REGION SEGMENTATION

Duangpen jetpipattanapong

REGION

- Region
  - Region should be uniform and homogenous with respect to some characteristics, such as color, texture
  - Region interiors should be simple and without many small holes
  - Adjacent regions of a segmentation should have significantly different values with respect to the characteristic on which they are uniform
  - Boundaries of each segment should be smooth, not ragged, and should be especially accurate

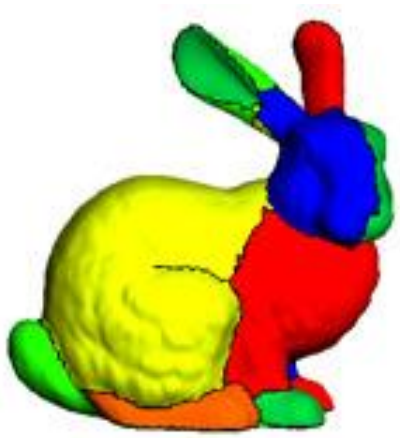


Image segmentation

- refers to partitioning image into a set of regions that cover it.
- The goal in many tasks is for the regions to represent meaningful areas of the image
- The regions might be sets of border pixels grouped in to such structures as line segments and circular arc segments
- Two approaches to partitioning an image into regions

Region-base segmentation

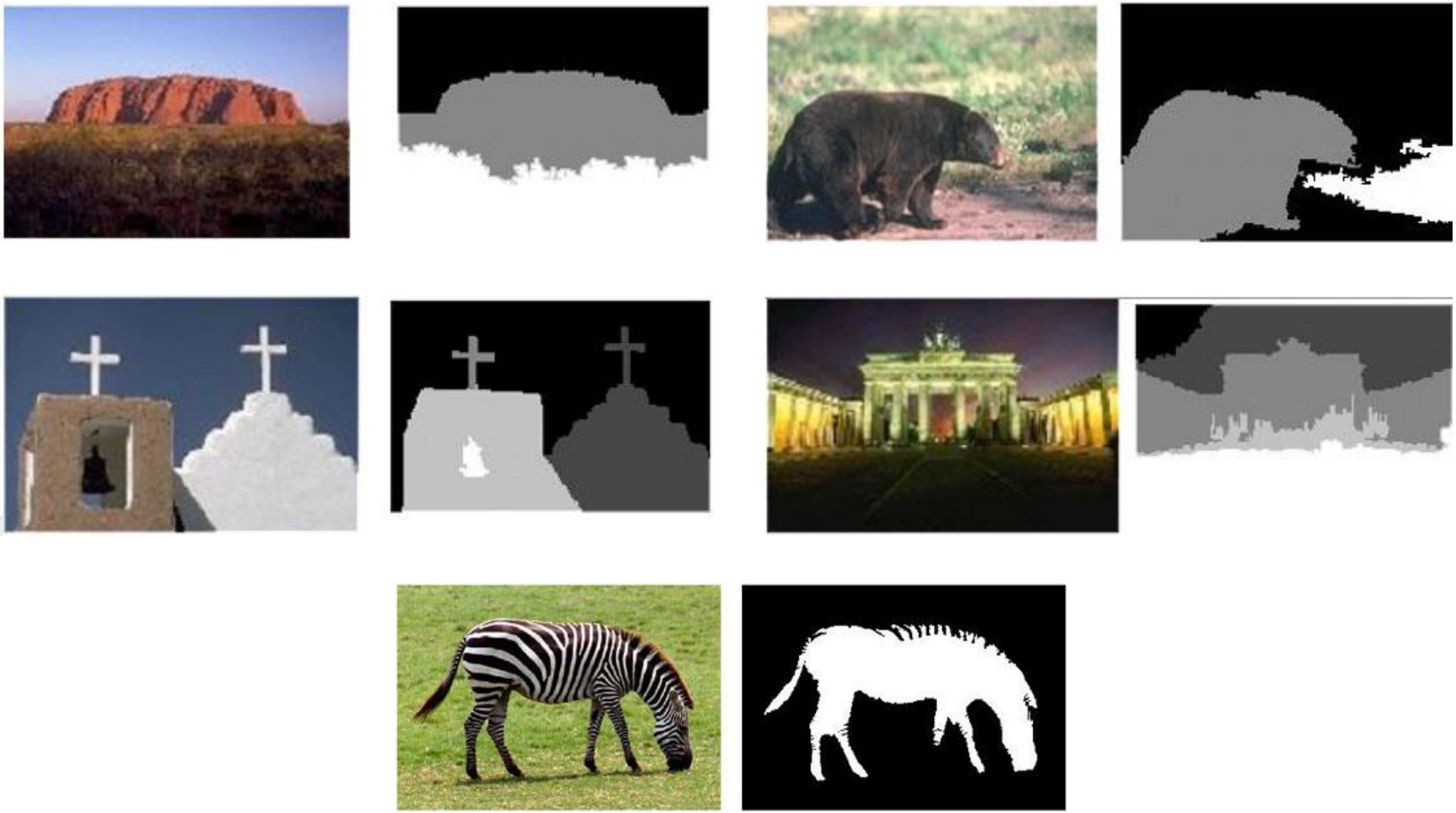
Assign pixels to regions using some similarity criterion.

- Value similarity
  - Ex. Intensity
- Spatial proximity
  - Close to the other

Edge-based segmentation

Use boundaries of region to segment image

- Base on edge detection

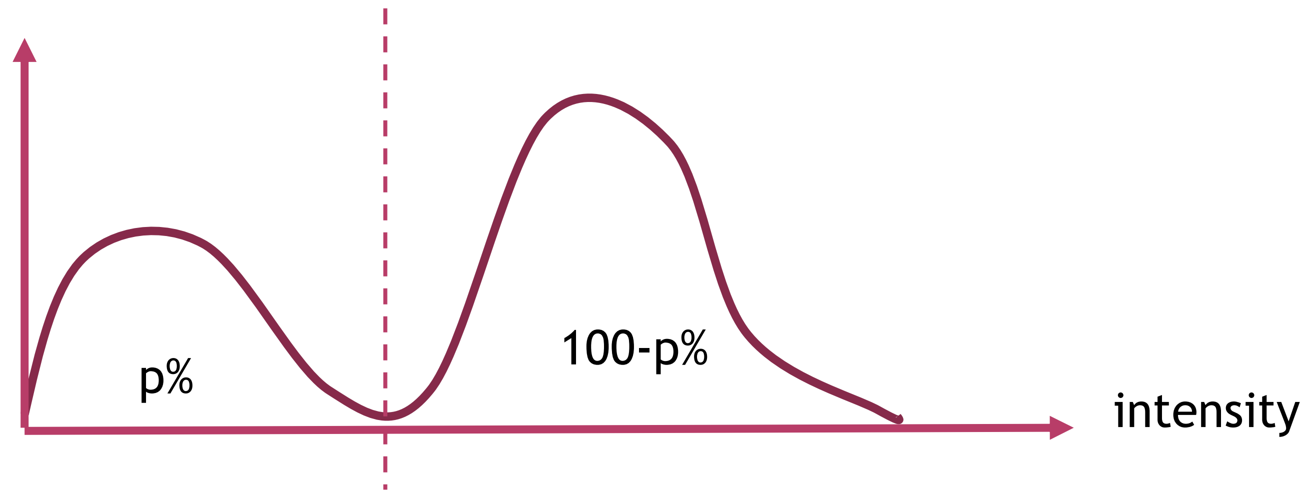


AUTOMATIC THRESHOLDING

- Automatic Thresholding
  - Thresholding should be selected by the system
  - The knowledge about the objects, applications, and environment should be used in segmentation rather than a fixed threshold value
    - Intensity characteristic of objects
    - Size of object
    - Fraction of an image occupied by the objects
    - Number of different types of objects

P-TILE METHOD

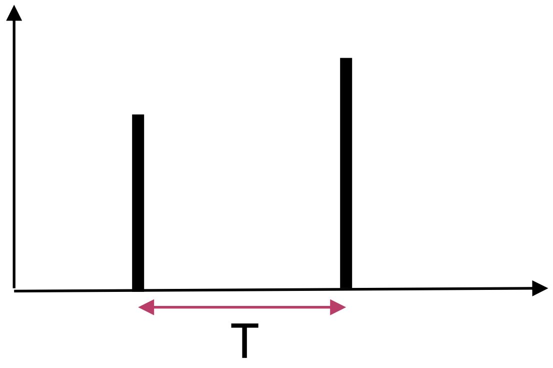
- Use knowledge about the area or size of the desired object to threshold an image
- Suppose that object occupies about p% of the image area then threshold the p% of pixels to object
- This method is very limited in use



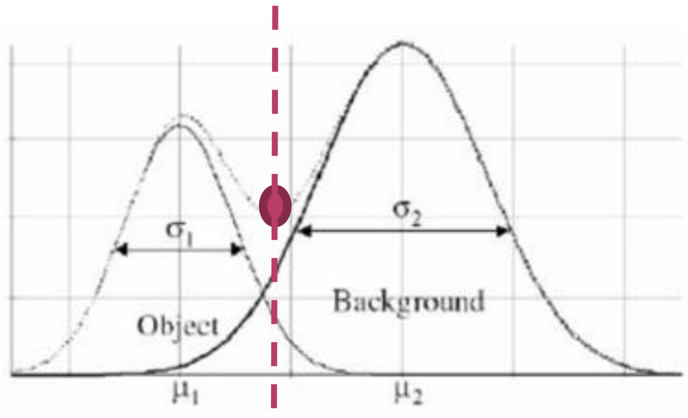
p% of pixels have intensity below threshold

MODE METHOD

- The object has a different intensity value to the background
- The intensity values are drawn from two normal distributions
- If the standard deviations are zero, there will be two spikes in histogram, and the threshold can be placed between them
- Detect peaks and valleys in the histogram, the threshold set to the valley intensity



Ideal image, SD = 0

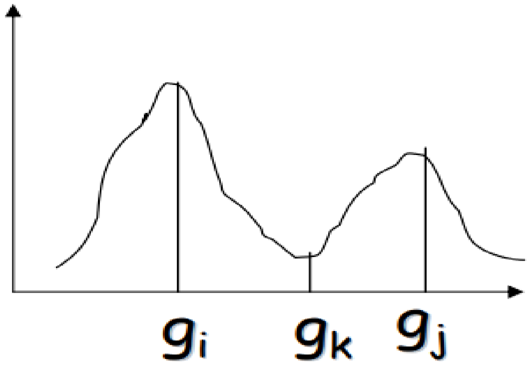


T

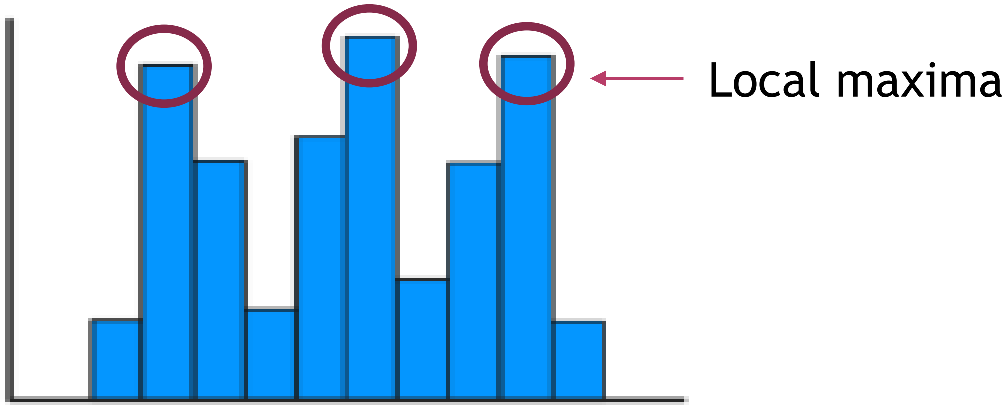
Peakiness Detection

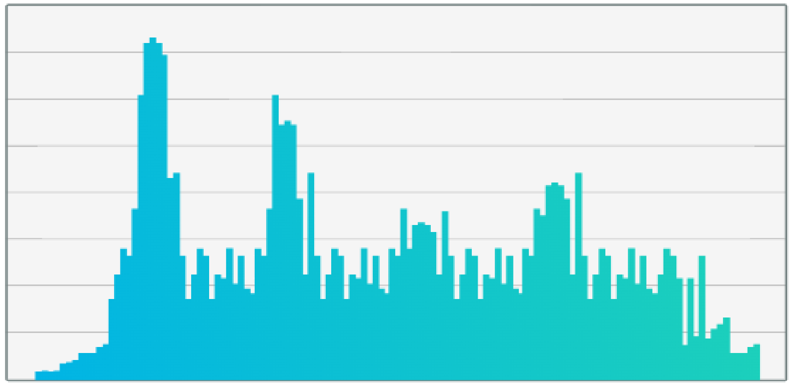
- Find the two HIGHEST **LOCAL MAXIMA** at a **MINIMUM DISTANCE APART**: g_i and g_j
- Find the lowest point between them: g_k
- Measure "peakiness":

$$peakiness = \frac{\min\left(H(g_i),\, H(g_j)\right)}{H(g_k)}$$

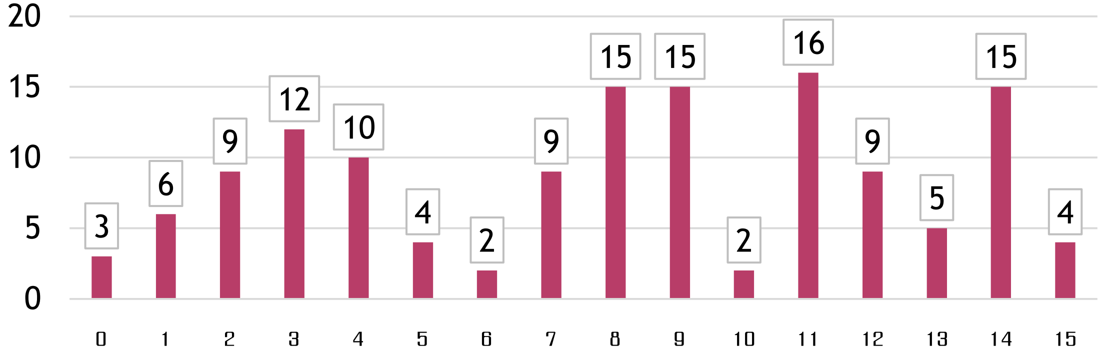


- Use a combination (g_i, g_j, g_k) with the highest peakiness to threshold the image





- Ex : find threshold value of mode method for the following histogram. Let minimum distance = 3



- For each combination of (g_i, g_j)

$$peakiness = \frac{\min\left(H(g_i),\, H(g_j)\right)}{H(g_k)}$$

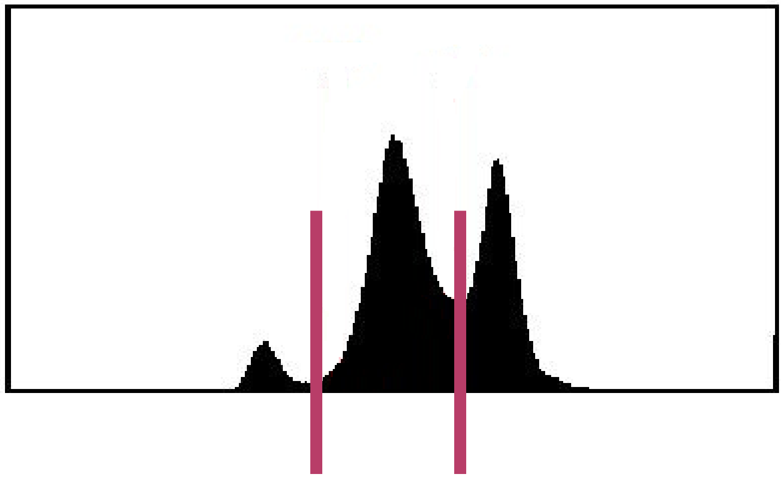
g_i	g_j	distance	g_k	peakiness

- find local maxima position
  - Local maxima = { g_3, }

Maximum peakiness =

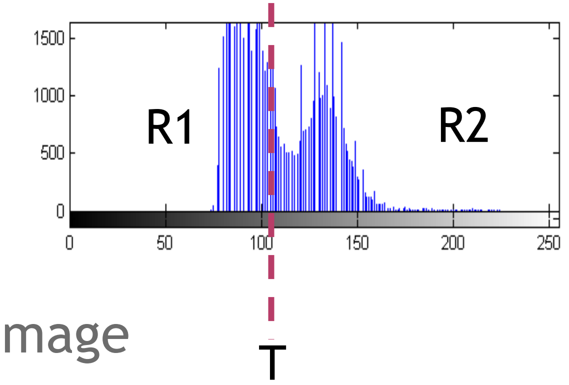
Threshold =

- The mode method can be applied to the image containing many objects with different gray values



ITERATIVE THRESHOLD SELECTION

- Start with approximating the threshold, then refine this threshold later

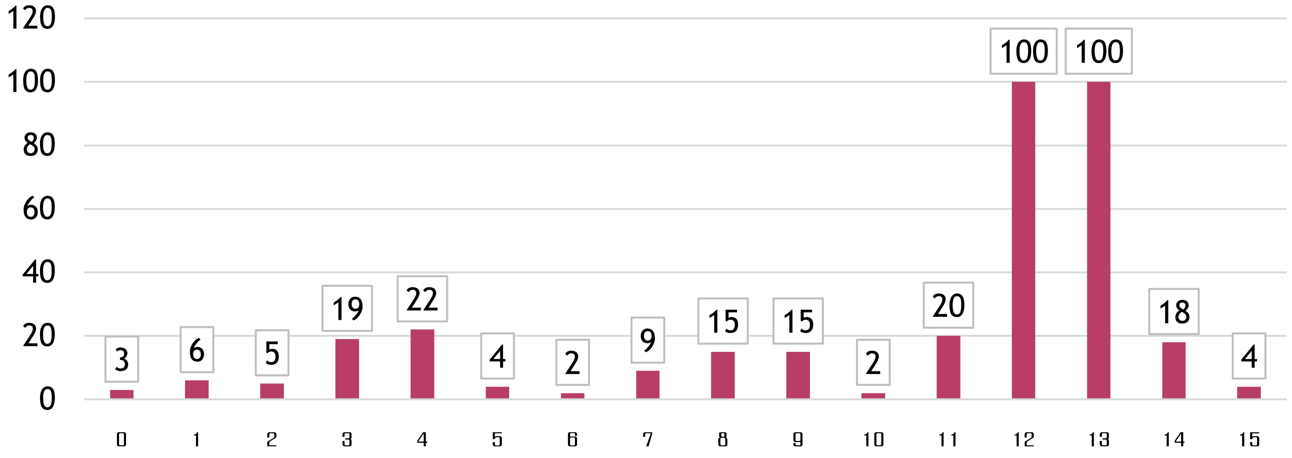


- Algorithm
  - Step1:Select initial threshold T -> average intensity of the image
  - Step2:Partition image into R1 and R2 using threshold T
  - Step3:Calculate the mean gray value μ_1 and μ_2 of R1 and R2
  - Step4:Compute new threshold

$$T = \frac{1}{2}(\mu_1 + \mu_2)$$

  - Step5:Repeat step 2-4 until mean value μ_1 and μ_2 do not change

- Ex



initial threshold T = μ_1 = $\mu_2 =$

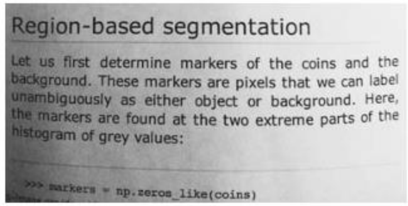
Iteration 1 $T_{new} = \frac{1}{2}(\mu_1 + \mu_2)$ = μ_1 = $\mu_2 =$

Iteration 2 $T_{new} = \frac{1}{2}(\mu_1 + \mu_2)$ = μ_1 = $\mu_2 =$

Iteration 3 $T_{new} = \frac{1}{2}(\mu_1 + \mu_2)$ = μ_1 = $\mu_2 =$

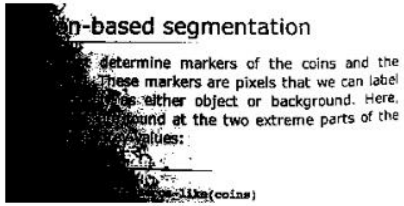
ADAPTIVE THRESHOLDING

- When the Illumination in a scene is uneven, the single threshold can't segment objects from the background correctly
- Due to shadow or due to the direction of illumination
- Partition the image into small regions and then analyse each subimage separately to threshold it.

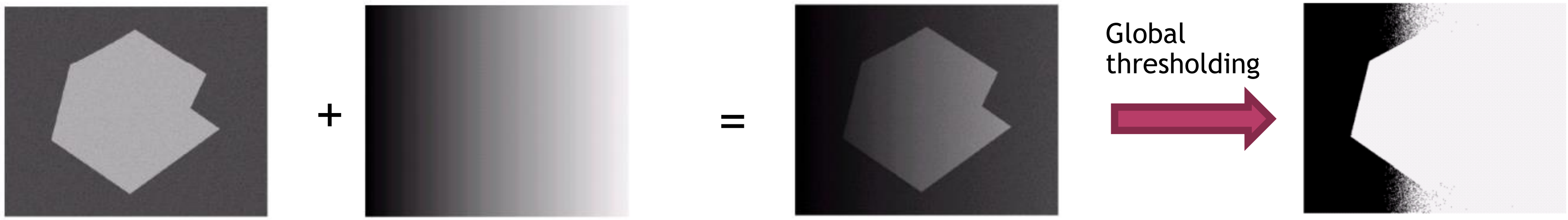

Region-based segmentation

Let us first determine markers of the coins and the background. These markers are pixels that we can label unambiguously as either object or background. Here, the markers are found at the two extreme parts of the histogram of grey values:

```
>>> markers = np.zeros_like(coins)
```

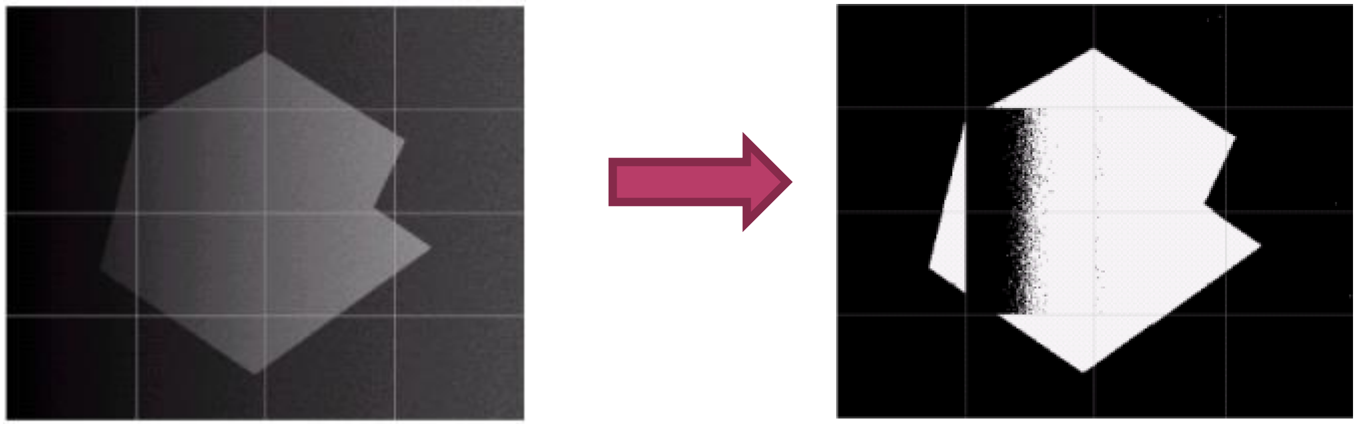

n-based segmentation

determine markers of the coins and the These markers are pixels that we can label es either object or background. Here, ound at the two extreme parts of the values:



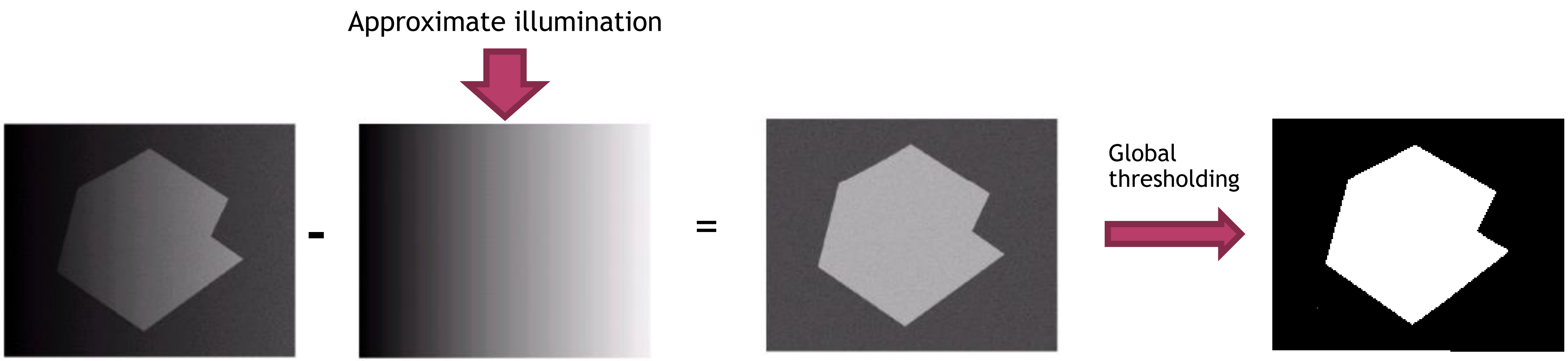
Uneven illumination

- Segment image to m x n subimage
- Select threshold T_{ij} for subimage (i,j)
- Union region of subimages together



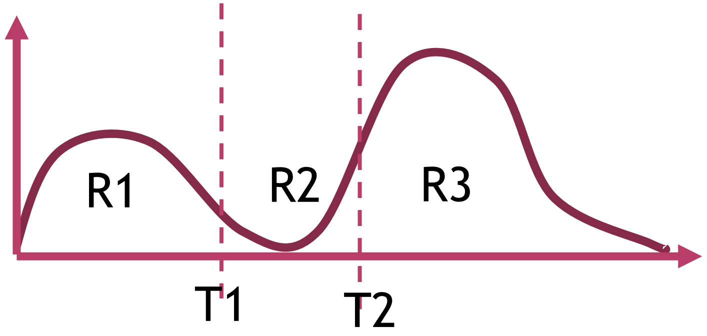
VARIABLE THRESHOLDING

- For uneven illumination, do approximate intensity value by simple function such as plane, biquadratic from gray value of the background
- Normalize the image with approximate illuminate function by subtraction

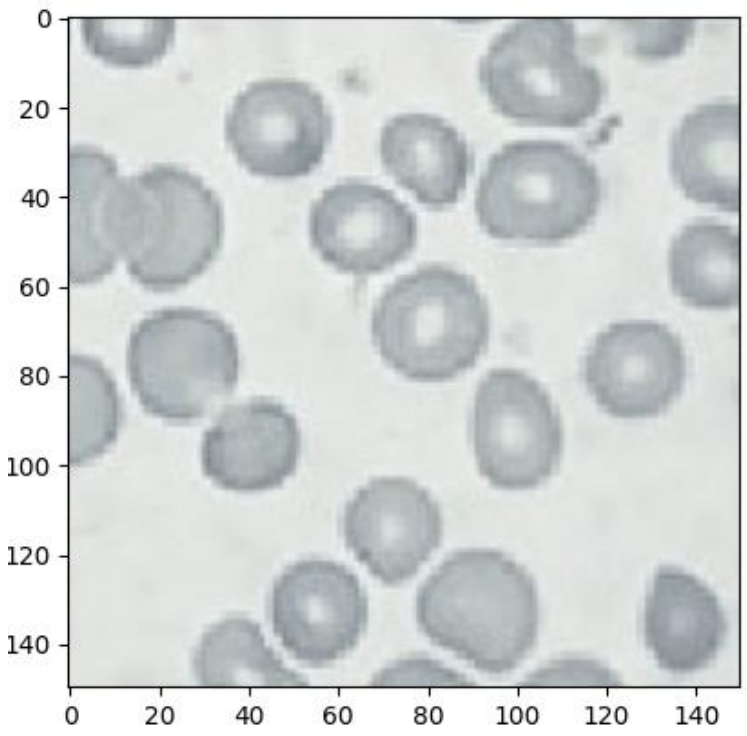


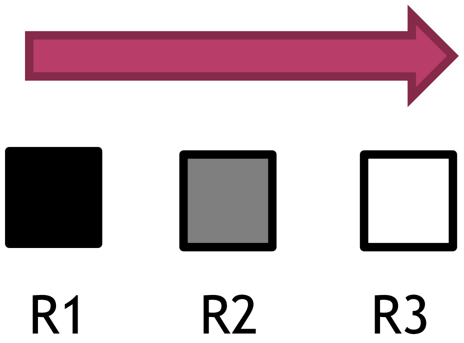
DOUBLE THRESHOLDING

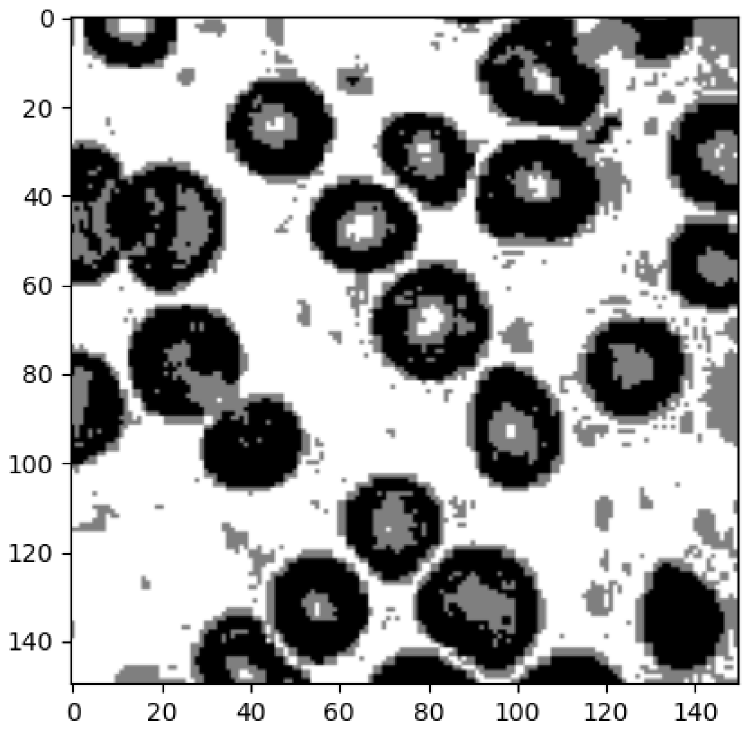
- This approach is to accept pixels if they have neighbor that is a core pixel of the object
- Algorithm
  - Step 1 : Select threshold T1 and T2 and partition image into 3 regions
    - R1 – all pixels with gray values below T1
    - R2 - all pixels with gray values between T1 and T2
    - R3 - all pixels with gray values above T2

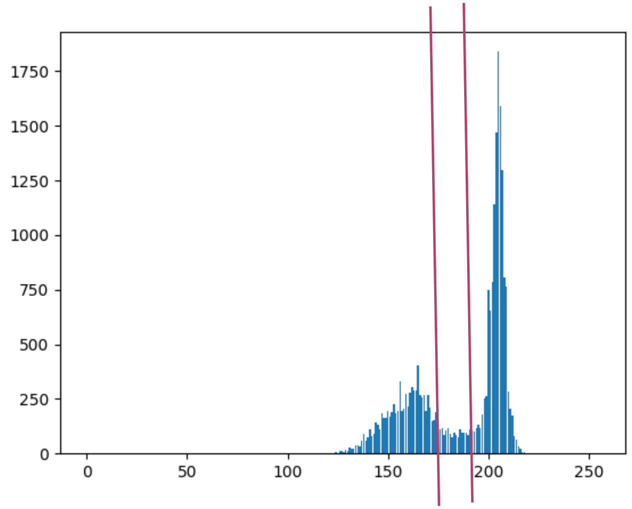


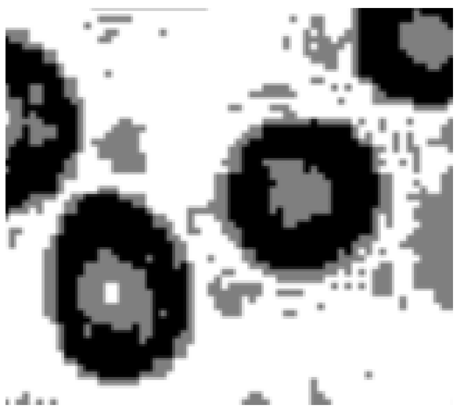
  - Step 2 : Visit each pixel in R2, if the pixel has neighbor in R1 then reassign pixel to R1
  - Step 3 : Repeat step2 until no pixels are reassigned
  - Step 4 : Reassign any pixels left in region R2 to R3











R2 pixels which connect to R1 become R1 otherwise R3

REGION REPRESENTATION

LABELING

- Use an array of the same size as the original image to indicate the region to which the pixel belong
- If element i,j has value r, then the corresponding pixel in image belong to region r

	1		1	1	1		
	1	1	1				
							2
	3	3	3	3			2
		3					2
	3	3	3				2
					2	2	2

Labeling

MASK / BITMAP

- Each region is associate with a binary image that indicate which pixels belong to the region
  - Allow pixel can be member of more than one region

Mask/ bitmap

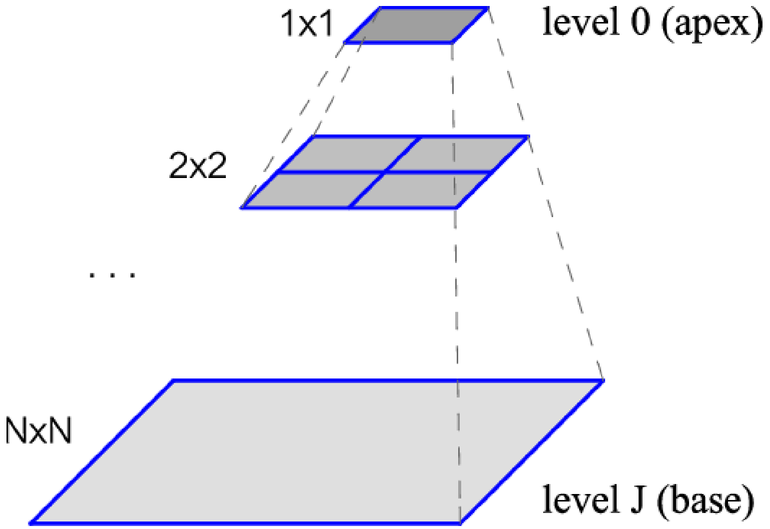
	1		1	1	1		
	1	1	1				

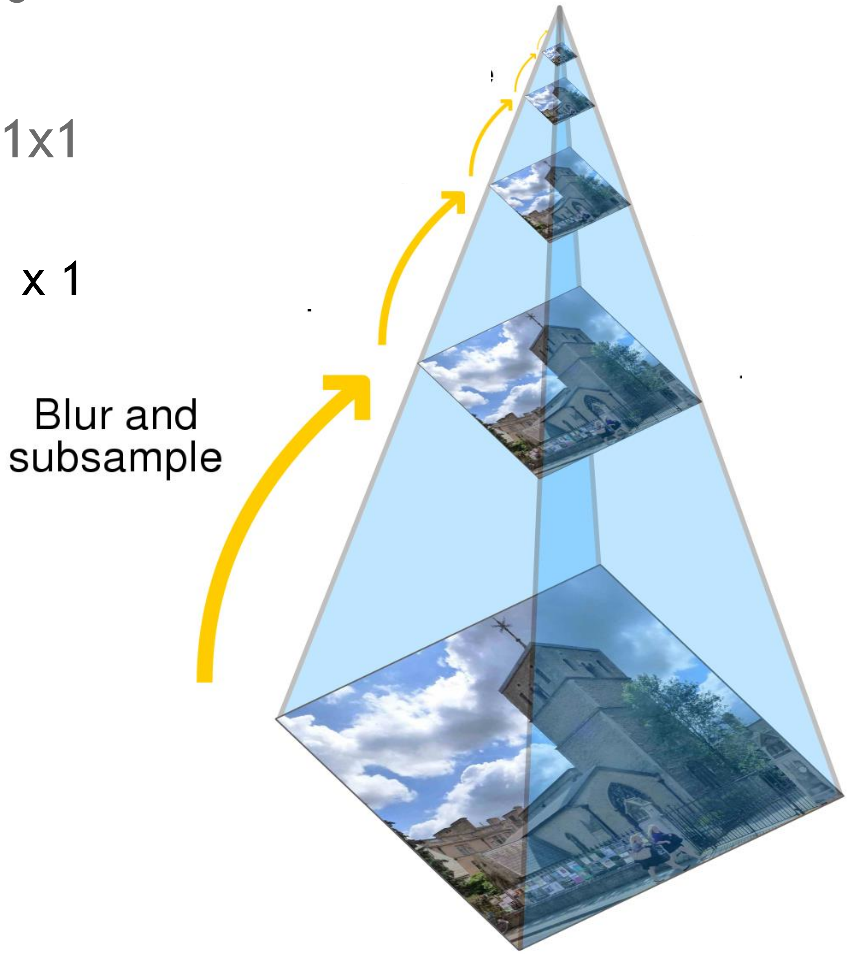
	1	1	1	1			
		1					
	1	1	1				

							1
							1
							1
							1
					1	1	1

PYRAMID

- Represent n x n image and k reduce version of image where n is power of 2
- Whole image represent in a single pixel at level 0
- The bottom level is the original image
- The list of image are n/2 x n/2 , n/4 x n/4, …., 1x1
  - Ex lv 6 = 64 x 64, lv5 = 32 x 32, lv4 = 16 x 16,
    lv3 = 8 x 8, lv2 = 4 x 4, lv1 = 2 x 2, lv0 = 1 x 1



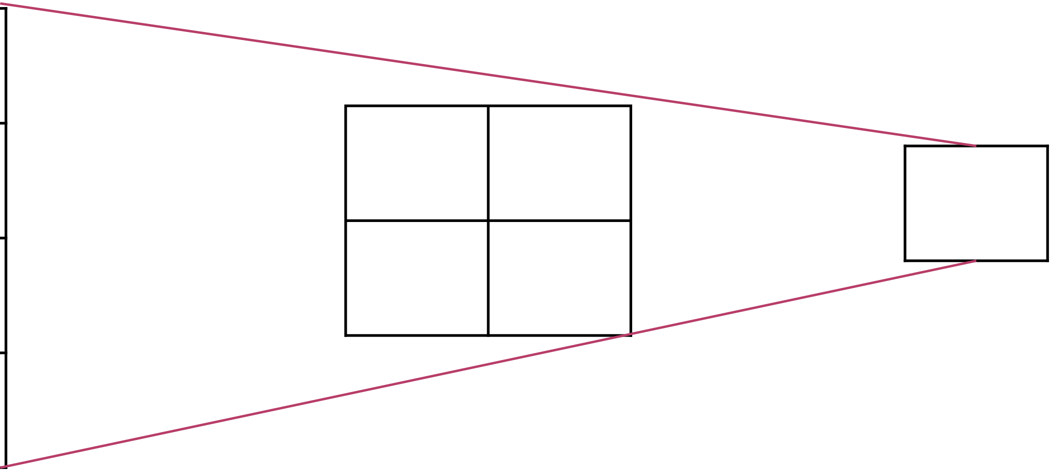


Level 2

120	100	150	128
145	155	160	45
80	200	200	33
90	170	160	59

Level 1

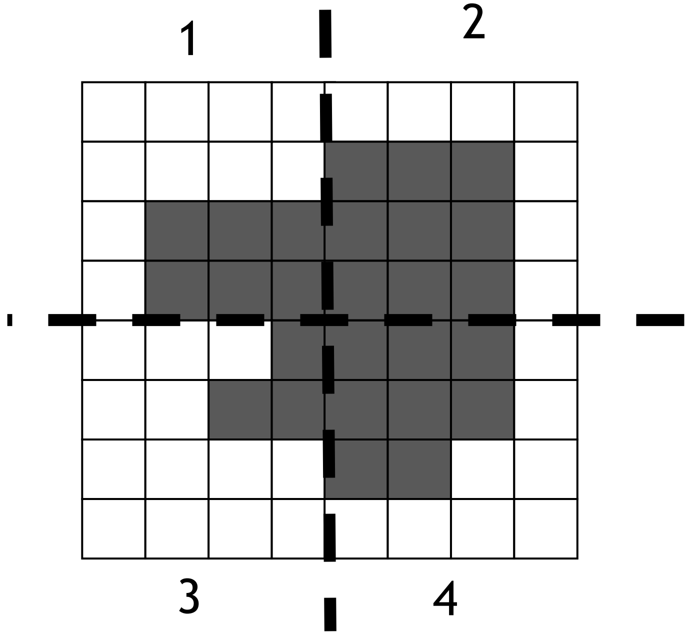
Level 0

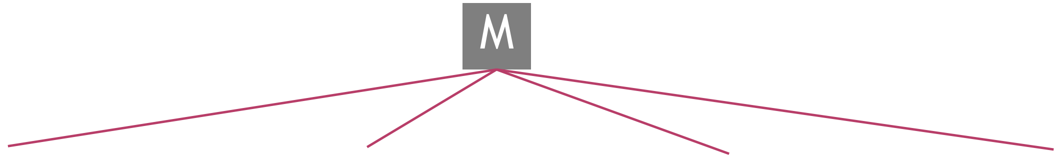


QUAD TREE

- The extension of pyramids for binary image
- Obtained by recursive splitting of an image into for sub-region of identical size
- Each node represented a square region in the image
- Each node have one of three labels Full, Empty, mixed

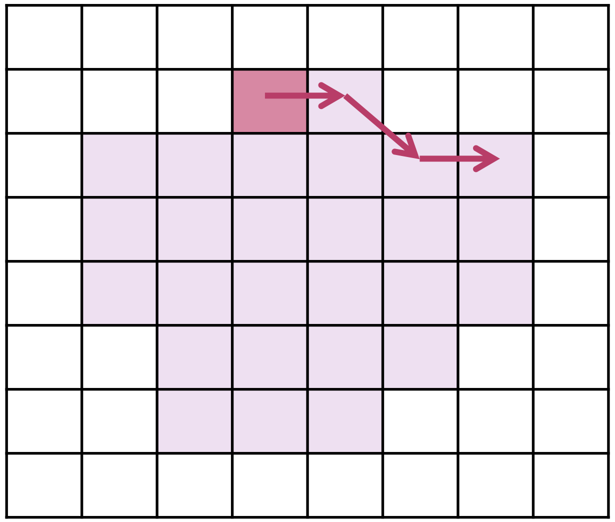
- Full : every pixels of square region it represents is a pixel of the region of interest
- Empty: there is no intersection between the square region it represents and the region of interest
- Mixed : some pixels of square region are pixels of region of interesting and some are not : split into for sub-region

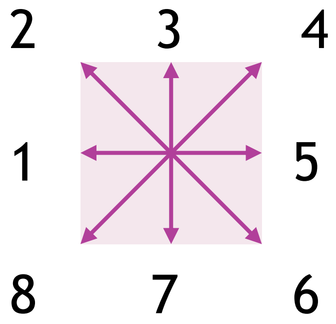




BOUNDARY CODING

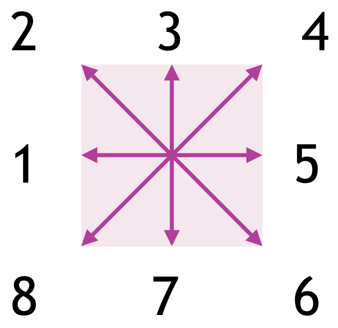
- Regions can also be represented by their boundary in a data structure in stead of an image
- The freeman chain code encodes information from list of edge point along contour
- Direction quantize to 8 directions





Encoding : 5 6 5 ……..

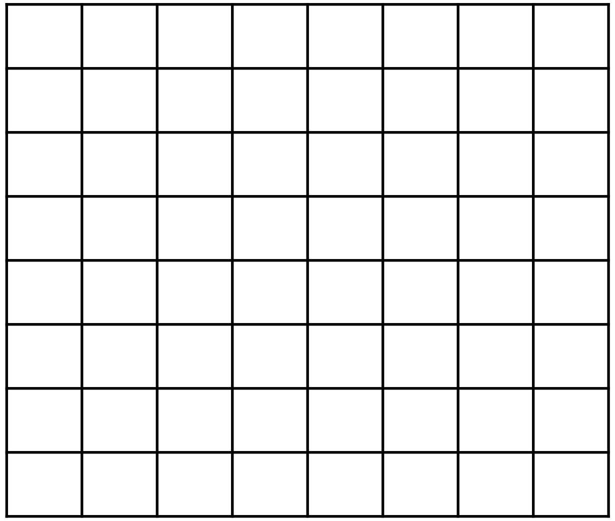
- Chain code can rotate by nx45 degree by adding n mod 8 to original code



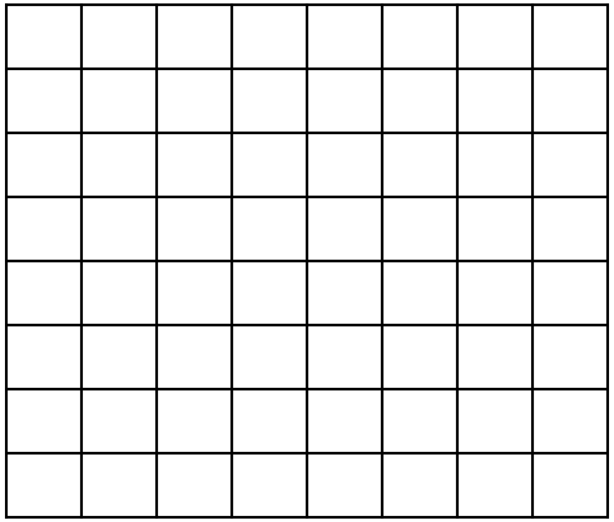
Original region encoding = 5 6 5 …

Rotation 45 degree encoding =

Rotation 135 degree encoding =



45 degree rotation



135 degree rotation

PROPERTY TABLE

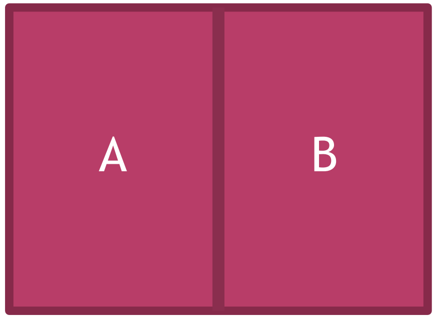
- Property table represent region by its properties rather than its pixels
- It's a table which has a row for each region in the image and a column for each property such as size, shape, intensity, color, texture.
- Property tables can be augmented to include or point to the chain code encoding or quad tree representation of region

region	area	width	height	hole	Chain code
A	26	8	5	1	1302435324
B	32	15	5	0	3543002352

- Some common property
  - Centroid
  - Moment
  - Euler number
  - Mean intensity
  - Variance intensity
  - Width
  - Height
  - Chain code
  - Quadtree

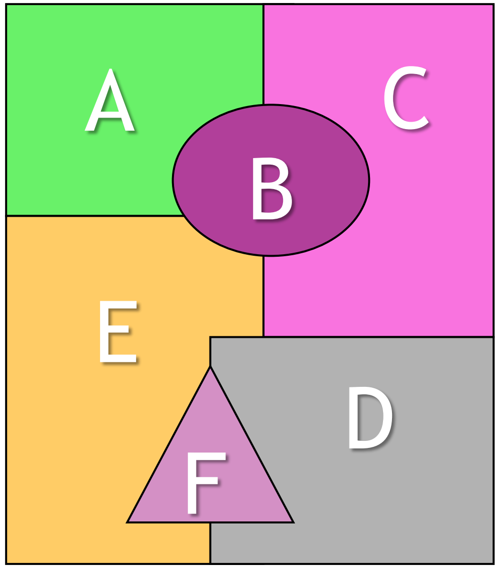
REGION ADJACENCY GRAPH (RAG)

- A region adjacency graph is used to represent regions and relationship among them in an image
- The emphasis is on the partitions of an image in the form of regions and the characteristics of each partition
- The nodes are used to represent region, and arcs between nodes represent a common boundary between regions
- Properties of regions may be stored in the node data structure



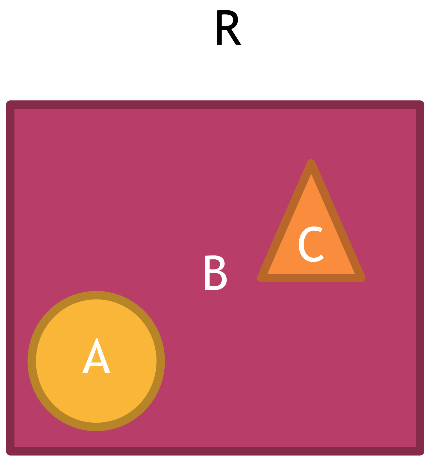
Region A connect to region B

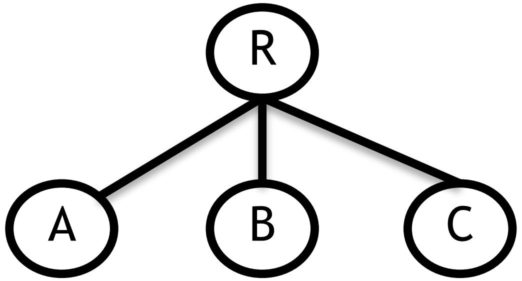
RAG



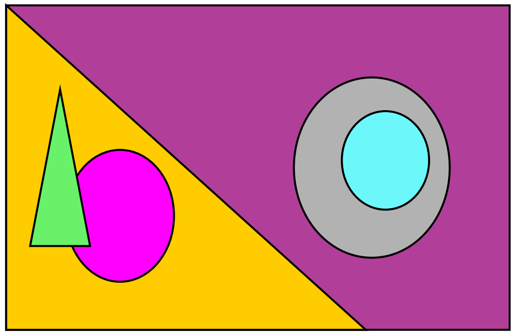
PICTURE TREE

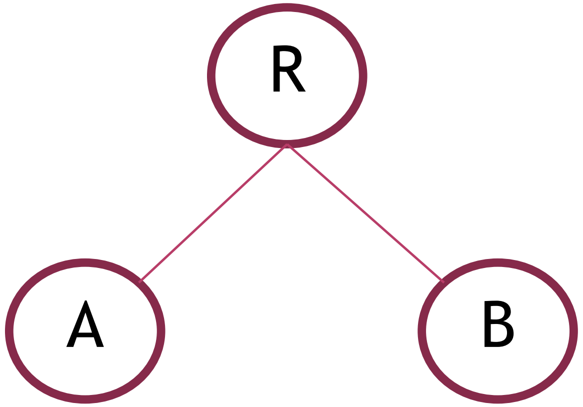
- The picture tree emphasizes the inclusion of a region with in another region as nesting region
- produced by recursively splitting an image into component parts
- Splitting stops when a region with constant characteristic has been reached

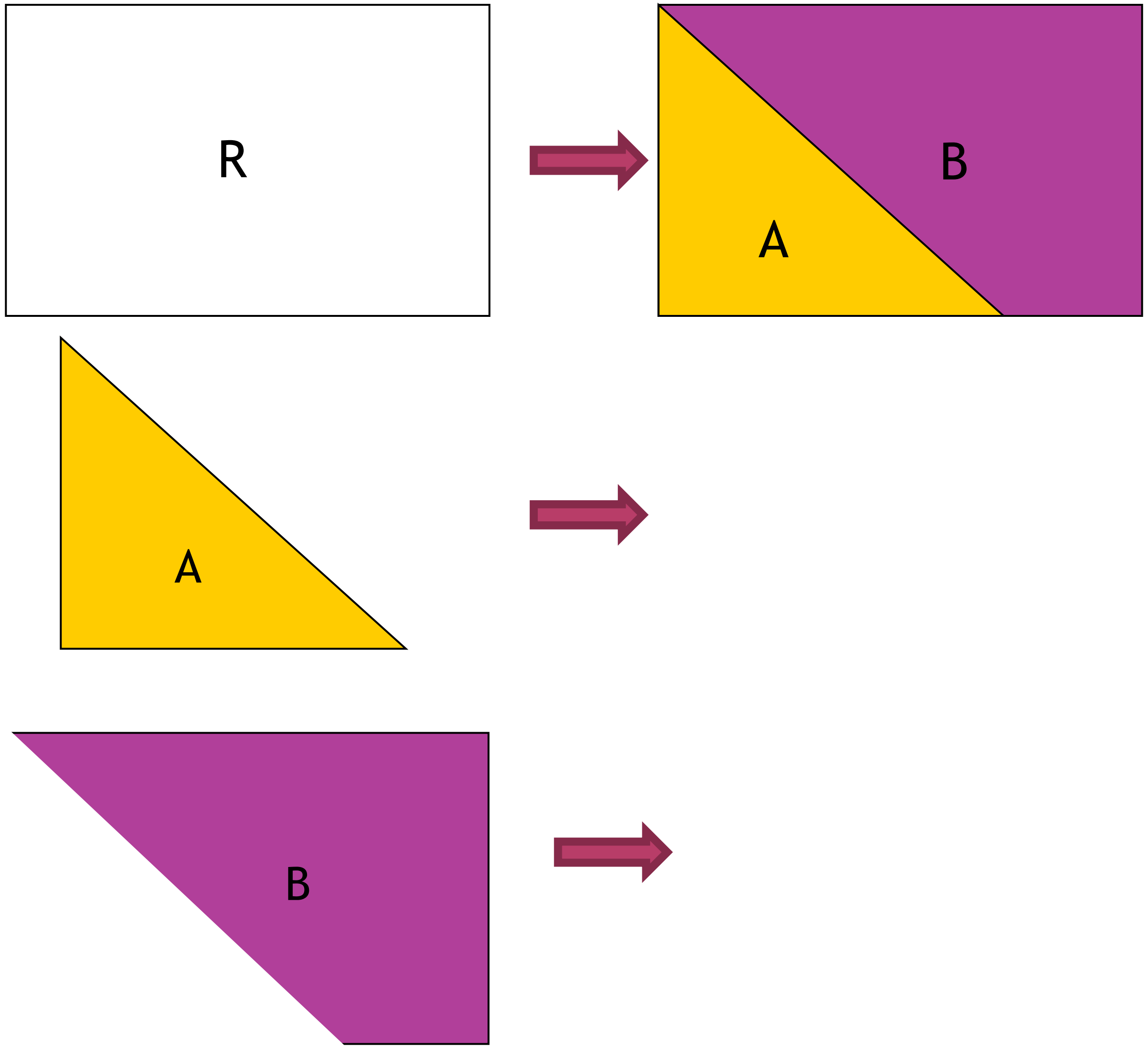




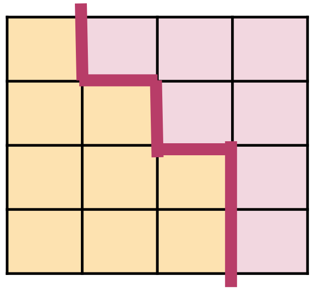
Region R compose of region A B and C




R
A
B

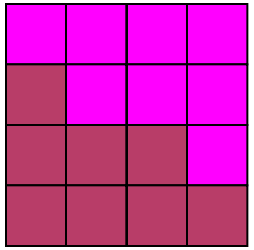

R
A
B
A
B

SUPER GRID

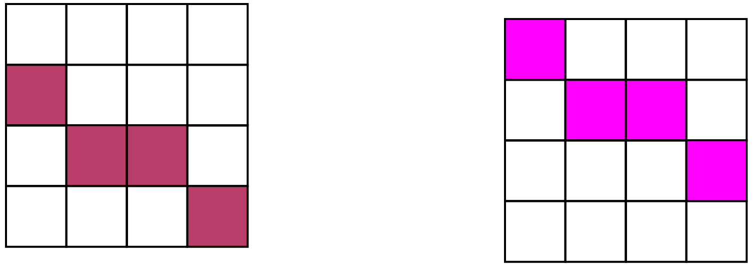


- The representation of boundary in an image array should be located between pixels of two adjacent regions
- This dilemma is solved by introducing a super grid on the image grid
- If the original image is NxN, then super grid is (2N+1)x(2N+1)
- Each pixel is surrounded by eight nonpixel points
- Nonpixel points are used to indicate whether or not there is a boundary between two pixels, and in what direction the boundary runs

N x N

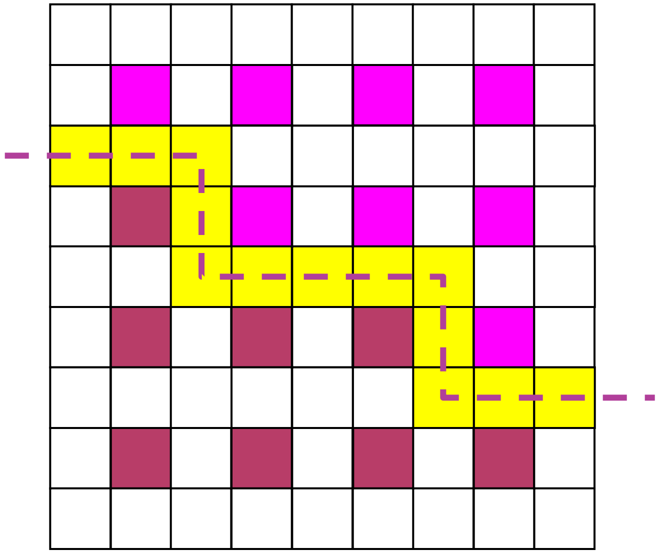


Original image



Traditional boundary representation

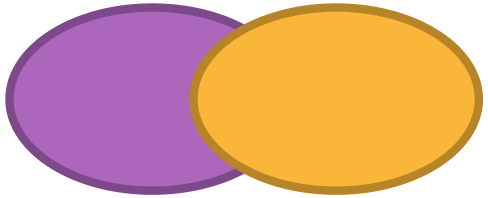
(2N+1) x (2N+1)



Super grid representation the boundary is now between the two regions

SPLIT AND MERGE

- Intensity-based segmentation usually results in too many regions
- The regions may need to refined or reform
- Automatic refinement is done by a combination of split and merge operation



Merge

- Eliminate false boundary from adjacent region that belong to the same object

Split

- Add missing boundary to the region that contain parts of difference objects

- Some approach for refinement may be compose of image intensity and other domain independent characteristic of region
- example
  - Merge adjacent regions with similar characteristic
  - Remove questionable edges
  - Use topological properties of the regions
  - Use shape information about object in the scene
  - Use semantic information about the scene

REGION MERGING

- Combine the regions that are considered similar
- The high level of merge algorithm as follow

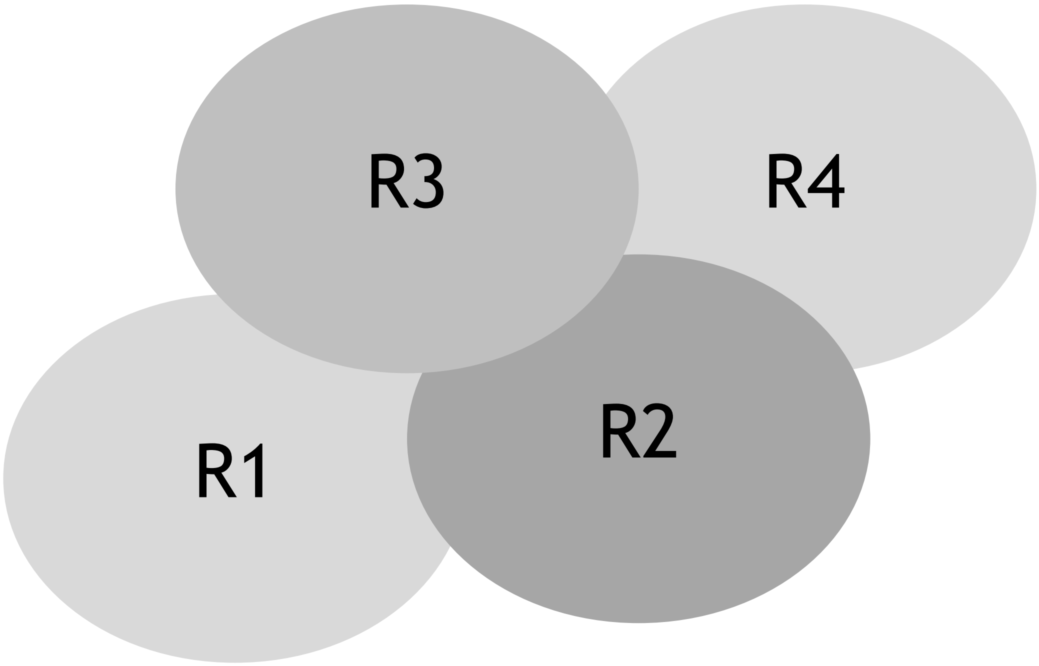
  - Step1 : Form initial regions using thresholding and labeling
  - Step 2 : Prepare a region adjacency graph (RAG)
  - Step 3 : For each region
    - Consider the similarity of its adjacent region
    - If the regions are similar then merge region and update RAG
  - Step 4 : Repeat step 3 consider region until no regions are merged

SIMILARITY OF REGION

- Two approach to just the similarity

Base on intensity	Base on the weakness of boundary
• Compare mean intensity • Compare the probability	• Combine two region if the boundary between them is weak

- Region similarity base on mean intensity
  - If mean intensity of two region do not differ more than the predetermined value, the region are similar

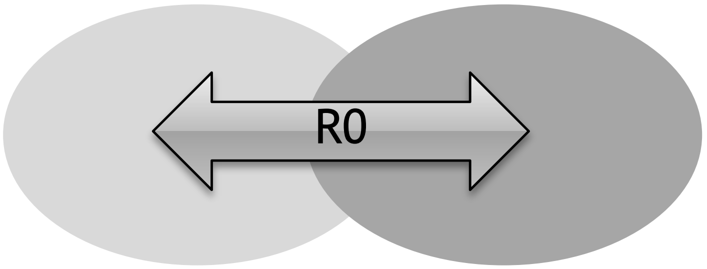


Find mean intensity of each region

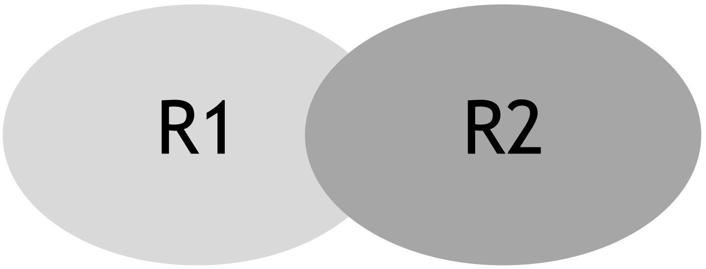
If $|\mu_1 - \mu_1| > T$
then merge region

- Region similarity base on probability
  - Merging region will have same statistic distribution of intensity
  - Use hypothesis testing

H0	H1
• The regions belong to the same object	• the regions belong to different object

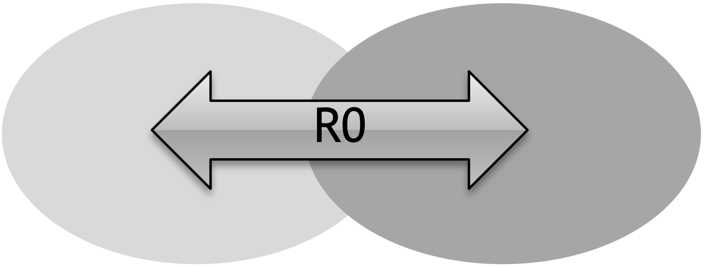


(μ_0, σ_0)



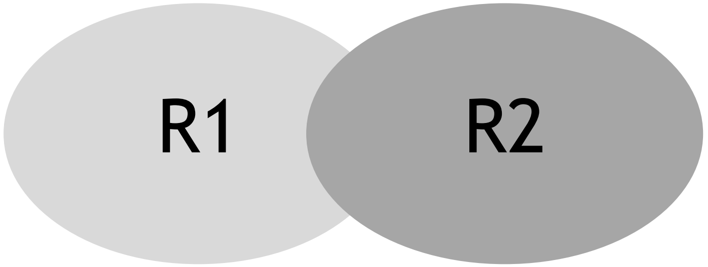
(μ_1, σ_1) (μ_2, σ_2)

m1+ m2 piexels



(μ_0, σ_0)

m1 piexels m2 piexels



(μ_1, σ_1) (μ_2, σ_2)

- Likelihood ratio (L) – the ratio of the probability densities under two hypothesis

$$L = \frac{{\sigma_0}^{m1+m2}}{{\sigma_1}^{m1} \cdot {\sigma_2}^{m2}}$$

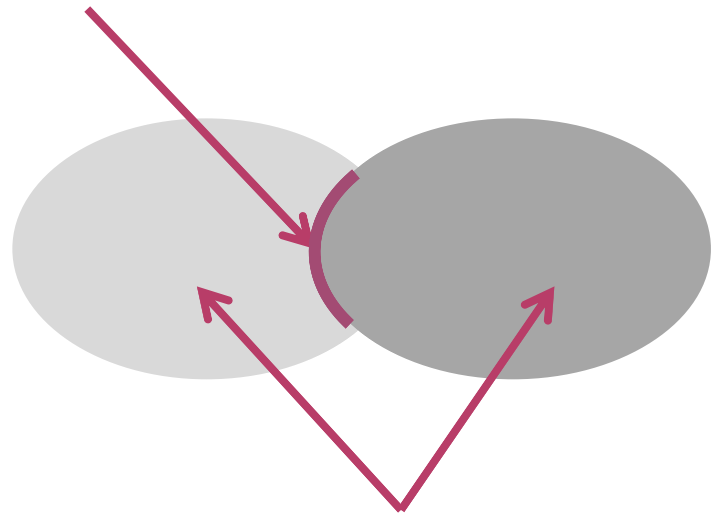
$$\mu = \frac{1}{n}\sum_{i=1}^{n} g_i$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (g_i - \mu)^2$$

$g_i = \text{intensity of pixel } i\text{th}$

If $L < threshold$ then merge region

- Boundary weakness
  - Combine two regions when boundary between them is weak
  - Weak boundary
    - Intensities in either side differ by less than an amount T
    - Determine the strength of the edgeness value of an edge point that is on the boundary separate two region
    - Length of the weak boundary

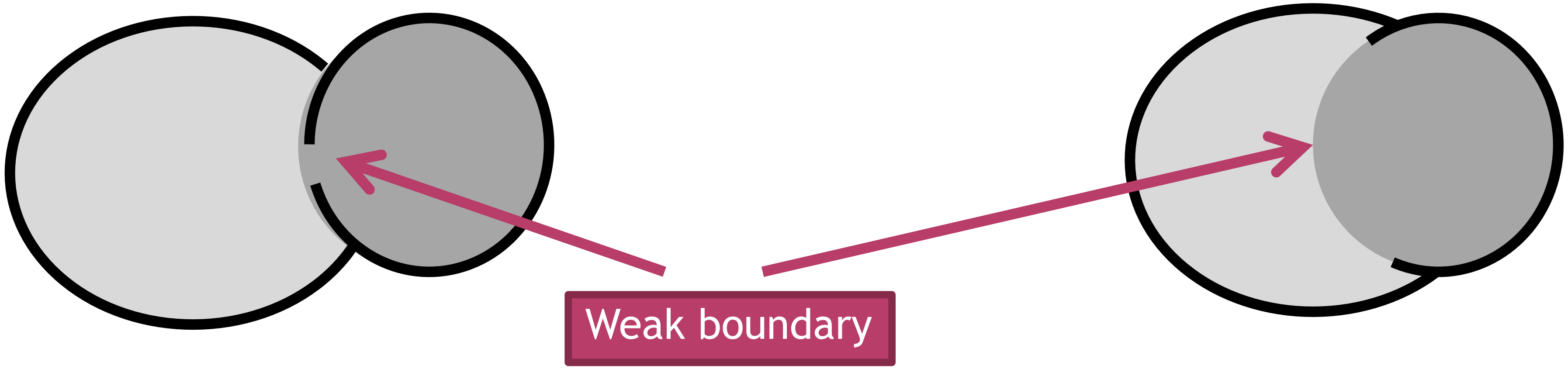


Weak boundary

- Approach 1 : remove weak boundary when ratio of the weak boundary to the minimum region perimeter > T

Merge adjacent region if $\frac{W}{S} > T$ Usually T = 0.5

W- length of weak part of the common boundary
S = min(S1,S2)



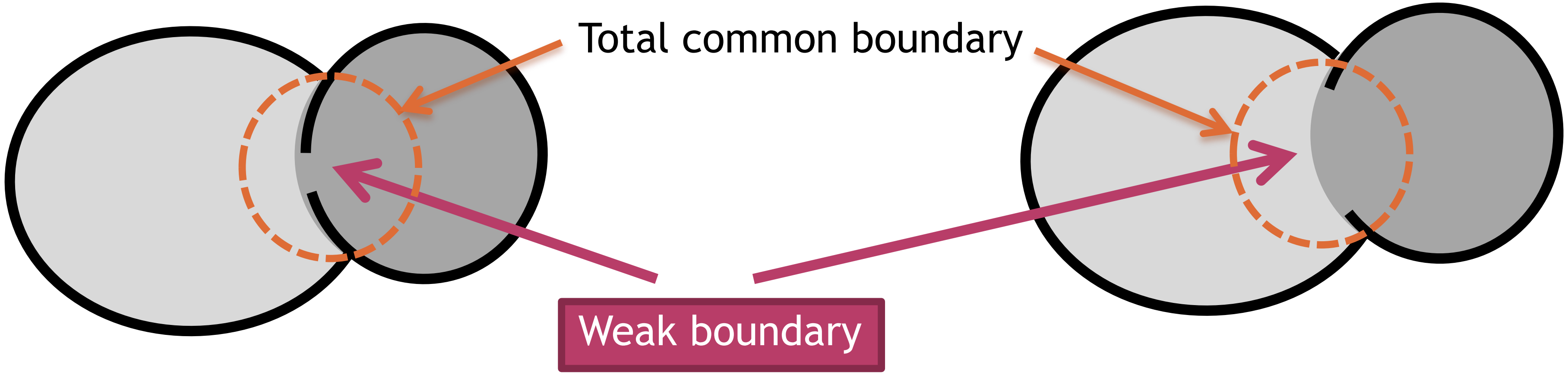
Not merge – weak boundary is very short when compare to the perimeter of smaller region

Merge – weak boundary is significant fraction of the perimeter of smaller region

- Weak boundary
  - Approach 2 : remove weak boundary when ratio of the weak boundary to the total common boundary > T

Merge adjacent region if $\frac{W}{S} > T$ Usually T = 0.75

W- length of weak part of the common boundary
S =total common boundary



Not merge – weak boundary is very short when compare to the perimeter of common boundary

Merge – weak boundary is significant fraction of the perimeter of common boundary