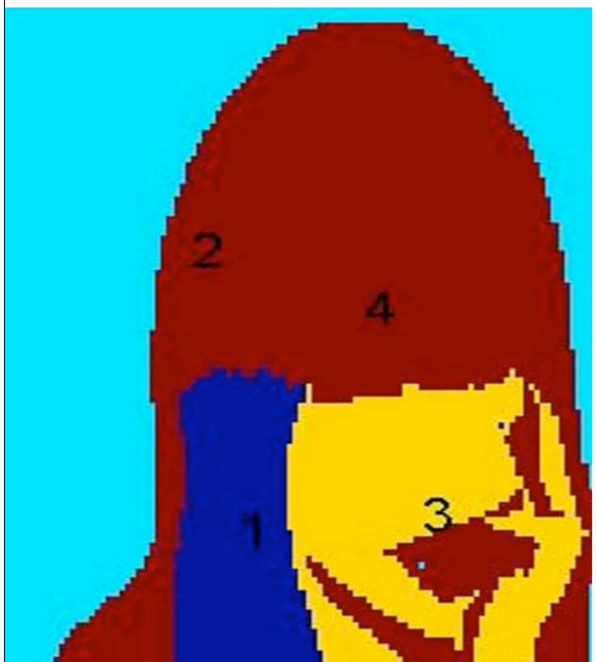
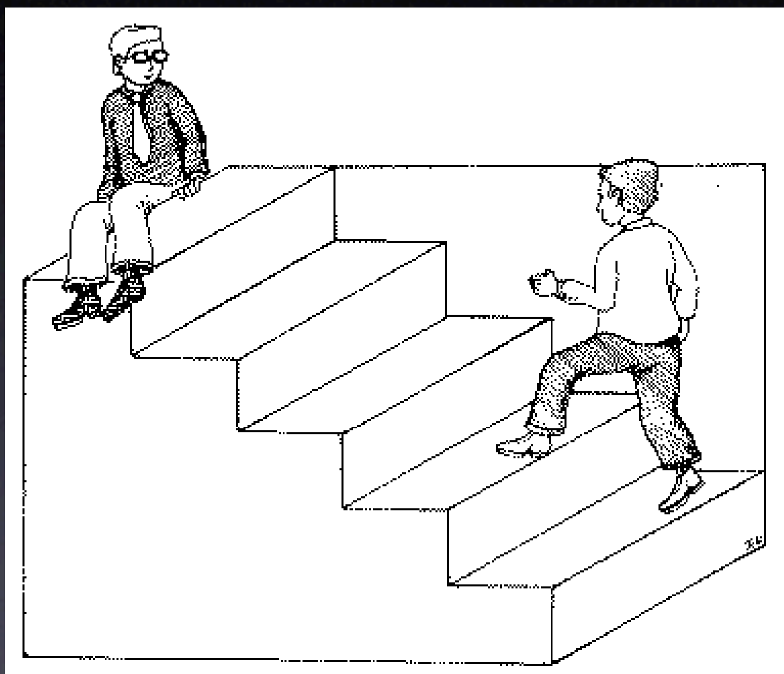
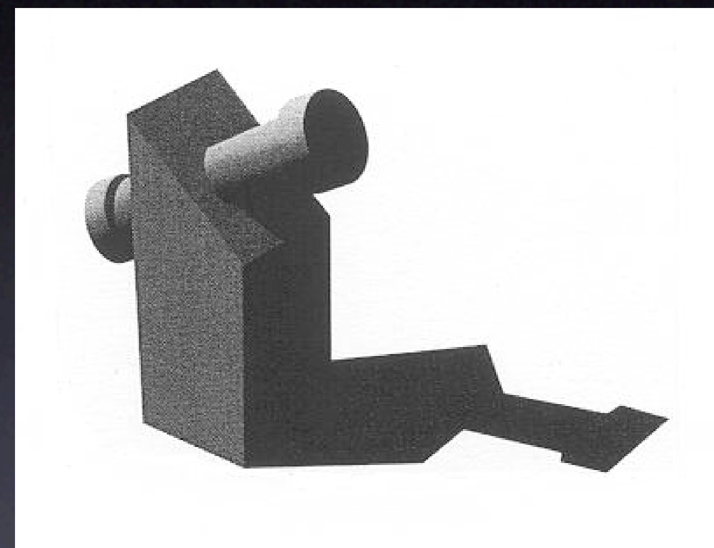


Graph Based Image Segmentation

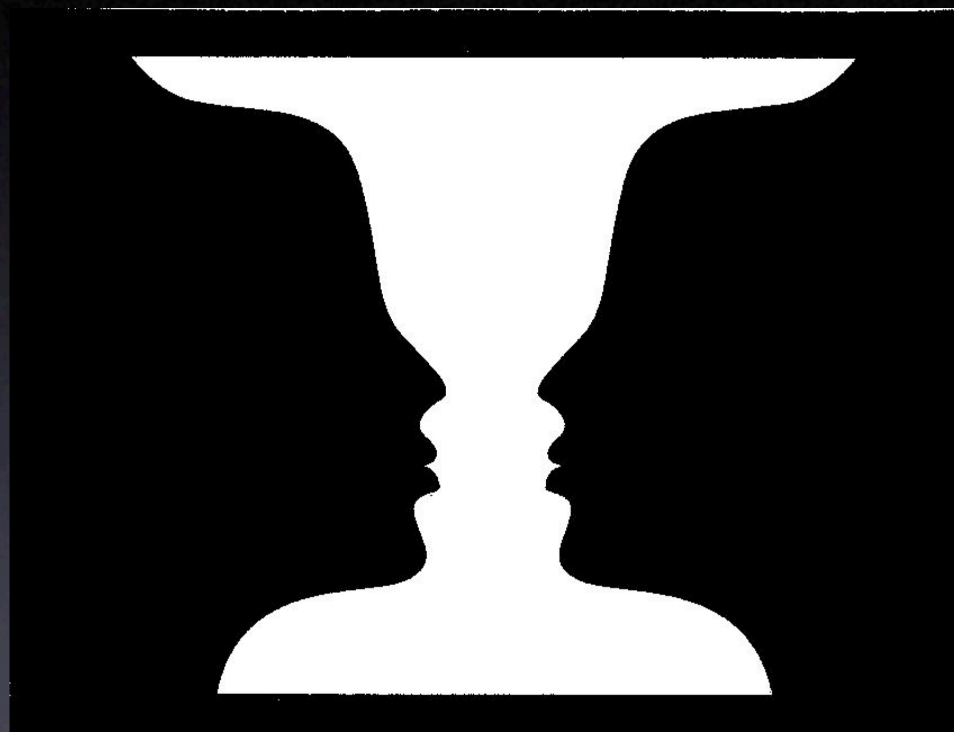
Jianbo Shi
University of Pennsylvania

A top-down process?



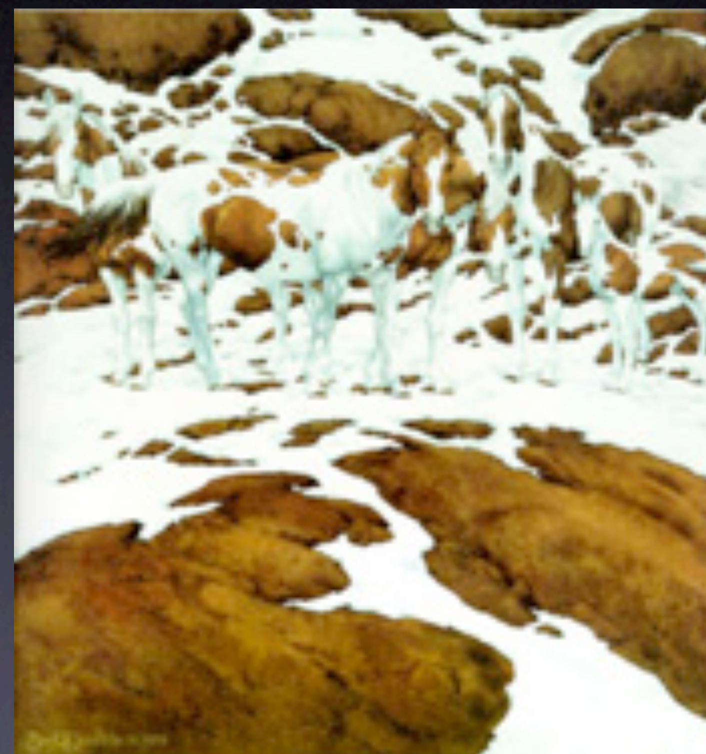


Or a bottom up process?



young woman, old woman

Both segmentation and recognition are context sensitive:
Need the whole to see its parts

segmentation ill defined?

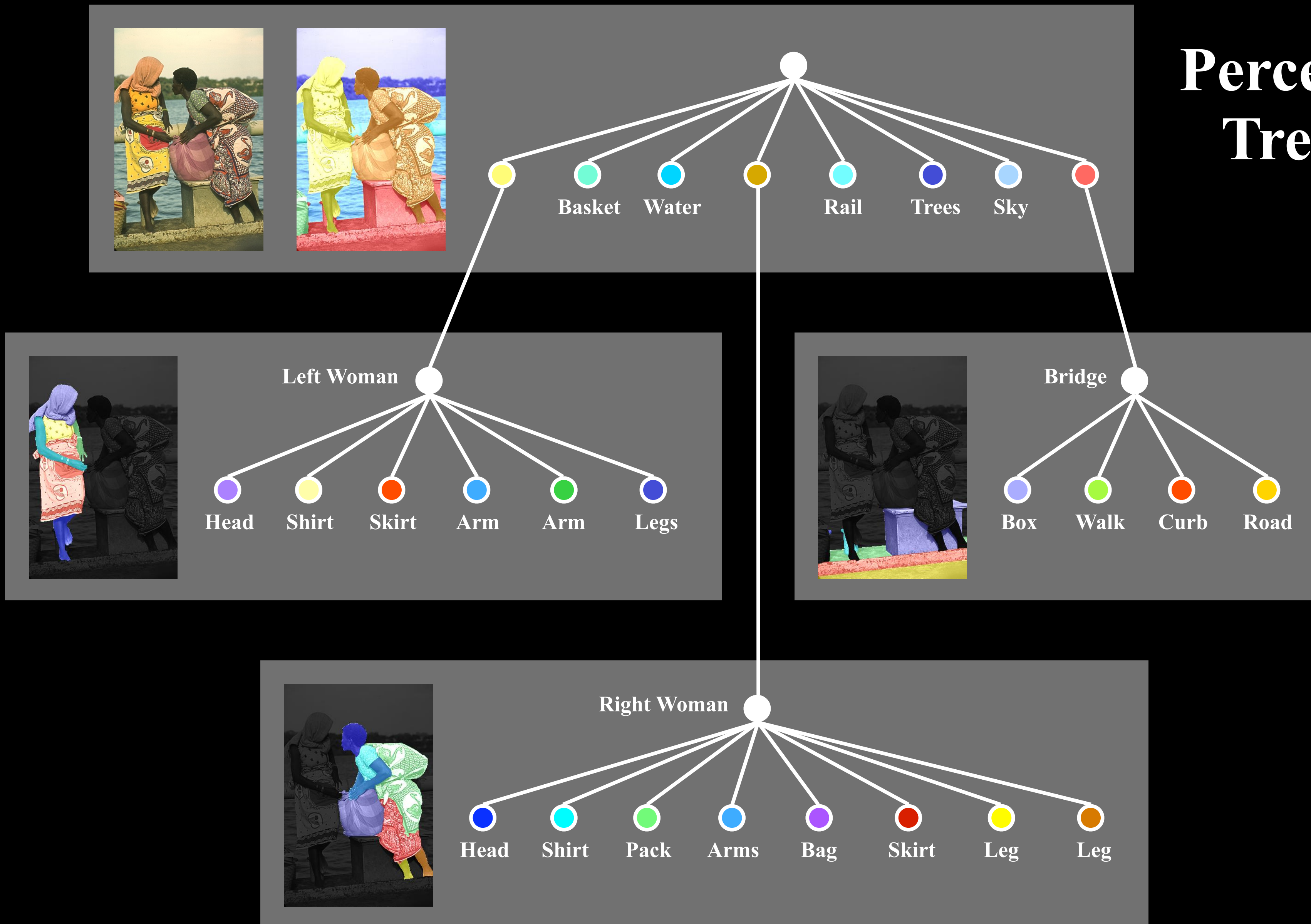
Berkeley Human Segmentation Dataset

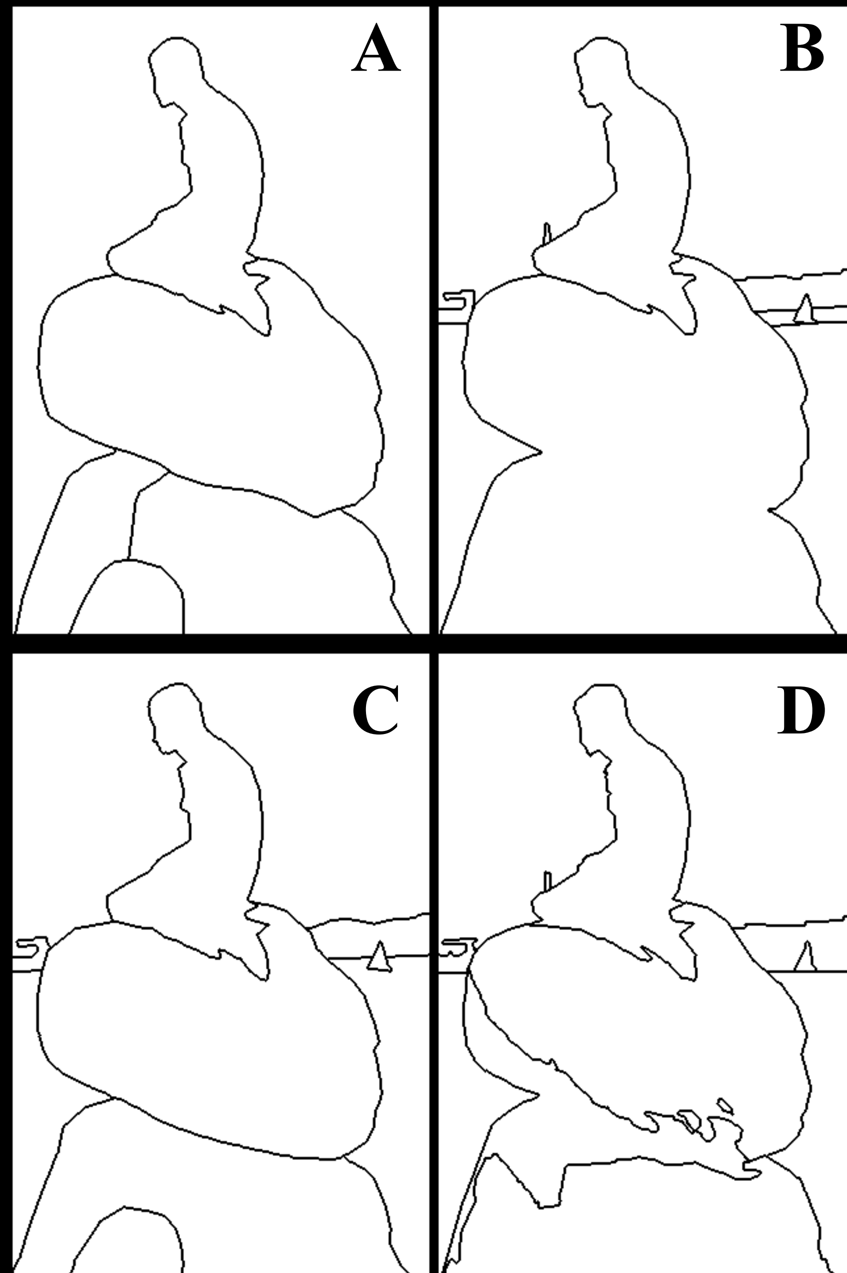
Dataset Summary

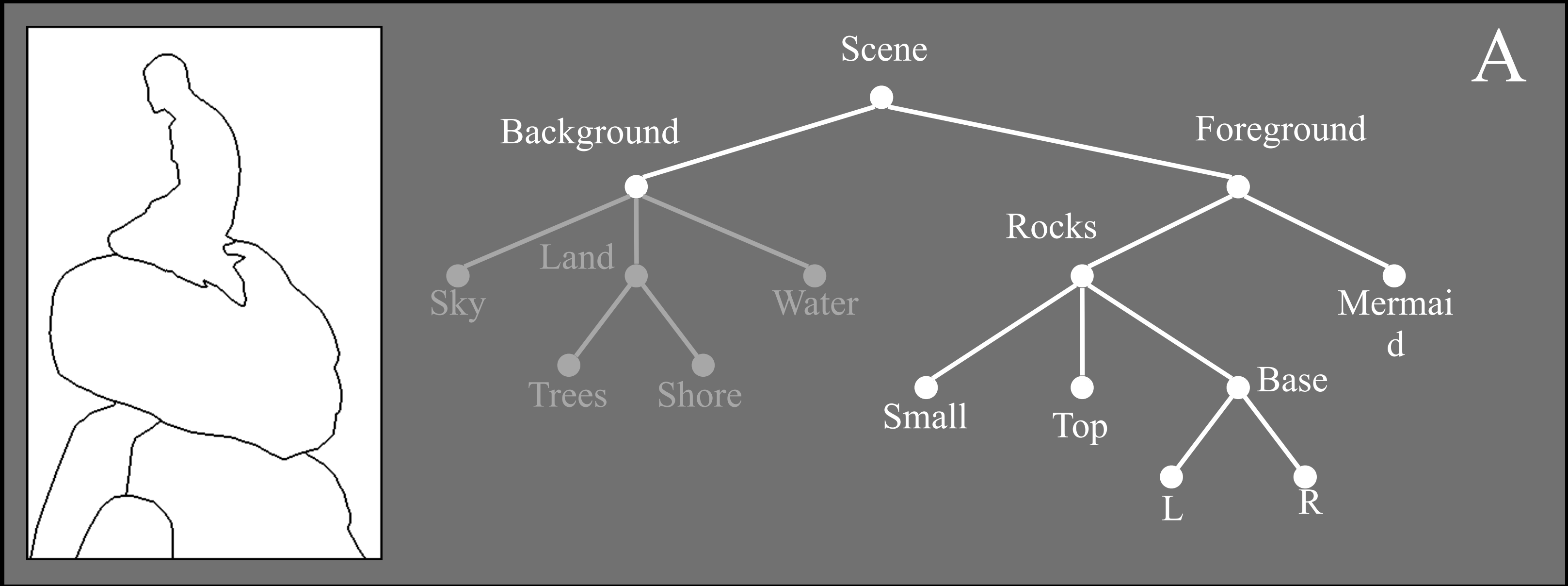
- 30 subjects, age 19-23
  - 17 men, 13 women
  - 9 with artistic training
- 8 months
- 1,458 person hours
- 1,020 Corel images
- 11,595 Segmentations
  - 5,555 color, 5,554 gray, 486 inverted/negated

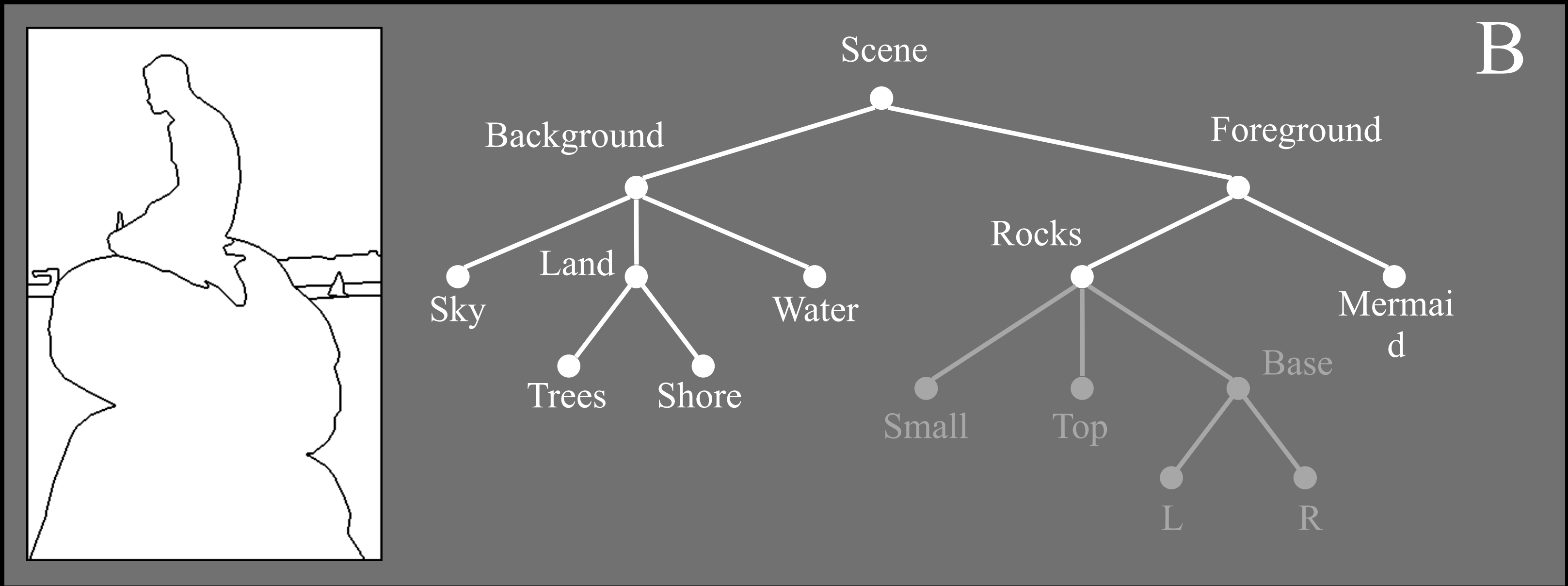
Do you even have one consistent segmentation?

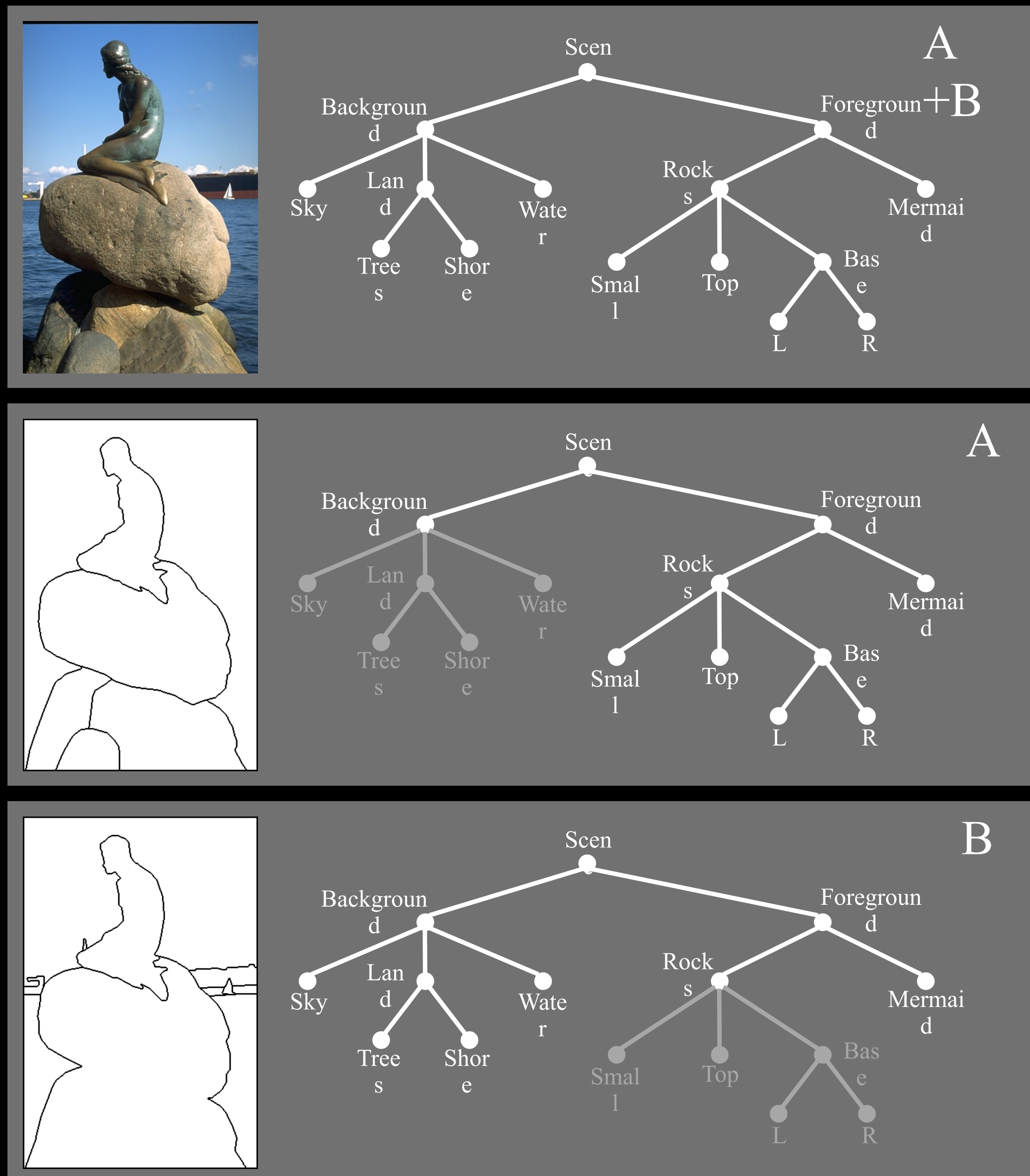
Percept Tree



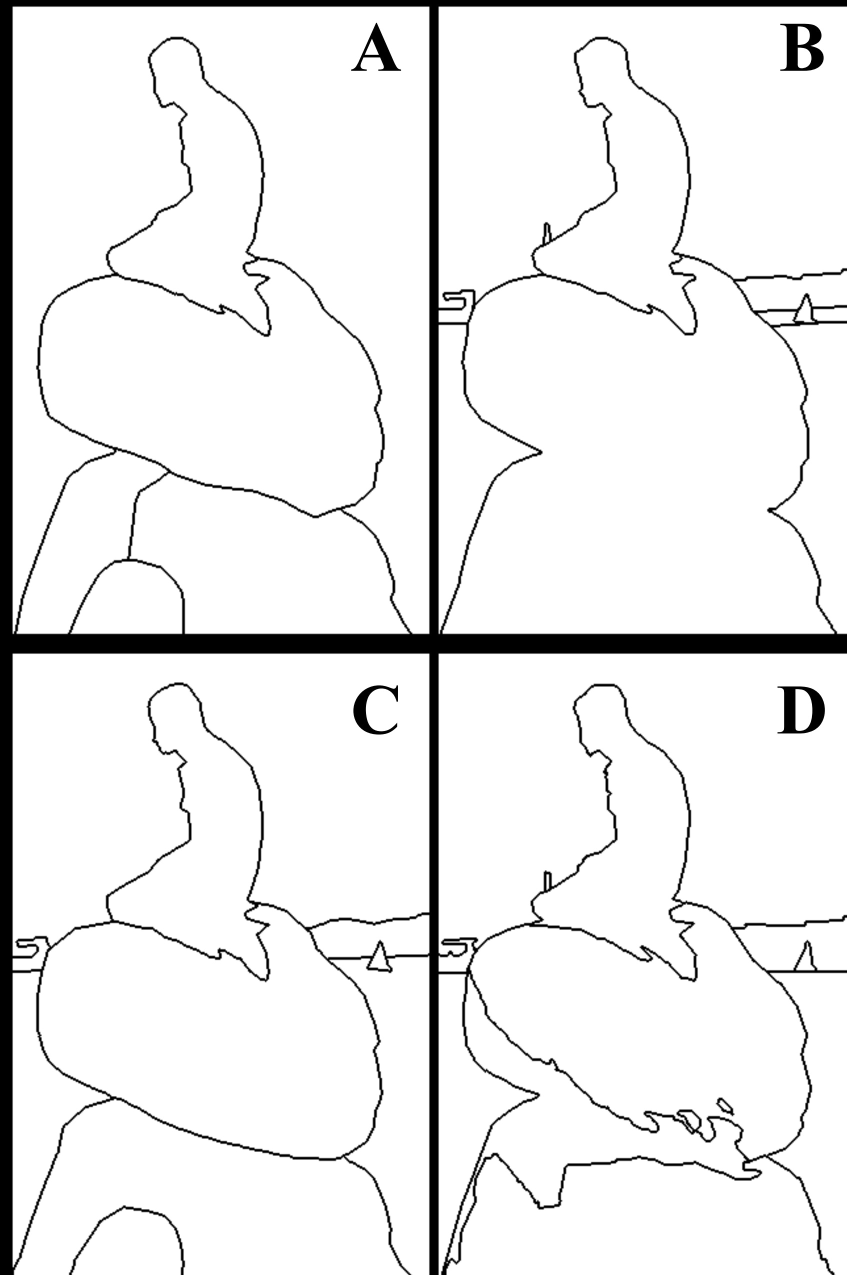

A
B
C
D


A
Scene
Background
Foreground
Sky
Land
Water
Trees
Shore
Rocks
Mermai
d
Small
Top
Base
L
R


B
Scene
Background
Foreground
Sky
Land
Water
Trees
Shore
Rocks
Mermai
d
Small
Top
Base
L
R



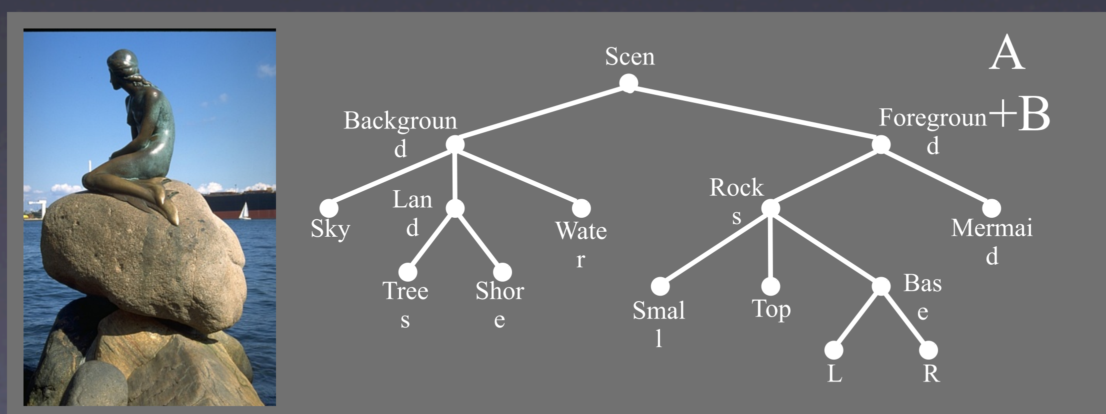
A
+B
Scen
Backgroun d
Foregroun d
Sky
Lan d
Wate r
Tree s
Shor e
Rock s
Mermai d
Smal l
Top
Bas e
L
R
A
Scen
Backgroun d
Foregroun d
Sky
Lan d
Wate r
Tree s
Shor e
Rock s
Mermai d
Smal l
Top
Bas e
L
R
B
Scen
Backgroun d
Foregroun d
Sky
Lan d
Wate r
Tree s
Shor e
Rock s
Mermai d
Smal l
Top
Bas e
L
R



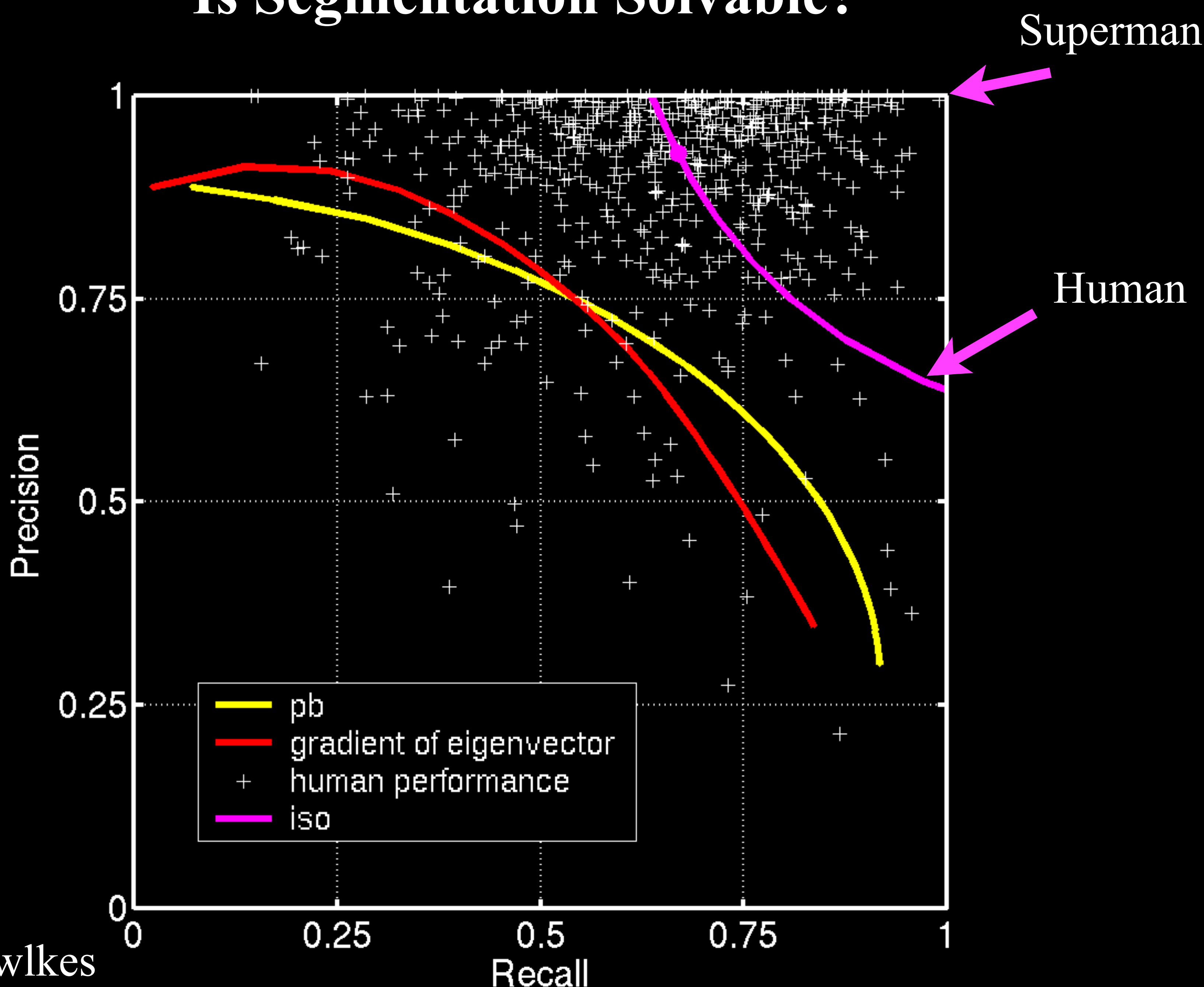
A
B
C
D

Insight 1: Segmentation/Clustering is always hierarchical

The difficult part is getting the top of the tree correct

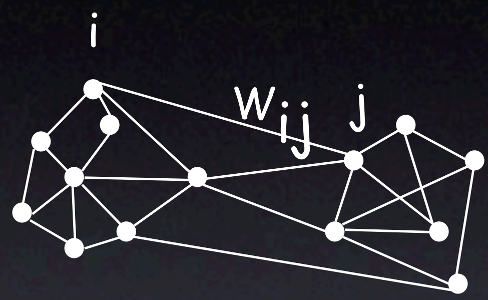


Is Segmentation Solvable?



C. Fowlkes

Graph Based Image Segmentation



$$\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$$

V: graph nodes

E: edges connection nodes

↔

Image = { pixels }

Pixel similarity

Segmentation = Graph partition

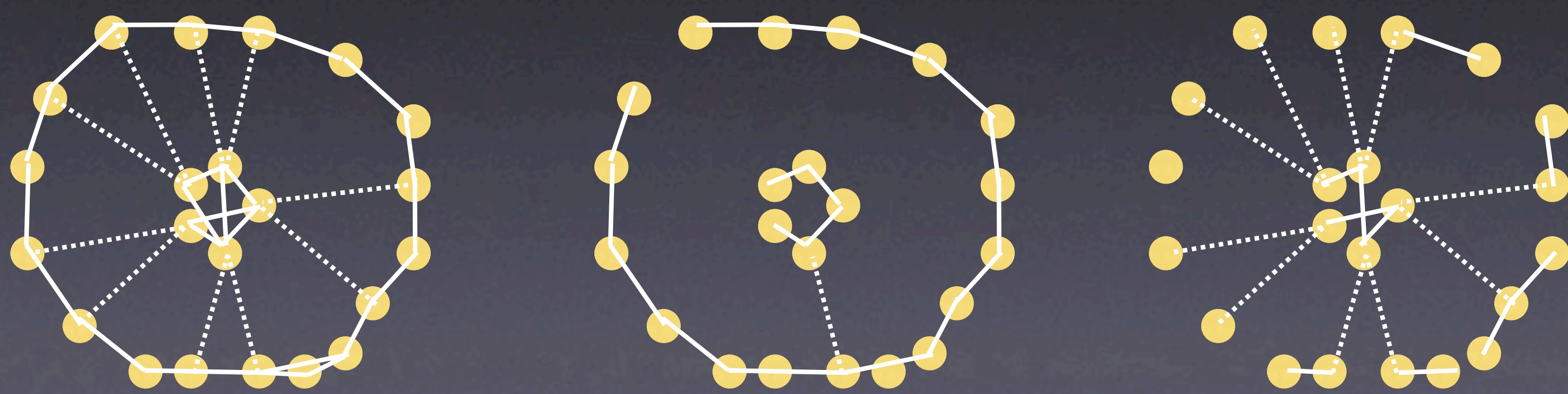
Right partition cost function?

Efficient optimization algorithm?

For simple cases, can try this:

Minimal/Maximal Spanning Tree

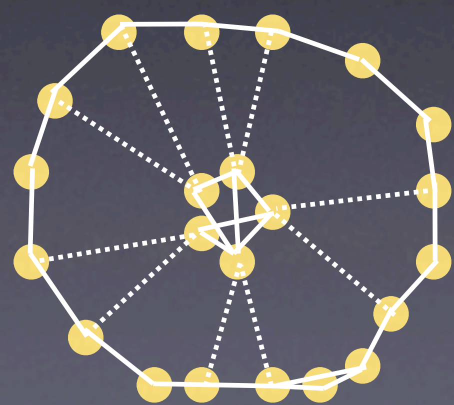
Tree is a graph G without cycle



Graph   Maximal   Minimal

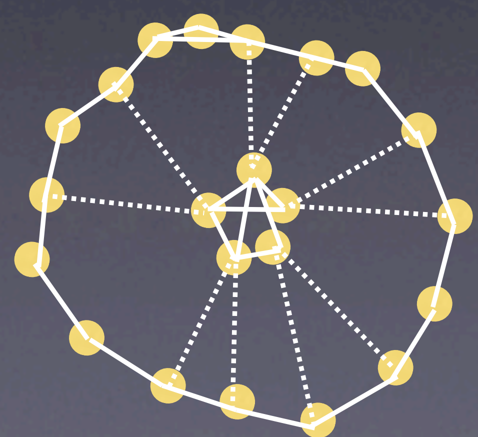
building a MST

Let X be any subset of the vertices of G, and let edge e be the smallest edge connecting X to G-X. Then e is part of the minimum spanning tree



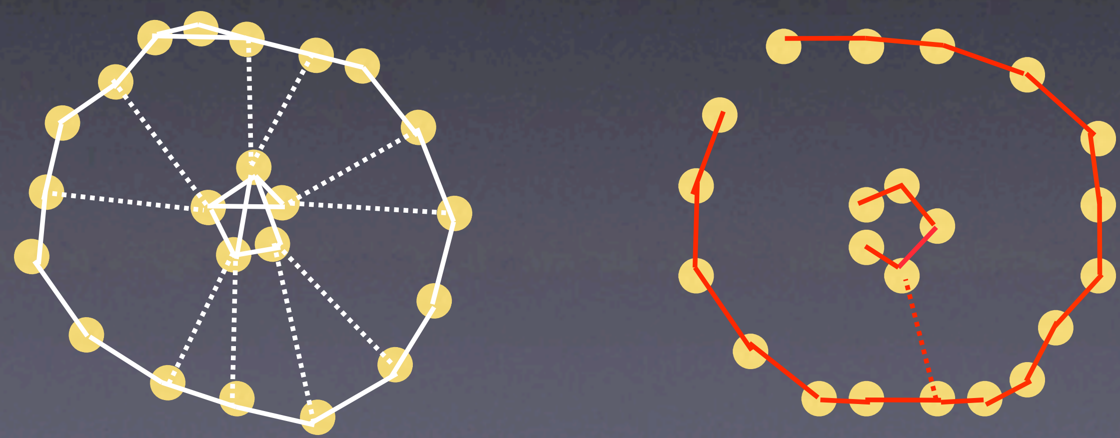
Prim’s algorithm

```
let T be a single vertex x
while (T has fewer than n vertices)
{
    find the smallest edge connecting T to G-T
    add it to T
}
```



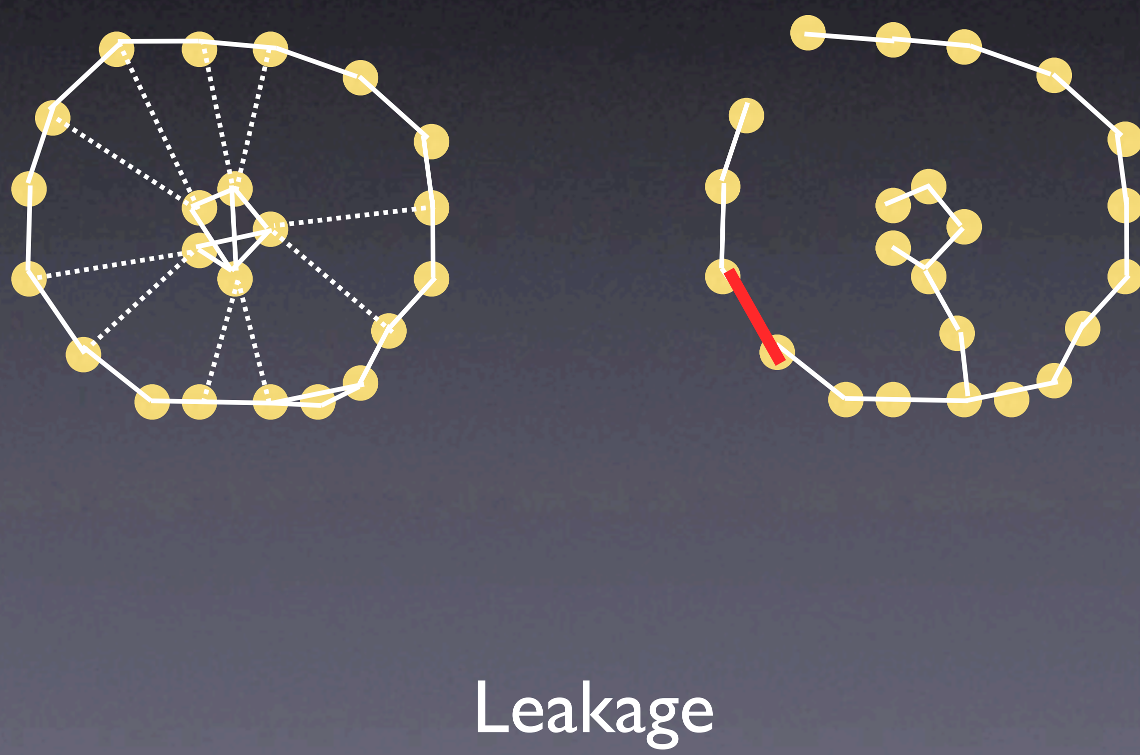
Kruskal's algorithm

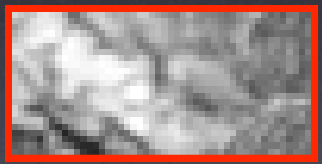
- sort the edges of G in increasing order by length
- for each edge e in sorted order
  if the endpoints of e are disconnected in S
  add e to S

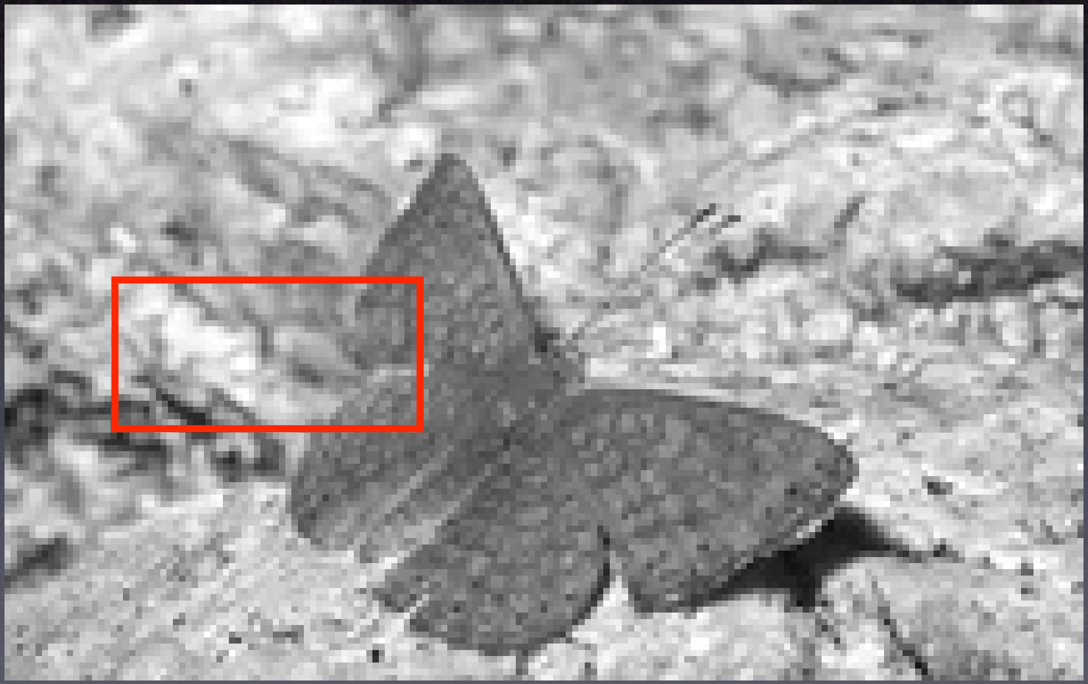


Randomized version can compute Typical cuts

Leakage problem in MST







Graph Segmentation: image cues

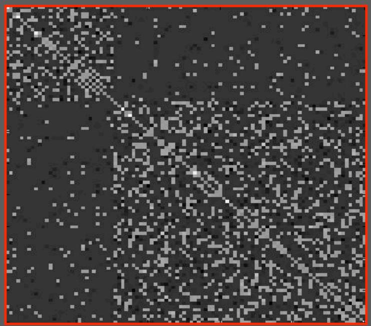
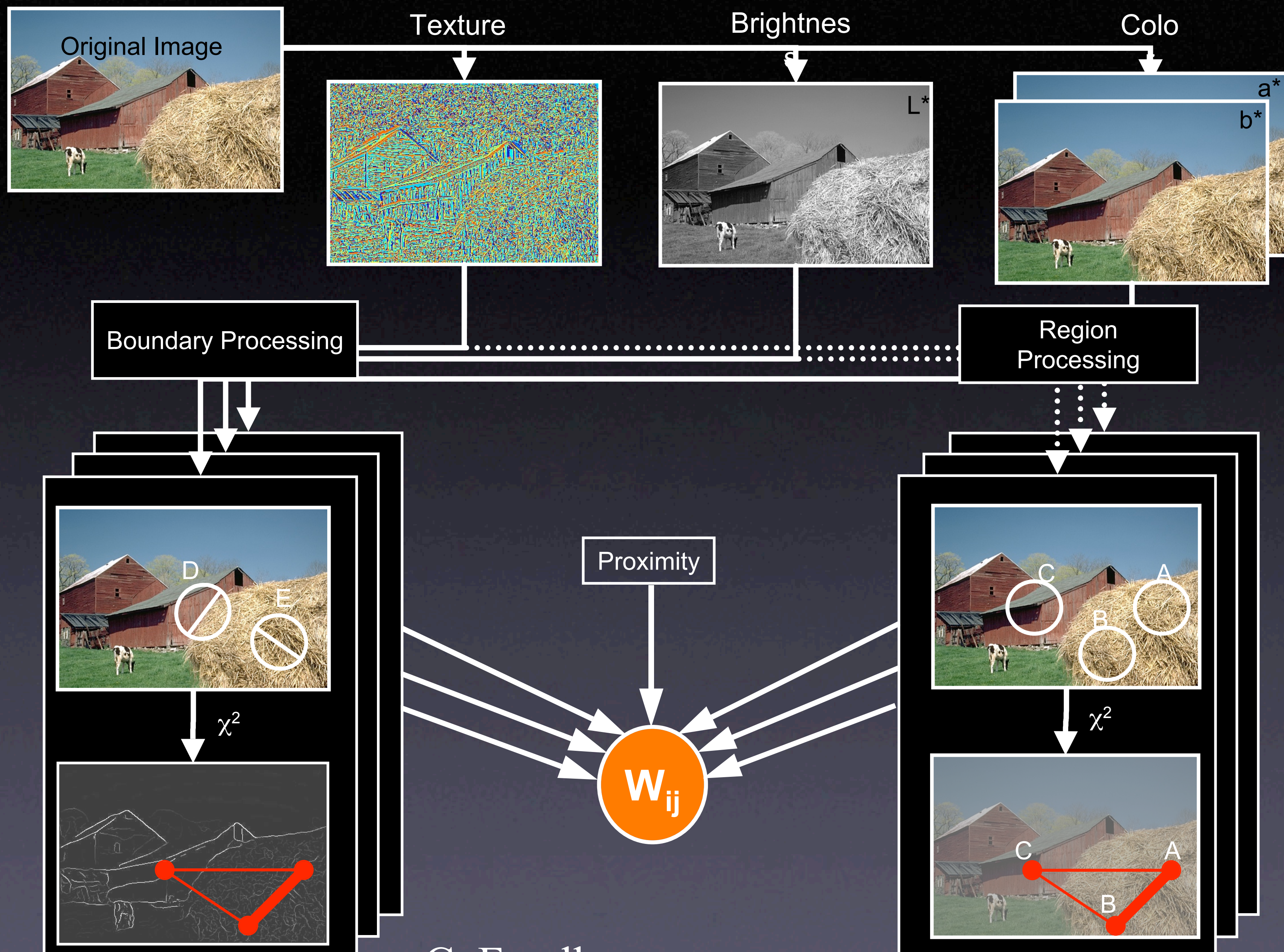


Image I → Graph Affinities

$W = W(I, \Theta)$

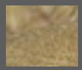
Intensity
Color
Edges
Texture
...

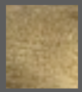


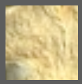
Original Image
Texture
Brightnes
s
Colo
L*
a*
b*
Boundary Processing
Region
Processing
D
E
χ2
Proximity
Wij
C
A
B
χ2
C
A
B

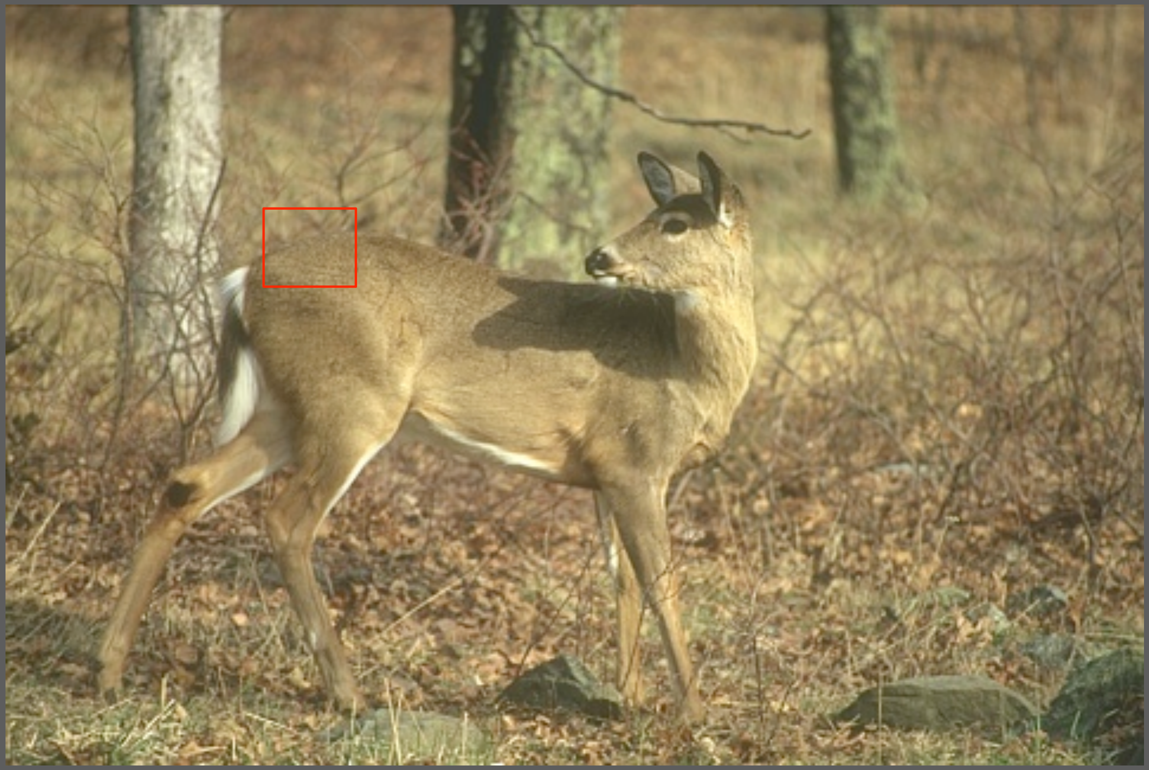
C. Fowlkes

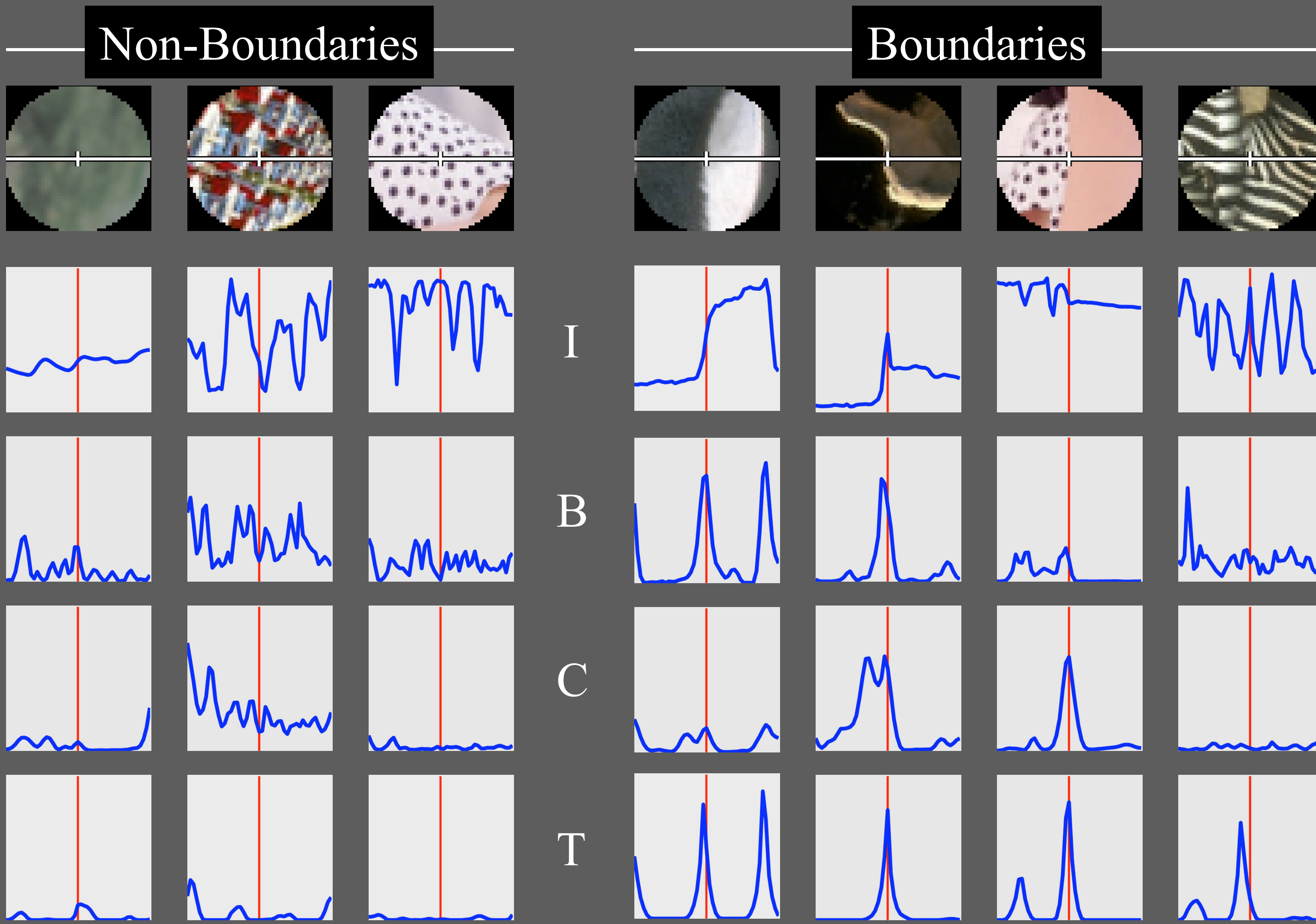
What is so hard about segmentation?





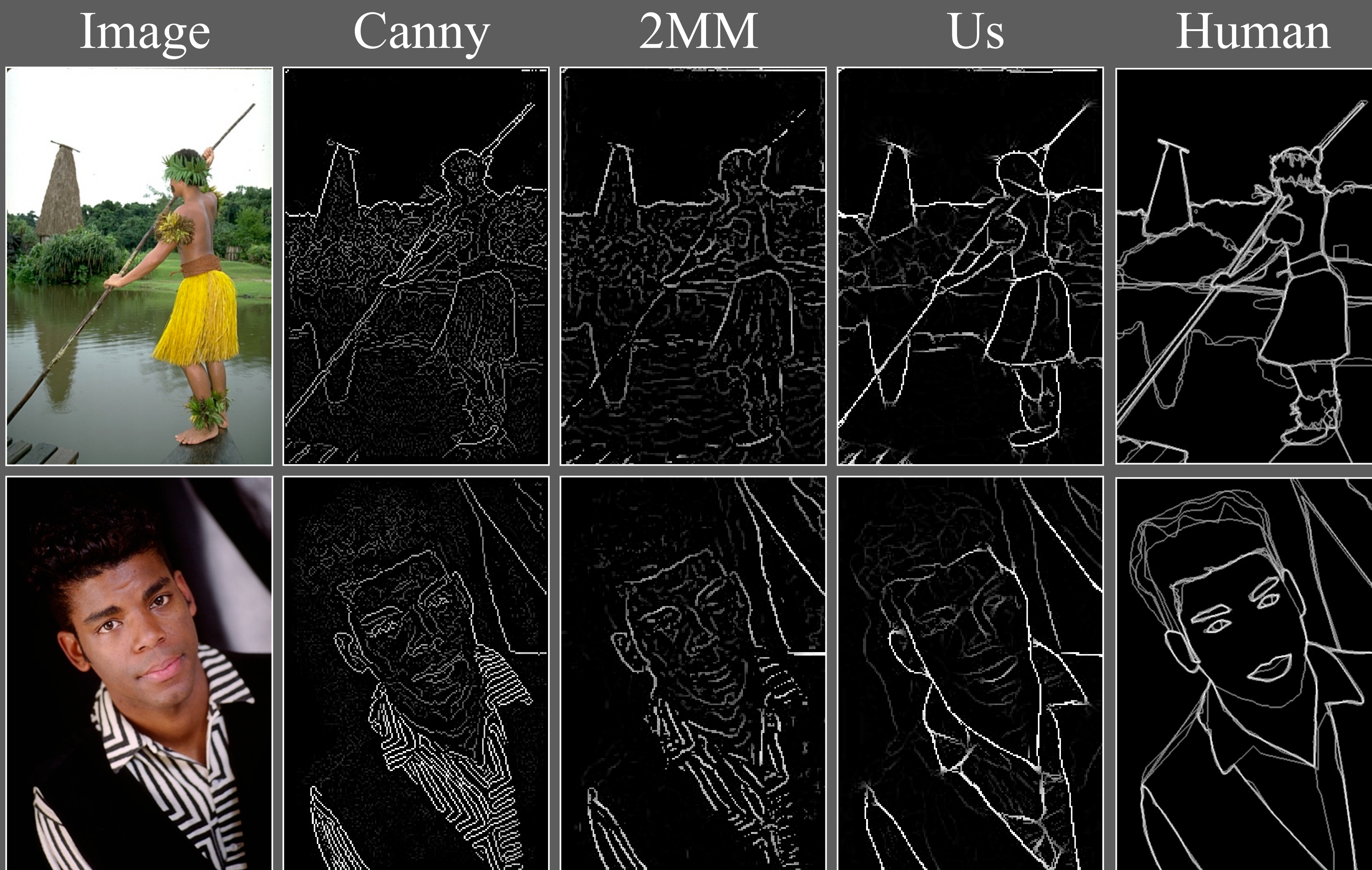






C. Fowlkes

P_b Images I

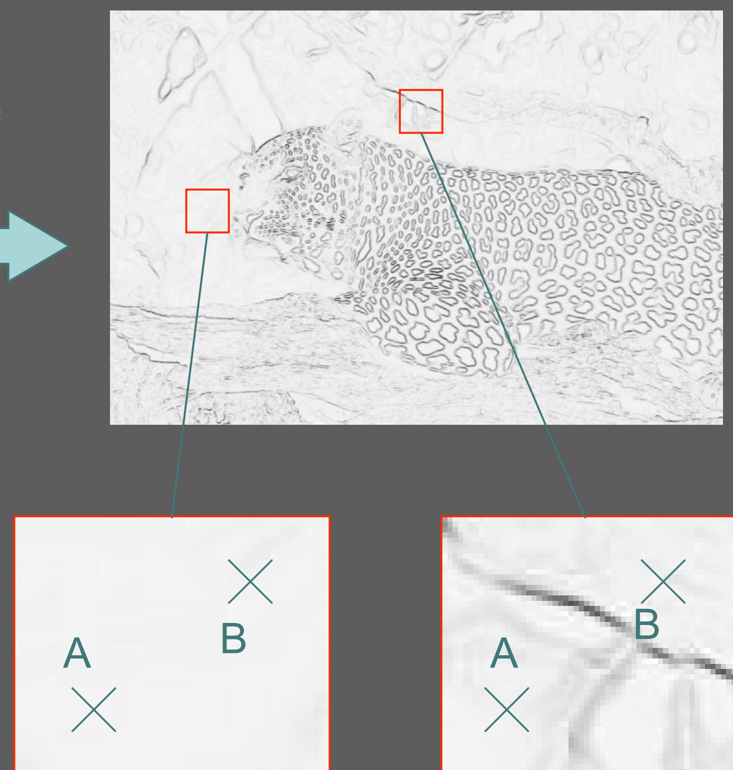


C. Fowlkes

Image I → Graph Affinities

$W=W(I,\Theta)$

Intensity
Color
Edges
Texture



Intervening Contour

…turning a boundary map into Wij

1 - maximum P_b along the line connecting i and j

Graph Segmentation

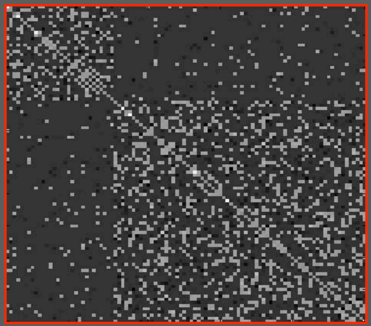


Image I → Graph Affinities

W=W(I,Θ)

Intensity
Color
Edges
Texture
…

Graph Segmentation: How to break the graph

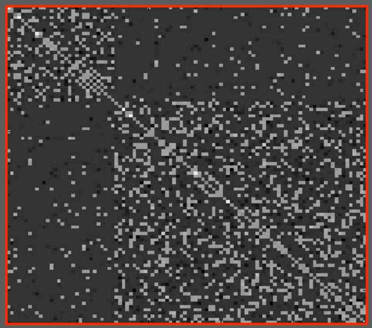
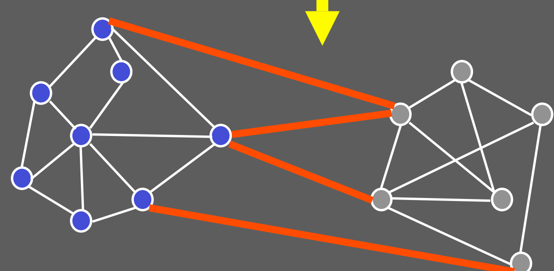


Image I

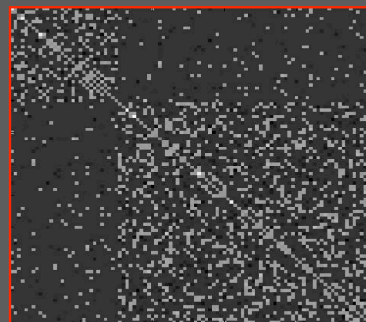
Intensity
Color
Edges
Texture
…

Graph Affinities
W=W(I,Θ)



Graph to encode Gestalt:
Getting the big picture
of scene

Spectral Graph Segmentation



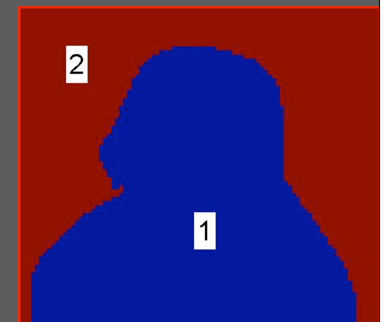


Image I → Graph Affinities W=W(I,Θ) → Eigenvector X(W) → Discretisation

$$NCut(A,B) = \frac{cut(A,B)}{Vol\ A'\ Vol\ B}$$

$$WX = l\ DX$$

$$X_A(i) = \begin{cases} 1 \text{ if } i \in A \\ 0 \text{ if } i \notin A \end{cases}$$

Graph Segmentation: How to break the graph

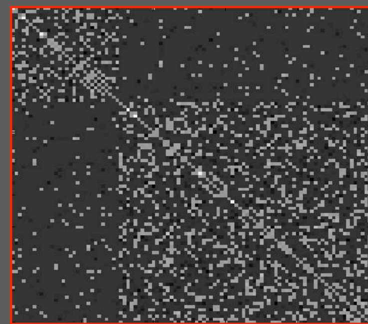
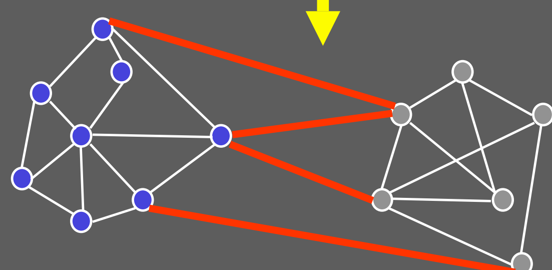


Image I

Intensity
Color
Edges
Texture
...

Graph Affinities
$W=W(I,\Theta)$



Graph to encode Gestalt:
Getting the big picture of scene

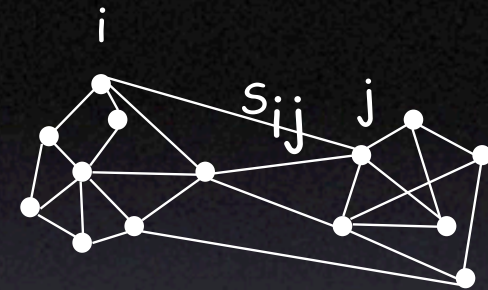
Graph Terminology

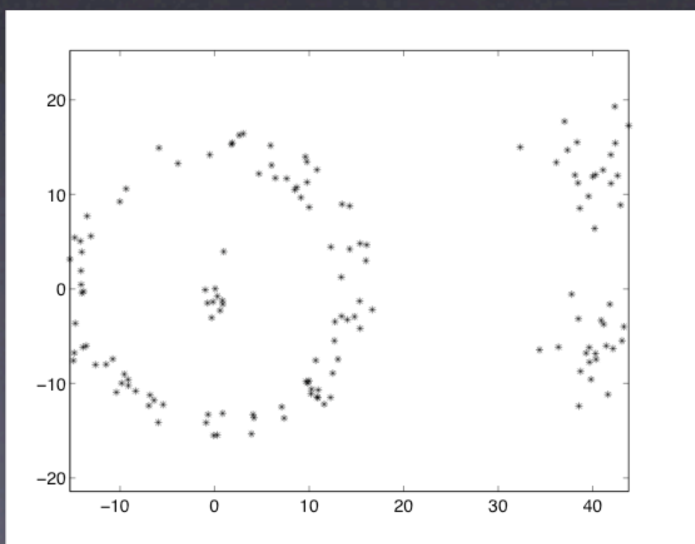
adjacency matrix,
degree,
volume,
graph cuts

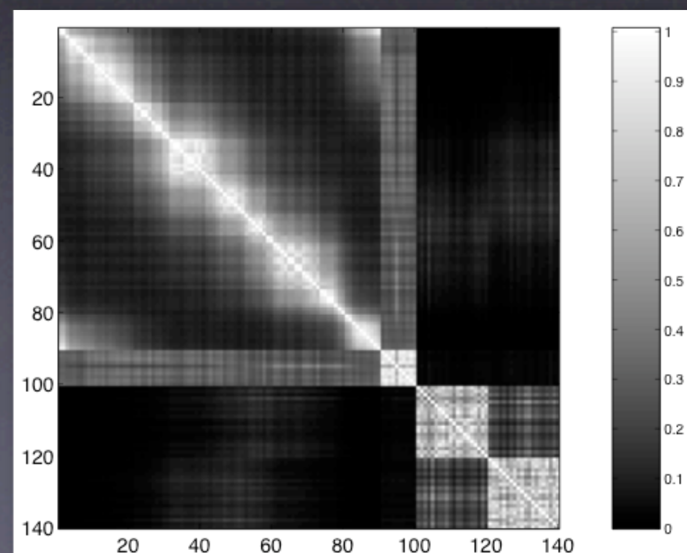
Graph Terminology

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

is generalized adjacency matrix

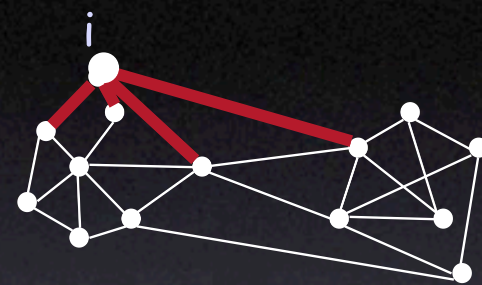


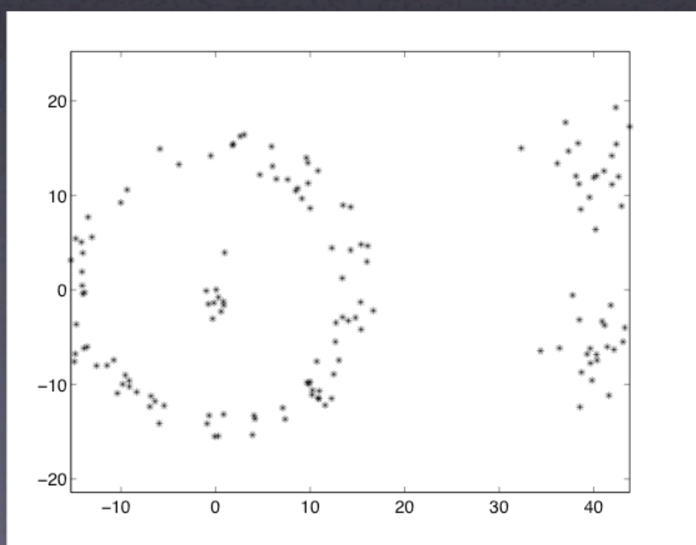


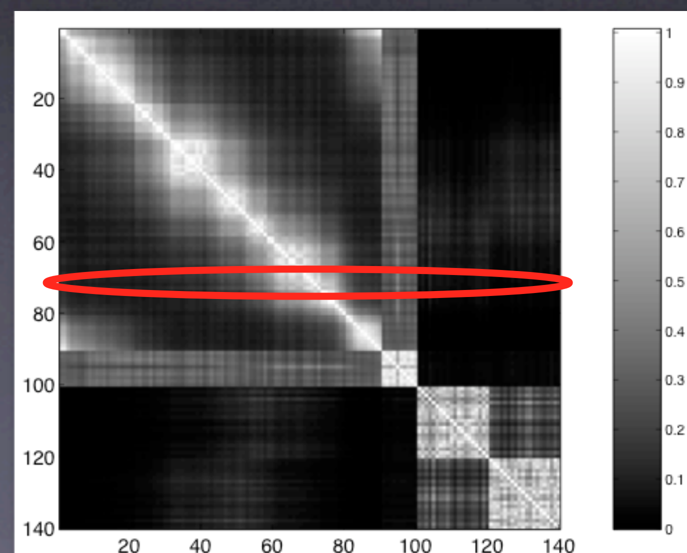


Graph Terminology

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



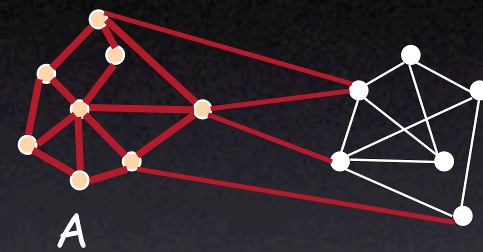


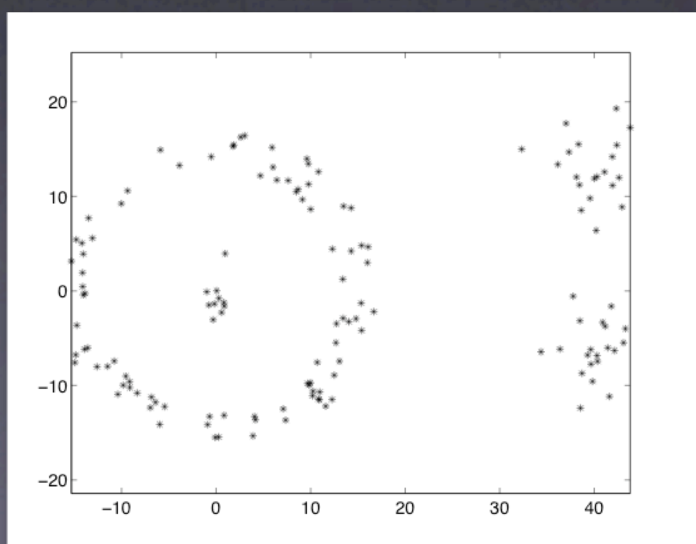


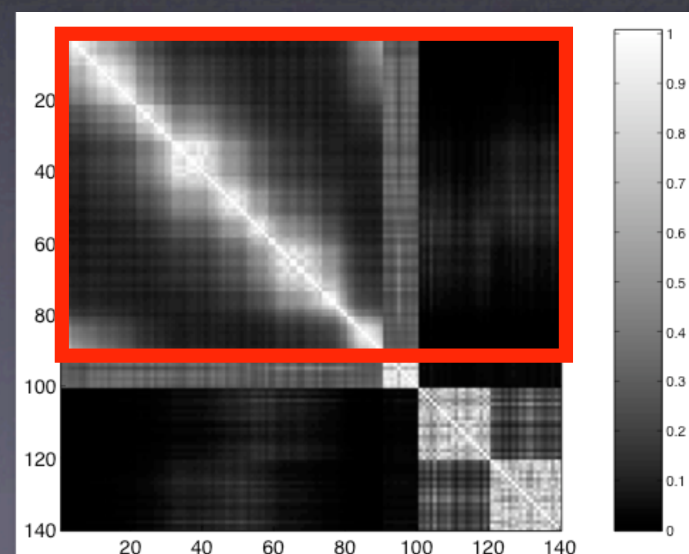
Graph Terminology

Volume of set:

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

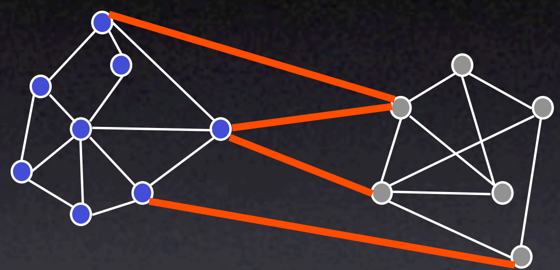


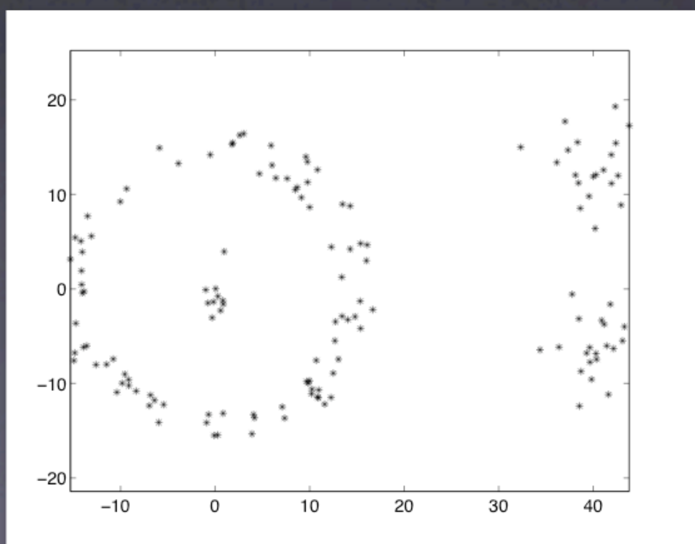


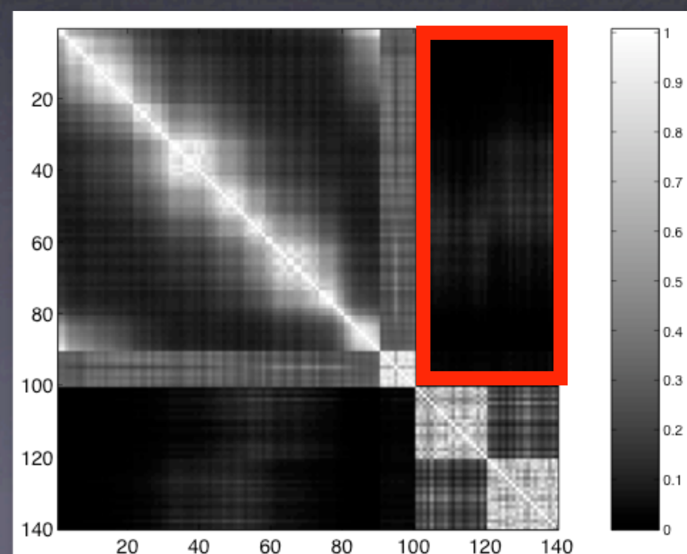


Cuts in a graph

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

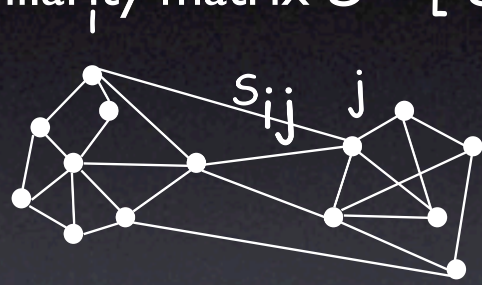




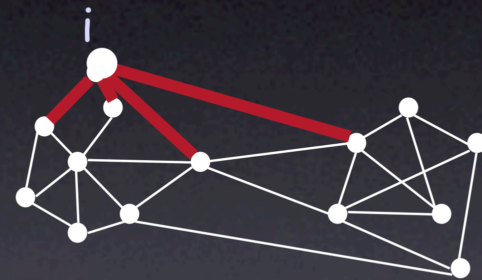


Graph Terminology

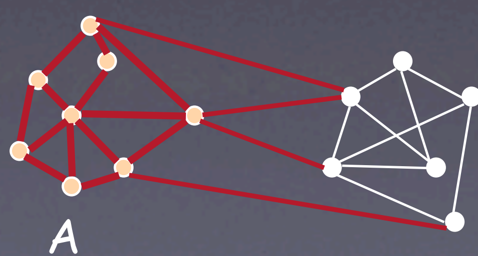
Similarity matrix $S = [\, S_{ij} \,]$



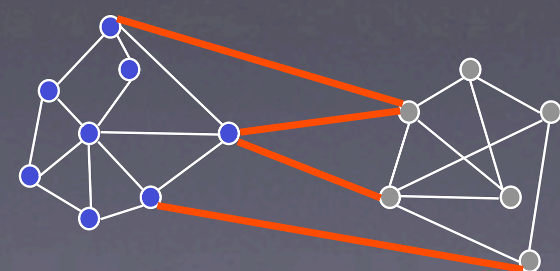
Degree of node: $d_i = \sum_j S_{ij}$



Volume of set:



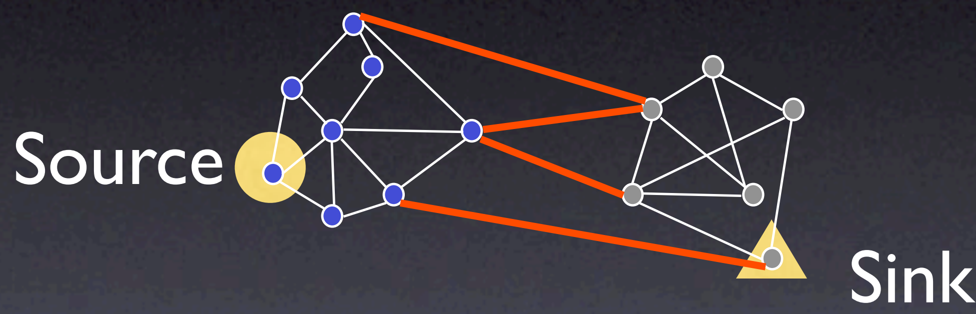
Graph Cuts



Useful Graph Algorithms

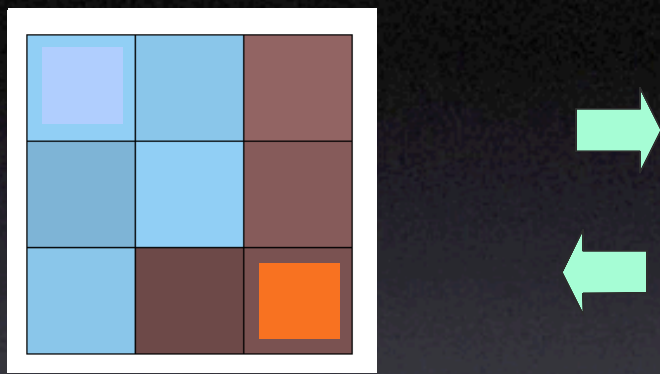
- Minimal Spanning Tree
- Shortest path
- s-t Max. graph flow, Min. cut

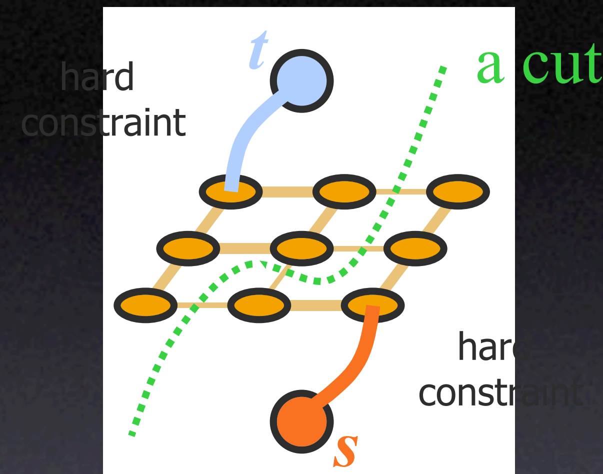
Graph Cut and Flow



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



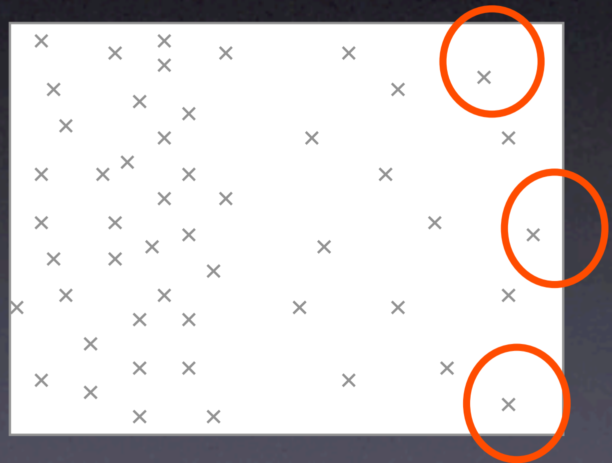


Minimum cost cut can be computed in polynomial time

(max-flow/min-cut algorithms)

(Boykov)

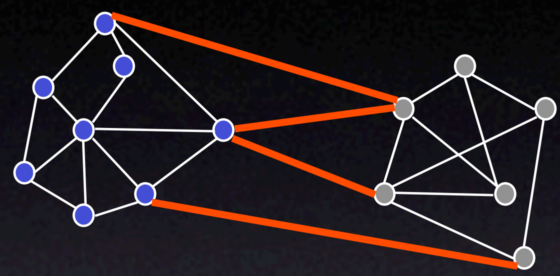
Problem with min cuts



Min. cuts favors isolated clusters

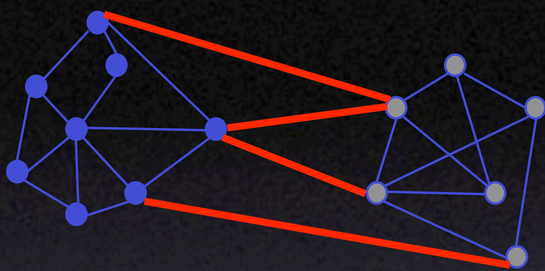
Normalize cuts in a graph

- (edge) Ncut = balanced cut



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

Normalized Cut and Normalized Association



$$Ncut(A,B) = \frac{cut(A,B)}{Vol(A)} + \frac{cut(A,B)}{Vol(B)}$$

$$Nassoc(A,B) = \frac{assoc(A,A)}{Vol(A)} + \frac{assoc(B,B)}{Vol(B)}$$

$$Ncut(A,B) = 2 - Nassoc(A,B)$$

- Minimizing similarity between the groups, and maximizing similarity within the groups can be achieved simultaneously.

Spectral Graph Segmentation

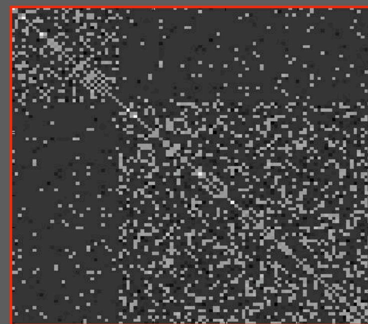


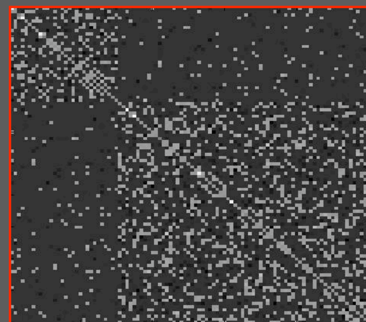
Image I → Graph Affinities W=W(I,Θ) → Eigenvector X(W)

$$NCut(A,B) = \frac{cut(A,B)}{Vol\ A'\ \ Vol\ B}$$

$$WX = \lambda DX$$

$$X_A(i) = \begin{cases} 1 & \text{if } i \in A \\ 0 & \text{if } i \notin A \end{cases}$$

Spectral Graph Segmentation

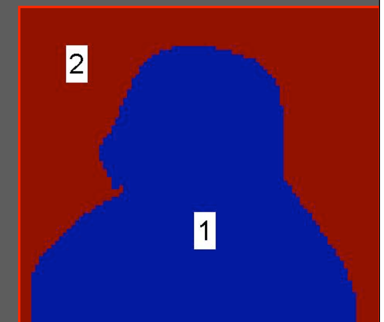


Image I → Graph Affinities $W=W(I,\Theta)$ → Eigenvector $X(W)$ → Discretisation

$$NCut(A,B) = \frac{cut(A,B)}{Vol\ A'\ \ Vol\ B}$$

$$WX = l\ DX$$

$$X_A(i) = \begin{cases} 1 \text{ if } i \in A \\ 0 \text{ if } i \notin A \end{cases}$$

Representation

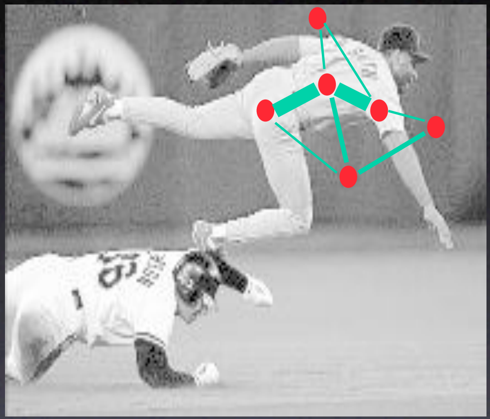
Partition matrix:

$$X = [X_1, ..., X_K]$$

segments

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

pixels



Representation

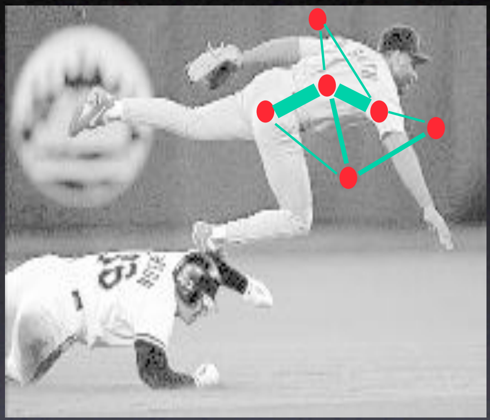
Partition matrix:

$$X = [X_1, ..., X_K]$$

segments

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

pixels



Pair-wise similarity matrix W

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

Representation

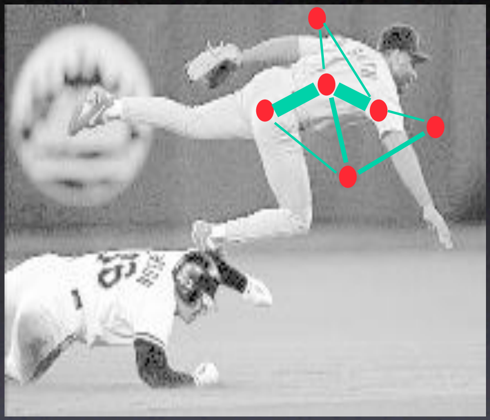
Partition matrix:

$$X = [X_1, \ldots, X_K]$$

segments

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

pixels

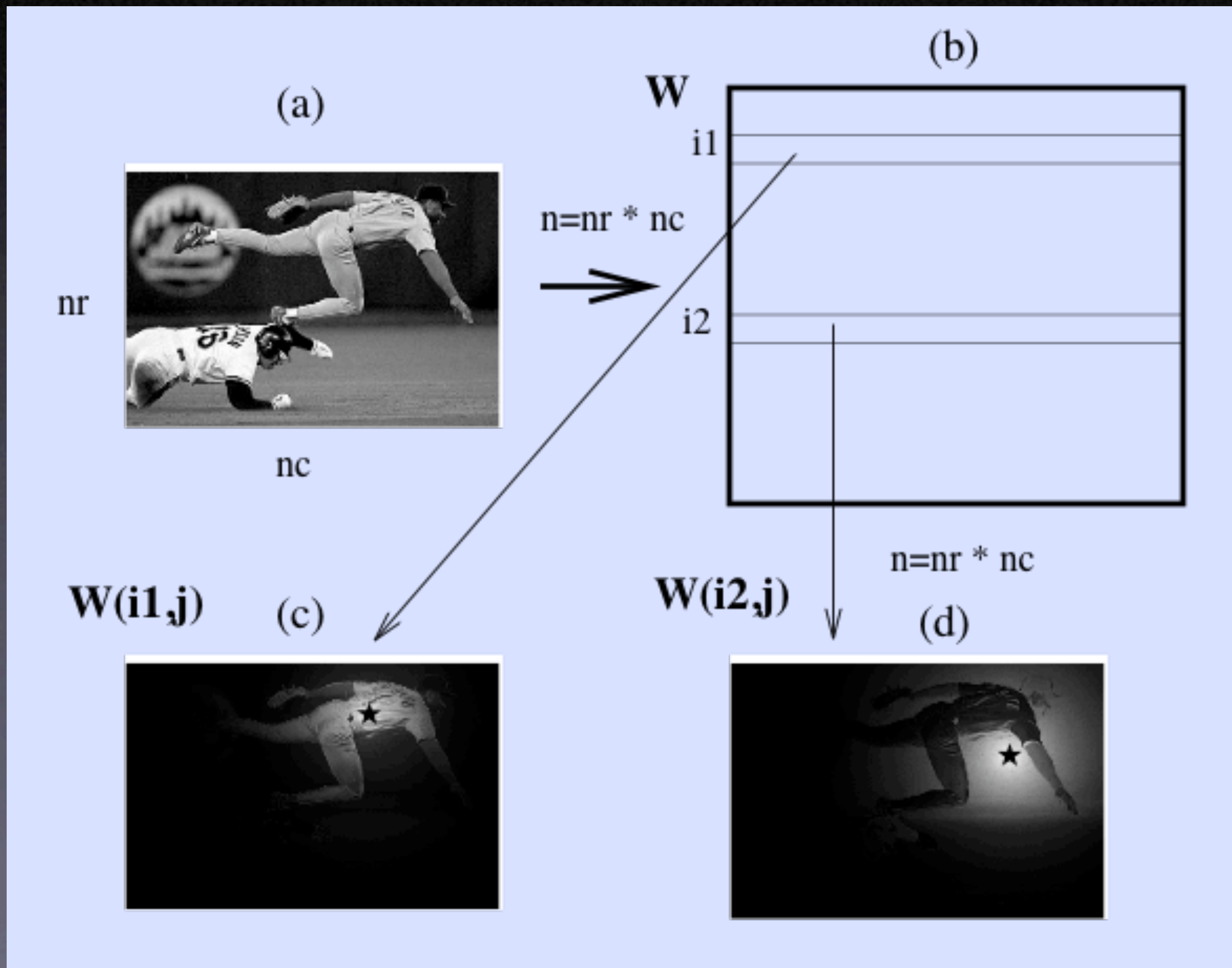


Pair-wise similarity matrix W

Laplacian matrix D-W

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

Graph weight matrix W

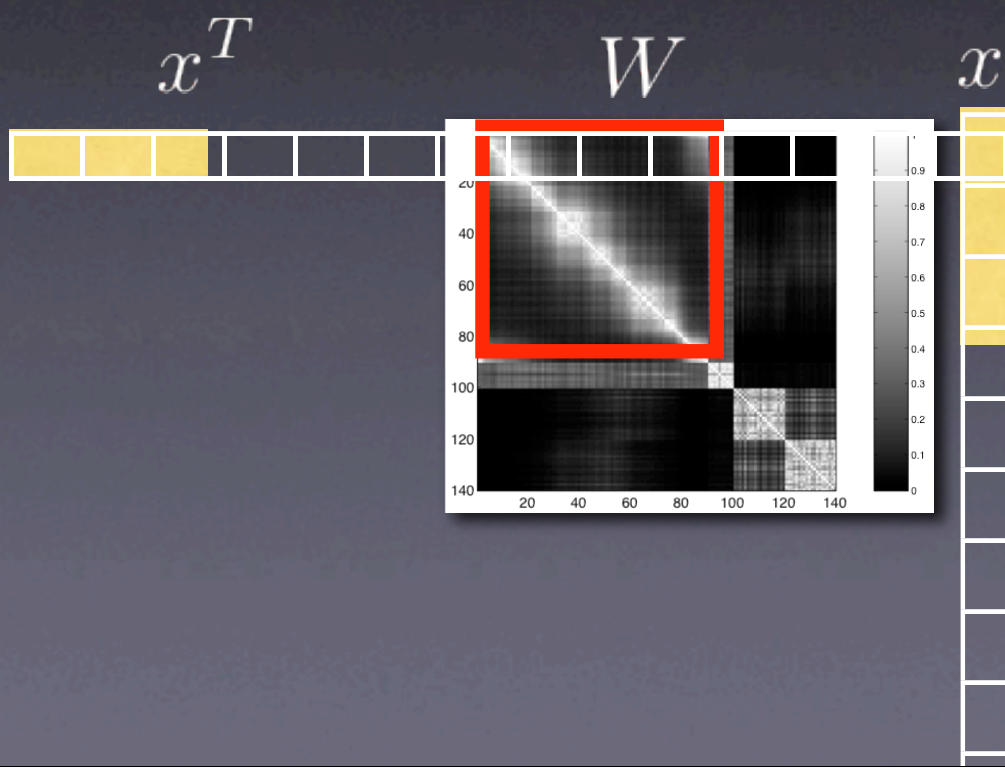


Laplacian matrix D-W

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

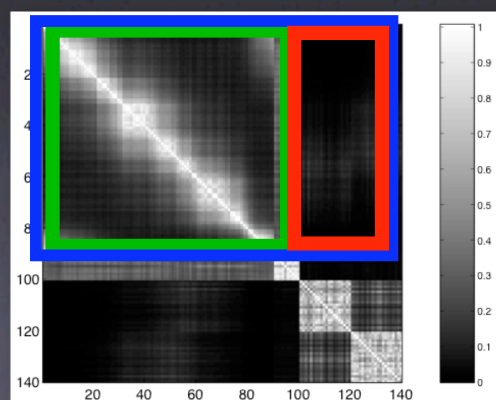
$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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



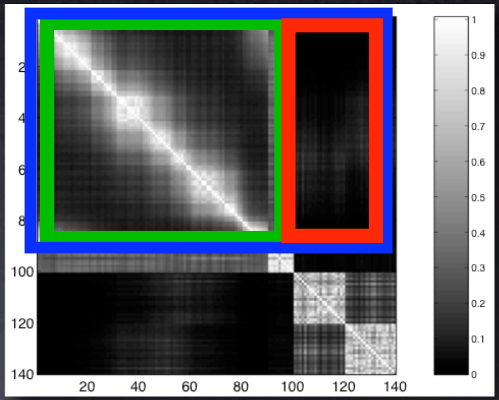
Laplacian matrix D-W

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



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

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



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

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

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

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

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

Minimize Ncut is NP-hard

Step 1: Find Continuous Global Optima

Scaled partition matrix. $Z = X(X^T D X)^{-\frac{1}{2}}$

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow Z = \begin{bmatrix} \frac{1}{\sqrt{vol(A)}} & 0 & 0 \\ \frac{1}{\sqrt{vol(A)}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{vol(B)}} & 0 \\ 0 & \frac{1}{\sqrt{vol(B)}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{vol(C)}} \end{bmatrix}$$

Step I: Find Continuous Global Optima

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

becomes

$$Ncut(Z) = \frac{1}{K} tr(Z^T W Z) \qquad Z^T D Z = I_K$$

becomes

$$Ncut(Z) = \frac{1}{K} tr(Z^T W Z) \qquad Z^T D Z = I_K$$

We use the generalization of the Rayleigh-Ritz theorem to solve it.

Rayleigh and…

Ritz

becomes

$$Ncut(Z) = \frac{1}{K} tr(Z^T W Z) \qquad Z^T D Z = I_K$$

Rayleigh and Ritz Says:

Eigensolutions

$$(D - W)z^* = \lambda D z^*$$

$$Z^* = [z_1^*, z_2^*, ..., z_k^*]$$

becomes

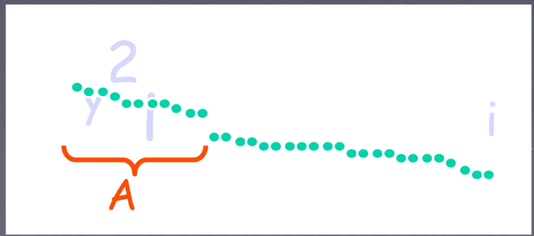
$$Ncut(Z) = \frac{1}{K} tr(Z^T W Z) \qquad Z^T D Z = I_K$$

Rayleigh and Ritz Says:

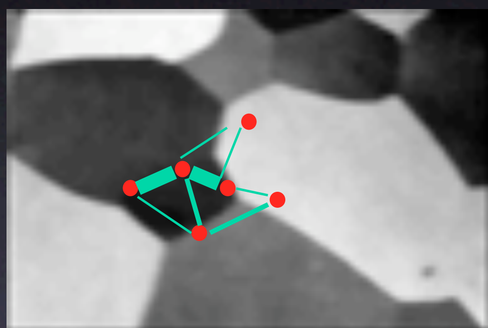
Eigensolutions

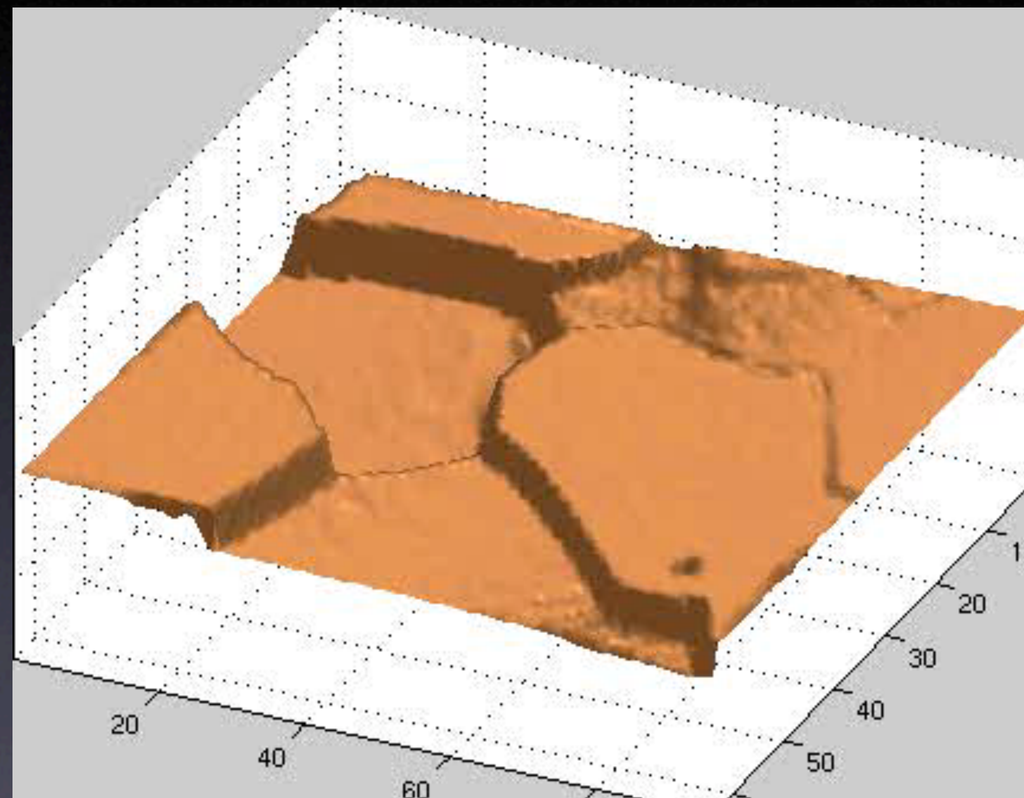
$$(D - W) z^* = \lambda D z^*$$

$$Z^* = [z_1^*, z_2^*, \ldots, z_k^*]$$



Interpretation as a Dynamical System





Step II: Discretize Continuous Optima

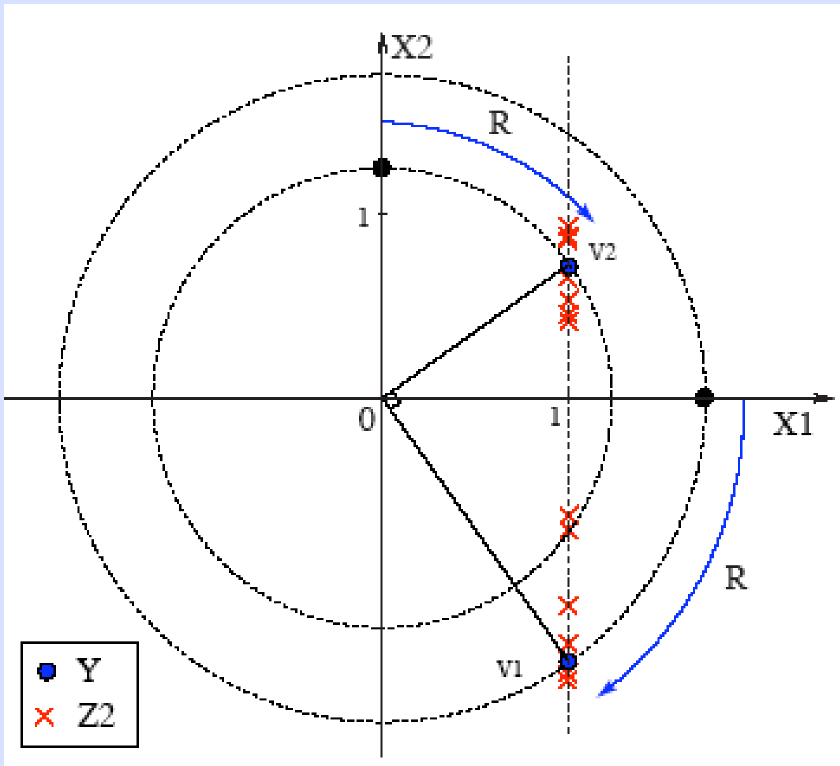
Partition	Scaled Partition	Eigenvector solution

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Leftrightarrow Z = \begin{bmatrix} \frac{1}{\sqrt{vol(A)}} & 0 & 0 \\ \frac{1}{\sqrt{vol(A)}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{vol(B)}} & 0 \\ 0 & \frac{1}{\sqrt{vol(B)}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{vol(C)}} \end{bmatrix} \Leftrightarrow Z^* = \begin{bmatrix} \frac{1}{\sqrt{vol(A)}} & 0 & 0 \\ \frac{1}{\sqrt{vol(A)}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{vol(B)}} & 0 \\ 0 & \frac{1}{\sqrt{vol(B)}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{vol(C)}} \end{bmatrix} \times \begin{bmatrix} cos(\theta) & \text{-}sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

$$(D - W)Z^* = \lambda D Z^*$$

If Z* is an optimal, so is $\{ZR : R^T R = I_K\}$

Step II: Discretize Continuous Optima



Target partition

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

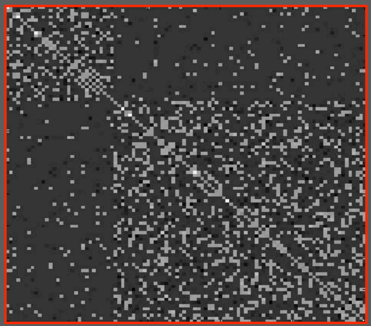
Rotation R

Eigenvector solution

$$Z^* = \begin{bmatrix} 1 & -1.4 \\ 1 & -1.3 \\ 1 & 0.8 \\ 1 & 0.9 \\ 1 & 0.7 \end{bmatrix}$$

Rotation R can be found exactly in 2-way partition

Spectral Graph Segmentation



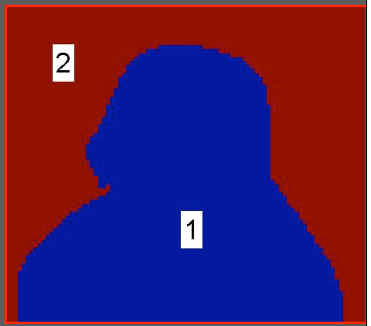
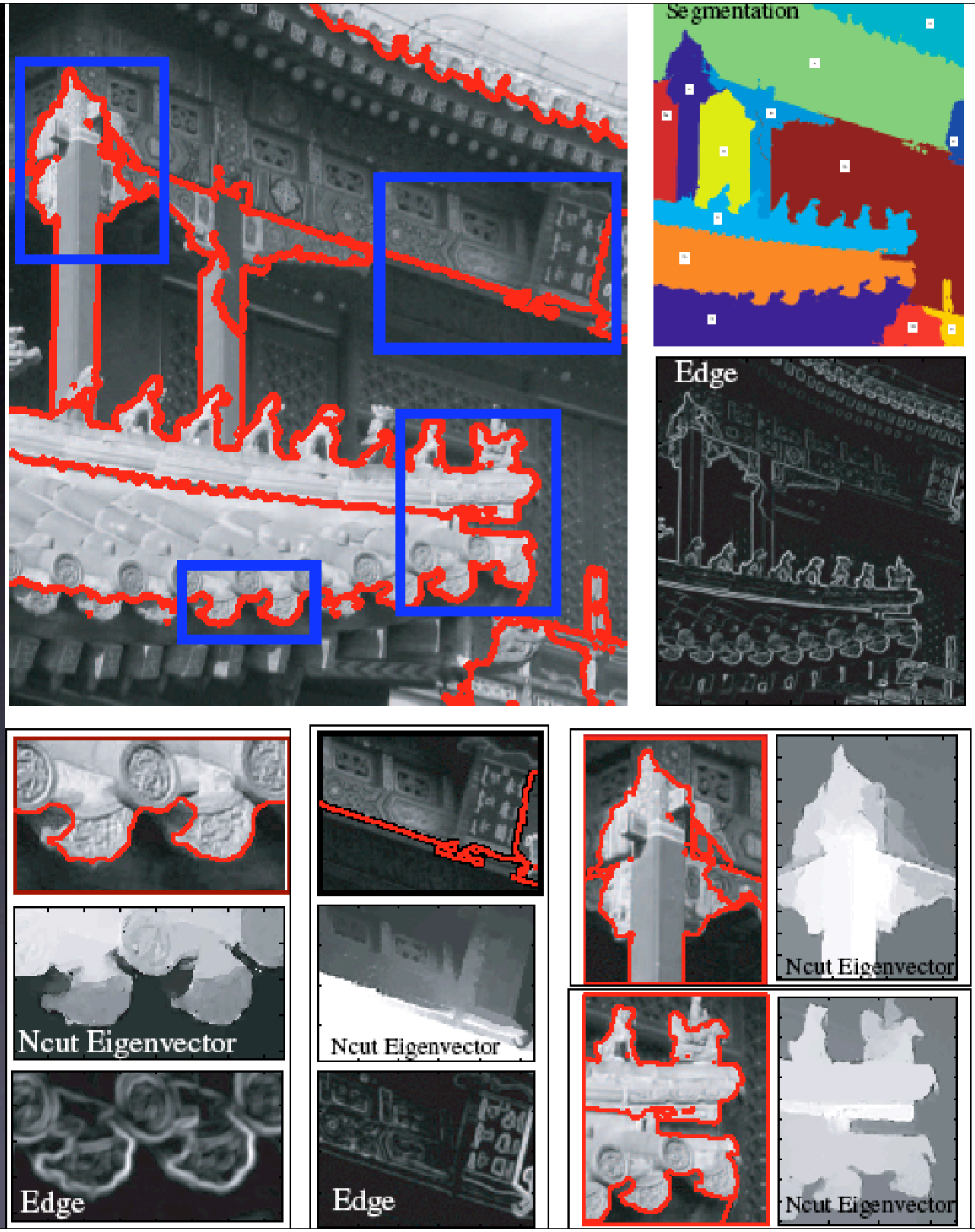


Image I → Graph Affinities W=W(I,Θ) → Eigenvector X(W) → Discretisation

$$NCut(A,B) = \frac{cut(A,B)}{Vol\ A' \ Vol\ B}$$

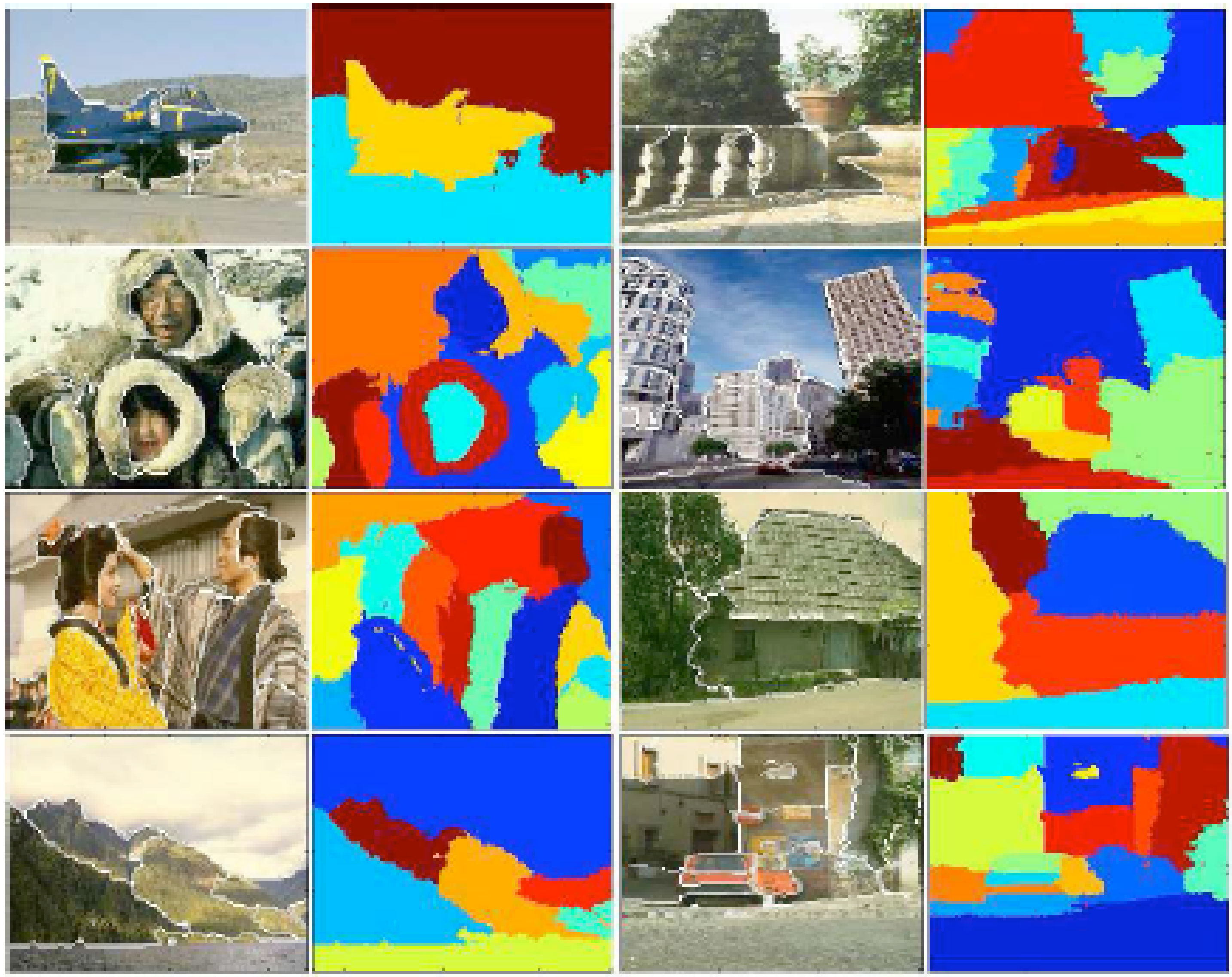
$$WX = l\ DX$$

$$X_A(i) = \begin{cases} 1 \text{ if } i \in A \\ 0 \text{ if } i \notin A \end{cases}$$

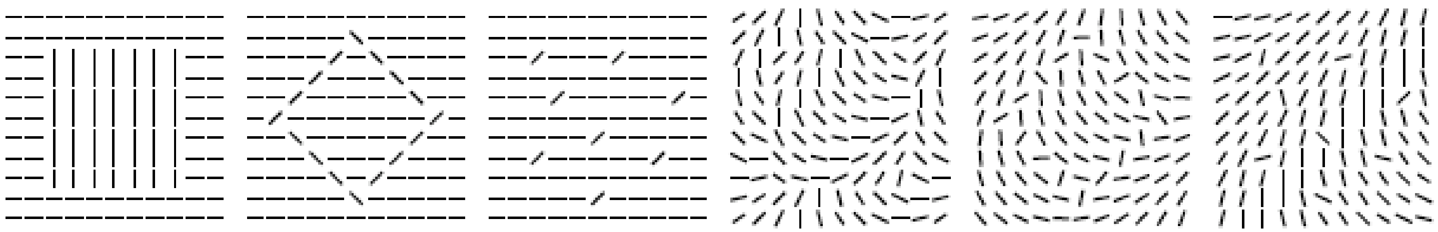


Multiscale NCut Segmentation

[Cour,Benezit,Shi, CVPR05]

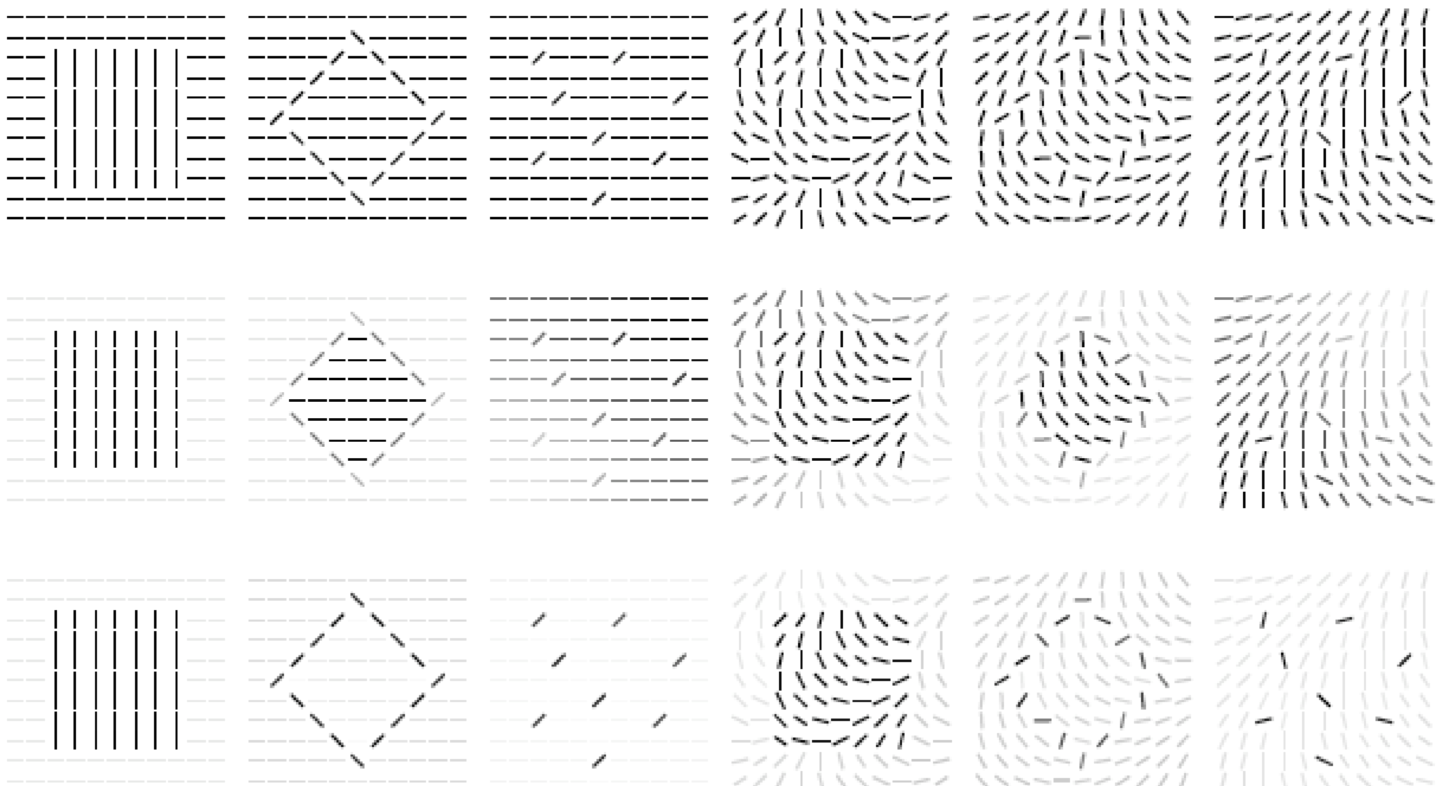


Visual Popout [Yu, Shi 2001]:

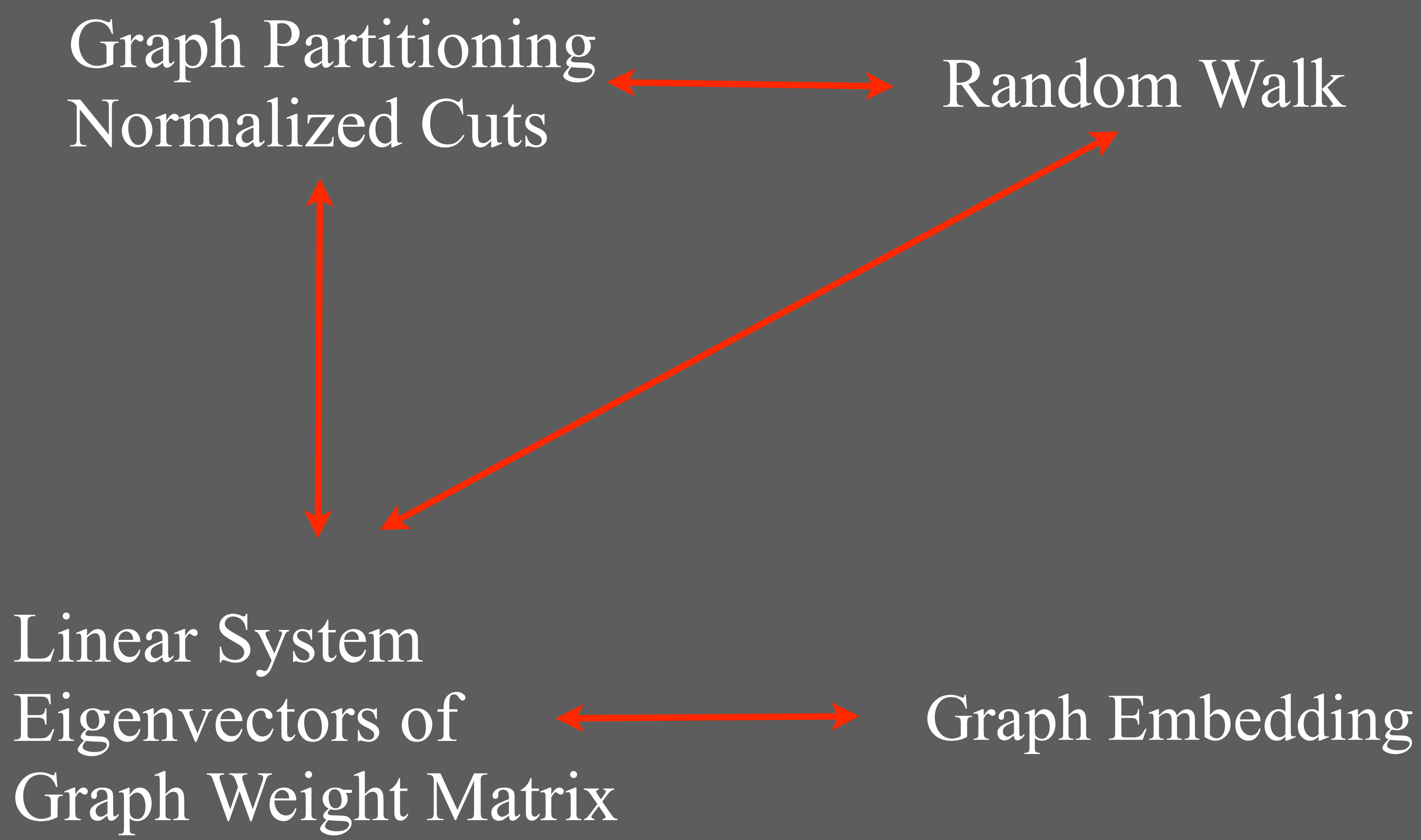


Visual Popout: Graph with negative weights

positive links only



$$W = W_+ - W_- = A - R,$$


Graph Partitioning
Normalized Cuts
Random Walk
Linear System
Eigenvectors of
Graph Weight Matrix
Graph Embedding

The random walks view

- Construct the matrix

$$P = D^{-1}S$$

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \cdots & \\ & & & d_n \end{bmatrix} \qquad S = \begin{bmatrix} S_{11} & S_{12} & S_{1n} \\ S_{21} & S_{22} & S_{2n} \\ & \cdots & \\ S_{n1} & S_{n2} & S_{nn} \end{bmatrix}$$

- P is stochastic matrix $\sum_j P_{ij} = 1$
- P is transition matrix of Markov chain with state space I

$\pi = \frac{1}{\text{vol } I}[\, d_1 \; d_2 \; \ldots \; d_n \,]^T$ is stationary distribution

Reinterpreting the NCut criterion

$$NCut(A, \bar{A}) = P_{A\bar{A}} + P_{\bar{A}A}$$

$$P_{AB} = Pr[\, A \dashrightarrow B \mid A \,] \text{ under } P, \pi$$

- NCut looks for sets that "trap" the random walk
- Related to Cheeger constant, conductivity in Markov chains

Reinterpreting the NCut algorithm

$$(D-W)y = \mu Dy$$

$$\mu_1 = 0 \leqslant \mu_2 \leqslant \ldots \leqslant \mu_n$$

$$y^1 \quad y^2 \quad \ldots \quad y^n$$

$$Px = \lambda x$$

$$\lambda_1 = 1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n$$

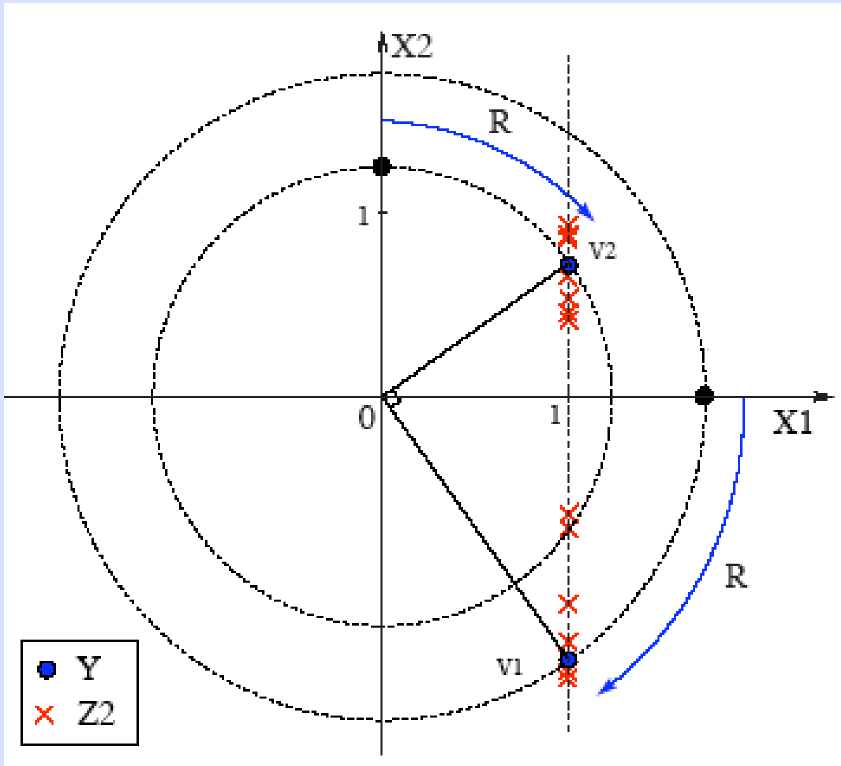
$$x^1 \quad x^2 \quad \ldots \quad x^n$$

$$\mu_k = 1 - \lambda_k$$

$$y^k = x^k$$

The NCut algorithm segments based on the second largest eigenvector of P

Relationship to Graph embedding



Target partition

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

Rotation R

Eigenvector solution

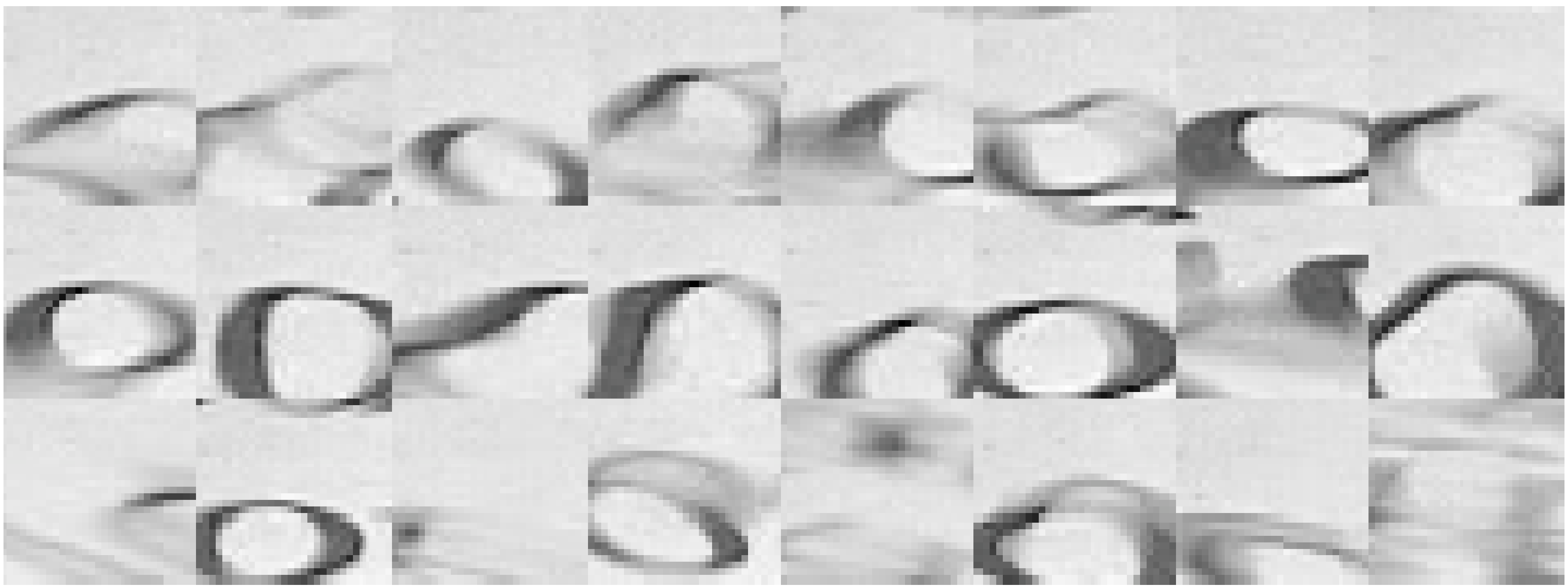
$$Z^* = \begin{bmatrix} 1 & -1.4 \\ 1 & -1.3 \\ 1 & 0.8 \\ 1 & 0.9 \\ 1 & 0.7 \end{bmatrix}$$

Rotation R can be found exactly in 2-way partition

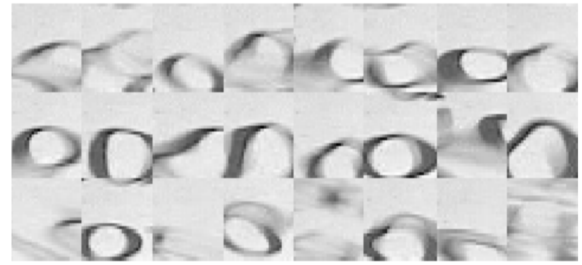
Seeing Through Water…

Efros, Shi, Visontai, Esler, NIPS 04

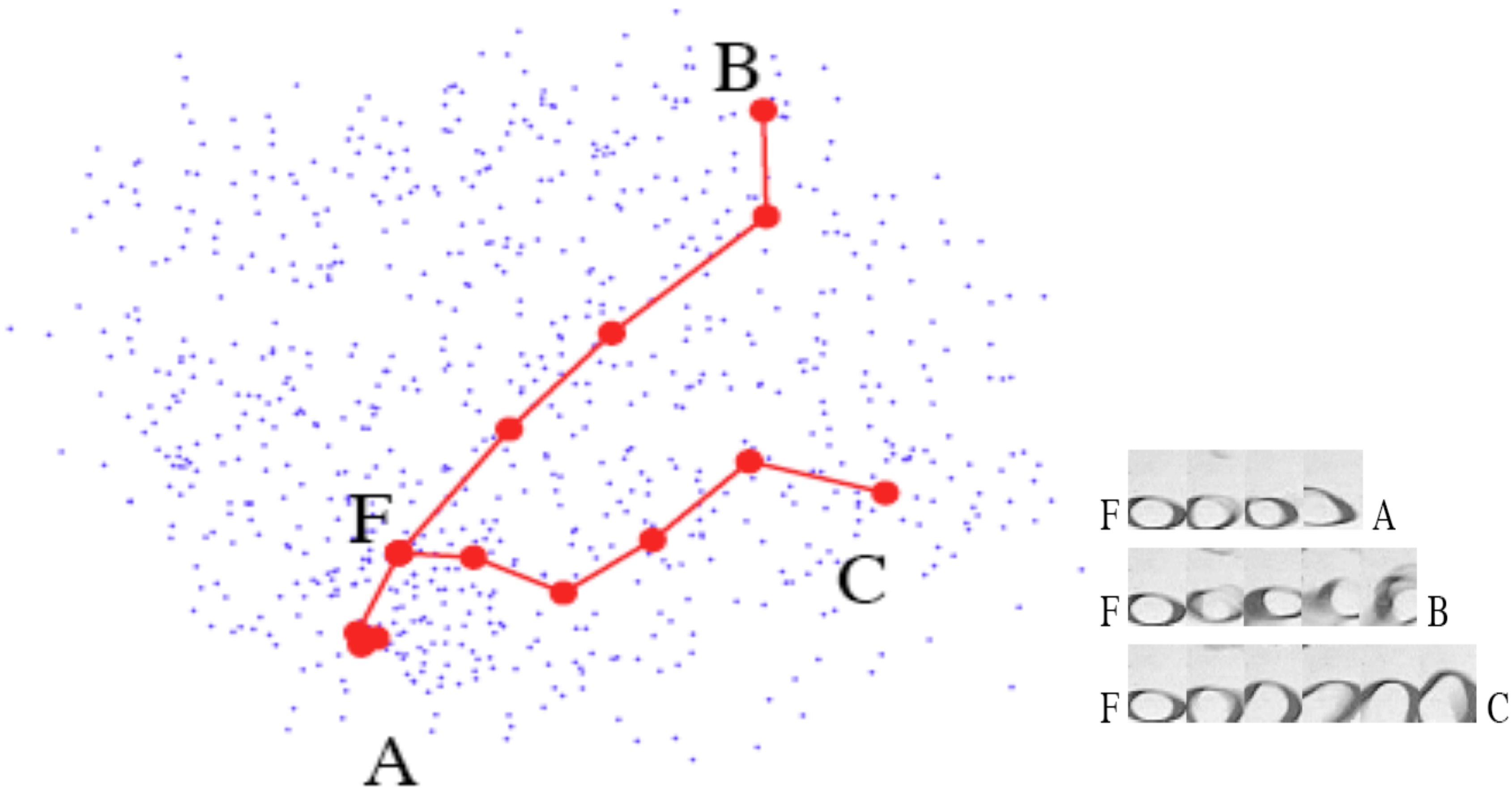
Patches observed at one fixed location from the previous slide:



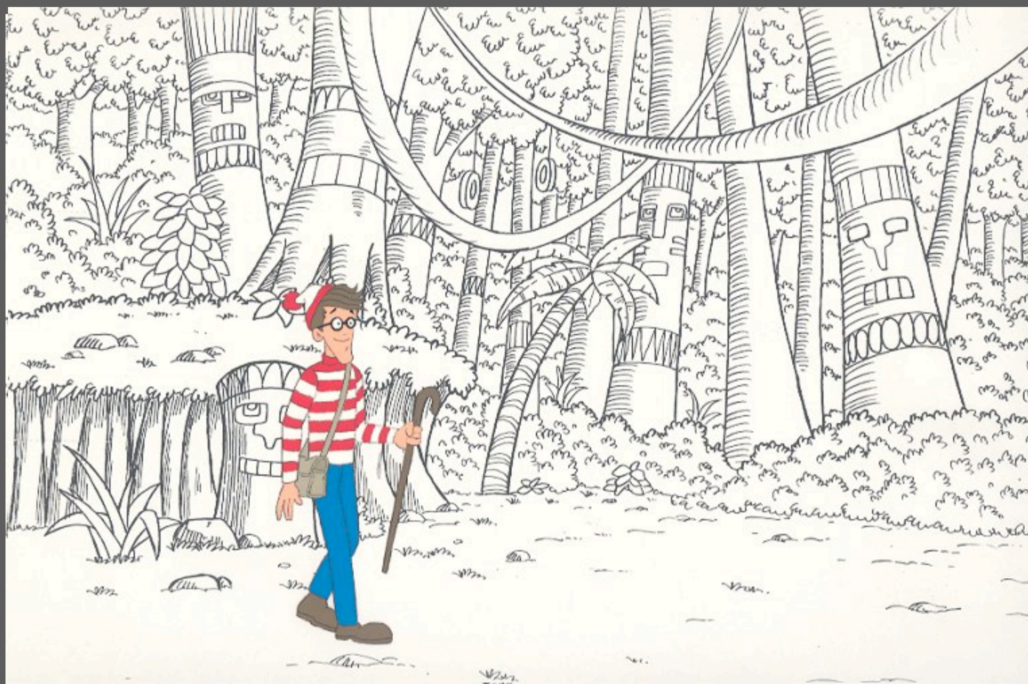
Patches observed at one fixed location:



Hypothesized embedding of these patches (we assume they form a manifold)



Where is Waldo ?

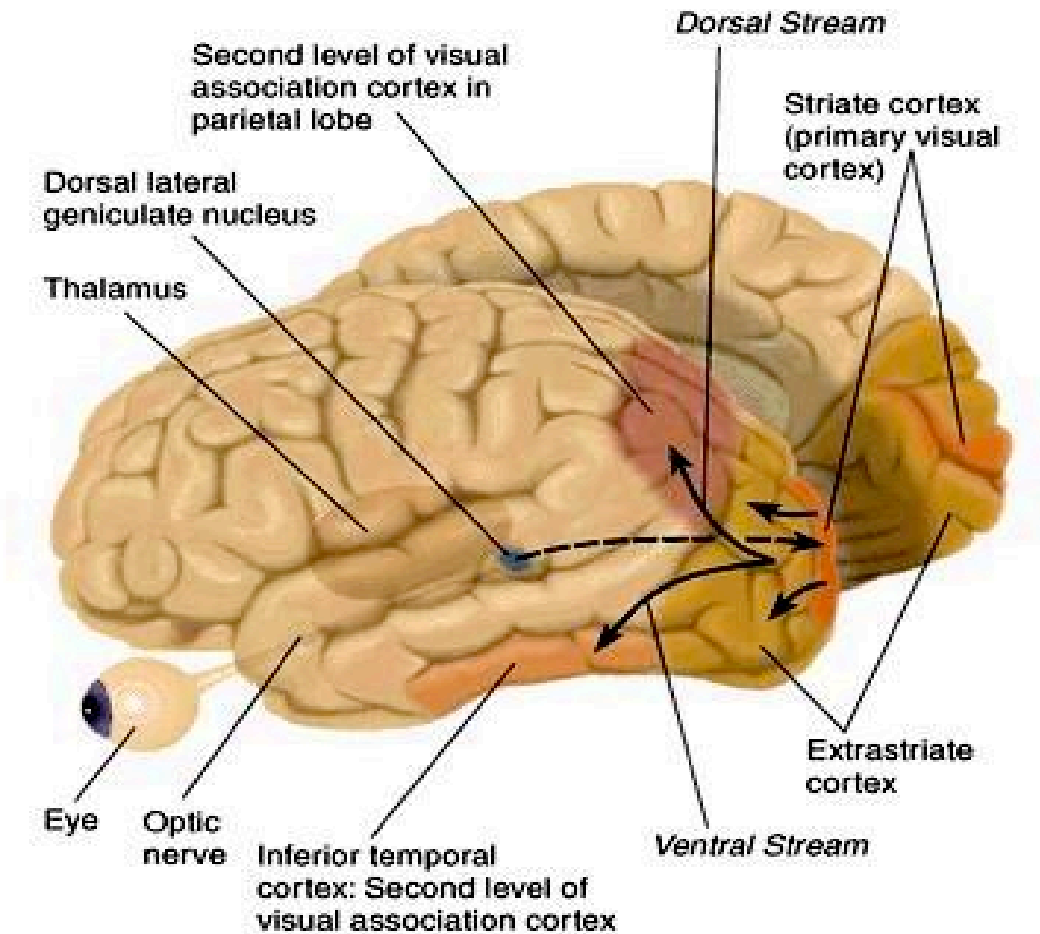


Do you use

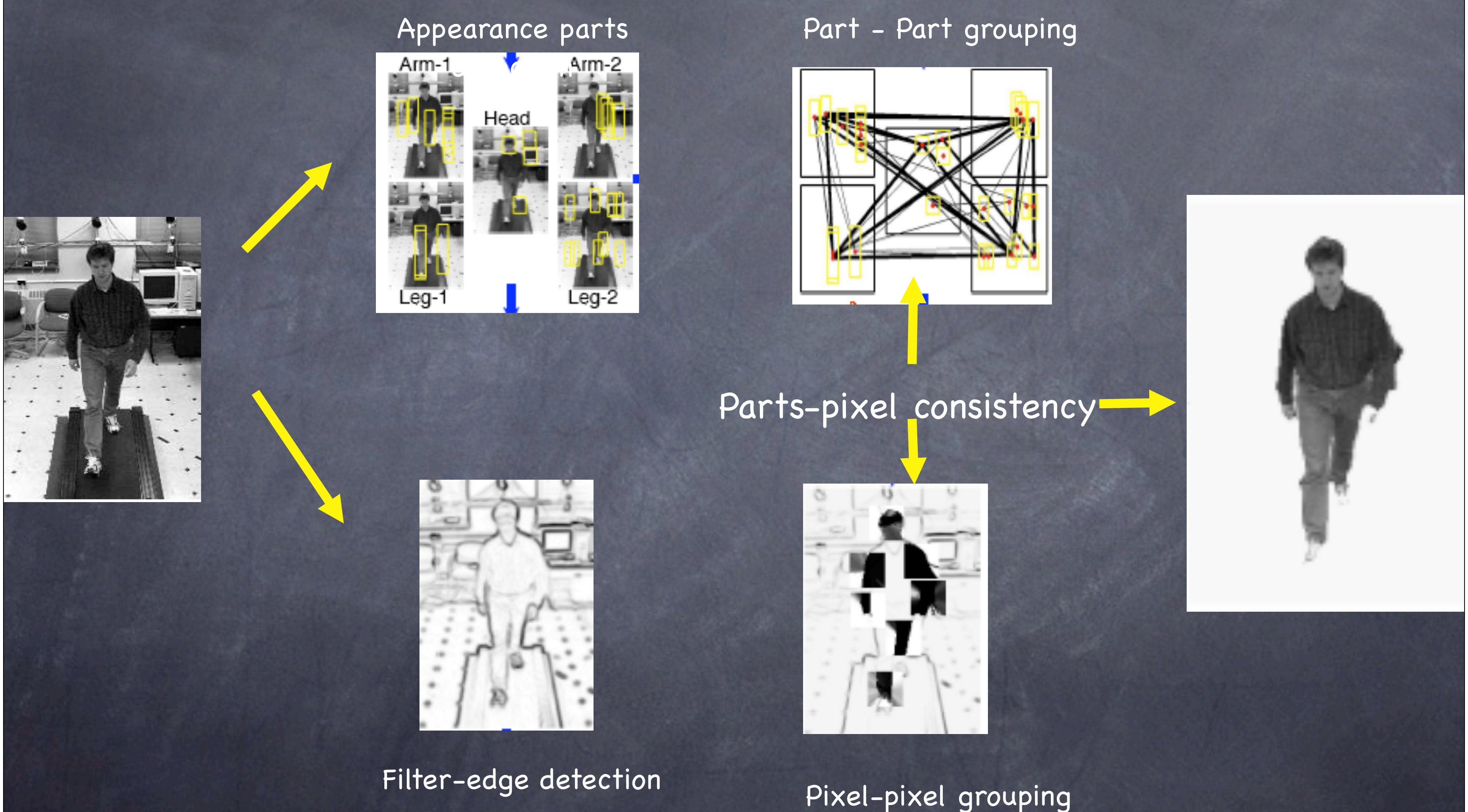
Edges cues ?
Color cues ?
Texture cues ?

-That’s not enough, you need
Shape cues
High-level object priors

► The Human Visual System



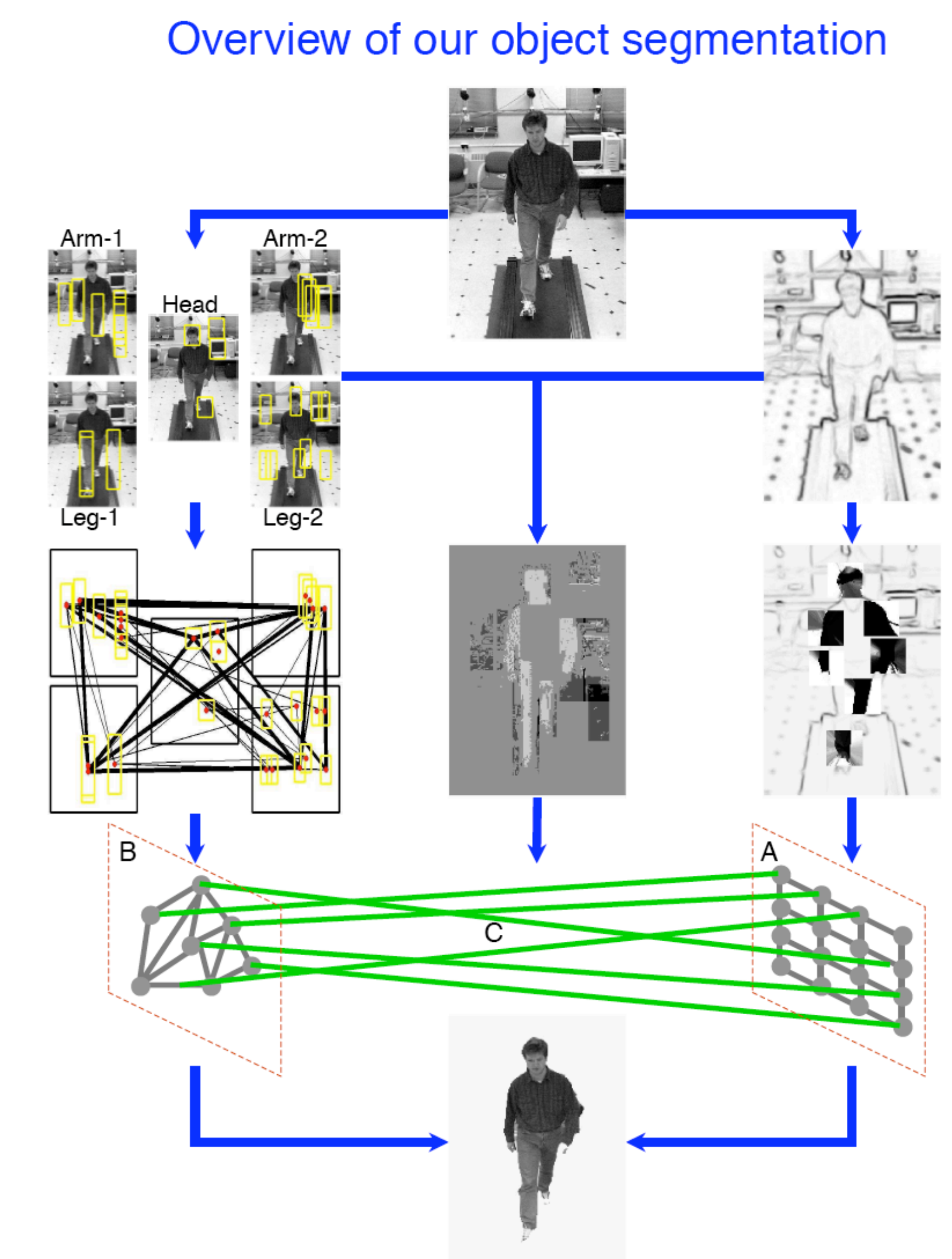
Con-current recognition-segmentation



[Yu, Shi, CVPR'03]

Con-current recognition-segmentation

[Yu, Shi, CVPR'03]



Representation

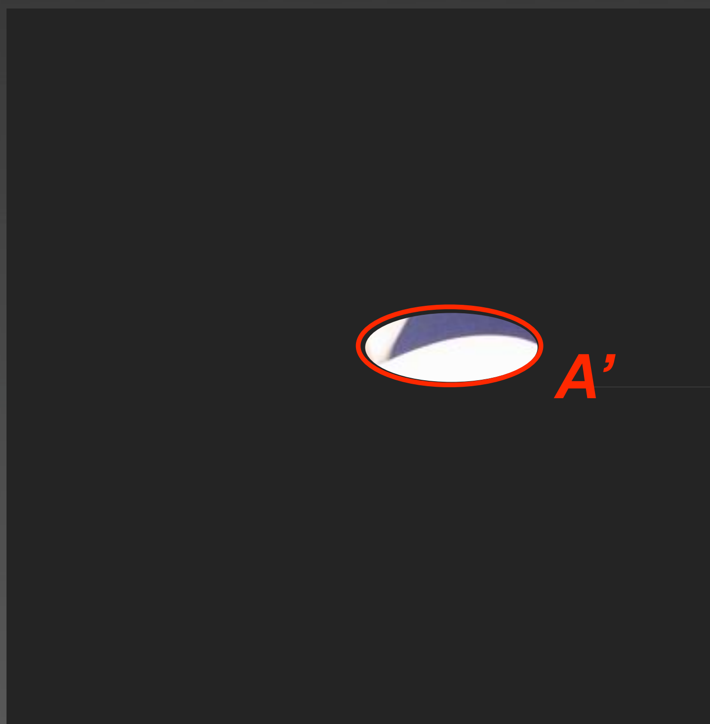
Graph: $\mathsf{G} = (\mathsf{V}, \mathsf{E}, W)$ = (nodes, edges, weights)

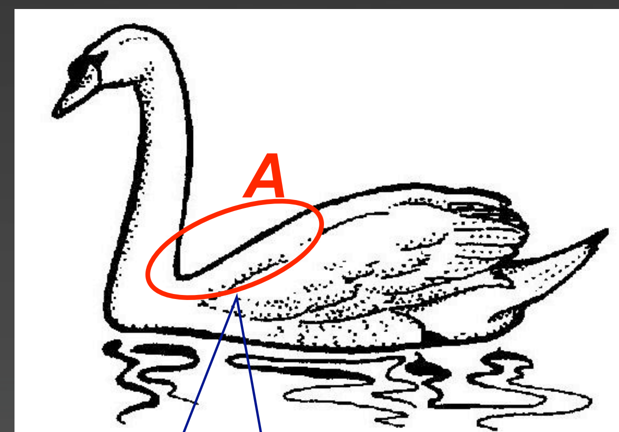
Node set: $\mathsf{V} = \mathsf{V}_{\mathrm{pixels}} \cup \mathsf{V}_{\mathrm{patches}}$

Edge set: $\mathsf{E} = \mathsf{E}_{\mathrm{pixel-pixel}} \cup \mathsf{E}_{\mathrm{patch-patch}} \cup \mathsf{E}_{\mathrm{pixel-patch}}$

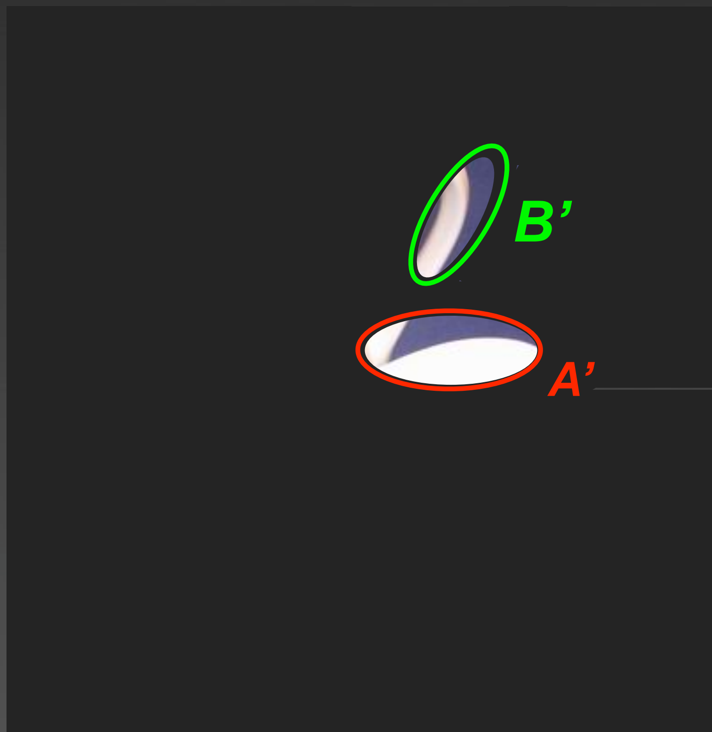
Weights: $W = \begin{bmatrix} A & C^T \\ C & B \end{bmatrix}$

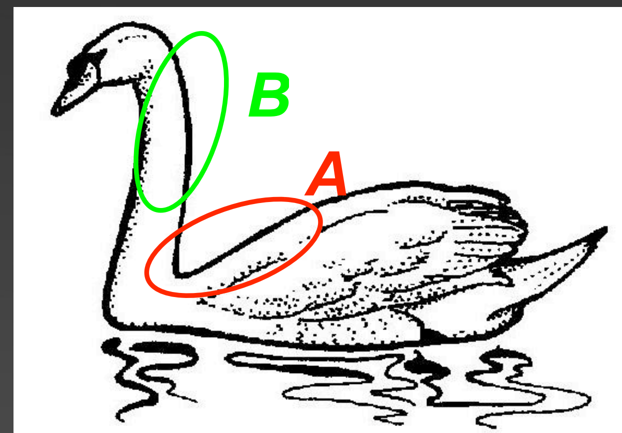
A: pixel-pixel similarity matrix
B: patch-patch compatibility matrix
C: pixel-patch association matrix





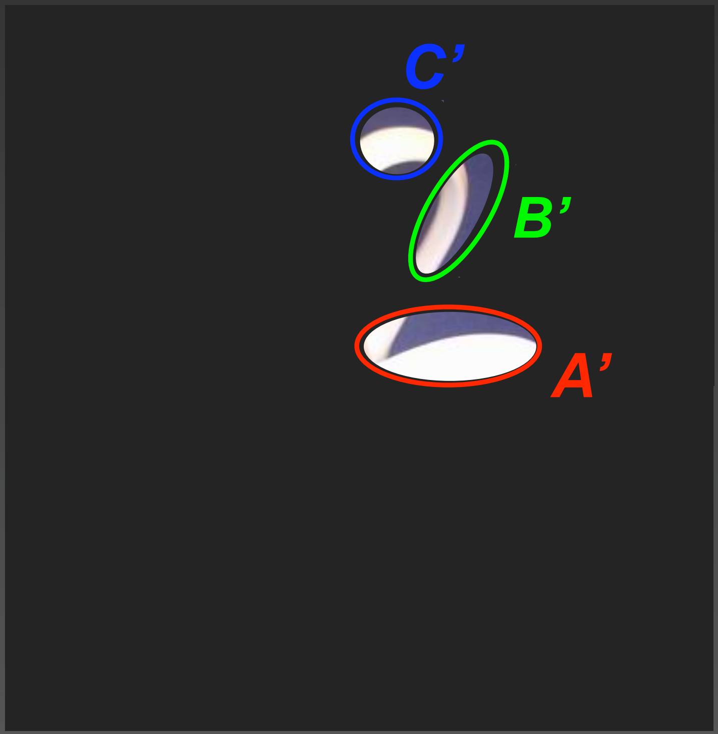
This could be the back

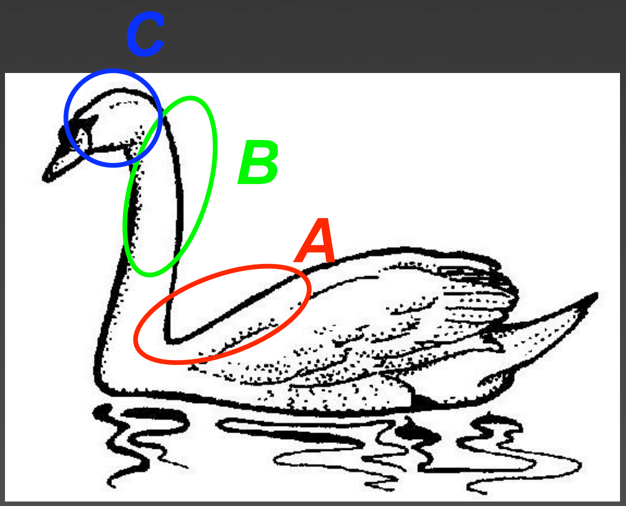




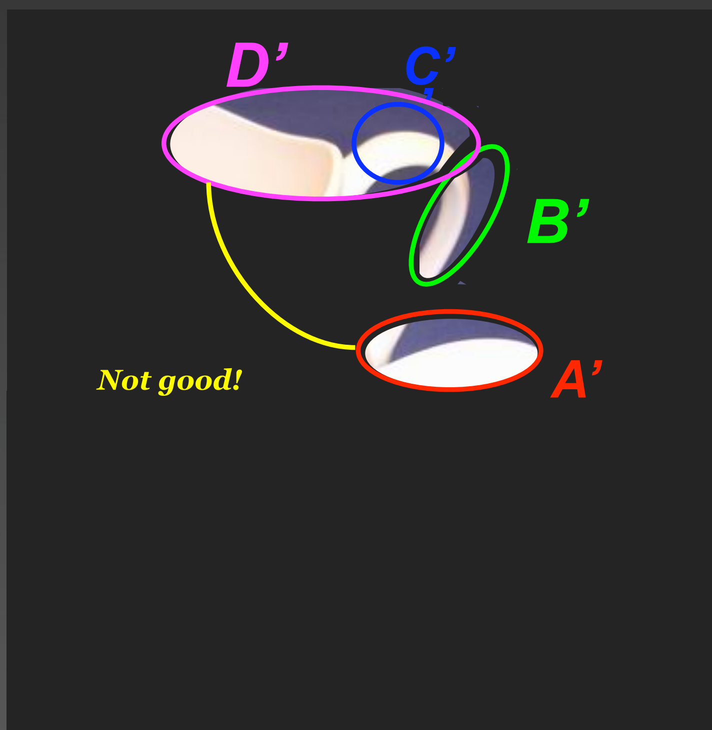
$$A \rightarrow B$$

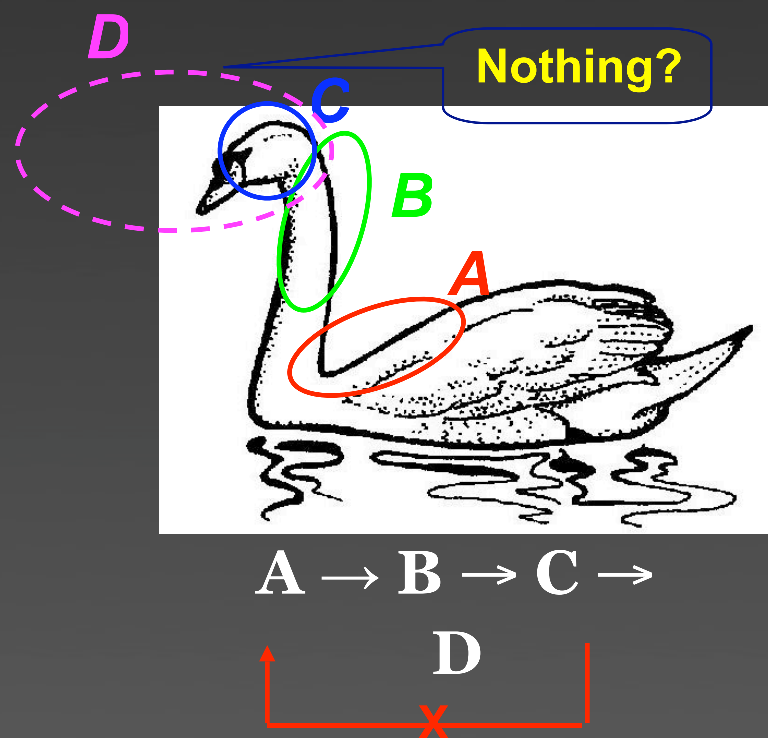
Context enhances contour perception





$$A \rightarrow B \rightarrow C$$


D’
C’
B’
A’
Not good!


D
Nothing?
C
B
A
A → B → C →
D
X

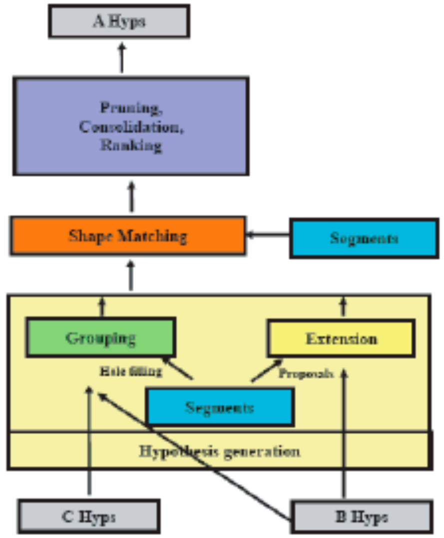
Accidental alignment happens. Context is very useful, but need to find the 'Right' context first.

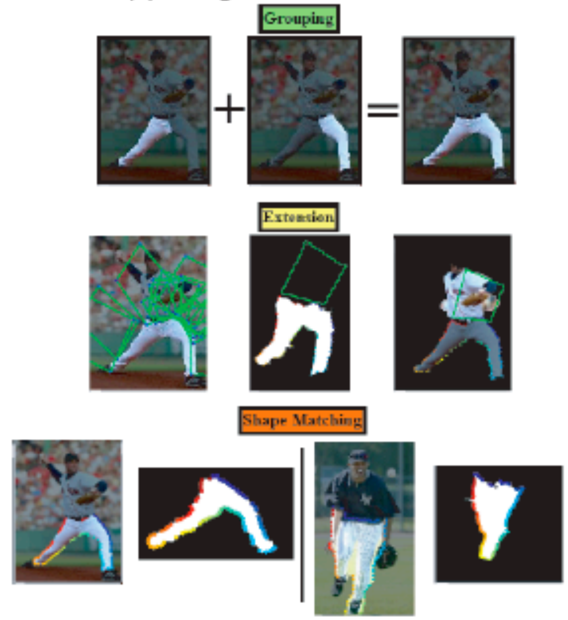
Bottom-up Recognition and Parsing of the Human Body

[Srinivasan&Shi, 07]

Segmentation

Composition rule for body parsing





Rule based search

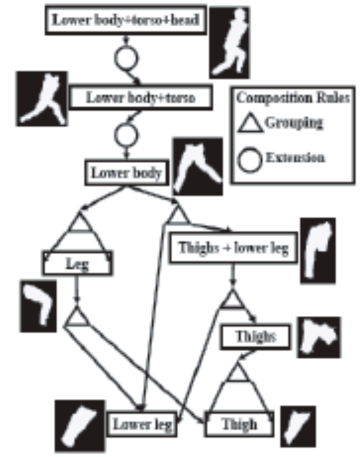
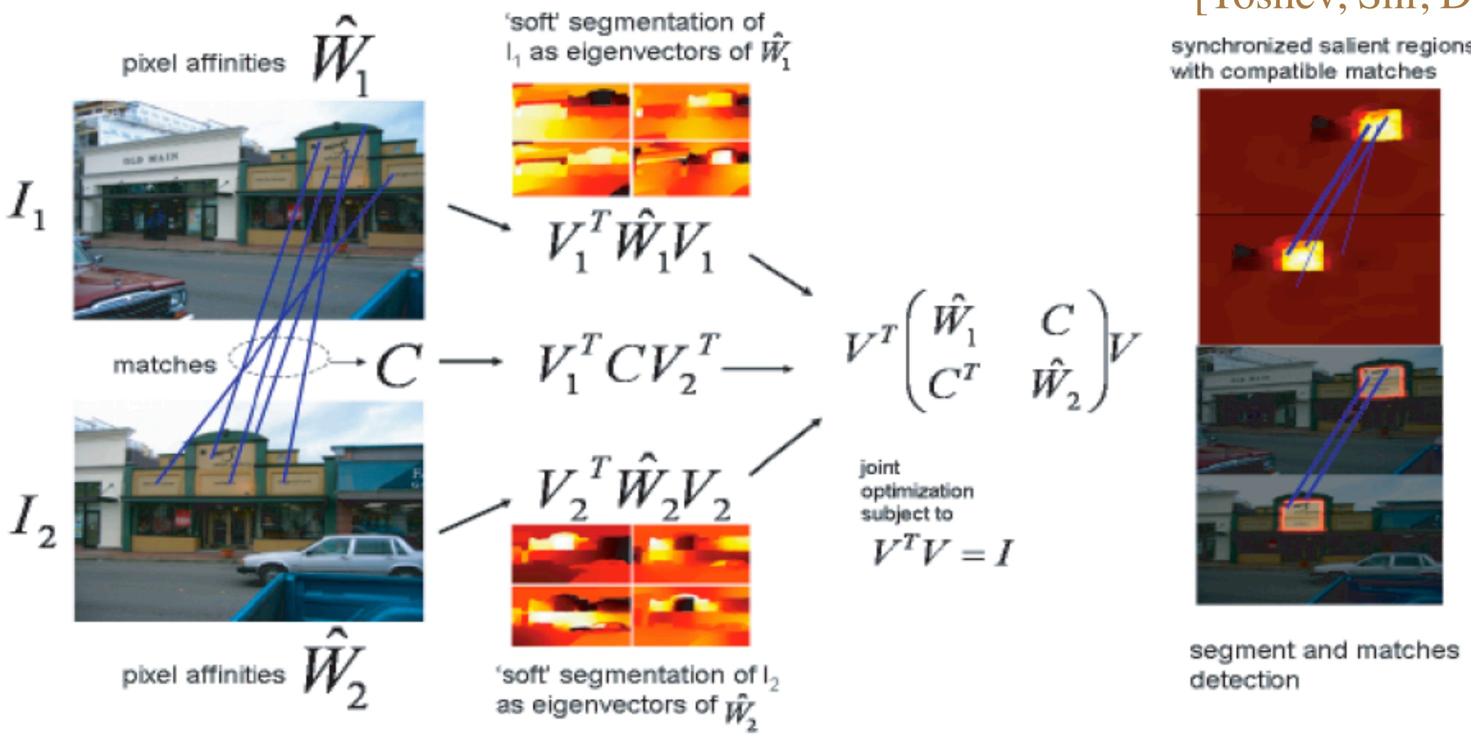
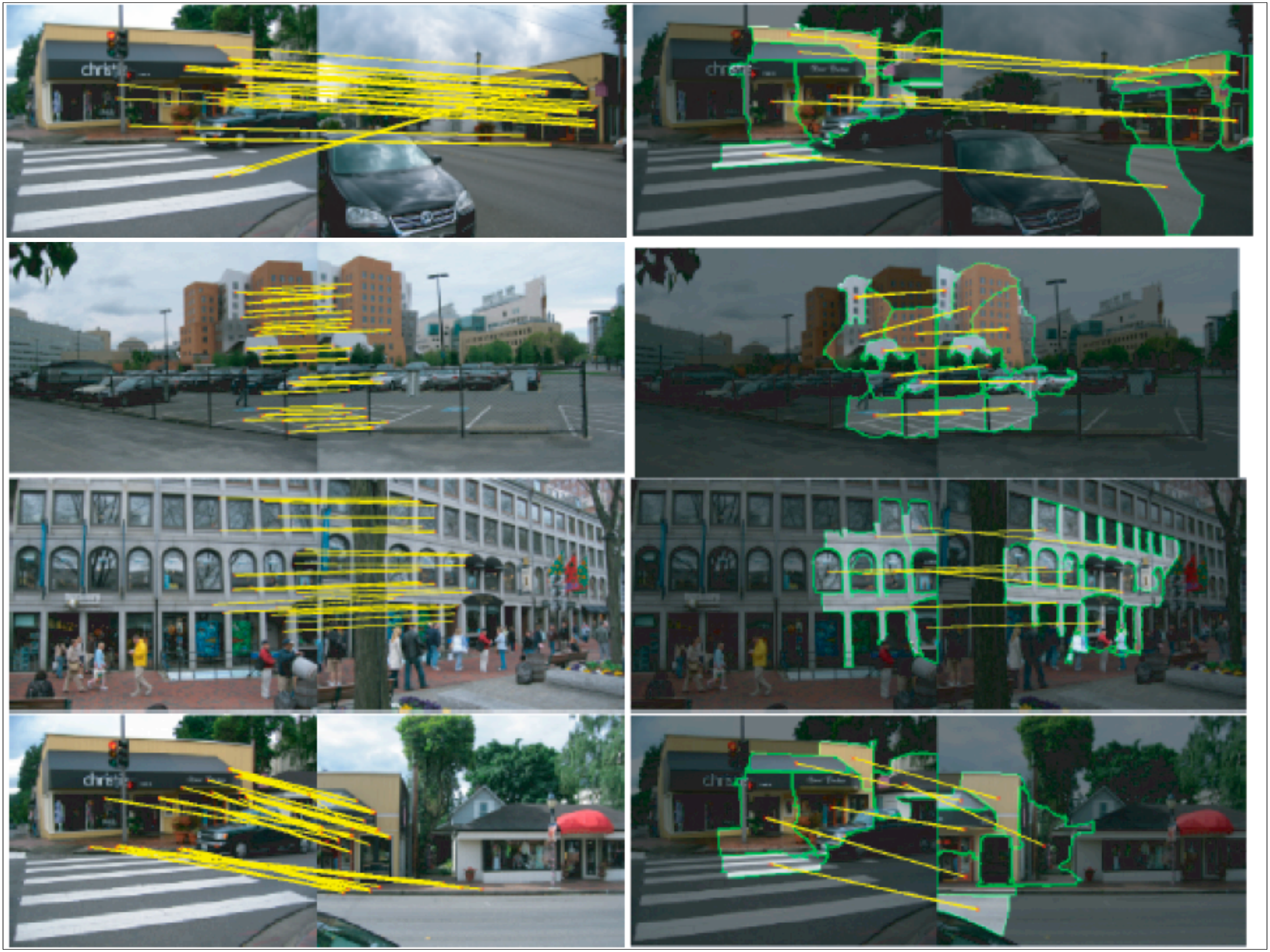


Image Matching via Saliency Region Correspondences

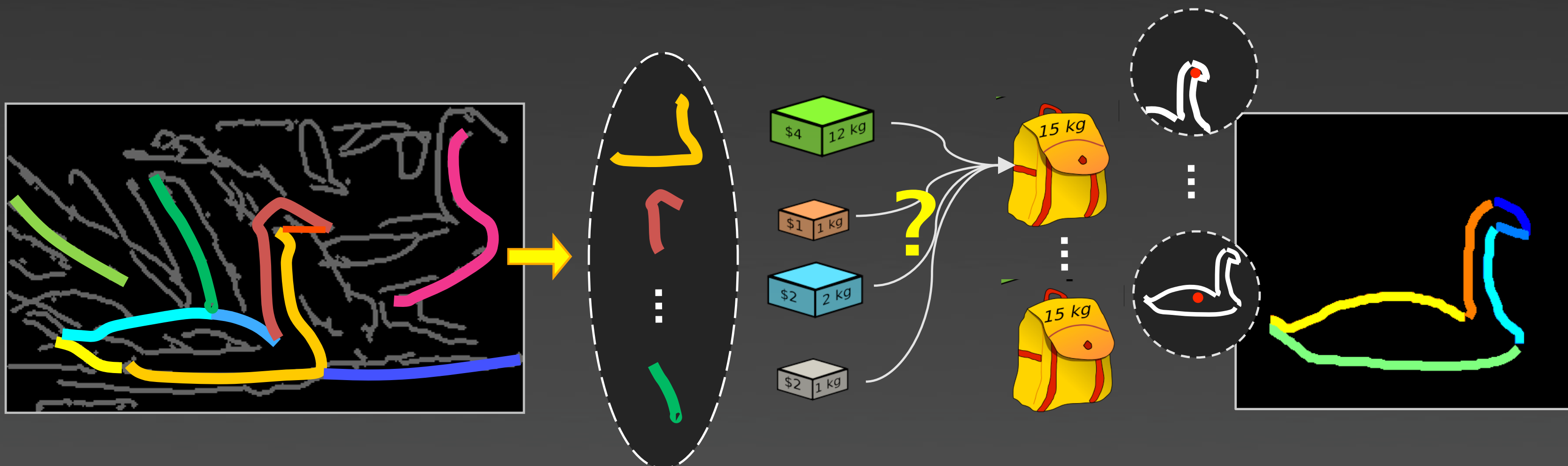
[Toshev, Shi, Daniidis, 07]



Our key observation is that point feature correspondence in isolateion is often wrong and uncertain, while segmentation alone produces different segments in different images. In our solution, we seek a correspondence that can lead to consistent joint-segmentation across two images. We utilize the 'soft' subspace generated by the spectral Ncut to reduce and stablize the correspondence search process.



Contour Packing for Object Recognition

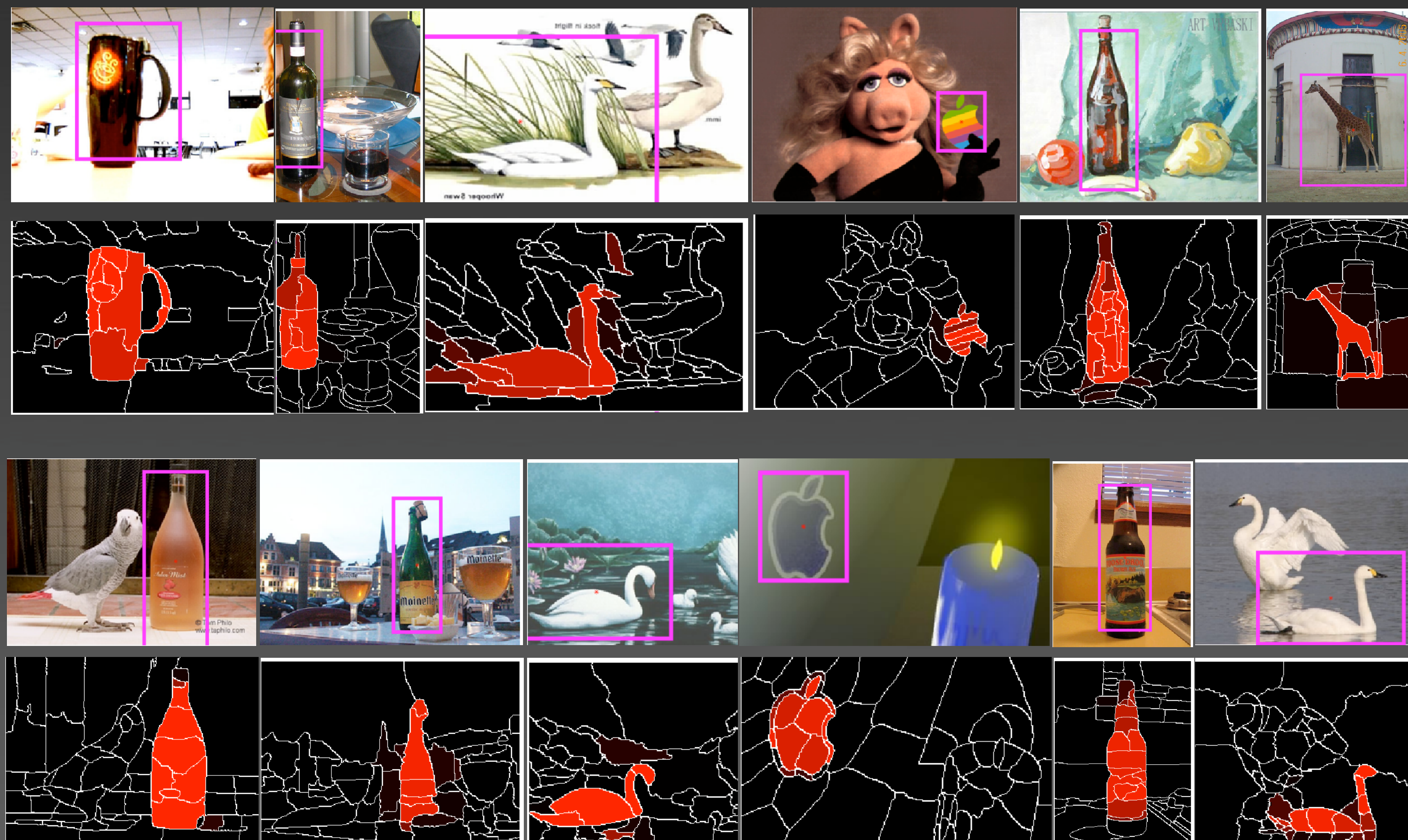


Jianbo Shi

Joint work with Qihui Zhu, Praveen Srinivasan, Liming Wang, Yang Wu

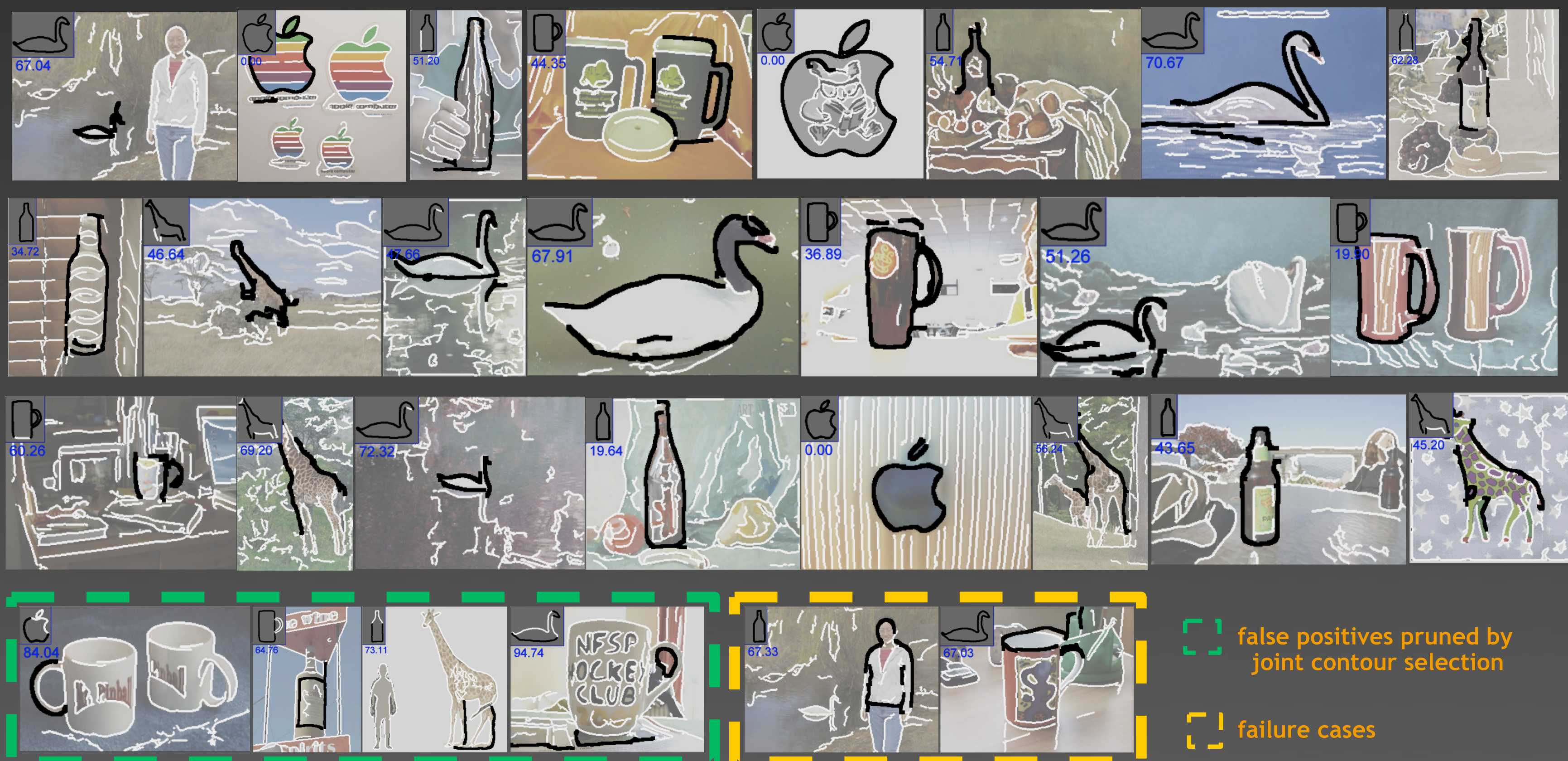
Penn

Object Recognition via Region Packing



Results on ETHZ

Used only one hand-drawn model per class



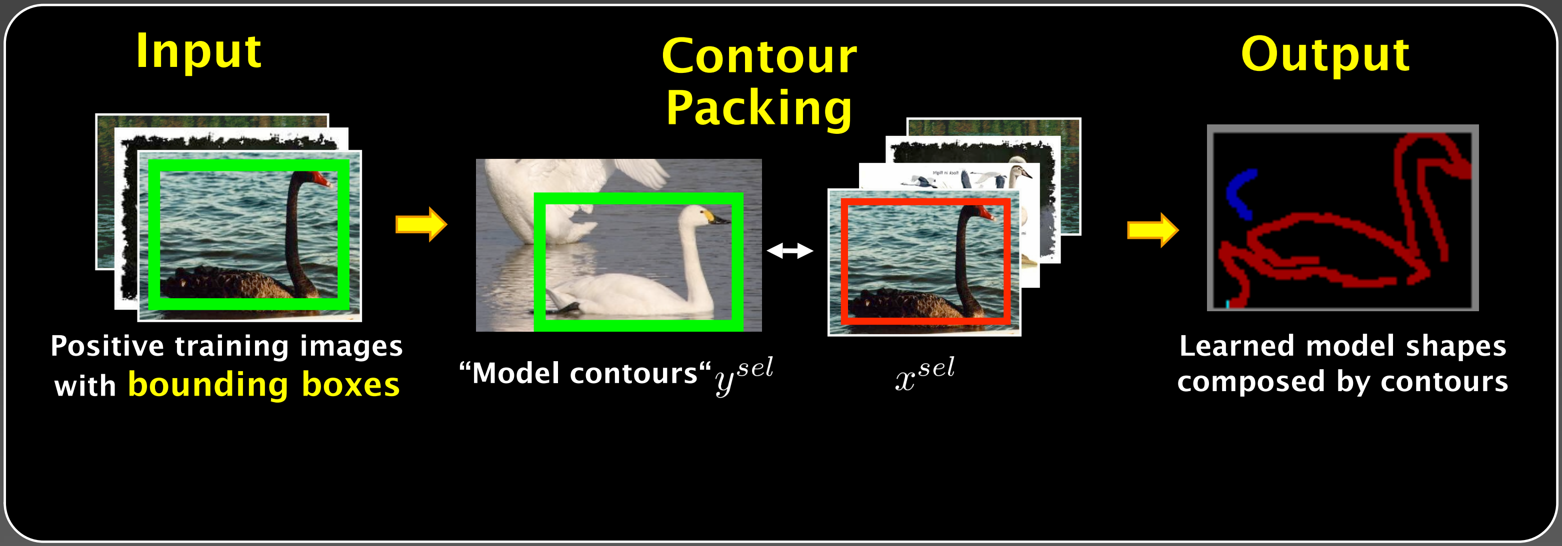
false positives pruned by joint contour selection

failure cases

Detection results of our method. Model selection is shown on the top-left corner.

Model Shape Learning

- Learn shapes without initial models
- Contour packing for finding common shapes
  - Choose all contours in each bounding box as model
  - Do contour packing across all other training images
  - Contours selected on the model compose common shapes



Preliminary Results

Top 5 learned model from bounding boxes