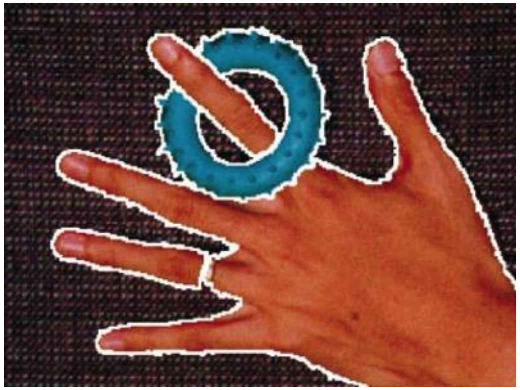
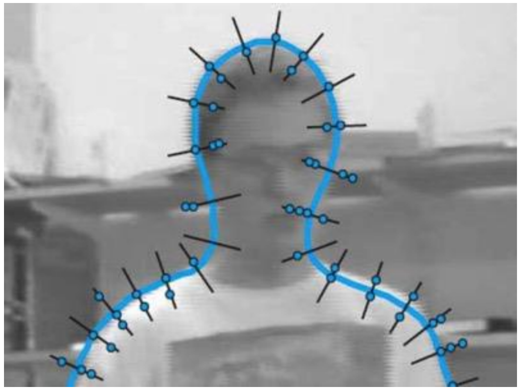
  

COMP9517: Computer Vision

Image Segmentation

Part 2

Outline

- Basic segmentation methods
  - Thresholding
  - K-means clustering
  - Feature extraction and classification

- More sophisticated segmentation methods
  - Region splitting and merging
  - Watershed segmentation
  - Maximally stable extremal regions
  - Mean-shift algorithm
  - Superpixel segmentation
  - Conditional random field
  - **Active contour segmentation**
  - **Level-set segmentation**

Outline

- **Region Representation**
  - **Labelled images (the most commonly used)**
  - **Overlays**
  - **Boundary coding**
  - **Quad trees**
  - **Property tables**

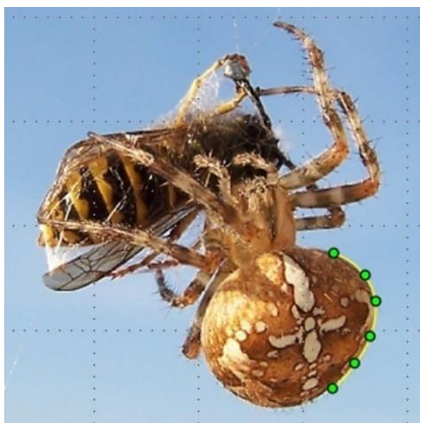
- **Evaluating segmentation methods**
  - **Quantitative evaluation metrics**
  - **Receiver operating characteristic**

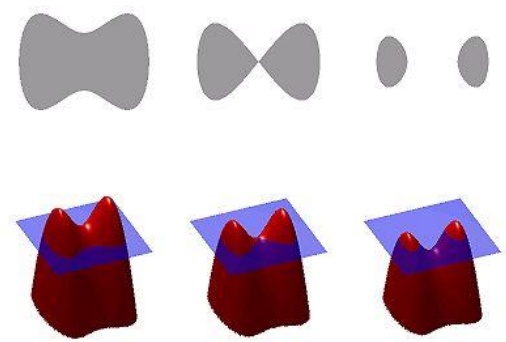
Active Contour Segmentation

- A contour based approach to object segmentation
  - Aims to locate object boundaries in images by curve fitting
  - Represents the curve by a set of control points and interpolation
  - Iteratively moves the control points to fit the curve to the object
  - Uses **image**, **smoothness** and **user-guidance forces** along curve

Active Contours

- Examples of implementations:
  - Snakes
  - Level sets





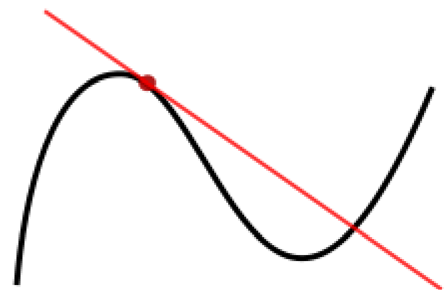
- Active contours can also be used in a wide variety of object-tracking applications
- Based on global optimisation

Global Optimisation

- formulate the goals of the desired transformation (e.g. segmentation) using some optimisation criterion, and then infer the solution that best meets this criterion
- Problem: finding a smooth surface that passes through a set of measured data points
  - ill-posed: many possible surfaces can fit this data
  - ill-conditioned: small changes in the input can sometimes lead to large changes in the fit
  - inverse problems: recover the unknown function form which data point were sampled
- therefore, regularisation is needed
  - to fit models to data that badly underconstrained the solution space

Global Optimisation

- in order to quantify what it means to find a smooth solution, we can define a norm on the solution space
- the **derivative** is a measure of how a function changes as its input changes.



  - a derivative can be thought of as how much one quantity is changing in response to changes in some other quantity
  - for example, the derivative of the position of a moving object with respect to time is the object's instaneous velocity

$$m = \frac{\Delta f(x)}{\Delta x} = \frac{f(x+h) - f(x)}{(x+h) - (x)} = \frac{f(x+h) - f(x)}{h}.$$

Global Optimisation

- For one dimensional function f(x), to find a smooth solution, we can define a **norm** on the solution space:
  - integrate the squared first derivative of the function

$$\varepsilon_1 = \int f_x^2(x)dx$$

  - integrate the squared second derivative

$$\varepsilon_2 = \int f_{xx}^2(x)dx$$

- For two dimensions(e.g. for images), the corresponding smoothness functions are（partial derivative)

$$\varepsilon_1 = \int f_x^2(x,y) + f_y^2(x,y)dxdy = \int \left\|\nabla f(x,y)\right\|^2 dxdy$$

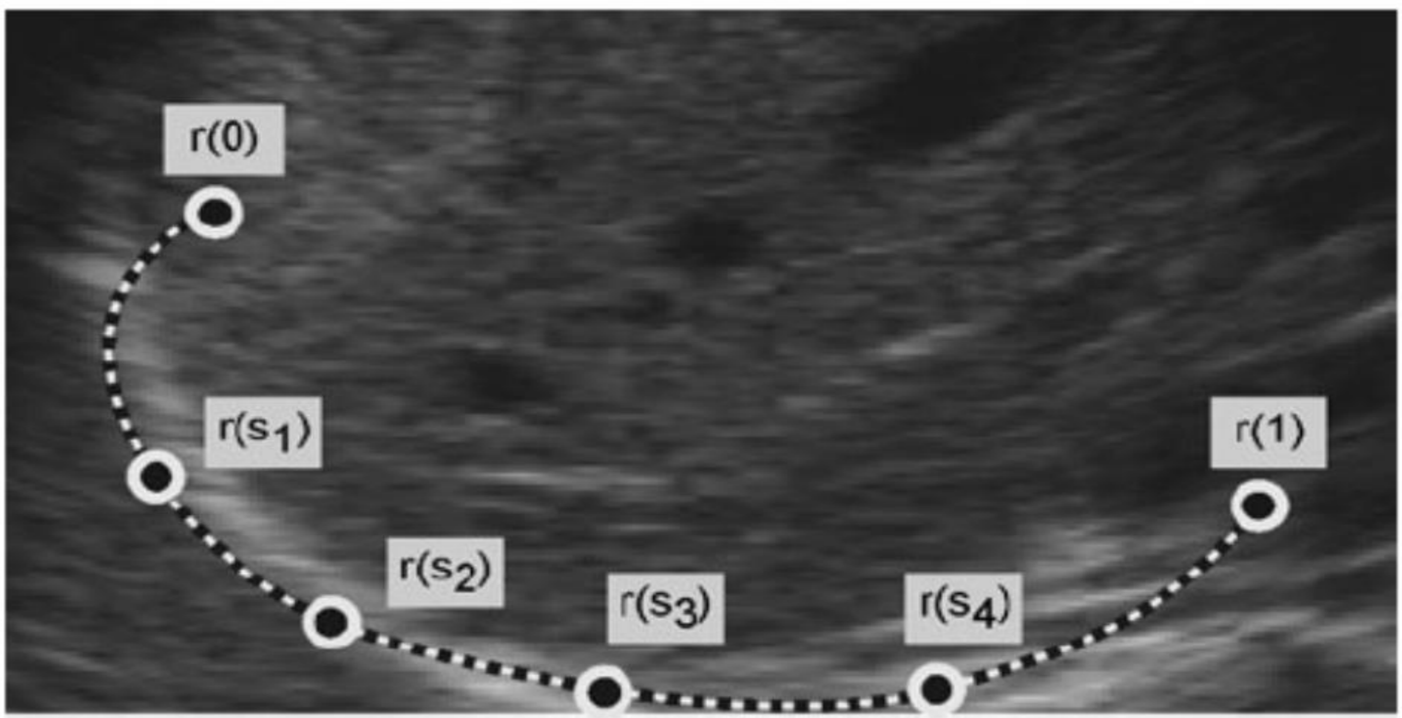
$$\varepsilon_2 = \int f_{xx}^2(x,y) + 2f_{xy}^2(x,y) + f_{yy}^2(x,y)dxdy$$

Global Optimization

- We also require a data term (or data penalty)
- For scattered data interpolation, the data term measures distance between function $f(x,y)$ and a set of data points $d_i = d(x_i, y_i)$
- By adding the norm and the data term, we get a global energy term that can be minimized

Active Contour Segmentation

- A well-known implementation is called the **snakes algorithm**
  - Smoothly follows high intensity gradients at the object boundary
  - Bridges areas of noise or missing gradients using smooth interpolation



Snakes

- Snakes are a two-dimensional generalisation of the 1-D energy minimising splines

$$\varepsilon_{\text{int}} = \int \alpha(s)\left\| f_s(s) \right\|^2 + \beta(s)\left\| f_{ss}(s) \right\|^2 ds$$

- discretised version of the energy function

$$E_{\text{int}} = \int \alpha(s)\left\| f(i+1) - f(i) \right\|^2 / h^2 + \beta(s)\left\| f(i+1) - 2f(i) + f(i-1) \right\|^2 / h^4$$

- additional external image-based and constraint-based potentials

$$\varepsilon_{image} = \omega_{line}\varepsilon_{line} + \omega_{edge}\varepsilon_{edge} + \omega_{term}\varepsilon_{term}$$

Snakes

- In practice, only the edge term is used
  - directly proportional to the image gradients

$$E_{edge} = \sum_i - \left\| \nabla I(f(i)) \right\|^2$$

  - a smoothed version of the image Laplacian

$$E_{edge} = \sum_i - \left\| (G_\delta \circ \nabla^2 I)(f(i)) \right\|^2$$

  - a distance map to the edges
- user-placed constraints

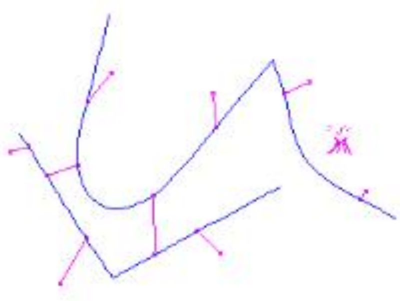
$$E_{spring} = k_i \left\| f(i) - d(i) \right\|^2$$

Snakes

- Put them together

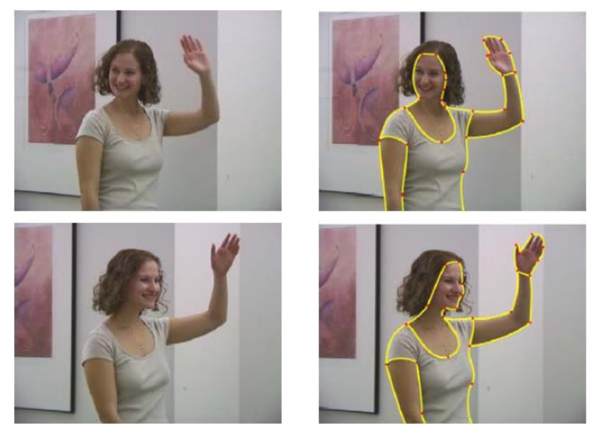
$$E = E_{\text{int}} + E_{image} + E_{spring}$$

- Let E to be zero, the energy E can be iteratively minimised by moving the curve



(a)

(b)

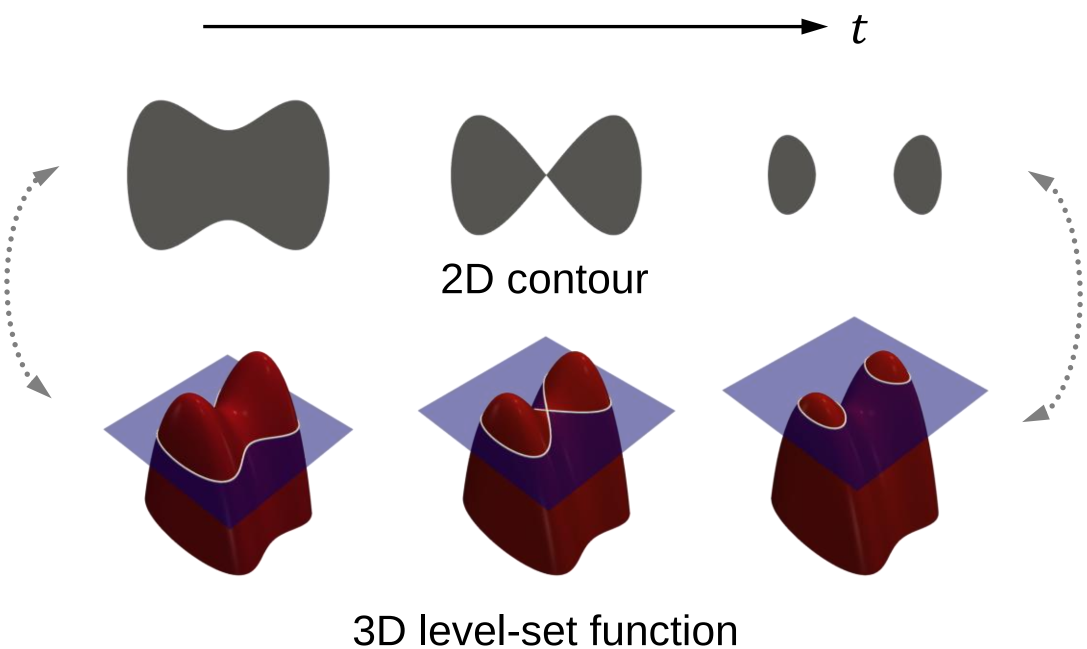


Active Contour Segmentation

- Active contours / snakes are parametric models
  - Explicit representations of the object boundaries
  - Typically requires manual interaction to initialize the curve
  - It is challenging to change the topology of the curve as it evolves
  - Curve reparameterization may be required for big shape changes
- Level-set methods have become more popular alternatives
  - Implicit representations of the object boundaries
  - Boundaries defined by the zero-set of a higher dimensional function
  - Level-set function evolves to make the zero-set fit and track objects
  - Easily accommodates topological changes in object shape
  - Computationally more demanding than active contours

Level Set Segmentation

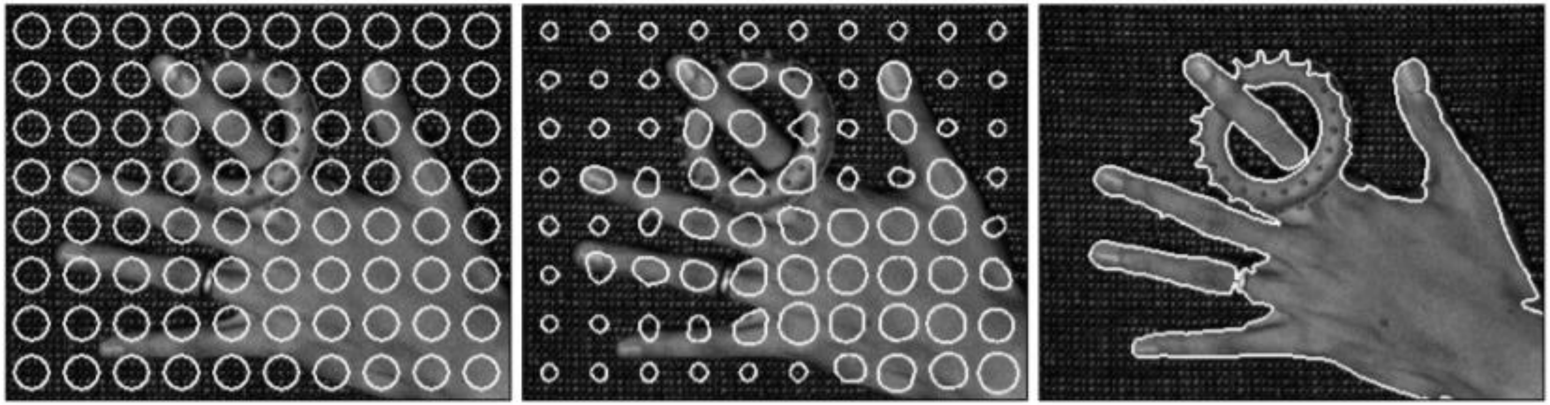
- Contour evolution over time (iterative)

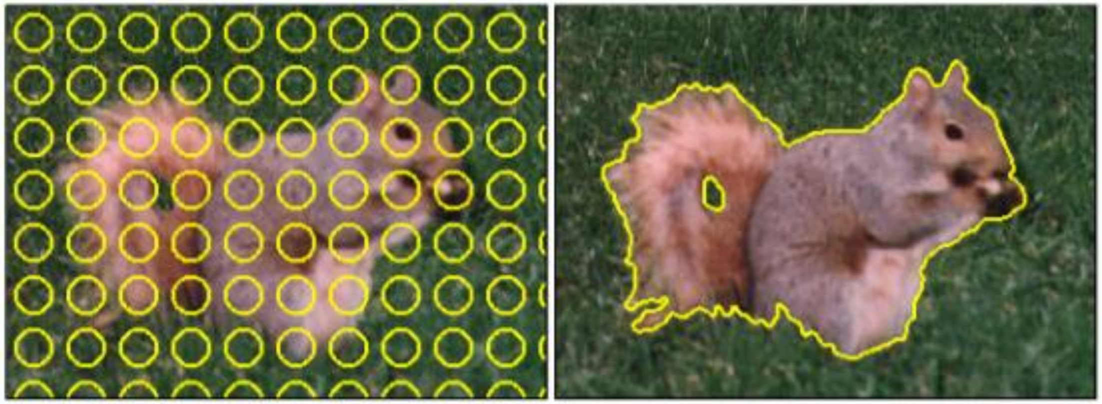


https://en.wikipedia.org/wiki/Level-set_method

Level Set Segmentation

- Examples of level-set based segmentation





 https://doi.org/10.1007/s11263-006-8711-1

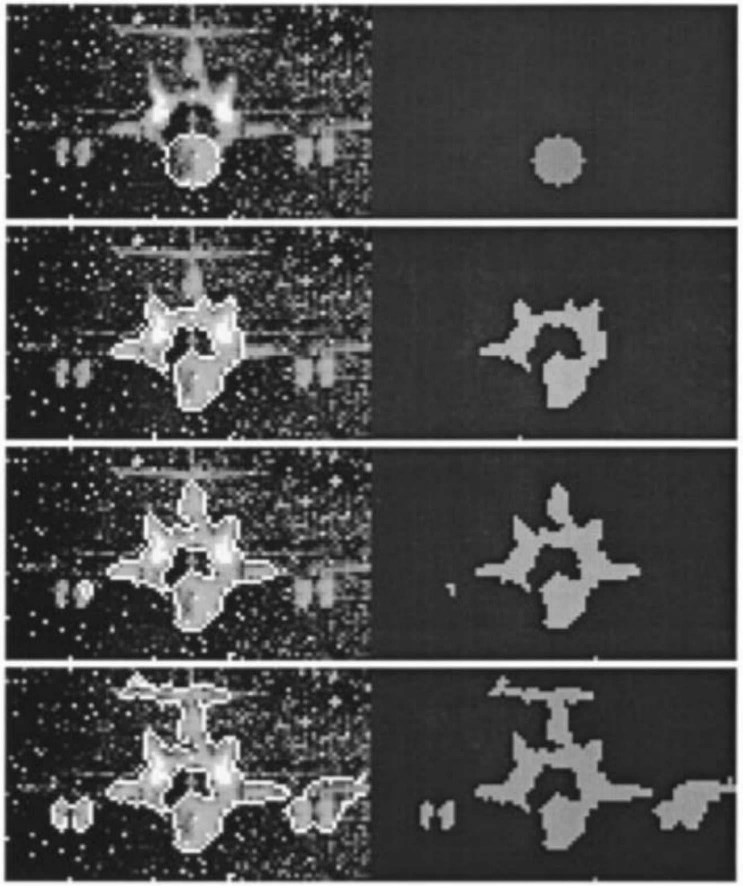
Level Set Segmentation

- A well-known implementation is the Chan-Vese model

Minimize the energy functional

$$E(C) = \mu \cdot \text{length}(C) + \nu \cdot \text{area}(C)$$
$$+ \lambda_1 \int_{\text{Inside}(C)} |u_0 - c_1|^2 \, dxdy$$
$$+ \lambda_2 \int_{\text{Outside}(C)} |u_0 - c_2|^2 \, dxdy$$

https://doi.org/10.1109/83.902291

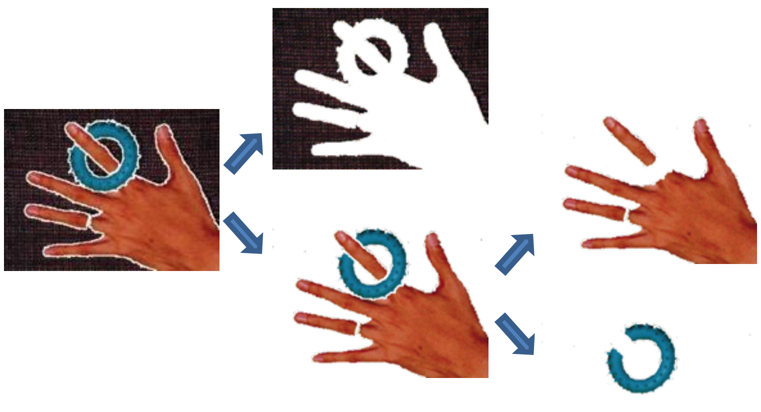


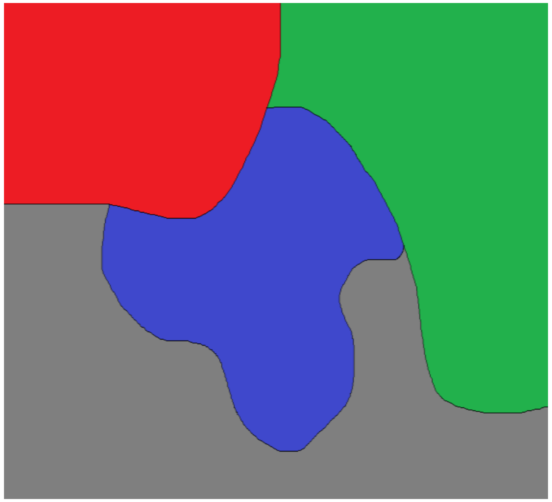
Region Representation

- The outputs from segmentation methods need to be represented and stored
- Methods to represent regions:
  - Labelled images (the most commonly used)
  - Overlays
  - Boundary coding
  - Quad trees
  - Property tables

Region Representation

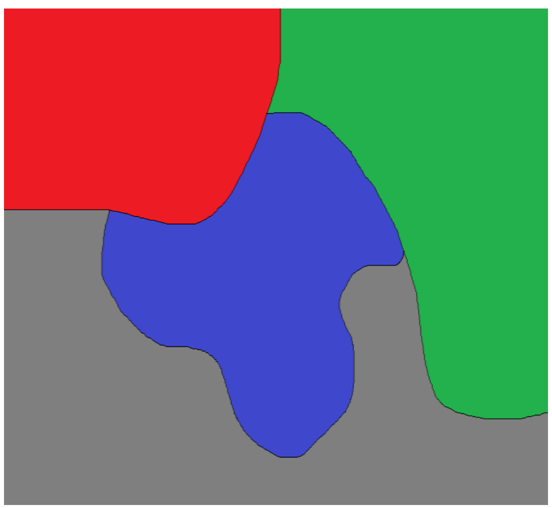
- Labelled images
  - assign each detected region a unique identifier (an integer)





Region Representation

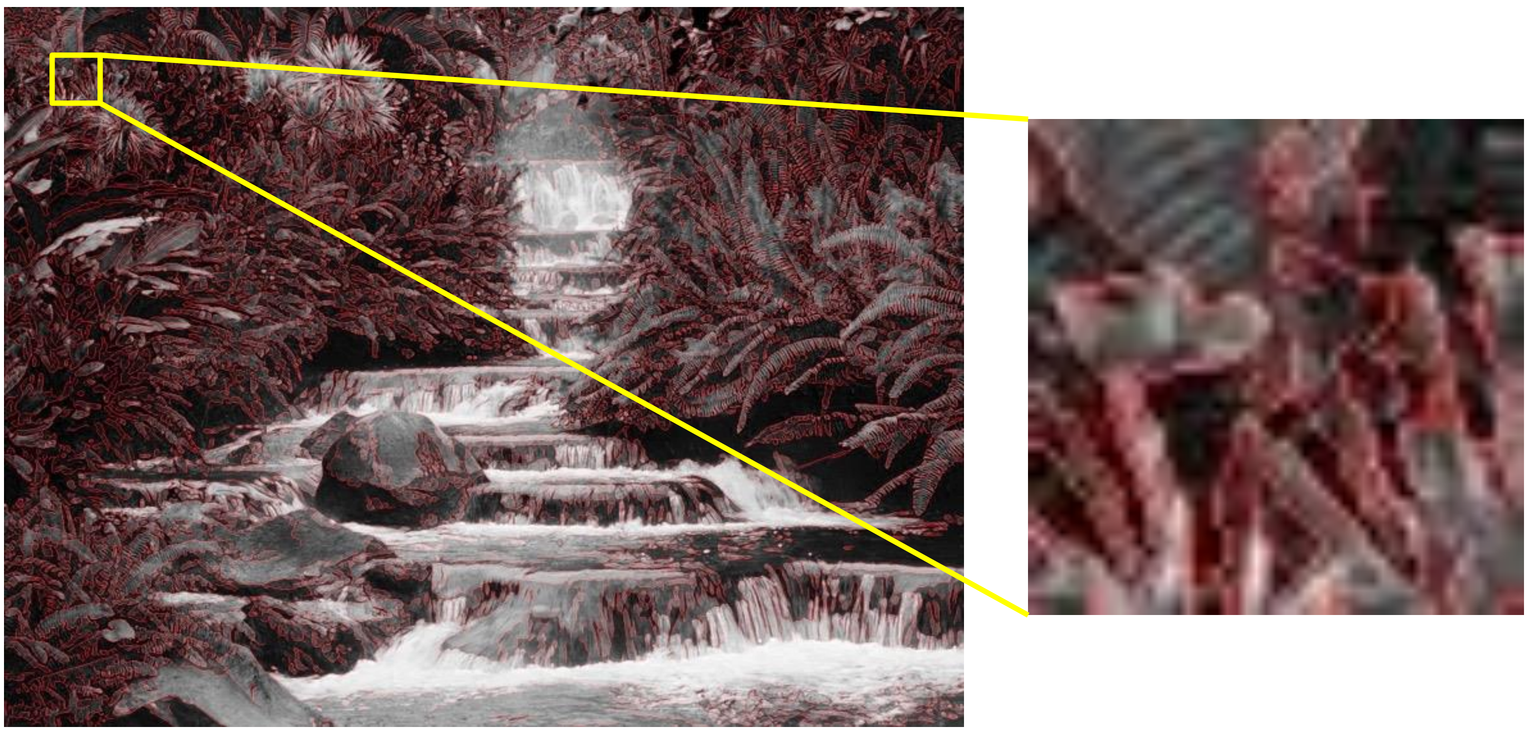
- Labelled images
  - create an image where all the pixels of a region will have its unique identifier as their pixel value



0	0	0	2	2
0	0	1	1	2
3	1	1	1	2
3	3	1	3	2
3	3	3	3	3

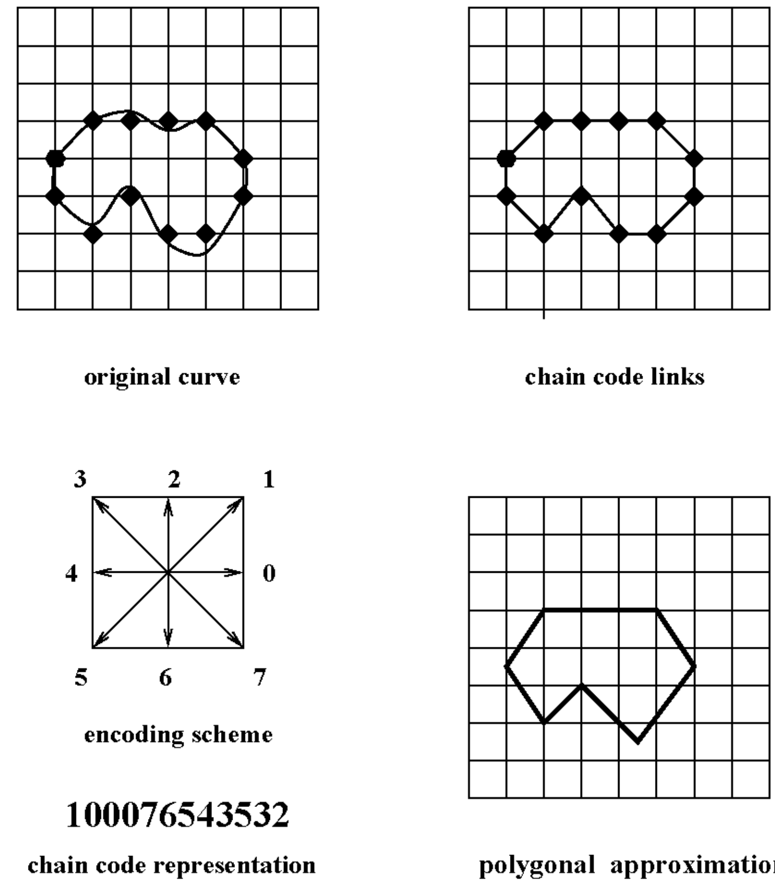
Region Representation

- Overlays
  - overlaying some colour or colours on top of the original image



Region Representation

- Boundary Coding
  - represents regions by their boundaries in a data structure instead of an image
    - a linear list of the border pixels
    - chain code
    - polygon approximation

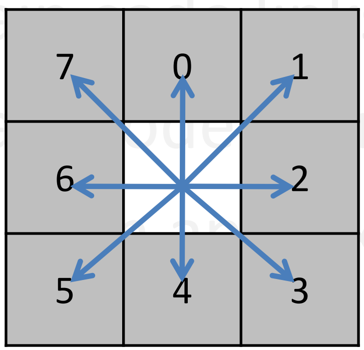


Region Representation

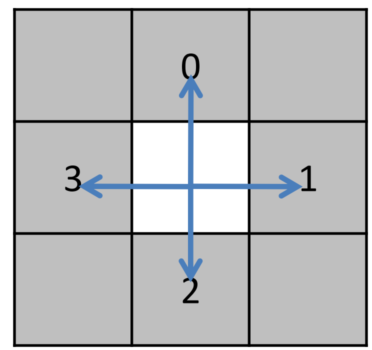
- Boundary Coding
  - encoding schema
  - start point
  - chain code links
  - chain code representation
  - polygon approximation

Region Representation

- Boundary Coding
  - encoding schema
  - start point



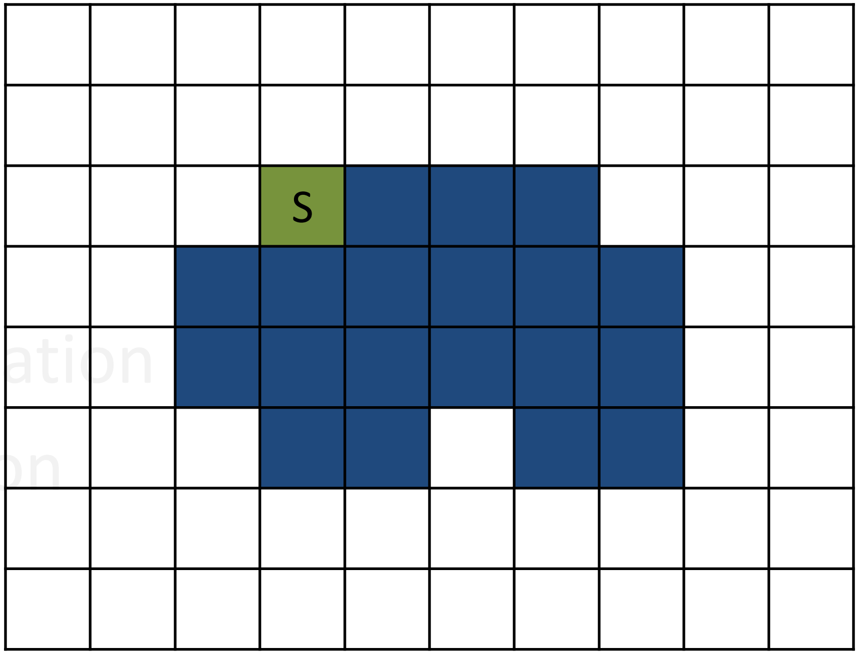
8-N



4-N

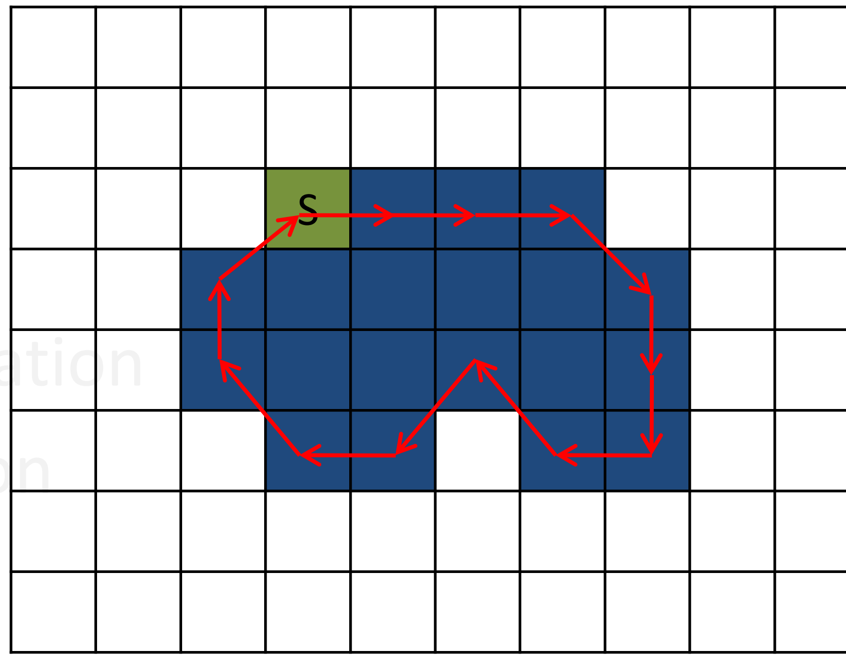
Region Representation

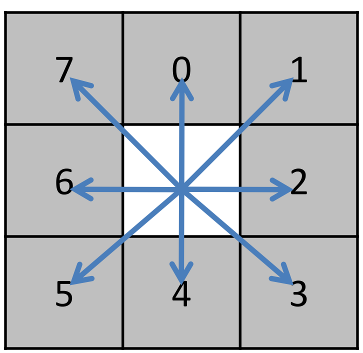
- Boundary Coding
  - encoding schema
  - start Point
  - chain code links
  - chain code representation
  - polygon approximation



Region Representation

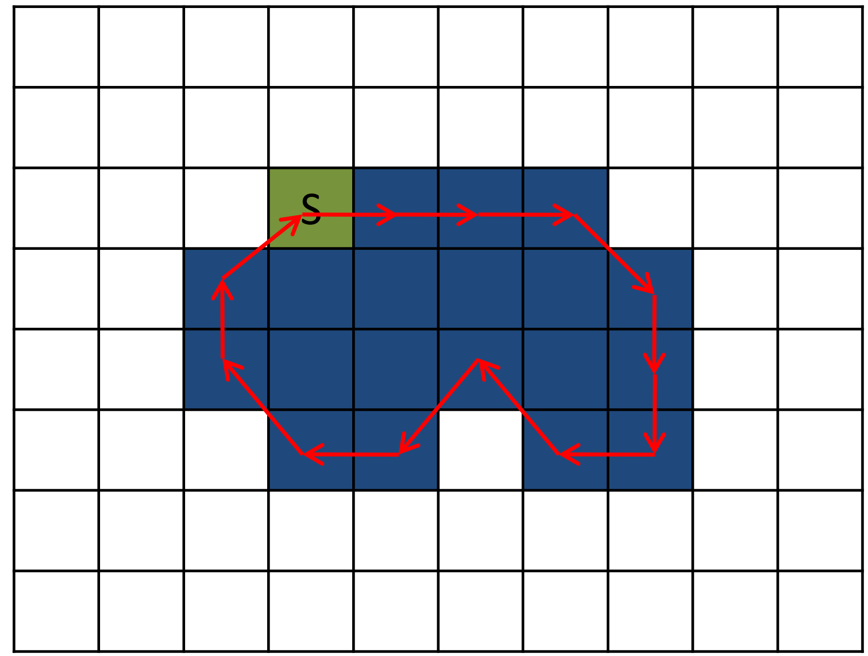
- Boundary Coding
  - encoding schema
  - start Point
  - chain code links
  - chain code representation
  - polygon approximation

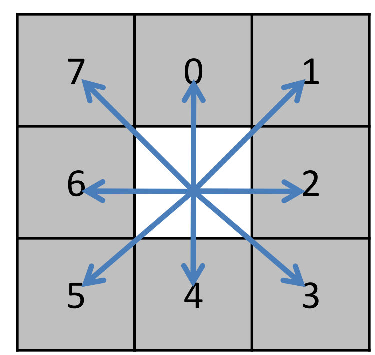




Region Representation

- Boundary Coding
  - encoding schema
  - start Point
  - chain code links
  - chain code representation
  - polygon approximation



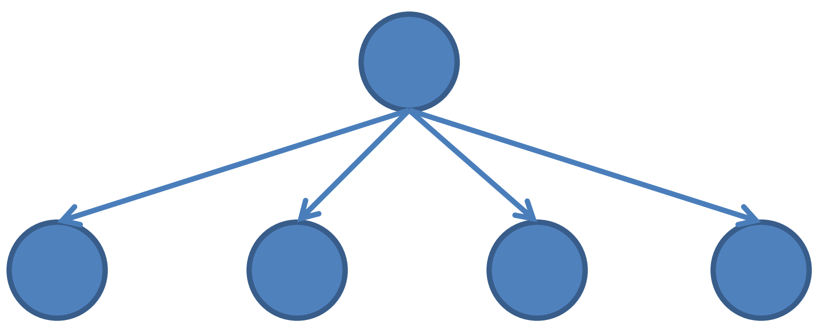


Chain Code:

2223446756701

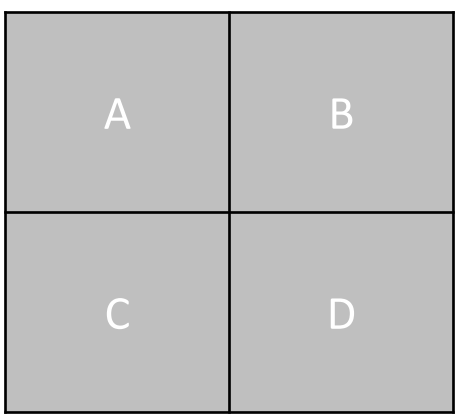
Region Representation

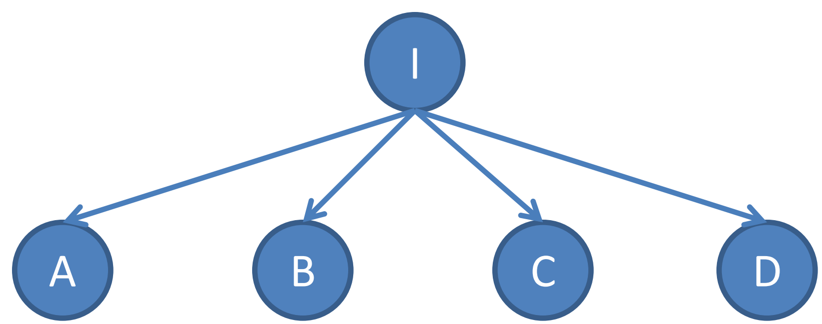
- Quadtrees
  - a **quadtree** is a tree data structure in which each internal node has exactly four children
  - quadtrees are most often used to partition a two dimensional space by recursively subdividing it into four quadrants or regions



Region Representation

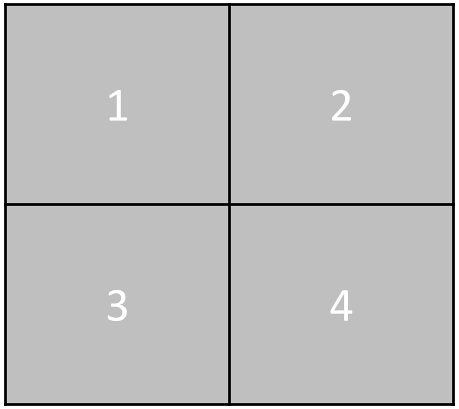
- Quadtrees
  - space-saving representation encoding the whole region
  - each region of interest would be represented by a quadtree structure.





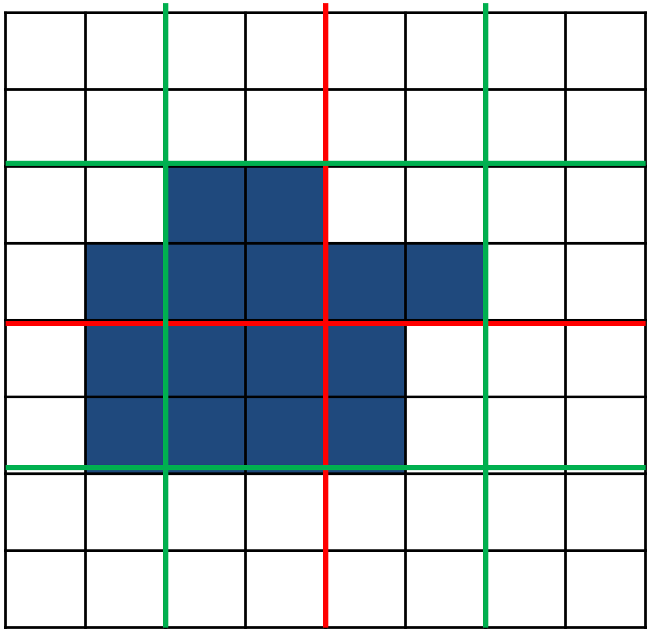
Region Representation

- Quadtrees
  - encoding Schema
  - split the image
  - create the tree



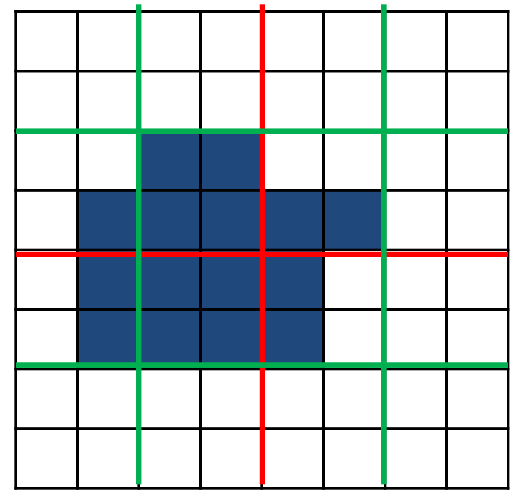
Region Representation

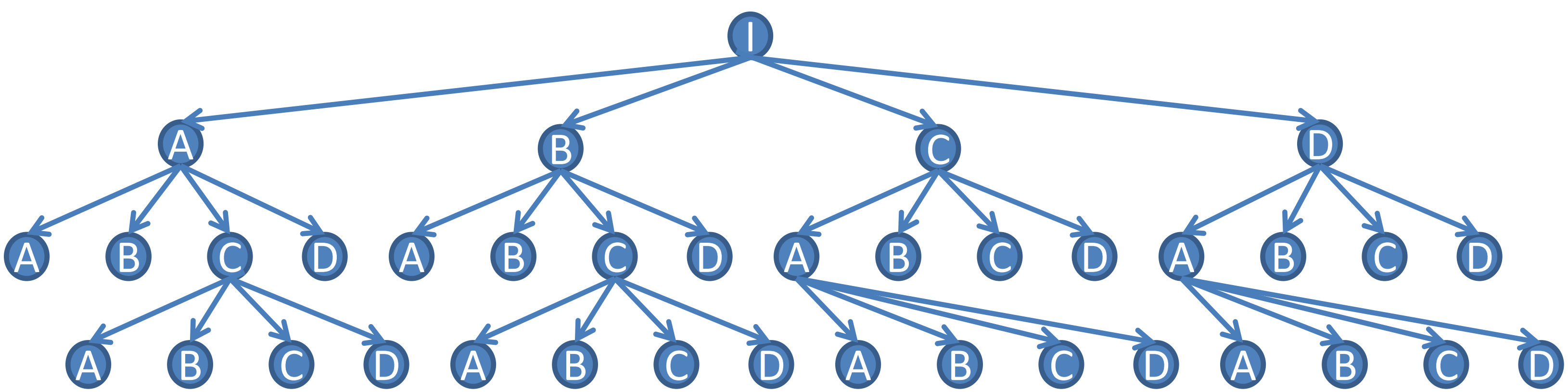
- Quadtrees
  - encoding Schema
  - split the image
  - create the tree



Region Representation

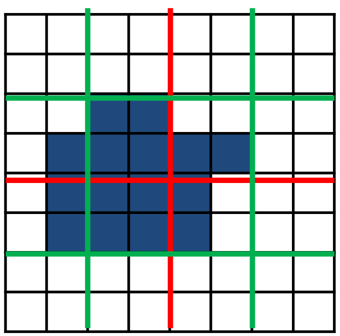
- Quadtrees
  - encoding Schema
  - split the image
  - create the tree

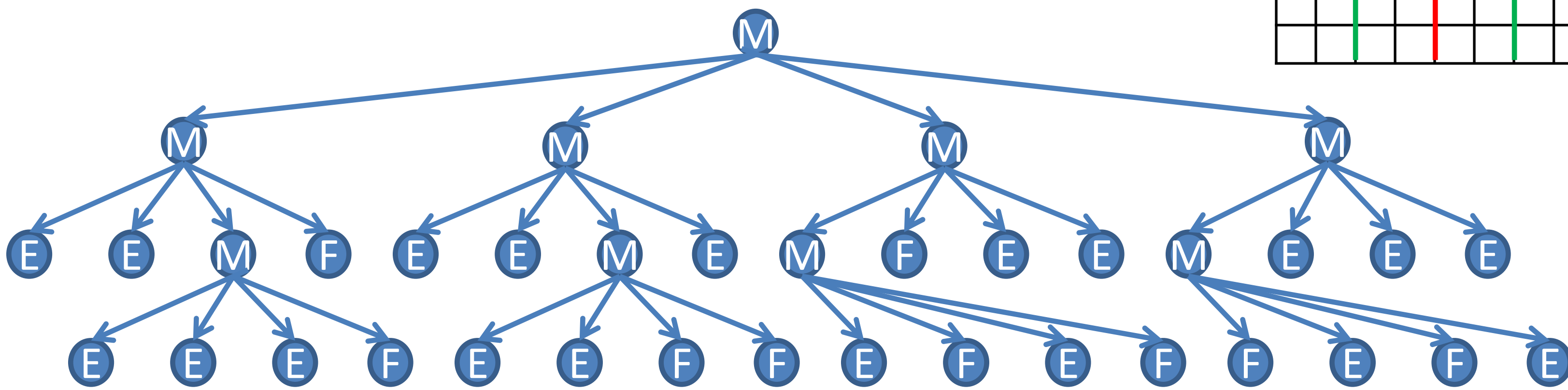




Region Representation

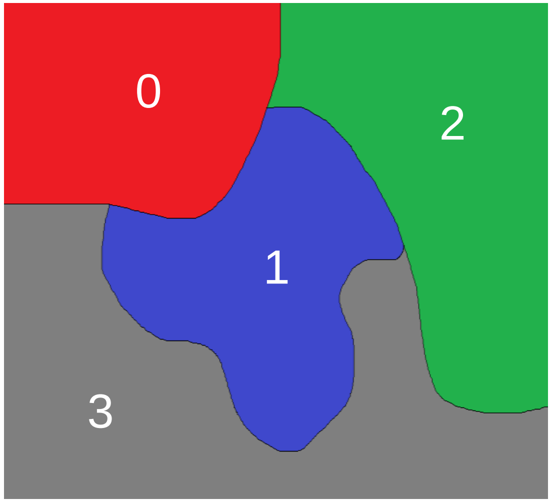
- Quadtrees
  - each node of a quadtree representing a square region in the image has one of three labels: *Full, Empty or Mixed*
  - only mixed nodes have child nodes



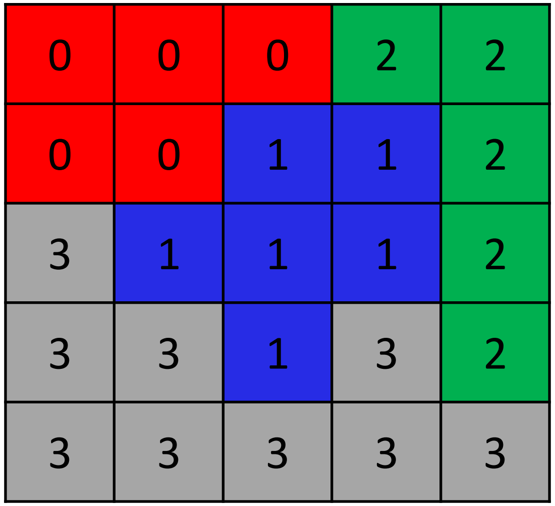


Region Representation

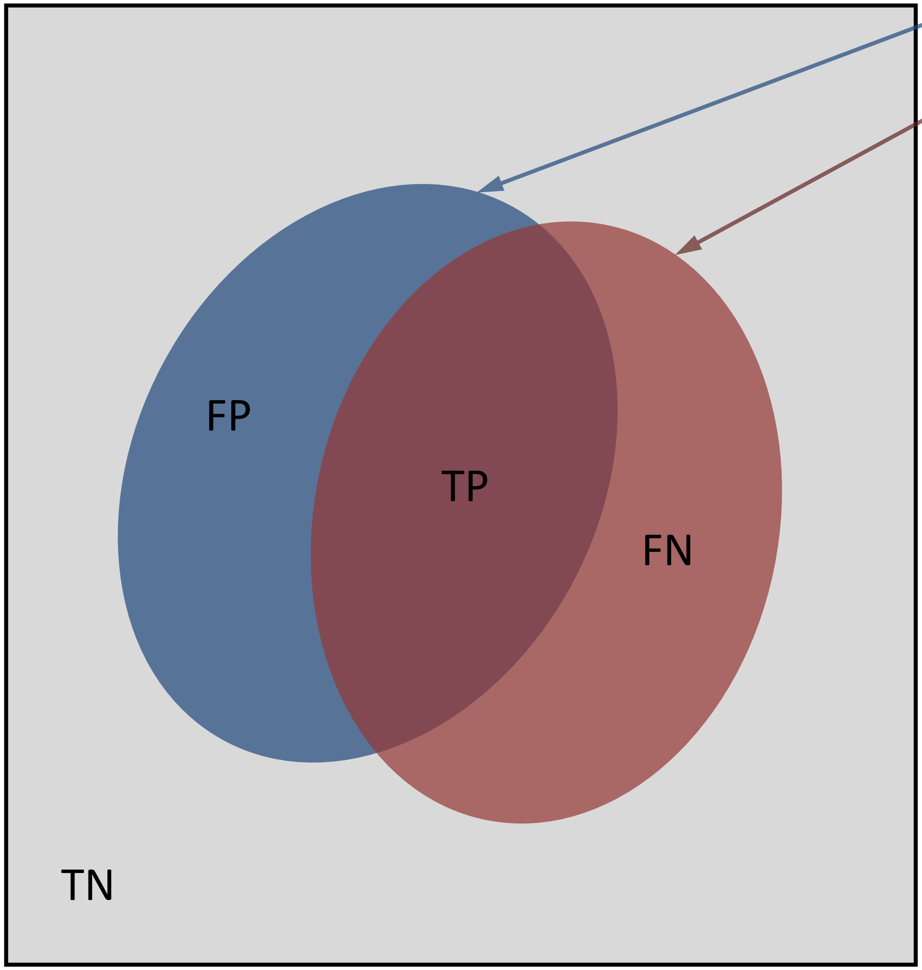
- Property Tables
  - represent a region by its extracted properties rather than by its pixels
  - properties can be the size, shape, intensity, colour or texture of the region



Region	Size	colour	…
0	5	Red	
1	6	Blue	
2	5	Green	
3	9	Grey	



Segmentation Evaluation Metrics



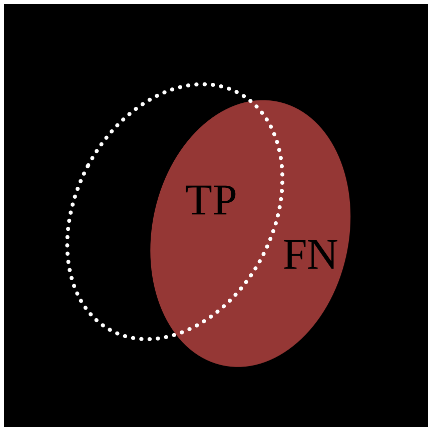
Segmented object pixels (set S)

True object pixels (set T)

Pixel classification

- True Positives $\mathrm{TP} = S \cap T$
  Correctly segmented as object
- True Negatives $\mathrm{TN} = S^c \cap T^c$
  Correctly segmented as background
- False Positives $\mathrm{FP} = S \cap T^c$
  Incorrectly segmented as object
- False Negatives $\mathrm{FN} = S^c \cap T$
  Incorrectly segmented as background

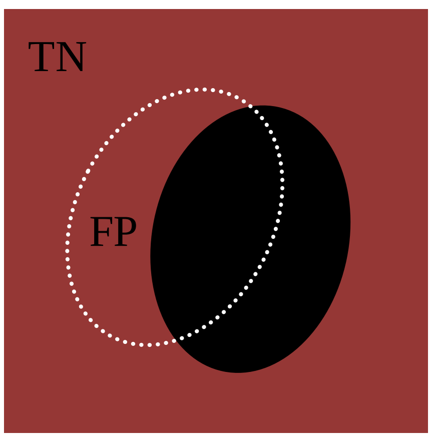
Evaluation Metrics



- Sensitivity (= true-positive rate)

  Fraction of the true object that is correctly segmented

$$\mathrm{TPR} = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|}$$



- Specificity (= true-negative rate)

  Fraction of the true background that is correctly segmented

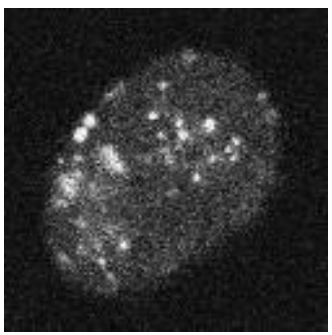
$$\mathrm{TNR} = \frac{|\mathrm{TN}|}{|\mathrm{TN}| + |\mathrm{FP}|}$$

Evaluation Metrics

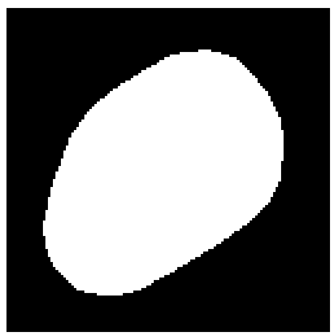
- Receiver operating characteristic (ROC) of a method

Plot of the true-positive rate (sensitivity) versus the false-positive rate (one minus the specificity) of a method as a function of its free parameter(s)

Example: Thresholding

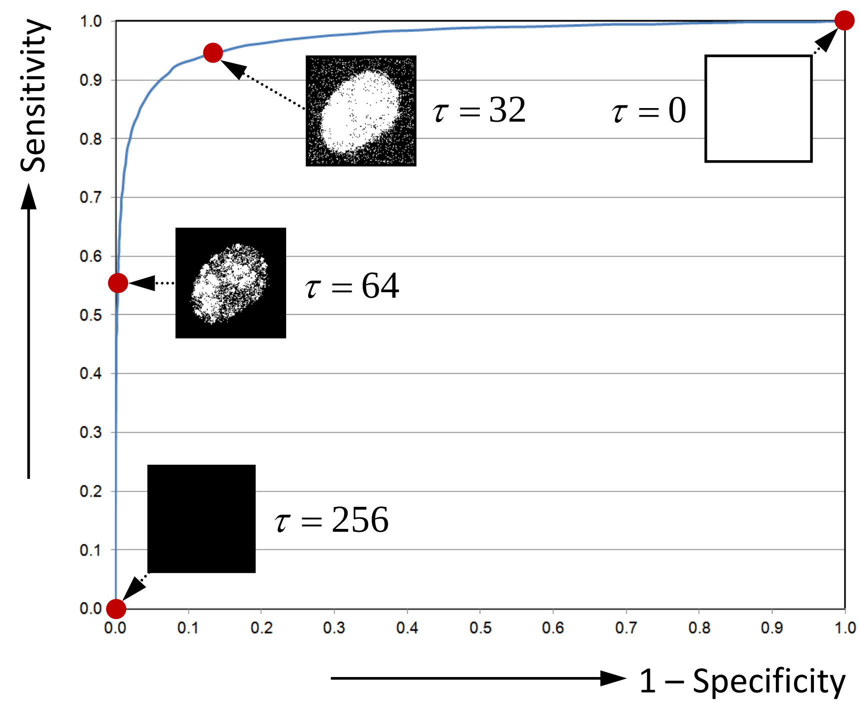


Image



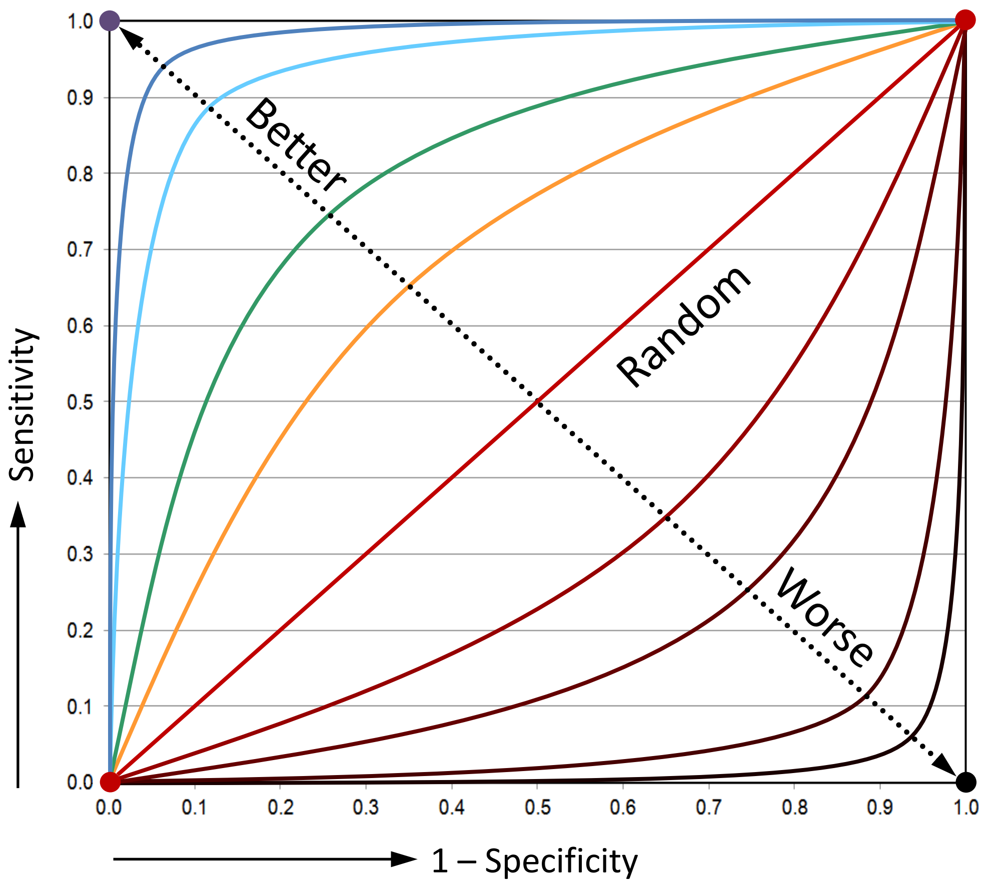
Truth

Compute the sensitivity versus the specificity for all possible intensity thresholds τ and plot the results



Evaluation Metrics

- Comparing the performance of methods by ROC analysis

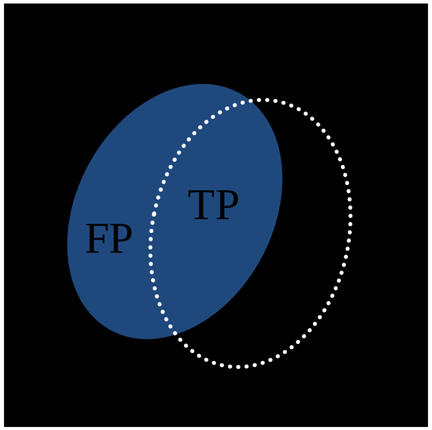


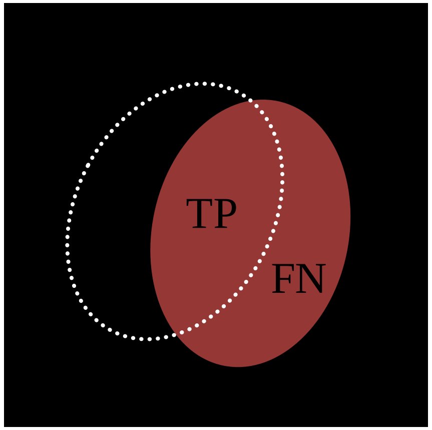
Area under the curve (AUC)

Worst	=	0.0
		0.1
		0.2
		0.3
		0.4
Random	=	0.5
		0.6
		0.7
		0.8
		0.9
Best	=	1.0

Higher AUC = better method

Evaluation Metrics





- Precision (= positive predictive value)

  Fraction of the segmented object that is correctly segmented

$$P = \frac{|TP|}{|TP| + |FP|}$$

- Recall (= sensitivity)

  Fraction of the true object that is correctly segmented

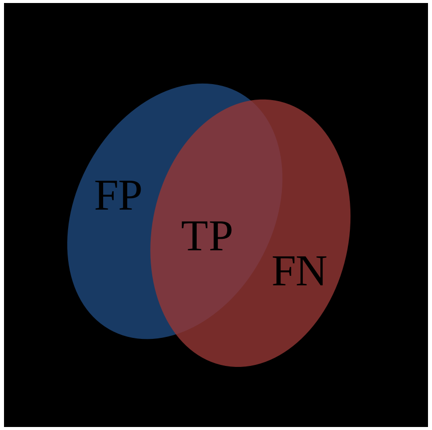
$$R = \frac{|TP|}{|TP| + |FN|}$$

- F-measure

  Harmonic mean of precision and recall

$$F = 2 \cdot \frac{R \cdot P}{R + P}$$

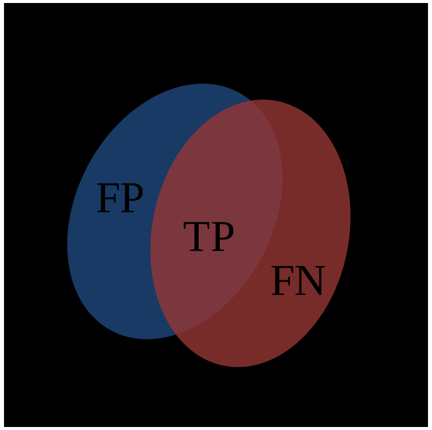
Evaluation Metrics



- Jaccard similarity coefficient (JSC)

  Fraction of the union of the segmented object and the true object that is correctly segmented

$$\mathrm{JSC} = \frac{|S \cap T|}{|S \cup T|} = \frac{|\mathrm{TP}|}{|\mathrm{FP}| + |\mathrm{TP}| + |\mathrm{FN}|}$$



- Dice similarity coefficient (DSC)

  Fraction of the segmented object set joined with the true object set that is correctly segmented

$$\mathrm{DSC} = 2\frac{|S \cap T|}{|S| + |T|} = \frac{2|\mathrm{TP}|}{|\mathrm{FP}| + 2|\mathrm{TP}| + |\mathrm{FN}|}$$

References and Acknowledgements

- Chapter 3 & 5 of Szeliski 2010
- Shapiro and Stockman 2001
- Some images drawn from Szeliski 2010
- Some images drawn from papers as indicated