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.**

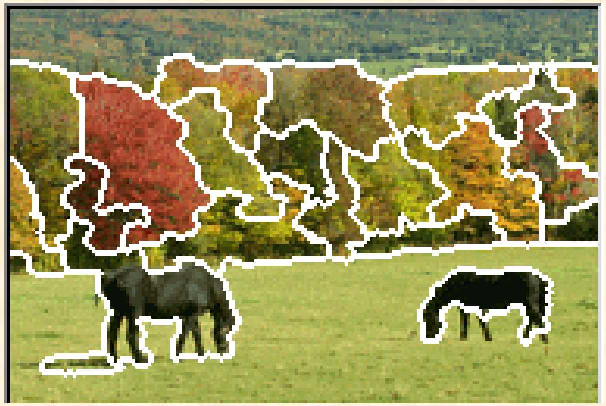
Region-based segmentation methods attempt to partition or group regions according to common image properties. These image properties consist of :

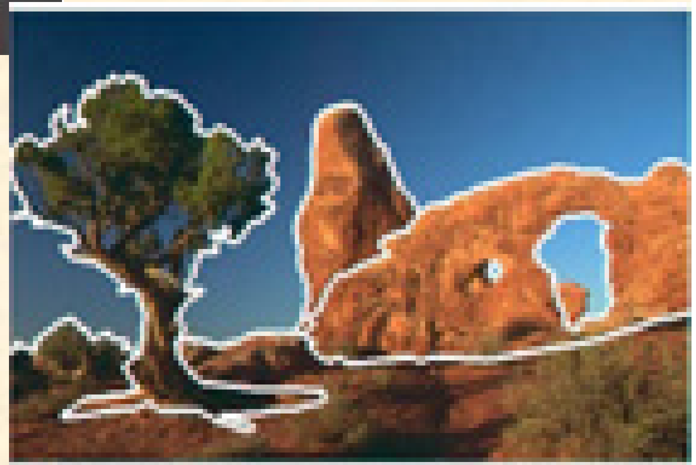
- **Intensity values from original images, or computed values based on an image operator**
- **Textures or patterns that are unique to each type of region**
- **Spectral profiles that provide multidimensional image data**

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:



10 *Image Segmentation* 567

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.

Purpose:

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

$$R = \bigcup_{i=1}^{N} R_i;\; R_1 \bigcap R_j = \Phi,\; i \neq j$$

$$H(R_i) = True;\; i = 1,2,...,N;$$

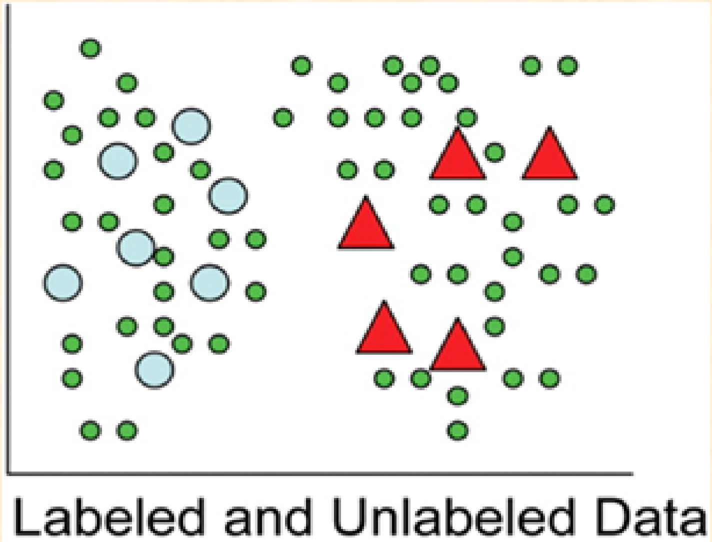
When, R_i and R_j are adjacent:

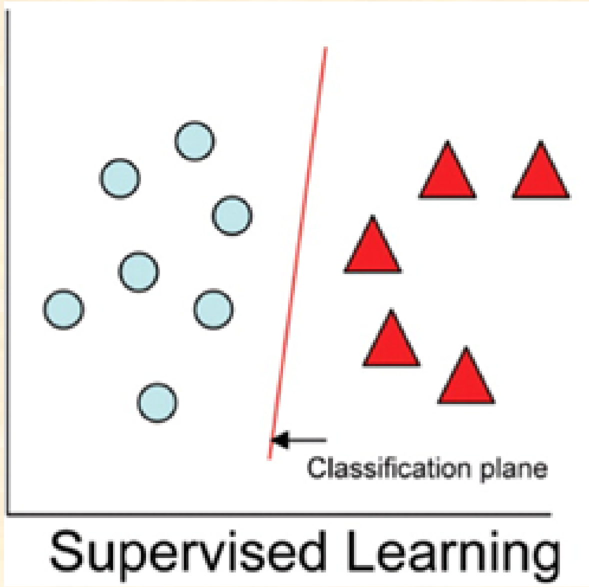
$$H(R_i \bigcup R_j) = False,\; i \neq j;$$

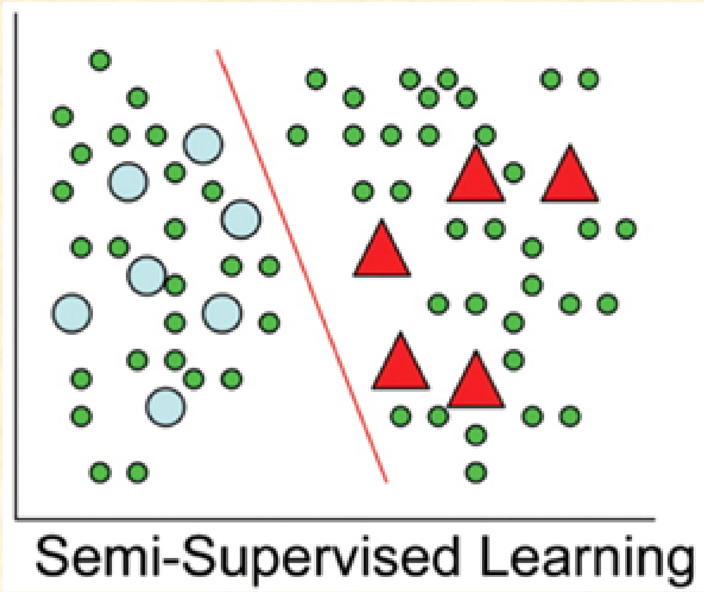
Typical algorithms of clustering data:

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

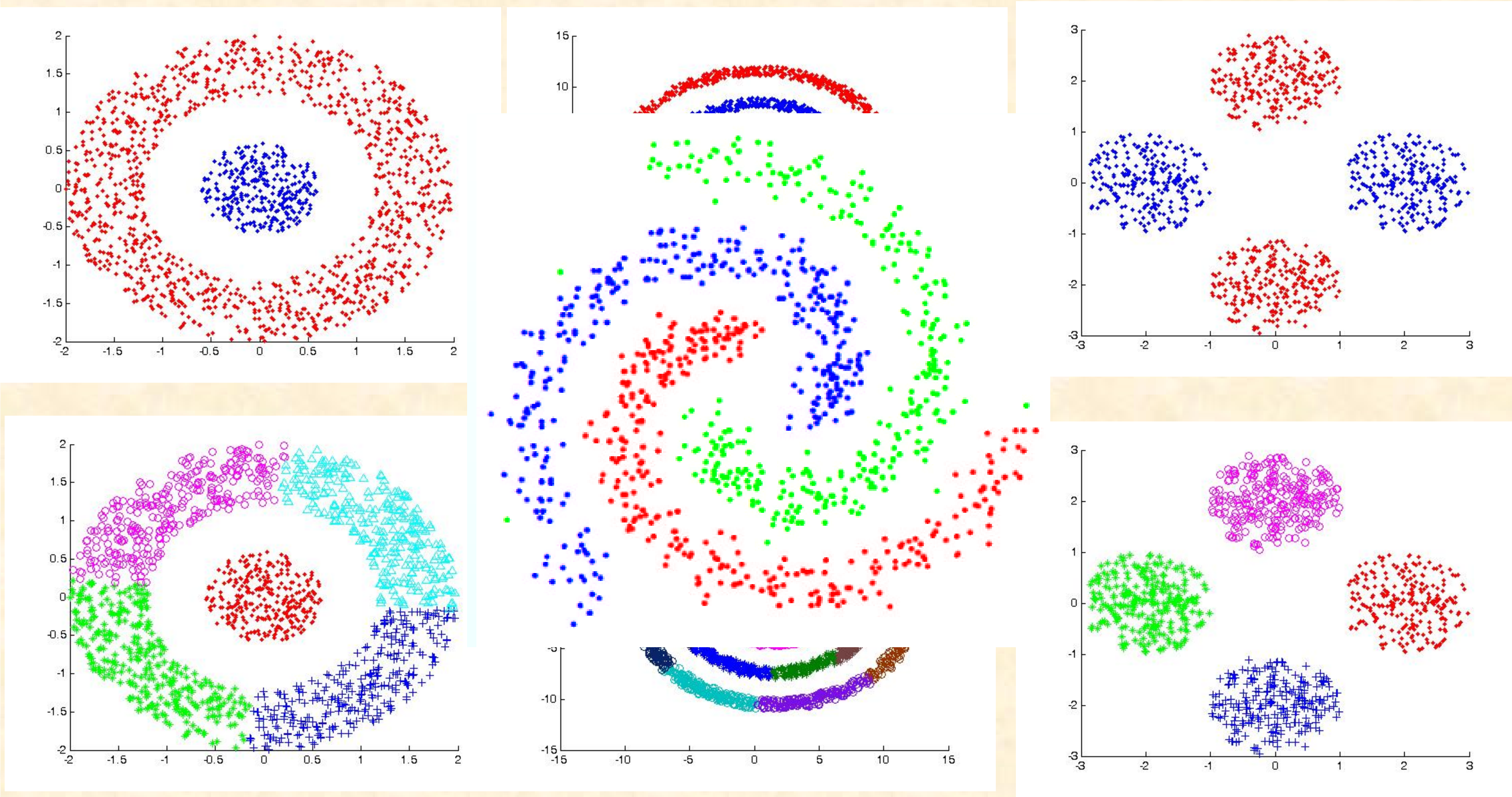
Clusters in Feature space







EXAMPLES of CLUSTERING



Categories of Image Segmentation Methods

- **Histogram-Based Methods**
- **Edge Detection Methods**
- **Region Growing Methods**
- **Clustering Methods**

- **Level Set Methods**
- **Graph Partitioning Methods**
- **Watershed Transformation**
- **Neural Network models**
- **Multi-scale Segmentation**
- **Probabilistic modeling**

- **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.**

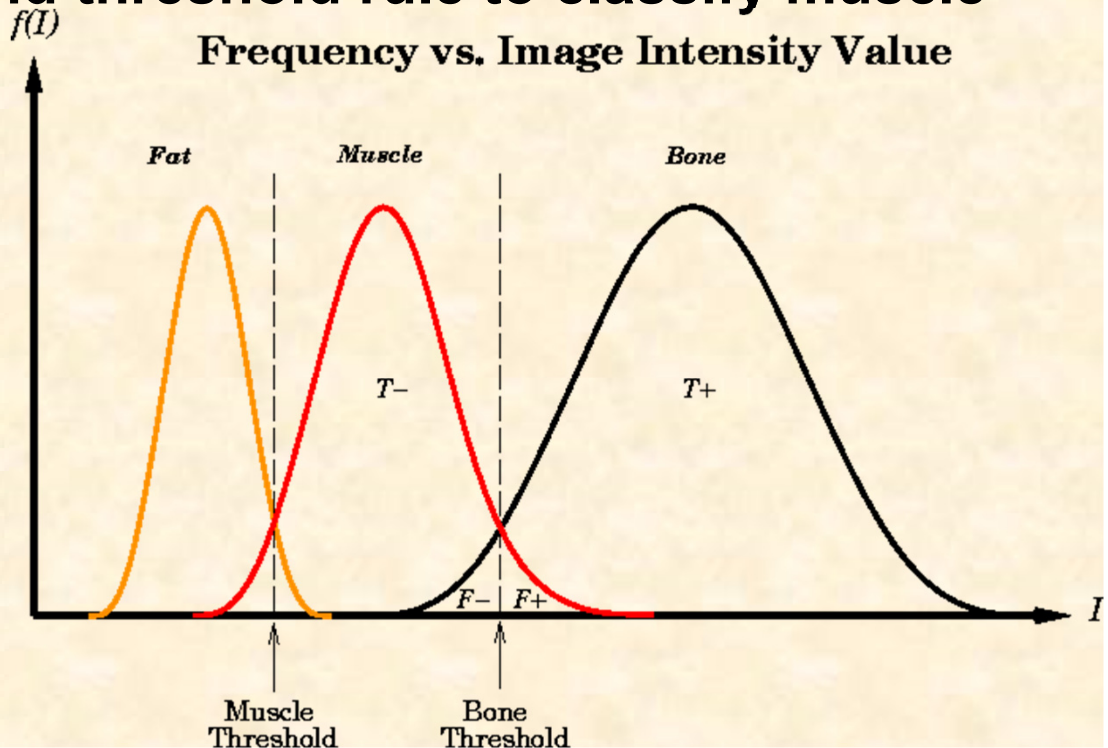
Thresholding is the simplest way to perform segmentation, and it is used extensively in many image processing applications. Thresholding is based on the notion that regions corresponding to different regions can be classified by using a range function applied to the intensity values of image pixels. 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).

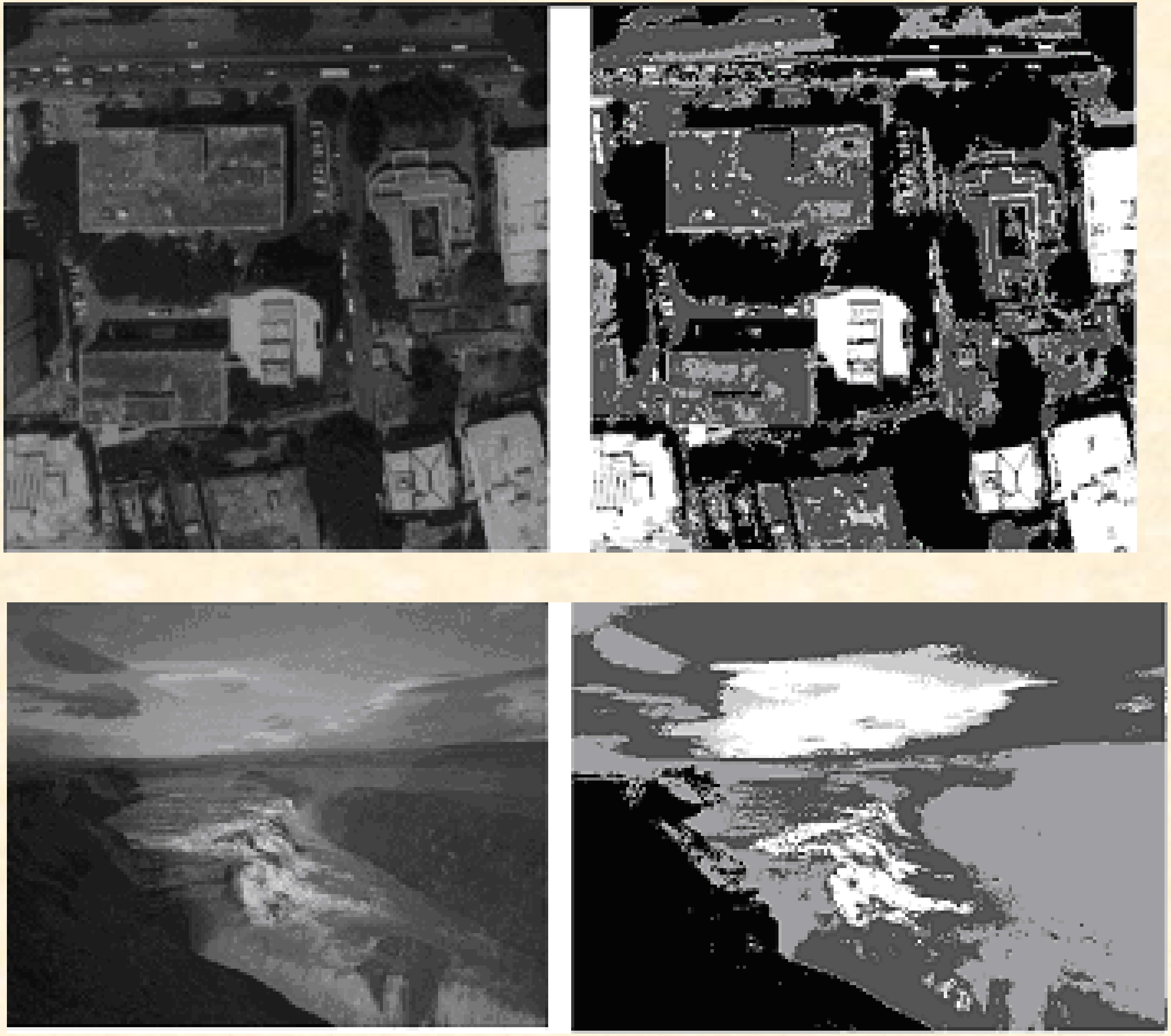
For example, given the histogram of a two-dimensional medical image I(x,y) , we can define a simple threshold rule to classify bony and fat tissues or a compound threshold rule to classify muscle tissue:

If, I(x,y) > T1 => Bony

If, I(x,y) < T0 => Fat

If, T0 < I(x,y) < T1 => 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:

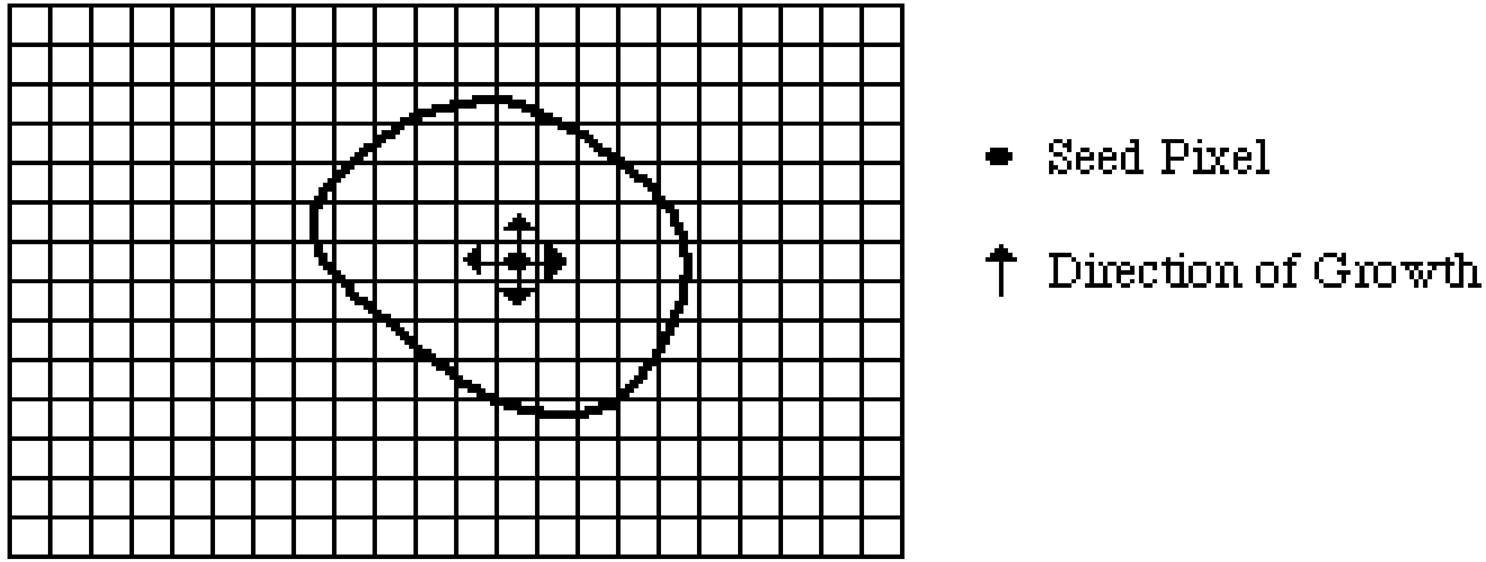
- gray level
- color
- texture
- shape
- model

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

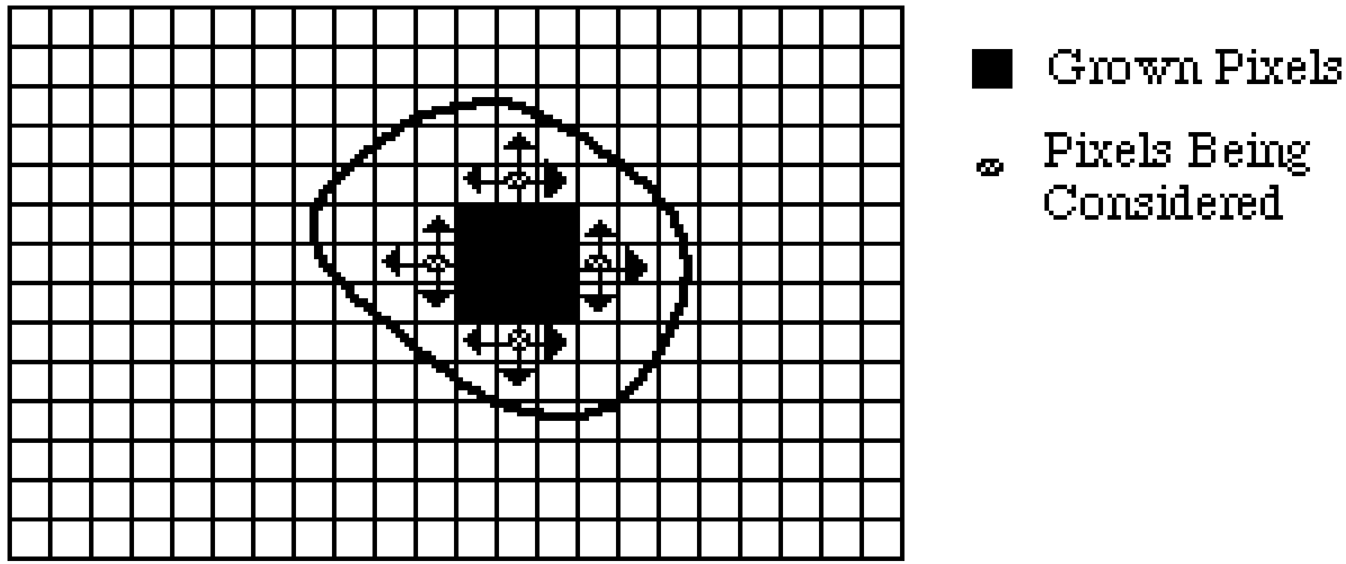
$$R = \bigcup_{i=1}^{N} R_i; \; R_1 \bigcap R_j = \Phi, \; i \neq j$$

$$H(R_i) = True; \; i = 1,2,\ldots, N;$$

When, R_i and R_j are adjacent: $$H(R_i \bigcup R_j) = False, \; i \neq j;$$



(a) Start of Growing a Region



(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.

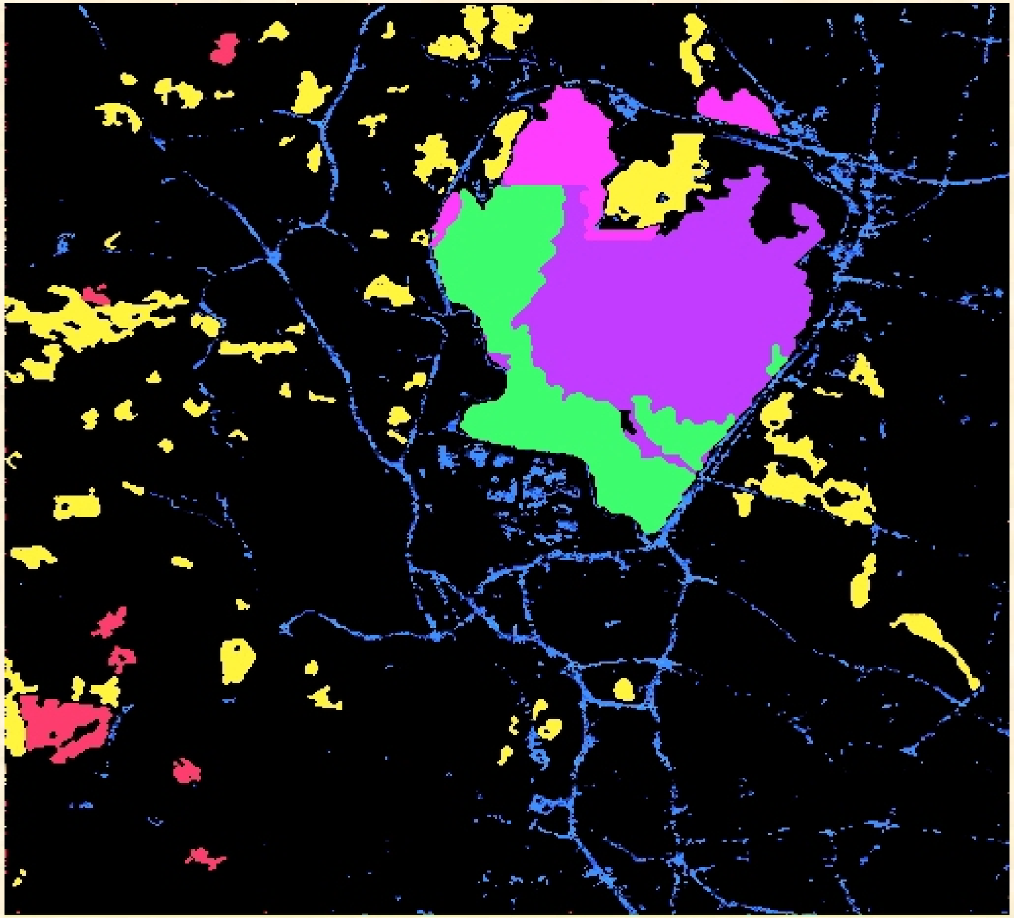
This can have several undesirable effects:

- 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.

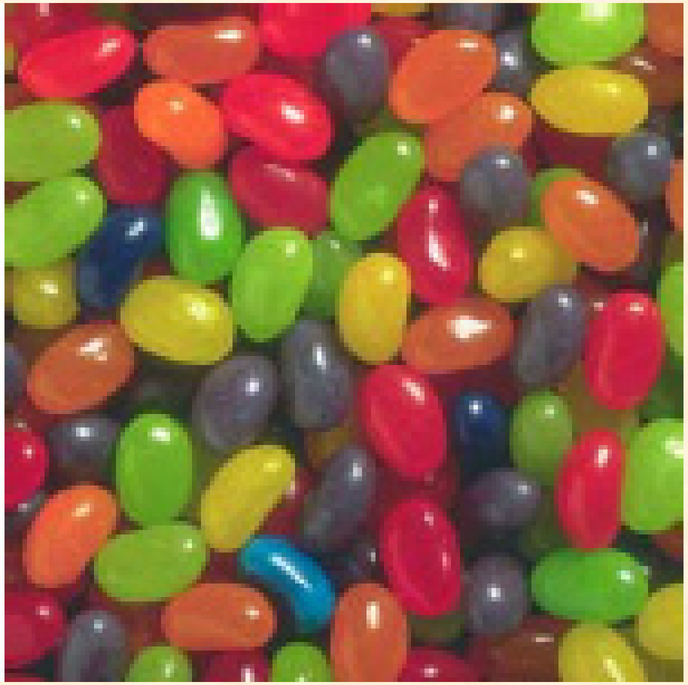
To counter the above problems, *simultaneous region growing* techniques have been developed.

- 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 (or pixels) 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.

Let's look at Pedro (MIT), Daniel's (Cornell) IJCV-2004:

Efficient Graph-Based Image Segmentation

For any region R, its *internal difference* is defined as the largest edge weight in the region's minimum spanning tree,

$$Int(R) = \max_{e \in MST(R)} w(e). \qquad (5.20)$$

For any two adjacent regions with at least one edge connecting their vertices, the difference between these regions is defined as the minimum weight edge connecting the two regions, as:

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

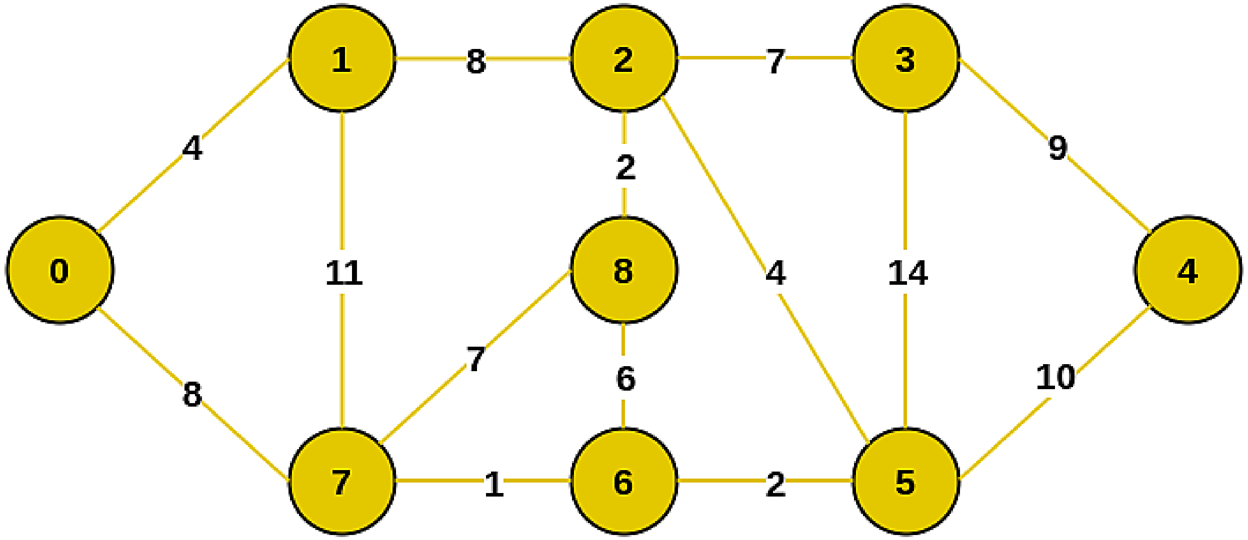
pairwise comparison predicate:

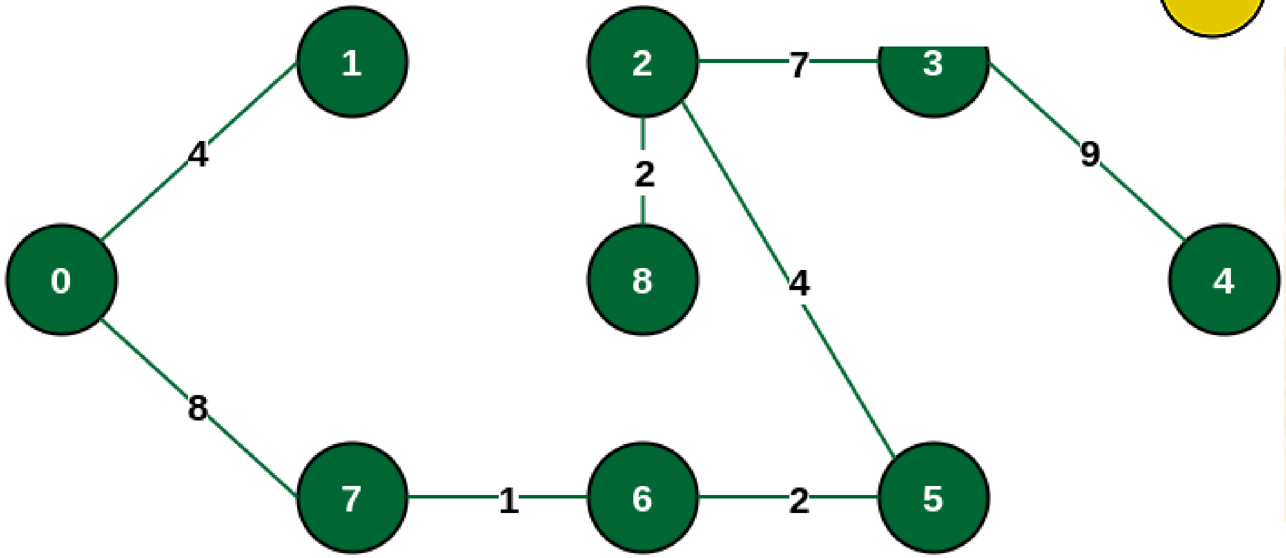
$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$

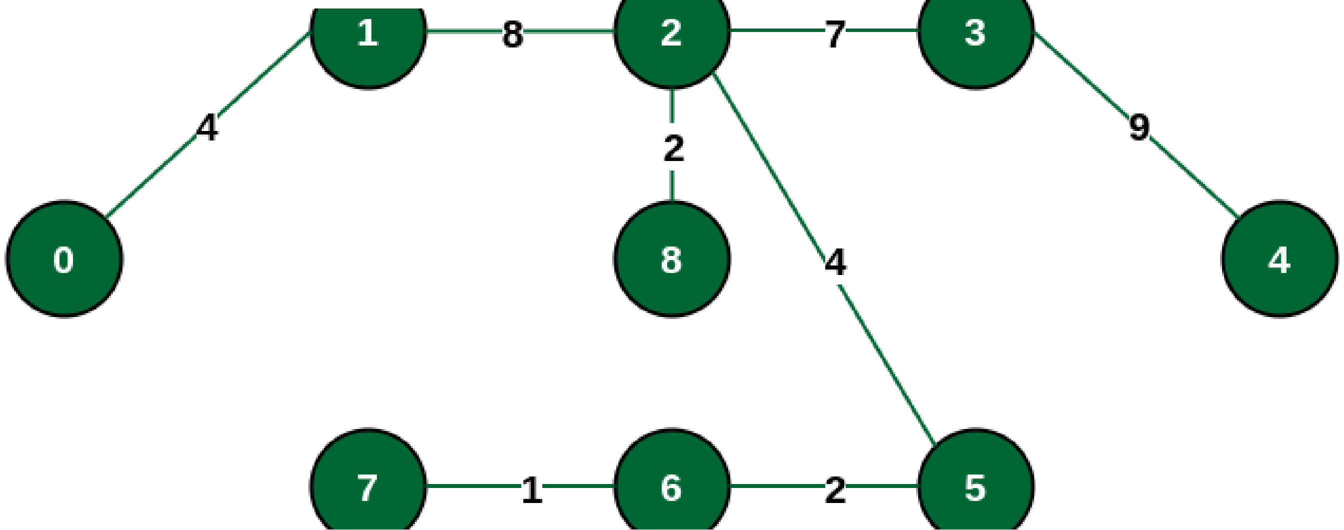
Their algorithm merges any two adjacent regions whose difference is smaller than the minimum internal difference of these two regions,

$$MInt(R_1, R_2) = \min(Int(R_1) + \tau(R_1), Int(R_2) + \tau(R_2)), \qquad (5.22)$$

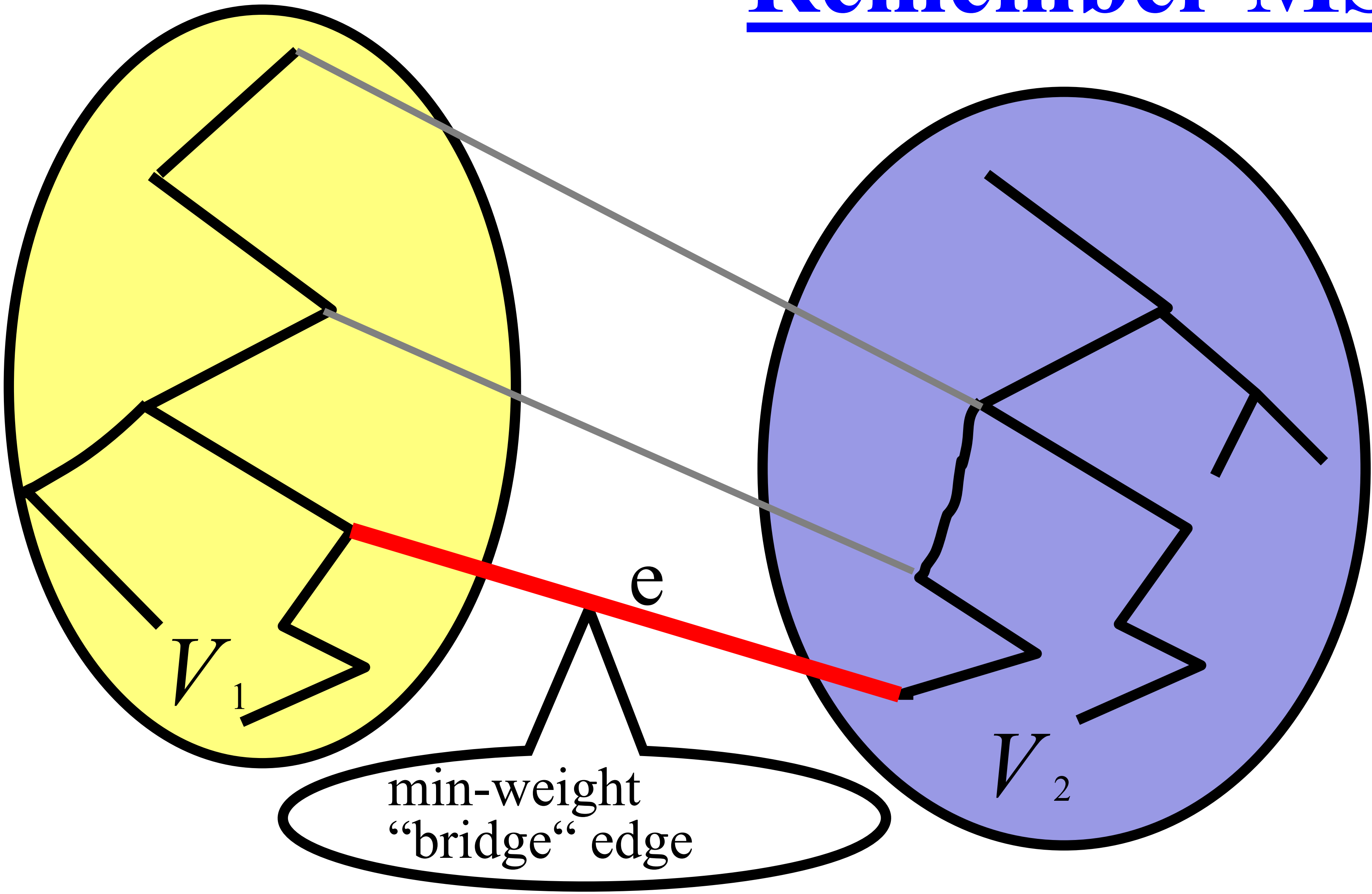
where $\tau(R)$ is a heuristic region penalty that Felzenszwalb and Huttenlocher (2004b) set to $k/|R|$, but which can be set to any application-specific measure of region goodness.


0
1
2
3
4
5
6
7
8
4
8
7
9
2
11
4
14
7
6
10
8
1
2


0
1
2
3
4
5
6
7
8
4
7
9
2
4
8
1
2

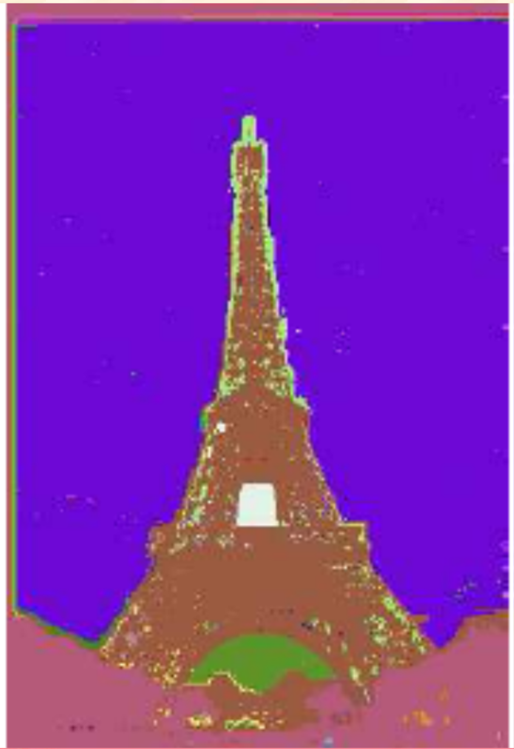

0
1
2
3
4
5
6
7
8
8
7
4
2
9
4
1
2

Remember MST



Merge regions in decreasing order of the edges separating them; From a lemma : edges causing merges are exactly the edges that would be selected by Kruskal's algorithm for constructing the minimum spanning tree (MST) of each component.





Segmentation and Graph Cut

- Similarity graphs: E-neighborhood, KNN, fully-connected
- 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'**

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*.

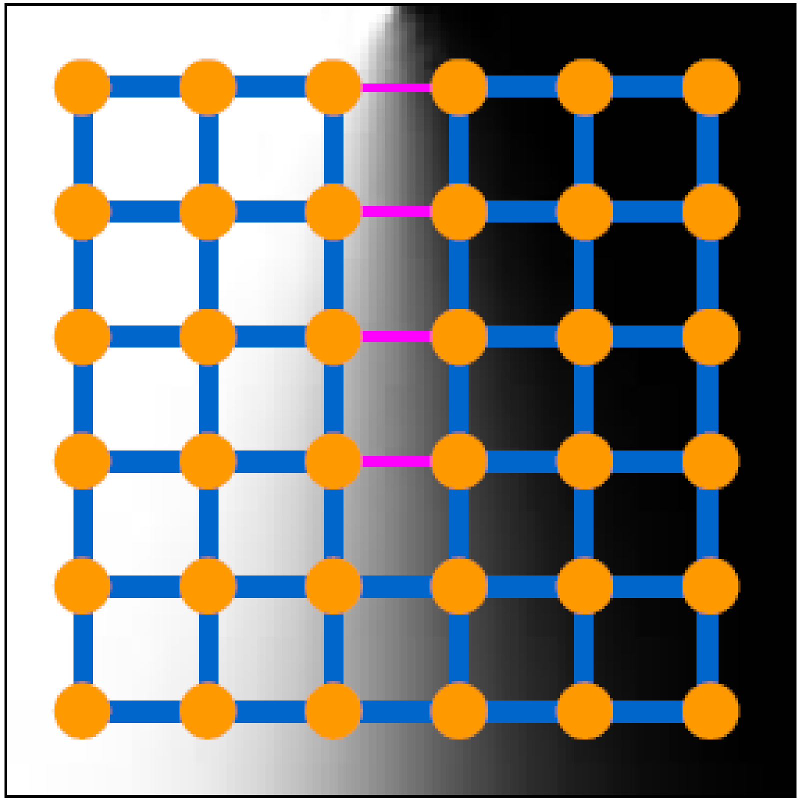
σ_I and σ_X are parameters set to 10-20 percent of the range of their related values.

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



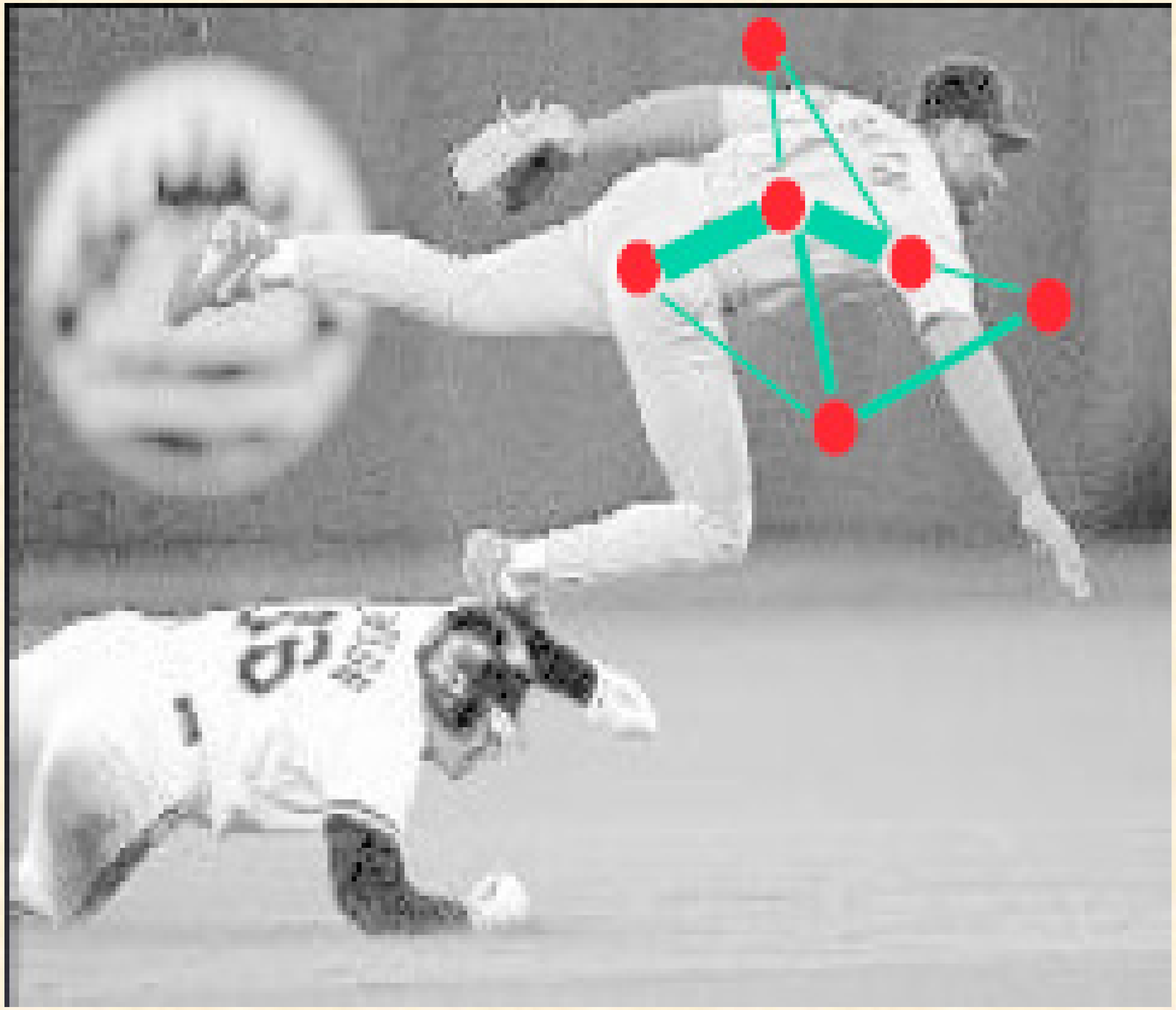
Low contrast – strong coupling

High contrast – weak coupling

V: graph nodes: ← → Image = { pixels }

E: edges connection nodes: ← → Pixel similarity

Representing Images as Similarity Graphs



Segmentation and Graph Cut

1) Given a source (**s**) and a sink node (**t**)

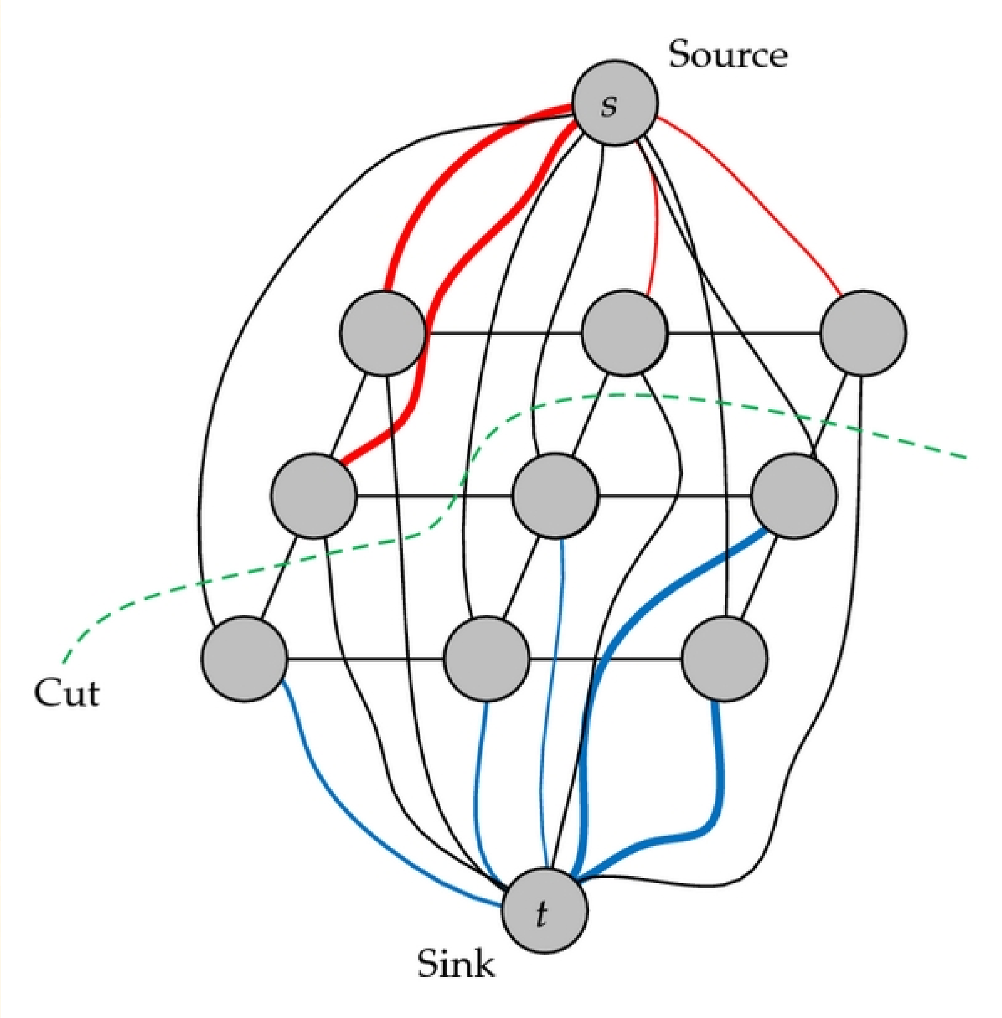
2) Define Capacity on each edge, $C_{ij} = W_{ij}$

3) Find the maximum flow from s->t, satisfying the capacity constraints

Min. Cut = Max. Flow

Max-flow/Min-cut theorem:

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



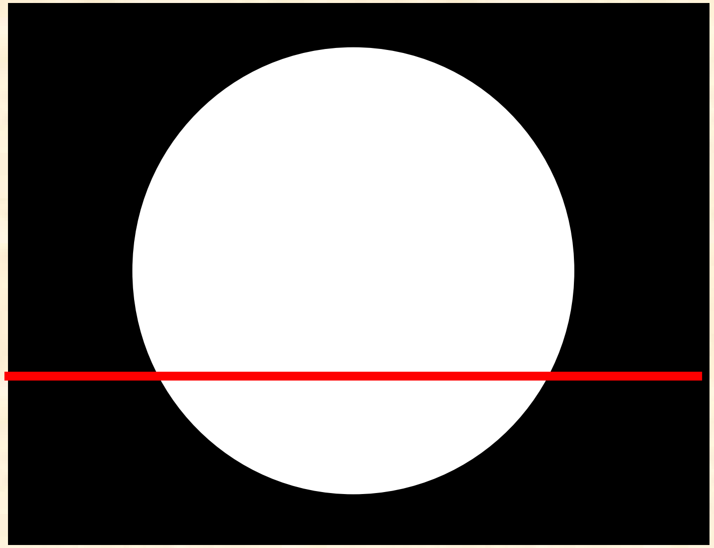
$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v).$$

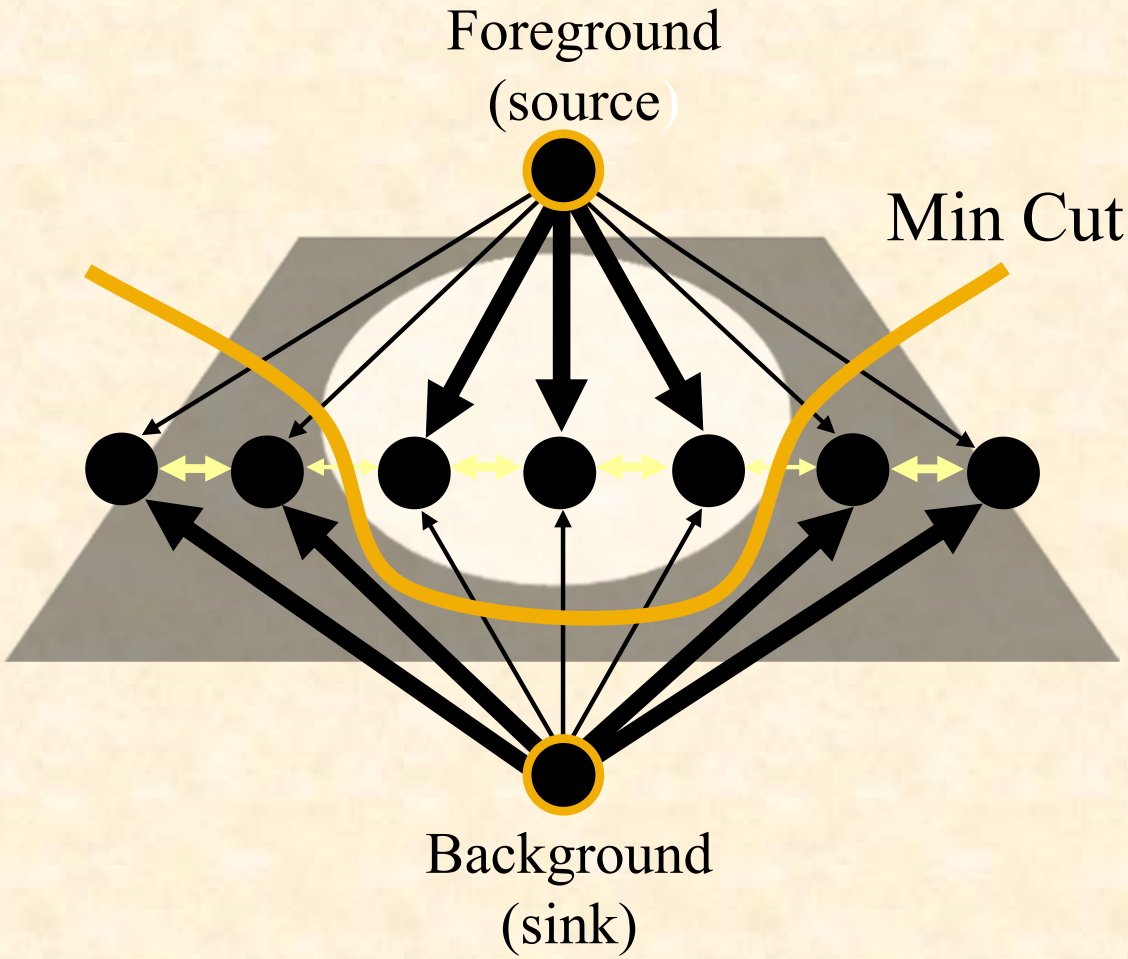
An example of min-cut/max-flow graph cut. The gray circles represent the nodes, and the solid lines are the edges between the nodes. The curve indicating each "flow" is connected to the source terminal or sink terminal. The potential of flow is measured by the width of line. The dotted line indicates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

Image source ***http://www.hindawi.com/journals/mpe/2012/814356/fig8/***

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**

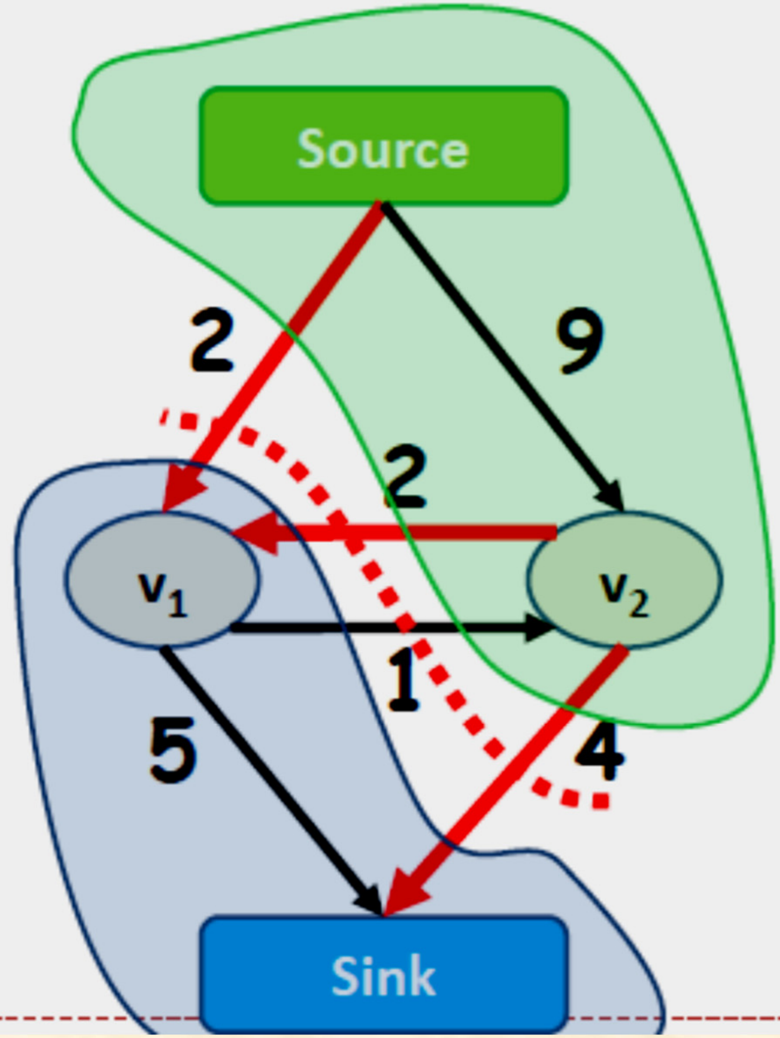
$$\text{cut}(A,B) = \sum_{u \in A,\, v \in B} w(u,v)$$

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

- Number of such partitions is exponential (2^N); but the minimum cut can be found efficiently.
- Ford and Fulkerson algo. is better than Linear Prog., to get the soln. efficiently. Edmonds-Karp uses idea of Ford, but uses breadth-first search to solve it with $O(V^2E)$. O(VE) algo. has also been suggested recently (2012) by J. Orlin + KRT.
-

Min-cut Problem

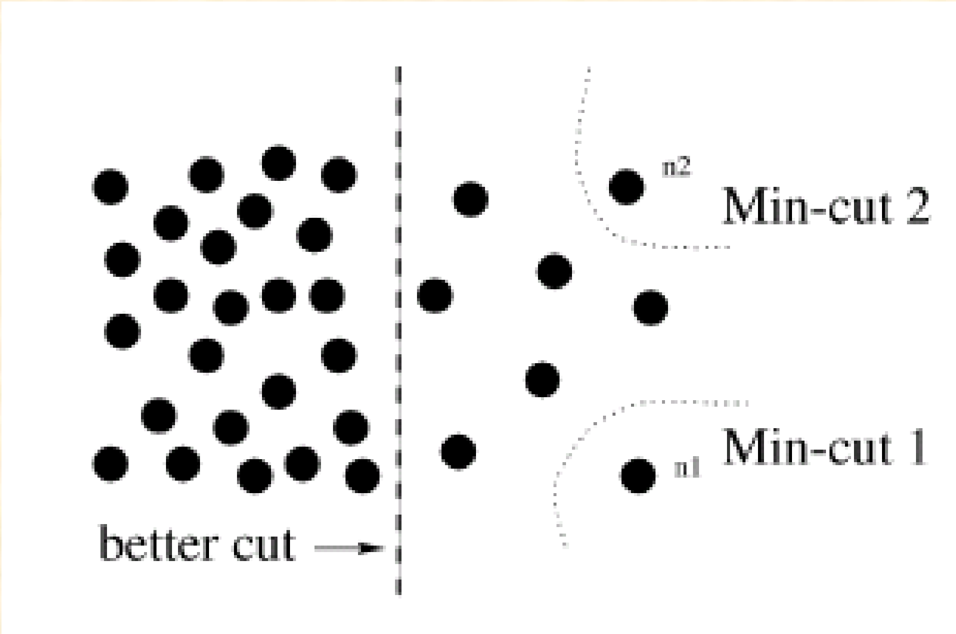
- Find the minimum s-t cut of a graph
  - An **s-t cut** *C* = (*S*,*T*) of a graph *G* = (*V*, *E*) is a cut of *G* such that $s \in S$ and $t \in T$, where *s* and *t* are the *source* and the *sink* of *N* respectively.
  - Cost of an s-t cut: $\sum_{(u,v)\in E, u\in S, v\in T} C_{u,v}$



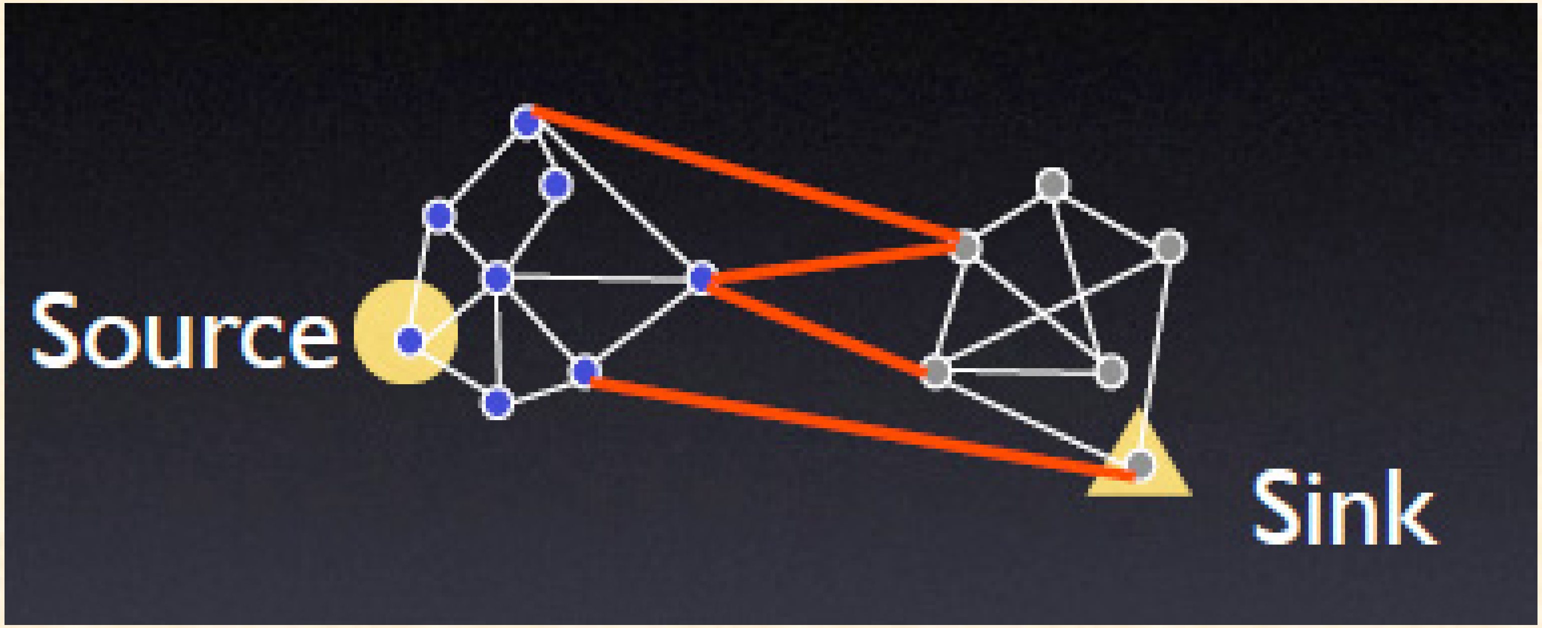
- Min-Cut = Max-Flow
- Polynomial-time solution
  - Augmenting Path: $O(|E|^2|V|)$
  - Push-Relabel: $O(|E||V|^2)$

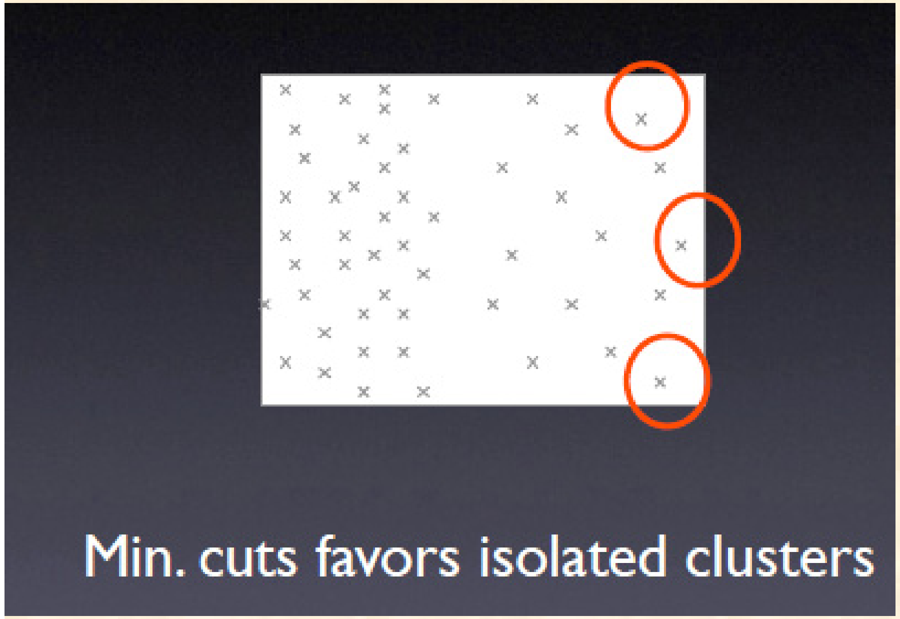
Problems with min-cut

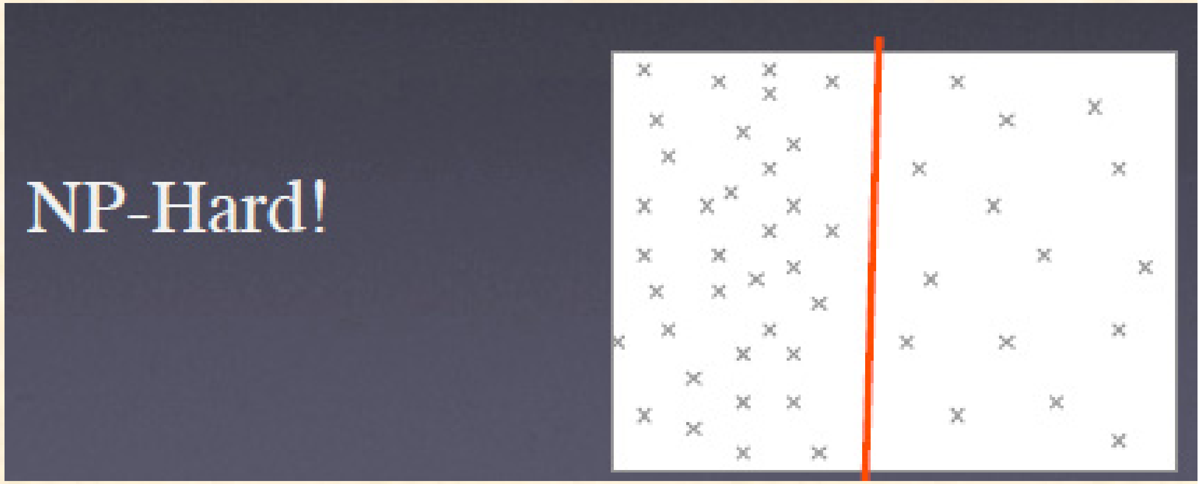
- Minimum cut criteria favors cutting small sets of isolated nodes in the graph.



A case where minimum cut gives bad partition


Source
Sink


Min. cuts favors isolated clusters


NP-Hard!

History of Graph partitioning:

- An Efficient Heuristic Procedure for Partitioning Graphs; B. W. Kernighan and S. Lin; Bell Syst Tech. J, vol. 49(2), 1970, pp 291-307. (***max-flow min-cut (Ford-Fulkerson-1962) is not suitable, as it has no size constraint***). /Princeton & Bell Labs.
- R. B. Boppana, "Eigenvalues and Graph Bisection: An Average-Case Analysis" in Proc. IEEE Symp. Found. Computer Sci., 1987, pp. 280-285. ***{Min. size (No. of edges cut) bisection algo.; This uses the largest eigenvalue of matrix (A+D); works well on average case}. /Rutgers- NJ***
- B Mohar, "The Laplacian spectrum of graphs," in Graph Theory, Combinatorics, and Applications, Y Alavi et a1. Eds , New York. Wiley, **1988/91**, pp 871-898. ***{Survey presenting the 2nd smallest eigenvalue of Laplacian, and many results}.*** */Yugoslavia*
- Lars Hagen and Andrew B Kahng, New Spectral Methods for Ratio cut Partitioning and Clustering; IEEE Transactions on CAD, vol. 11(9), Sept. **92**, pp 1074-1090. **{ 2nd eigenvector for cut - partitioning ckts. In VLSI}. /UCLA**

Pothen, H. D. Simon, and K.-P. Liou, Partitioning sparse matrices with eigenvectors of graphs, SIAM J. Matrix Anal. Appl., 11 (**1990**), pp. 430-452.

George Karypis AND Vipin Kumar; A FAST AND HIGH QUALITY MULTILEVEL SCHEME FOR PARTITIONING IRREGULAR GRAPHS; SIAM J. SCI. COMPUT., 1998, Vol. 20, No. 1, pp. 359-392. //CSE-Univ, of Minnesota

Ulrike von Luxburg; A tutorial on spectral clustering; Stat Comput (2007) 17: 395–416; DOI 10.1007/s11222-007-9033-z //Tubingen, Germany.

Path-Finding Methods for Linear Programming - Solving Linear Programs in ˜O(√rank) Iterations and Faster Algorithms for Maximum Flow; **Yin Tat Lee**,, **Aaron Sidford**; Department of Mathematics/EECS, **MIT**, Cambridge, USA **2014** IEEE Annual Symposium on Foundations of Computer Science.

V. Osipov, P. Sanders, and C. Schulz. Engineering Graph Partitioning Algorithms. In Proceedings of the 11th International Symposium on Experimental Algorithms (SEA'12), volume 7276, pages 18–26. Springer, 2012.

Year	Author	Running Time
1972	Edmonds, Karp [6]	$\tilde{O}(\|E\|^2 \log(U))$
1984	Tardos [40]	$O(\|E\|^4)$
1984	Orlin [31]	$\tilde{O}(\|E\|^3)$
1986	Galil, Tardos [10]	$\tilde{O}(\|E\|\|V\|^2)$
1987	Goldberg, Tarjan [12]	$\tilde{O}(\|E\|\|V\| \log(U))$
1988	Orlin [32]	$\tilde{O}(\|E\|^2)$
2008	Daitch, Spielman [5]	$\tilde{O}(\|E\|^{3/2} \log^2(U))$
2013	This paper	$\tilde{O}(\|E\|\sqrt{\|V\|} \log^2(U))$

Figure 2. Minimum Cost Flow Running Times Improvements

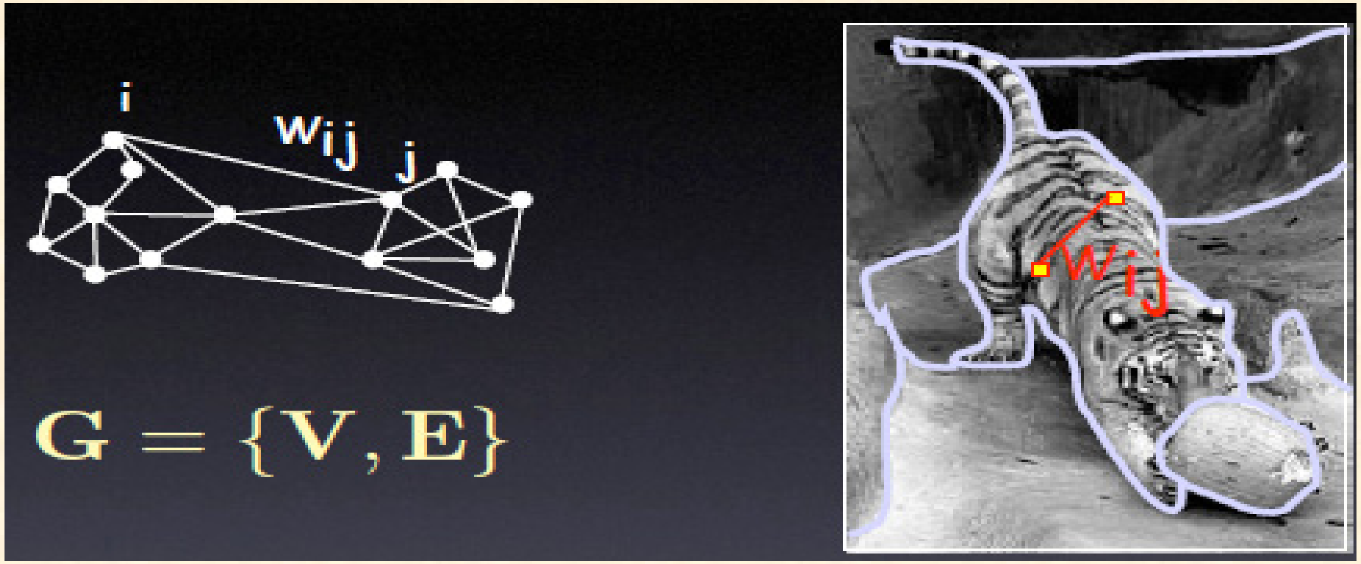
Solution – Normalized Cut

- 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

- Also see Spectral Clustrering, from ML literature.

Illustrations follow, Tutorial of:
Graph Based Image Segmentation; CVPR-2004
Jianbo Shi, David Martin, Charless Fowlkes, Eitan Sharon

NORMALIZED GRAPH CUT



V: graph nodes: ←→ Image = { pixels }

E: edges connection nodes: ← → Pixel similarity

A graph G = { V, E } can be partitioned into two disjoint sets: A, B; A U B = V , $A \bigcap B = \Phi$, 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 V1,V2, . . . ,Vm, where by some measure the similarity among the vertices in a set Vi is high and, across different sets Vi, Vj is low.

Mincut creates a optimal bi-partioning 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

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

where, assoc(A, V) is the total connection from nodes in A to all nodes in the graph.

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

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

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

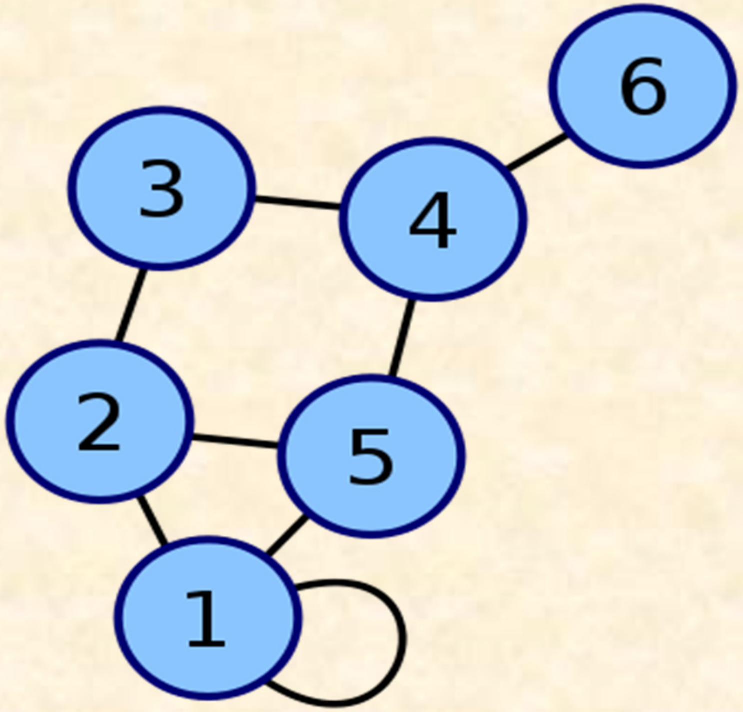
be the total connection from node i to all other nodes.

Let D be an NxN diagonal matrix with d on its diagonal;

W be an NxN symmetrical **matrix with W(i, j) = w(i, j);**

W is also an adjacency matrix.

The adjacency matrix of a finite graph G on n vertices is the n × n matrix where the non-diagonal entry a_{ij} is the number of edges from vertex i to vertex j, and the diagonal entry aii, depending on the convention, is either once (directed) or twice (undirected) the number of edges (loops) from vertex i to itself. In the special case of a finite simple graph, the adjacency matrix is a (0,1)-matrix with zeros on its diagonal. If the graph is undirected, the adjacency matrix is symmetric.



Labeled graph

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Adjacency matrix

Computation for normalized graph- cut

- x is an N =|V| dimensional indicator vector, $x_i = 1$ if node i is in A and -1, otherwise

- $d(i) = \Sigma_j w(i,j)$ be the total connection from node i to all other nodes.
- D is a N x N diagonal matrix with d on its diagonal
- W is a N x N symmetrical matrix with $W(i,j) = w_{ij}$
- k is defined as $k = \frac{\sum_{xi>0} d_i}{\sum_i d_i}$

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 1 be a Nx1 vector of all ones.**

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

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

Solving and setting,
b = k/(1-k) :

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

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

Using, y = (1 + x) - b(1 – x)

we have :

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

under the condition

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

The above expression is the **Rayleigh quotient**. If y is relaxed to take on real values, the above eq^{n}. can be minimized 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.

L = (D-W) is called the Laplacian matrix (symmetric and +ve Semi-Defnt.). Rayleigh quotient can be reduced to:

$$z = \lambda z \quad \Rightarrow \quad Az = \lambda z;$$

where, $z = D^{\frac{1}{2}} y$; A is sparse, as 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 = λDx 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

Rayliegh 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 x1, . . . , x_{j-1}, the quotient $\mathbf{x^T A x / x^T x}$ 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

The Rayleigh quotient reaches its minimum value $\lambda_{\min}$ (the smallest eigenvalue of *M*) when *x* is $v_{\min}$ (the corresponding eigenvector).

Let A be Hermitian. Then the Rayleigh quotient satisfies

$$\lambda_1 = \min \rho(\mathbf{x}), \qquad \lambda_n = \max \rho(\mathbf{x}).$$

$$R(M, x) := \frac{x^* M x}{x^* x}.$$

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:

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

$$R(H; x, y) := \frac{y^* H x}{\sqrt{y^* y \cdot x^* x}}$$

Altn. Formulation:

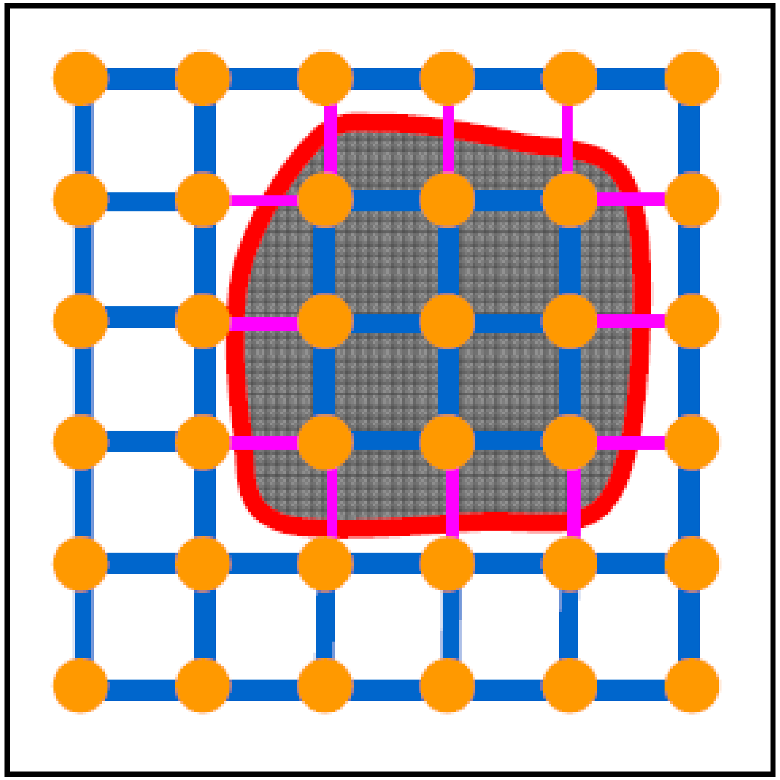
Normalized-Cut Measure

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

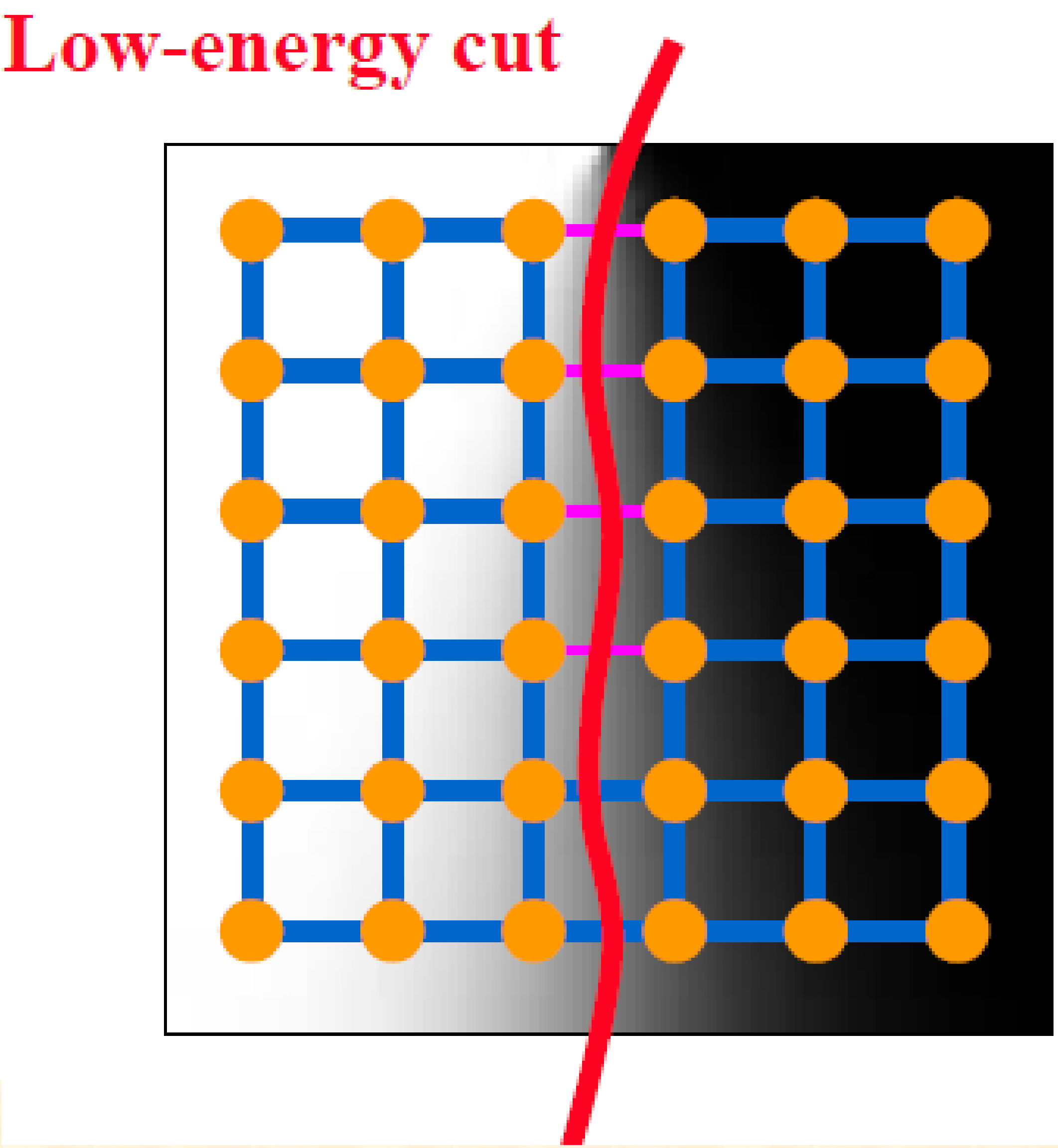
$$N(S) = \sum w_{ij} u_i u_j$$

Minimize:

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



Normalized-Cut Measure



Minimize:

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

Matrix Formulation

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

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

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

Read about Spectral-cut methods

Properties of L:

(1) *For every vector* $f \in \mathbb{R}^n$ *we have*

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

(2) *L is symmetric and positive semi-definite.*

(3) *The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector* $\mathbb{1}$.

(4) *L has n non-negative, real-valued eigenvalues* $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

Normalized Laplacian:

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2};$$

$$L_{rw} = D^{-1} L = I - D^{-1} W \ \text{(random walk)}$$

RANDOM WALKS on GRAPHS

- $G = (V, E)$: a simple connected graph on n vertices
- $A(G)$: the adjacency matrix
- $D(G) = \mathrm{diag}(d_1, d_2, \ldots, d_n)$: the diagonal degree matrix
- $L = D - A$: the combinatorial Laplacian
- L is semi-definite and $\mathbf{1}$ is always an eigenvector for the eigenvalue 0.

Normalized Laplacian: $\mathcal{L} = I - D^{-1/2} A D^{-1/2}$.

- $\mathcal{L}$ is always semi-definite.
- 0 is always an eigenvalue of $\mathcal{L}$ with eigenvector $(\sqrt{d_1}, \ldots, \sqrt{d_n})'$.

- Laplacian eigenvalues: $\lambda_0, \ldots, \lambda_{n-1}$

$$0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{n-1} \le 2.$$

- $\lambda_{n-1} = 2$ if and only if G is bipartite.
- $\lambda_1 > 1$ if and only if G is the complete graph.

A walk on a graph is a sequence of vertices together a sequence of edges:

$$v_0, v_1, v_2, v_3, \ldots, v_k, v_{k+1}, \ldots$$

$$v_0v_1, v_1v_2, v_2v_3, \ldots, v_kv_{k+1}, \ldots$$

Random walks on a graph G:

$$f_{k+1} = f_k D^{-1} A.$$

$$D^{-1}A \sim D^{-1/2}AD^{-1/2} = I - \mathcal{L}.$$

$\bar{\lambda}$ determines the mixing rate of random walks.

Normalized spectral clustering according to Shi and Malik (2000)

Input: Similarity matrix $S \in \mathbb{R}^{n\times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Sect. 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k generalized eigenvectors $u_1,\dots,u_k$ of the generalized eigenproblem $Lu = \lambda Du$.**
- Let $U \in \mathbb{R}^{n\times k}$ be the matrix containing the vectors $u_1,\dots,u_k$ as columns.
- For $i = 1,\dots,n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1,\dots,C_k$.

Output: Clusters $A_1,\dots,A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Normalized spectral clustering according to Ng et al. (2002)

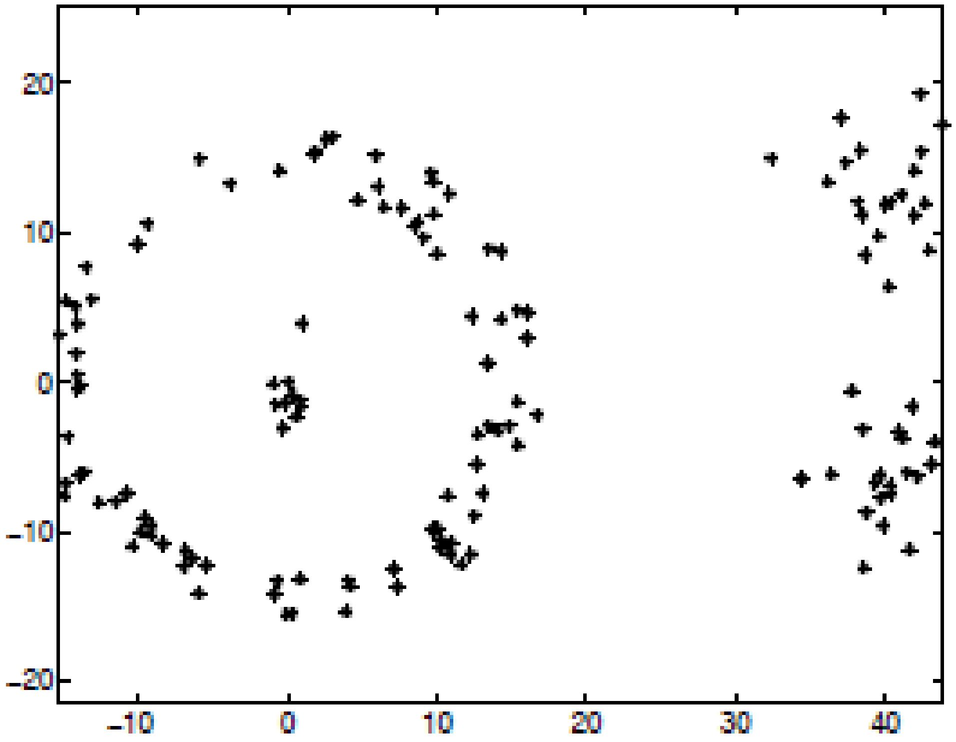
Input: Similarity matrix $S \in \mathbb{R}^{n\times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Sect. 2. Let W be its weighted adjacency matrix.
- Compute the normalized Laplacian L_{sym}.
- **Compute the first k eigenvectors $u_1,\dots,u_k$ of L_{sym}.**
- Let $U \in \mathbb{R}^{n\times k}$ be the matrix containing the vectors $u_1,\dots,u_k$ as columns.
- **Form the matrix $T \in \mathbb{R}^{n\times k}$ from U by normalizing the rows to norm 1,** that is set $t_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{1/2}$.
- For $i = 1,\dots,n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of T.
- Cluster the points $(y_i)_{i=1,\dots,n}$ with the k-means algorithm into clusters $C_1,\dots,C_k$.

Output: Clusters $A_1,\dots,A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

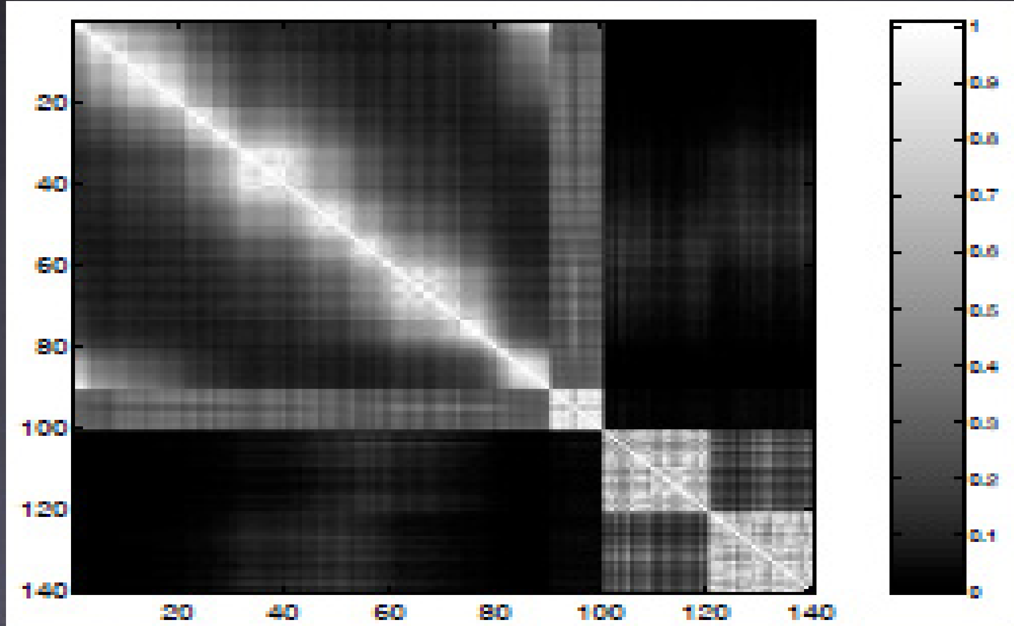
A tutorial on spectral clustering; Ulrike von Luxburg; Stat Comput (2007) 17: pp 395–416.

A Graphical Illustration of GRAPHCUT



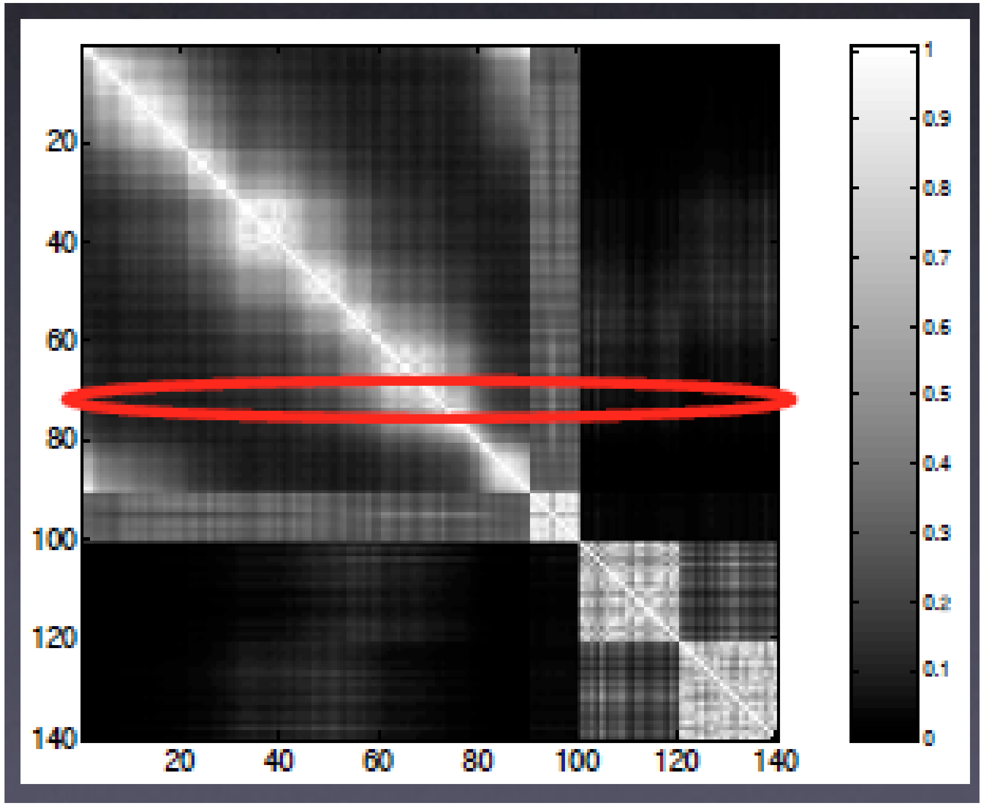
90 points in circular ring;
10 points in inside cluster;
20 points each in the right-hand Clusters.

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



Jianbo Shi, David Martin, Charless Fowlkes, Eitan Sharon

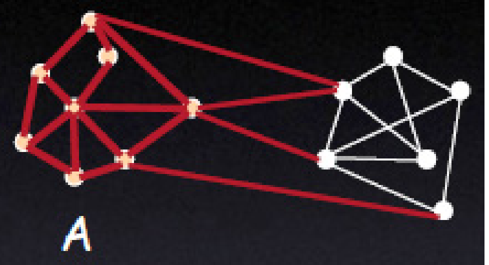
$$d_i = \sum_j S_{ij}$$

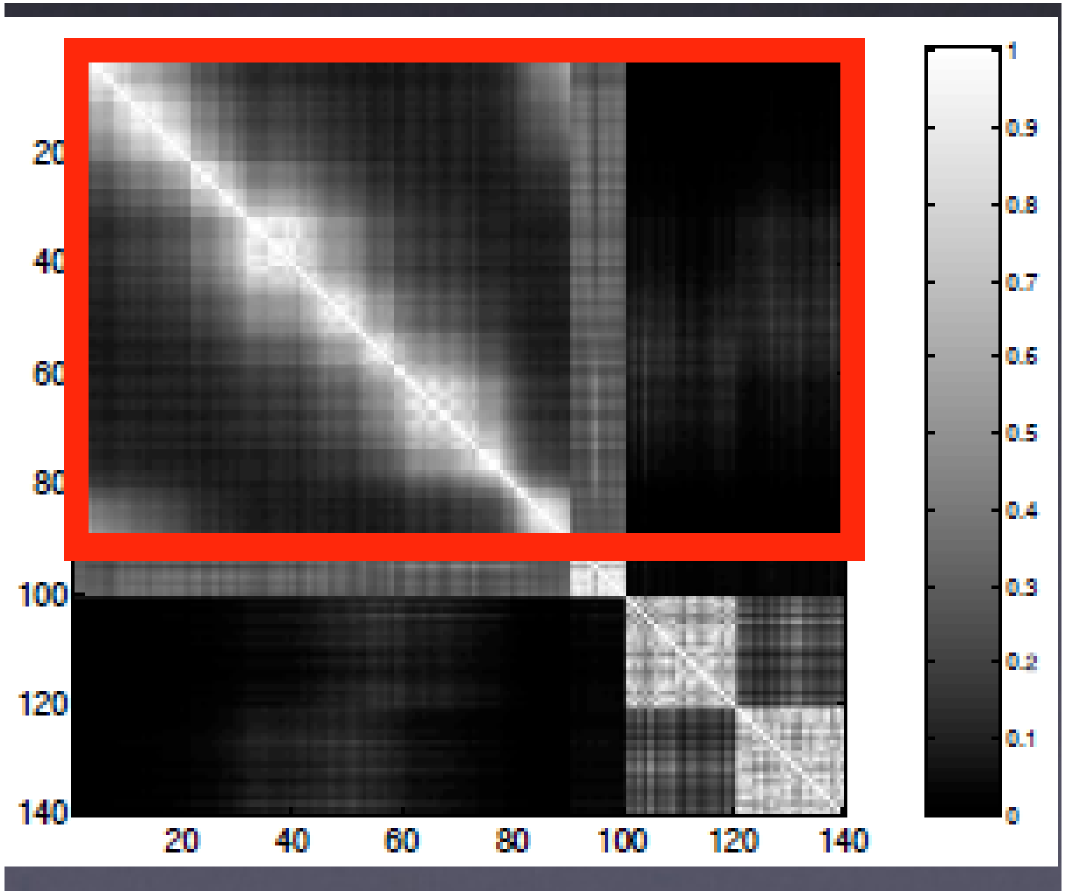


Tutorial
Graph Based Image Segmentation; CVPR-2004
Jianbo Shi, David Martin, Charless Fowlkes, Eitan Sharon

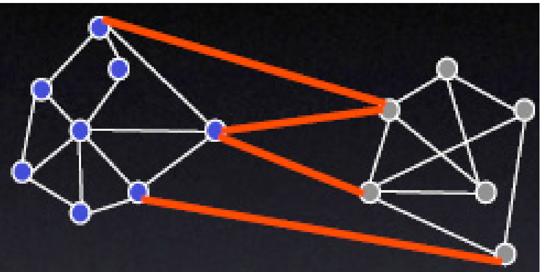
Volume of set:

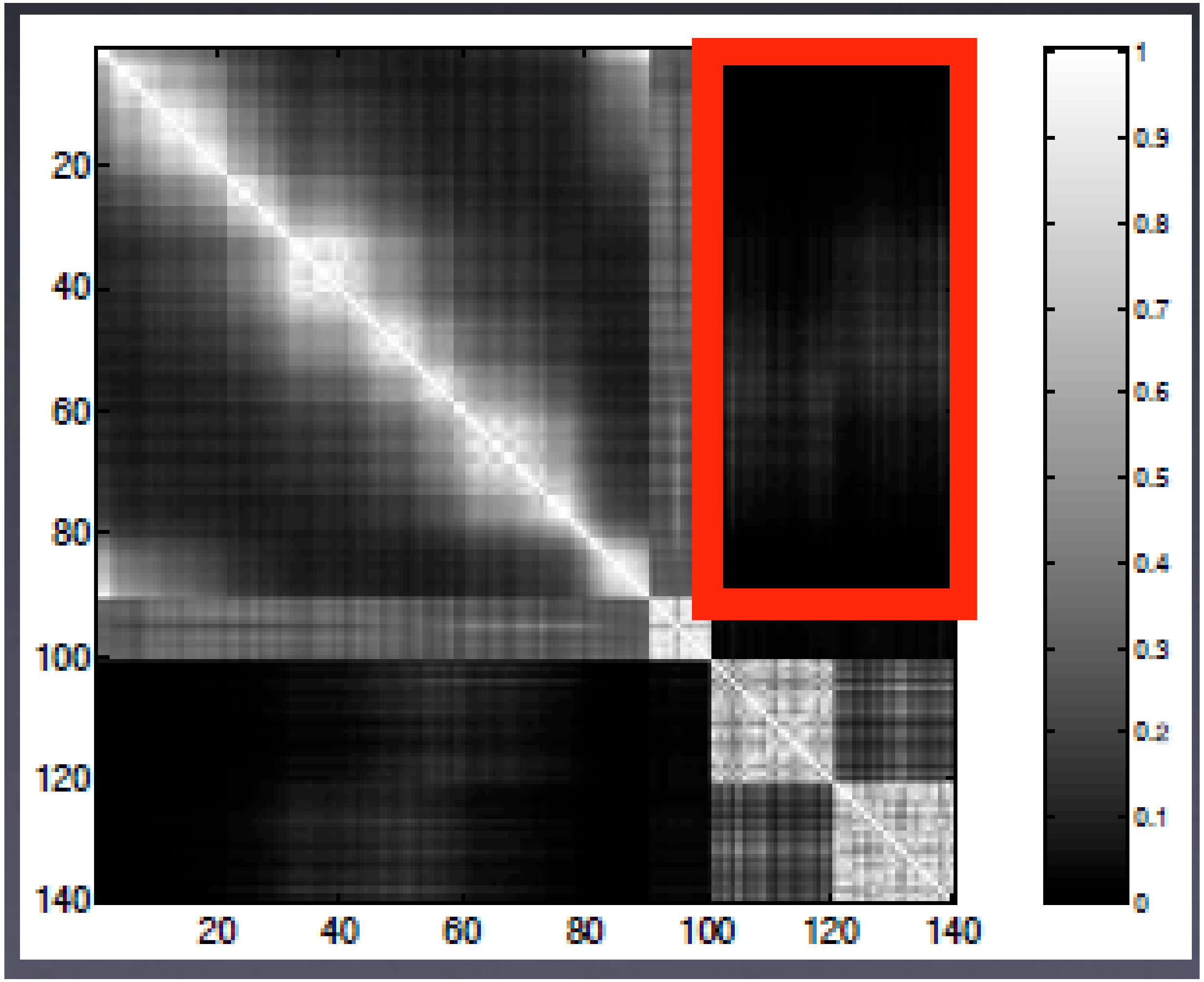
$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$

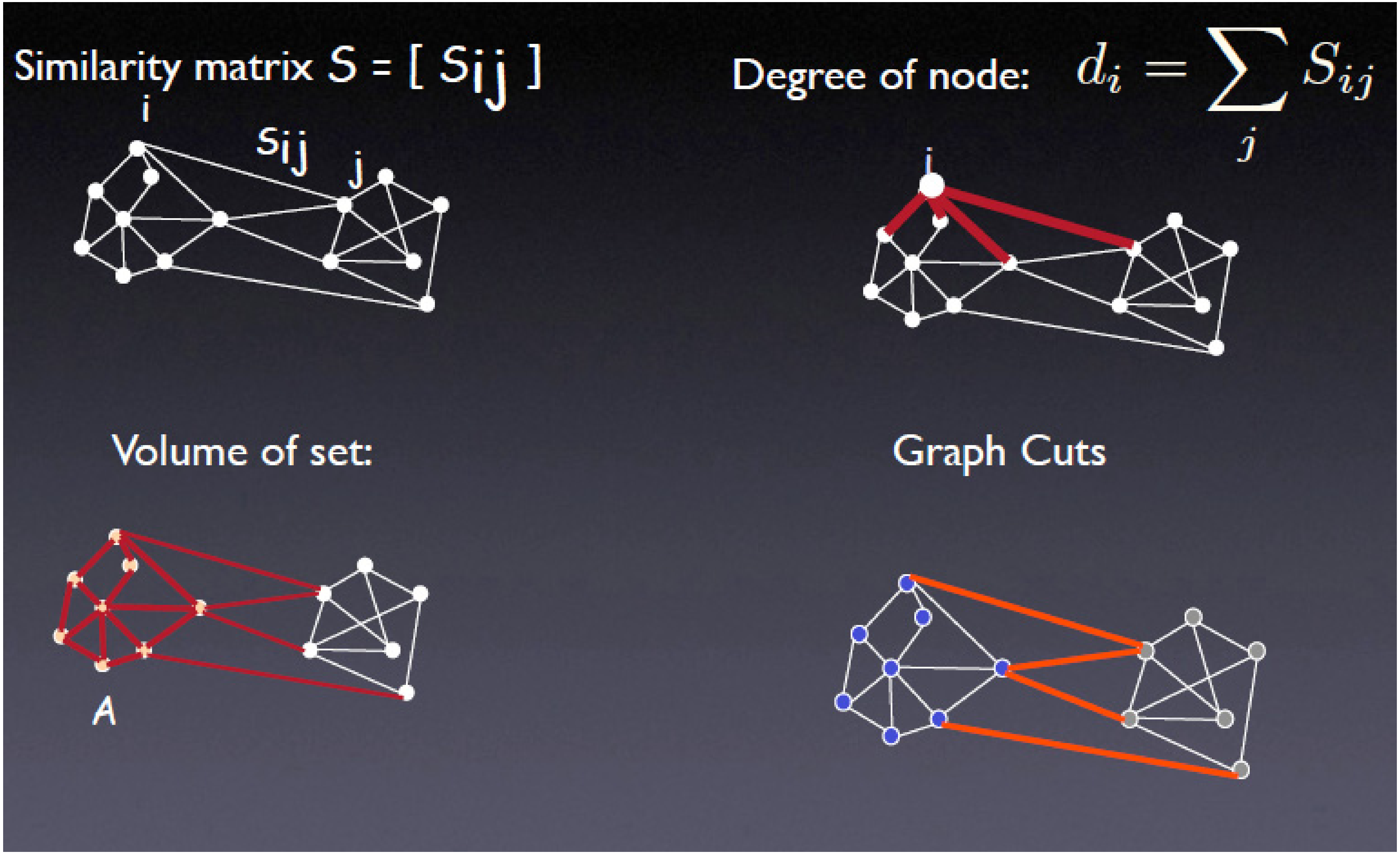




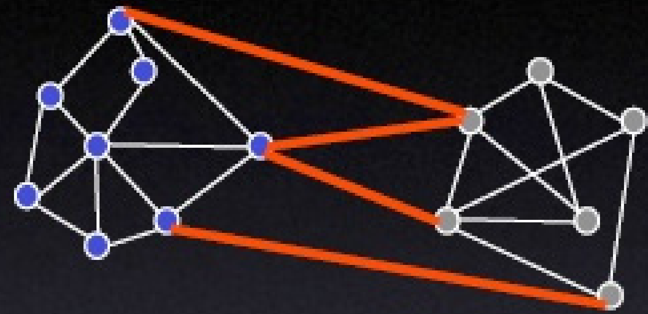
$$cut(A,\bar{A}) = \sum_{i \in A, j \in \bar{A}} S_{i,j}$$



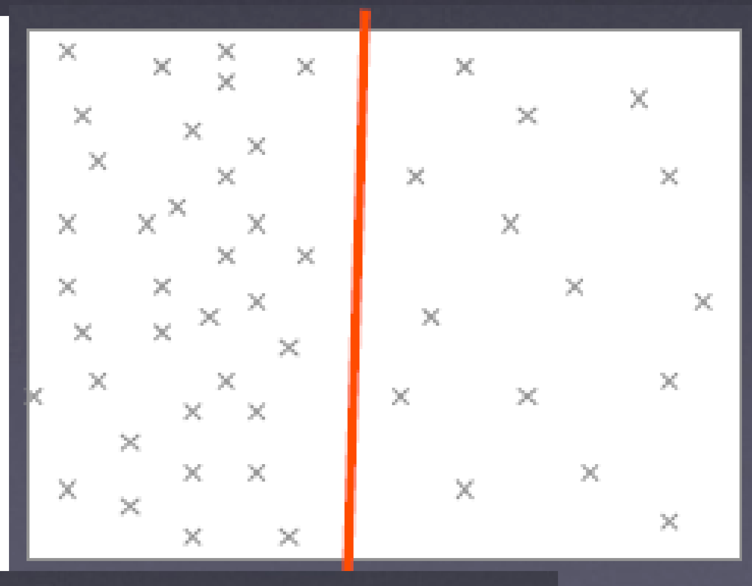



Similarity matrix S = [Sij]
i
Sij
j
Degree of node: $d_i = \sum_j S_{ij}$
i
Volume of set:
A
Graph Cuts

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

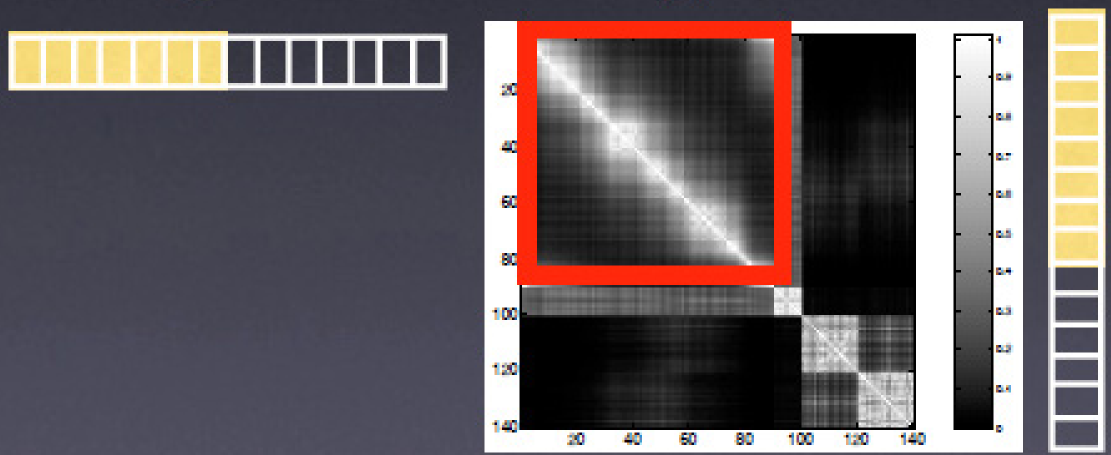
Laplacian matrix D-W

Let x = X(1,:) be the indicator of group 1

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

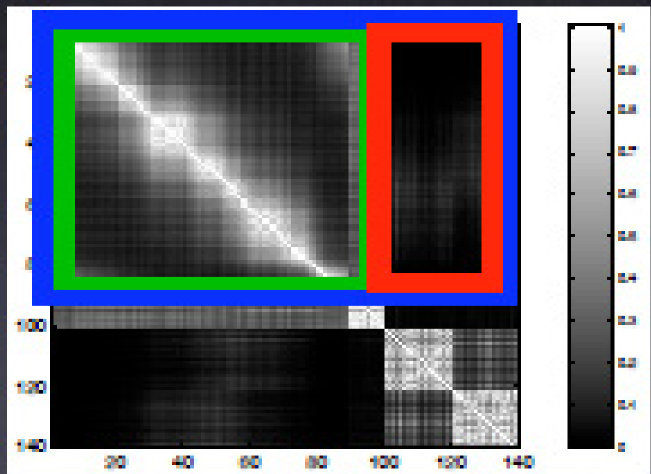
asso(A,A) = $x^T W x$

x^T W x



Laplacian matrix D-W

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



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

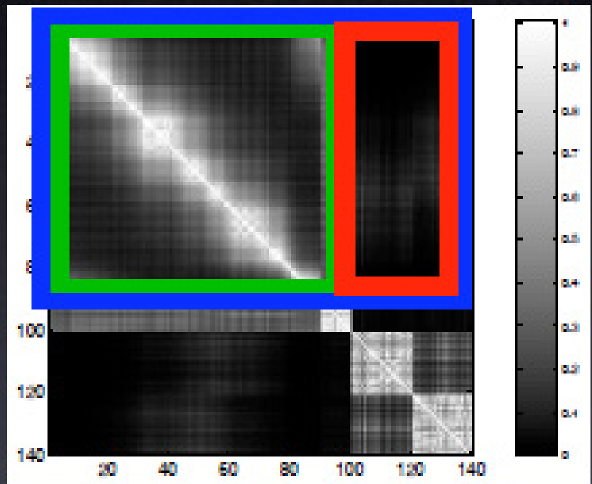
$$Ncut(A,B) = cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$$

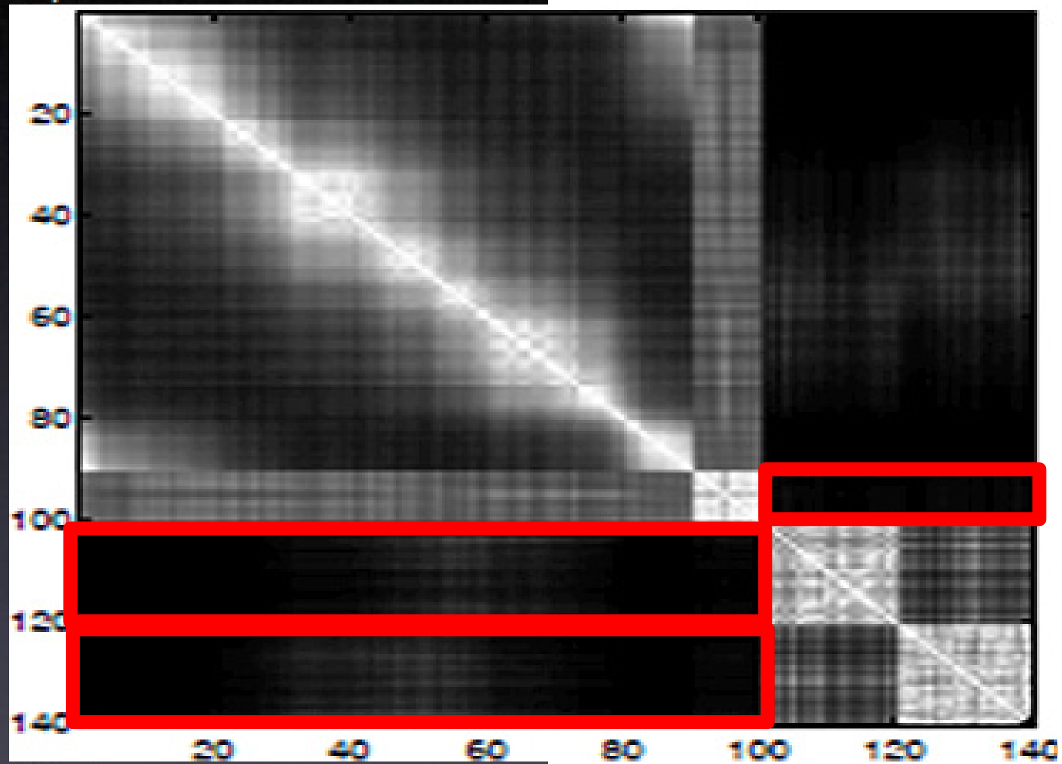
	$x^T D x$		$x^T W x$
Cut(A,V-A) =	vol(A)	-	asso(A,A)

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

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

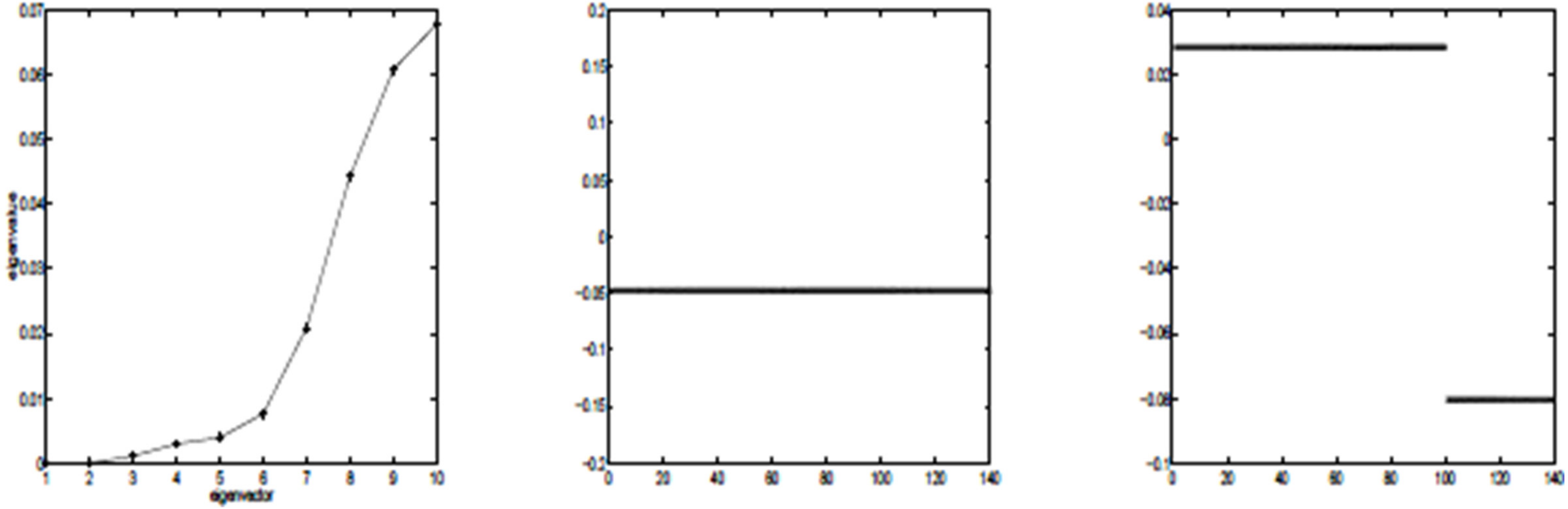
$$Ncut(X) = \frac{1}{K}\sum_{l=1}^{K}\frac{cut(V_l, V - V_l)}{vol(V_l)}$$





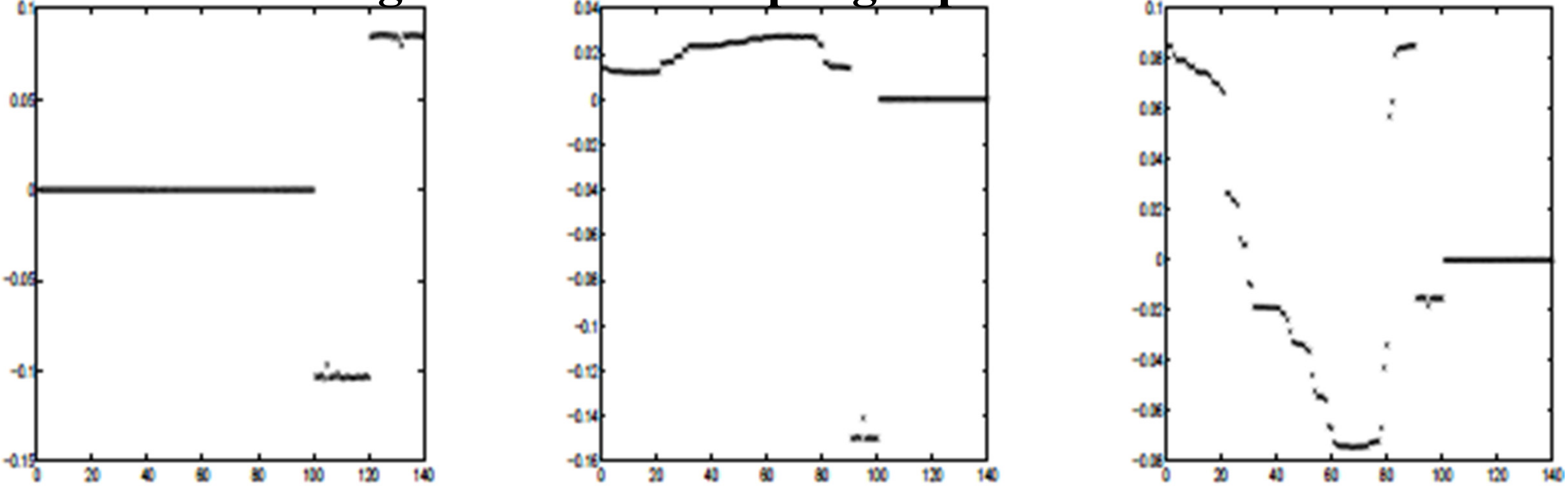
$$= \frac{1}{K}\sum_{l=1}^{K}\frac{X_l^T(D-W)X_l}{X_l^T D X_l}$$

$$X \in \{0,1\}^{N\times K}, X1_K = 1_N$$



First plot shows the 10 smallest eigenvalues; and subplots show the eigenvectors of the 5 smallest eigenvalues.

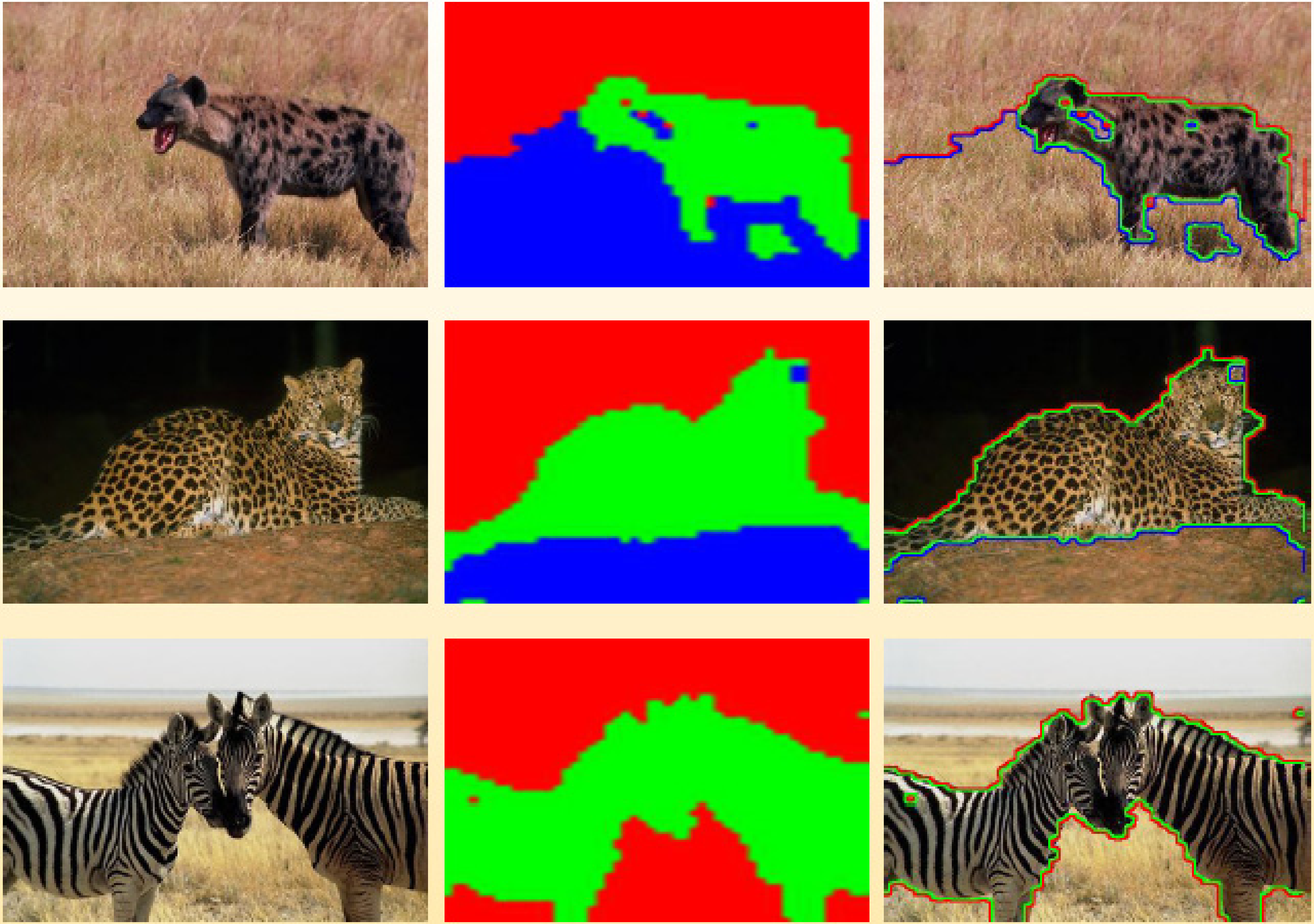
Desired Eigenvector is on top-right plot – the indicator vector.



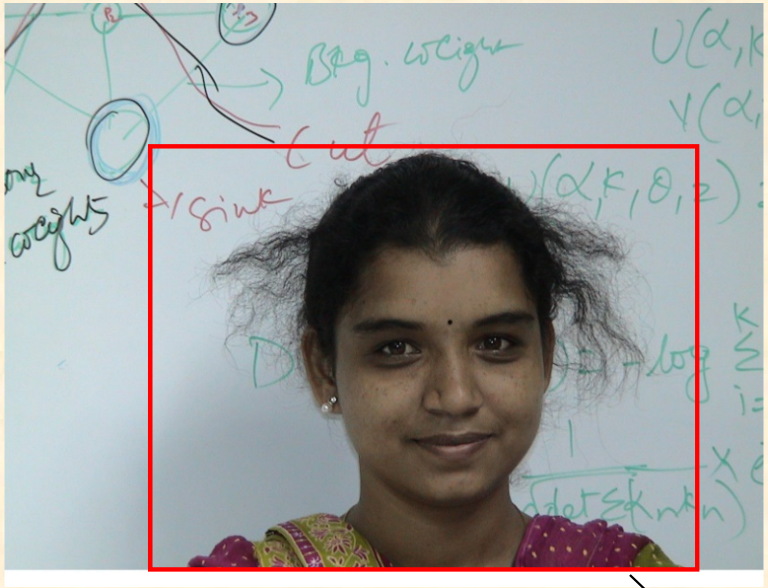
Example Normalized Cut

(a) (b)

Shi & Malik, 2000



Object Extraction From an Image



Snake

N-Cut

Figure 1. (a) An image with sparse constraints: white scribbles indicate foreground, black scribbles indicate background. Applying Bayesian matting to such sparse input produces a completely erroneous matte (b). Foreground extraction algorithms, such as [9, 11] produce a hard segmentation (c). An automatically generated trimap from a hard segmentation may miss fine features (d). An accurate hand-drawn trimap (e) is required in this case to produce a reasonable matte (f). (Images taken from [15])

[15] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2005.

A Closed Form Solution to Natural Image Matting

Anat Levin, Dani Lischinski, Yair Weiss; CVPR-2006; PAMI(2008)-30(2).

[11] C. Rother, V. Kolmogorov, and A. Blake. ”grabcut”: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[9] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i$$

Iterated Graph Cut

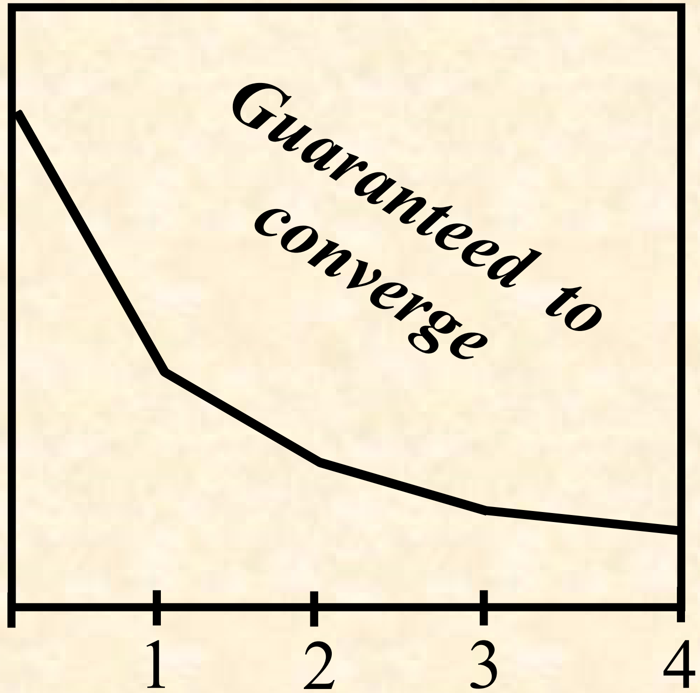
User Initialisation

K-means for learning colour distributions

Graph cuts to infer the segmentation

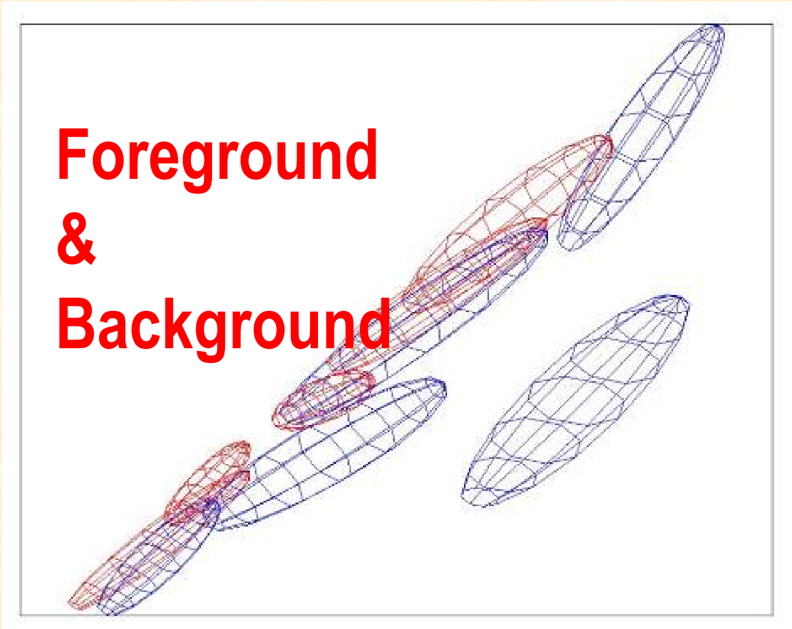
Iterated Graph Cuts

Result



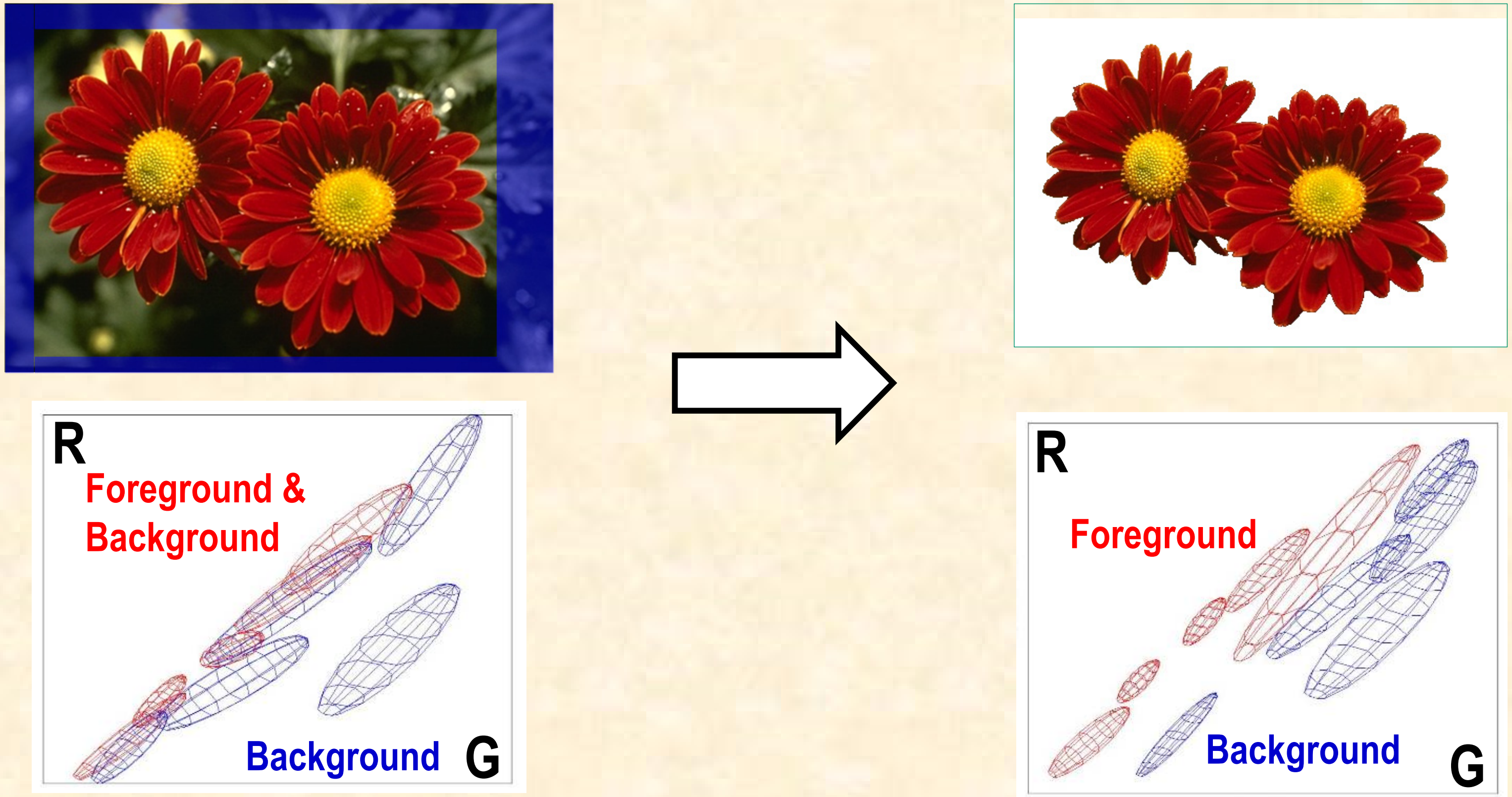
Energy after each Iteration

Colour Model



Gaussian Mixture Model (typically 5-8 components)

Colour Model



Gaussian Mixture Model (typically 5-8 components)

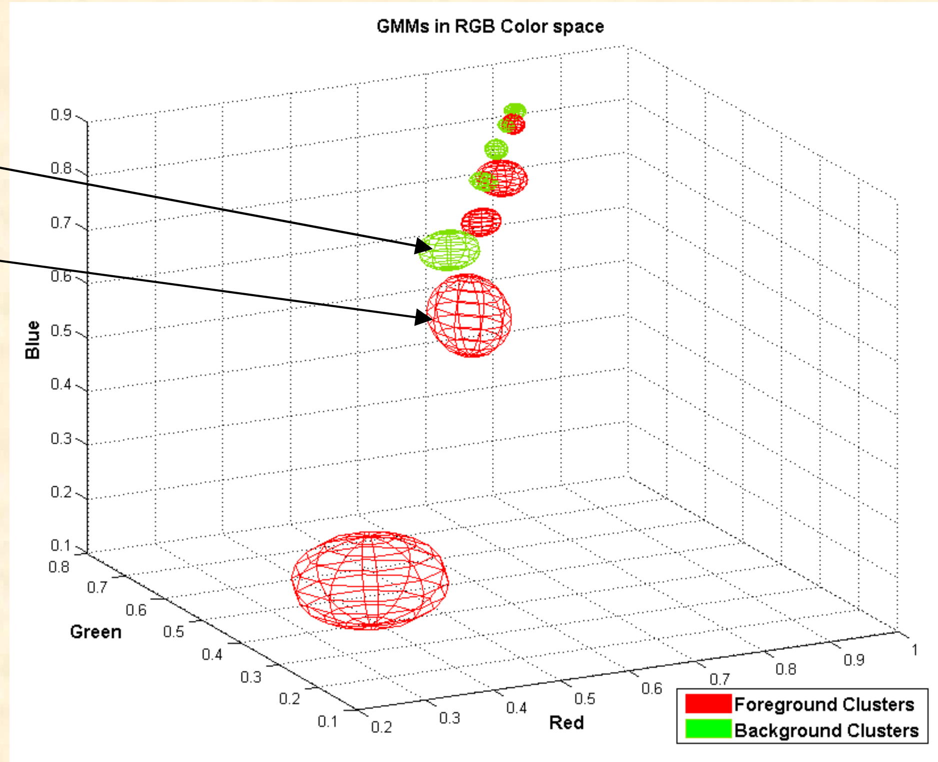
Initially both GMMs overlap considerably, but are better separated after convergence, as the foreground/background labelling has become accurate.

Object Extraction From an Image

Alpha-Matte based Foreground Extraction:

Unknown foreground

Known Background

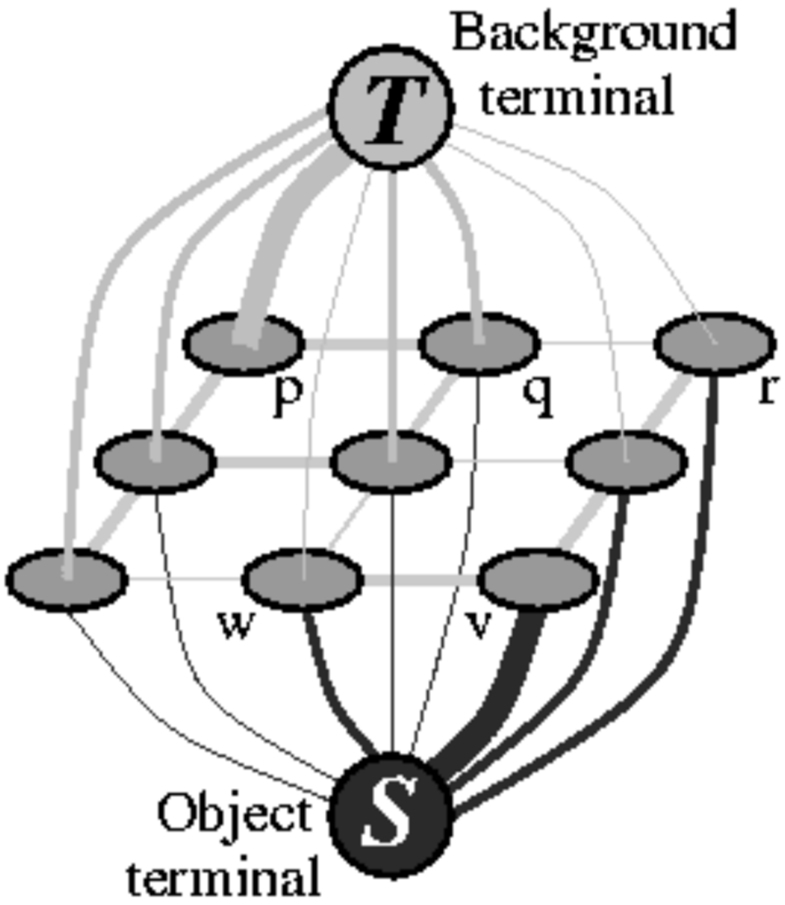


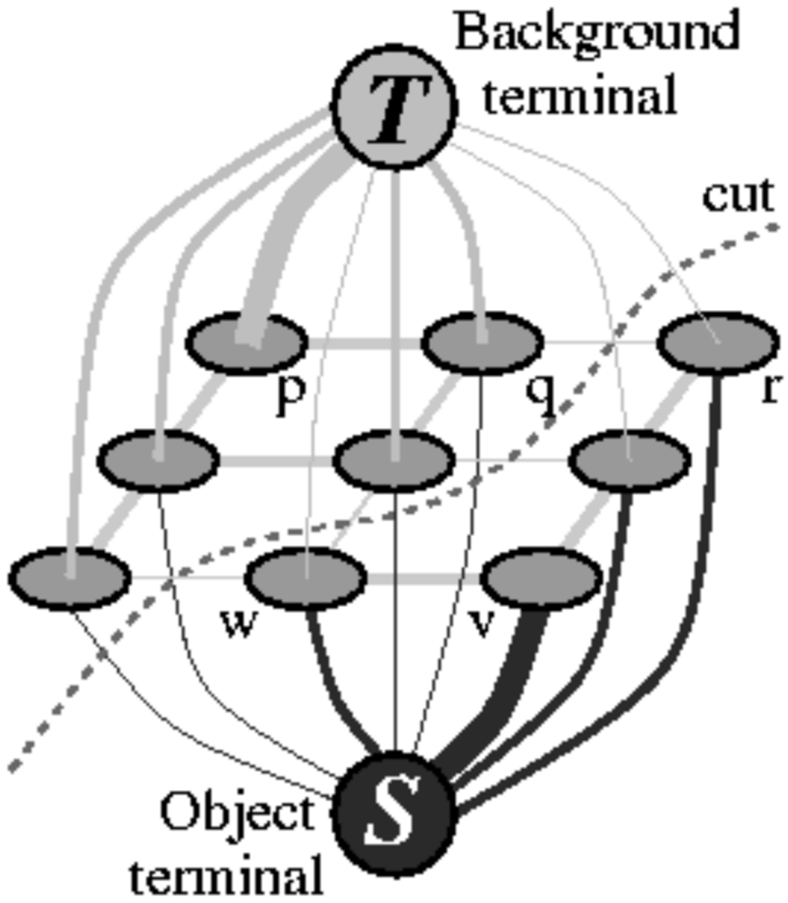
Create GMMs with K components for foreground and background separately

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

Image with seeds.

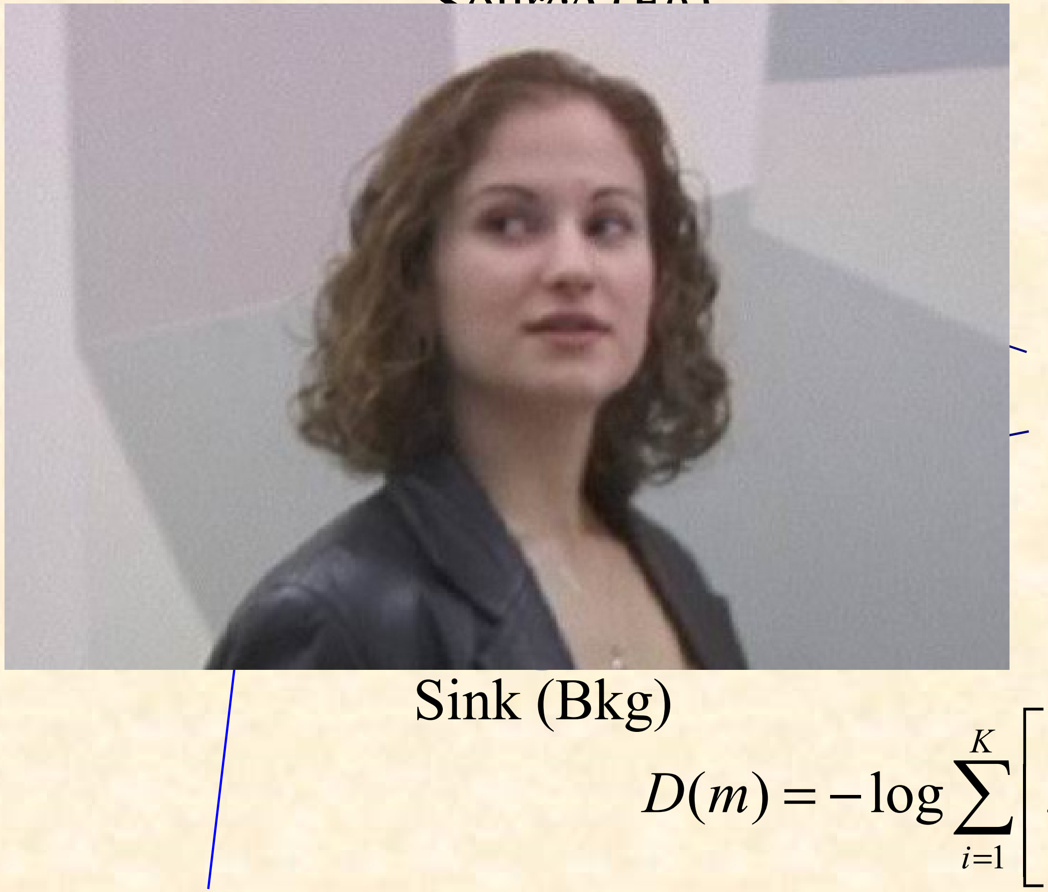
Segmentation results





Object Extraction From an Image

Source (Fg)



Sink (Bkg)

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

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

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

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

GrabCut segmentation

$$E(L) = \sum_{p} D_p(f_p) + \sum_{p,q \in N} V(f_p, f_q)$$

1. Define graph
   – usually 4-connected or 8-connected
2. Define unary potentials (data/region term; t-links)
   – 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 (smoothness / boundary term; interaction/n-links)

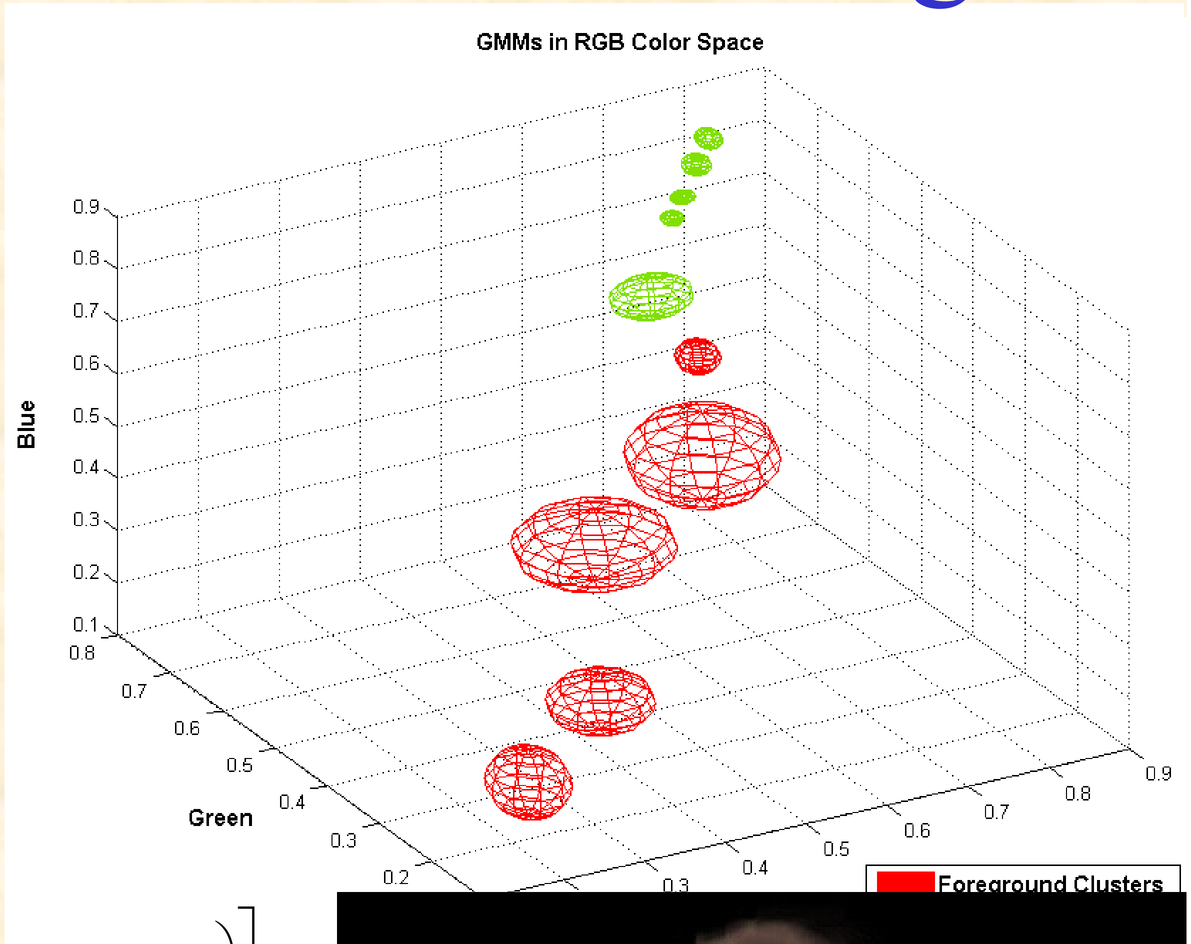
$$edge\_potential(x,y) = k_1 + k_2 \exp\left\{\frac{-\|c(x)-c(y)\|^2}{2\sigma^2}\right\}$$

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

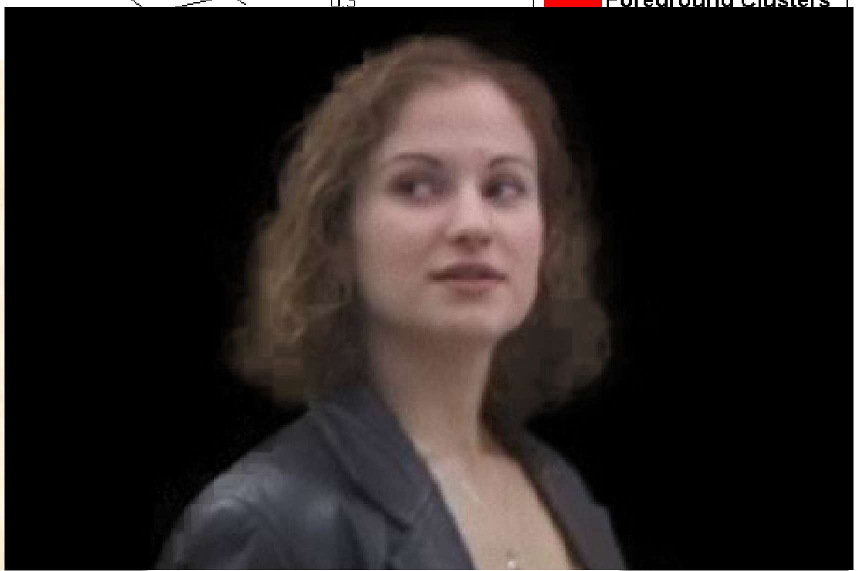
Final State



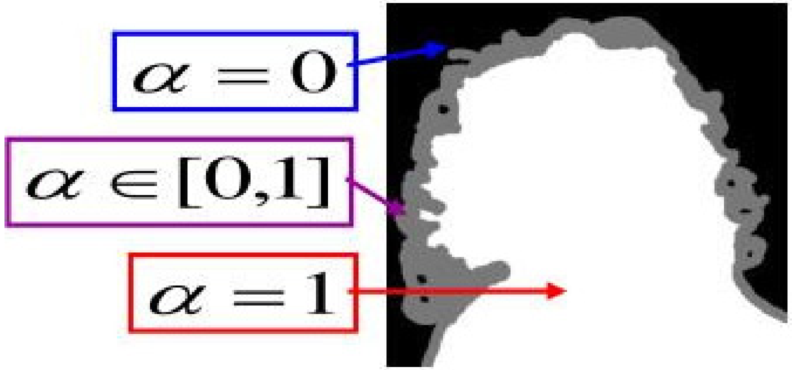
$$P(m) = \log \sum_{i=1}^{K} \left[w_i \frac{1}{\sqrt{\det \Sigma_i}} \times \exp\left(\frac{1}{2} [I_m - \mu_i]^T \Sigma_i^{-1} [I_m - \mu_i] \right) \right]$$

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

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

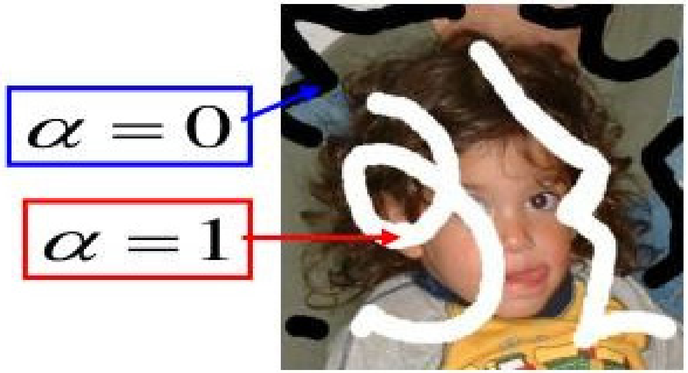


Previous approaches



The trimap interface:

- Bayesian Matting (Chuang et al, CVPR01)
- Poisson Matting (Sun et al SIGGRAPH 04)
- Random Walk (Grady et al 05)



Scribbles interface:

- Wang&Cohen ICCV05

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i$$

$$\alpha_i \approx a I_i + b, \quad \forall i \in w,$$

where $a =$ and w is a small image window.

goal in this paper will be to find α, a and b minimizing the cost function

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - a_j I_i - b_j)^2 + \varepsilon a_j^2 \right), \qquad (3)$$

where w_j is a small window around pixel j.

The cost function is quadratic in **α, a**, and **b**, with **3N** unknowns for an image with **N** pixels.

Proof of Eqn (3) to Eqn. (5)/Theorem (1), is derived in paper (read it):

A Closed Form Solution to Natural Image Matting

Anat Levin, Dani Lischinski, Yair Weiss; CVPR-2006.

where Σ_k is a 3×3 covariance matrix, μ_k is a 3×1 mean vector of the colors in a window w_k, and I_3 is the 3×3 identity matrix.

We refer to the matrix L in equations (5) and (12) as the *matting Laplacian*. Note that the elements in each row of L sum to zero, and therefore the nullspace of L includes the constant vector. If $\varepsilon = 0$ is used, the nullspace of L also includes every color channel of I.

$$\sum_{k|(i,j)\in w_k} \left(\delta_{ij} - \frac{1}{|w_k|} \left(1 + \frac{1}{\frac{\varepsilon}{|w_k|} + \sigma_k^2} (I_i - \mu_k)(I_j - \mu_k) \right) \right) \tag{5}$$

Here δ_{ij} is the Kronecker delta, μ_k and σ_k^2 are the mean and variance of the intensities in the window w_k around k, and $|w_k|$ is the number of pixels in this window.

Color-channel:

Here L is an $N \times N$ matrix, whose (i,j)-th element is:

$$\sum_{k|(i,j)\in w_k} \left(\delta_{ij} - \frac{1}{|w_k|} \left(1 + (I_i - \mu_k)(\Sigma_k + \frac{\varepsilon}{|w_k|} I_3)^{-1} (I_j - \mu_k) \right) \right)$$

To extract an ***alpha matte*** matching the user's constraints, we solve for:

$$\alpha = \operatorname{argmin}\ \alpha^T L\,\alpha + \lambda(\alpha^T - b_S^T)\,D_S\,(\alpha - b_S), \qquad (13)$$

where λ is some large number, D_S is a diagonal matrix whose diagonal elements are one for constrained pixels and zero for all other pixels, and b_S is the vector containing the specified alpha values for the constrained pixels and zero for all other pixels.

Since the above cost is quadratic in alpha, the global minimum may be found by differentiating (13) and setting the derivatives to zero. This amounts to solving the following sparse linear system

$$(L + \lambda D_S)\alpha = \lambda b_S$$

Theorem 3. *Let I be an image formed from F and B according to the compositing equation (1), and let α^* denote the true alpha matte. If F and B satisfy the color line model in every local window w_k and if the user-specified constraints S are consistent with α^*, then α^* is an optimal solution for the system (13), where L is constructed with $\epsilon = 0$.*

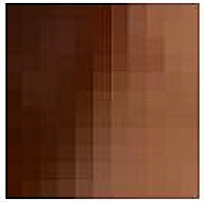
RY 2008

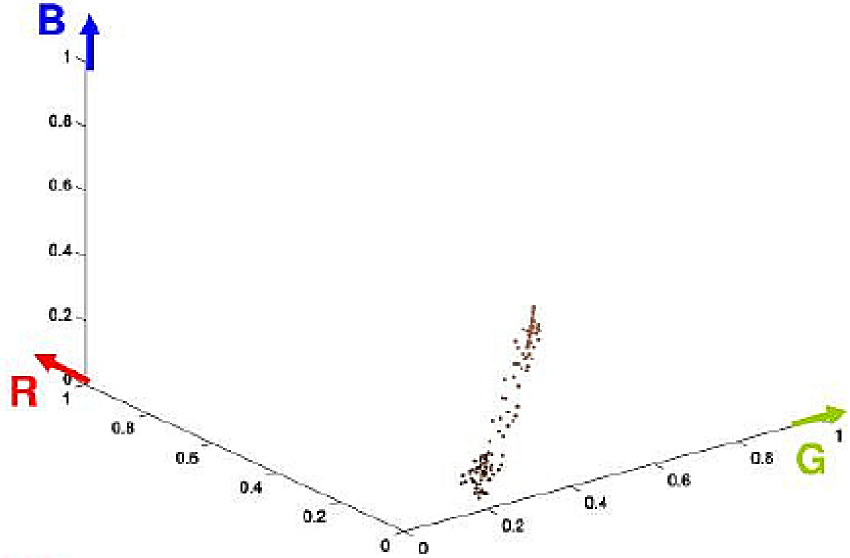
Natural

Anat Levin, Dani Lischinski, and Yair Weiss

Color lines

Color Line: $\{C_i \in R^3 \mid C_i = \beta_i C_1 + (1-\beta_i) C_2\}$





The matting equations

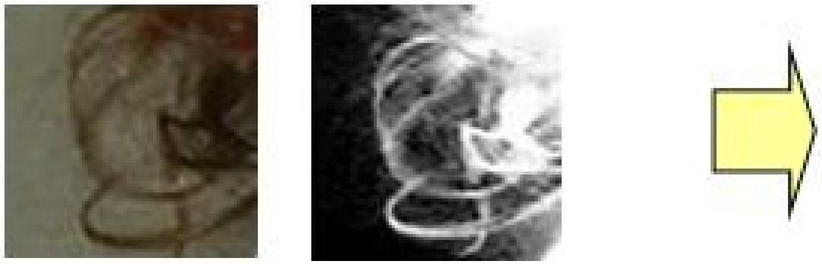
$$I_i = \alpha_i F_i + (1-\alpha_i) B_i$$

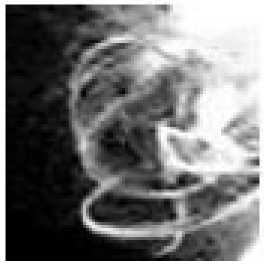
Linear model from color lines

Observation:

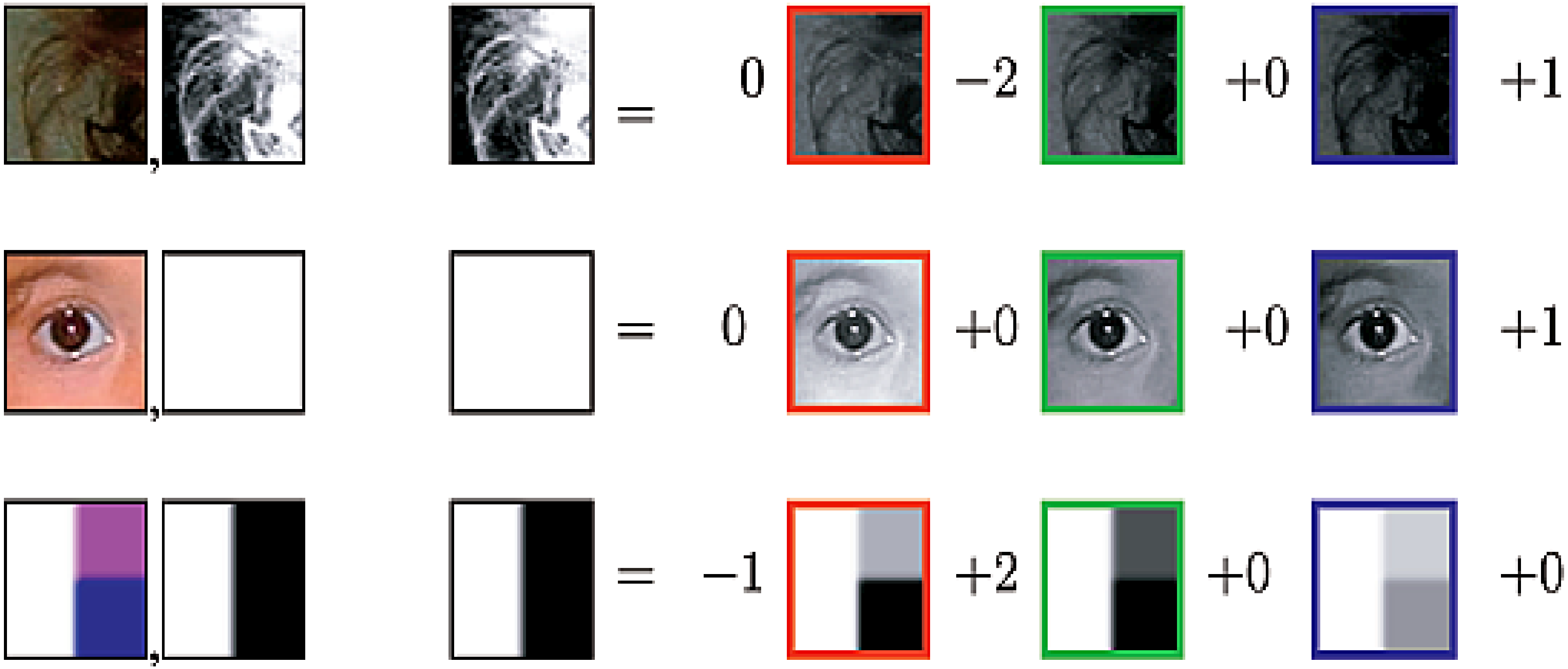
If the F,B colors in a local window lie on a color line, then

$$\alpha_i = a^R R_i + a^G G_i + a^B B_i + b \qquad \forall i \in w$$





$= -2$  $+1$



$$I_i = \alpha_i F + (1 - \alpha_i) B$$

Image and matte pair

Expressing the matte as a linear combination of color channels

Local linear relations between alpha windows and image windows.

$$\alpha_i \approx a I_i + b, \quad \forall i \in w,$$

(a) (b) (c) (d) (e)

Fig. 11. Foreground and background reconstruction: (a) input, (b) Alpha matte, (c) foreground reconstruction, (d) background reconstruction, and (e) F composited over a novel background.

Why is matting hard?

$$I_i = \alpha_i F_i + (1-\alpha_i) B_i$$

Matting is ill posed: 7 unknowns but 3 constraints per pixel

Input Image | Conventional Segmentation Result | Spectral Matting Result

Supervised Matting

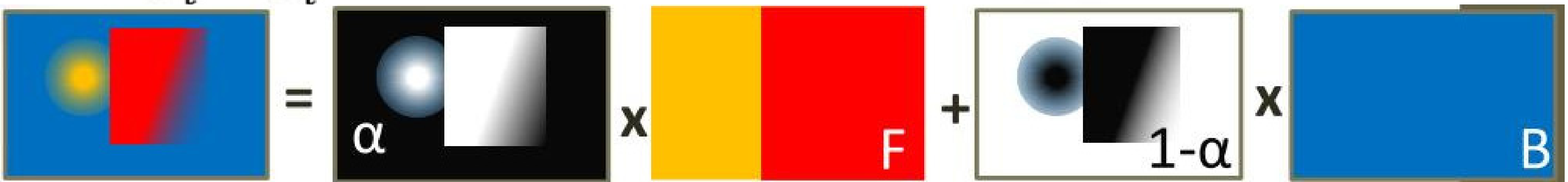
Input Image | Trimap (user's constraint) | Alpha Matte

Local Models for Alpha Mattes

$I_i = \alpha_i F_i + (1 - \alpha_i) B_i$ *α, F,* and *B* are unknown → ill-posed problem

$$\alpha_i = \frac{I_i - B_i}{F_i - B_i} \approx aI_i + b \quad , \forall i \in w$$

Assume a and b are constant in a small window

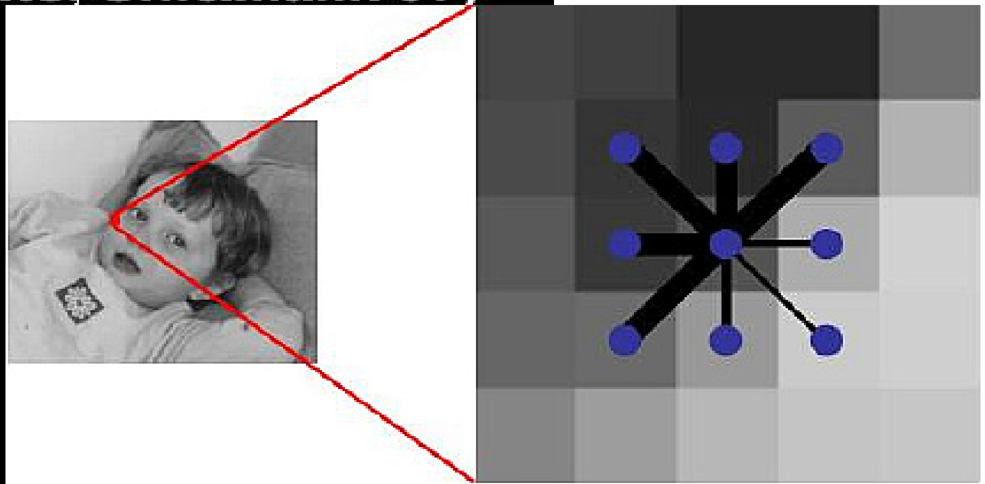


$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - a_j I_i - b_j)^2 + \varepsilon a_j^2 \right)$$

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

Matting and spectral segmentation

Spectral segmentation: Analyzing smallest eigenvectors of a graph Laplacian L (E.g. Normalized Cuts, Shi&Malik 97)



$$L = D - W$$

$$D(i,i) = \sum_j W(i,j)$$

$$W_{Global}(i,j) = e^{-\|C_i - C_j\|^2 / \sigma^2}$$

$$W_{Matting}(i,j) \propto \sum_{k|(i,j)\in w_k} \left(1 + (C_i - \mu_k)^T (\Sigma_k + \varepsilon I_3)^{-1} (C_j - \mu_k)\right)$$

•Analytically eliminate F,B and obtain quadratic cost $\alpha^T L \alpha$

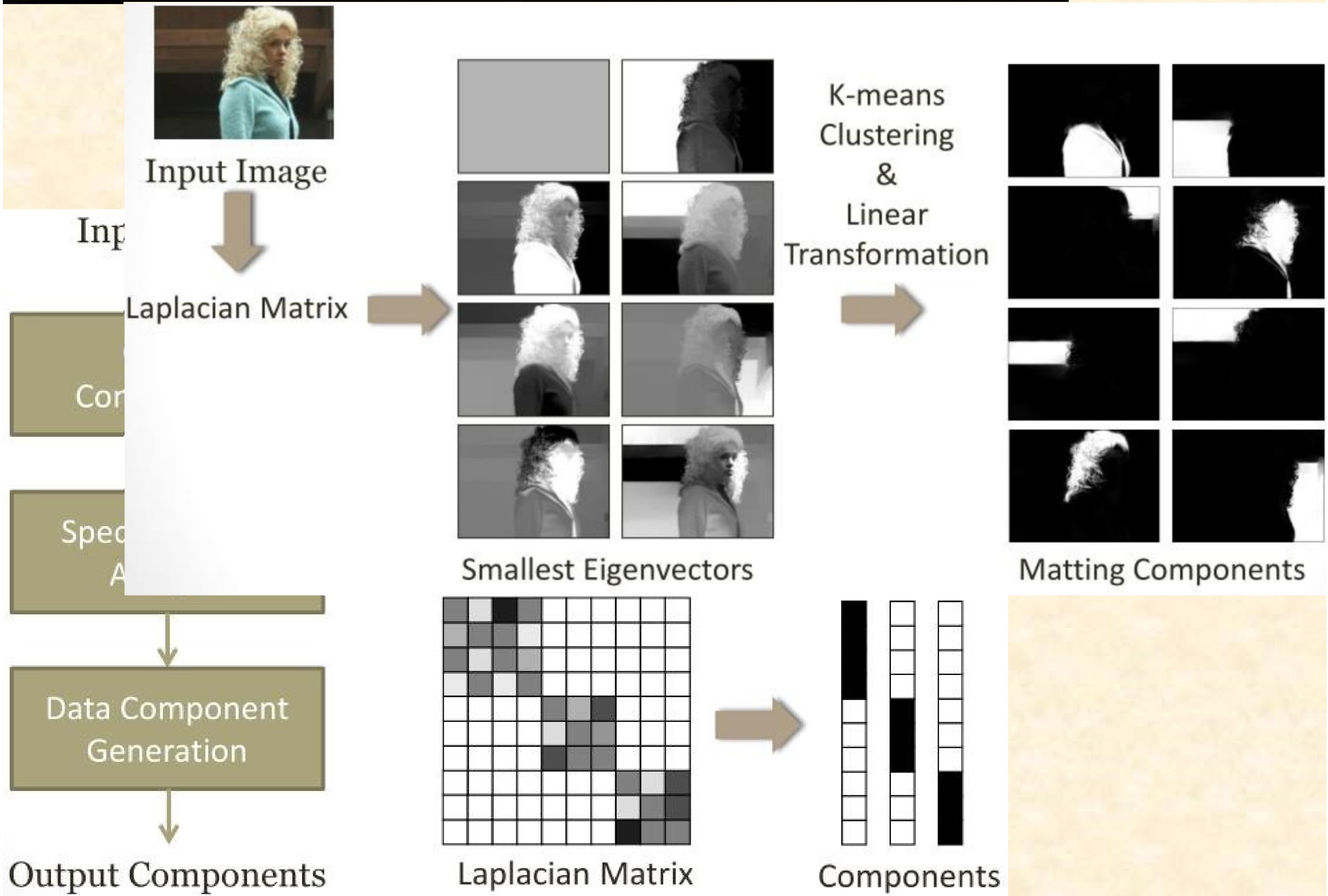
Solve efficiently using linear algebra.

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i$$

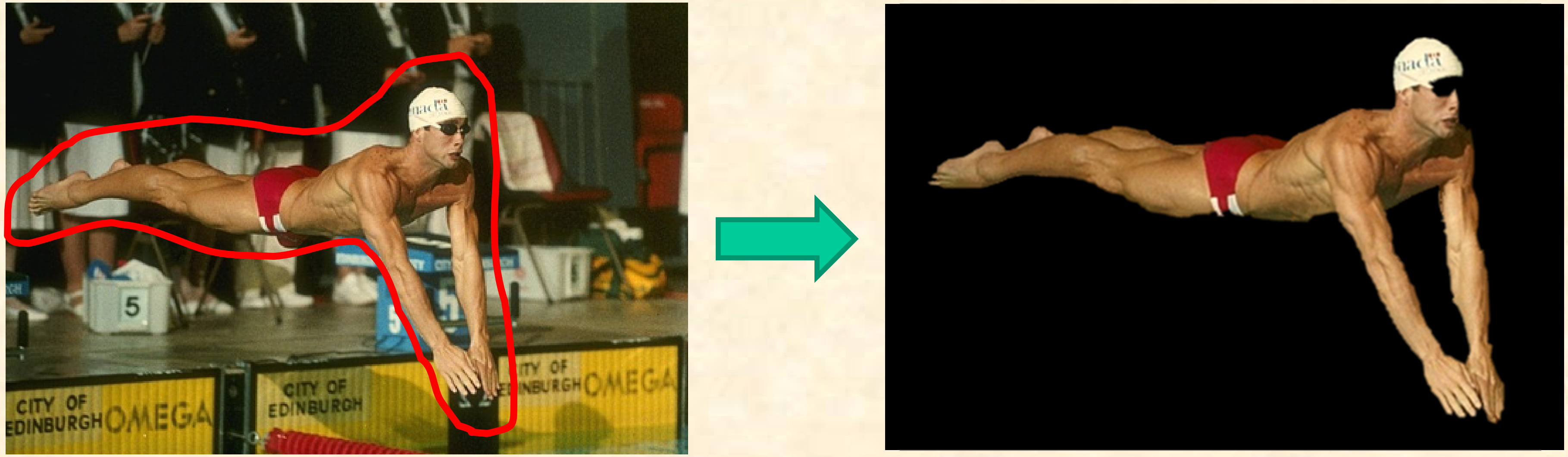
$$\alpha_i \approx a I_i + b, \quad \forall i \in w,$$

where $a = \frac{1}{F-B}$, $b = -\frac{B}{F-B}$ and w is a small image window.

Overview of Spectral Matting



GrabCut segmentation - Brief



User provides rough indication of foreground region.

Goal: Automatically provide a pixel-level segmentation.

$$\min \; J(\alpha) = \alpha^T L \alpha; \qquad I_i = \alpha_i F + (1-\alpha_i) B$$

For an image with N pixels:

L is an N ×N matrix (matting Laplacian) , whose (i, j)-th element is:

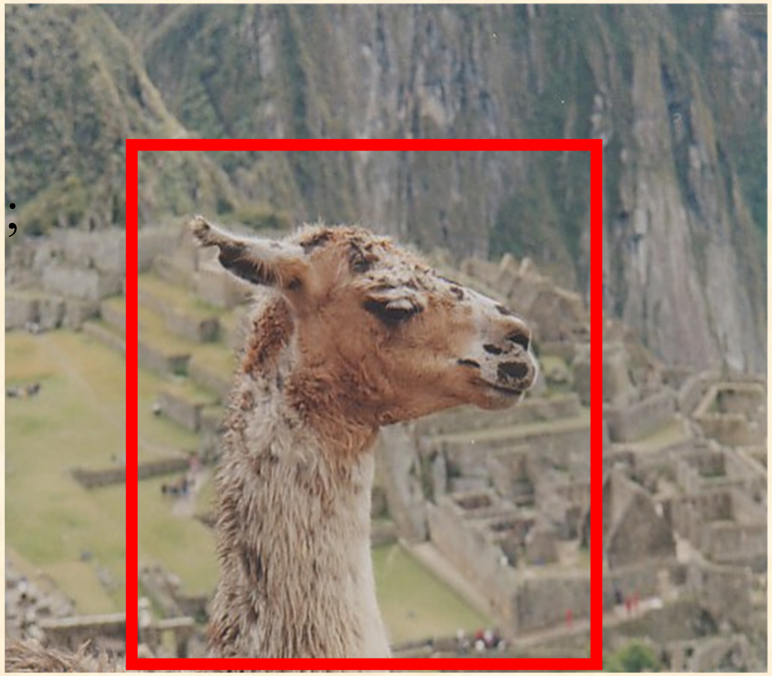
$$\sum_{k \mid (i,j) \in W_k} \left[\delta_{ij} - \frac{1}{|W_k|} \left(1 + (I_i - \mu_k) \left(\Sigma_k + \frac{\varepsilon}{|W_k|} I_3 \right)^{-1} (I_j - \mu_k) \right) \right]$$

GrabCut segmentation - Overview

$$I_i = \alpha_i F + (1-\alpha_i)B \qquad \min\ J(\alpha) = \alpha^T L \alpha;$$

$$L = \sum_{k|(i,j)\in W_k} \left[\delta_{ij} - \frac{1}{|W_k|}\left(1 + (I_i - \mu_k)\left(\Sigma_k + \frac{\varepsilon}{|W_k|} I_3\right)^{-1}(I_j - \mu_k)\right)\right]$$

User Input



Result

$|W_k|$ is the no. of pixels in window W_k;

δ_{ij} is the Kronecker delta;

Σ_k is a 3x3 covariance matrix;

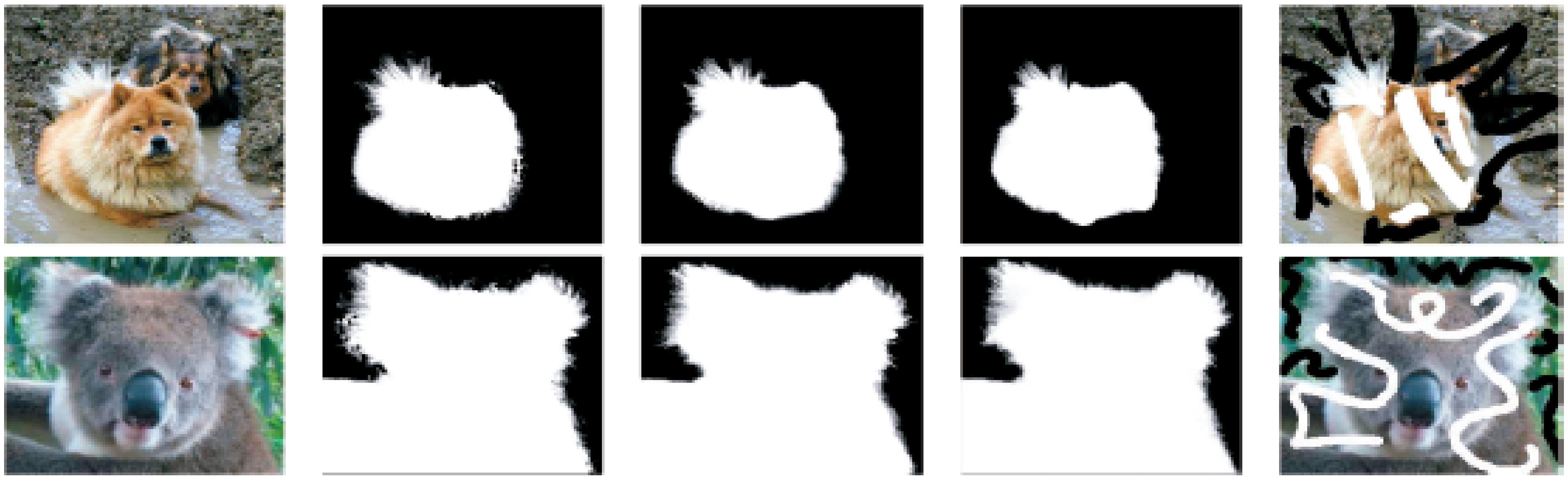
I_3 is a 3x3 Identity matrix.

For gray-scale data:

$$L = \sum_{k|(i,j)\in W_k} \left[\delta_{ij} - \frac{1}{|W_k|}\left(1 + \frac{1}{\frac{\varepsilon}{|W_k|} + \sigma_k^2}(I_i - \mu_k)(I_j - \mu_k)\right)\right]$$

(a) (b) (c) (d) (e)

Fig. 12. Comparison with Bayesian matting [6]. (a) Input image. (b) Trimap. (c) Bayesian matting result (obtained from the Bayesian Matting Web page). (d) Scribbles. (e) Our result.

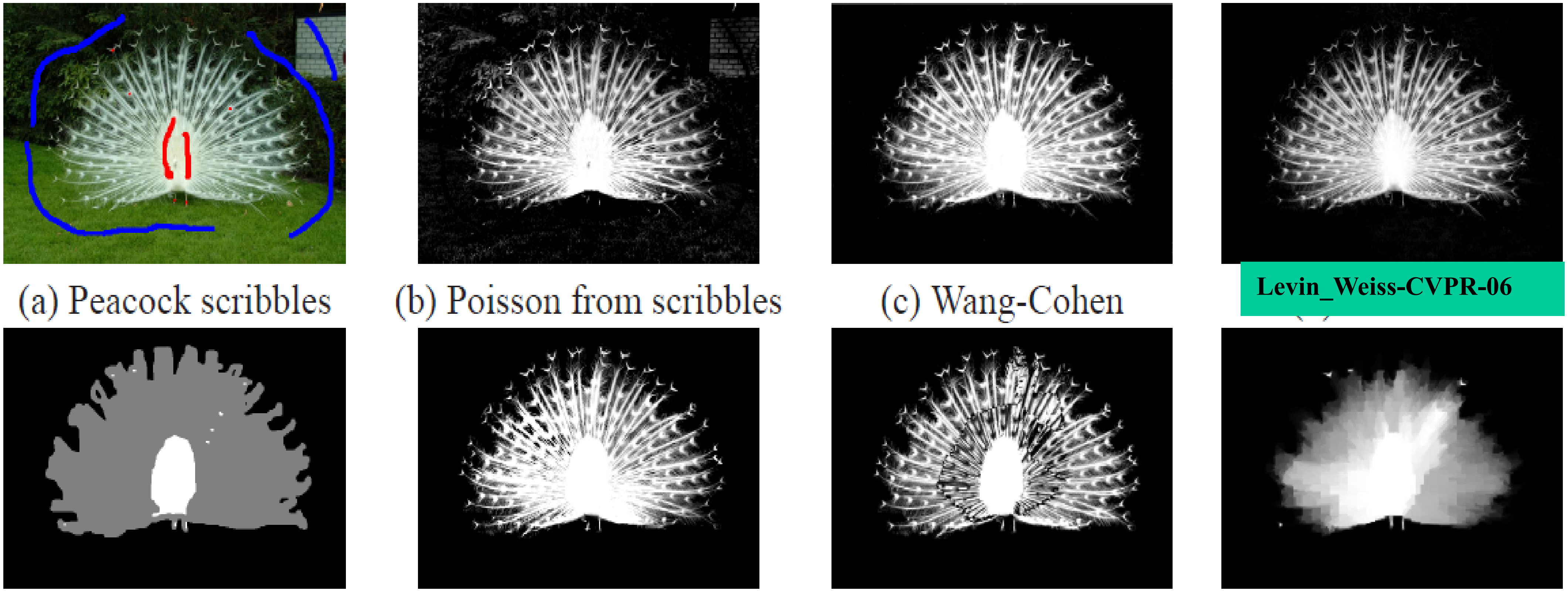


(a) (b) (c) (d) (e)

Fig. 13. Result on Poisson matting examples. (a) Input image. (b) Bayesian matting (obtained from the Poisson matting paper). (c) Poisson matting (obtained from the Poisson matting paper). (d) Our result. (e) Scribbles.

To design a matting Laplacian, as L = D-W,
Use the following "matting affinity" matrix:

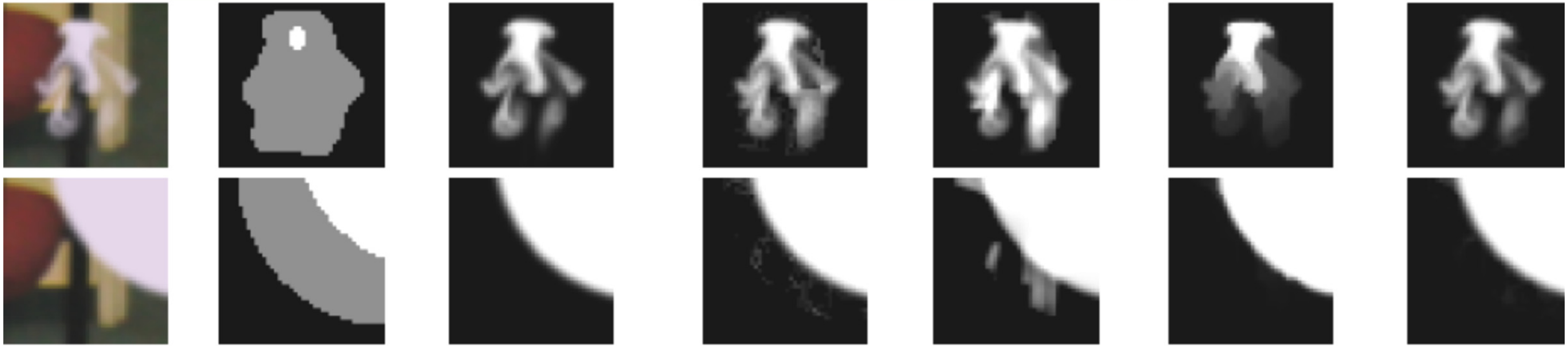
$$W_M(i,j) = \sum_{k|(i,j)\in W_k} \left[\frac{1}{|W_k|} \left(1 + (I_i - \mu_k) \left(\Sigma_k + \frac{\varepsilon}{|W_k|} I_3 \right)^{-1} (I_j - \mu_k) \right) \right]$$



(a) Peacock scribbles (b) Poisson from scribbles (c) Wang-Cohen

Levin_Weiss-CVPR-06

(e) Peacock trimap (f) Poisson from trimap (g) Bayesian (h) Random walk



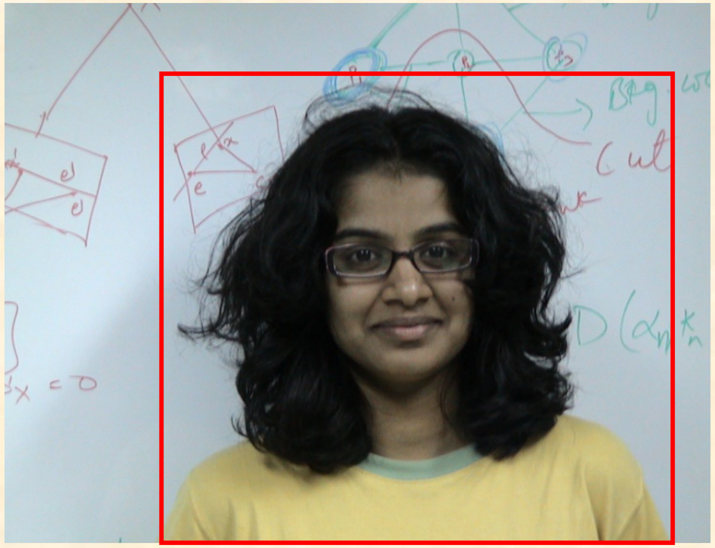
a) Composite (b) Trimap (c) Ground truth (d) Wang-Cohen (e) Poisson (f) Random walk (g) Our resul

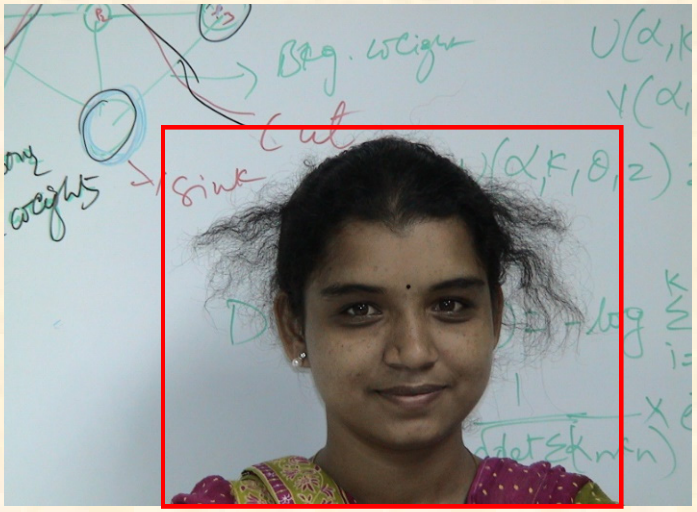
Input image Global σ eigenvectors Matting eigenvectors

Figure 3. Smallest eigenvectors of the different Laplacians.

Object Extraction From an Image

Results:





Object Extraction From an Image

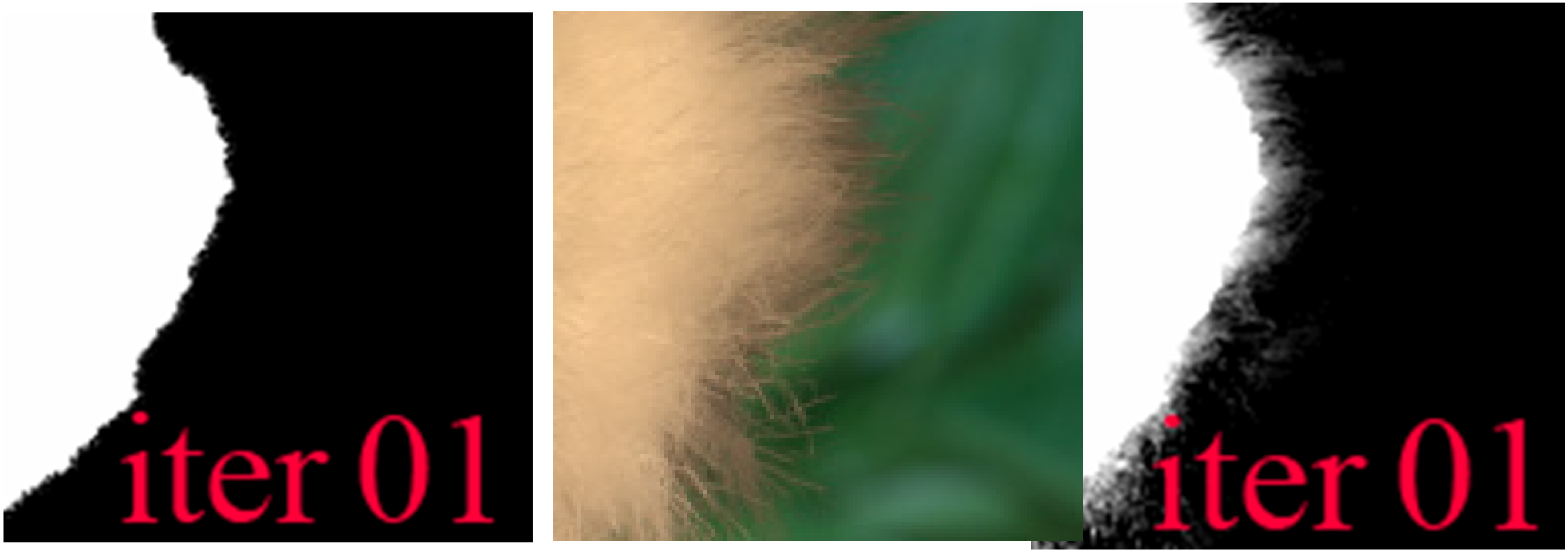
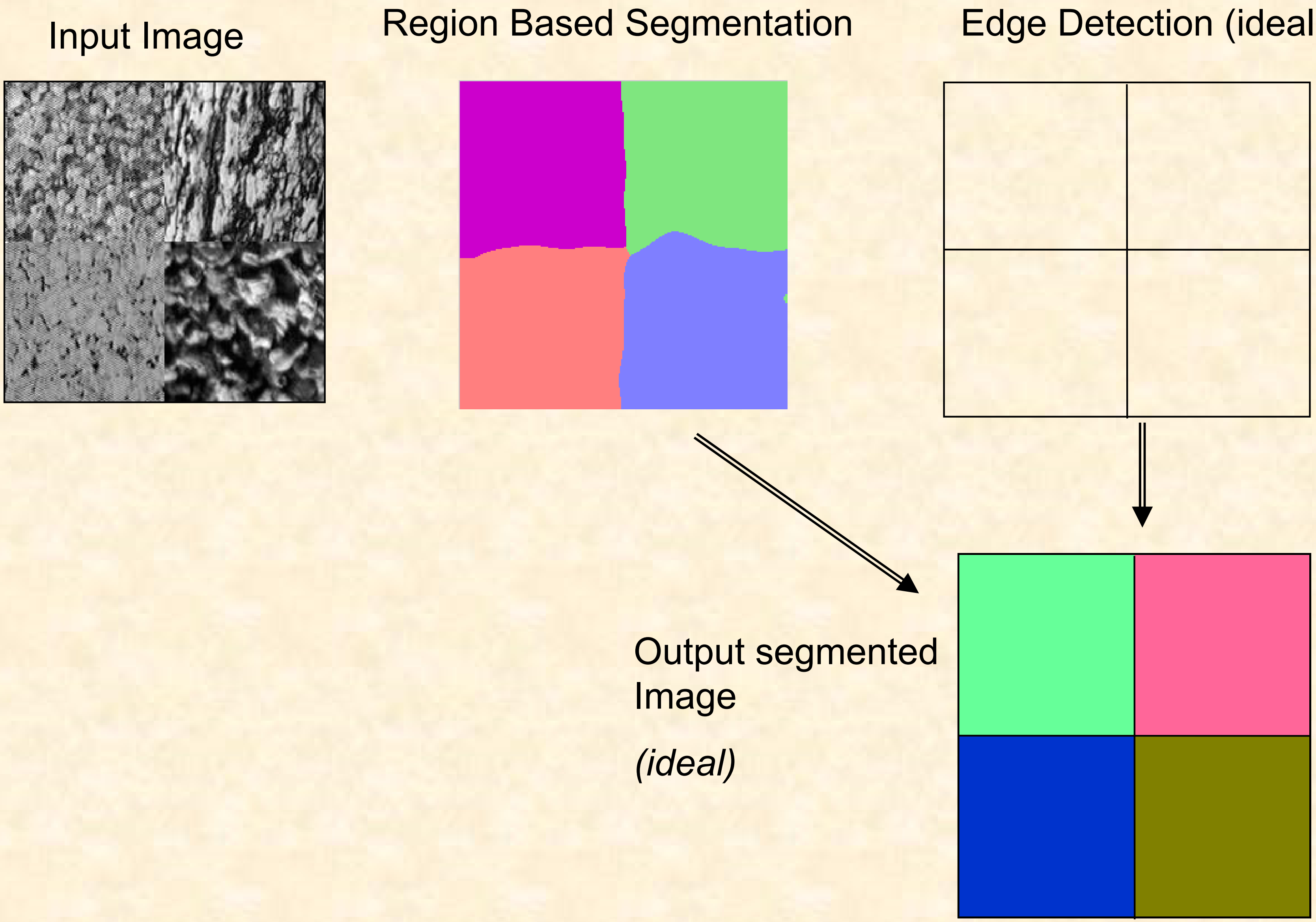

iter 01
iter 01

Image Segmentation –

Combining edge and region information

Example of Image Segmentation (ideal) based on fusion



CVPR 2004 Graph-Based Image Segmentation Tutorial

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**

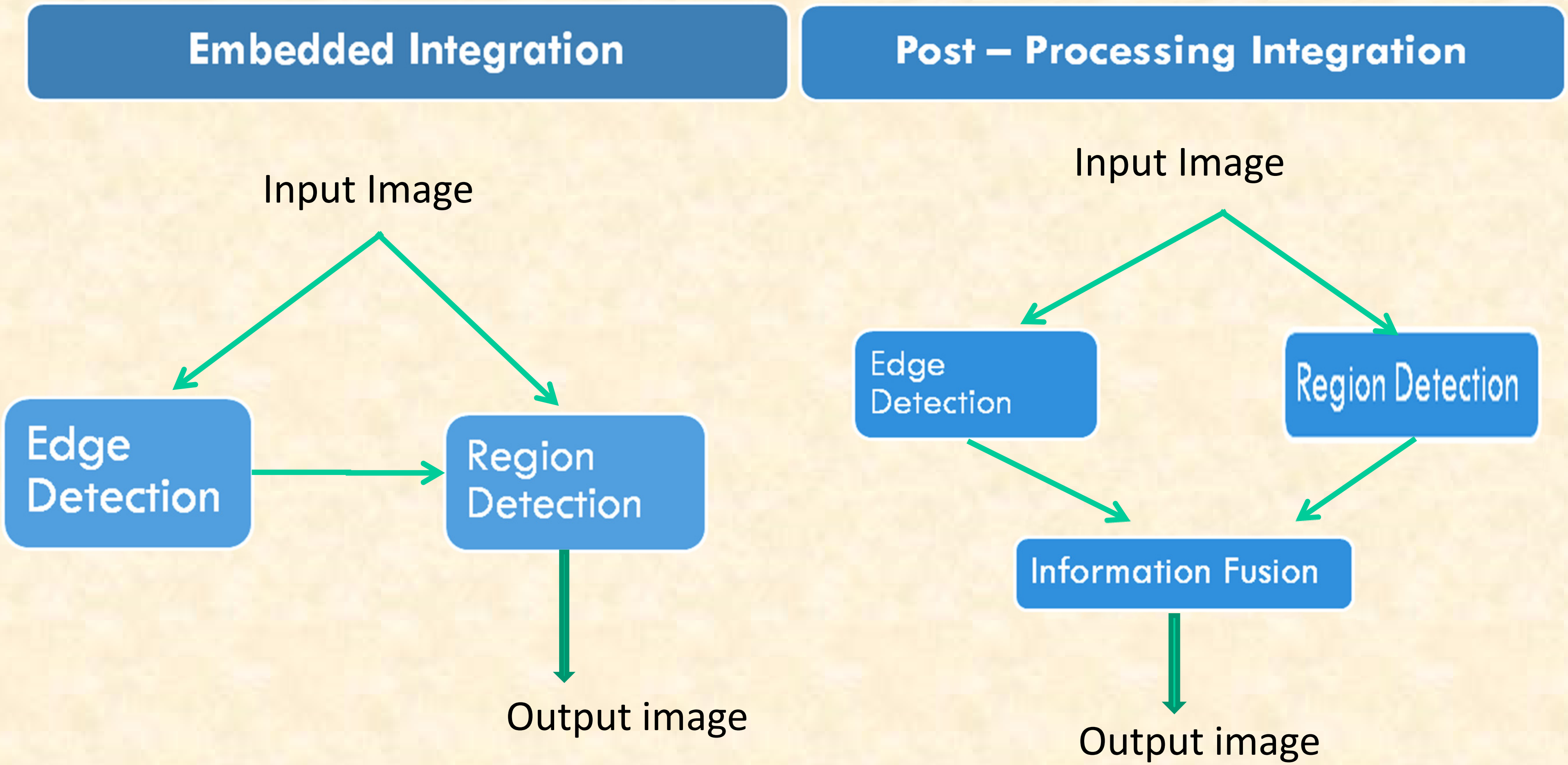
Integration Techniques

- 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.**

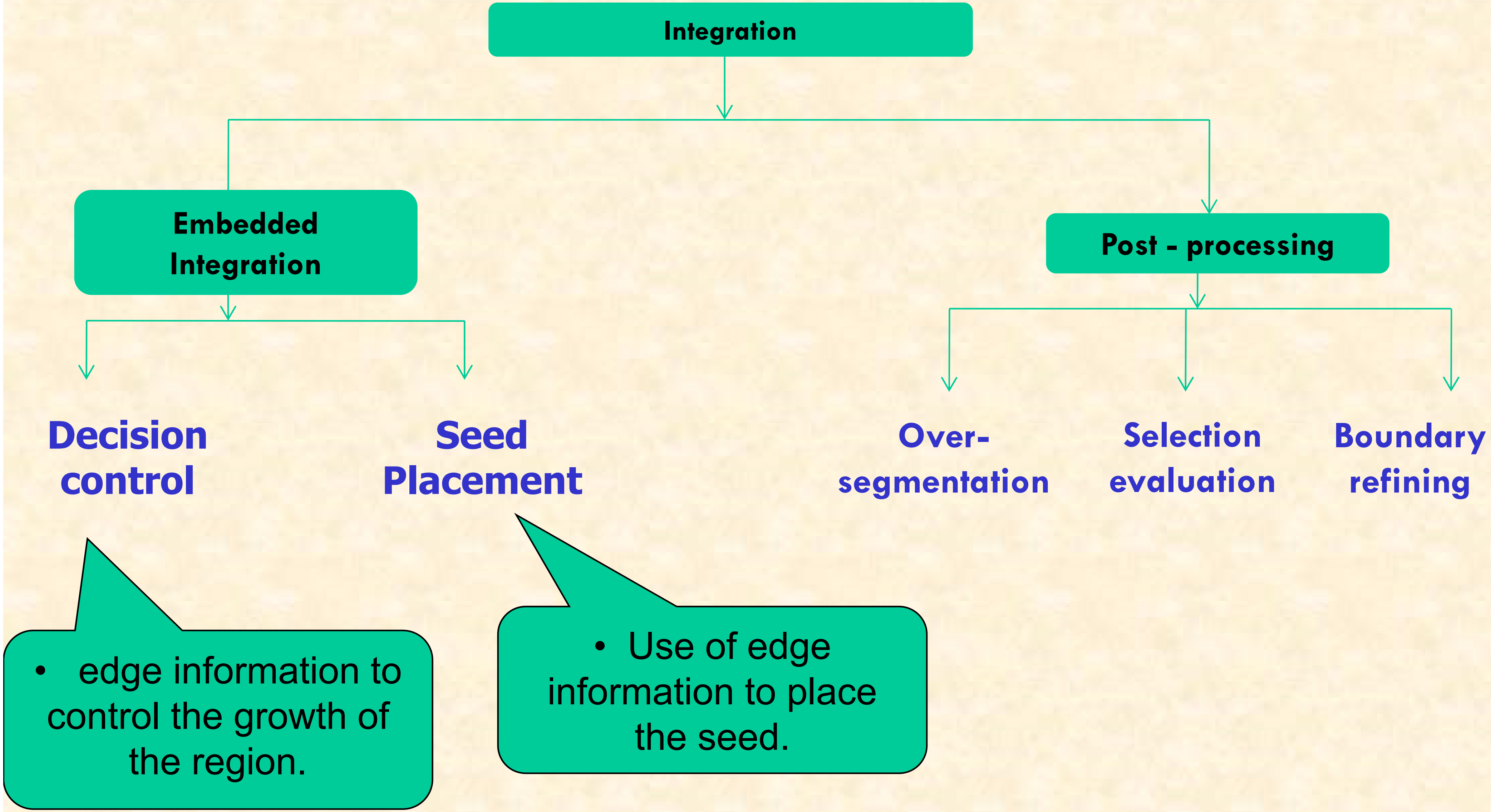
X. Munoz, J.freixenet, X. Cufi, J. Marti,
Strategies for image segmentation combining region and boundary information, Pattern Recognition Letters 24 (2003).

Integration Techniques



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

- Extracted edge information is used within region segmentation algorithm.

- Edge Information can be used in two ways

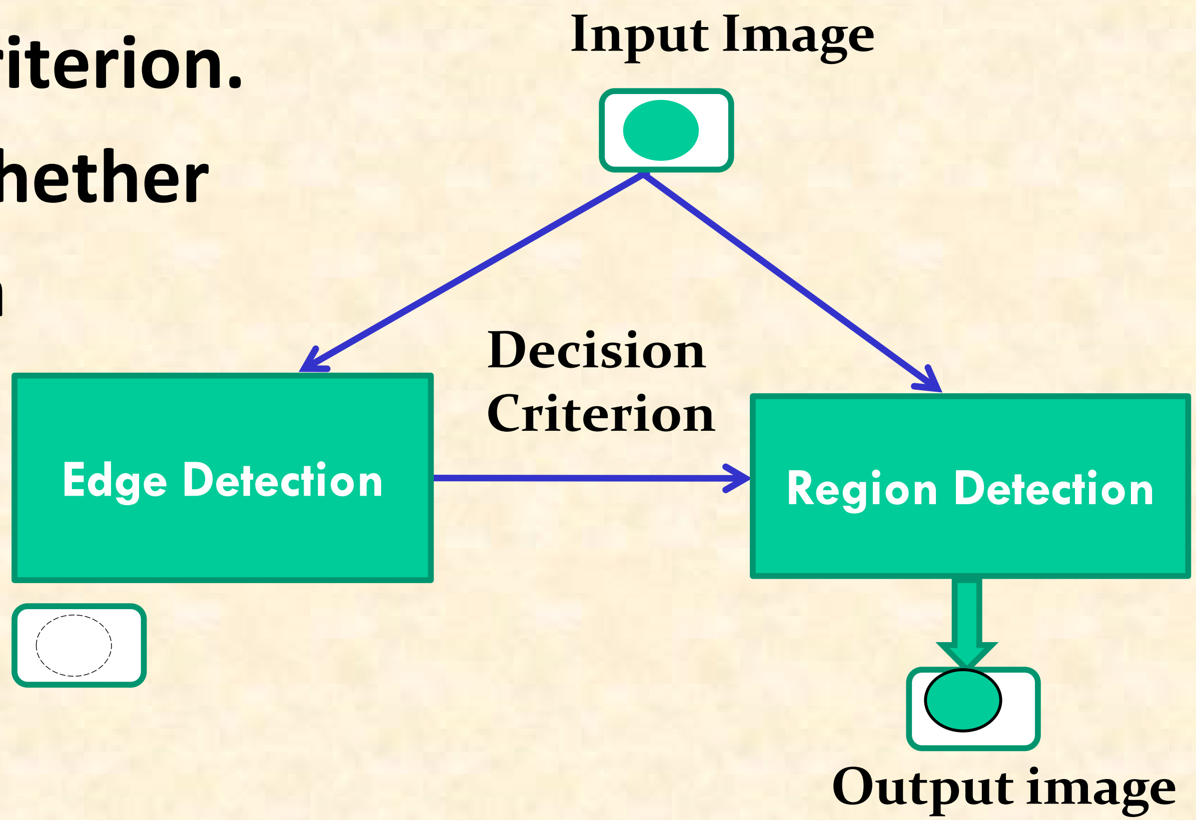
1. ***Control of decision criterion -*** edge information is included in the definition of decision criterion which controls the growth of the region.

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.

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

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

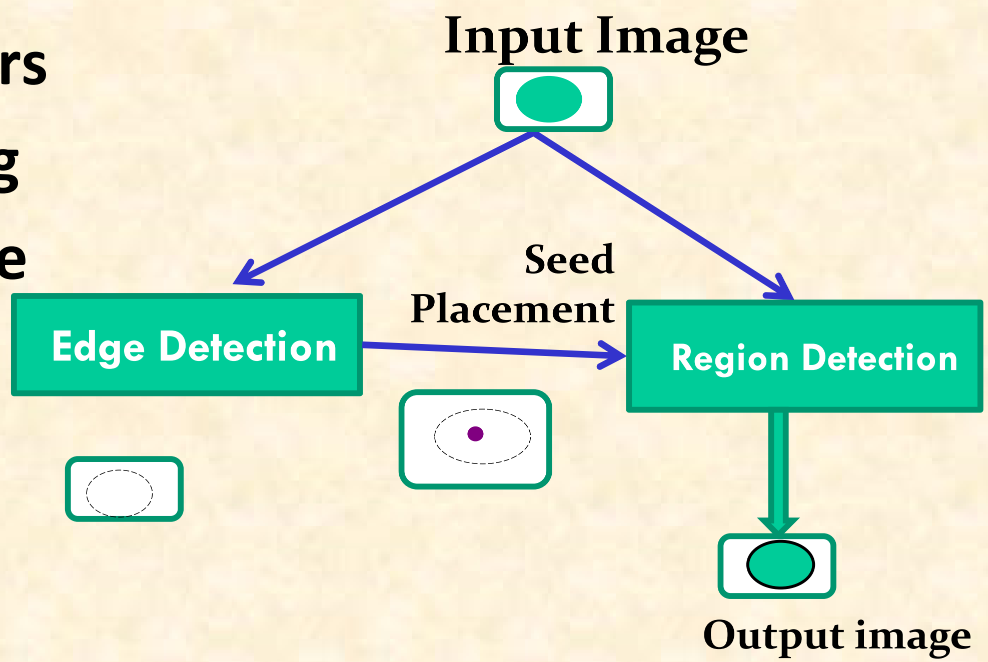


X. Munoz, J.freixenet, X. Cufi, J. Marti,
Strategies for image segmentation combining region and boundary information, Pattern Recognition Letters 24 (2003).

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.

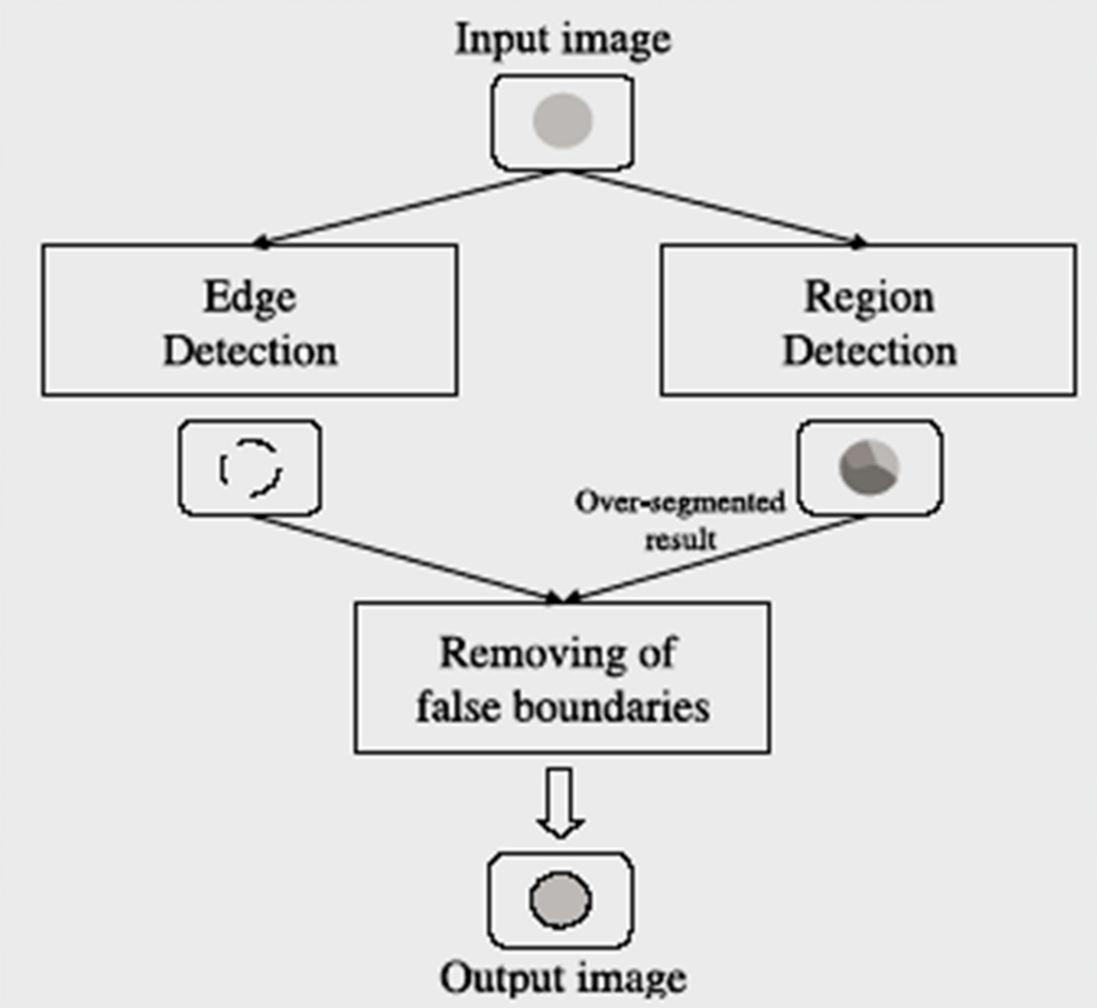
- Three different approaches

  (1) Over- segmentation

  (2) Boundary refinement

  (3) Selection- evaluation

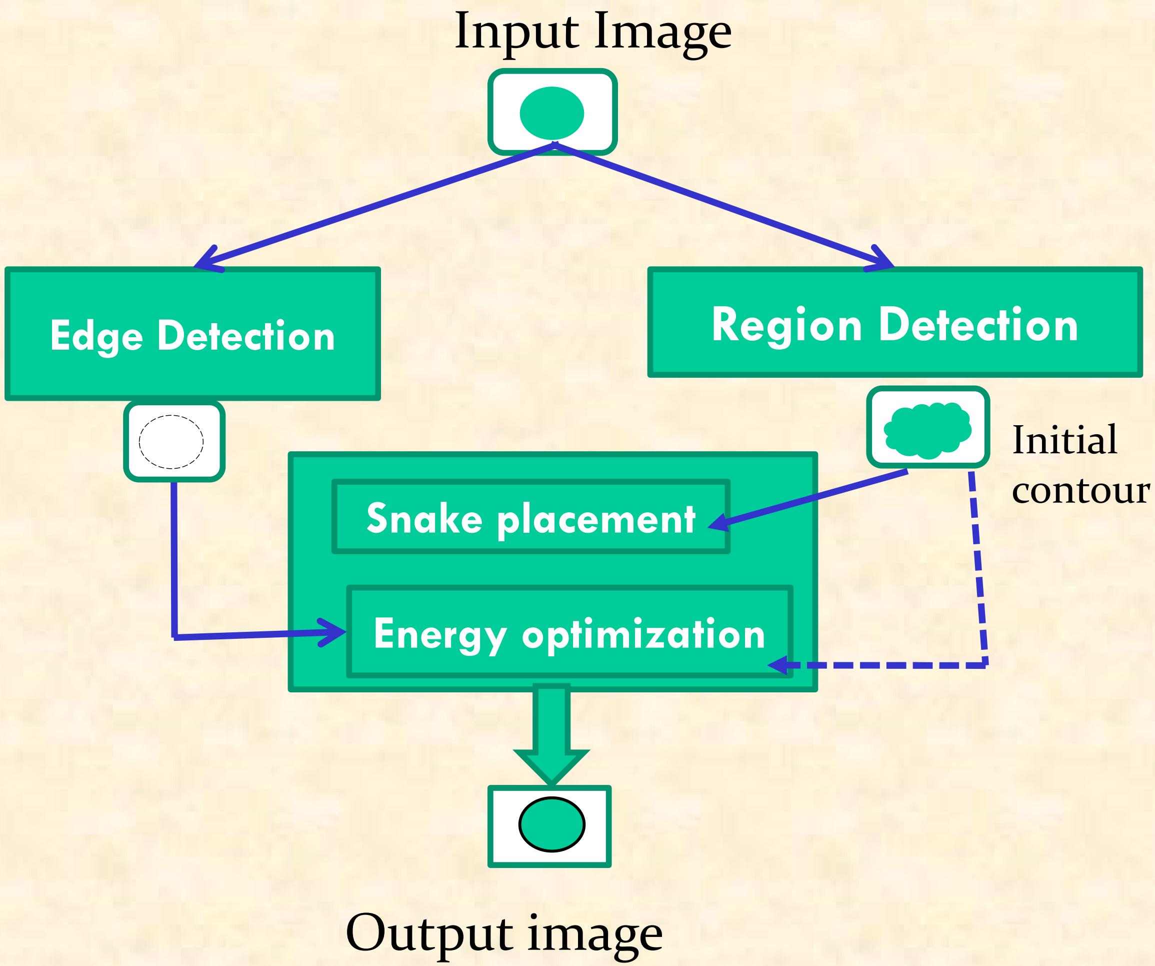
Over-segmentation



- 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.

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

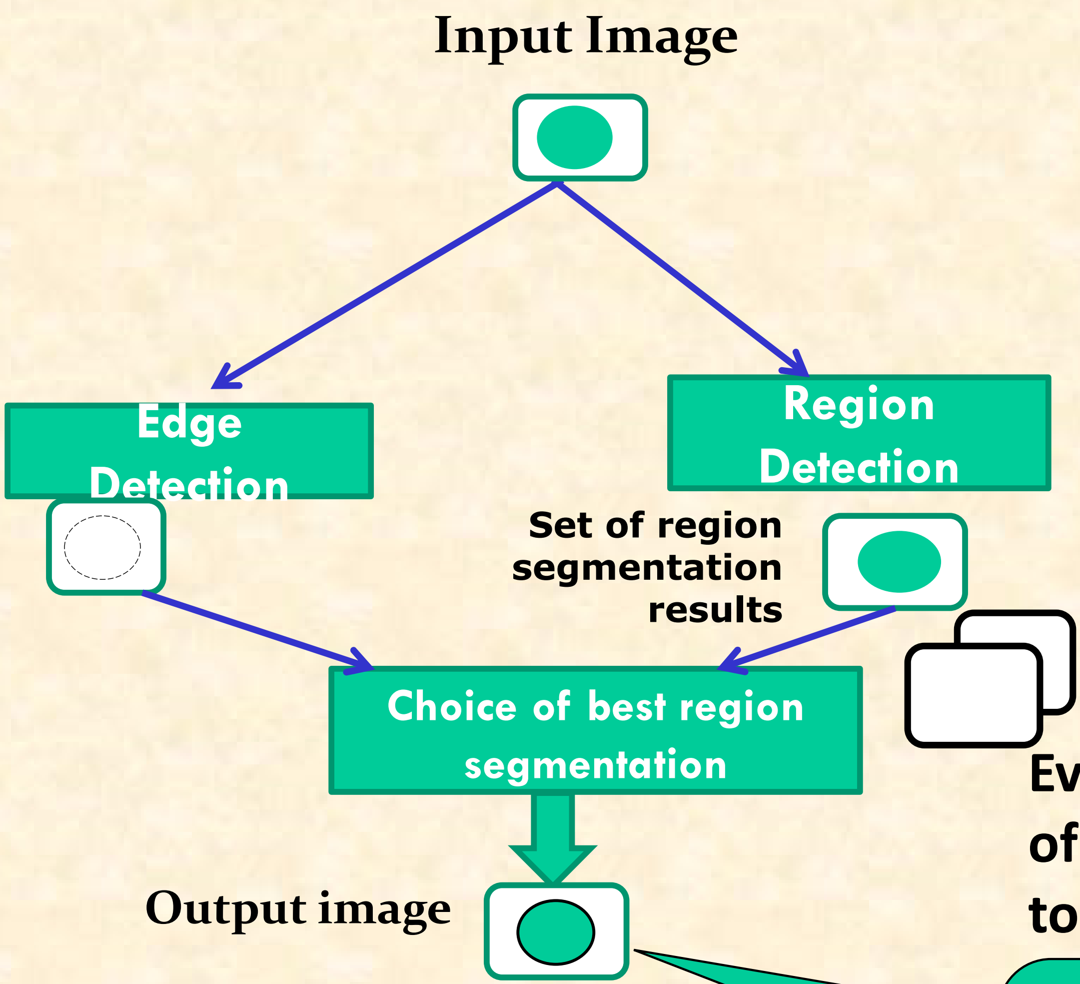
Boundary refinement



- 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

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

Selection- evaluation



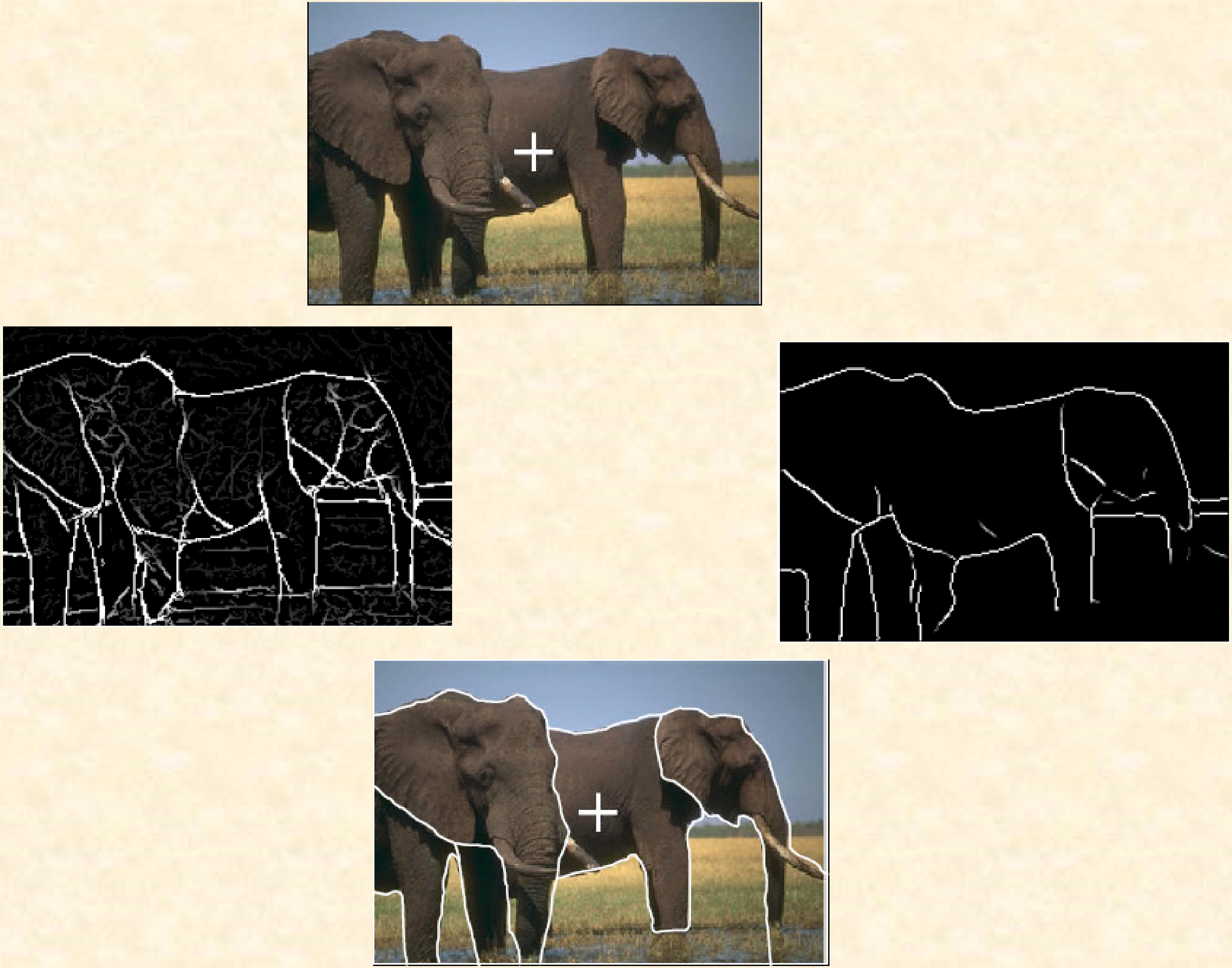
- Different results are achieved by changing parameters and thresholds in a region- segmentation algorithm

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

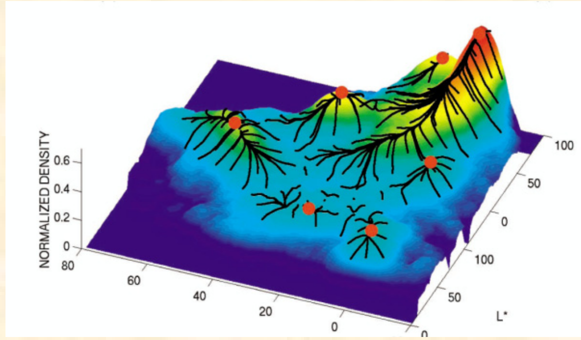
Evaluation function measures the quality of a region-based segmentation according to its consistency with the edge map

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

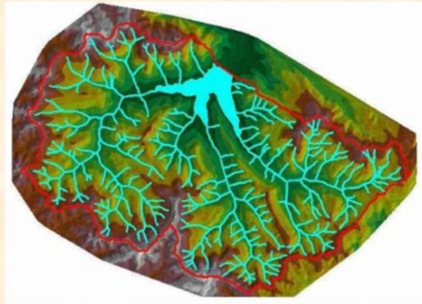
X. Munoz, J.freixenet, X. Cufi, J. Marti,
Strategies for image segmentation combining region and boundary information, Pattern Recognition Letters 24 (2003).



Segmentation

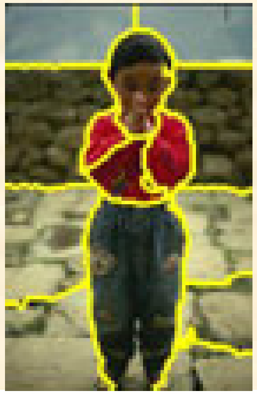


- Mean-shift segmentation
  - Flexible clustering method, good segmentation



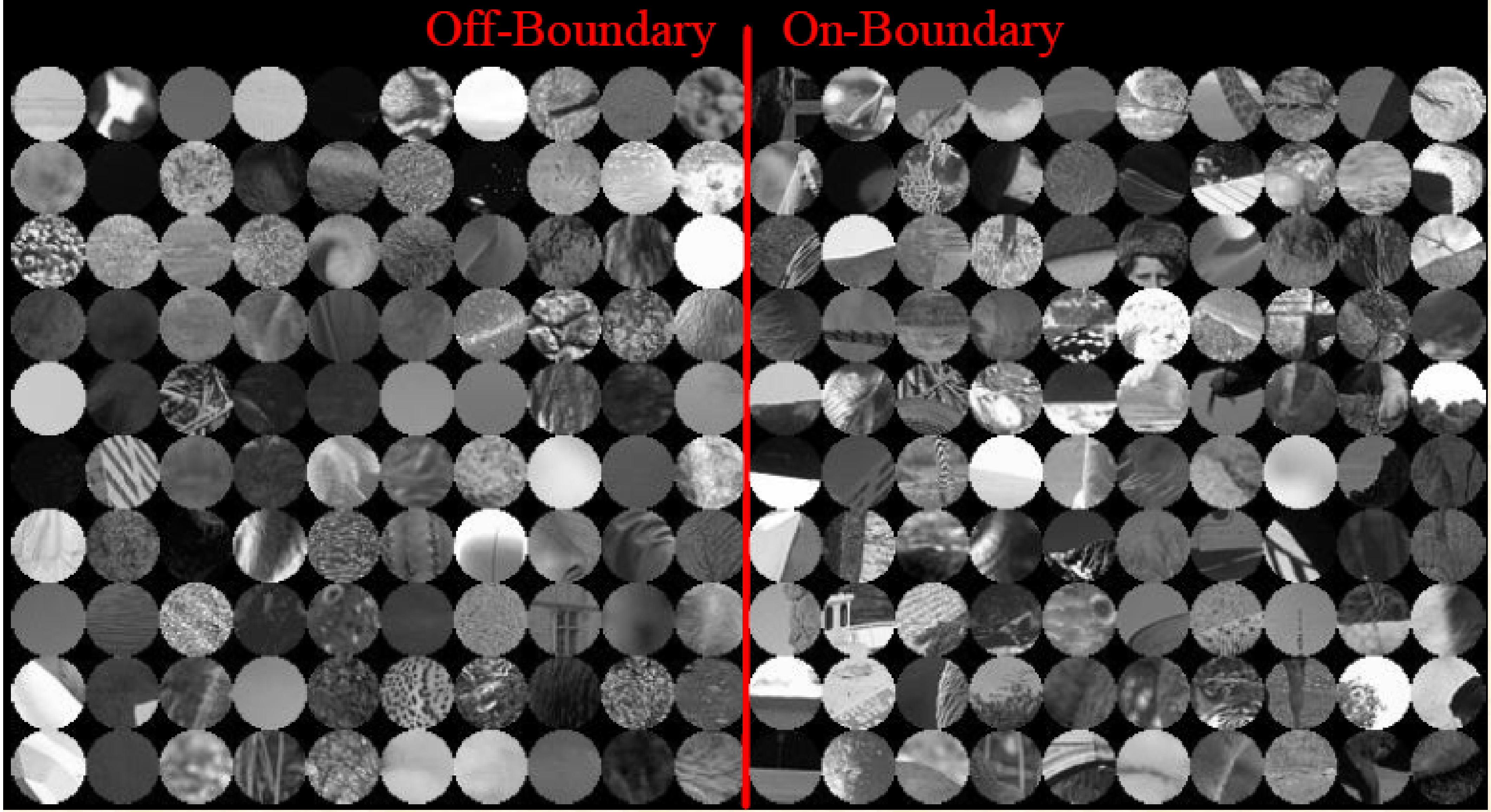
- Watershed segmentation
  - Hierarchical segmentation from soft boundaries

- **Normalized cuts**
  - Produces regular regions
  - Slow but good for oversegmentation



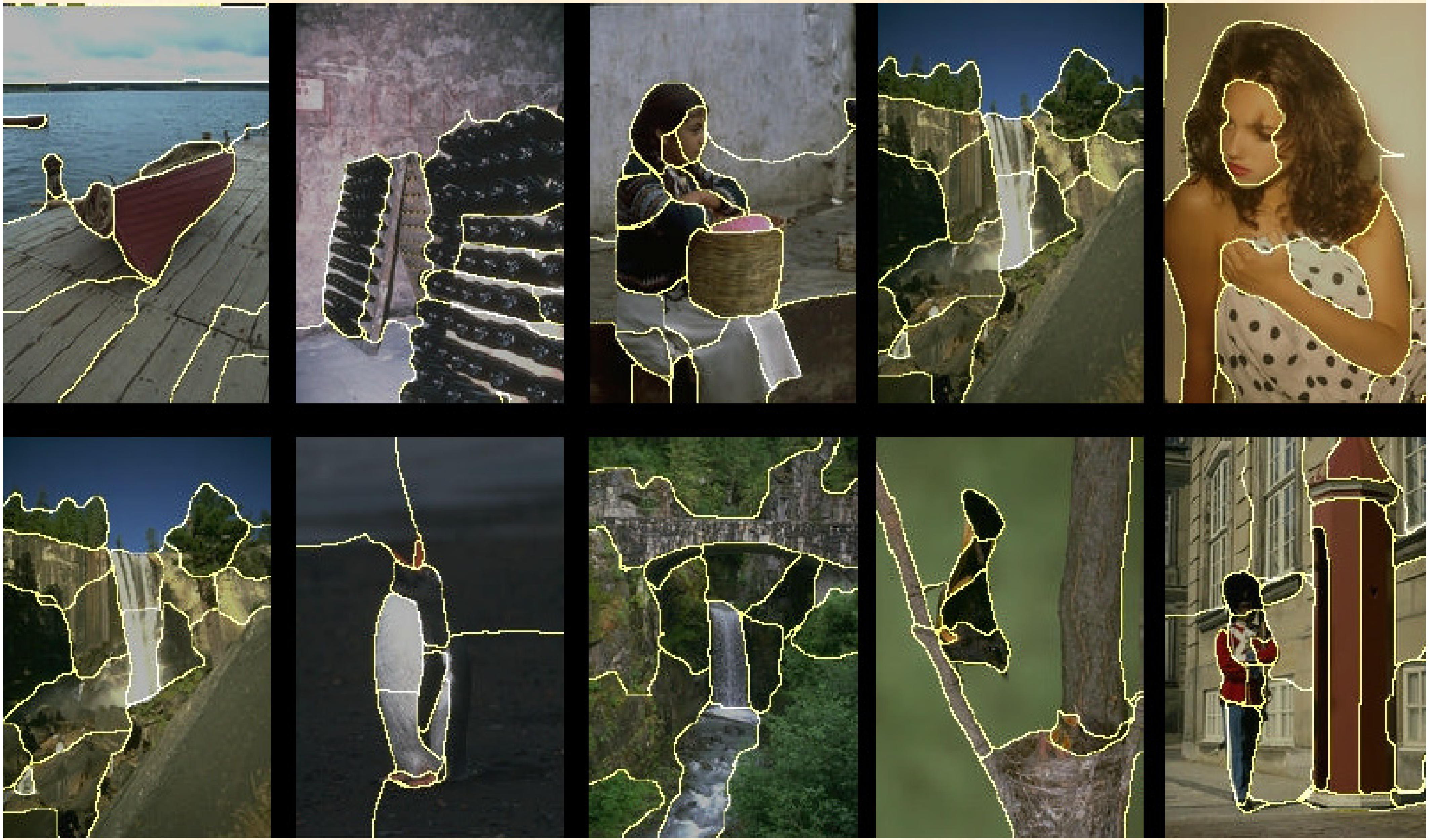
- **MRFs with Graph Cut**
  - Incorporates foreground/background/object model and prefers to cut at image boundaries
  - Good for interactive segmentation or recognition

How good are humans locally?



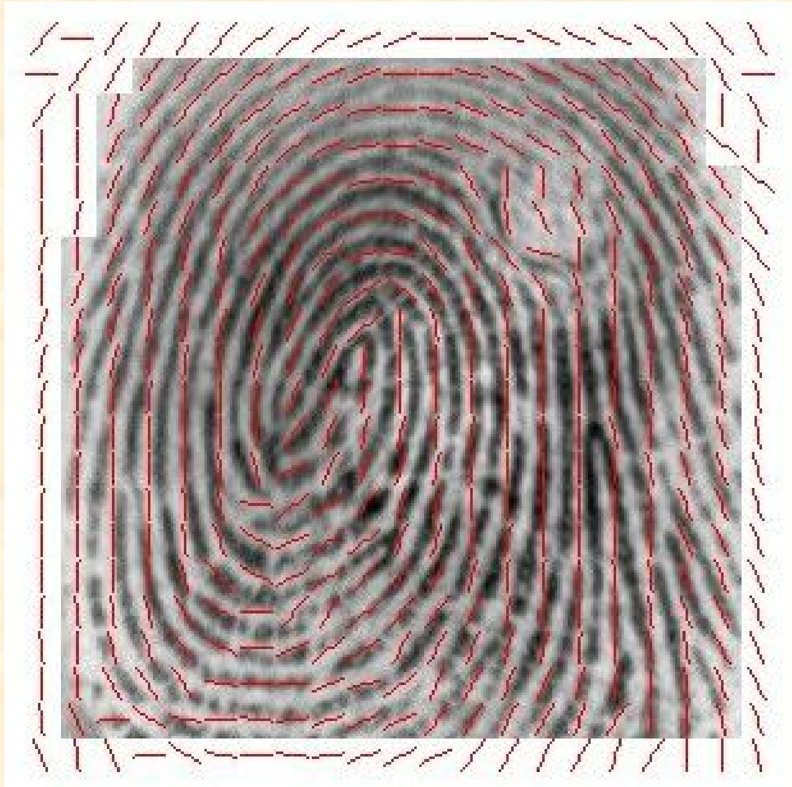
Modern methods for Image segmentation involve:

- **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;**
- **α Expansion; α–β swaps**
- **Snakes (Active Contours); Loopy-BP; QBPO; BP-M, BP-S**
- **Parametric Distributional clustering, Probabilistic approaches**
- **Deformable Templates, AAM, ASM**
- **Decision Trees and hierarchical analysis**
- **Neuro-fuzzy and soft-computing techniques – ACO, PSO etc.**
- **Mean-Shift;**



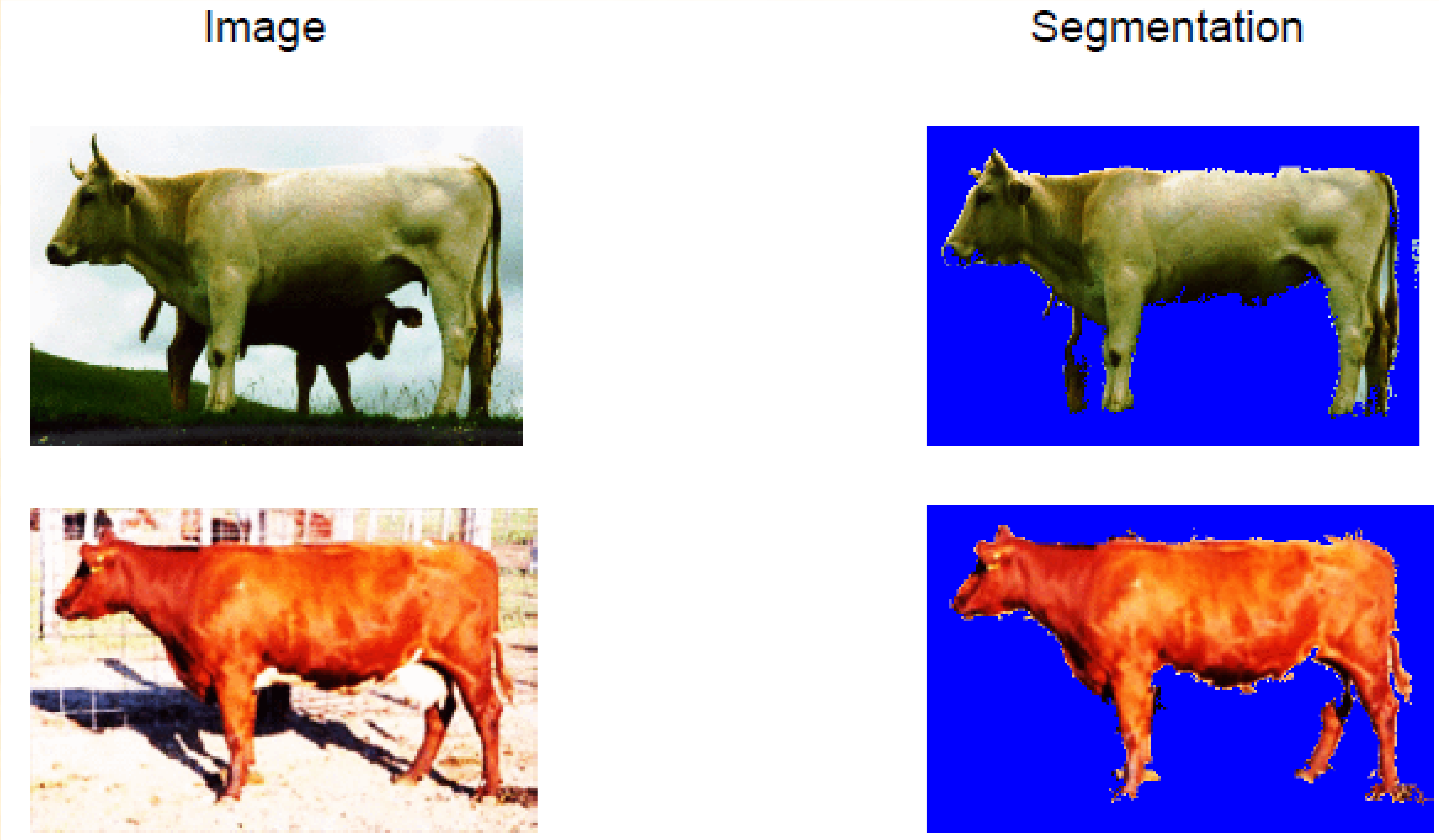
http://www.cs.berkeley.edu/~fowlkes/BSE/

Why not OK ?



OK

Object Detection



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

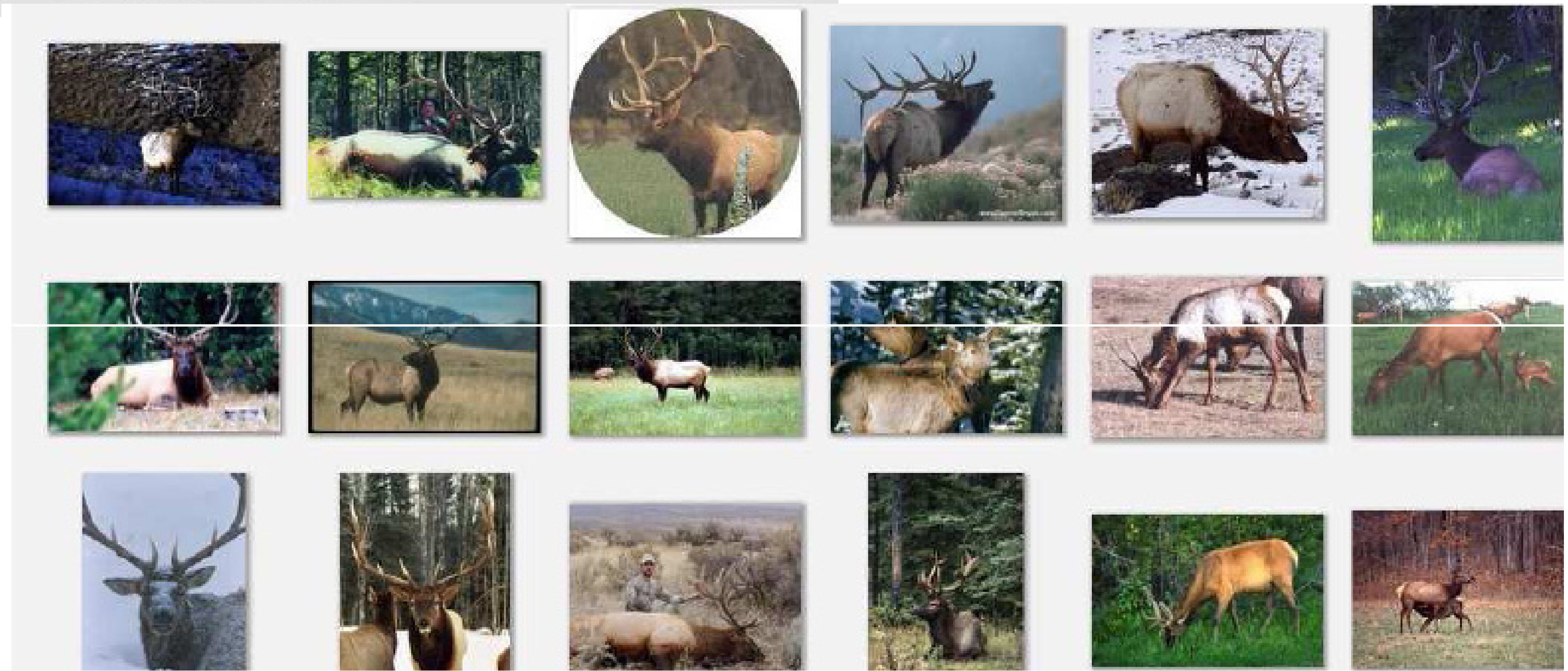
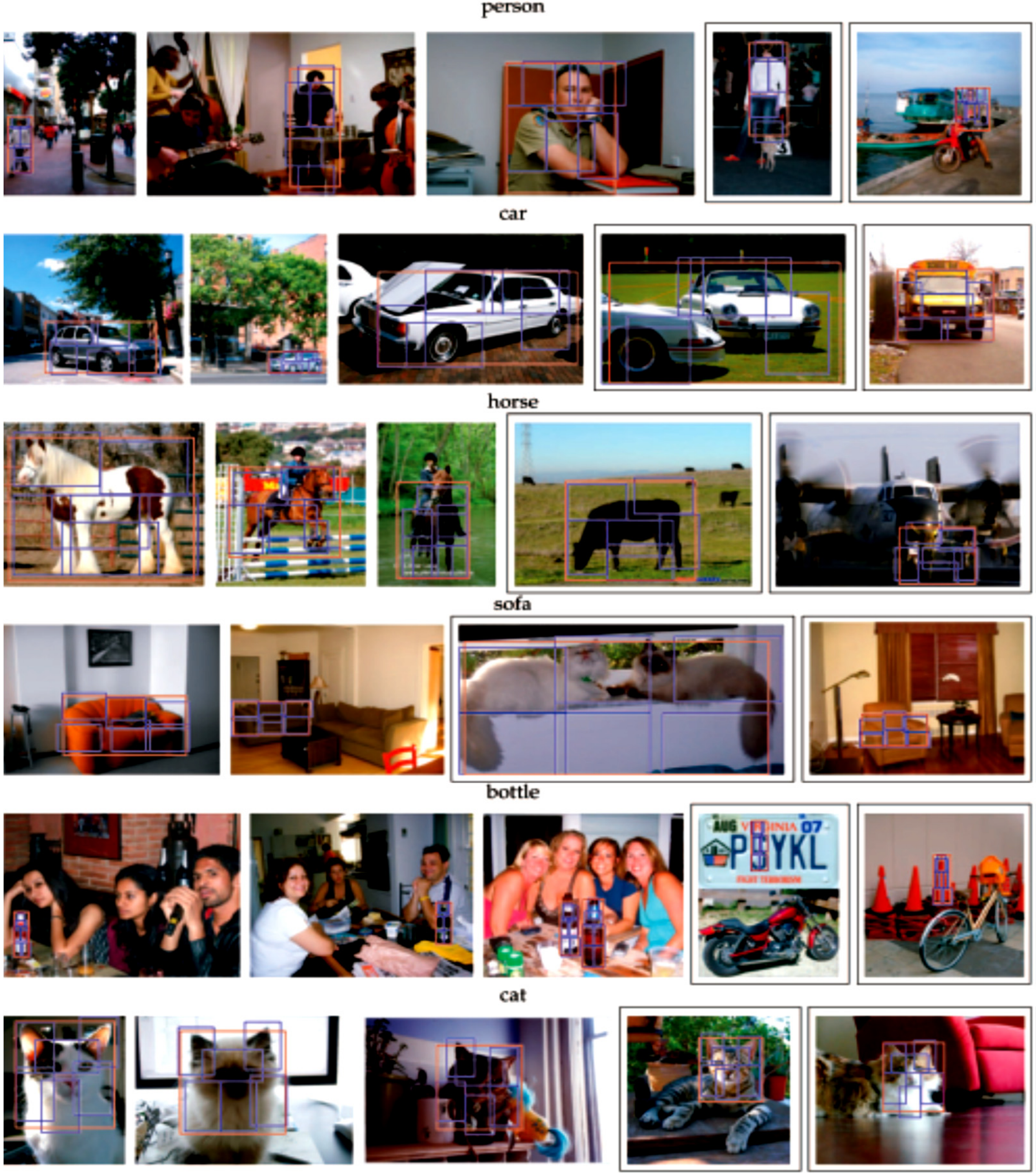
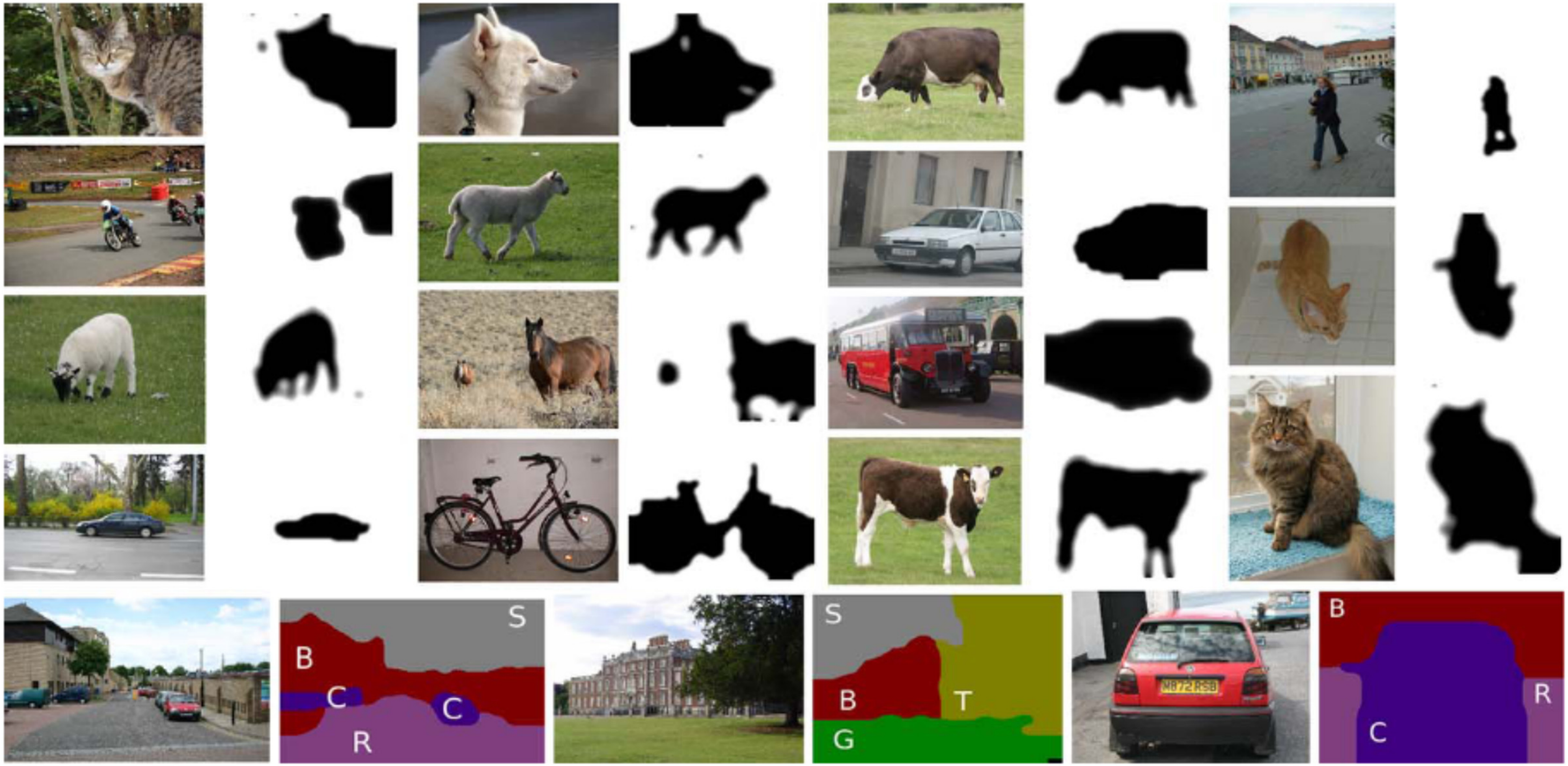


Fig. 22 (Color online) Example detections of our approach on difficult crowded scenes from the TUD Pedestrian test set (at the EER). Correc
detections are shown in *yellow*, false positives in *red*


person
car
horse
sofa
bottle
cat



IJCV – 2010; **Category Level Object Segmentation by Combining Bag-of-Words Models with Dirichlet Processes and Random Fields**
Diane Larlus · Jakob Verbeek · Frédéric Jurie

Think on your own now:

Image segmentation, object detection and recognition/identification are often intertwined topics. Does segmentation leads to recognition, or recognition leads to segmentation?

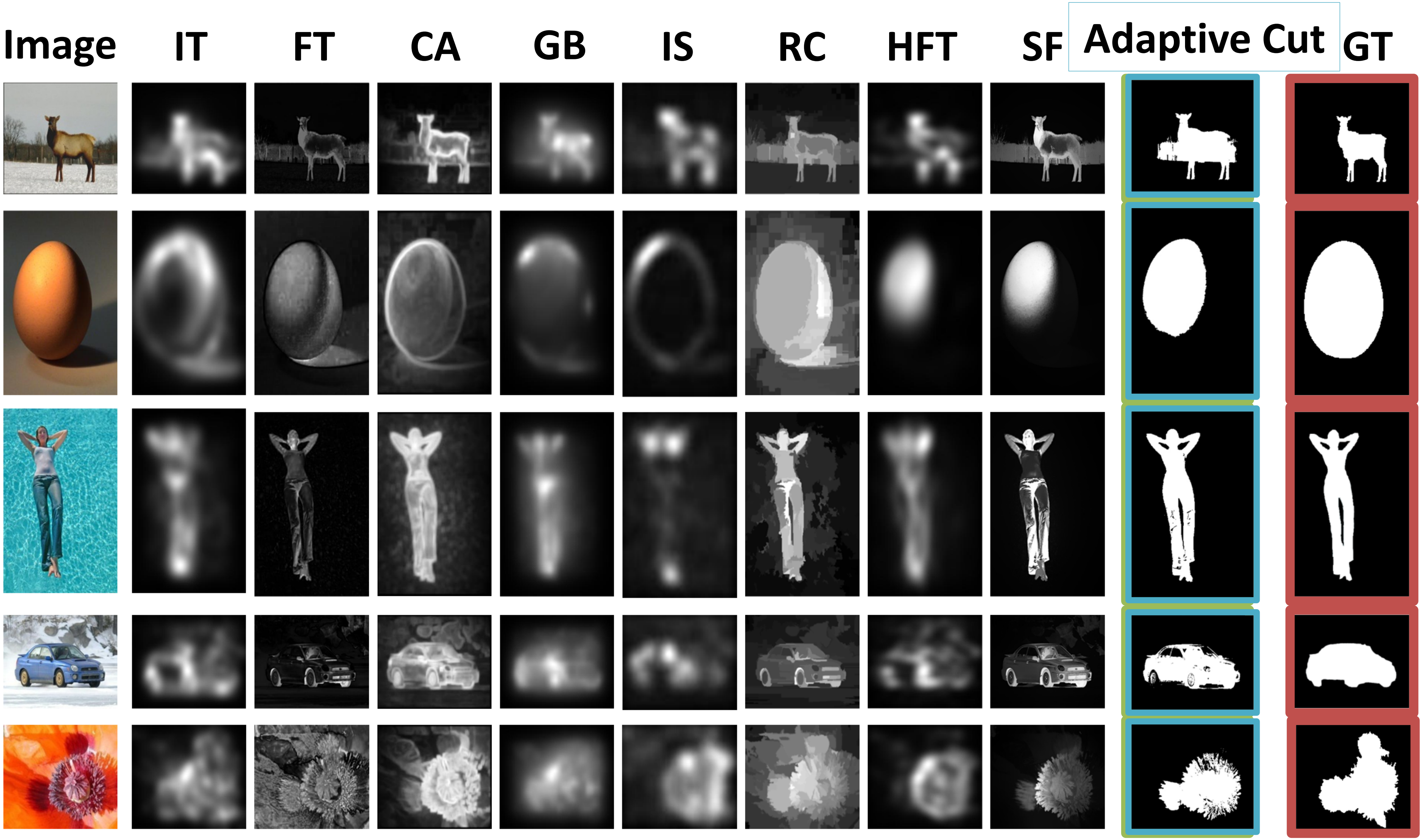
Can we do detection without having knowledge of the category (*e.g. what are U looking for ? – anything I can eat; I am hungry*) ?

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. Any situation where both processes are necessary ?

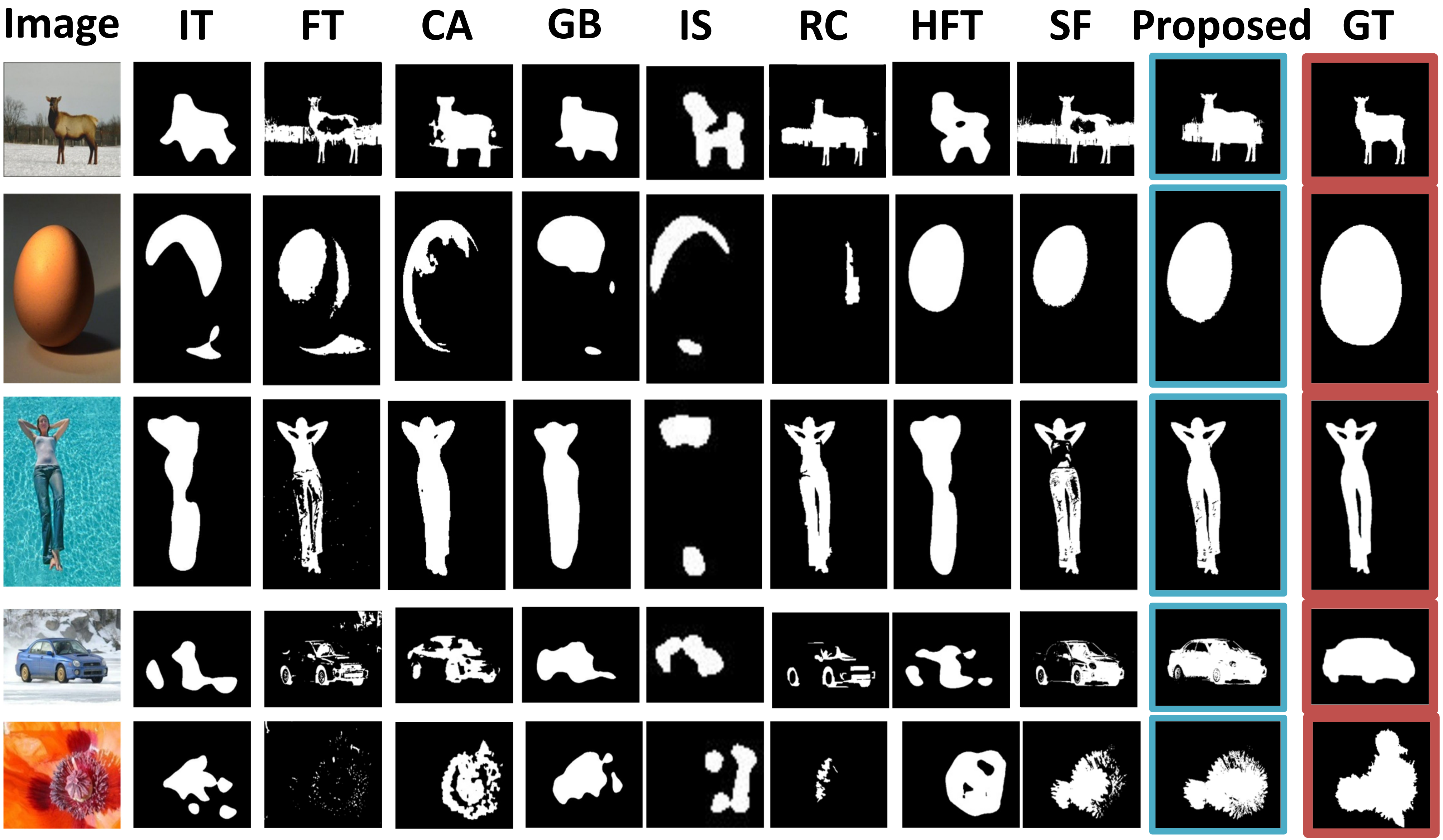
We will come back to OBJ. RECOGN. & Detection (supervised methods) in this course later, and then revisit this question.

Visual Results



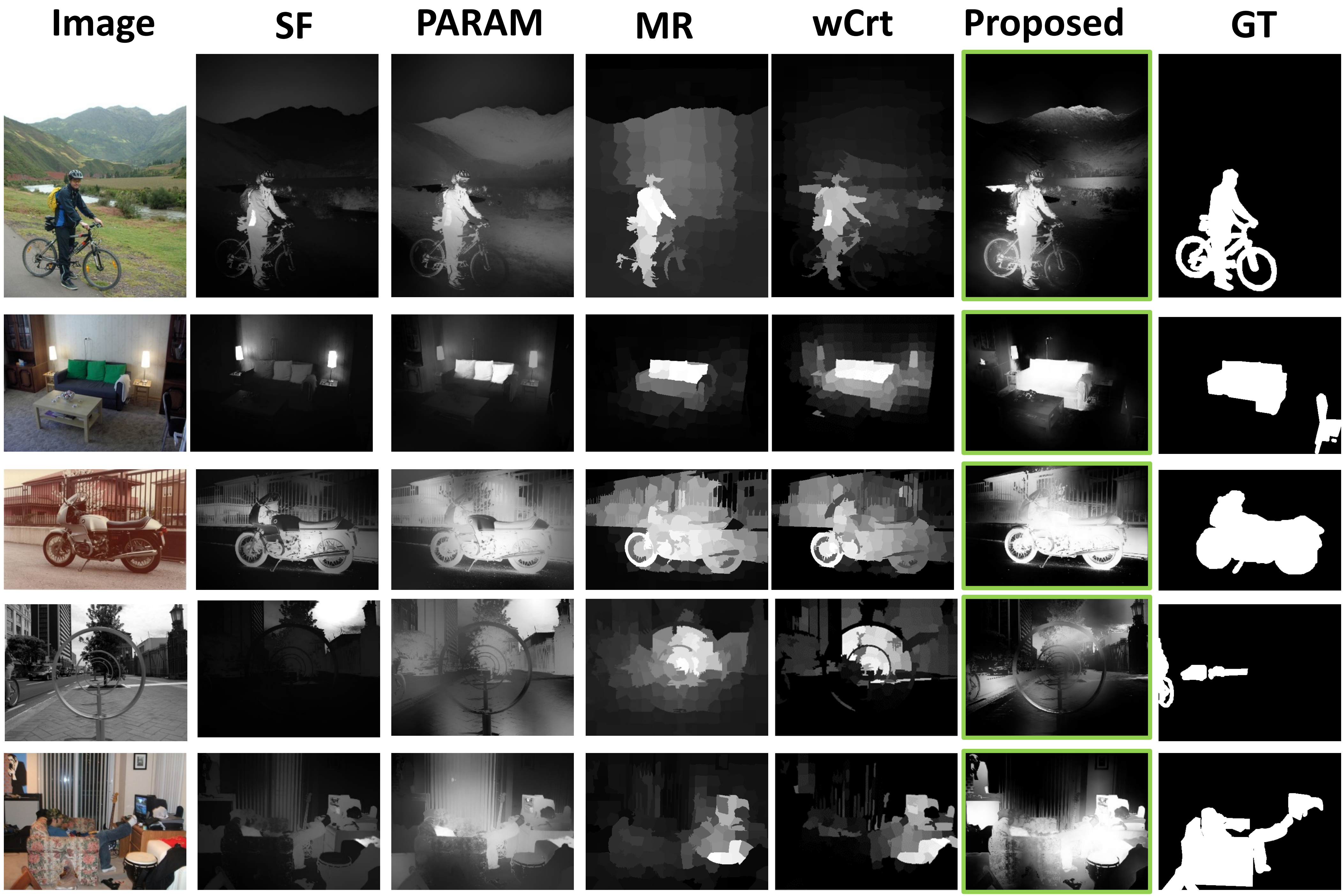
Images from MSRA B 5000 image Dataset

Adaptive Cut

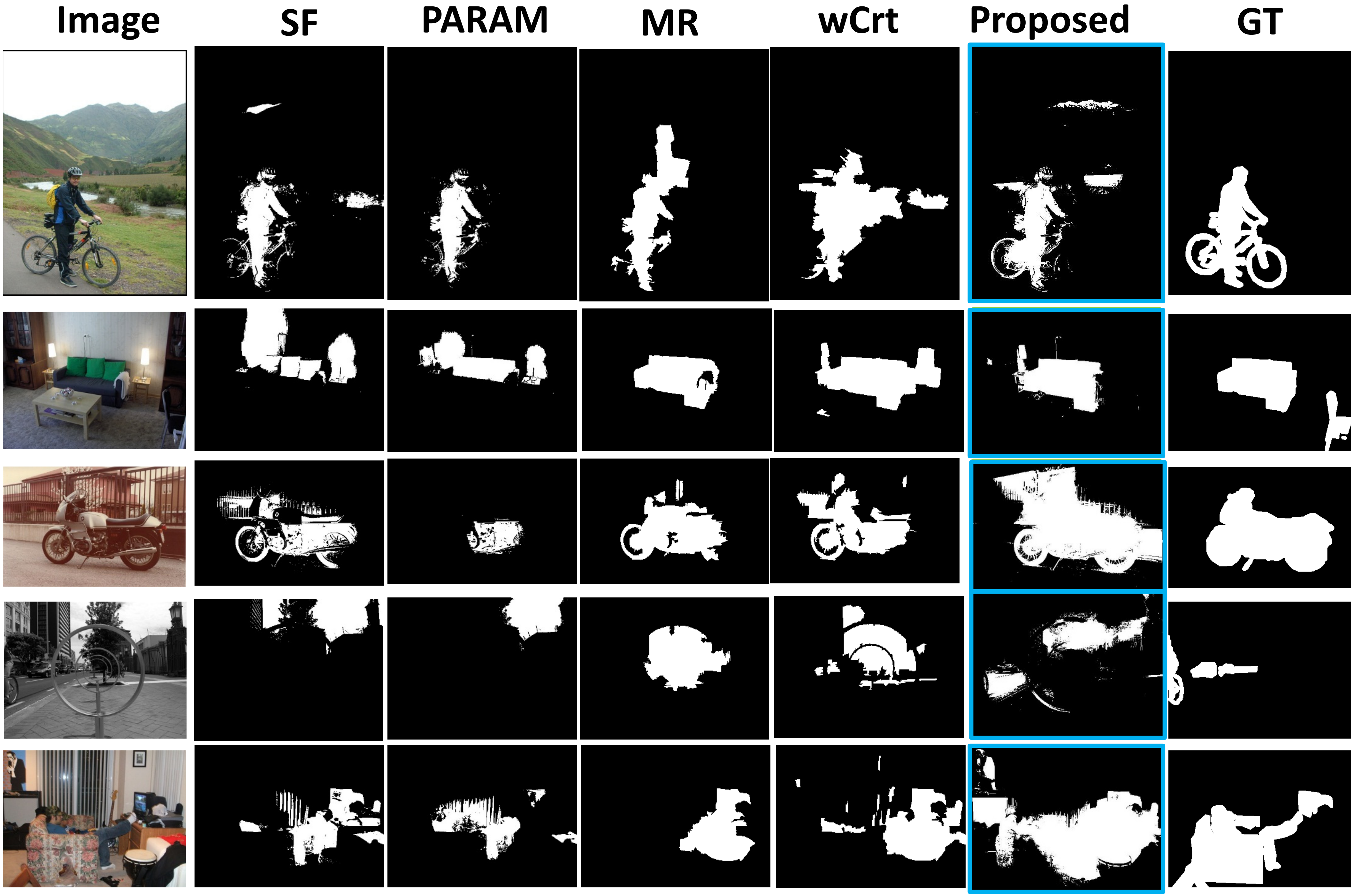


Images from MSRA B 5000 image Dataset

Visual Results on PASCAL



Visual Results on PASCAL



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.

- M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models; International Journal of Computer Vision, 1(4):321–331, January 1988.

- Boykov, Y. and Funka-Lea, G. (2006). Graph cuts and efficient N-D image segmentation. International Journal of Computer Vision, 70(2):109–131.

- 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.

- 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.

- Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Dietterich, T., Becker, S., Ghahramani, Z. (eds.); Advances in Neural Information Processing Systems (NIPS)14, pp. 849–856. MIT Press, Cambridge (2002).

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

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

-PF Felzenszwalb, DP Huttenlocher; Efficient graph-based image segmentation; International Journal of Computer Vision; Volume 59 Issue 2, September 2004 , ages 167 – 181.

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.

A. 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.

J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In Proc. IEEE Intl. Conf. on Computer Vision, 2005.

Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In Proc. ICCV, 2001.

Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In Proc. CVPR, 2001.

Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping; ACM Trans. Graph., 23(3):303–308, 2004.

C. Rother, V. Kolmogorov, and A. Blake; ”grabcut”: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3):309–314, 2004.

[1] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 23, pp. 1222-1239, 2001.

[2] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *Proceedings of the Ninth IEEE International Conference on Computer Vision-Volume 2,* 2003, pp. 26.

[3] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 26, pp. 147, 2004.

[4] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *International Journal of Computer Vision,* vol. 70, pp. 109-131, 2006.

[5] C. Rother, V. Kolmogorov and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH 2004 Papers,* 2004, pp. 314.

[6] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts-a review," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 29, pp. 1274-1279, 2007.

[7] Hiroshi Ishikawa, "Transformation of General Binary MRF Minimization to the First Order Case," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 Mar. 2010.


Bruegel, 1564