

Segmentation of Images

SEGMENTATION

If an image has been preprocessed appropriately to remove noise and artifacts, segmentation is often the key step in **interpreting the image**. Image segmentation is a process in which *regions or features sharing similar characteristics are identified and grouped together*.

Image segmentation may use **statistical classification, thresholding, edge detection, region detection, or any combination of these techniques**. The output of the segmentation step is usually a set of classified elements,

Segmentation techniques are either **region-based or edge-based**.

- **Region-based techniques** rely on common patterns in intensity values within a cluster of neighboring pixels. The cluster is referred to as the region, and the goal of the segmentation algorithm is to group regions according to their anatomical or functional roles.
- **Edge-based techniques** rely on discontinuities in image values between distinct regions, and the goal of the segmentation algorithm is to accurately demarcate the boundary separating these regions.

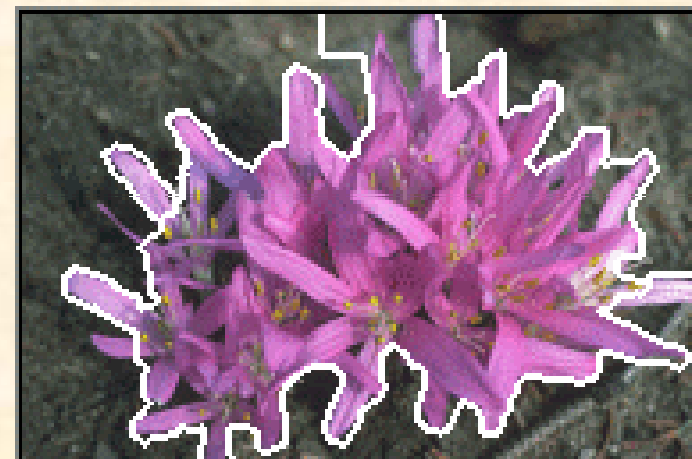
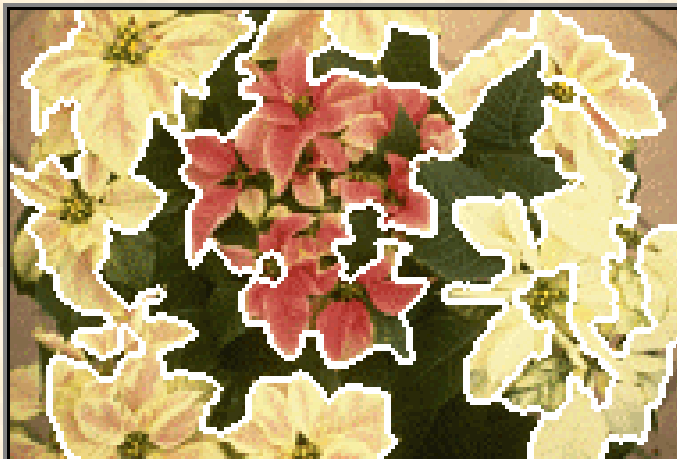
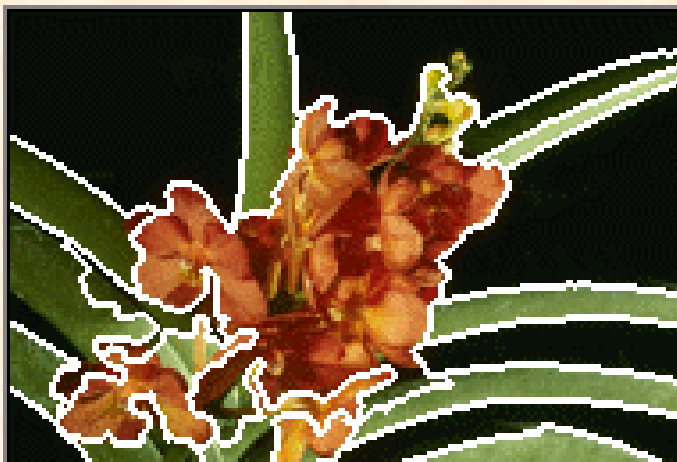
***Segmentation* is a process of extracting and representing information from an image is to group pixels together into regions of similarity.**

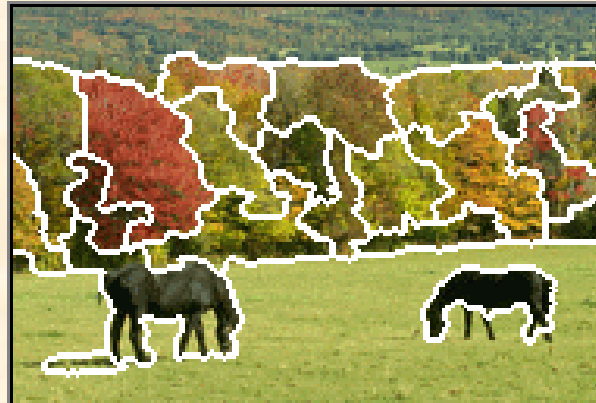
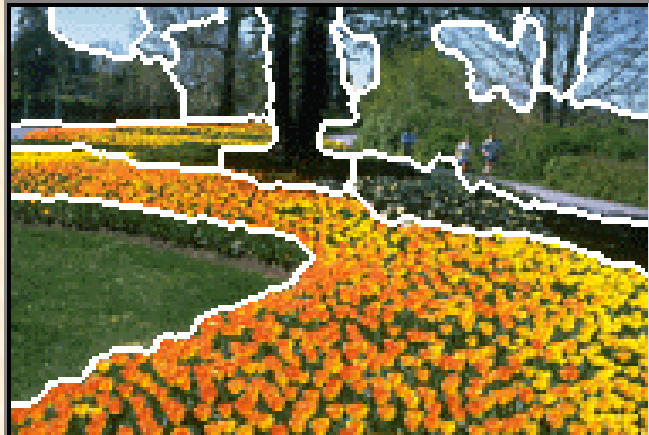
Region-based segmentation methods attempt to partition or group regions according to common image properties. These image properties consist of :

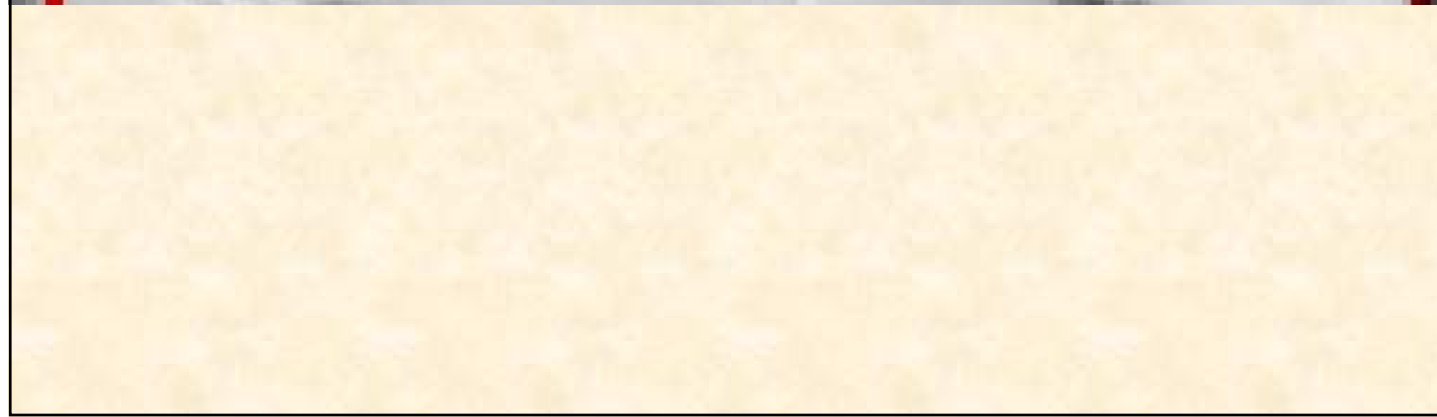
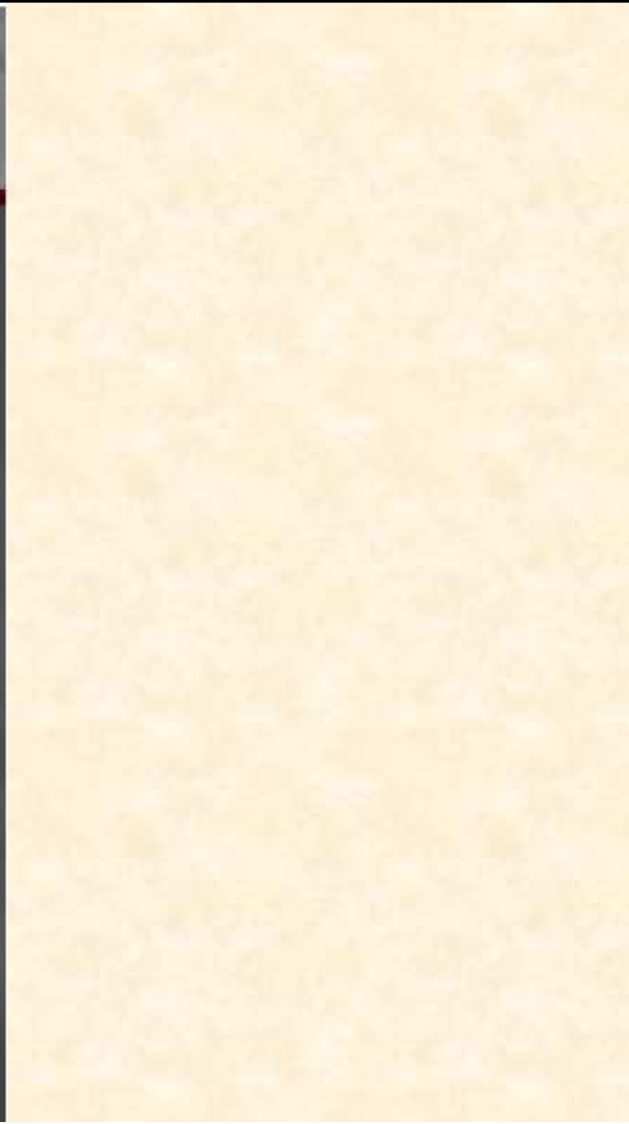
- **Intensity values from original images, or computed values based on an image operator**
- **Textures or patterns that are unique to each type of region**
- **Spectral profiles that provide multidimensional image data**

Elaborate systems may use a combination of these properties to segment images, while simpler systems may be restricted to a minimal set on properties depending of the type of data available.

Lets observe some examples from recent literature:







The problem of image Segmentation:

Decompose a given image into segments/regions/sub-areas/partitions/blobs, each containing similar pixels (or having similar statistical characteristics or similarity).

Target is to have regions of the image depicting the same object.

Semantics:

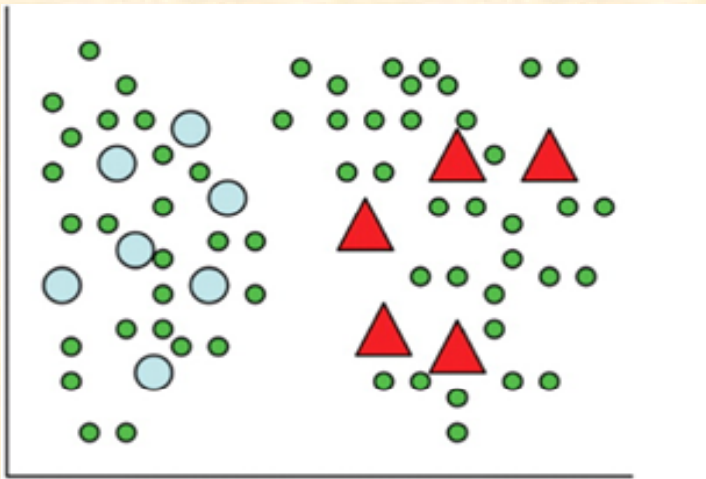
- How to get the idea of an object in the algorithm ?**
- How should we infer the objects from segments ??**

Segmentation problem is often posed or solved by pattern **classification or CLUSTERING (unsupervised).**

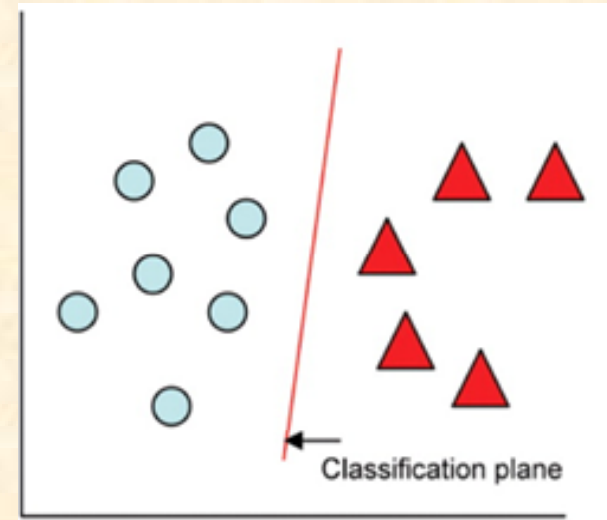
Are features from pixels from a particular region form a unique cluster or pattern ??

Segments must be connected regions assigned to the same cluster.

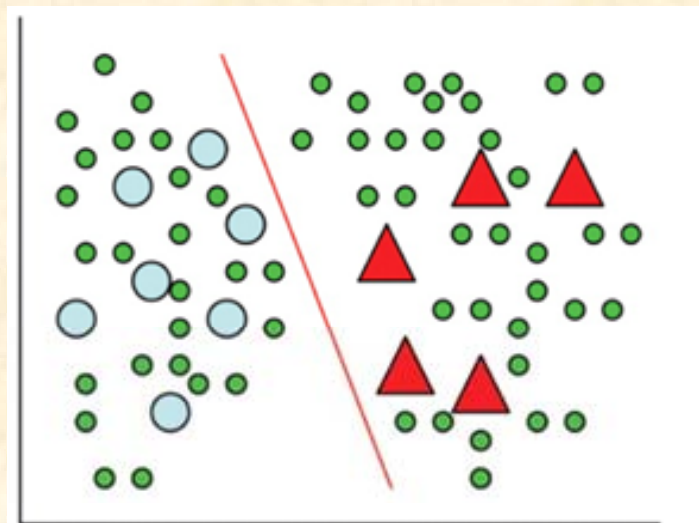
Clusters in Feature space



Labeled and Unlabeled Data

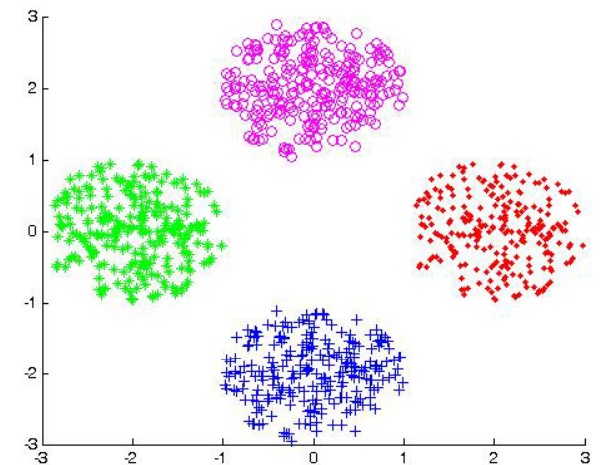
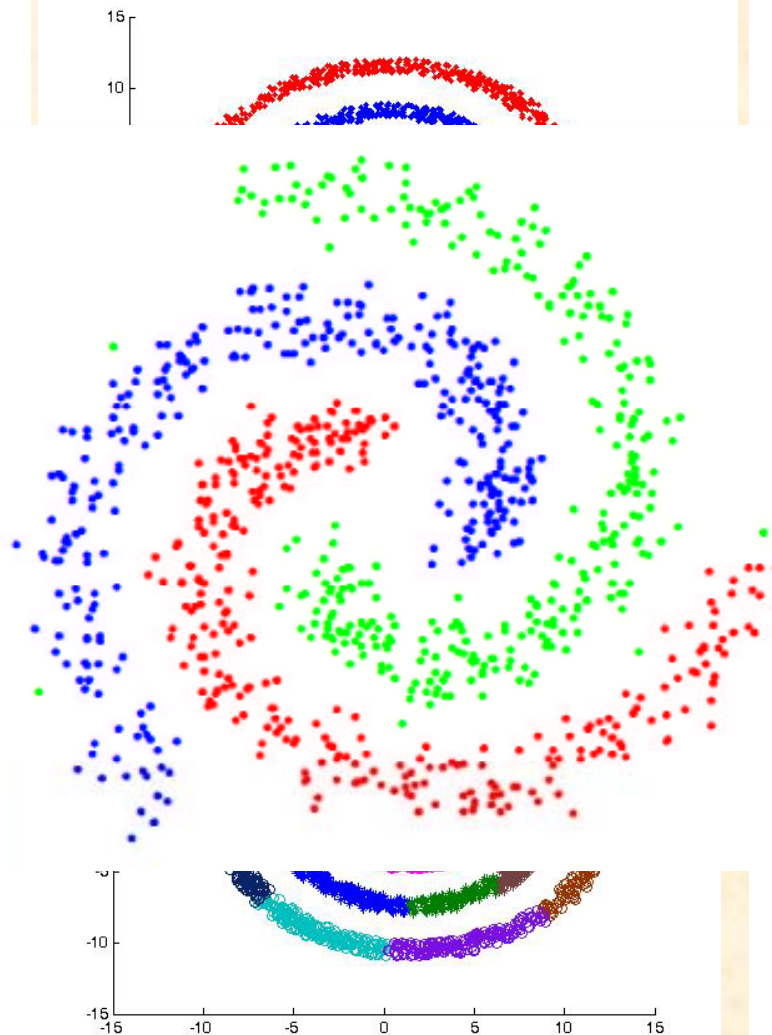
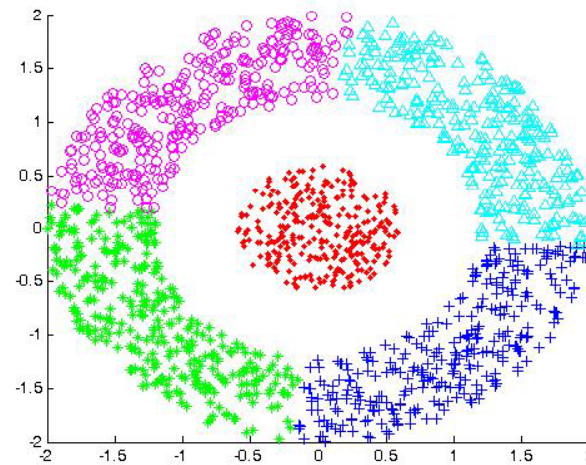
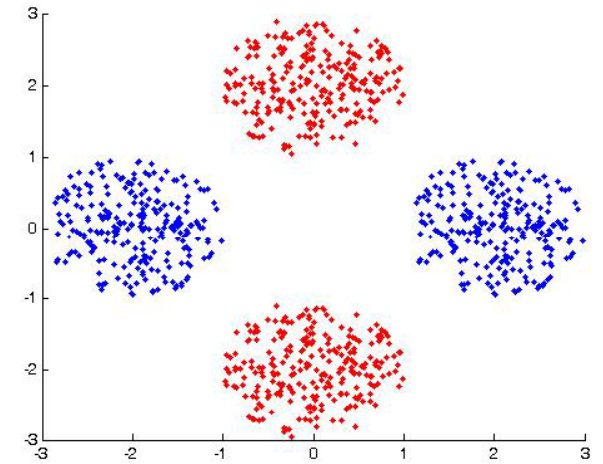
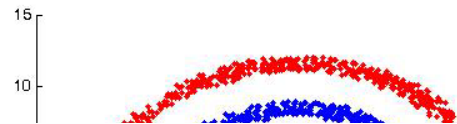
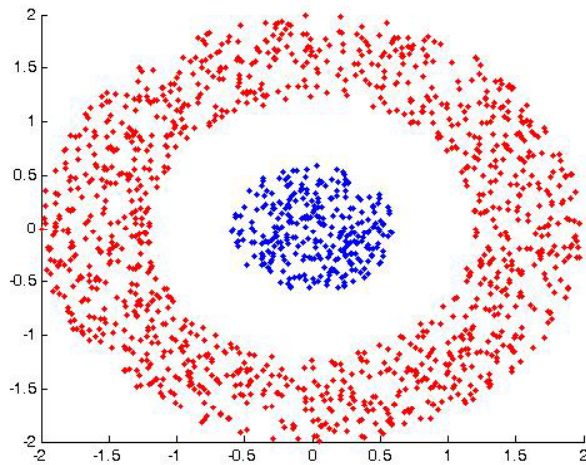


Supervised Learning



Semi-Supervised Learning

EXAMPLES of CLUSTERING



Purpose:

Segment an entire image R into smaller sub-images, R_i , $i=1,2,\dots,N$. which satisfy the following conditions:

$$R = \bigcup_{i=1}^N R_i; R_i \cap R_j = \Phi, i \neq j$$

$$H(R_i) = \text{True}; i = 1, 2, \dots, N;$$

When, R_i and R_j are adjacent: $H(R_i \cup R_j) = \text{False}, i \neq j;$

Typical algorithms of clustering data:

- **Agglomerative clustering**
- **K-means, K-medoids, DB-SCAN**
- **check PR literature for more (cluster validity index etc.)**

Categories of Image Segmentation Methods

- **Clustering Methods**
- **Histogram-Based Methods**
- **Edge Detection Methods**
- **Region Growing Methods**
- **Level Set Methods**
- **Graph Partitioning Methods**
- **Watershed Transformation**
- **Neural Networks Segmentation**
- **Multi-scale Segmentation**
- **Model based Segmentation/knowledge-based segmentation** - involve active shape and appearance models, active contours and deformable templates.
- **Semi-automatic Segmentation** - Techniques like Livewire or Intelligent Scissors are used in this kind of segmentation.

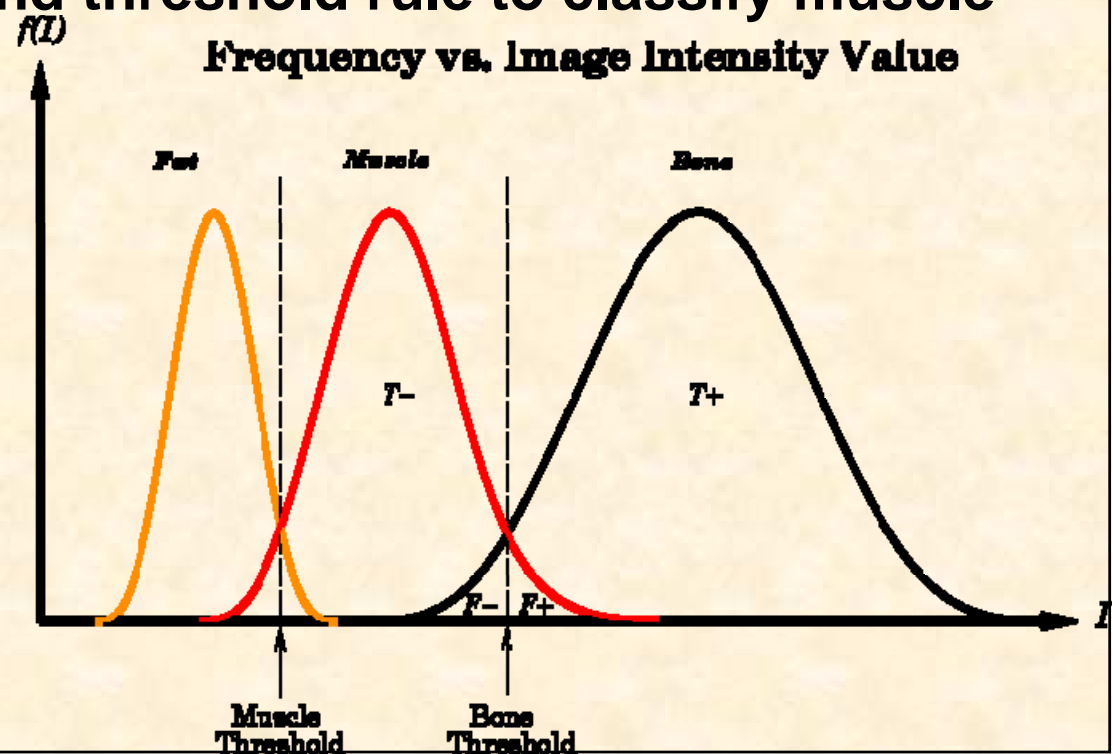
Thresholding is the simplest way to perform segmentation, and it is used extensively in many image processing applications. Thresholding is based on the notion that regions corresponding to different regions can be classified by using a range function applied to the intensity values of image pixels. The assumption is that different regions in an image will have a distinct frequency distribution and can be discriminated on the basis of the mean and standard deviation of each distribution (see Figure).

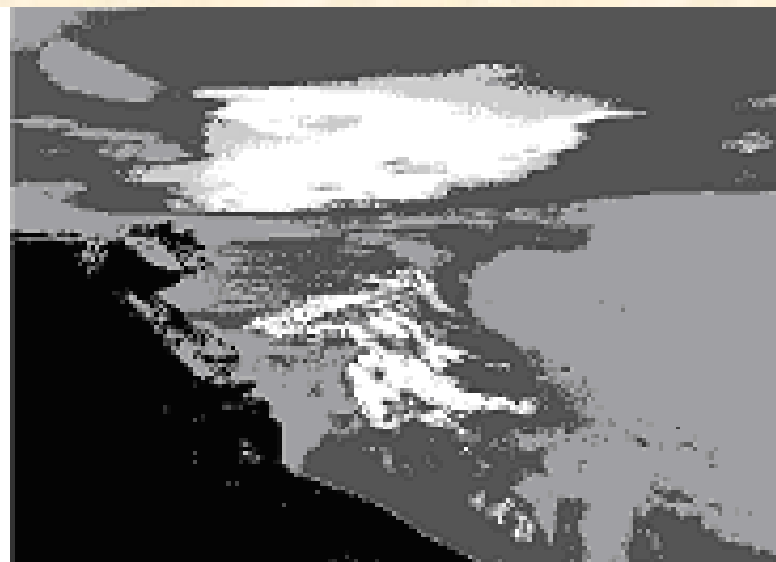
For example, given the histogram of a two-dimensional medical image $I(x,y)$, we can define a simple threshold rule to classify bony and fat tissues or a compound threshold rule to classify muscle tissue:

If, $I(x,y) > T_1 \Rightarrow$ Bony

If, $I(x,y) < T_0 \Rightarrow$ Fat

If, $T_0 < I(x,y) < T_1 \Rightarrow$ Muscle





Two examples of gray level thresholding based segmentation



**Typical segmentation output of a satellite image
using recursive multi-level thresholding method
with statistical features**

Read Otsu's method of multi-modal thresholding:

Limitations of thresholding:

- **The major drawback to threshold-based approaches is that they often lack the sensitivity and specificity needed for accurate classification.**
- **The problem gets severe in case of multi-modal histograms with no sharp or well-defined boundaries.**
- **It is often difficult to define functional and statistical measures only on the basis of gray level value (histogram).**

Solution:

Region Growing based segmentation techniques, such as:

Region splitting, Region merging, Split and Merge and Region growing techniques.

Region-Growing based segmentation

Homogeneity of regions is used as the main segmentation criterion in region growing.

The criteria for homogeneity:

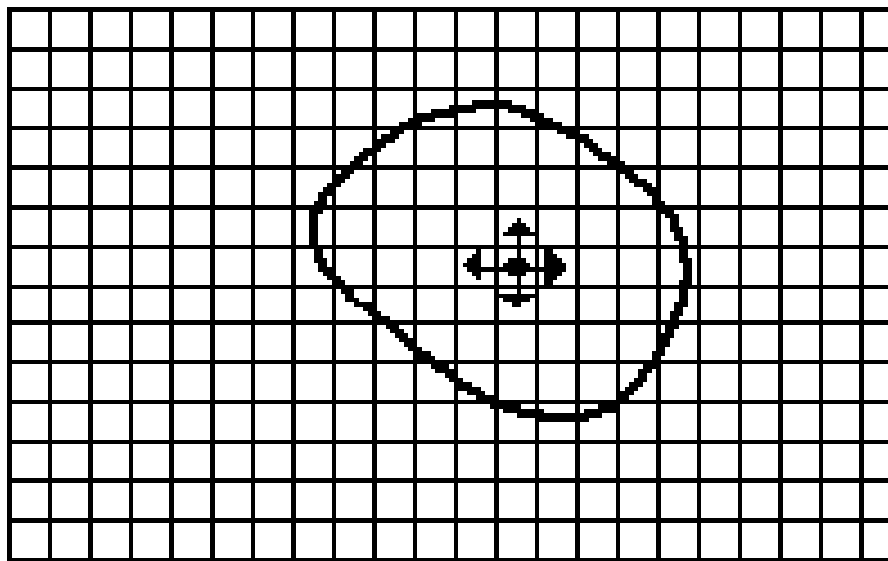
- **gray level**
- **color**
- **texture**
- **shape**
- **model**

The basic purpose of region growing is to segment an entire image R into smaller sub-images, R_i , $i=1,2,\dots,N$. which satisfy the following conditions:

$$R = \bigcup_{i=1}^N R_i; R_i \cap R_j = \Phi, i \neq j$$

$$H(R_i) = \text{True}; i = 1,2,\dots, N;$$

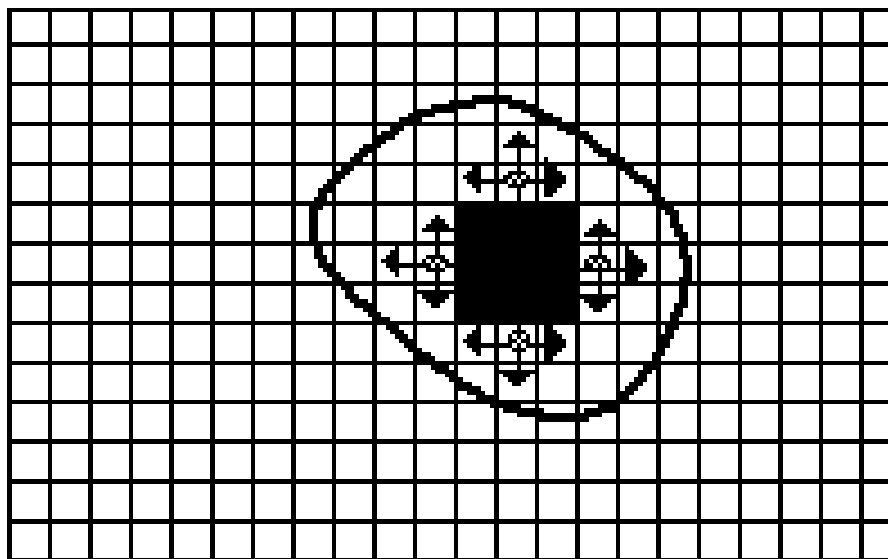
When, R_i and R_j are adjacent: $H(R_i \cup R_j) = \text{False}, i \neq j$;



• Seed Pixel

↑ Direction of Growth

(a) Start of Growing a Region



■ Grown Pixels

⊗ Pixels Being Considered

(b) Growing Process After a Few Iterations

Region Growing

Region growing approach is the opposite of the split and merge approach:

- **An initial set of small areas is iteratively merged according to similarity constraints.**
- **Start by choosing an arbitrary *seed pixel* and compare it with neighboring pixels (see Fig).**
- **Region is *grown* from the seed pixel by adding in neighboring pixels that are similar, increasing the size of the region.**
- **When the growth of one region stops we simply choose another seed pixel which does not yet belong to any region and start again.**
- **This whole process is continued until all pixels belong to some region.**
- ***A bottom up* method.**

Region growing methods often give very good segmentations that correspond well to the observed edges.

However starting with a particular seed pixel and letting this region grow completely before trying other seeds biases the segmentation in favour of the regions which are segmented first.

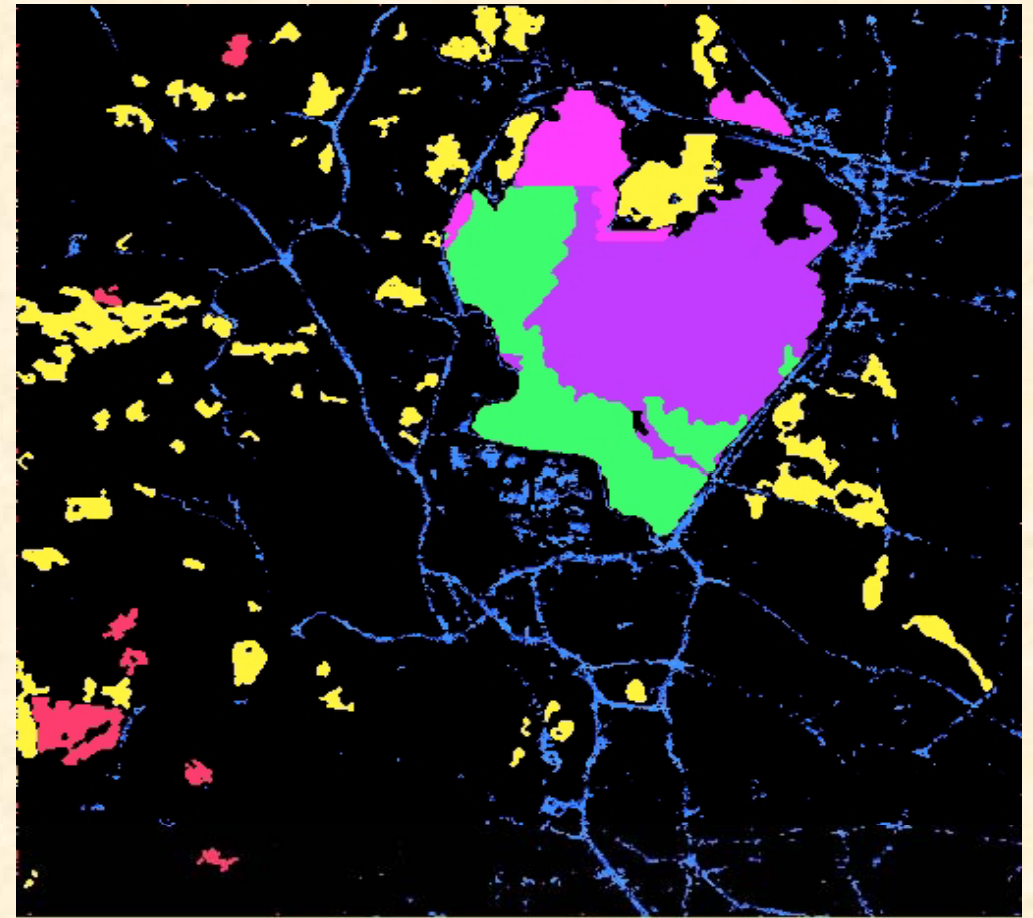
This can have several undesirable effects:

- **Current region dominates the growth process -- ambiguities around edges of adjacent regions may not be resolved correctly.**
- **Different choices of seeds may give different segmentation results.**
- **Problems can occur if the (arbitrarily chosen) seed point lies on an edge.**

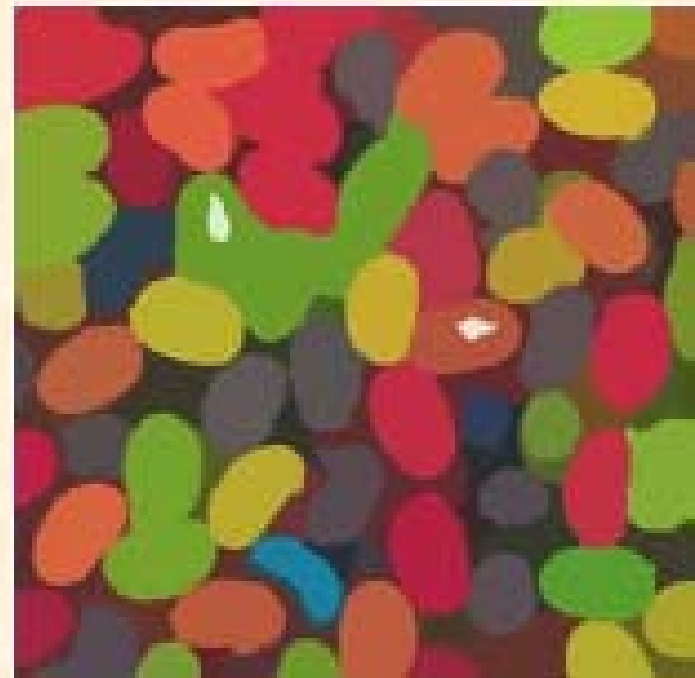
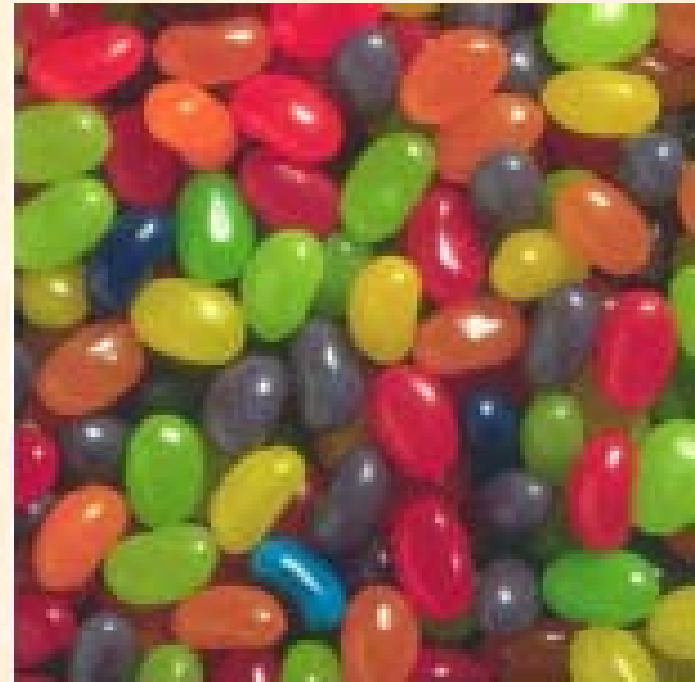
To counter the above problems, *simultaneous region growing* techniques have been developed.

- **Similarities of neighboring regions are taken into account in the growing process.**
- **No single region is allowed to completely dominate the proceedings.**
- **A number of regions are allowed to grow at the same time.**
- **Similar regions will gradually coalesce into expanding regions.**
- **Control of these methods may be quite complicated but efficient methods have been developed.**
- **Easy and efficient to implement on parallel computers.**





**Terrain classification based on color properties
of a satellite Image of Hyderabad lake area**



Modeling as a Graph Partitioning problem

- Set of points of the feature space represented as a weighted, undirected graph, $G = (V, E)$
- The points of the feature space are the nodes of the graph.
- Edge between every pair of nodes.
- Weight on each edge, $w(i, j)$, is a function of the similarity between the nodes i and j .
- Partition the set of vertices into disjoint sets where similarity within the sets is high and across the sets is low.

Weight Function for Brightness Images

- Weight measure (reflects likelihood of two pixels belonging to the same object)

$$w_{ij} = \exp - \frac{(I(i) - I(j))^2}{\sigma_I^2} * \begin{cases} \exp - \frac{\|X(i) - X(j)\|_2^2}{\sigma_X^2} & \text{if } \|X(i) - X(j)\|_2 < R \\ 0 & \text{otherwise} \end{cases}$$

For brightness images, $I(i)$ represents normalized intensity level of node i and $X(i)$ represents spatial location of node i .


σ_I and σ_X are parameters set to 10-20 percent of the range of their related values.


R is a parameter that controls the sparsity of the resulting graph by setting edge weights between distant pixels to 0.

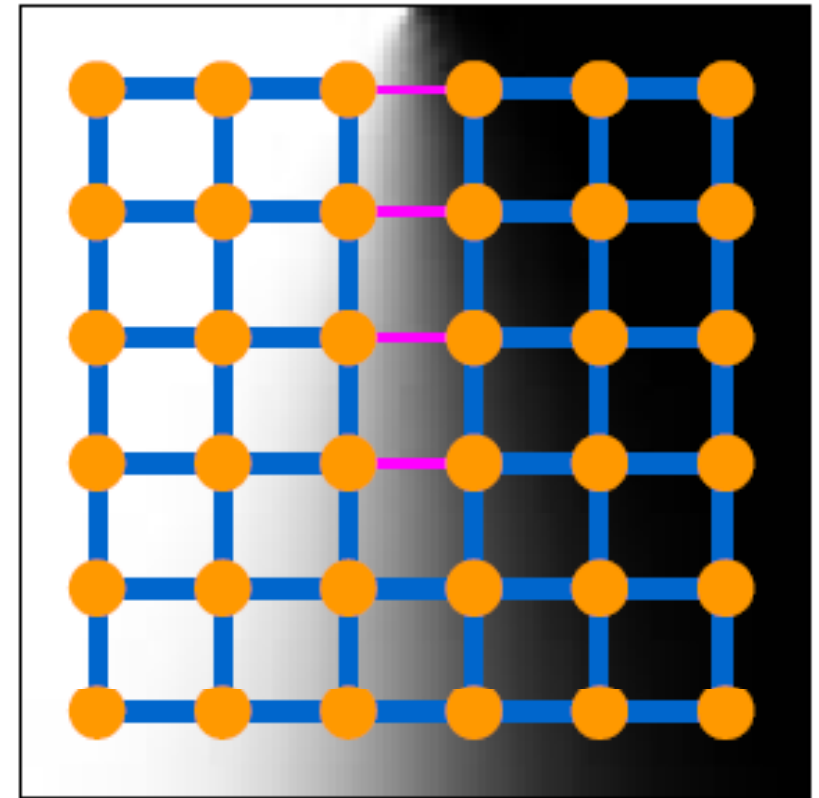
The Pixel Graph

Couplings $\{w_{ij}\}$

Reflect intensity similarity

 Low contrast –
strong coupling

 High contrast –
weak coupling



V: graph nodes:

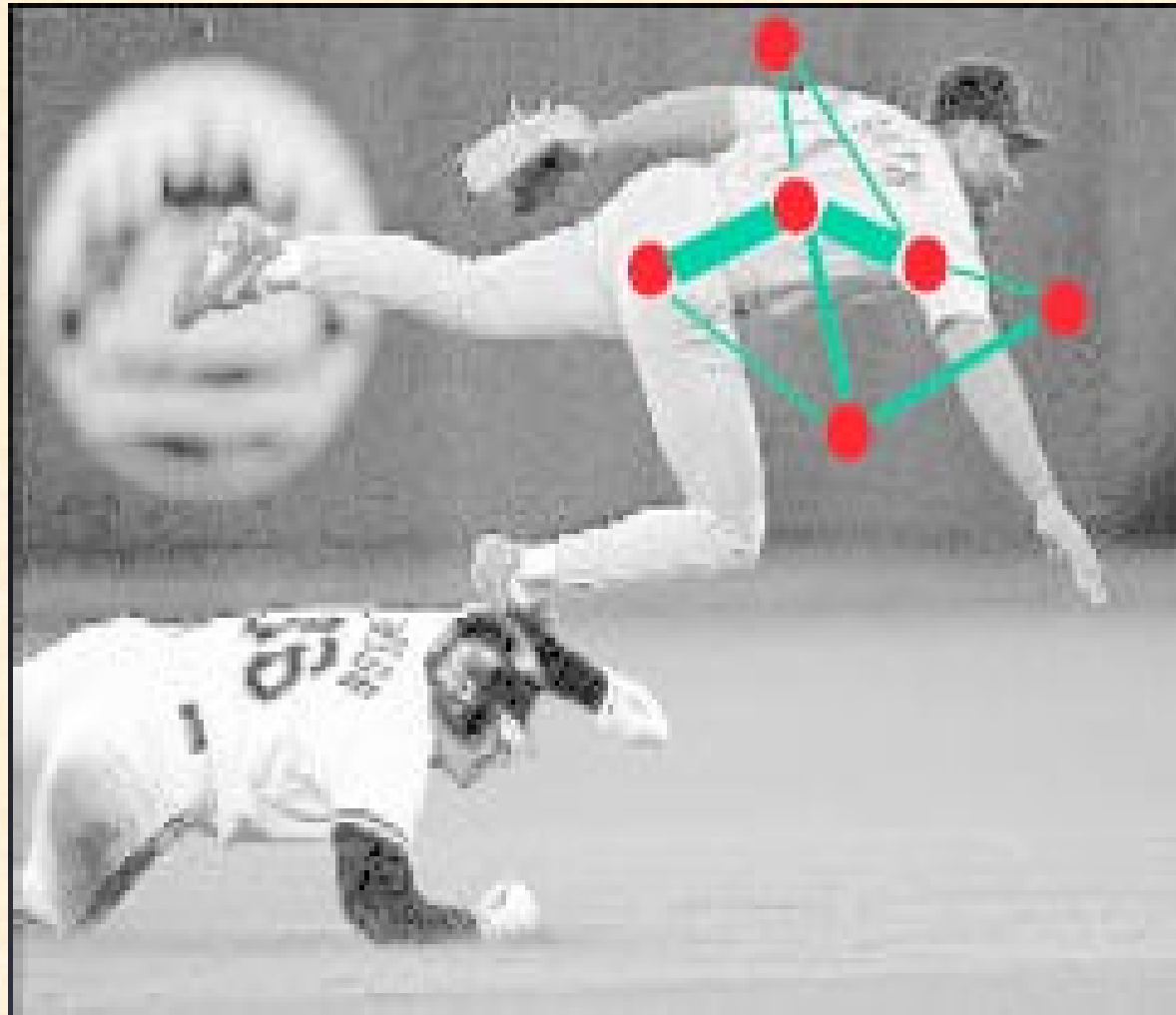


Image = { pixels }

E: edges connection nodes:  

Pixel similarity

Representing Images as Graphs



Segmentation and Graph Cut

- A graph can be partitioned into two disjoint sets by simply removing the edges connecting the two parts
- The degree of dissimilarity between these two pieces can be computed as total weight of the edges that have been removed
- More formally, it is called the **'cut'**

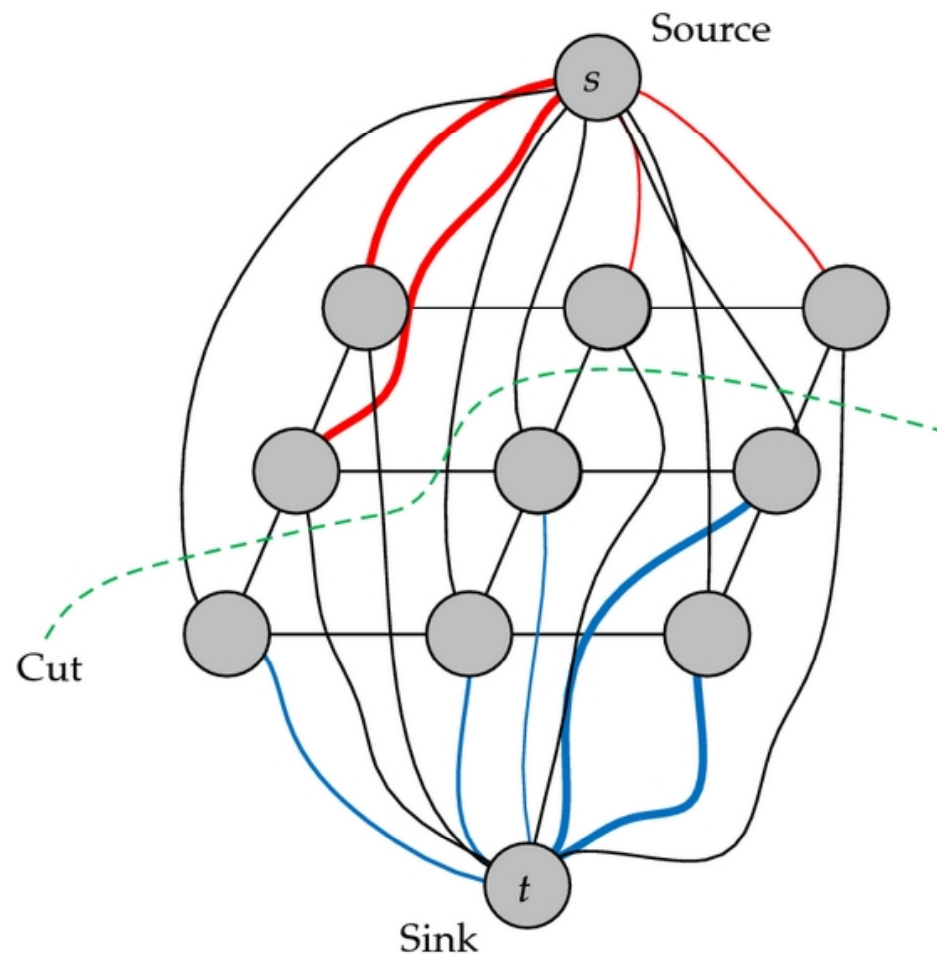
Segmentation and Graph Cut

- 1) Given a source (s) and a sink node (t)
- 2) Define Capacity on each edge, $C_{ij} = W_{ij}$
- 3) Find the maximum flow from $s \rightarrow t$, satisfying the capacity constraints

Min. Cut = Max. Flow

Max-flow/Min-cut theorem:

For any network having a single origin node and destination node, the maximum flow from origin to destination equals the minimum cut value for all cuts in the network.

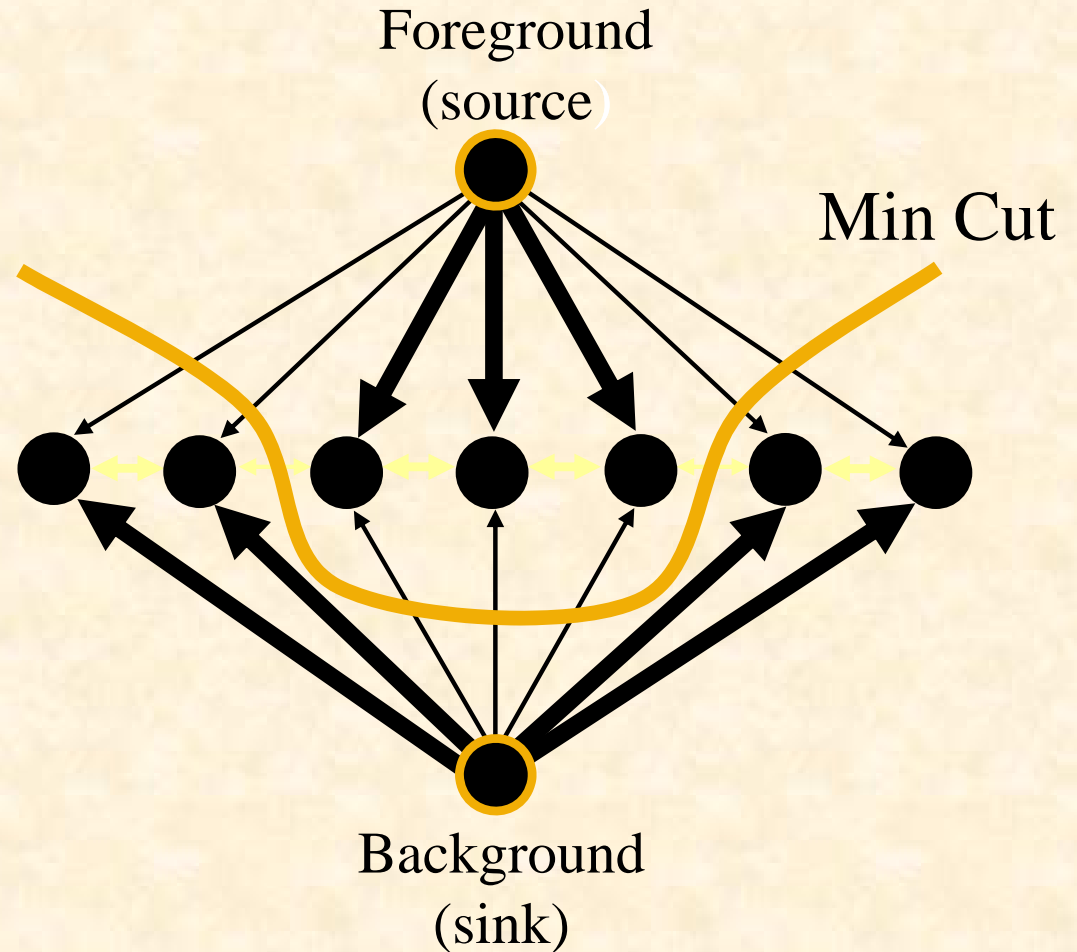
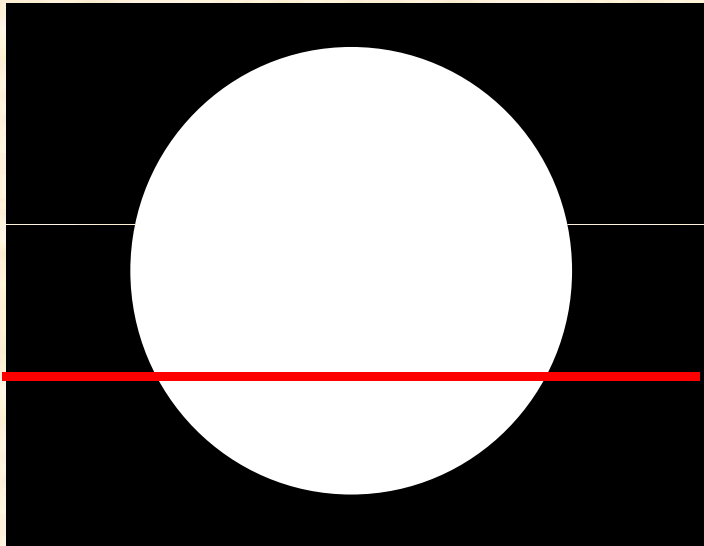


$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v).$$

An example of min-cut/max-flow graph cut. The gray circles represent the nodes, and the solid lines are the edges between the nodes. The curve indicating each “flow” is connected to the source terminal or sink terminal. The potential of flow is measured by the width of line. The dotted line indicates a cut of graph partition.

Graph cuts

Image



Cut: separating source and sink; Energy: collection of edges

Min Cut: Global minimal energy in polynomial time

Optimization Problem

- Minimize the *cut* value

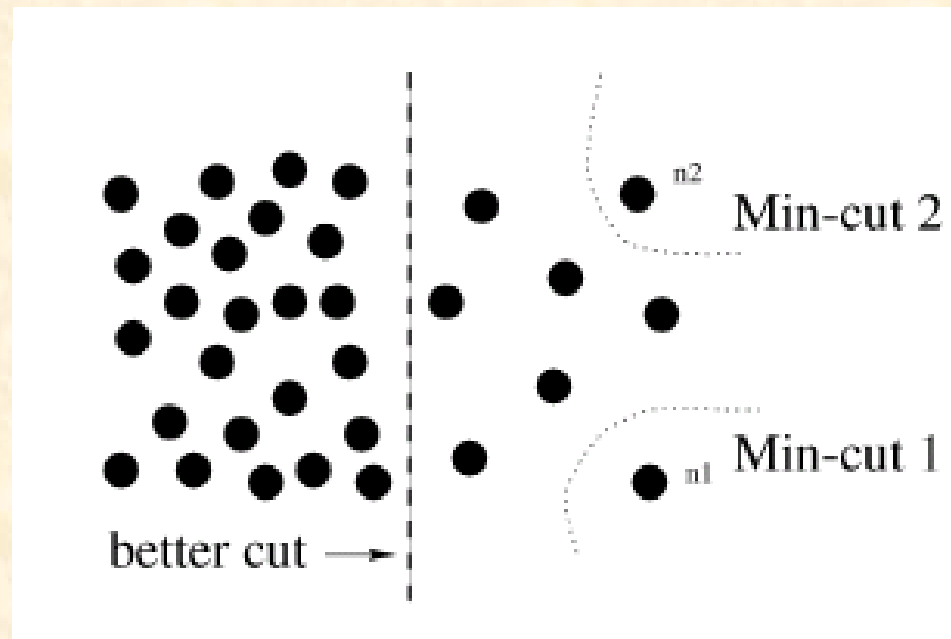
$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

$$A \cup B = V, A \cap B = \emptyset$$

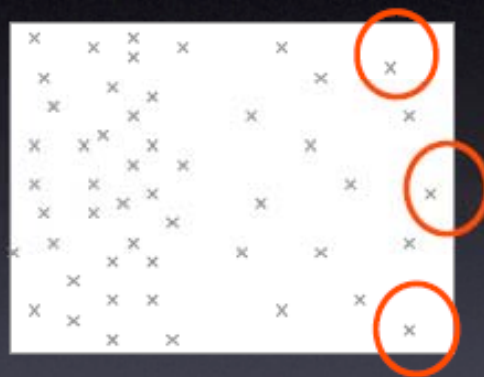
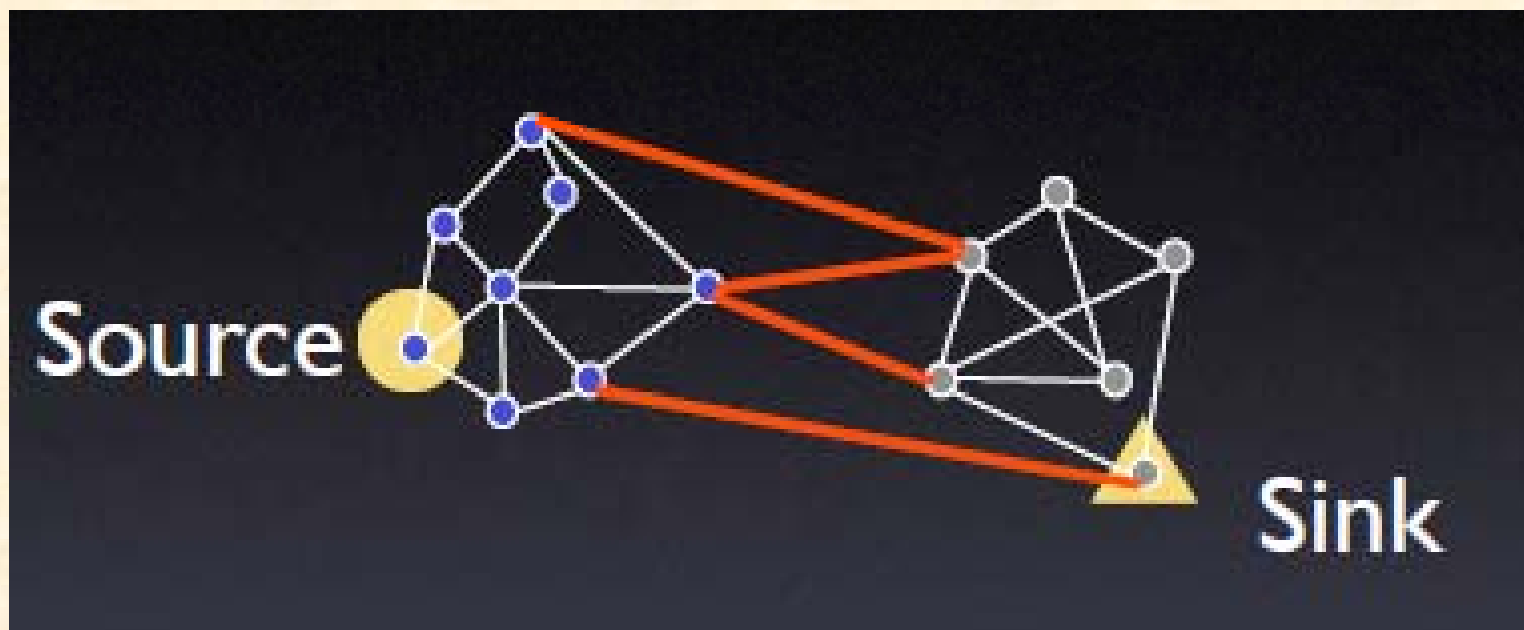
- Number of such partitions is exponential (2^N);
but the minimum cut can be found efficiently

Problems with min-cut

- Minimum cut criteria favors cutting small sets of isolated nodes in the graph.



A case where minimum cut gives bad partition



Min. cuts favors isolated clusters

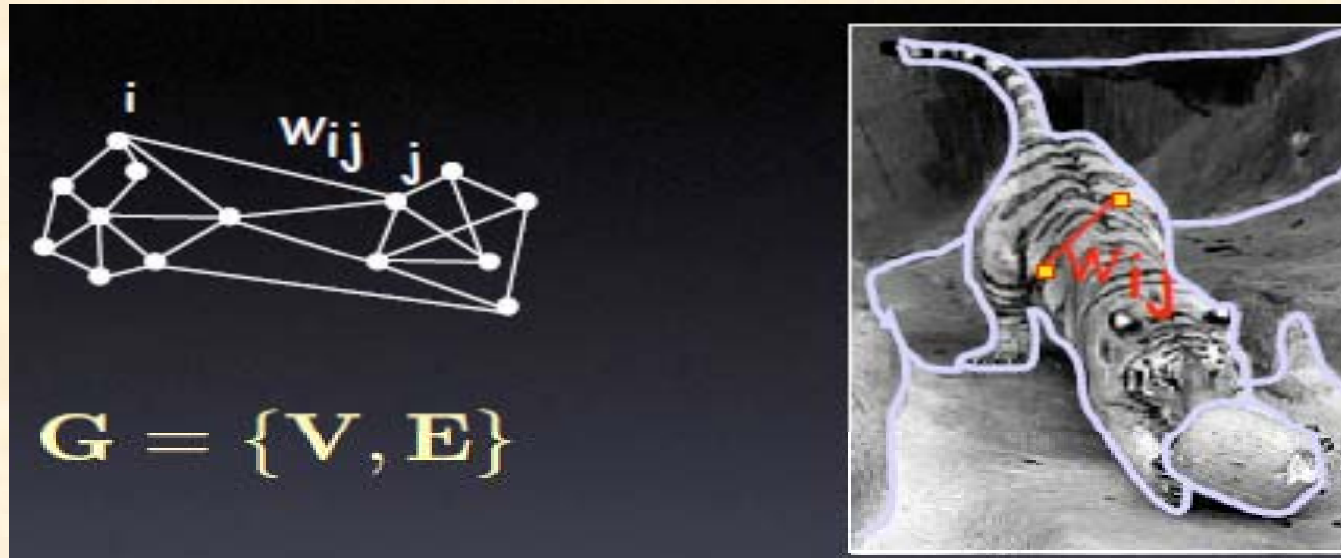
NP-Hard!



Solution – Normalized Cut

- Avoid unnatural bias for partitioning out small sets of points
- Normalized Cut - computes the cut cost as a fraction of the total edge connections to all the nodes in the graph

NORMALIZED GRAPH CUT



V: graph nodes: \longleftrightarrow Image = { pixels }

E: edges connection nodes: \longleftrightarrow Pixel similarity

A graph $G = \{V, E\}$ can be partitioned into two disjoint sets: A, B ; $A \cup B = V$, $A \cap B = \Phi$, by simply removing edges connecting the two parts.

The degree of dissimilarity between these two pieces can be computed as total weight of the edges that have been removed.

In graph theoretic language, it is called the cut:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

In grouping, we seek to partition the set of vertices into disjoint sets V_1, V_2, \dots, V_m , where by some measure the similarity among the vertices in a set V_i is high and, across different sets V_i, V_j is low.

Mincut creates a optimal bi-partitioning of the graph. Instead of looking at the value of total edge weight connecting the two partitions, a normalized measure computes the cut cost as a fraction of the total edge connections to all the nodes in the graph.

This disassociation measure is called the normalized cut (Ncut):

Minimize the cut, while maximize the association

$$H^{\text{NCut}}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(B, A)}{\text{assoc}(B, V)}$$

where, $\text{assoc}(A, V)$ is the total connection from nodes in A to all nodes in the graph.

$$\text{assoc}(A, V) = \sum_{u \in A, t \in V} w(u, t);$$

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

Computational Issues

- Exact solution to minimizing normalized cut is an NP-complete problem
- However, approximate discrete solutions can be found efficiently
- Normalized cut criterion can be computed efficiently by solving a generalized eigenvalue problem

Need to partition the nodes of a graph, V , into two sets A and B .

Let x be an $N = |V|$ dimensional indicator vector, $x_i = 1$, if node i is in A , else -1 .

Let , $d(i) = \sum_j w(i, j)$

be the total connection from node i to all other nodes.

Let D be an $N \times N$ diagonal matrix with d on its diagonal;

W be an $N \times N$ symmetrical matrix with $W(i, j) = w(i, j)$;

W is also an adjacency matrix.

The normalized cut is defined as :

$$N_{cut}(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(B, A)}{assoc(B, V)}$$

$$N_{cut}(A, B) = \frac{\sum_{(x_i > 0, x_j < 0)} -w_{ij} x_i x_j}{\sum_{x_i > 0} d(i)} + \frac{\sum_{(x_i < 0, x_j > 0)} -w_{ij} x_i x_j}{\sum_{x_i < 0} d(i)}$$

**$x_i = 1$, if node i is in A , else -1 ;
(assignment done, post-optimization)**

$$d(i) = \sum_j w(i, j)$$

Let, $k = \frac{\sum_{x_i > 0} d(i)}{\sum_i d(i)}$; and $\mathbf{1}$ be a $N \times 1$ vector of all ones.

- The $N_{cut}(x)$ can be rewritten as

$$\frac{(1+x)^T (\mathbf{D} - \mathbf{W})(1+x)}{k \mathbf{1}^T \mathbf{D} \mathbf{1}} + \frac{(1-x)^T (\mathbf{D} - \mathbf{W})(1-x)}{(1-k) \mathbf{1}^T \mathbf{D} \mathbf{1}}$$

Solving and setting,

$b = k/(1-k) :$

$$= \frac{[(1+x) - b(1-x)]^T (\mathbf{D} - \mathbf{W}) [(1+x) - b(1-x)]}{b \mathbf{1}^T \mathbf{D} \mathbf{1}}$$

$$N_{cut}(x) = \frac{\left[(1+x) - b(1-x)\right]^T (D-W) \left[(1+x) - b(1-x)\right]}{b1^T D1}$$

Using, $y = (1+x) - b(1-x)$
we have :

under the condition

$y(i) \in \{1, -b\}$ and $y^T D1 = 0$

$$\min_x N_{cut}(x) = \min_y \frac{y^T (D-W)y}{y^T D y}$$

The above expression is the **Rayleigh quotient**. If y is relaxed to take on real values, we can minimize above eqⁿ. by solving the generalized eigenvalue system:

$$\mathbf{L}y = (\mathbf{D} - \mathbf{W})y = \lambda \mathbf{D}y$$

Refer – Golub & Van Loan for above theory.

$\mathbf{L} = (\mathbf{D}-\mathbf{W})$ is called the Laplacian matrix (+ve Semi-Defnt.).

Rayleigh quotient can be reduced to:

$$D^{-\frac{1}{2}} L D^{-\frac{1}{2}} z = \lambda z \Rightarrow A z = \lambda z;$$

**where \mathbf{A} is sparse, as \mathbf{W} is sparse;
the above can be solved in $O(n)$ time.**

Partition (grouping) algorithm steps:

1. Given an image or image sequence, set up a weighted graph $G = (V, E)$, and set the weight on the edge connecting two nodes to be a measure of the similarity between the two nodes.
2. Solve $(D - W)x = \lambda Dx$ for eigenvectors with the smallest eigenvalues.
3. Use the eigenvector with the **second smallest eigenvalue** to bipartition the graph.
4. Decide if the current partition should be subdivided and recursively

Rayleigh Quotient: $\min_x NCut(x) = \min_y \frac{y^T (D - W) y}{y^T D y}$

A simple fact about the Rayleigh quotient

Let A be a real symmetric matrix. Under the constraint that x is orthogonal to the $j-1$ smallest eigenvectors x_1, \dots, x_{j-1} , the quotient $\mathbf{x}^T \mathbf{A} \mathbf{x} / \mathbf{x}^T \mathbf{x}$ is minimized by the next smallest eigenvector x_j and its minimum value is the corresponding eigenvalue j .

Thus, the second smallest eigenvector of the generalized eigen-system is the real valued solution to our normalized cut problem

Generalization: For a given pair (A, B) of real symmetric positive-definite matrices, and a given non-zero vector x , the generalized Rayleigh quotient is defined as:

$$R(A, B; x) = \frac{x^T A x}{x^T B x}$$

The Generalized Rayleigh Quotient can be reduced to the Rayleigh Quotient $\mathbf{R}(\mathbf{D}, \mathbf{C} \mathbf{x})$, through the transformation $D = C^{*-1} A C^{-1}$

where C is the Cholesky decomposition of matrix B .

Altn. Formulation:

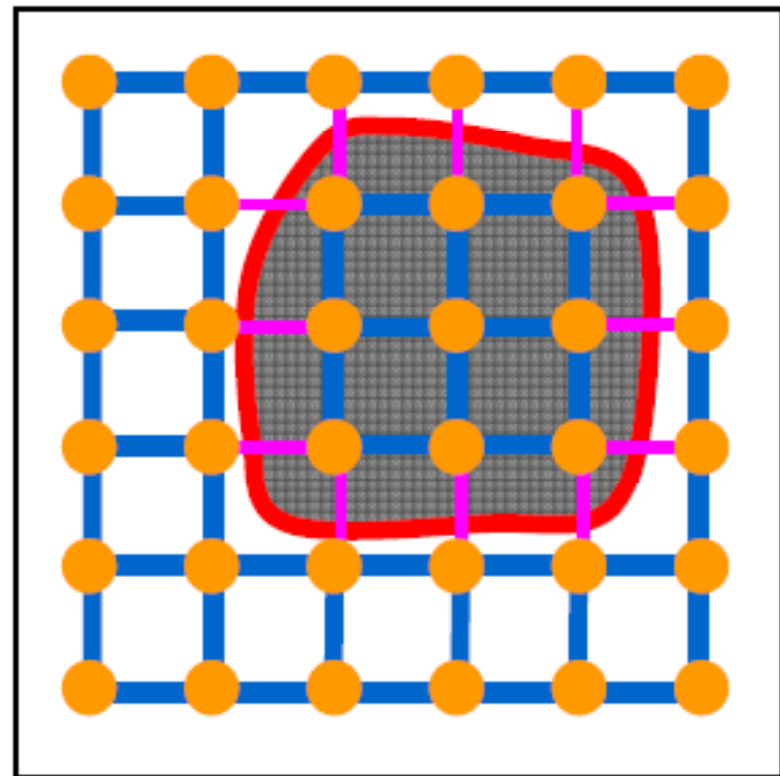
Normalized-Cut Measure

$$E(S) = \sum_{i \neq j} w_{ij} (u_i - u_j)^2 \quad u_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$

$$N(S) = \sum w_{ij} u_i u_j$$

Minimize:

$$\Gamma(S) = \frac{E(S)}{N(S)}$$

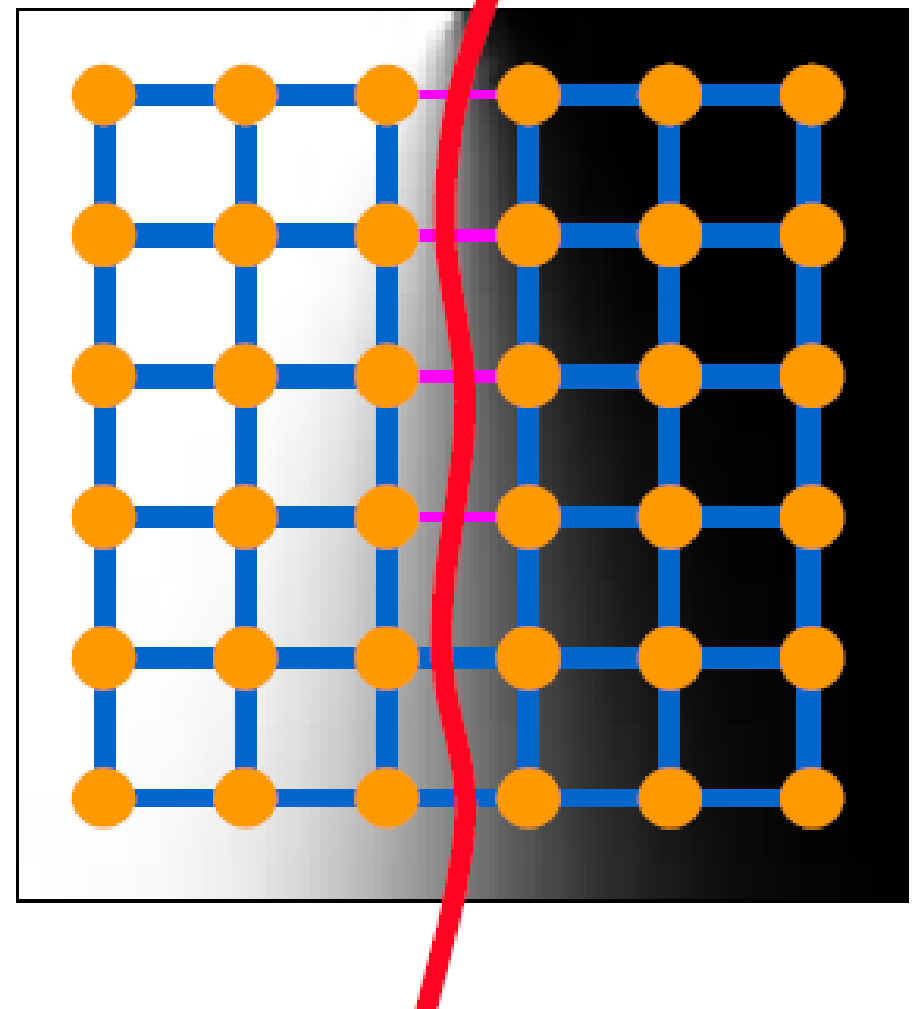


Normalized-Cut Measure

Low-energy cut

Minimize:

$$\Gamma(S) = \frac{E(S)}{N(S)}$$



Matrix Formulation

Define matrix W by $w_{ij} > 0$ $w_{ii} = 0$

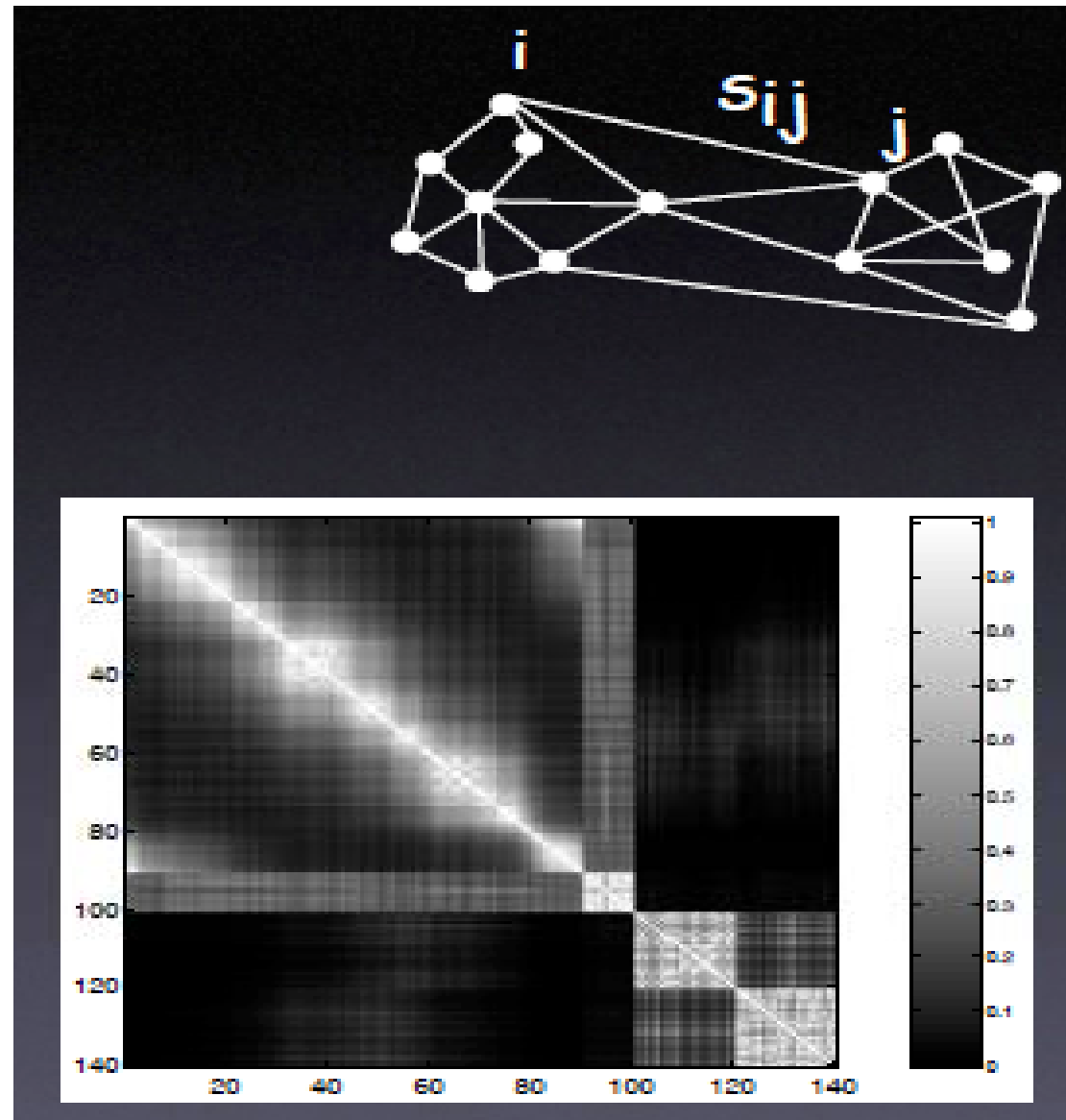
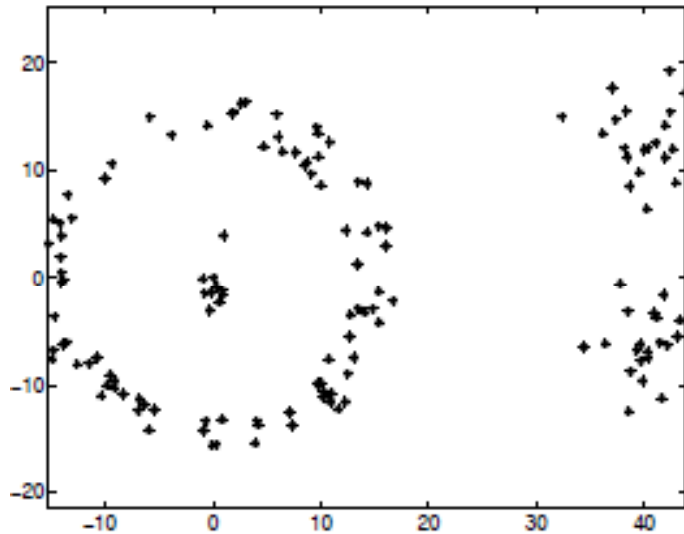
Define matrix L by
$$l_{ij} = \begin{cases} \sum_{k, (k \neq i)} w_{ik} & i = j \\ -w_{ij} & i \neq j \end{cases}$$

We minimize $\Gamma(u) = \frac{u^T L u}{\frac{1}{2} u^T W u}$

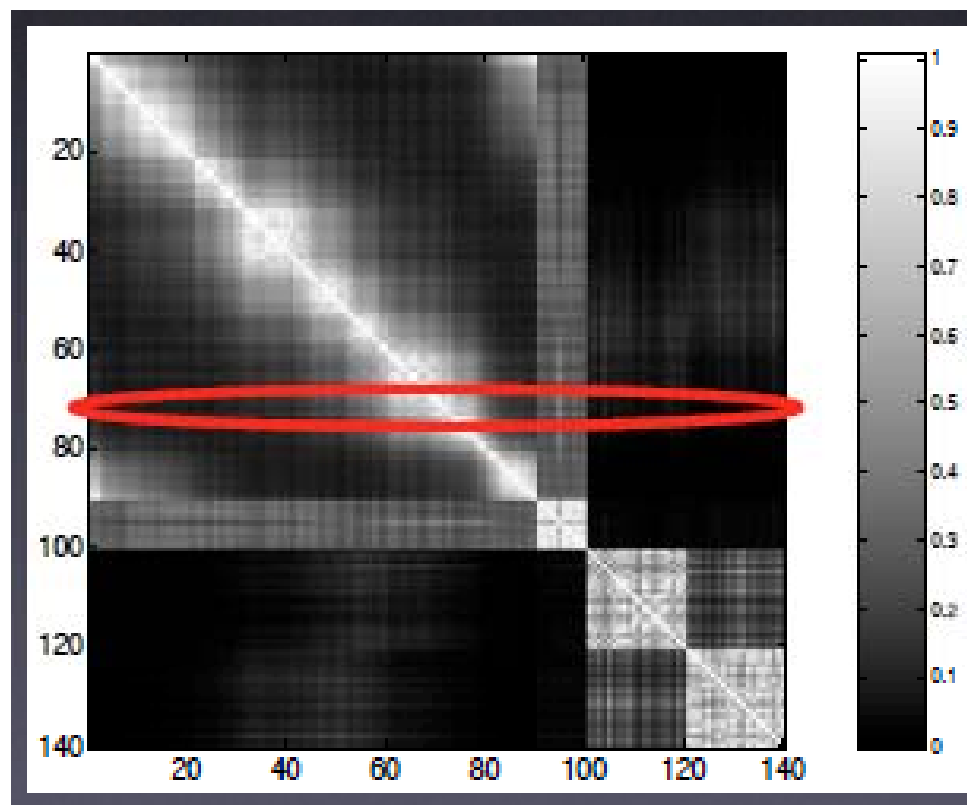
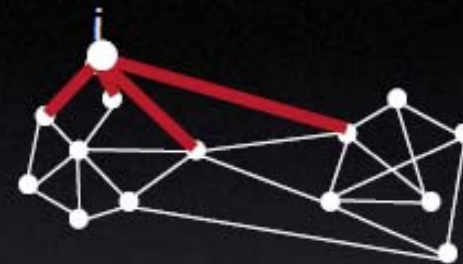
Read about Spectral-cut methods

A Graphical Illustration of GRAPHCUT

**Generalized
Adjacency (W)
Or Similarity (S)
Matrix :**

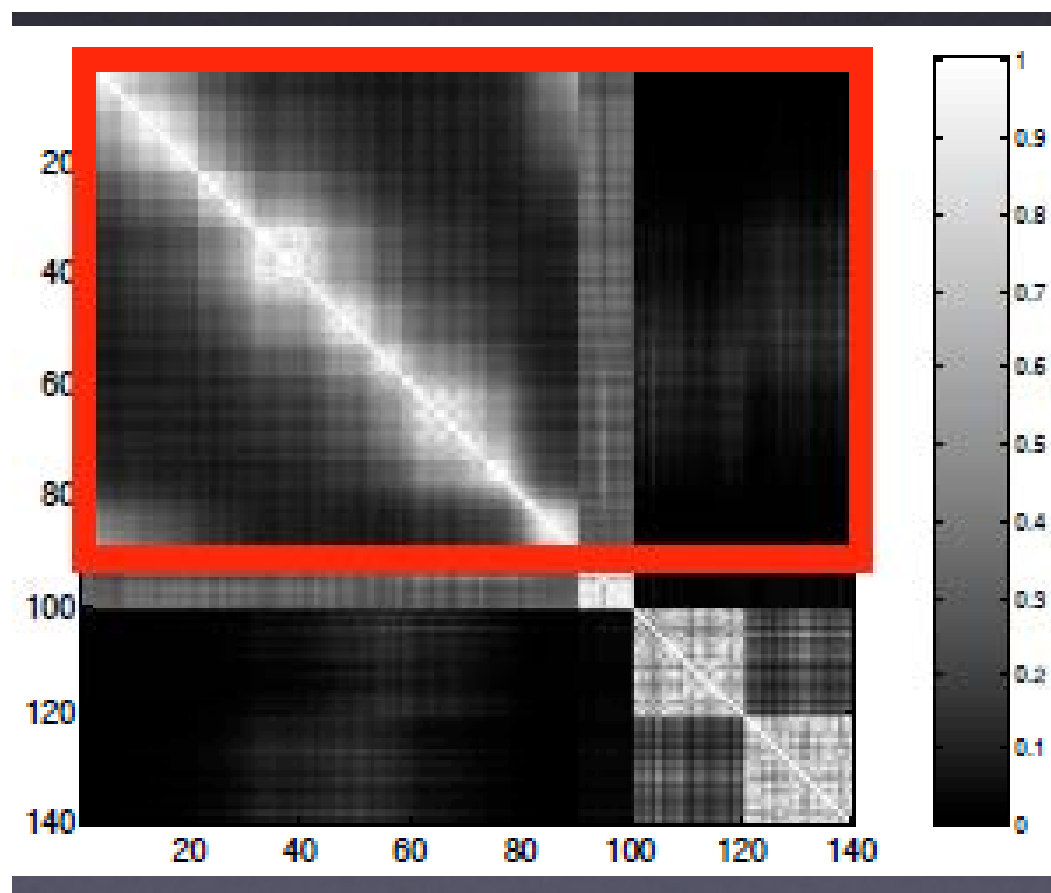


Degree of node: $d_i = \sum_j S_{ij}$

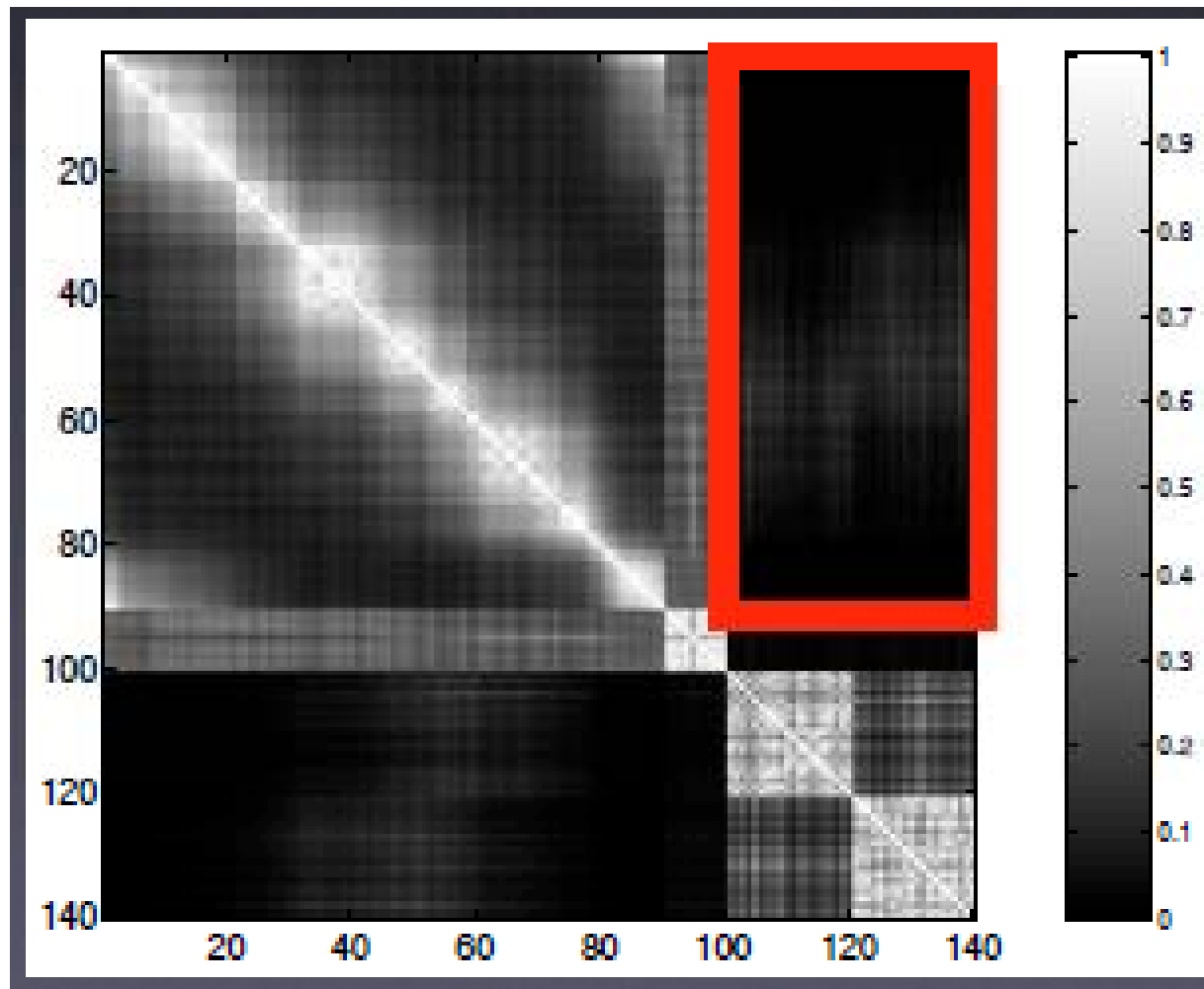
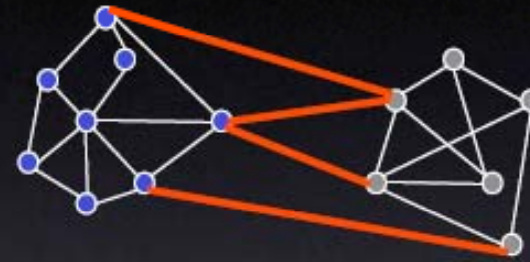


Volume of set:

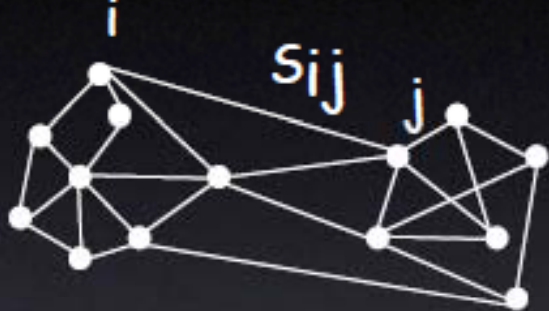
$$\text{vol}(A) = \sum_{i \in A} d_i, A \subseteq V$$



$$\text{cut}(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} S_{i,j}$$



Similarity matrix $S = [S_{ij}]$



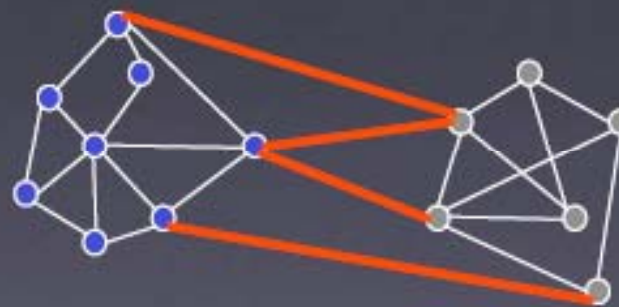
Degree of node: $d_i = \sum_j S_{ij}$



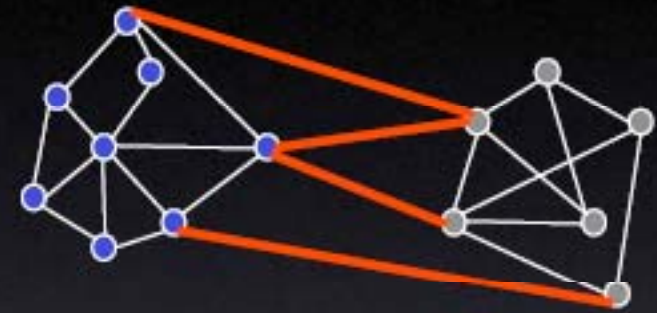
Volume of set:



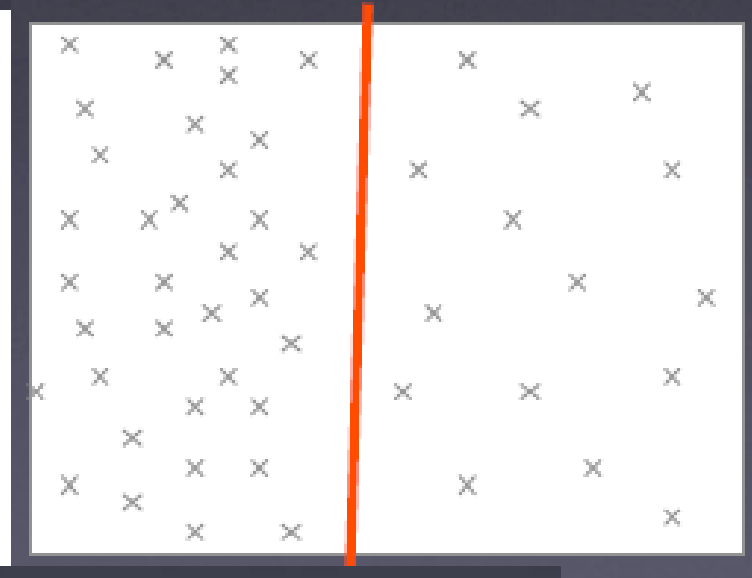
Graph Cuts



- (edge) Ncut = balanced cut



$$Ncut(A, B) = cut(A, B) \left(\frac{1}{vol(A)} + \frac{1}{vol(B)} \right)$$



Pair-wise similarity matrix W

Laplacian matrix $D-W$

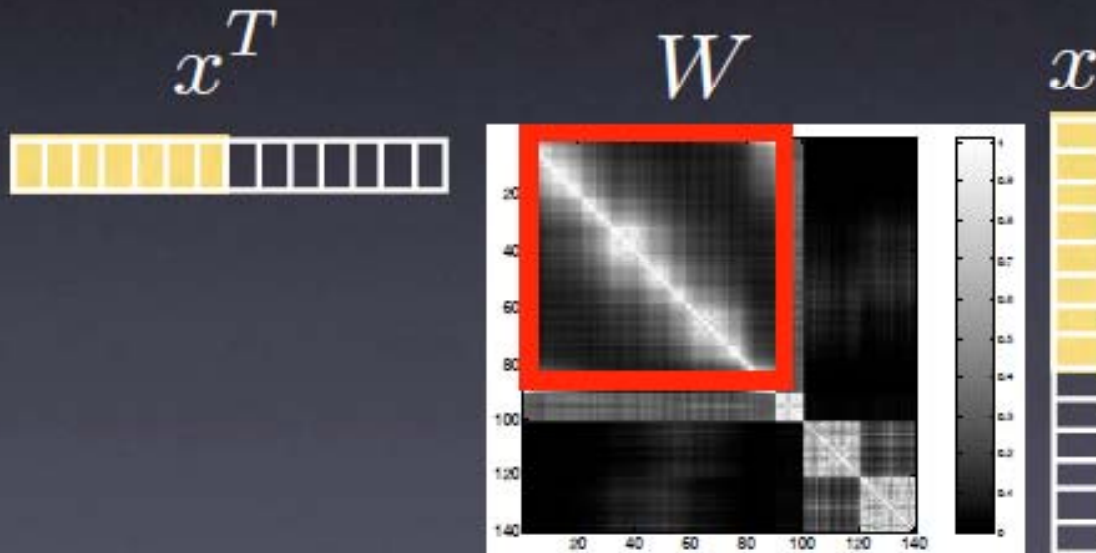
Degree matrix D : $D(i, i) = \sum_j W_{i,j}$

Laplacian matrix D-W

Let $x = X(l,:)$ be the indicator of group l

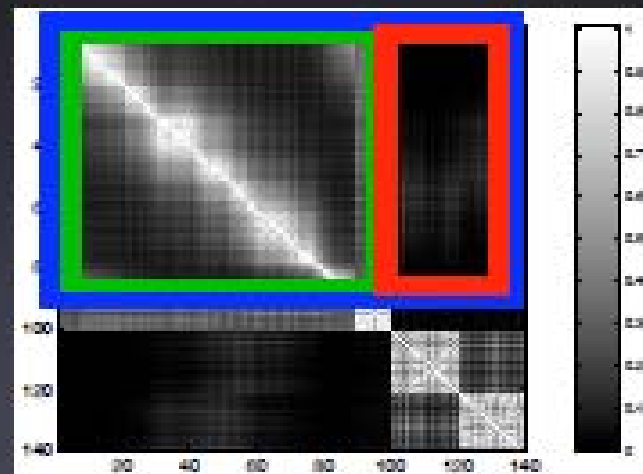
$$\text{asso}(A,A) = x^T W x$$

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Laplacian matrix D-W

$$\text{Cut}(A, V-A) = \frac{x^T D x}{\text{vol}(A)} - \frac{x^T W x}{\text{asso}(A, A)}$$



$$\text{Cut}(A, V - A) = x^T (D - W) x$$

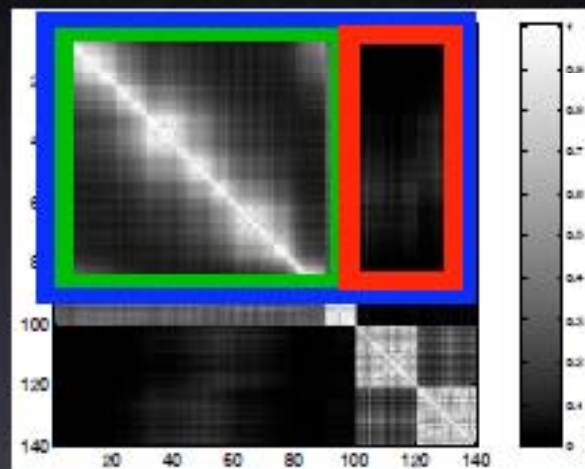
$$Ncut(A, B) = cut(A, B) \left(\frac{1}{vol(A)} + \frac{1}{vol(B)} \right)$$

$$Cut(A, V-A) = \frac{x^T D x}{vol(A)} - \frac{x^T W x}{asso(A, A)}$$

$$Cut(A, V - A) = x^T (D - W)x$$

$$\min_x Ncut(x) = \min_y \frac{y^T (\mathbf{D} - \mathbf{W})y}{y^T \mathbf{D}y}$$

$$Ncut(X) = \frac{1}{K} \sum_{l=1}^K \frac{cut(V_l, V - V_l)}{vol(V_l)}$$



$$= \frac{1}{K} \sum_{l=1}^K \frac{X_l^T (D - W) X_l}{X_l^T D X_l}$$

Example Normalized Cut



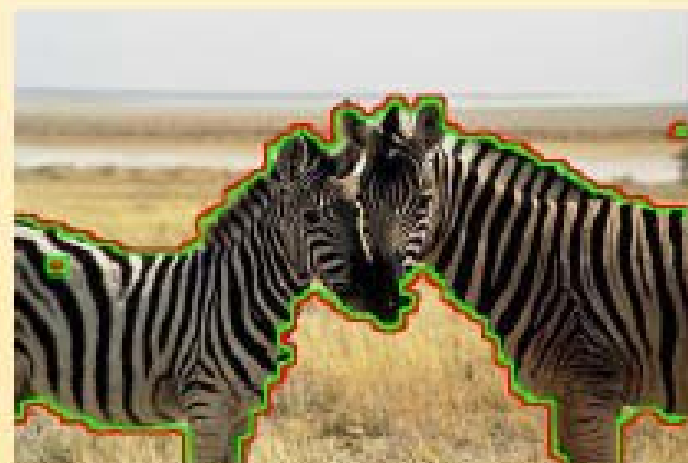
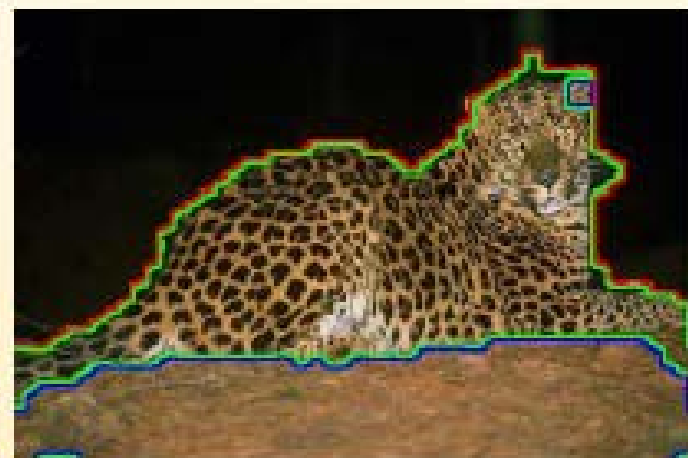
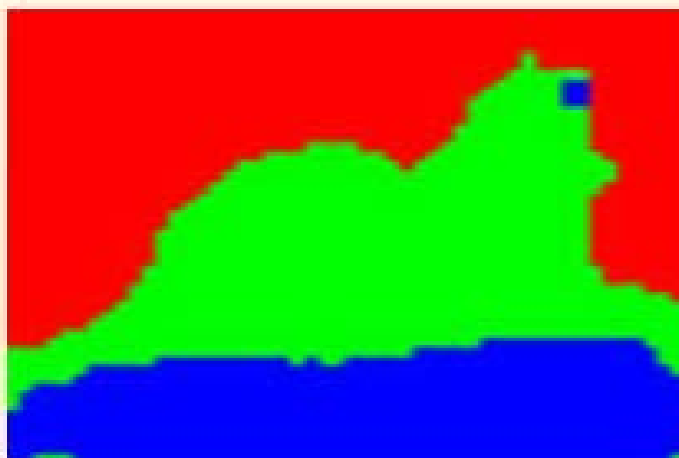
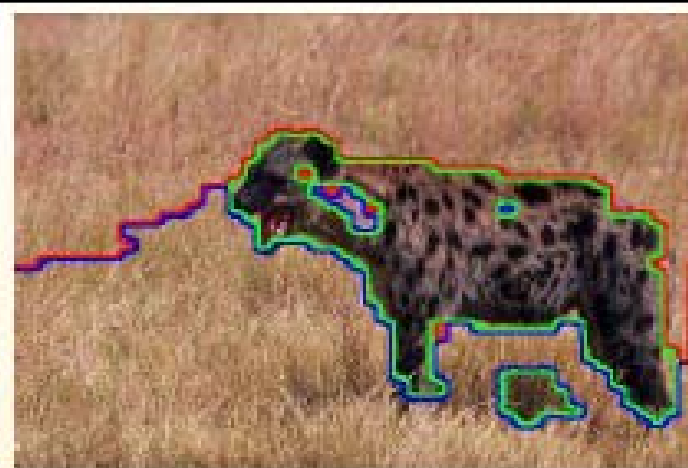
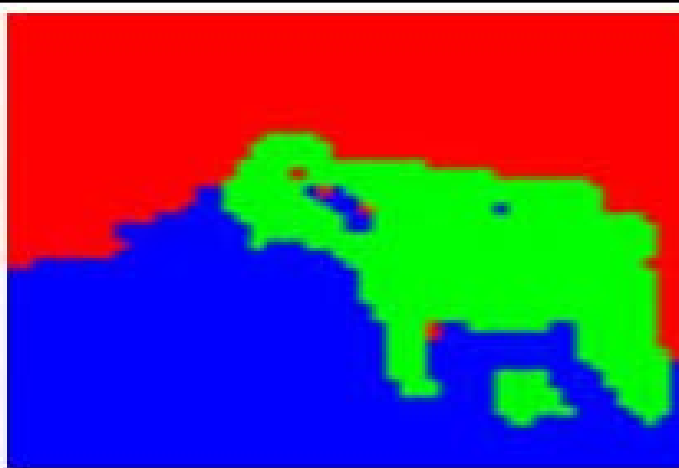
(a)

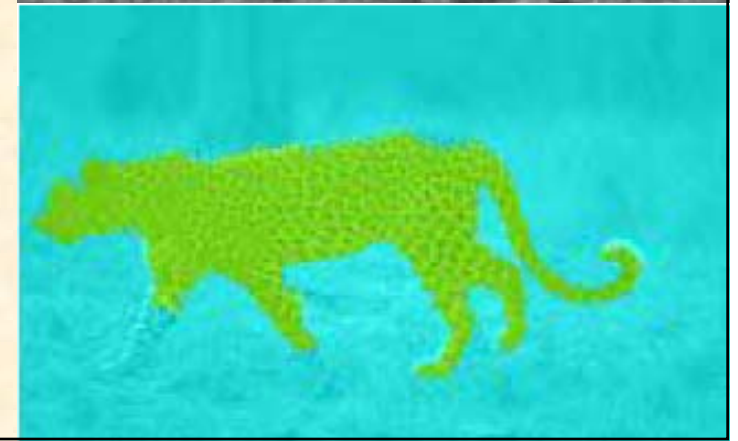
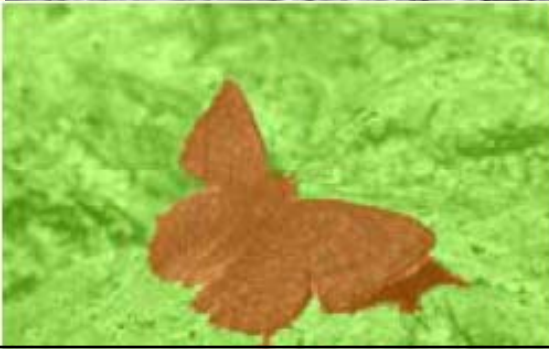


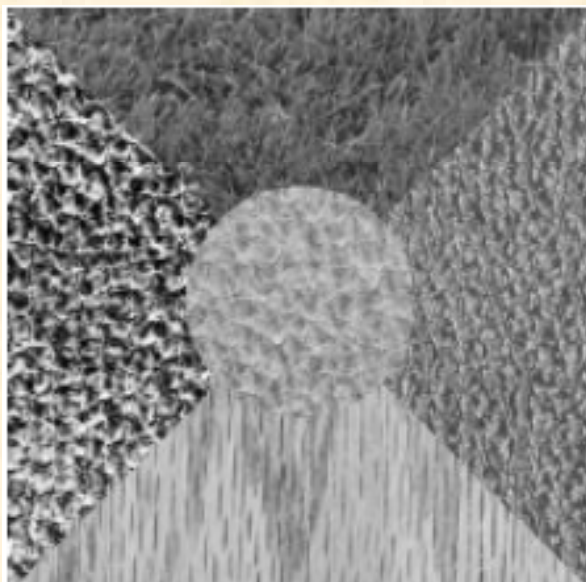
(b)



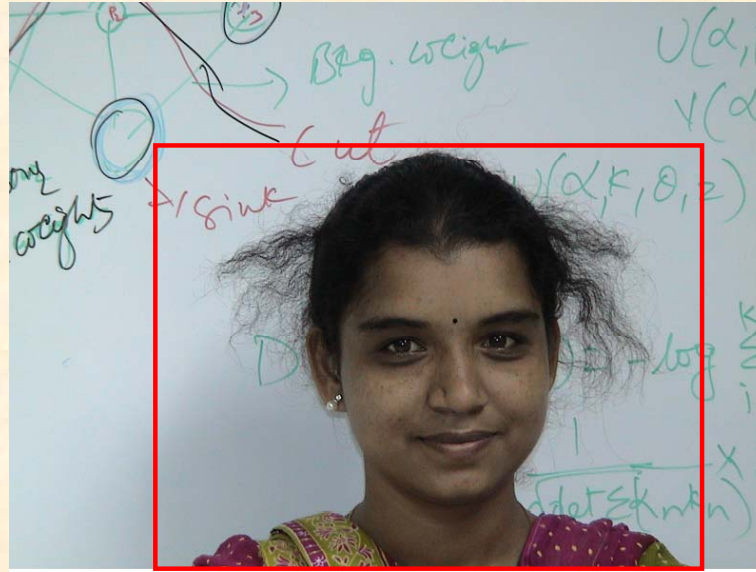
Shi & Malik, 2000







Object Extraction From an Image



Snake



N-Cut



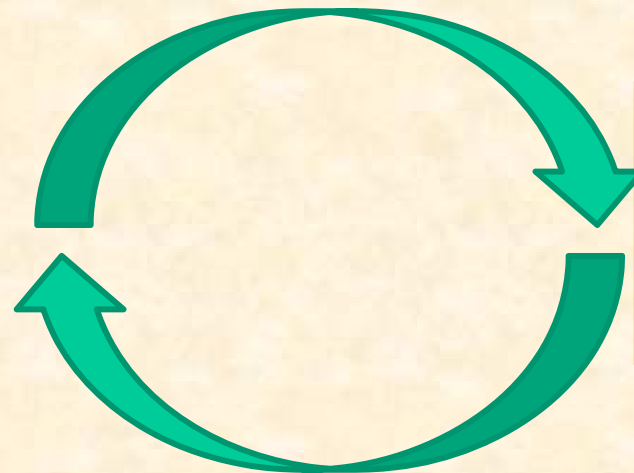
Iterated Graph Cut



User Initialisation



**K-means for learning
colour distributions**

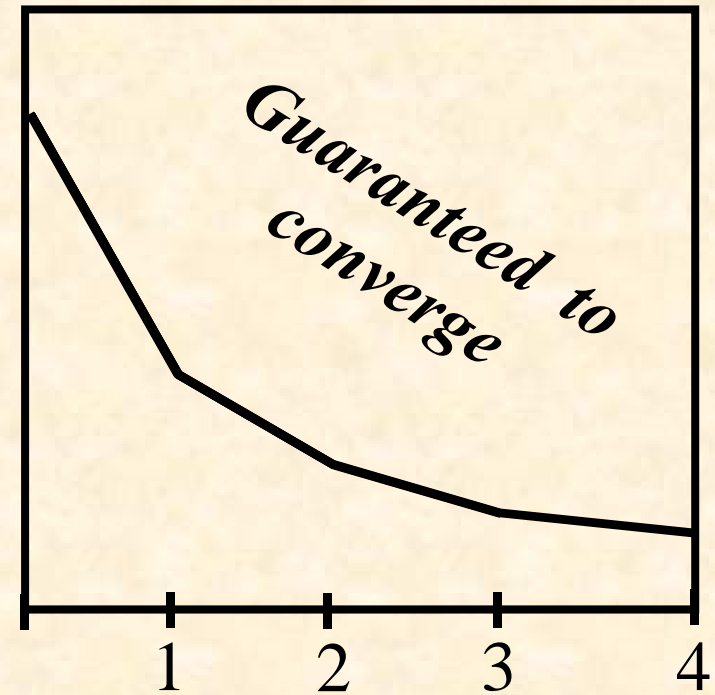


**Graph cuts to
infer the
segmentation**

Iterated Graph Cuts



Result

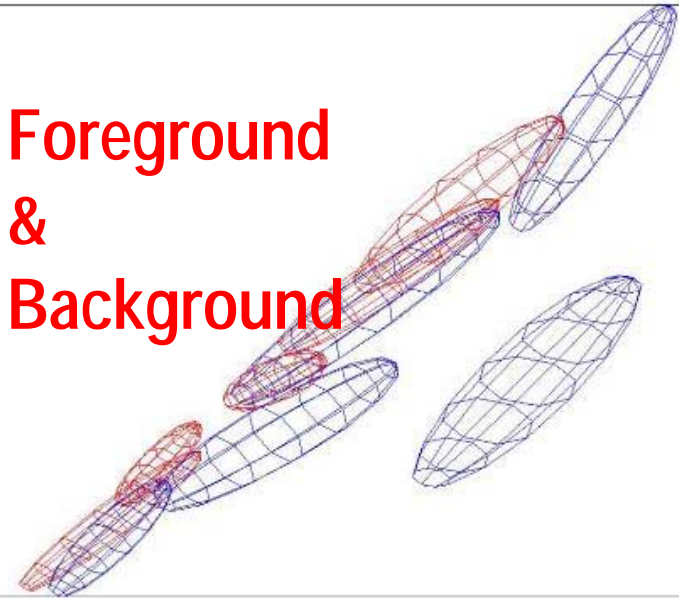


Energy after each Iteration

Colour Model

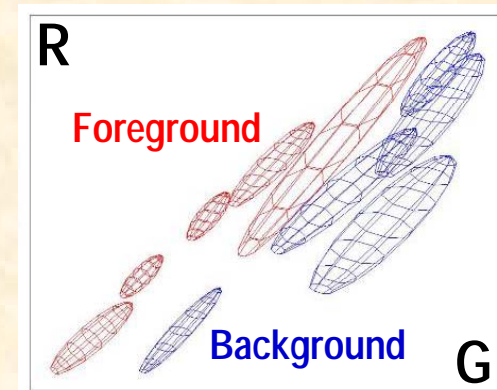
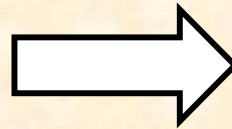
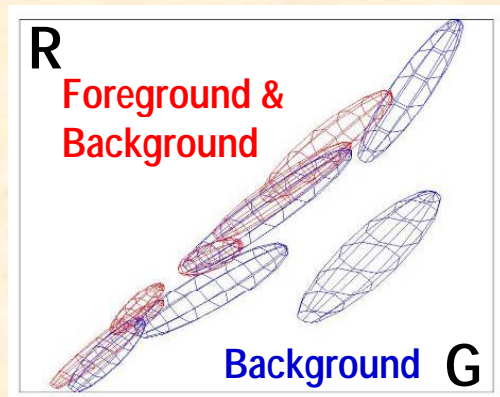


Foreground
&
Background



Gaussian Mixture Model (typically 5-8 components)

Colour Model



Gaussian Mixture Model (typically 5-8 components)

Initially both GMMs overlap considerably, but are better separated after convergence, as the foreground/background labelling has become accurate.

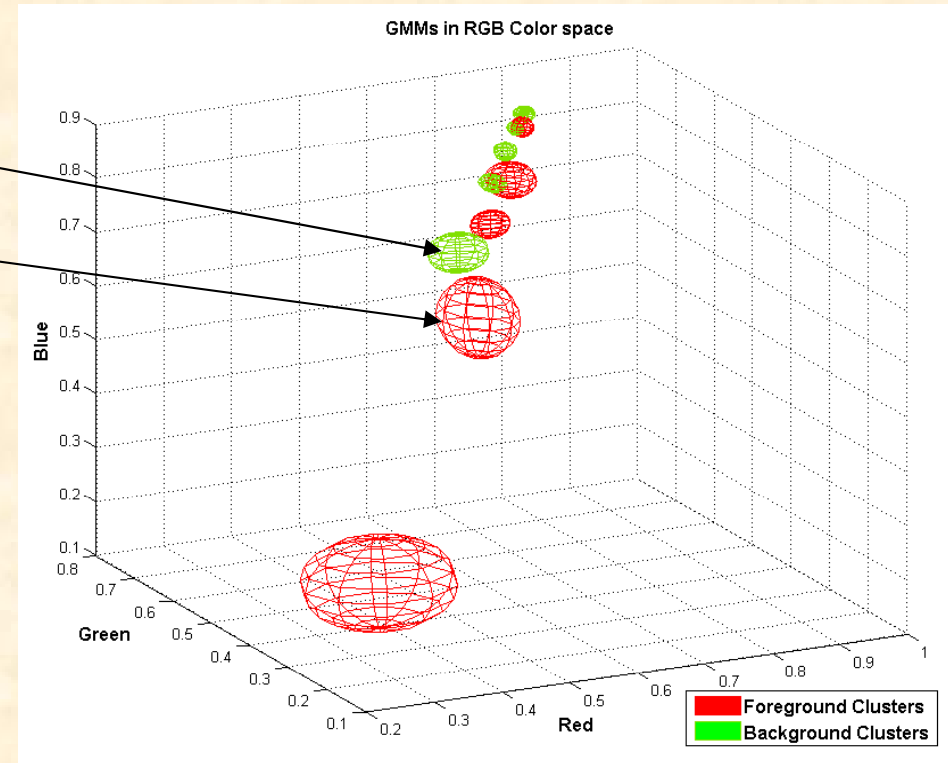
Object Extraction From an Image

Alpha-Matte based Foreground Extraction:



Unknown
foreground

Known
Background



Create GMMs with K components for foreground and background separately

Learn GMMs and perform GraphCut to find tentative classification of foreground and background

Object Extraction From an Image

Source (Eq)



Pixel type (m)	BackGR T-link	Fore –GR T-link
Foreground	0	constant X
Background	constant X	0
Unknown	D_{Fore}	D_{Back}

Sink (Bkg)

$$D(m) = -\log \sum_{i=1}^K \left[\pi_i \frac{1}{\sqrt{\det \Sigma_i}} \exp \left(\frac{1}{2} [z_m - \mu_i]^T \Sigma_i^{-1} [z_m - \mu_i] \right) \right]$$

$$N(m, n) = \frac{\gamma}{\text{dist}(m, n)} e^{-\beta \|z_m - z_n\|^2}$$

Learn GMMs with newly classified set, and repeat the process until classification converges

GrabCut segmentation

1. Define graph

- usually 4-connected or 8-connected

2. Define unary potentials

- Color histogram or mixture of Gaussians for background and foreground

$$unary_potential(x) = -\log\left(\frac{P(c(x); \theta_{foreground})}{P(c(x); \theta_{background})}\right)$$

3. Define pairwise potentials

$$edge_potential(x, y) = k_1 + k_2 \exp\left\{\frac{-\|c(x) - c(y)\|^2}{2\sigma^2}\right\}$$

4. Apply graph cuts

5. Terminate iteration when potential ceases to decrease significantly

6. Else return to 2, using current labels to compute foreground, background models

Object Extraction From an Image

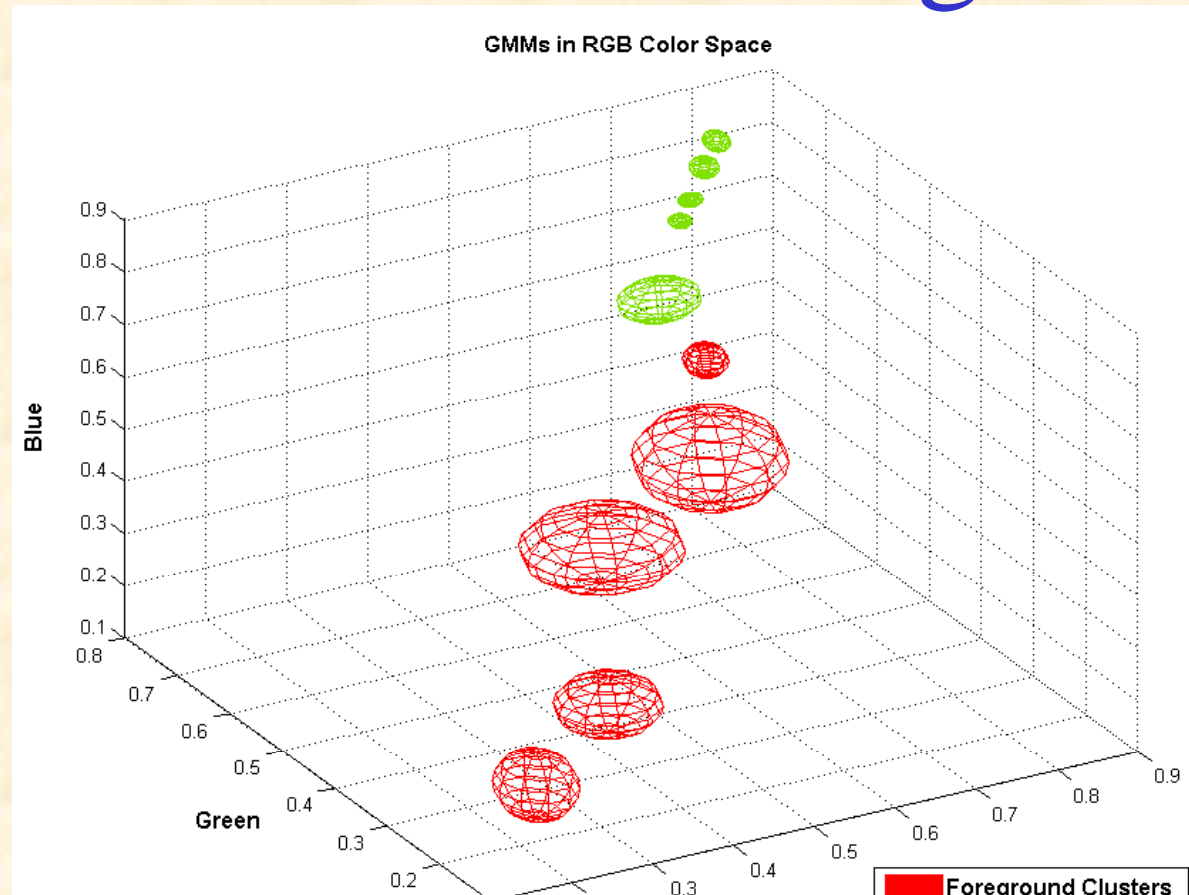


Initial State

$$P(m) = \log \sum_{i=1}^K \left[w_i \frac{1}{\sqrt{\det \Sigma_i}} \times \exp \left(\frac{1}{2} [I_m - \mu_i]^T \Sigma_i^{-1} [I_m - \mu_i] \right) \right]$$

$$\alpha_m = \begin{cases} 1 & \text{if } (P_{fore}(m) - P_{back}(m)) > \tau \\ 0 & \text{if } (P_{back}(m) - P_{fore}(m)) > \tau \\ \text{unknown} & \text{if } |P_{fore}(m) - P_{back}(m)| < \tau \end{cases}$$

$$\min J(\alpha) = \alpha^T L \alpha;$$



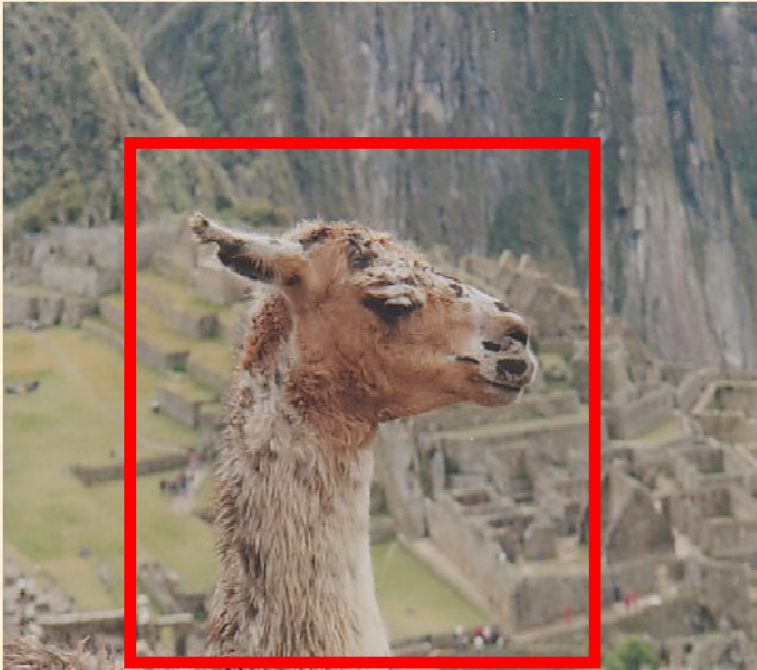
GrabCut segmentation



User provides rough indication of foreground region.

Goal: Automatically provide a pixel-level segmentation.

GrabCut segmentation



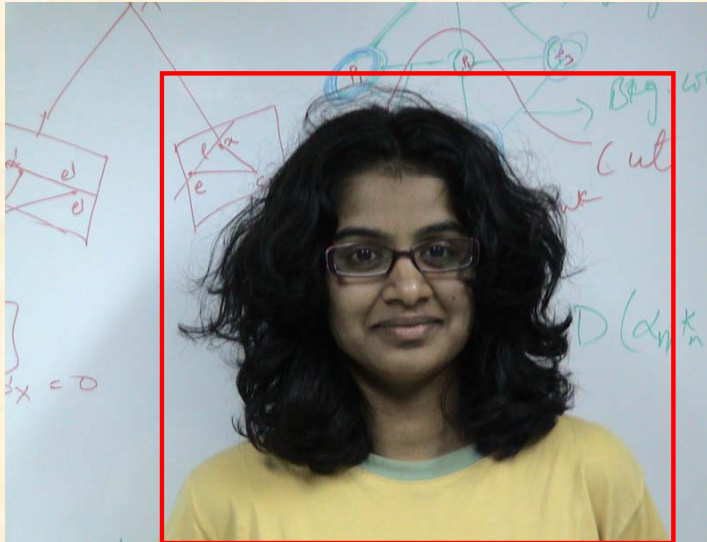
User Input



Result

Object Extraction From an Image

Results:



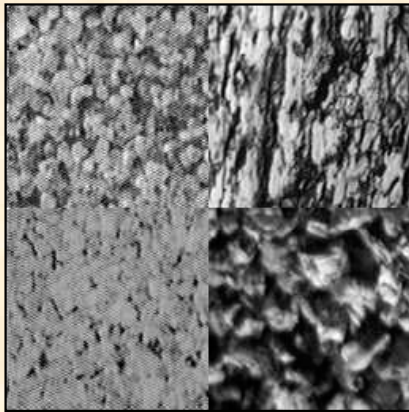
Object Extraction From an Image



Image Segmentation - Combining edge and region information

Example of Image Segmentation (ideal) based on fusion

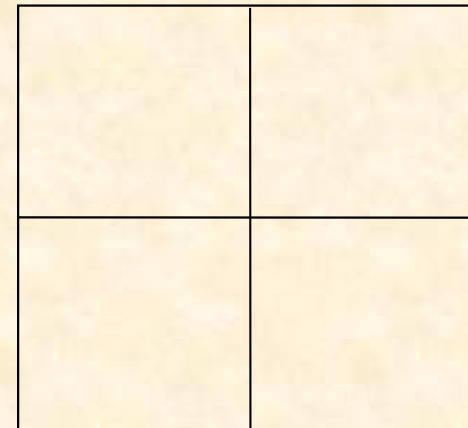
Input Image



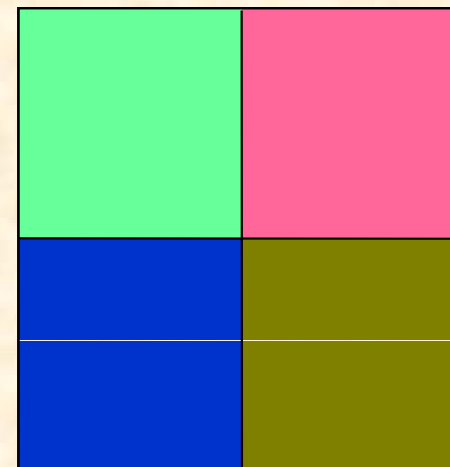
Region Based Segmentation

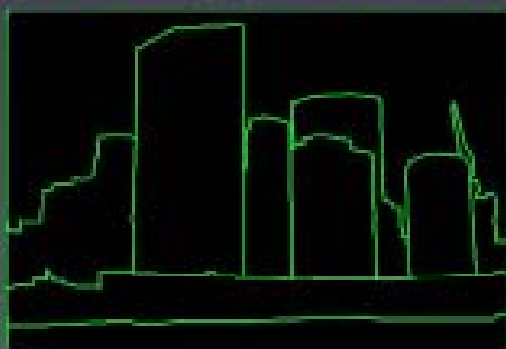
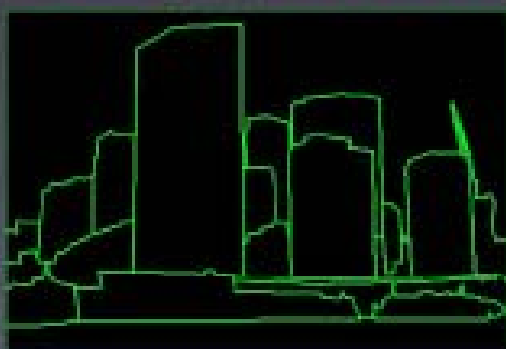
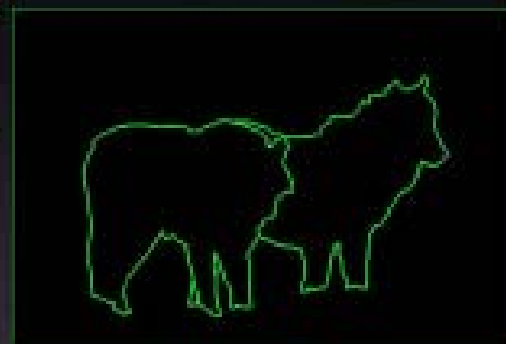
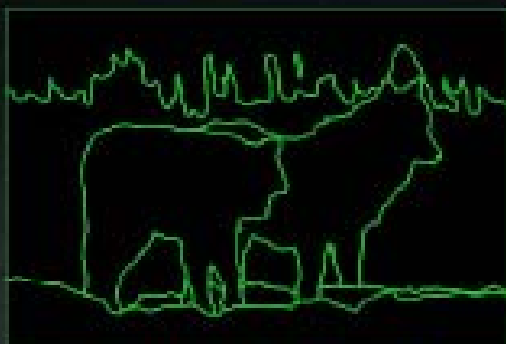
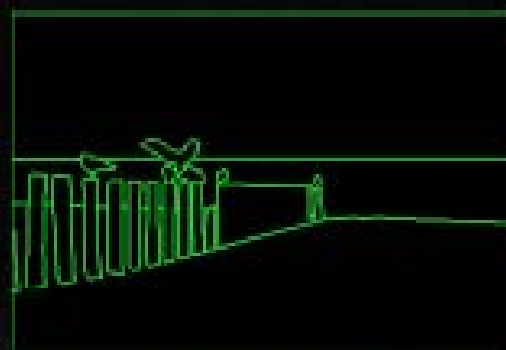
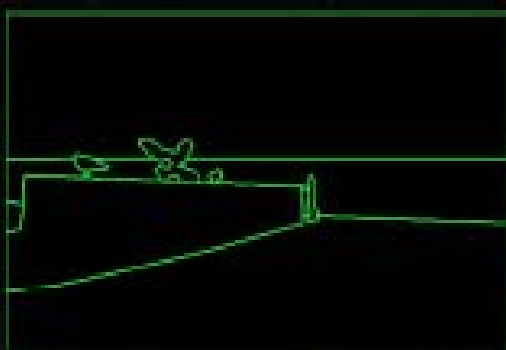
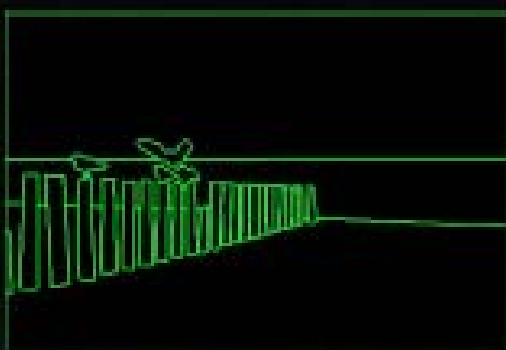
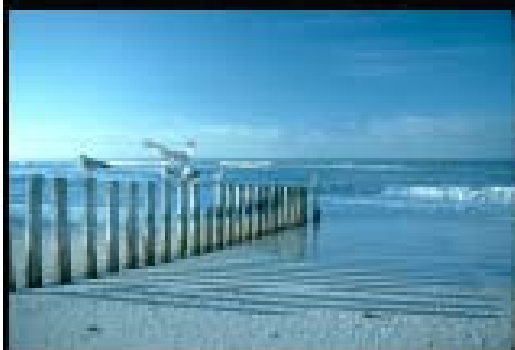


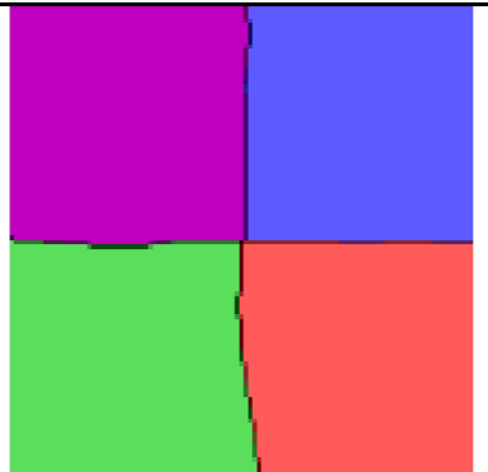
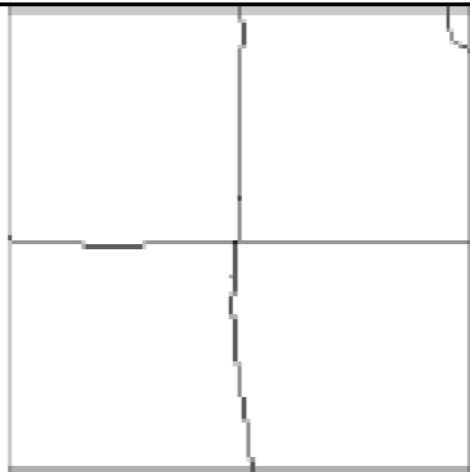
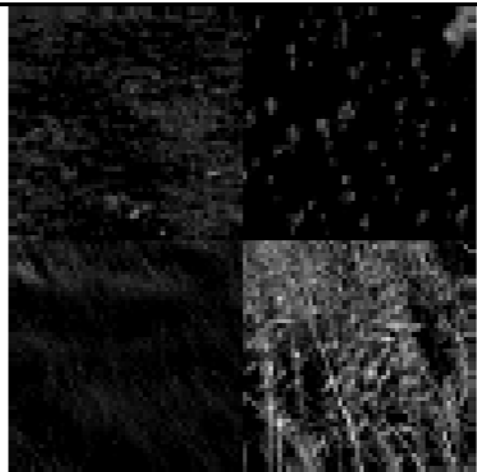
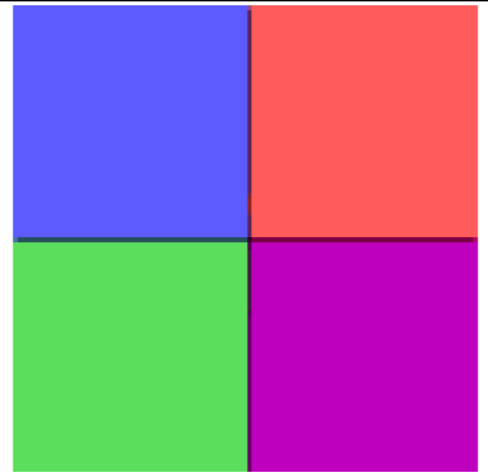
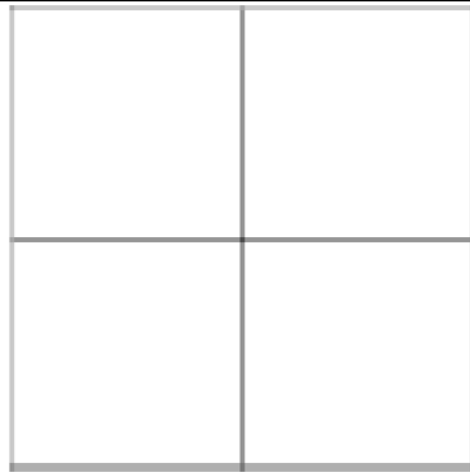
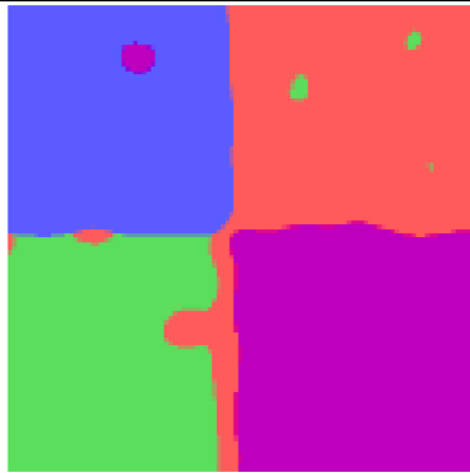
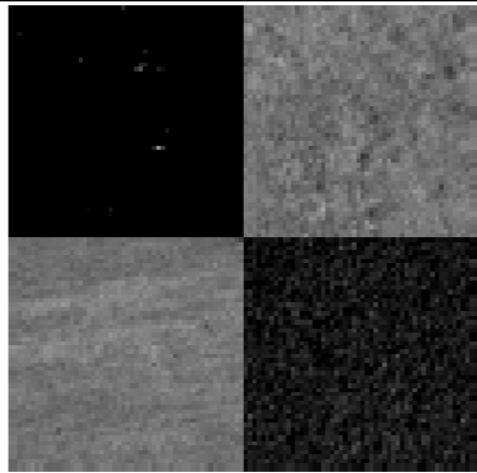
Edge Detection (ideal)



Output segmented
Image
(ideal)







Fusion of Complimentary Information

- **Region-based methods sacrifices resolution and details in the image while calculating useful statistics for local properties – leads to segmentation errors at the boundaries**
- **Difficult to choose initial seed points and stopping criteria in the absence of priori information.**
- **Boundary-based methods fail if image is noisy or if its attributes differ only by a small amount between regions**
- **Both Boundary-based and region based method often fail to produce accurate segmentation results, although the location in which each of these methods fail may not be identical (often complimentary).**
- **Both approaches suffer from a lack of information since they rely on ill-defined hard thresholds, which may lead to wrong decisions**

Integration Techniques

- By using the complementary information of edge-based and region-based information, it is possible to reduce the problems that arise in each individual methods.

1. Embedded Integration

2. Post- processing integration.

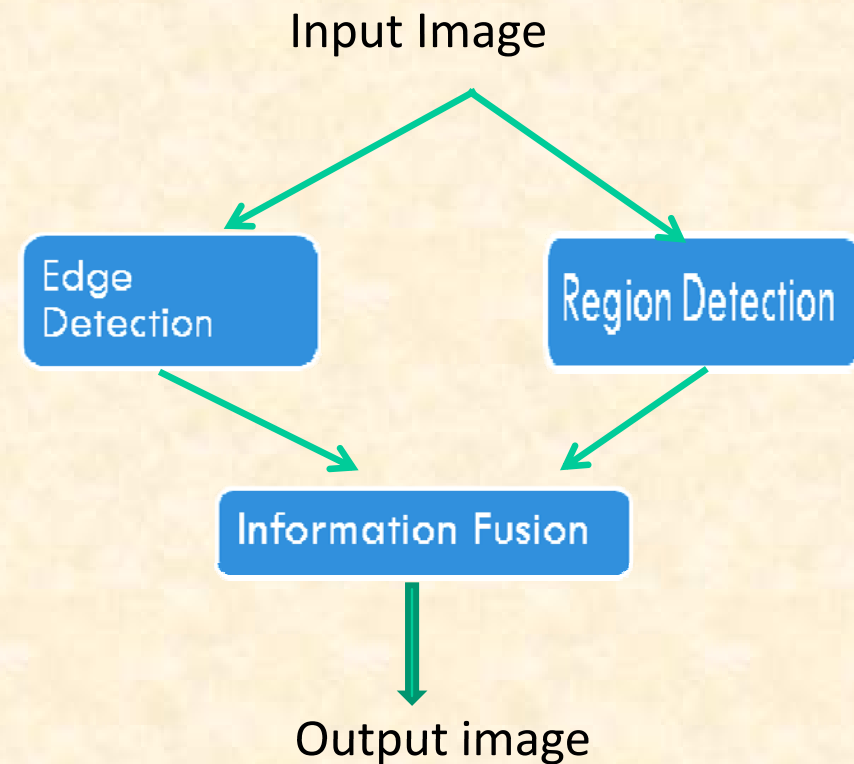
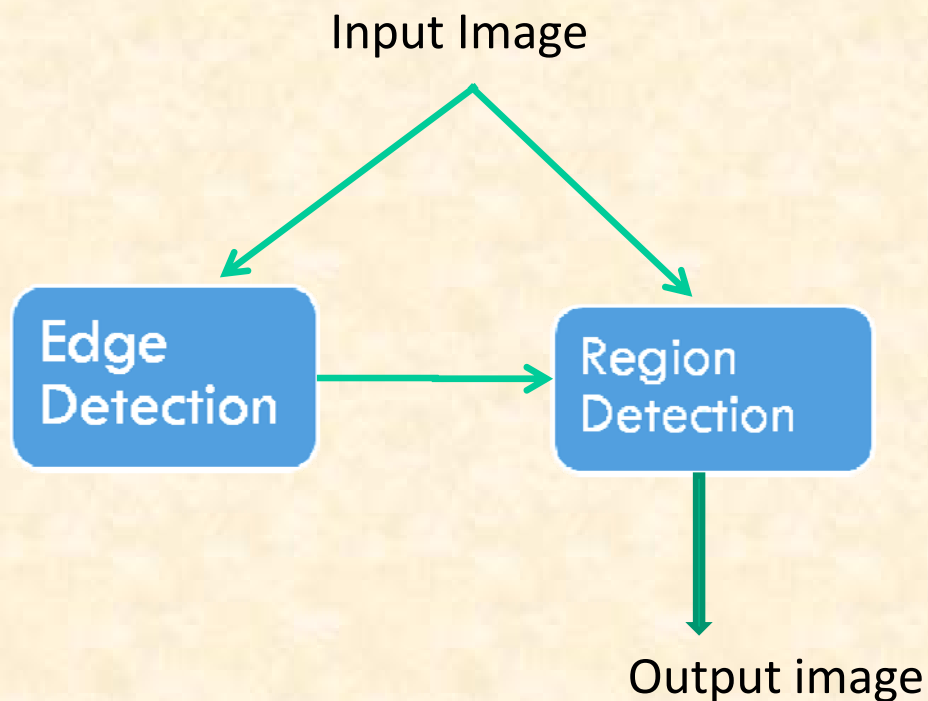
X. Munoz, J. freixenet, X. Cufi, J. Marti,

Strategies for image segmentation combining region and boundary information, Pattern Recognition Letters 24 (2003).

Integration Techniques

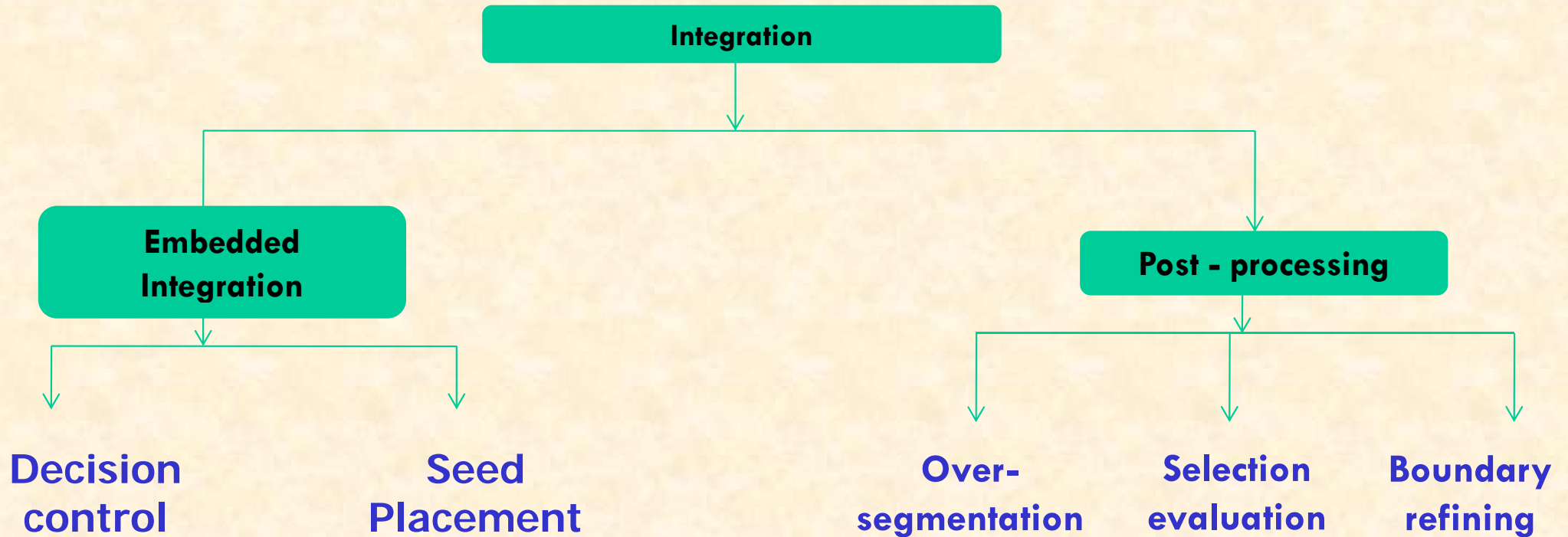
Embedded Integration

Post – Processing Integration



X. Munoz, J. freixenet, X. Cufi, J. Marti,
Strategies for image segmentation combining region and boundary information, Pattern Recognition Letters 24 (2003).

Integration Techniques



- edge information to control the growth of the region.

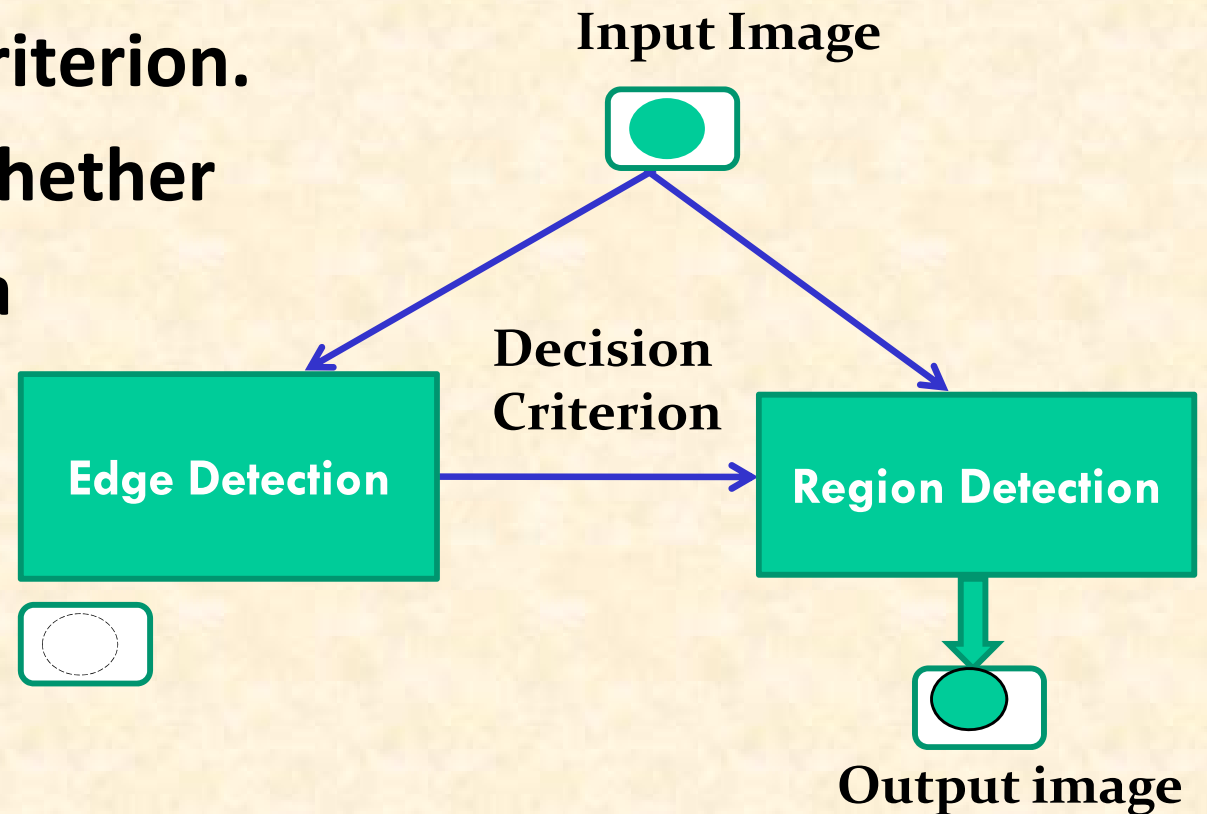
- Use of edge information to place the seed.

Embedded Integration

- Extracted edge information is used within region segmentation algorithm.
- Edge Information can be used in two ways
 1. ***Control of decision criterion*** - edge information is included in the definition of decision criterion which controls the growth of the region.
 2. ***Seed placement guidance*** - edge information used to decide which is the most suitable position to place the seed of the region region growing process.

Decision control-based Region Growing

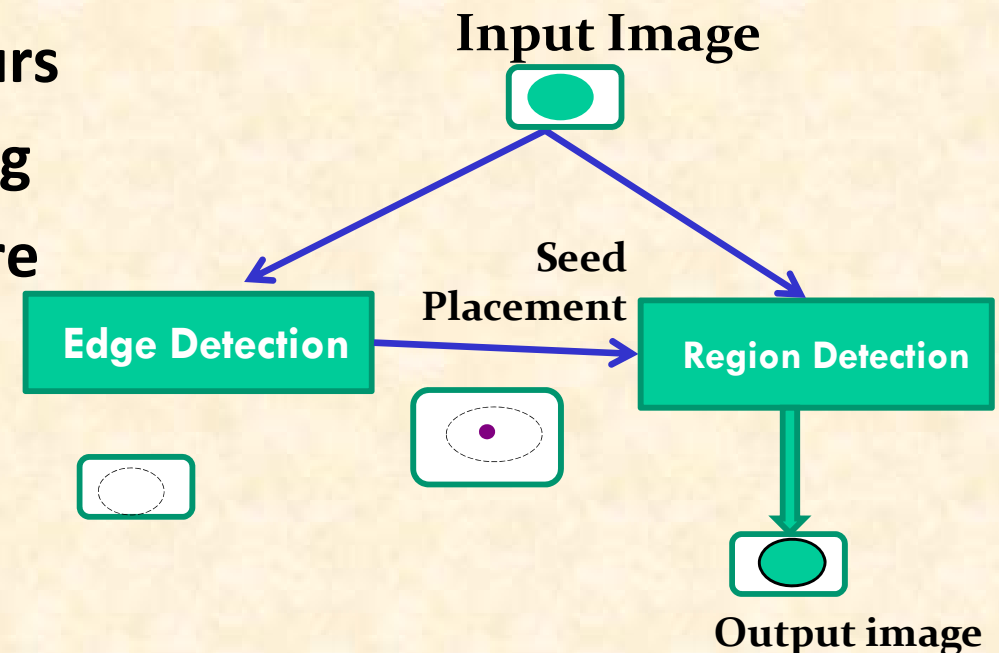
- Choose a starting point or a pixel.
- Add neighboring pixels that are similar based on homogeneity criterion.
- Criterion determines whether or not a pixel belongs to a growing region
 - ▣ Region growing stops if there is a edge
- Merge if there is no edge



Seed placement guidance

- Placement of initial seed points influences the result of region- based segmentation.
- Edge information is used to decide the best position to place the seed point

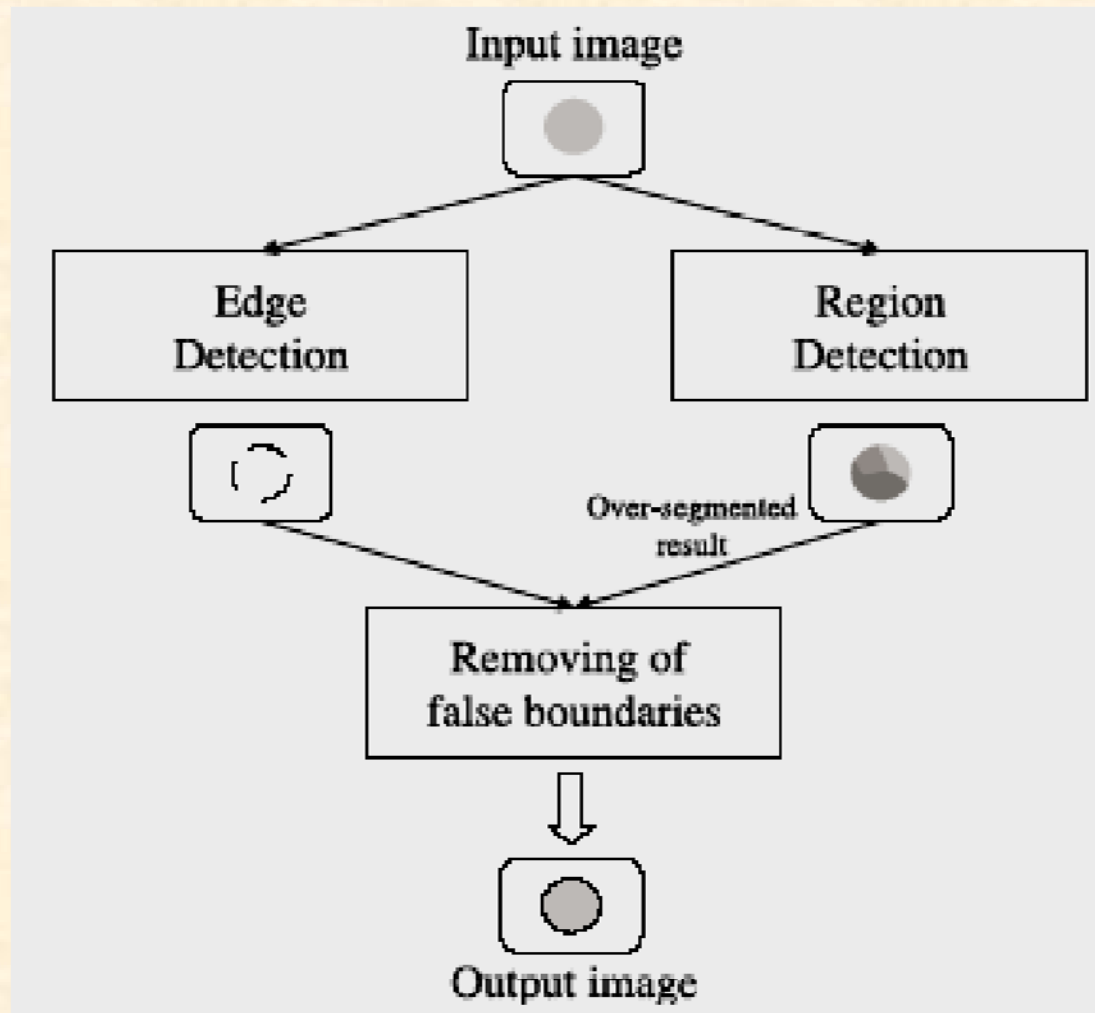
- ▣ Seeds are placed in the core of regions which are far away from contours
- ▣ Disadvantage of region growing and merging – sequential nature



Post-processing Integration

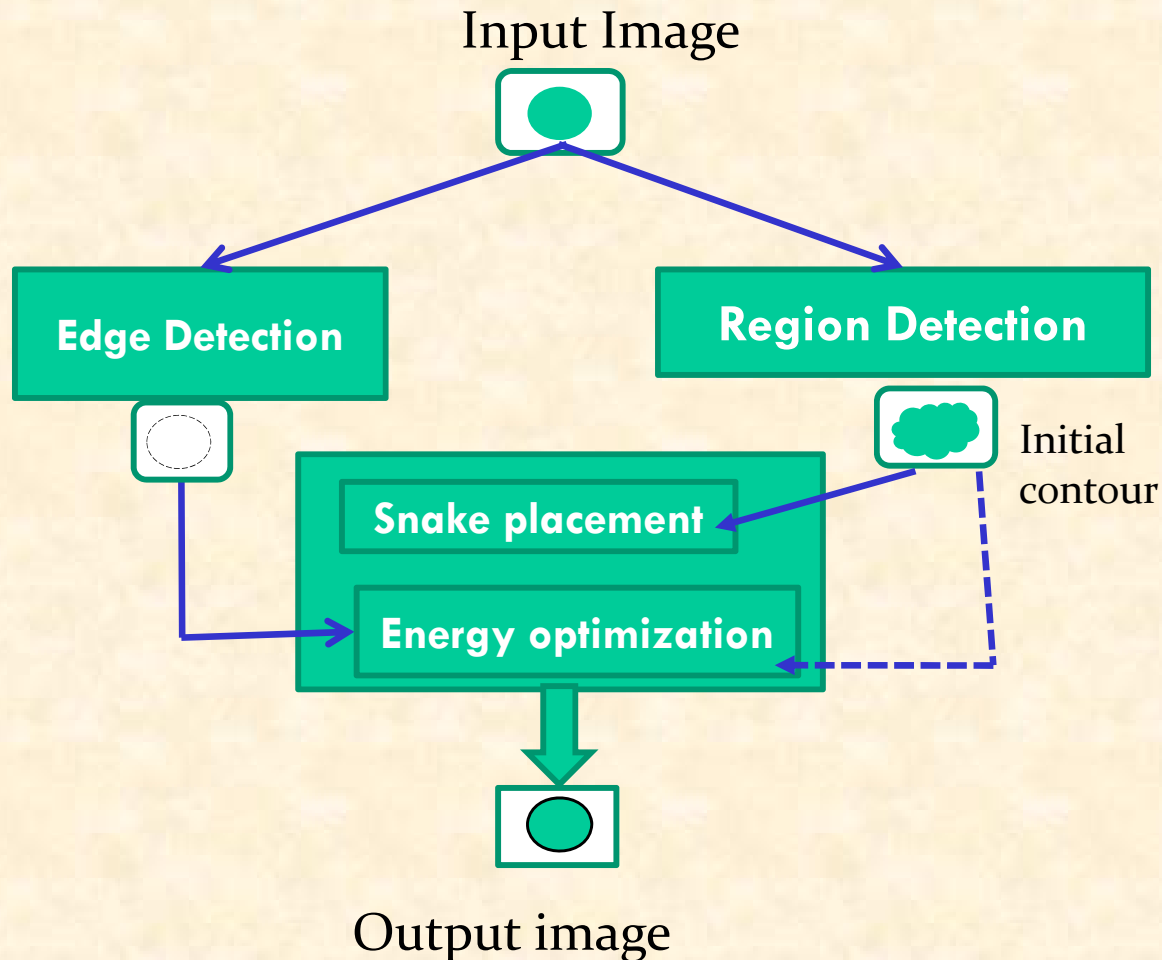
- Combines the map of regions and the map of edge outputs with the aim of providing an accurate and meaningful segmentation.
- Three different approaches
 - (1) Over- segmentation*
 - (2) Boundary refinement*
 - (3) Selection- evaluation*

Over-segmentation



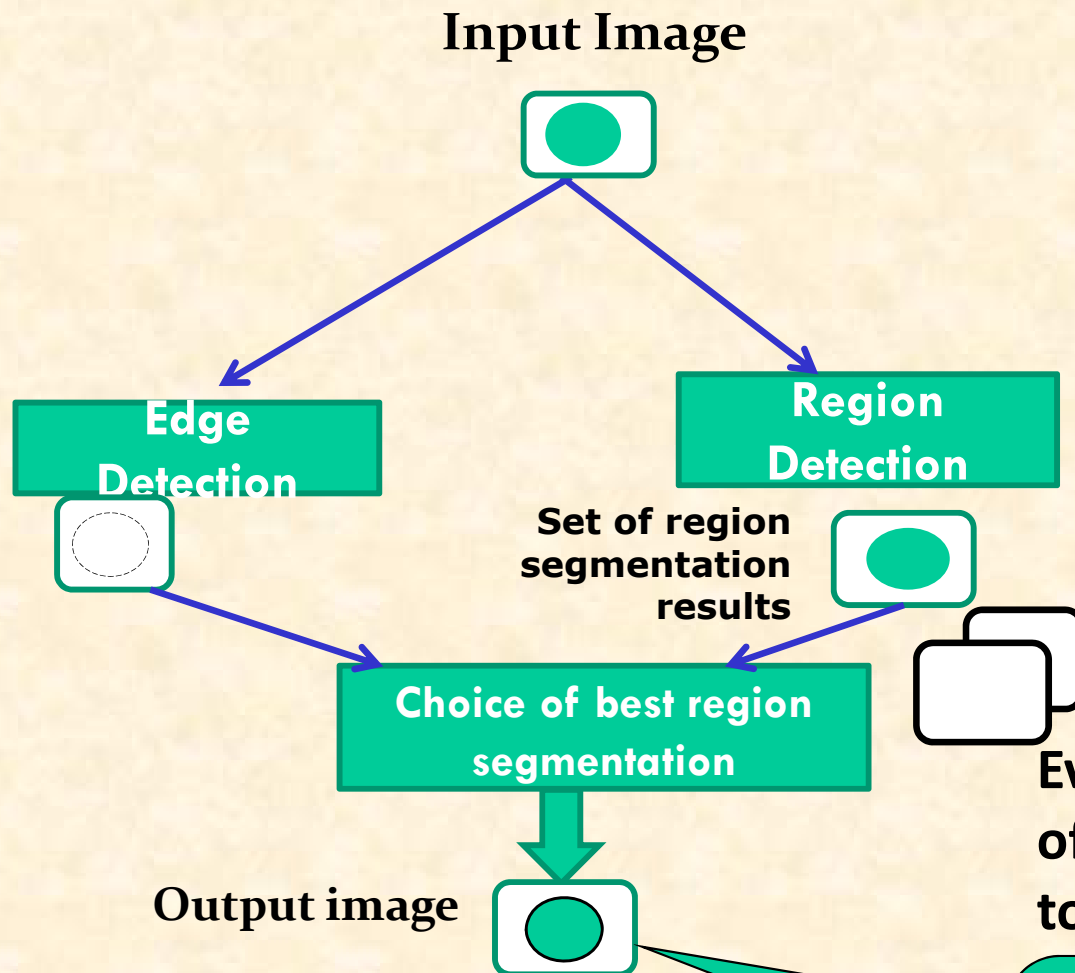
- Region segmentation algorithm may produce false boundaries
- It is compared with edge detection results.
- Eliminate boundaries that are not in Edge detection results
- Only real boundaries are preserved.

Boundary refinement



- A region-based segmentation is used to get an initial estimate of the region.
- It is combined with salient edge information to achieve more accurate representation of the target boundary

Selection- evaluation

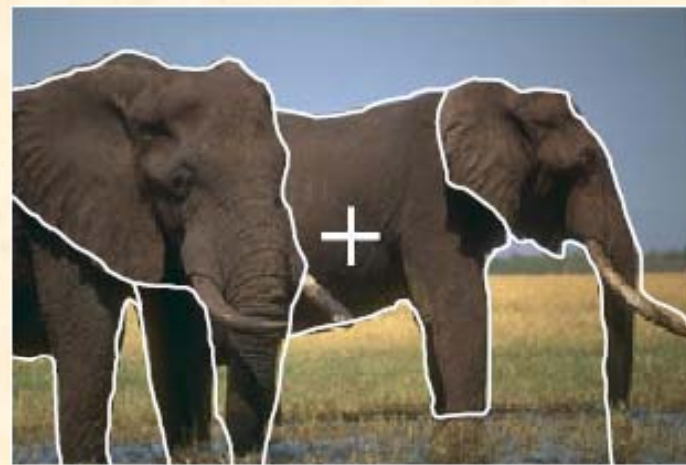
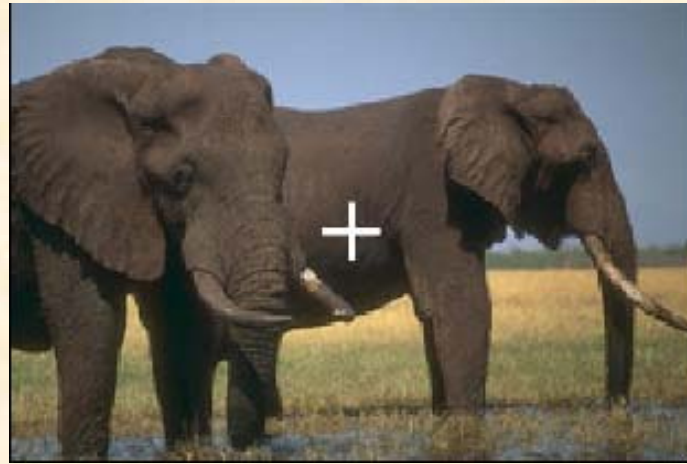


- Different results are achieved by changing parameters and thresholds in a region- segmentation algorithm

- Evaluation function is used to choose the best result obtained.

Evaluation function measures the quality of a region-based segmentation according to its consistency with the edge map

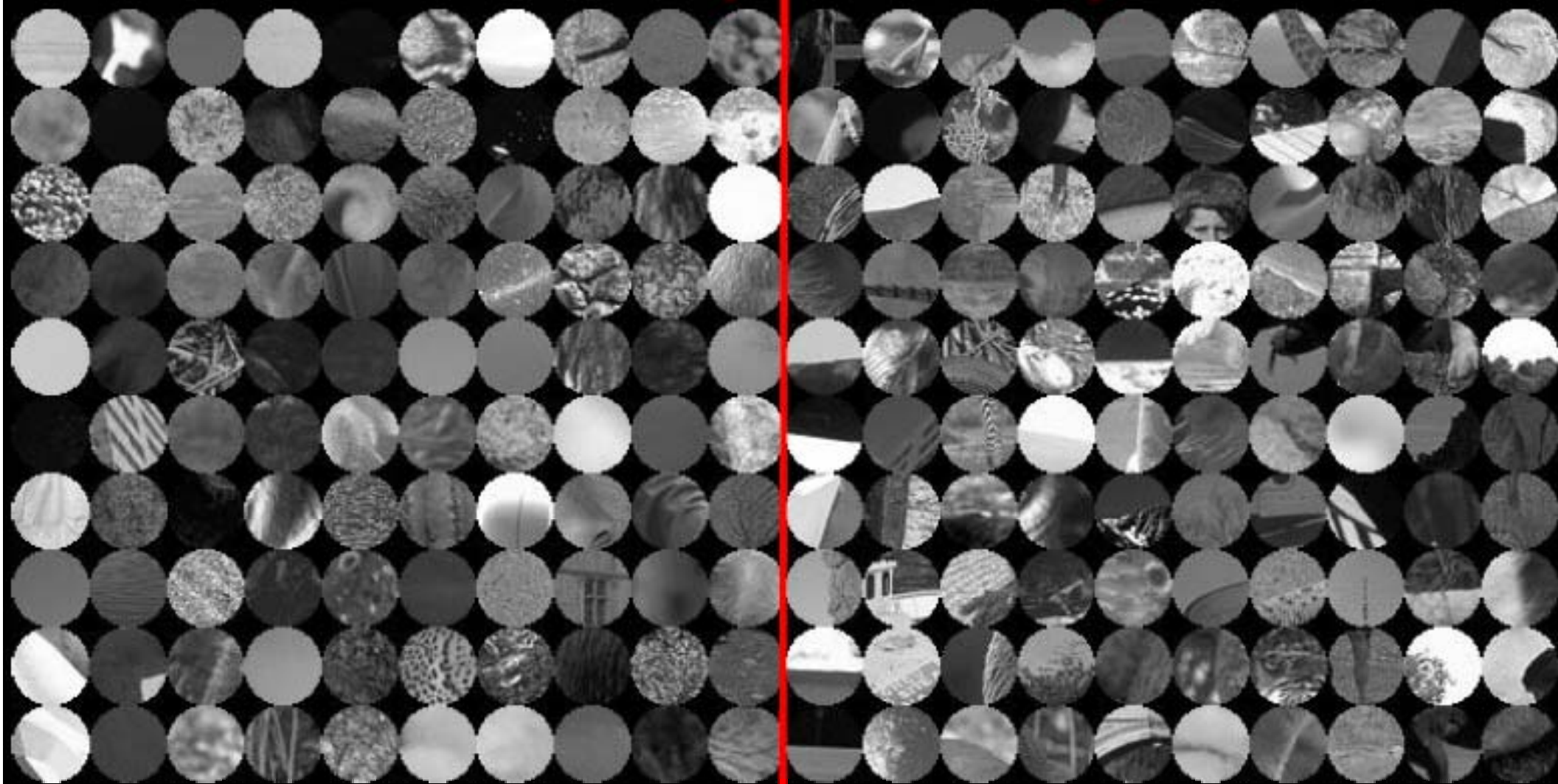
The best region segmentation is the one where the region boundaries correspond most closely to the contours



How good are humans locally?

Off-Boundary

On-Boundary

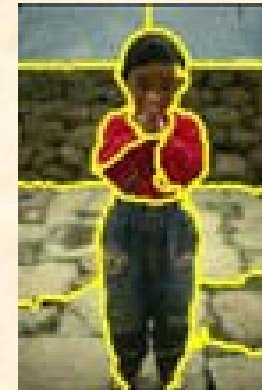
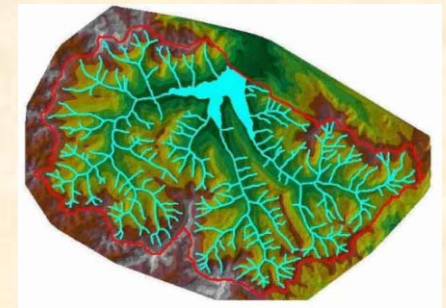
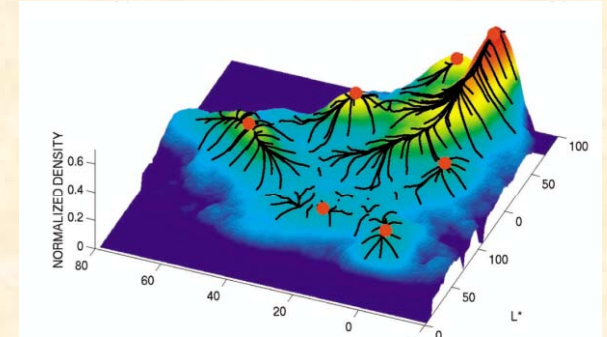


Modern methods for Image segmentation involve:

- **Multi-resolution and multi-channel features**
- **Feature fusion (selection) techniques**
- **Multi-classifier decision combination**
- **HMM, GMM, CRF- and GMRF-based techniques**
- **Artificial Neural Networks – SOM and Hopfield/Bolztmann**
- **Watershed transform**
- **Grabcut (Graph cut); normalized cut.**
- **Snakes (Active Contours); Snake-cut;**
- **Parametric Distributional clustering**
- **Deformable Templates, AAM, ASM**
- **Decision Trees and hierarchical analysis**
- **Probabilistic approaches**
- **Neuro-fuzzy and soft-computing techniques – ACO, PSO etc.**
- **Mean-Shift**

Segmentation

- Mean-shift segmentation
 - Flexible clustering method, good segmentation
- Watershed segmentation
 - Hierarchical segmentation from soft boundaries
- Normalized cuts
 - Produces regular regions
 - Slow but good for oversegmentation
- MRFs with Graph Cut
 - Incorporates foreground/background/object model and prefers to cut at image boundaries
 - Good for interactive segmentation or recognition



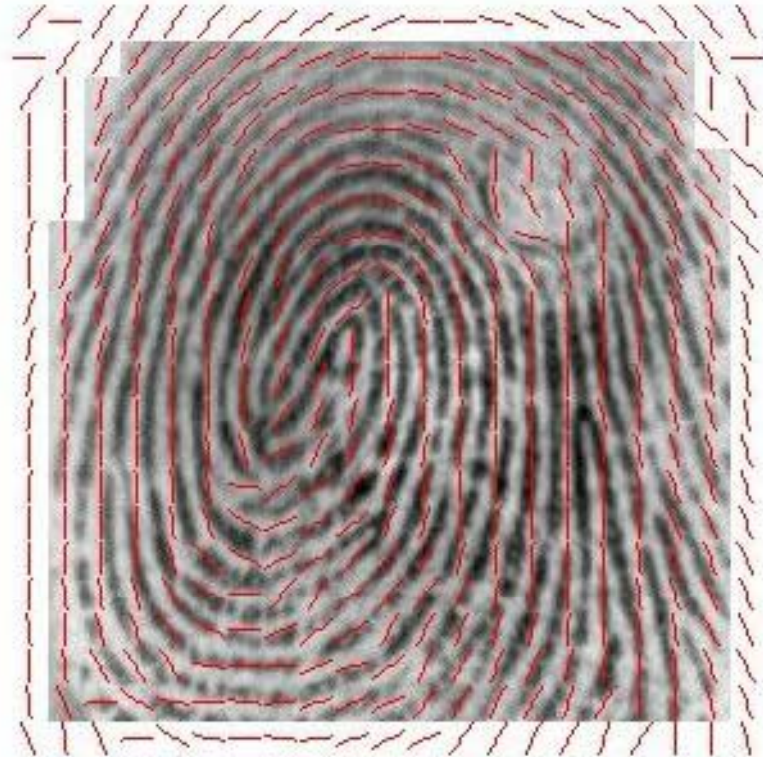


<http://www.cs.berkeley.edu/~fowlkes/BSE/>



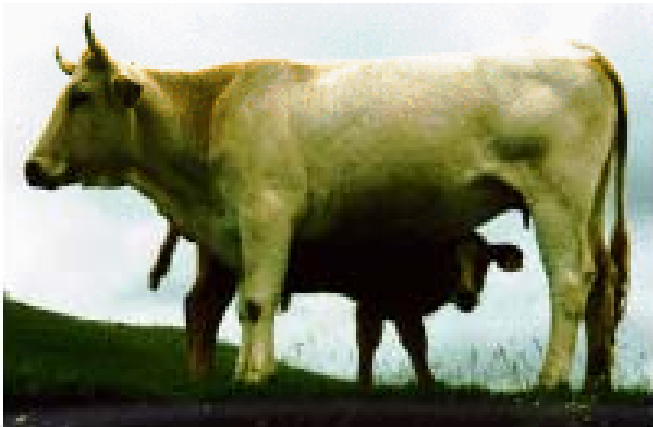


**Why not
OK ?**

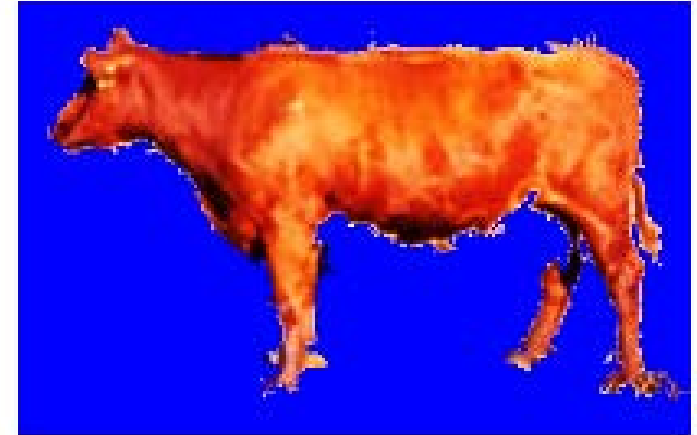
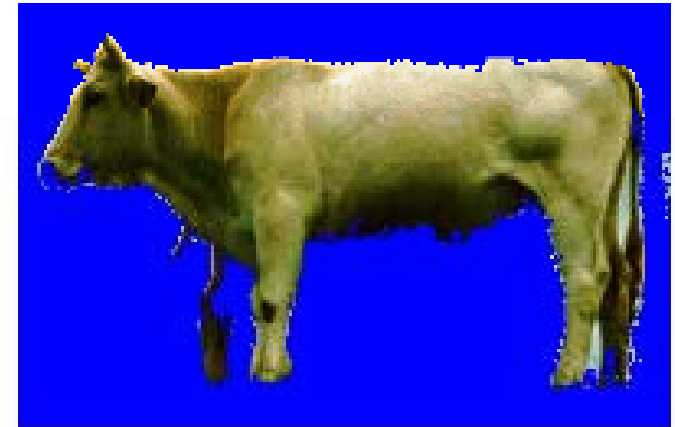


OK

Image



Segmentation



**Purposeful image segmentation – involves object
Detection and recognition modules (non-trivial tasks)**

Think on your own now:

Image segmentation and object recognition are two intertwined topics. Does segmentation leads to recognition, or recognition leads to segmentation?

Several proposals have emerged recently, some uses top-down recognition process to guide image segmentation, while others use bottom-up segmentation to guide object recognition. The results have been surprisingly good in their limited domain.

Regardless one's philosophical stand on this question, it is undeniable a tight connection exists between them.

We will come back to OBJ. RECOGN. in this course later, and then revisit this question.

REFERENCES

- **Jianbo Shi and Jitendra Malik; Normalized Cuts and Image Segmentation;, Member, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 22, NO. 8, AUGUST 2000, pp 888-905.**
- **B. Leibe, A. Ettlin, B. Schiele. Learning Semantic Object Parts for Object Categorization, Image and Vision Computing, Vol. 26, No. 1, pp. 15-26, 2008.**
- **Pedro F. Felzenszwalb and Ross B. Girshick and David McAllester and Deva Ramanan, "Object Detection with Discriminatively Trained Part-Based Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, pp. 1627-1645, 2010.**
- **B. Leibe, A. Leonardis and B. Schiele. Robust object detection with interleaved categorization and segmentation; International Journal of Computer Vision 77 (1-3), pp. 259-289, 2007.**
- **Anat Levin and Yair Weiss; Learning to Combine Bottom-Up and Top-Down Segmentation"; International Journal on Computer Vision, Vol. 81, pp. 105–118, 2009.**
- **Lalit Gupta, Utthara Gosa Mangai and Sukhendu Das, "Integrating Region and Edge Information for Texture Segmentation using a modified Constraint Satisfaction Neural Network", Journal of Image and Vision Computing, Vol.26, No.8, 1106-1117, August 2008.**

- **Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9 (1): 62–66.**
- **Lie, Wen-Nung, Automatic target segmentation by locally adaptive image thresholding; 1995 IEEE Transactions on Image Processing 4 (7), pp. 1036-1041.**
- **M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. International Journal of Computer Vision, 1(4):321–331, January 1988.**
- **Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: Interactive foreground extraction using iterated graph-cuts. ACM Transactions on Graphics, 23(3):309–34, 2004.**
- **S.X. Yu and J. Shi. Multiclass spectral clustering. Int'l Conf. on Computer Vision, 2003.**
- **M. Meila and L. Xu. Multiway cuts and spectral clustering. U. Washington Tech Report, 2003.**

Smith and J. Blinn. "Blue screen matting", SIGGRAPH, Aug.1996, pp.259 – 268.

Yuri Boykov, Vladimir Kolmogorov "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision". IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 26, no. 9, pp. 1124-1137, Sept. 2004.

Levin D. Lischinski and Y. Weiss. "A Closed Form Solution to Natural Image Matting". IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 2006.

Ruzon, M., and Tomasi, C. "Alpha estimation in natural images". In Proc. IEEE Conf. Comp. Vision and Pattern Recognition, June 2000, vol 1, pp.18-25.

X. Munoz, J. Freixenet, X. Cu, J. Marti, Strategies for image segmentation combining region and boundary information, Pattern Recognition, 24, 375-292, 2003.



Bruegel, 1564