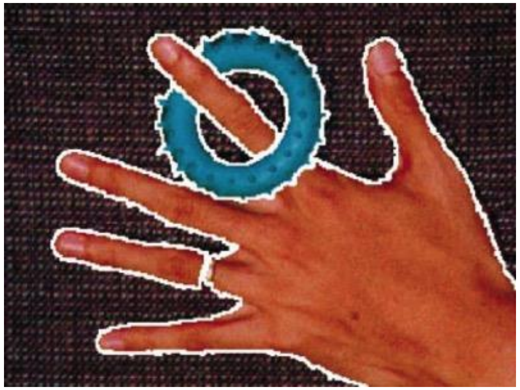
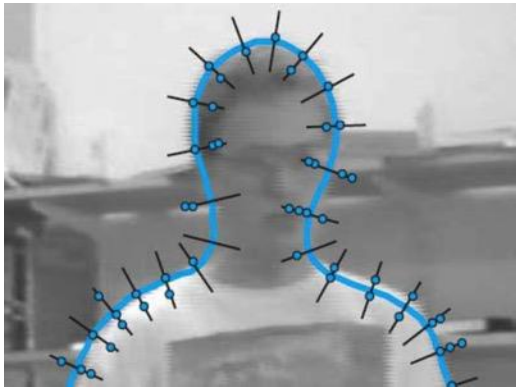
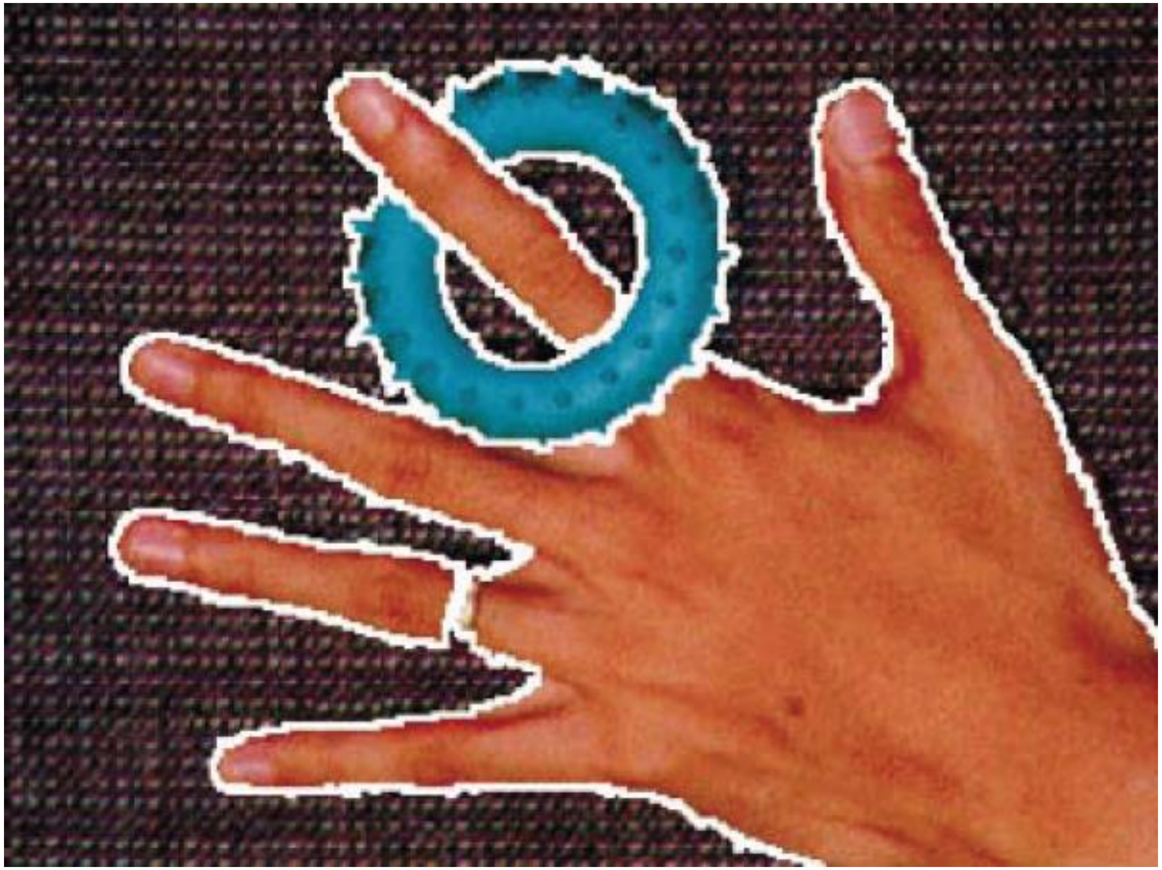
  

COMP 9517 Computer Vision

Segmentation

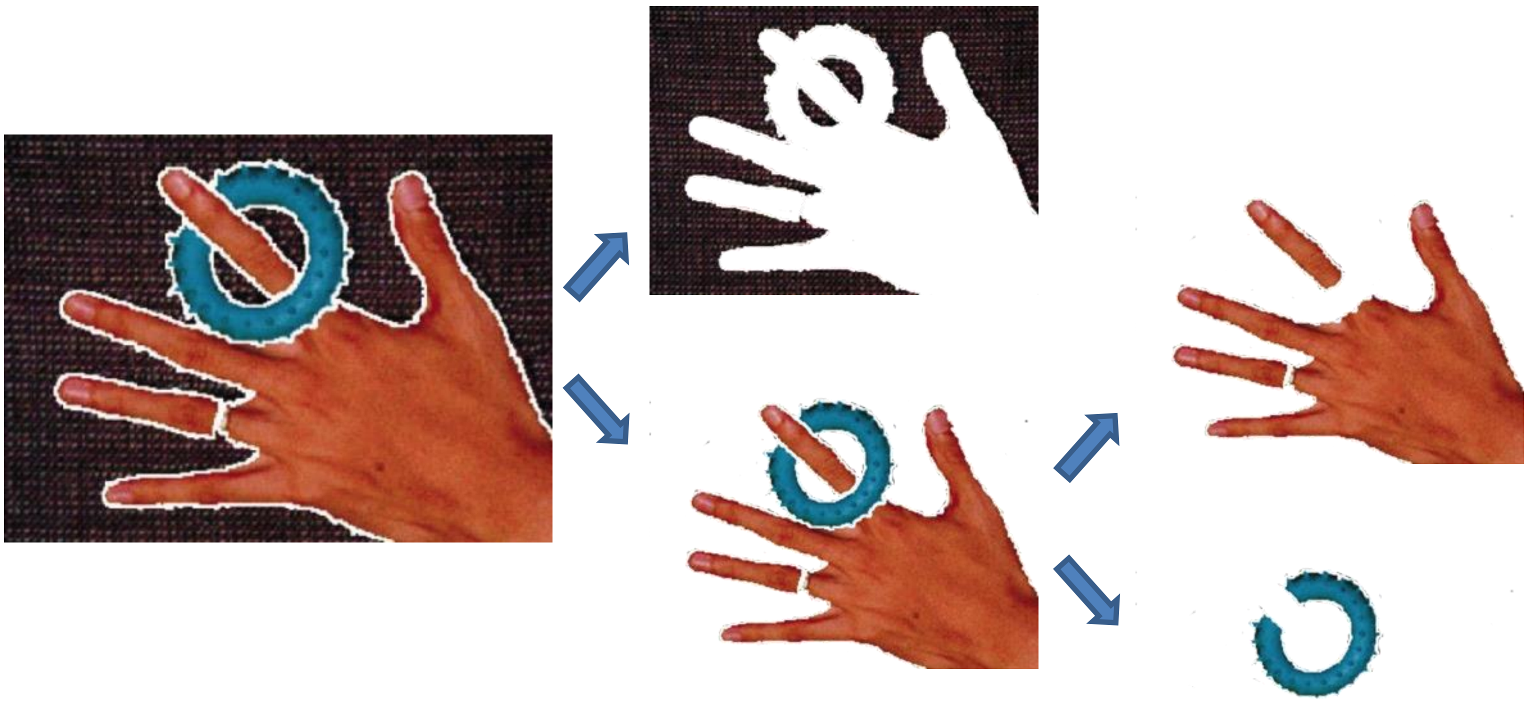
Introduction

- What can you see from this image?



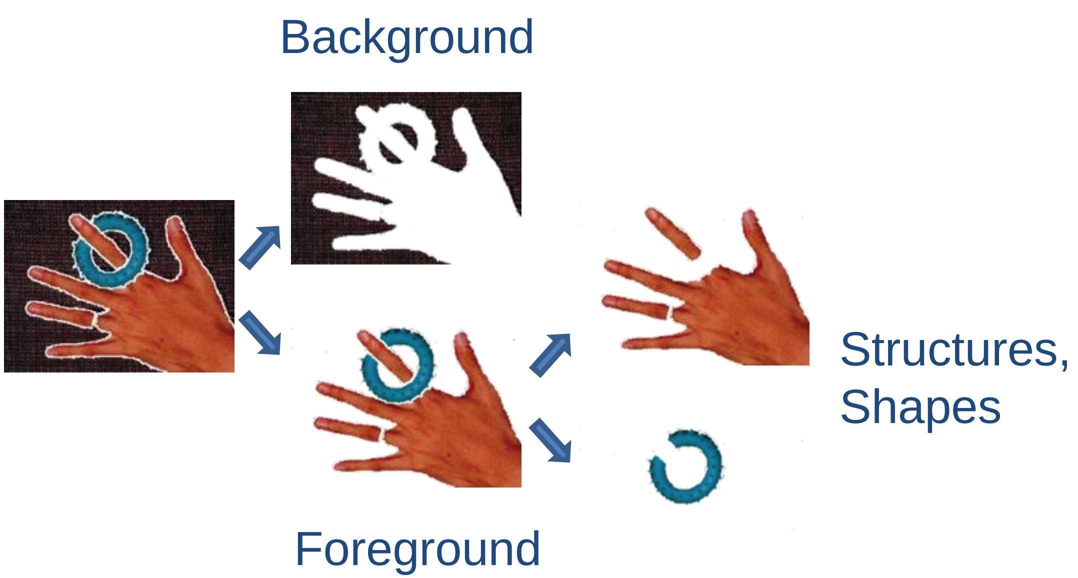
Introduction

- Image segmentation
  - one of the oldest and most widely studied problems in computer vision



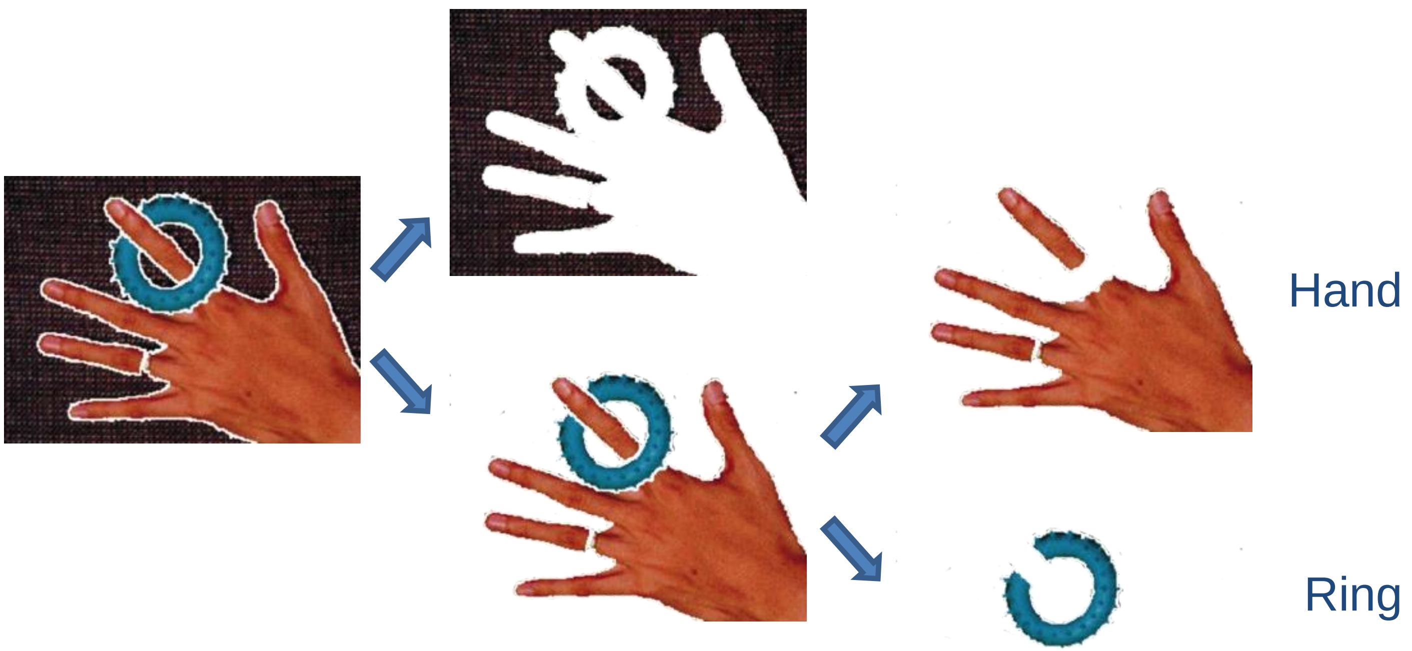
Introduction

- Image segmentation is the partitioning of an image into a set of regions
  - meaningful areas
  - border pixels grouped into structures
  - groups of pixels with shapes
  - foreground and background



Introduction

- Two objectives of segmentation
  - **decompose** image into parts for further analysis
  - **represent** image using meaningful and/or efficient higher-level units for further analysis



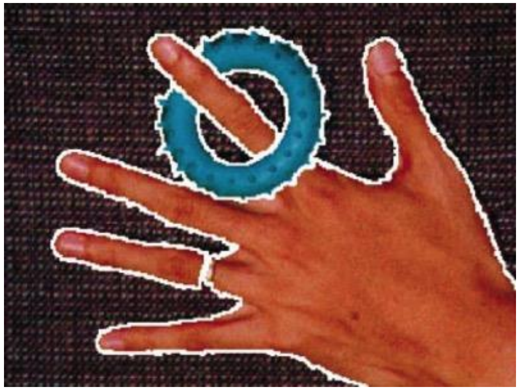
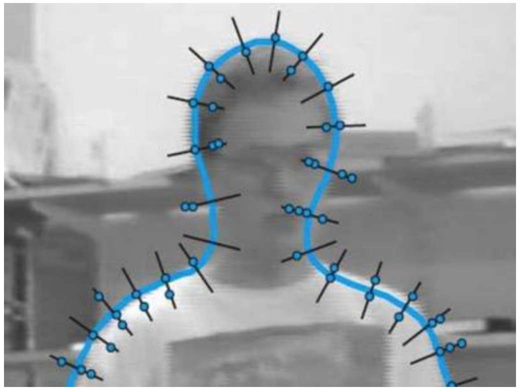
Introduction

- Identifying Regions
  - regions should be ***uniform and homogeneous*** with respect to some characteristic
  - region interiors should be ***simple*** and without many small holes
  - adjacent regions should have ***significantly different*** values with respect to the characteristics on which they are uniform
  - boundaries of each segment should be ***smooth*** and spatially accurate



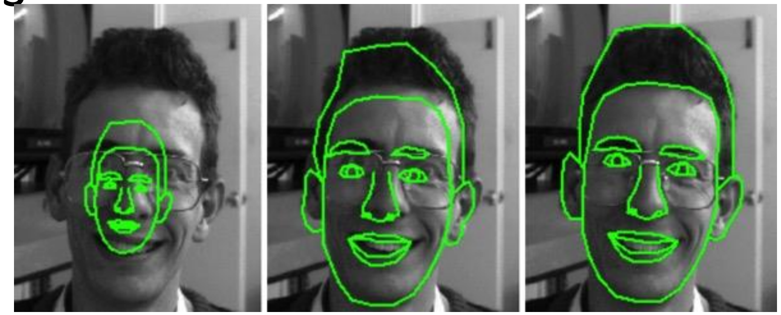
Segmentation Approaches

- region-based
- curve-based
- early techniques tend to use region splitting and/or merging
- recent algorithms optimise some global criterion

Segmentation Issues

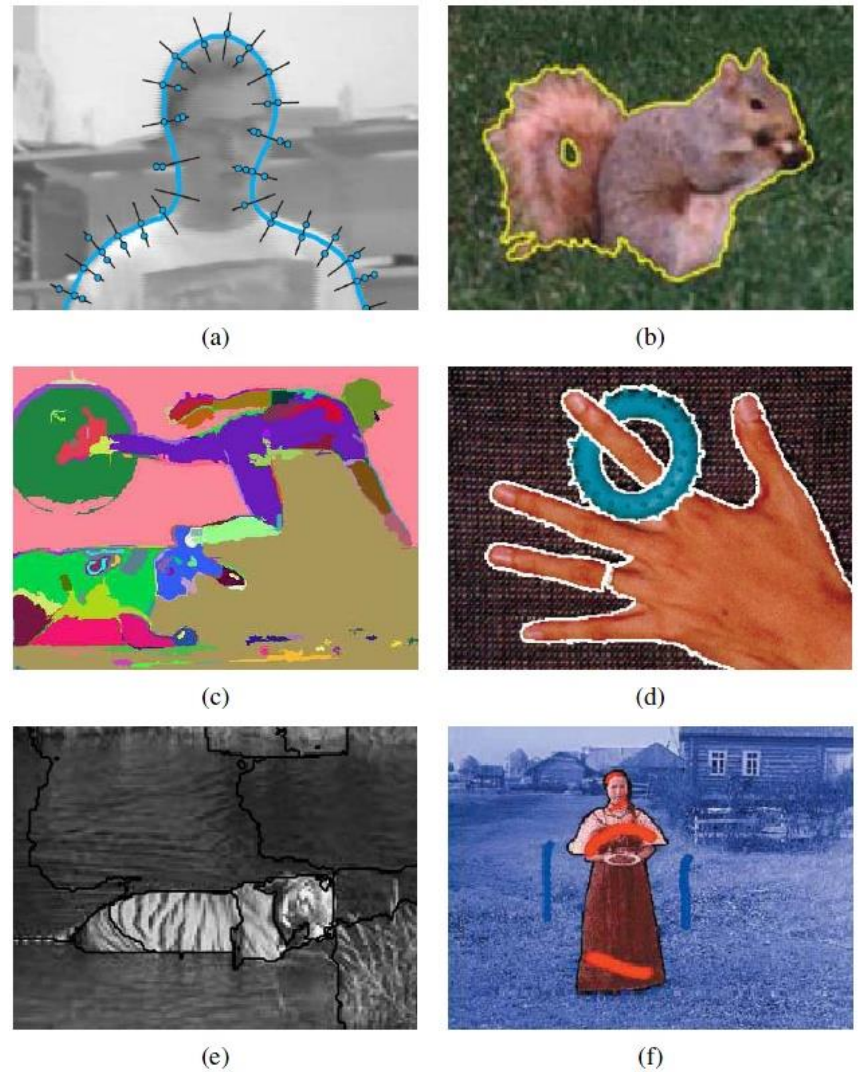
- **no single** segmentation method that works well for all problems
- special **domain knowledge** of the application is typically essential for implementation of computer vision applications



Source: http://www.isbe.man.ac.uk/~bim/Models/asms.html

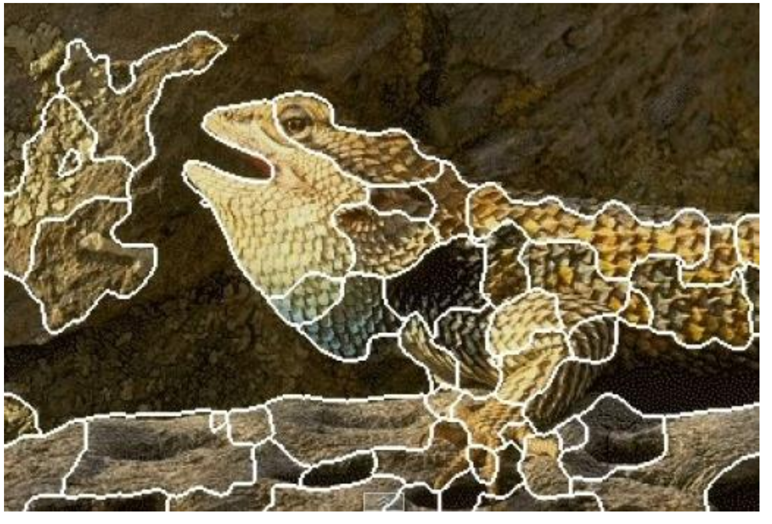
Popular Techniques

- Results from some popular segmentation techniques
  a) Active contours
  b) Level sets
  c) Graph-based merging
  d) Mean shift
  e) Normalised cuts
  f) Binary MRF



Region Split and Merge

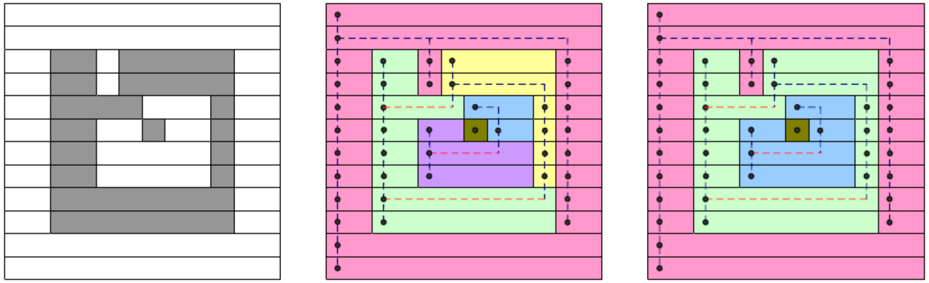
- Overview
  - recursively split whole image into parts, based on region statistics
  - recursively merge pixels and regions together in hierarchical fashion
  - combine both splitting and merging together



Region Split and Merge

- Simplest possible techniques
  - use a threshold and then compute connected components
  - rarely sufficient due to lighting and intra-object statistical variations

$$\theta(f,t) = \begin{cases} 1 & \text{if } f \geq t, \\ 0 & \text{else,} \end{cases}$$



Split and Merge

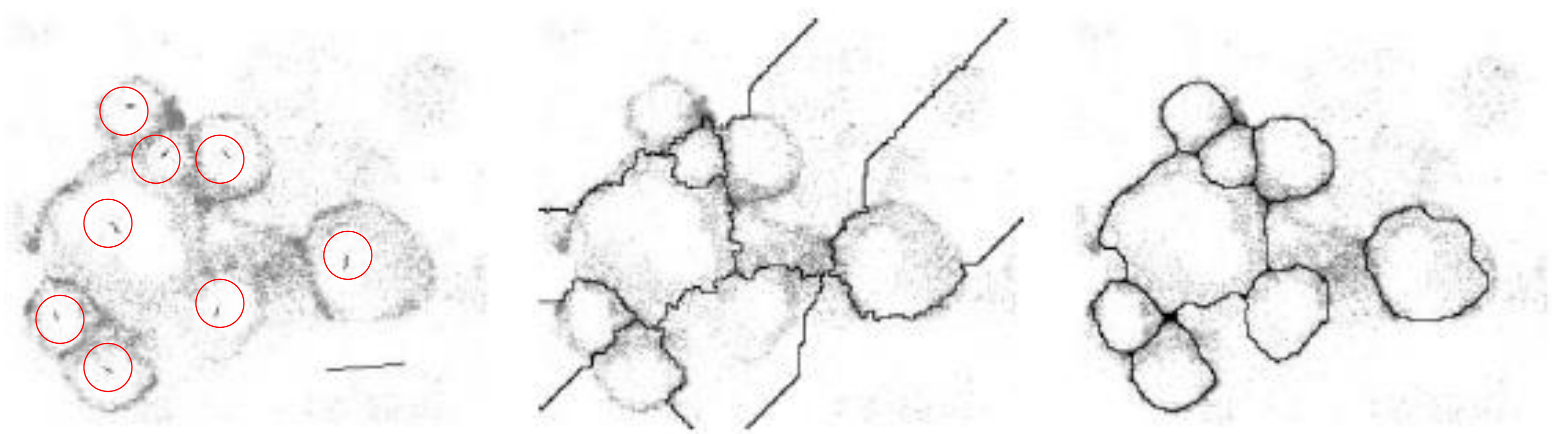
- Watershed
  - technique related to **thresholding**
  - operates on **grayscale** image
  - segments an image into several **catchment basins**
  - **analogy**: regions of an image where rain would flow into the same lake
  - start flooding the landscape at all of the **local minima** and label ridges whenever different evolving components meet
  - often used as part of an interactive system, where the user first marks seed locations that correspond to the centres of different desired components

Split and Merge

- Watershed continued
  - Meyer's flooding algorithm
    1) Choose a set of markers to start the flooding, each is given a different label
    2) The neighbouring pixels of each marked area are inserted into a priority queue, with a priority level corresponding to the gray level of the pixel
    3) The pixel with the highest priority level is popped from the queue. If the neighbours of the popped pixel that have already been labelled all have the same label, the pixel is labelled with the label. All non-labelled neighbours that are not yet in the queue are put into the queue.
    4) Repeat step 3 until the queue is empty
  - The non-labelled pixels are the watershed lines

Split and Merge

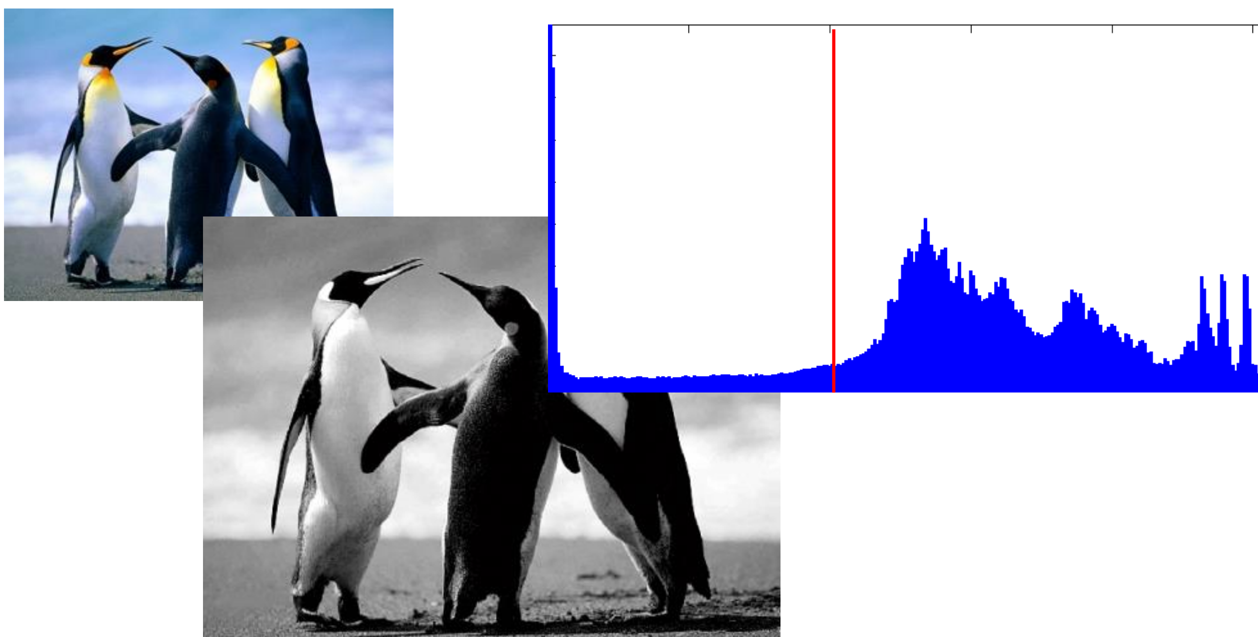
- Watershed



(a) (b) (c)

Split and Merge

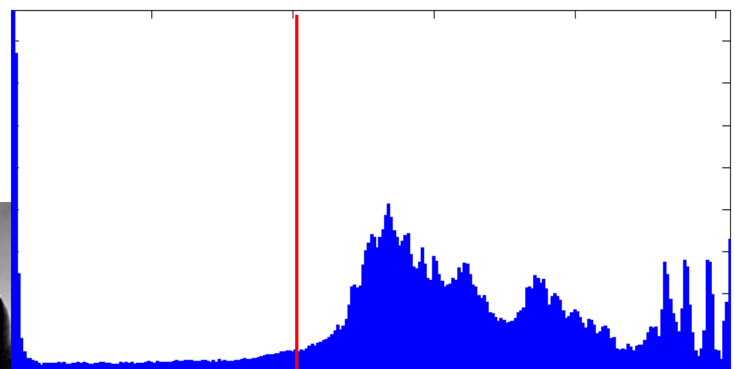
- Region Splitting
  - one of the oldest techniques in computer vision
  - first computes a histogram for the whole image
  - then finds a threshold that best separates the large peaks in the histogram



Split and Merge

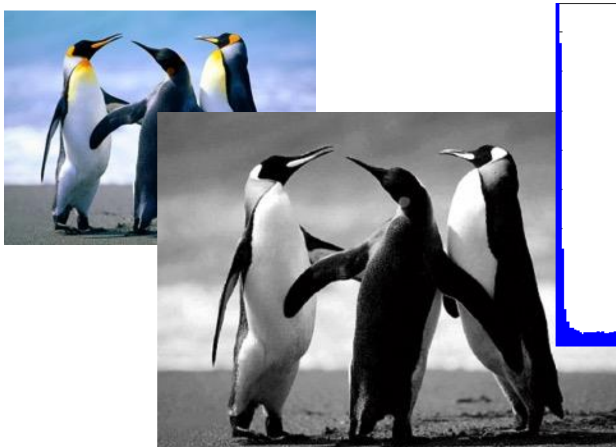
- Region Splitting
  - repeat until regions are either fairly uniform or below a certain size
  - more recent approaches often optimise some metric of **intra-region similarity** and **inter-region dissimilarity**

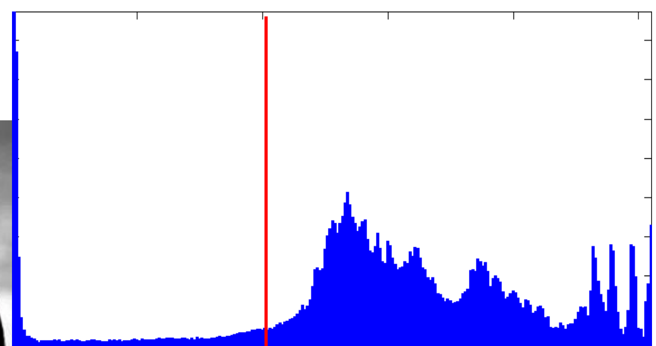





Split and Merge

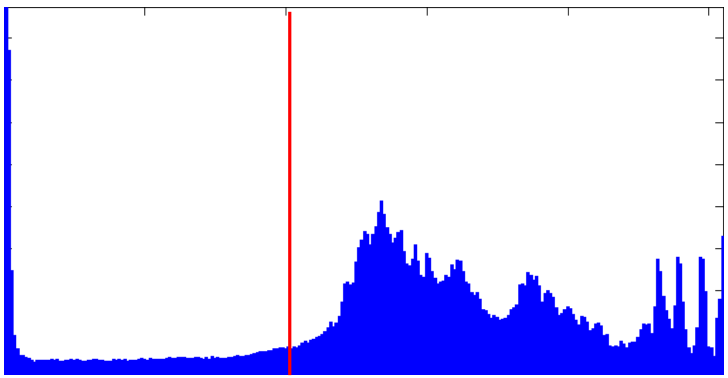
- Histogram-based methods
  - follows a measurement-space clustering process
  - assumes homogeneous objects in the image appear as clusters in measurement space (histogram)
  - measurement-space clustering can be accomplished by determining the valleys in the histogram and declaring the clusters to be the interval of values between valleys

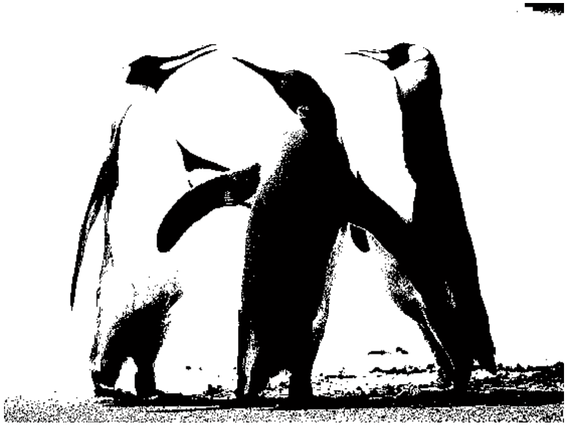




Split and Merge

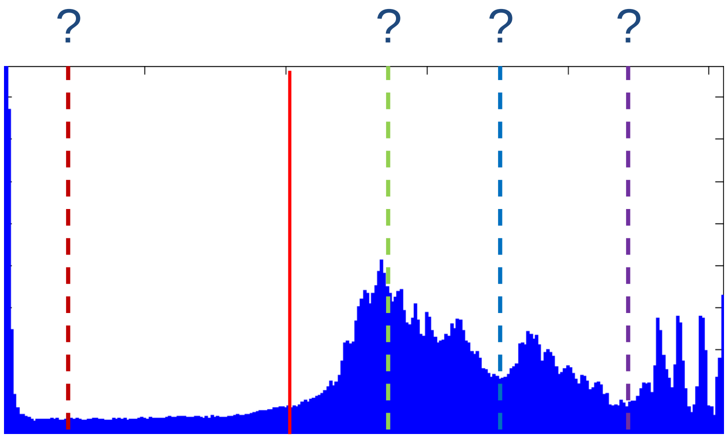
- Histogram-based methods
  - segmentation is accomplished by mapping the clusters back to the image domain
  - maximum connected components of the cluster labels constitute the segments





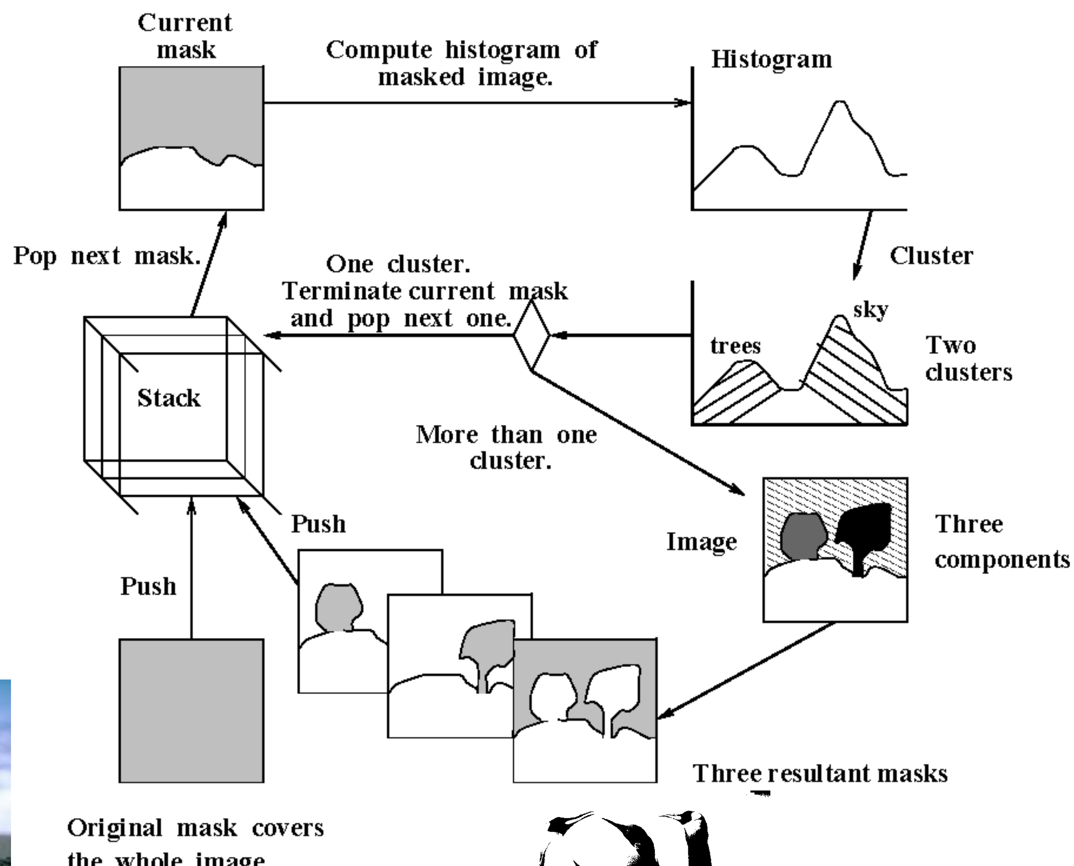
Split and Merge

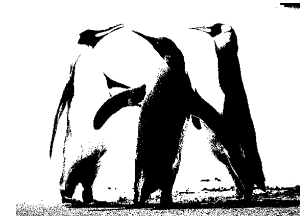
- Histogram-based methods
  - measurement-space clustering can be accomplished by determining the valleys in the histogram and declaring the clusters to be the interval of values between valleys



Split and Merge

- A recursive version of histogram-based clustering
  - perform histogram mode seeking first on the whole image
  - then on each of the regions obtained from the resultant clusters
  - until regions obtained cannot be decomposed further





Split and Merge

- Region Merging
  - merge regions based on their relative **boundary lengths** and the strength of the visible edges at the boundaries
  - link clusters together based either on the distance between their **closest points**, their **farthest points** or some thing in between
  - merge adjacent regions whose average colour difference is below a threshold or whose regions are too small
  - useful pre-processing stage to make higher-level algorithms faster and more robust

Split and Merge

- Graph-based Segmentation
  - use relative dissimilarities between regions to merge
  - start with a pixel-to-pixel dissimilarity measure such as intensity differences between N_8 neighbours
  - Internal difference

$$Int(R) = \max_{e \in MST(R)} w(e)$$

  - Difference between regions

$$Dif(R_1, R_2) = \min_{e \in (v_1, v_2) | v_1 \in R_1, v_2 \in R_2} \omega(e)$$

  - merge any two adjacent regions whose difference is smaller than the minimum internal difference of these two regions

NW	N	NE
W	C	E
SW	S	SE

Split and Merge

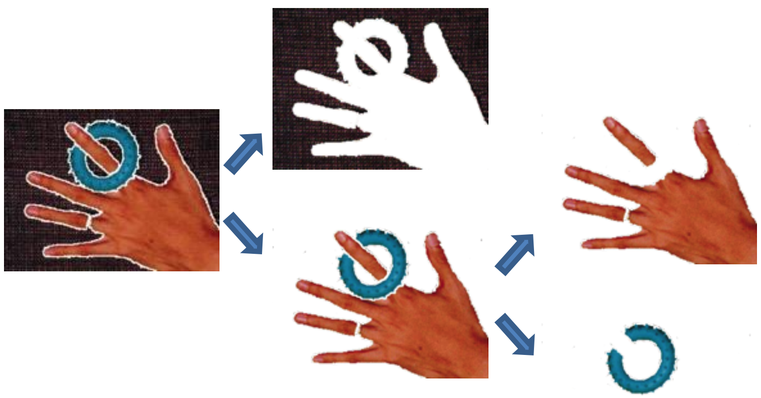
- Graph-based Segmentation

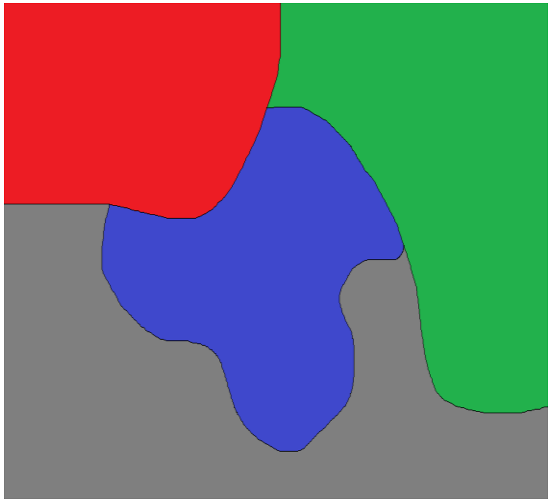
Region Representation

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

Region Representation

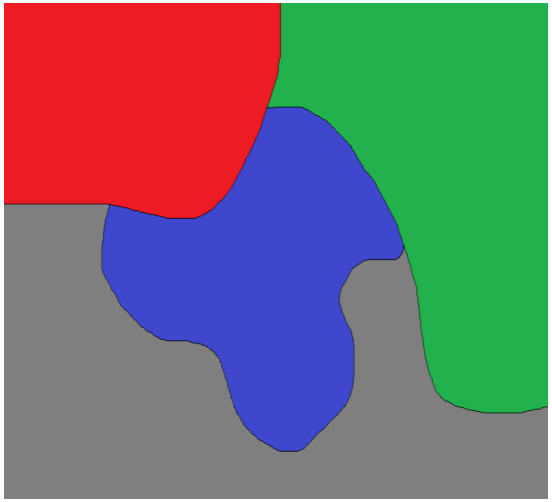
- Labelled images
  - assign each detected region a unique identifier (an integer)





Region Representation

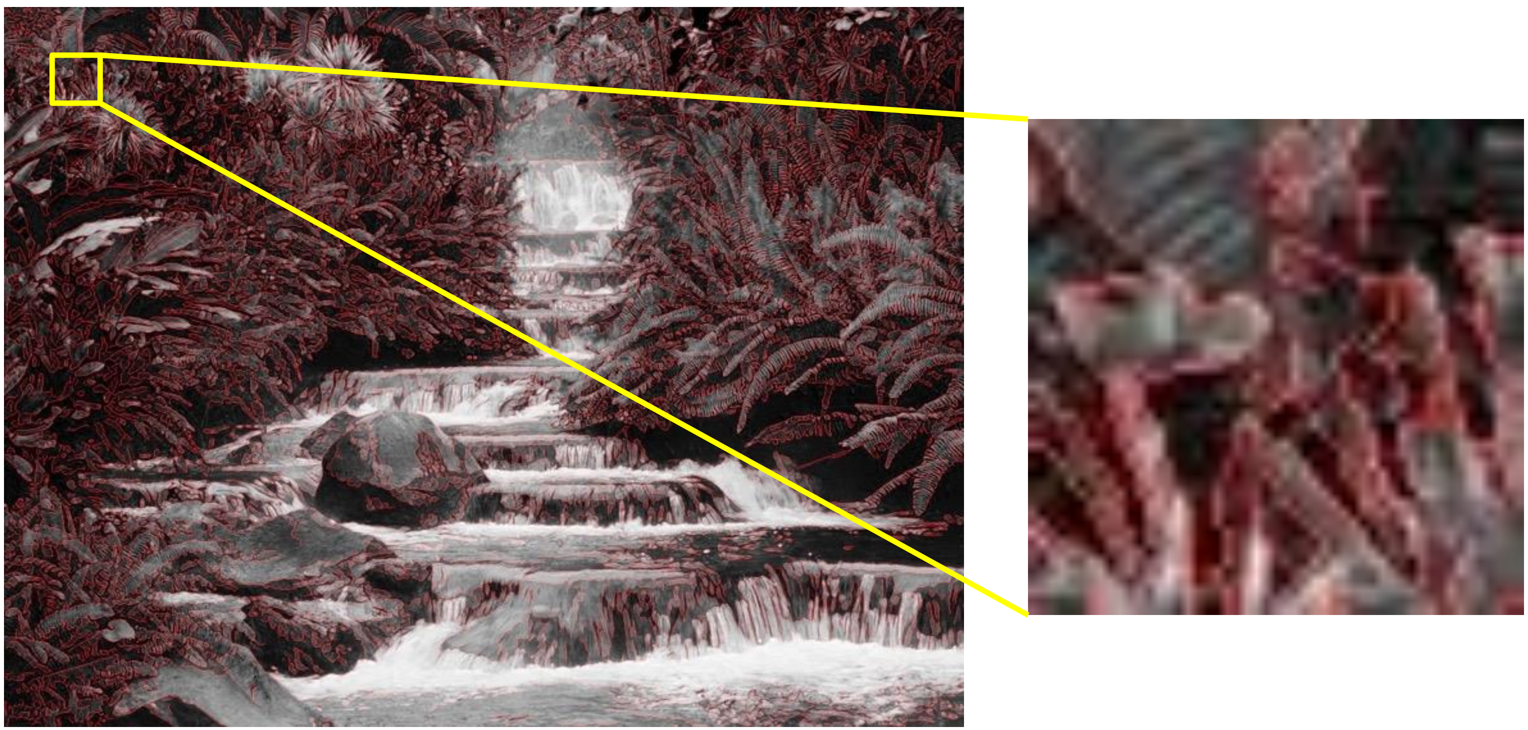
- Labelled images
  - create an image where all pixels of a region will have a unique identifier as their pixel value



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

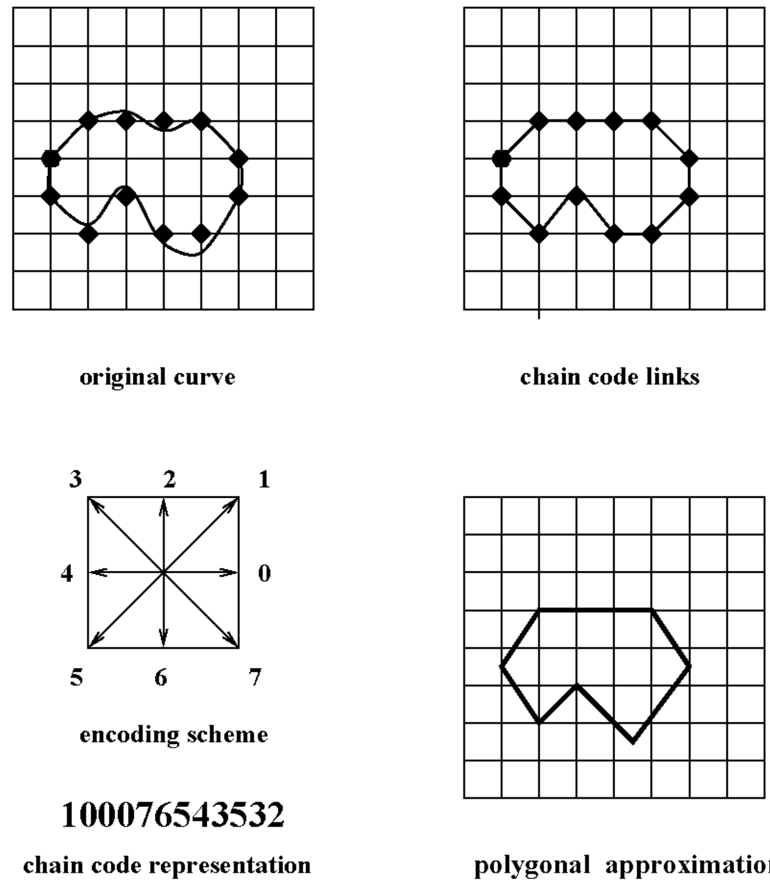
Region Representation

- Overlays
  - overlaying some colour(s) 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

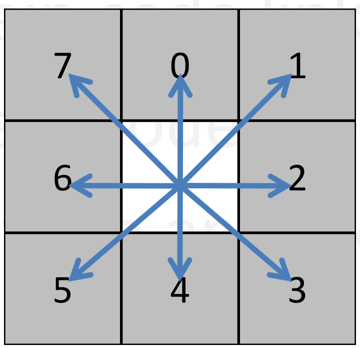


Region Representation

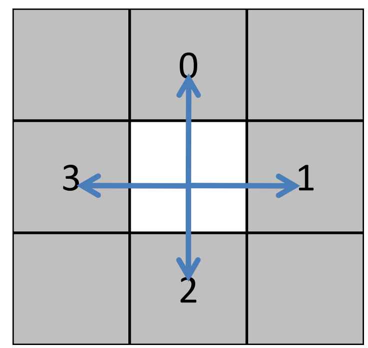
- Boundary Coding
  - encoding schema
  - start point
  - chain code links
  - chain code representation
  - polygon approximation

Region Representation

- Boundary Coding
  - Encoding schema
  - Start point



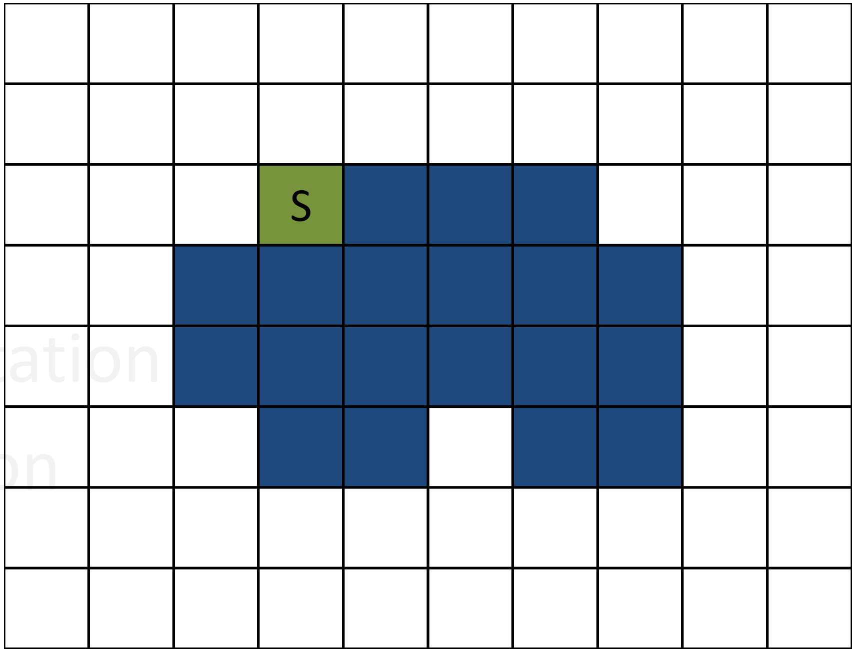
8-N



4-N

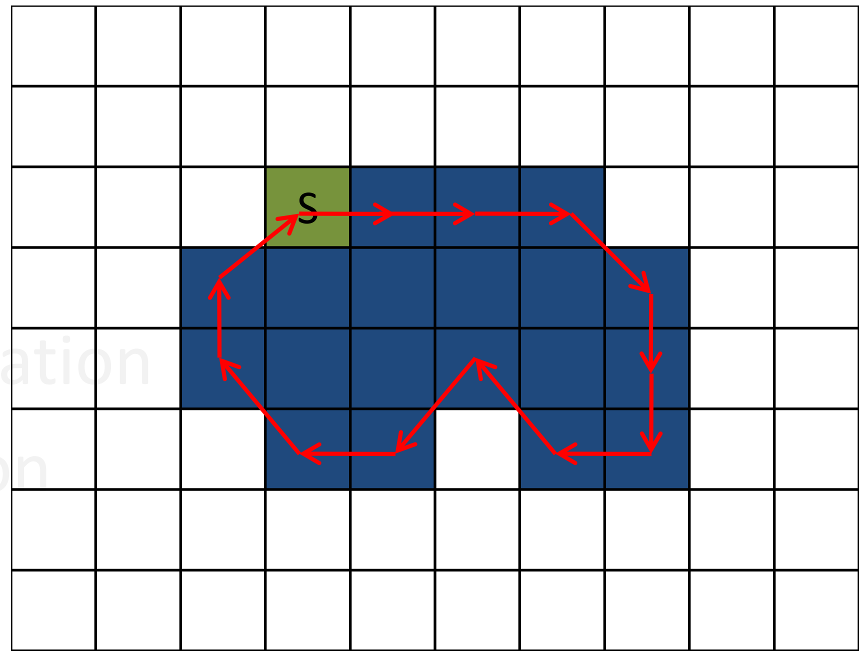
Region Representation

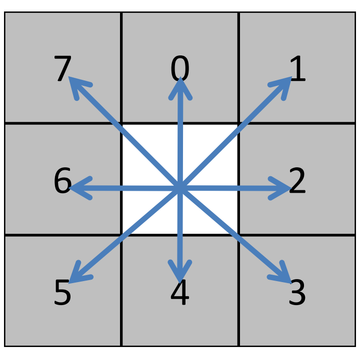
- Boundary Coding
  - Encoding schema
  - start Point
  - Chain code links
  - Chain code representation
  - Polygon approximation



Region Representation

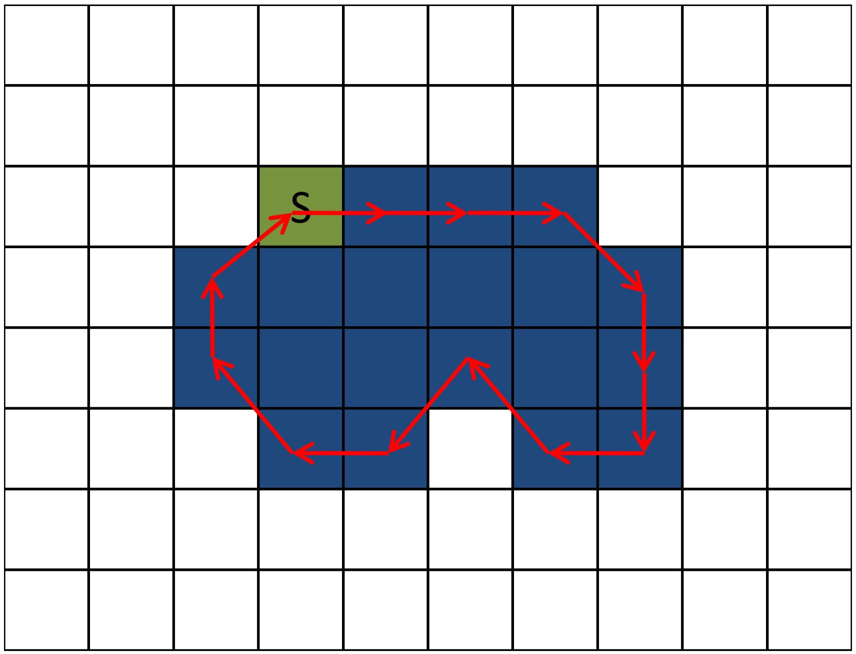
- Boundary Coding
  - Encoding schema
  - Start Point
  - chain code links
  - Chain code representation
  - Polygon approximation





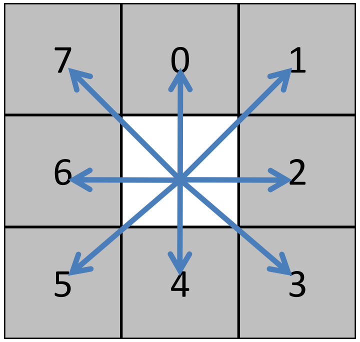
Region Representation

- Boundary Coding
  - Encoding schema
  - Start Point
  - Chain code links
  - chain code representation
  - Polygon approximation



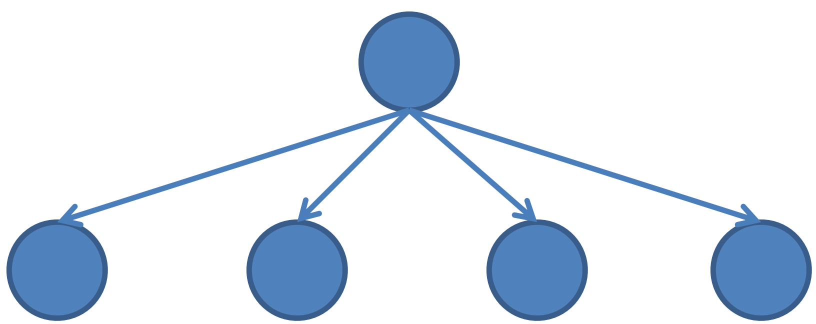
Chain Code:

2223446756701



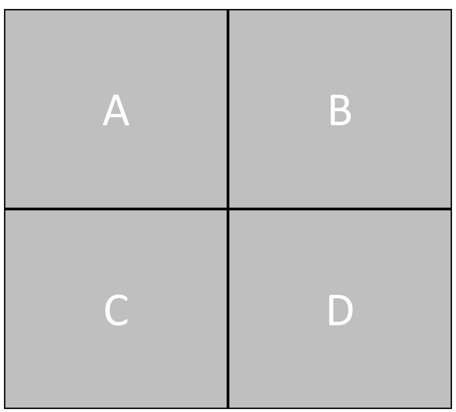
Region Representation

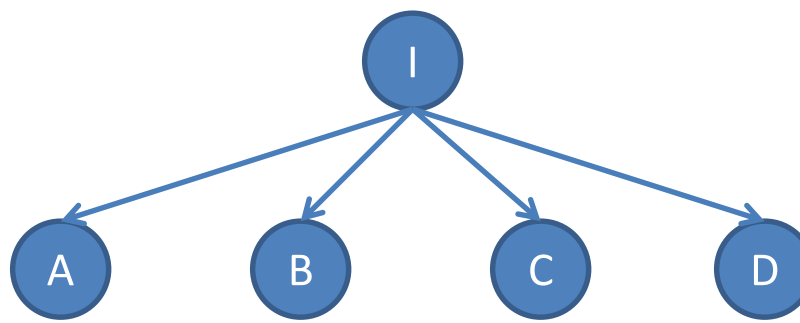
- Quadtrees
  - tree data structure in which each internal node has exactly four children
  - most often used to partition a 2-D space by recursively subdividing it into four quadrants or regions



Region Representation

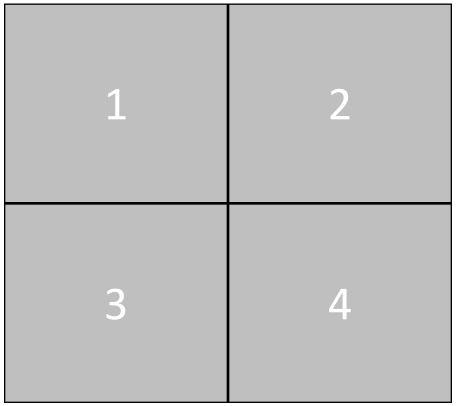
- Quadtrees
  - space-saving representation encoding the whole region
  - each region of interest represented by a quadtree structure





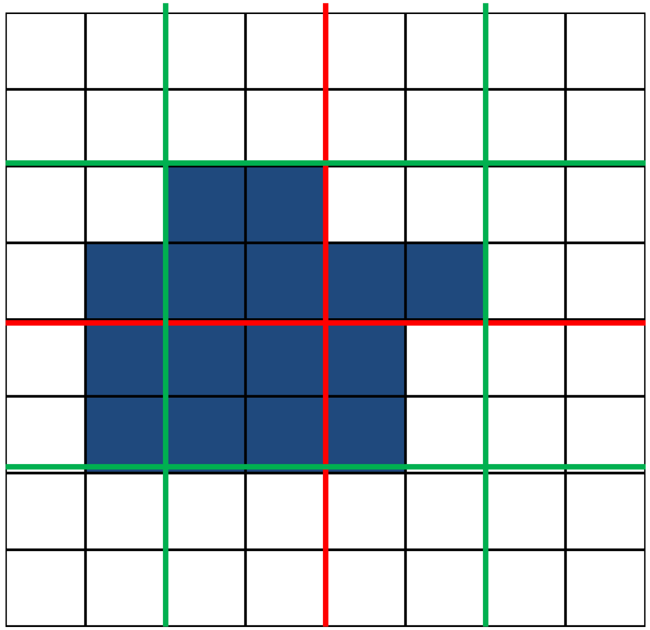
Region Representation

- Quadtrees
  - encoding Schema
  - Split the image
  - Create the tree



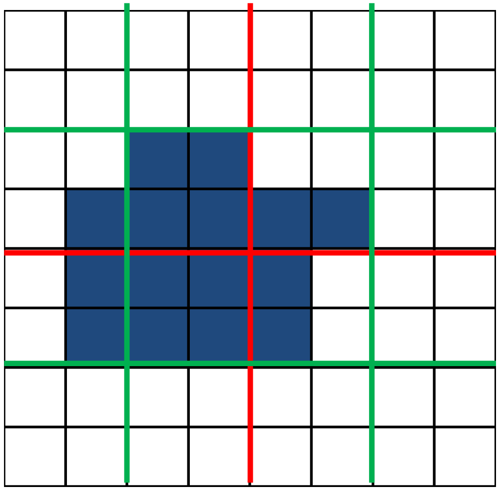
Region Representation

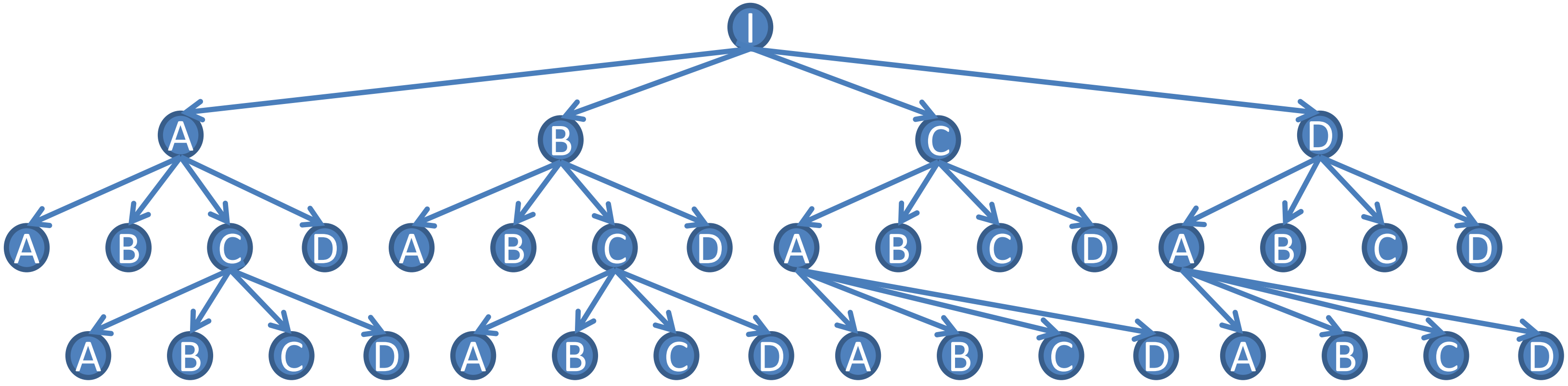
- Quadtrees
  - Encoding Schema
  - split the image
  - Create the tree



Region Representation

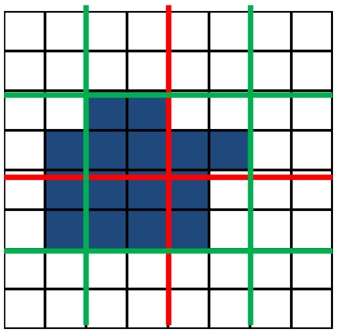
- Quadtrees
  - Encoding Schema
  - Split the image
  - create the tree

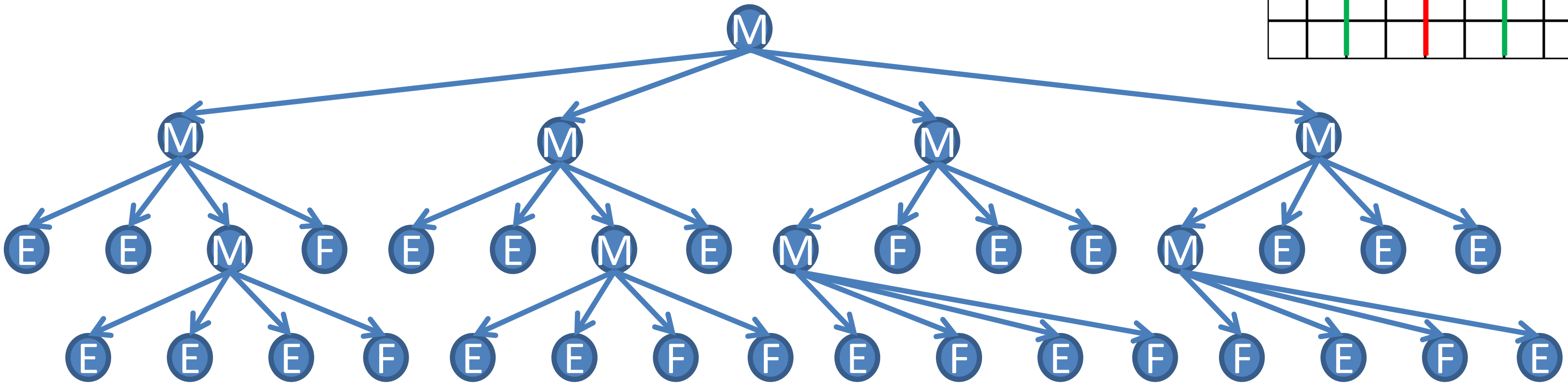




Region Representation

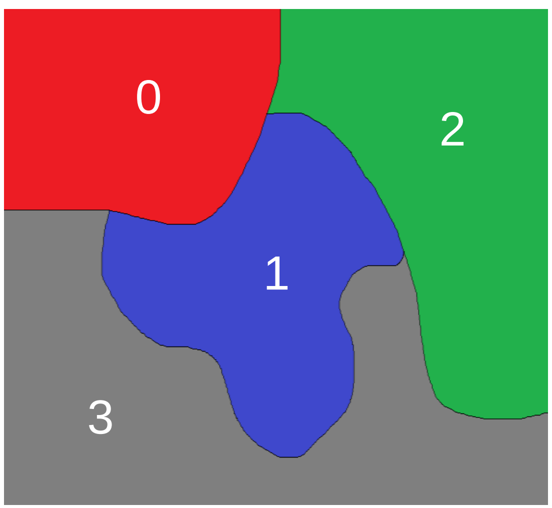
- Quadtrees
  - each node of a quadtree representing a square region in the image has one of three labels: *Full, Empty or Mixed*
  - Only mixed nodes have child nodes





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	

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

References and Acknowledgements

- Chapter 3, 5 Szeliski 2010
- Chapter 10, Shapiro and Stockman 2001
- Some images drawn from Szeliski 2010 and other papers