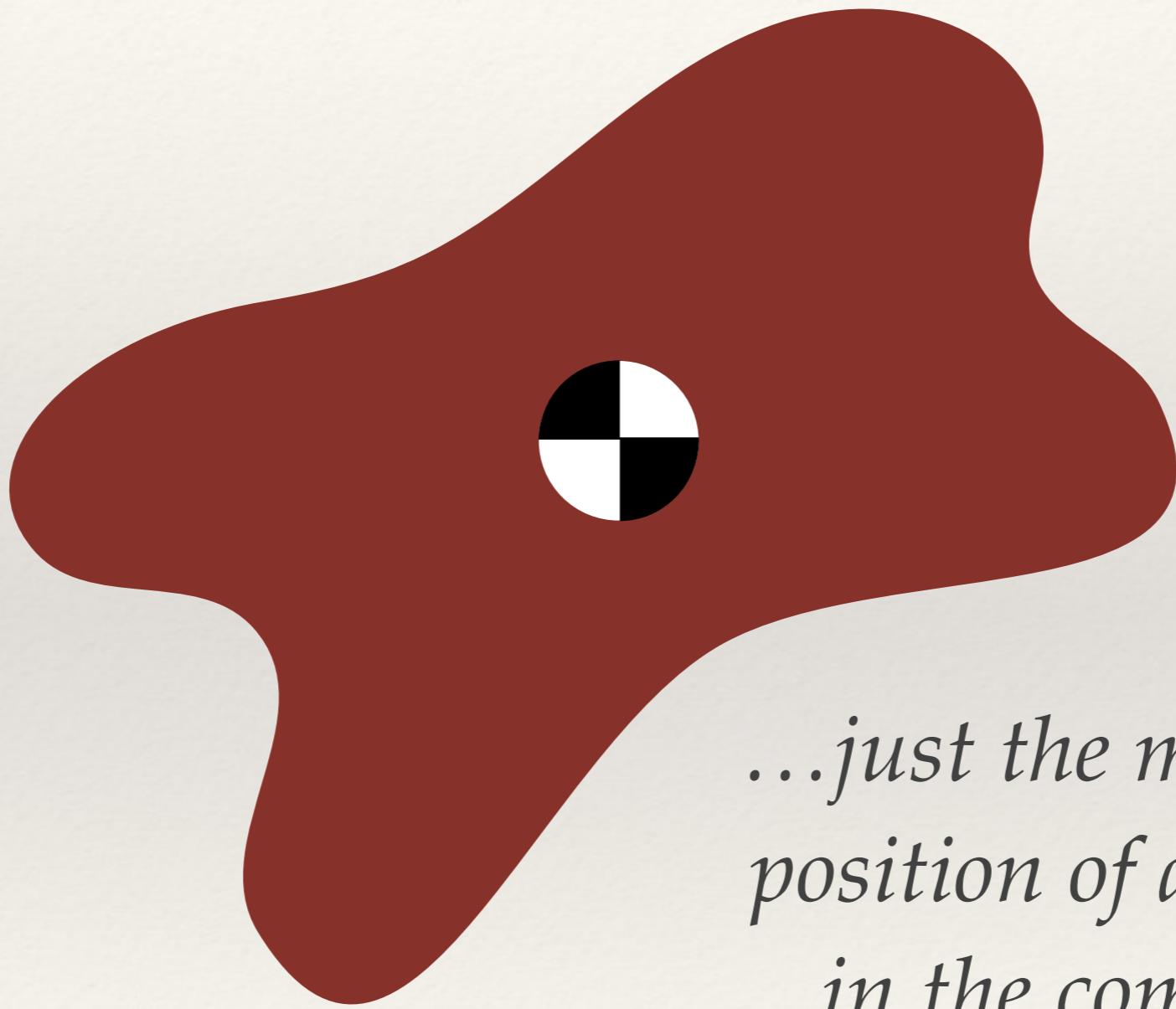
$C \approx 0.11$

$C \approx 0.79$

---

# Centre of Mass

---



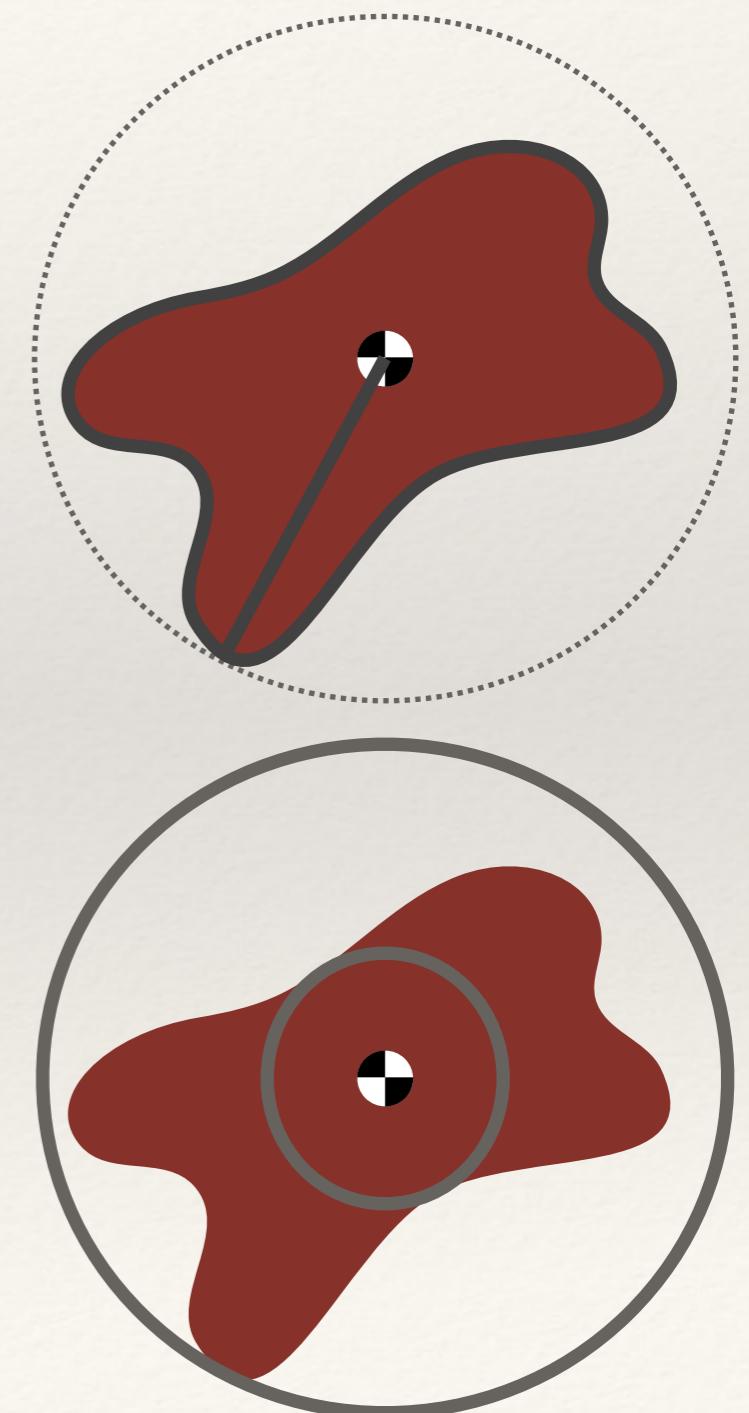
*...just the mean  $x$  and  $y$   
position of all the pixels  
in the component...*

# Irregularity/dispersion

A measure of how ‘spread-out’ the shape is

$$I(s) = \frac{\pi \max\left((x_j - \bar{x})^2 + (y_j - \bar{y})^2\right)}{A(s)}$$

$$IR(s) = \frac{\max\left(\sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2}\right)}{\min\left(\sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2}\right)}$$



*Demo: scalar region features*

# Moments

# Standard Moments

- ❖ Moments describe the distribution of pixels in a shape.
  - ❖ Moments can be computed for any grey-level image. For the purposes of describing shape, we'll just focus on moments of a connected component.
  - ❖ Standard two-dimensional Cartesian moment of an image, with order p and q is defined as:

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y) \Delta A$$

- ❖ In the case of a connected component, this simplifies to:

$$m_{pq} = \sum_i x_i^p y_i^q$$

- ❖ The zero order moment of a connected component  $m_{00}$  is just the area of the component. The centre of mass is (centroid):

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$$

---

# Central Moments

---

- ❖ Standard 2d moments can be used as shape descriptors
  - ❖ But, they're not invariant to translation, rotation and scaling
- ❖ **Central Moments** are computed about the centroid of the shape, and are thus translation invariant:

$$\mu_{pq} = \sum_i (x_i - \bar{x})^p (y_i - \bar{y})^q$$

- ❖ Note:  $\mu_{01}$  and  $\mu_{10}$  are always 0, so have no descriptive power!

# Normalised Central Moments

---

- ❖ Normalised Central Moments are both scale and translation invariant:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \text{ where } \gamma = \frac{(p+q)}{2} + 1$$

*Demo: Moments*

---

# Hu Moments

---

- ❖ Hu Moments are a set of 7 translation, scale and rotation invariant moments:

$$M1 = \eta_{20} + \eta_{02}$$

$$M2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$M3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$M4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\begin{aligned} M5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \end{aligned}$$

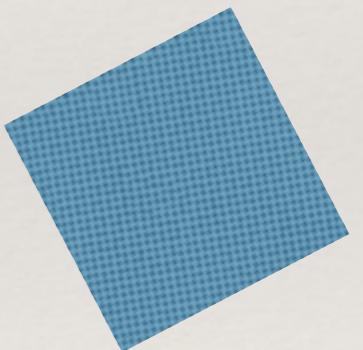
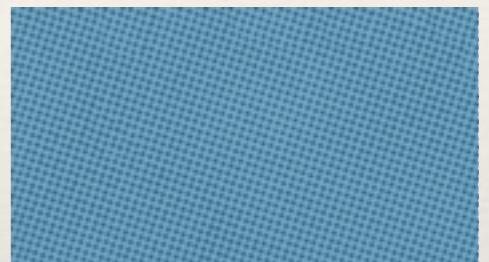
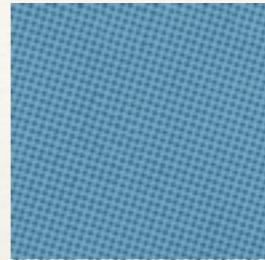
$$M6 = (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\begin{aligned} M7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) \\ &\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})(3(\eta_{12} + \eta_{30})^2 - (\eta_{21} + \eta_{03})^2) \end{aligned}$$

# *Demo: Hu Moments*

# Even more invariance

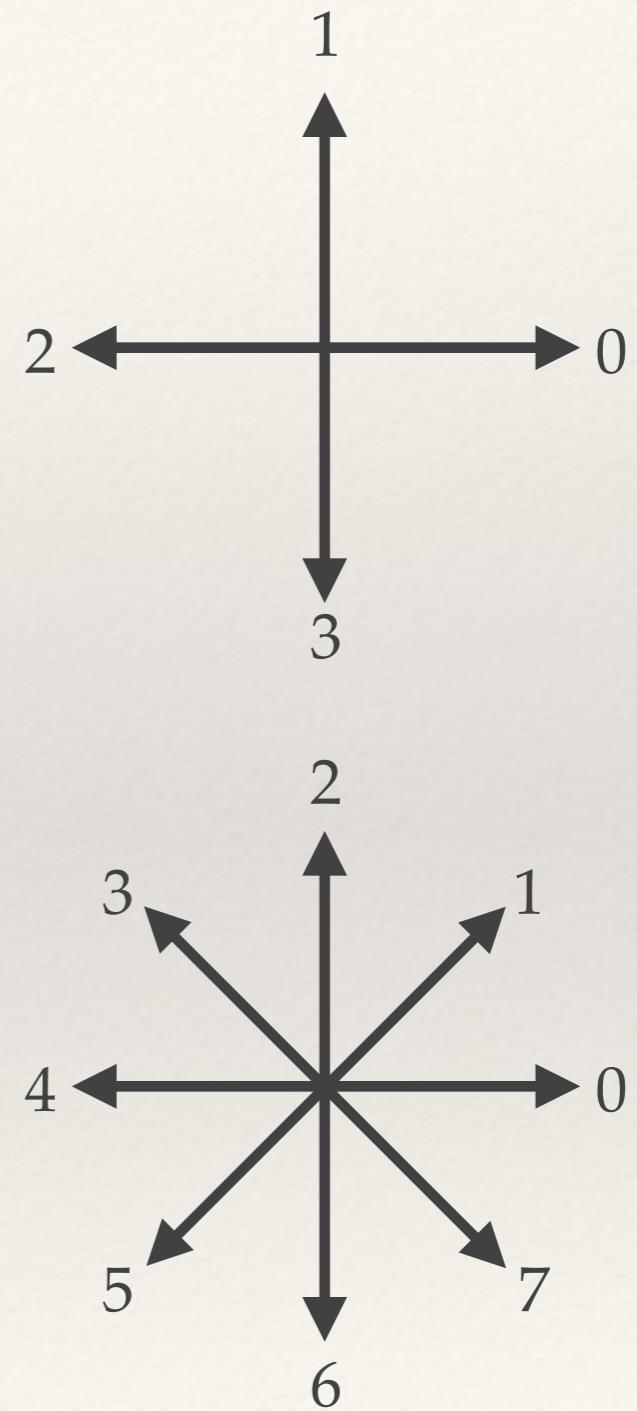
- ❖ There's another common set of rotation invariant moments called *Zernike moments*.
  - ❖ Used a lot in medical imaging for describing the shape of tumours.
- ❖ Other sets moments can be even more invariant to shape transformation
  - ❖ e.g. *Affine invariant moments*



# Boundary Description

# Chain Codes

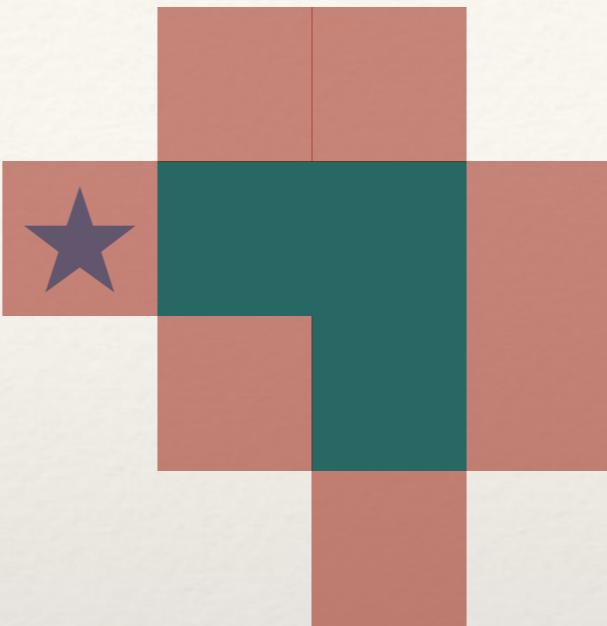
- ❖ Simple way of encoding a boundary.
  - ❖ Walk around the boundary and encode the direction you take on each step as a number
  - ❖ Then cyclically shift the code so it forms the smallest possible integer value (making it translation invariant)



---

# Chain Codes Example

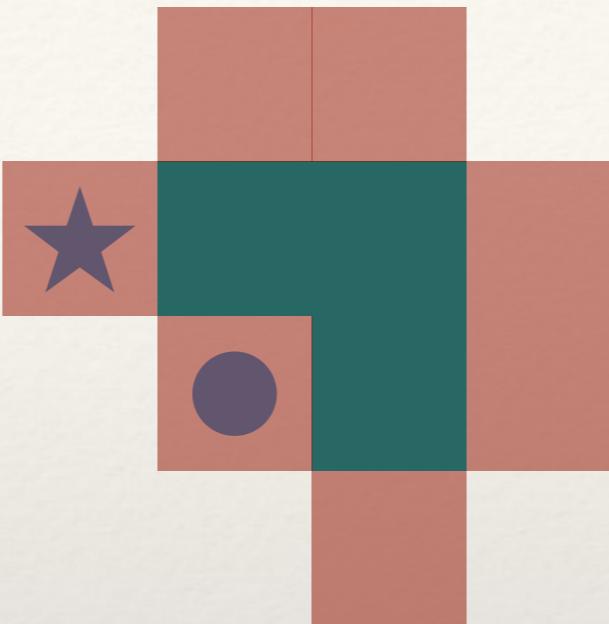
---



---

# Chain Codes Example

---

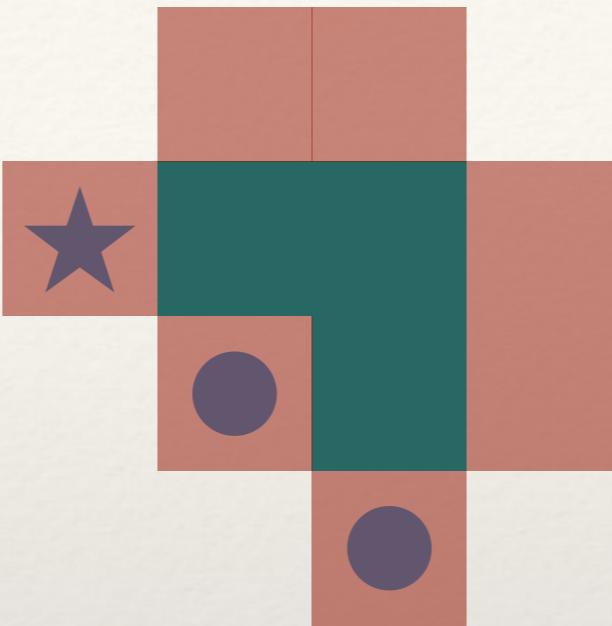


7

---

# Chain Codes Example

---

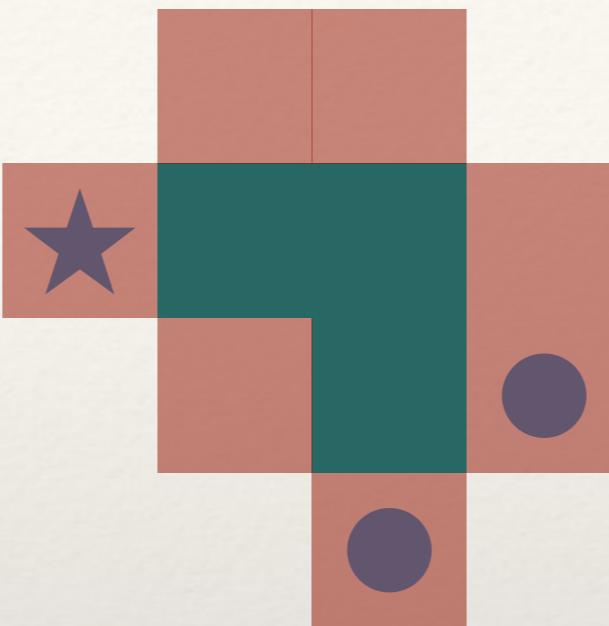


77

---

# Chain Codes Example

---

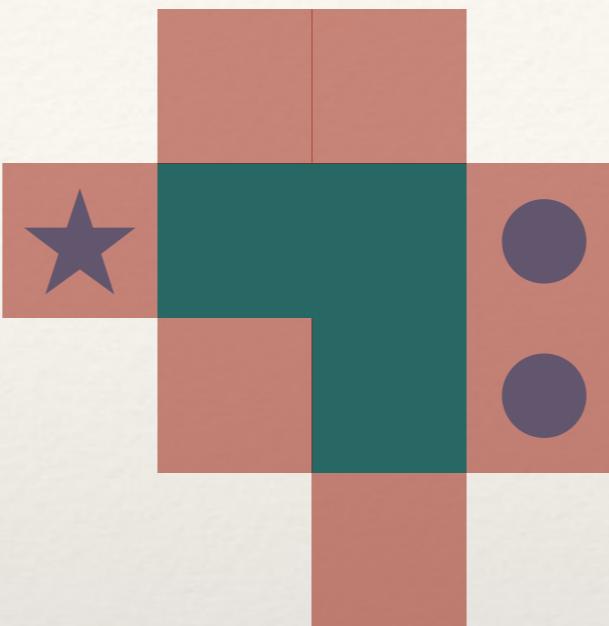


771

---

# Chain Codes Example

---

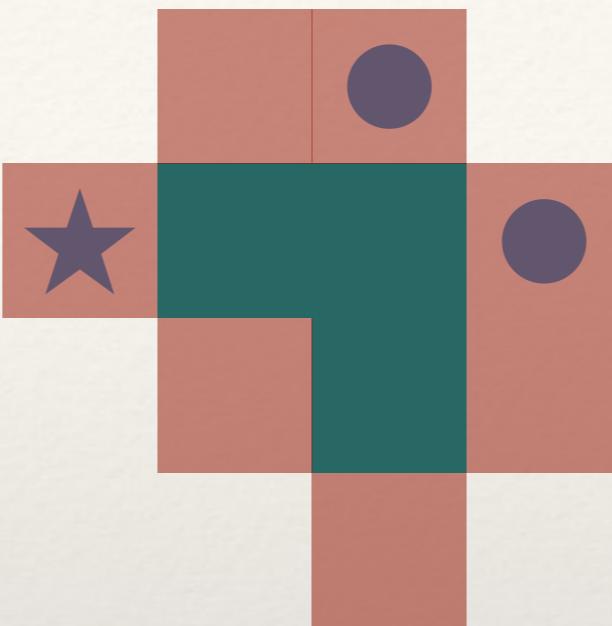


7712

---

# Chain Codes Example

---

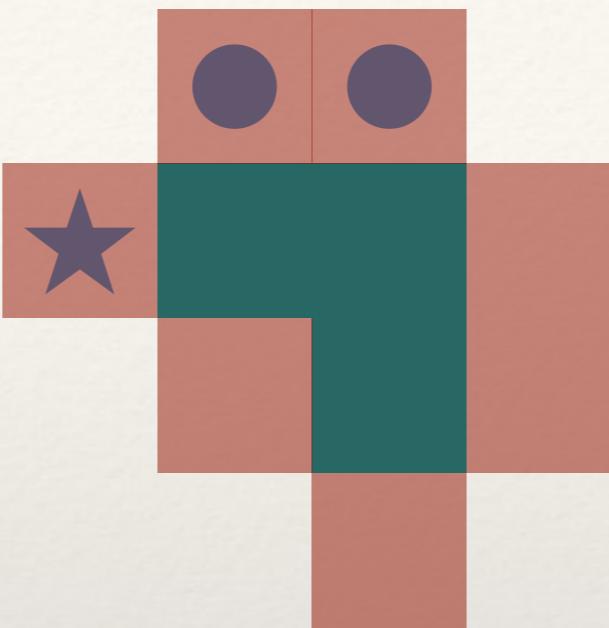


77123

---

# Chain Codes Example

---

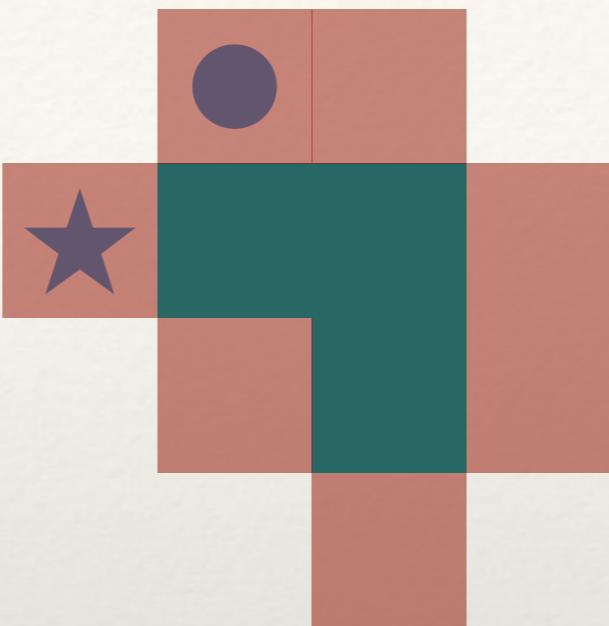


771234

---

# Chain Codes Example

---

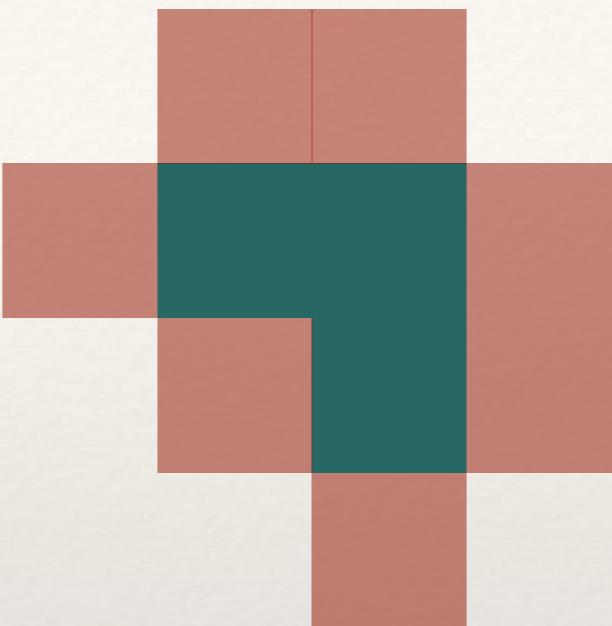


7712345

---

# Chain Codes Example

---

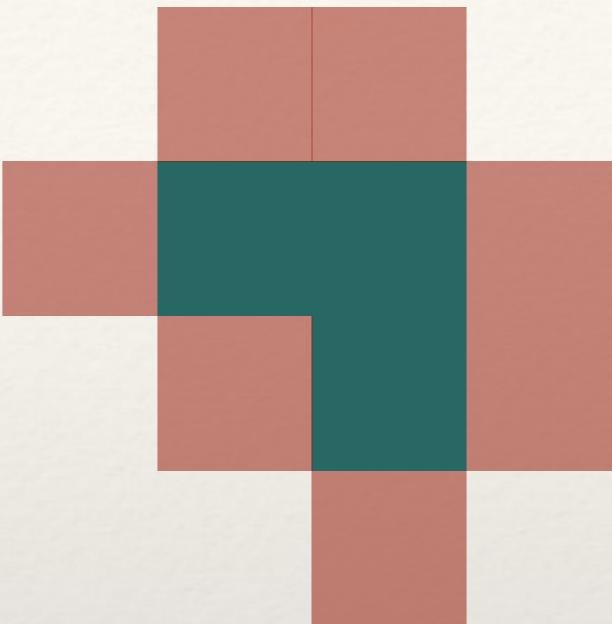


7123457

---

# Chain Codes Example

---



1234577

---

# Chain Code Invariance

---

- ❖ Can be made rotation invariant:
  - ❖ Encode the differences in direction rather than absolute values.
- ❖ Can be made scale invariant:
  - ❖ Resample the component to a fixed size
    - ❖ Doesn't work well in practise...

# Chain Code Advantages and Limitations

---

- ❖ Can be used for computing perimeter area, moments, etc.
  - ❖ Perimeter for an 8-connected chain code is  $N(\text{even numbers in code}) + \sqrt{2}N(\text{odd numbers in code})$
- ❖ Practically speaking, not so good for shape matching
  - ❖ Problems with noise, resampling effects, etc
  - ❖ Difficult to find good similarity / distance measures

---

# Fourier Descriptors

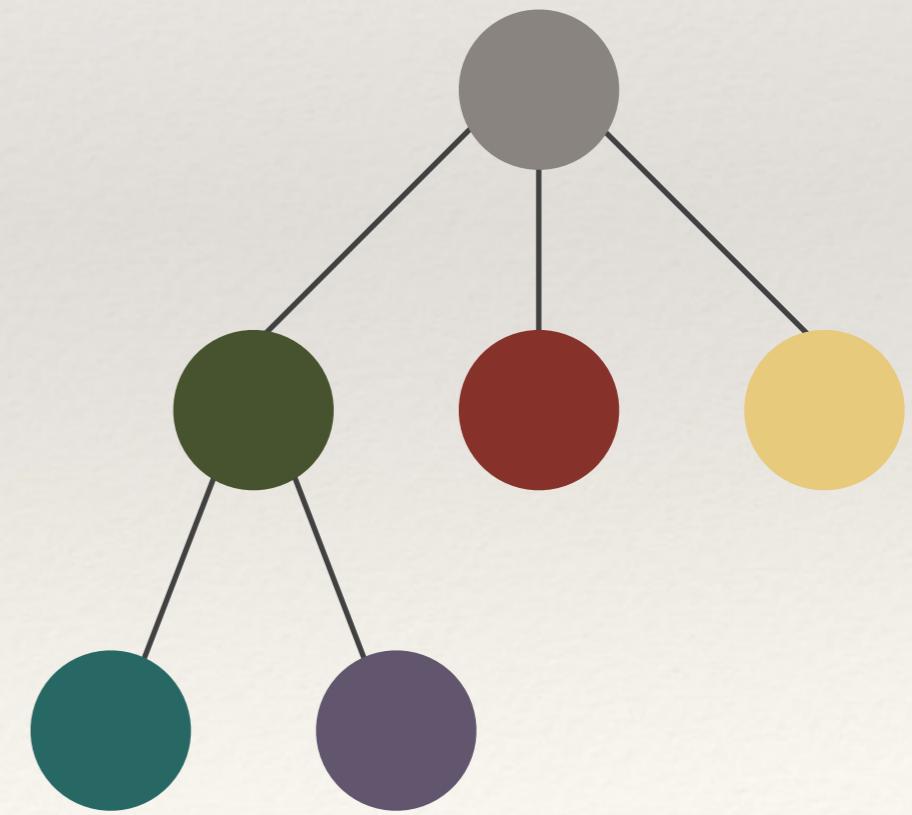
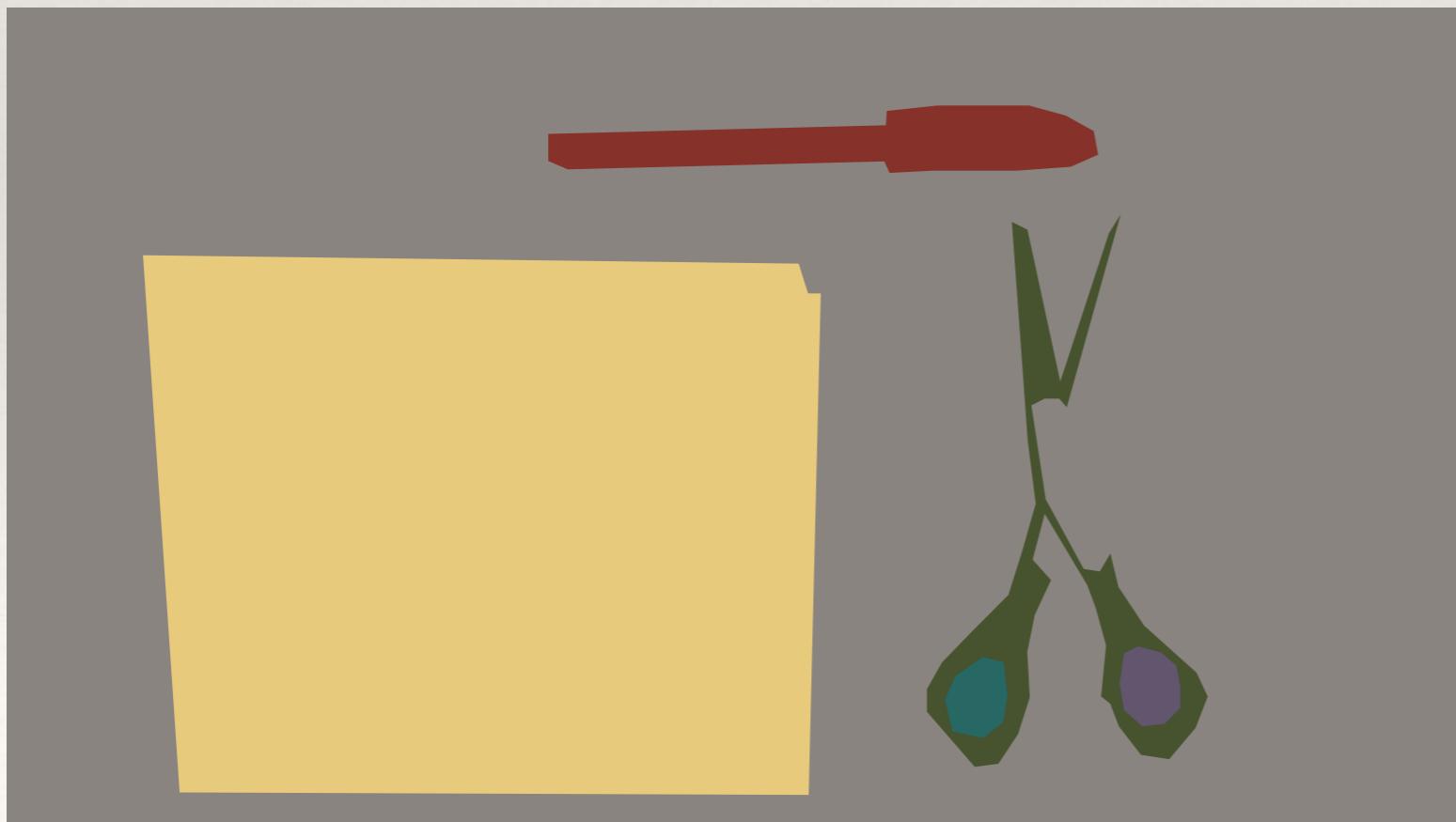
---

- ❖ The Fourier transform can be used to encode shape information by decomposing the boundary into a (small) set of frequency components.
  - ❖ There are two main steps to consider:
    - ❖ Defining a representation of a curve (the boundary)
    - ❖ Expanding the representation using Fourier theory.
  - ❖ By choosing these steps carefully it is possible to create rotation, translation and scale invariant boundary descriptions that can be used for recognition, etc.

# Describing Multiple Shapes

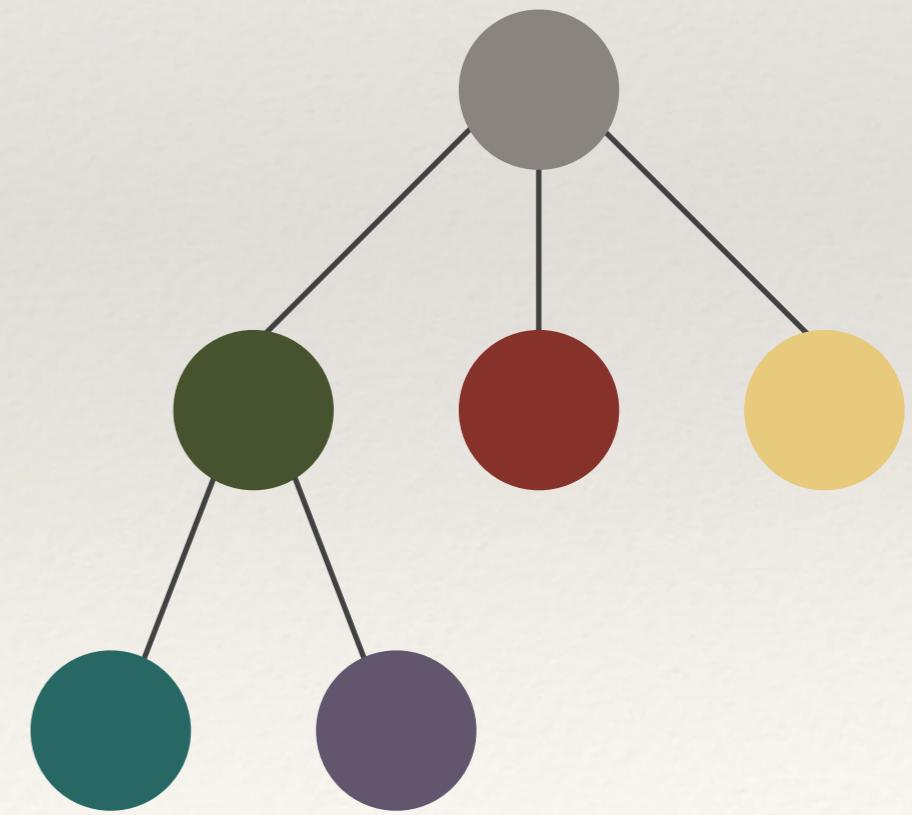
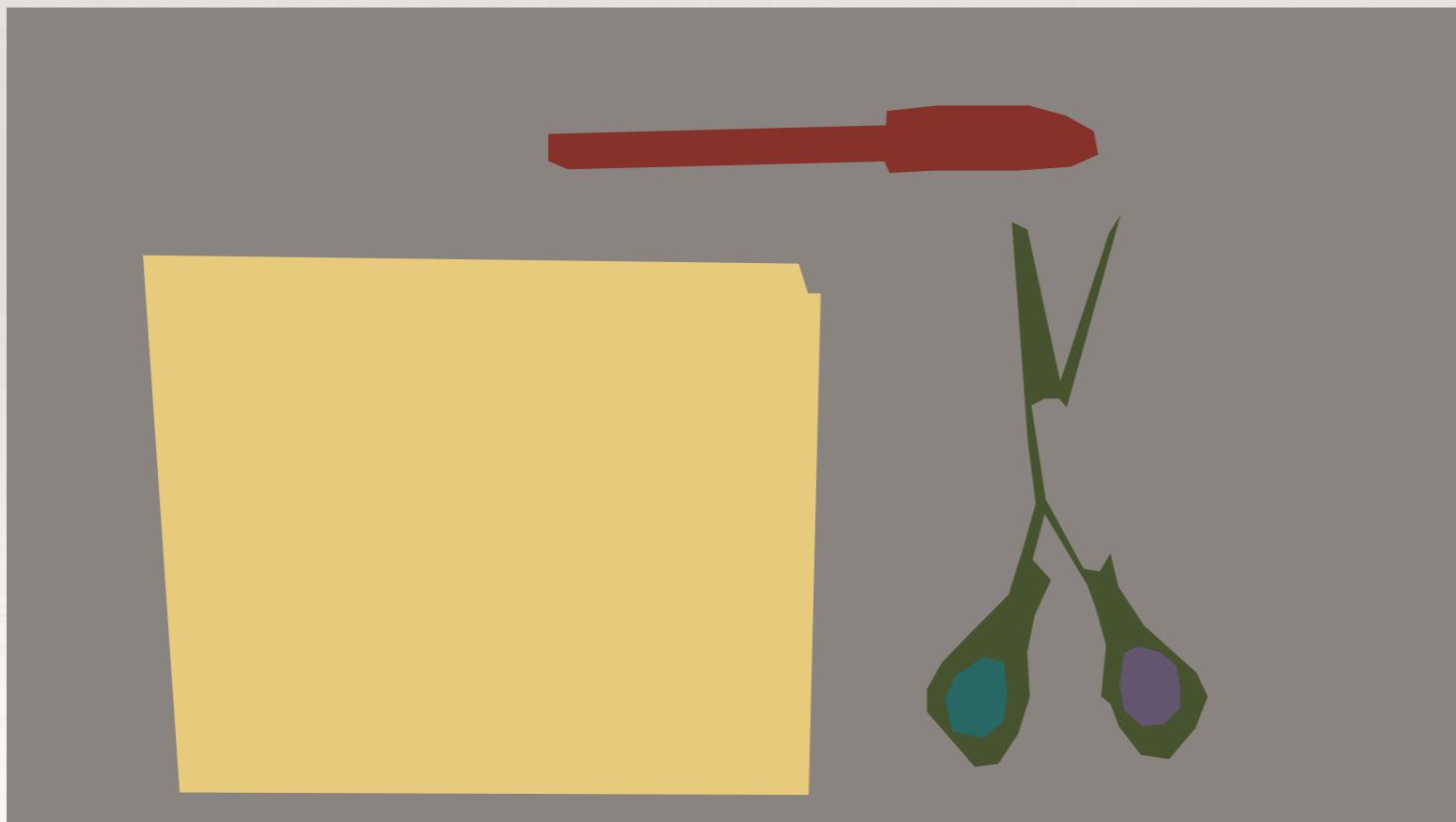
# Region Adjacency Graphs

- ❖ Build a graph from a set of connected components
  - ❖ Each node corresponds to a component
  - ❖ Nodes connected if they share a border



# Region Adjacency Graphs

- ❖ Can easily detect patterns in the graph
  - ❖ e.g. “a node with one child with four children”
- ❖ Invariant to non-linear distortions, but not to *occlusion*



*Demo: RAGs*

# Flexible Shape Modelling and Fitting

# Point Distribution Models

- ❖ Idea: learn a low dimensional parametric model of how the shape of an object can vary.
  - ❖ Shape represented by a fixed number of 2d points at distinguishable locations on the object.
    - ❖ e.g. for faces, points at the corners of the mouth and eyes, tip of the nose, etc
    - ❖ Need a training set of images with the same points marked...

# *Demo: Shape Data*

---

# Procrustes Analysis

---

- ❖ Need to align (rotate and scale) all training points.
  - ❖ All point-sets must have the same size and be about the origin.
  - ❖ *Generalised procrustes analysis* is an iterative technique for achieving this:
    1. arbitrarily choose a reference shape (typically by selecting it among the available instances and centring it about the origin)
    2. superimpose all instances to current reference shape (by computing the optimal translation, scaling and rotation for each)
    3. compute the mean shape of the current set of superimposed shapes
    4. if the Procrustes distance (the Euclidean distance between  $[x_1, y_1, x_2, y_2, \dots]$  and  $[x'_1, y'_1, x'_2, y'_2, \dots]$ ) between mean and reference shape is above a threshold, set reference to mean shape and continue to step 2.

# *Demo: Procrustes Analysis*

# Form the data points into a shape matrix

**S =**

x	y	x	y	...	...	x	y
...	...	...	...	...	...	...	...
x	y	x	y	...	...	x	y

Each pair of columns contains  
(x, y) coordinates of aligned  
corresponding points across the  
images

Each row corresponds to  
an image (it's essentially  
a featurevector for the  
shape in that image!)

---

# Apply PCA to learn a low-dimensional basis

---

$$\mathbf{S}^T \mathbf{S} = \mathbf{Q} \Lambda \mathbf{Q}^T$$

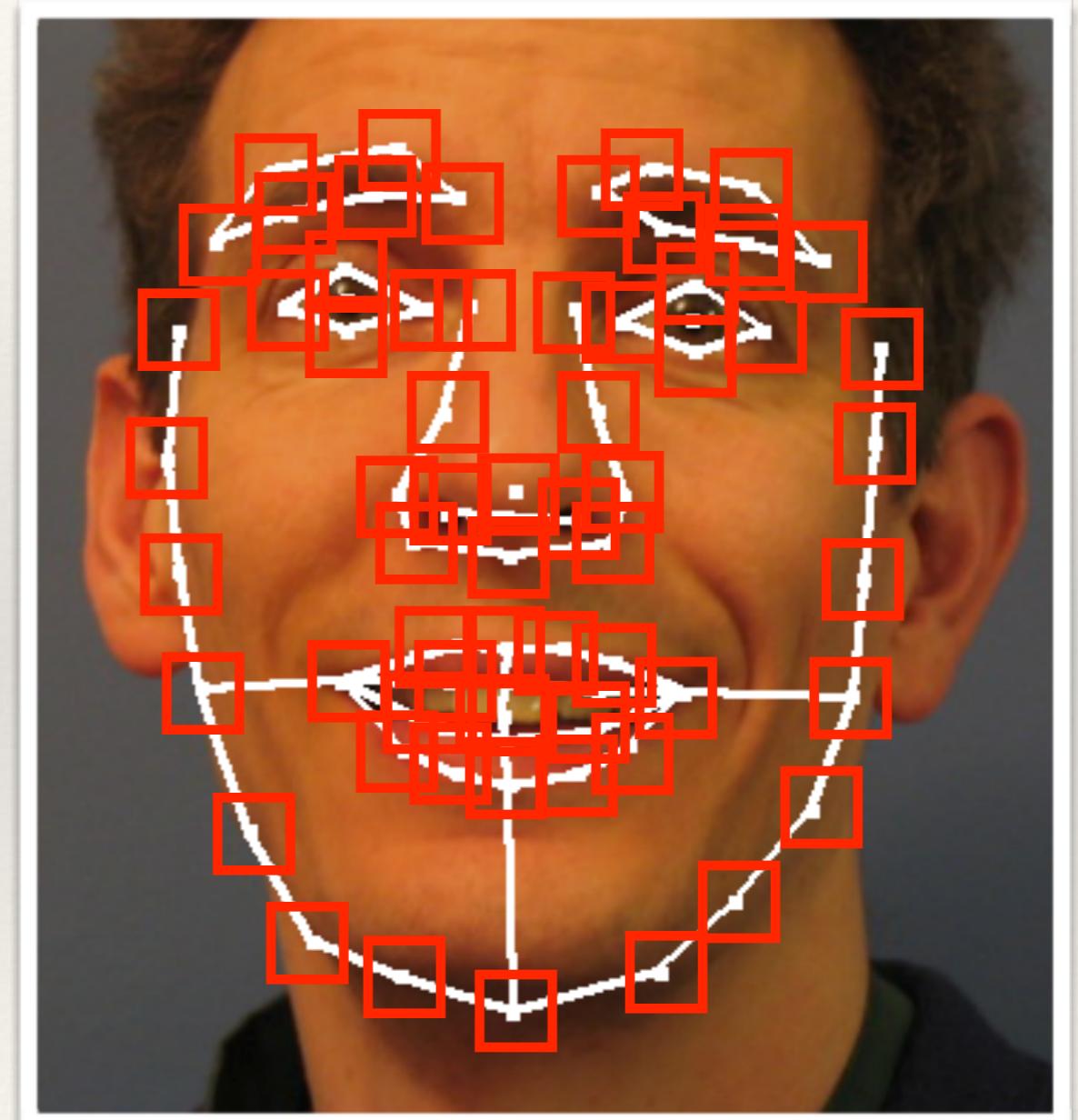
$\mathbf{Q}$  can be used to *generate* shapes from a low-dimensional weight vector,  $w$ :

$$v = w \mathbf{Q}^T$$

*Demo: PDM*

# Active Shape Models and Constrained Local Models

- ❖ ASMs/CLMs extend a PDM by also learning local appearance around each point
  - ❖ Typically just as an image *template*
- ❖ Using a constrained optimisation algorithm, the shape can be optimally fitted to an image
  - ❖ Constraints:
    - ❖ *Plausible shape*
    - ❖ *Good template matching*



*Demo: CLM fitting*

---

# Active Appearance Models

---

- ❖ AAMs go even further and model the global appearance of the shape
  - ❖ ...using Eigenfaces!
- ❖ Optimiser fits the model to new images by searching for both plausible shape and plausible global appearance

---

# Summary

---

- ❖ Many different ways to describe shape of connected components
  - ❖ Choice really depends on required invariance
- ❖ Multiple shapes can efficiently be represented by a RAG
  - ❖ Very robust!
- ❖ Point distribution models apply PCA to x-y coordinate pairs across multiple images to produce a low-dimensional parametric model
  - ❖ ASMs/CLMs also model local appearance, and can use this to optimise the fit of the model parameters to match an image