Image Segmentation
continued
Graph Based Methods

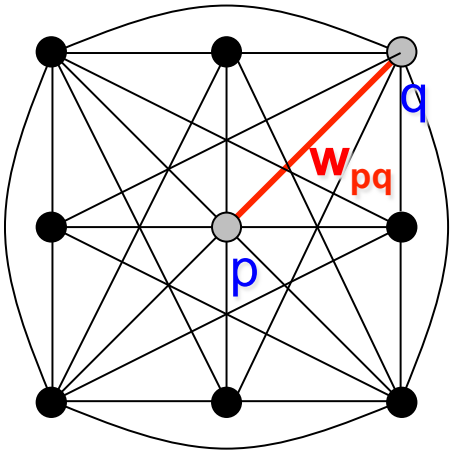
Some slides: courtesy of O. Capms, Penn State,
J.Ponce and D. Fortsyth, Computer Vision Book

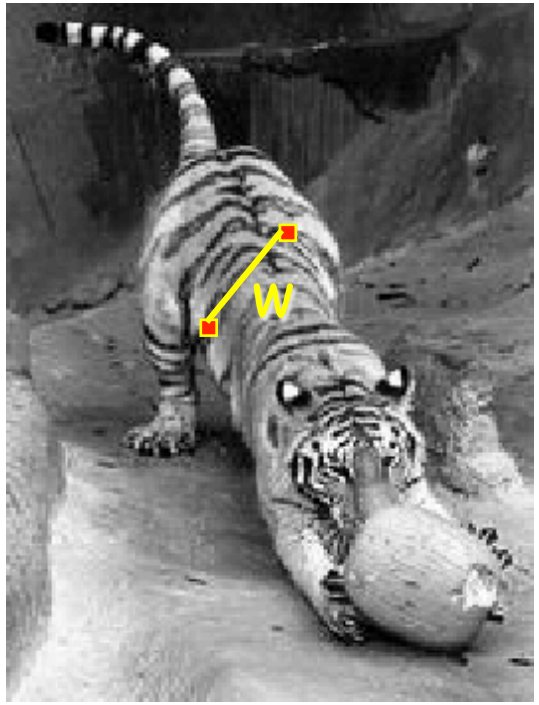
Previously

- Binary segmentation
- Segmentation by thresholding
- Background subtraction
- Motion segmentation
- K-means clustering

- Grouping and graph based segmentation

Images as graphs





- *Fully-connected* graph
  - node (vertex) for every pixel
  - link between *every* pair of pixels, **p**,**q**
  - affinity weight $\mathbf{w_{pq}}$ for each link (edge)
    - $\mathbf{w_{pq}}$ measures *similarity*
      - similarity is *inversely proportional* to difference (in color and position…)

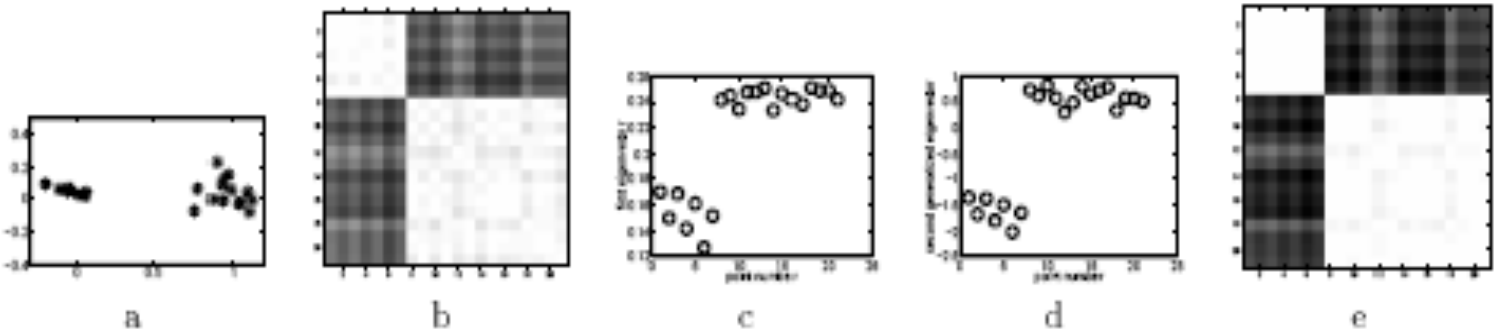
Source: Steve Seitz

Segmentation as Graph Partitioning

- (Shi & Malik ‘’97) Normalized Cut Algorithm
- Idea – each pixel in the image is a node in the graph
- Arcs represent similarities between adjacent pixels
- Graph is fully connected
- Goal – partition the graph into a sets of vertices (regions), such that the similarity within the region is high – and similarity across the regions is low.

Segmentation

- Toy example
- Bright entries in the affinity matrix high
- Likely to belong together

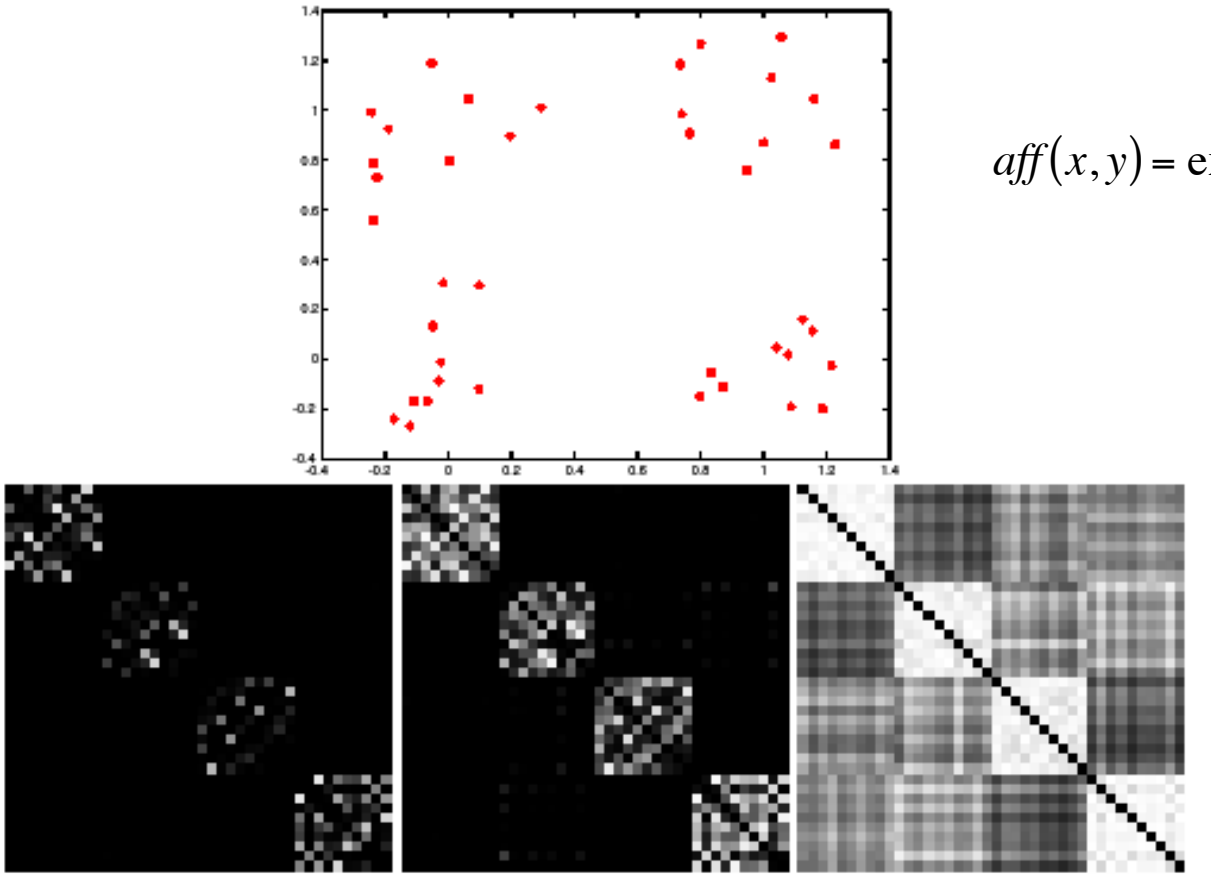


- one possible affinity based on distance

$$aff(x,y) = \exp\left\{-\left(1/2\sigma_d^2\right)\left(\|x-y\|^2\right)\right\}$$

Scale affects affinity

Depending on the scale the blocks are more (middle)
Or less obvious (left and right)



$$aff(x,y) = \exp\left\{-\left(1/2\sigma_d^2\right)\left(\|x-y\|^2\right)\right\}$$

$$\sigma_d = \{0.1, 0.2, 1\}$$

Measuring Affinity

Intensity

$$aff(x,y) = \exp\left\{-\left(1/2\sigma_i^2\right)\left(\|I(x) - I(y)\|^2\right)\right\}$$

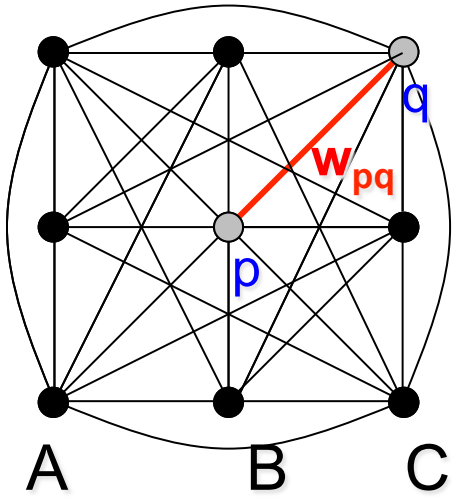
Distance

$$aff(x,y) = \exp\left\{-\left(1/2\sigma_d^2\right)\left(\|x - y\|^2\right)\right\}$$

Texture

$$aff(x,y) = \exp\left\{-\left(1/2\sigma_t^2\right)\left(\|c(x) - c(y)\|^2\right)\right\}$$

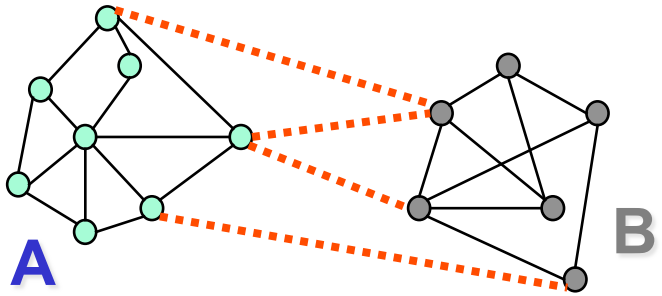
Segmentation by Graph Cuts



- Break Graph into Segments
  - Want to delete links that cross **between** segments
  - Easiest to break links that have low similarity (low weight)
    - similar pixels should be in the same segments
    - dissimilar pixels should be in different segments

Source: Steve Seitz

Cuts in a graph: Min cut



- Link Cut
  - set of links whose removal makes a graph disconnected
  - cost of a cut: $cut(A,B) = \sum_{p \in A, q \in B} w_{p,q}$

Find minimum cut

- gives you a segmentation
- fast algorithms exist for doing this

Source: Steve Seitz

Minimum cut

- Problem with minimum cut:

  Weight of cut proportional to number of edges in the cut; tends to produce small, isolated components.

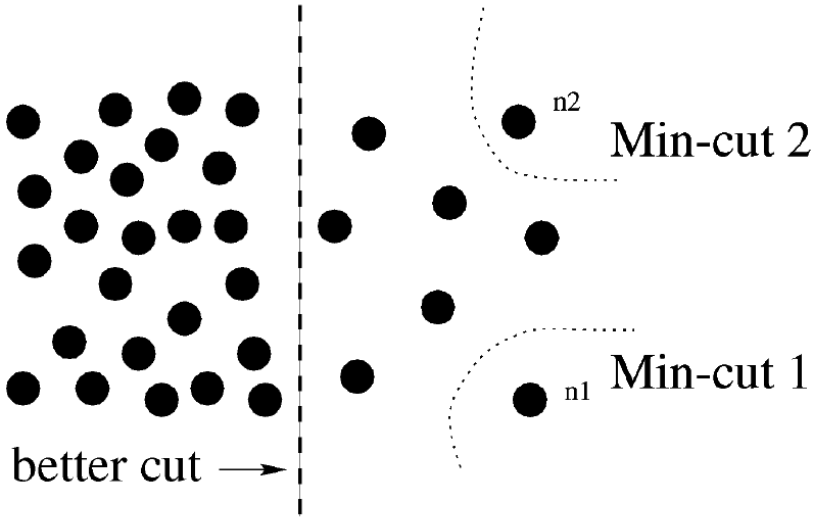
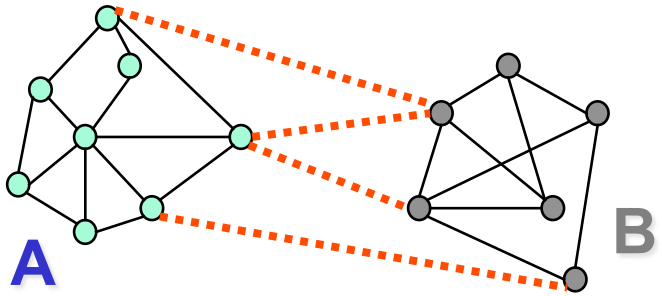


Fig. 1. A case where minimum cut gives a bad partition.

[Shi & Malik, 2000 PAMI]

Cuts in a graph: Normalized cut



Normalized Cut

- fix bias of Min Cut by **normalizing** for size of segments:

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

assoc(A,V) = sum of weights of all edges that touch A

- Ncut value small when we get two clusters with many edges with high weights, and few edges of low weight between them
- Approximate solution for minimizing the Ncut value : generalized eigenvalue problem.

J. Shi and J. Malik, Normalized Cuts and Image Segmentation, CVPR, 1997

Source: Steve Seitz

Eigenvectors and segments

- Simplest idea: we want a vector giving the association between each element and a cluster
- We want elements within this cluster to, on the whole, have strong affinity with one another
- We could maximize

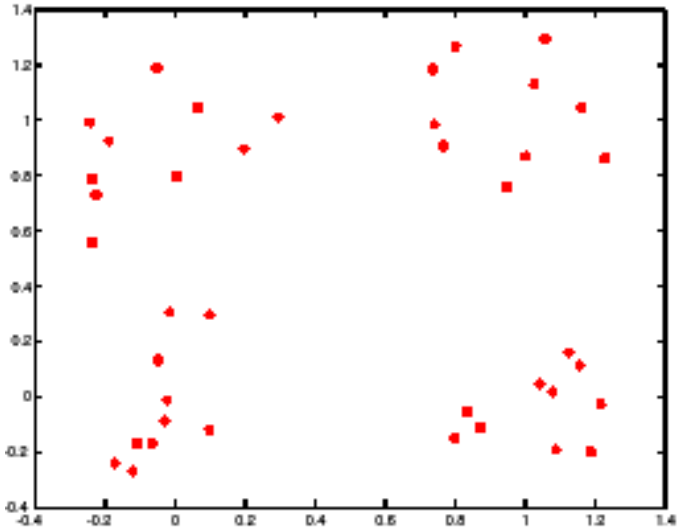
$$a^T A a$$

- But need the constraint

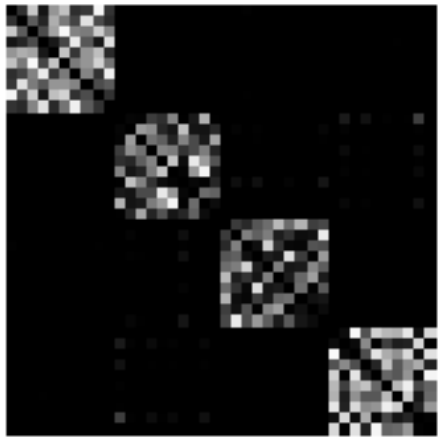
$$a^T a = 1$$

- This is an eigenvalue problem - choose the eigenvector of A with largest eigenvalue - single good cluster

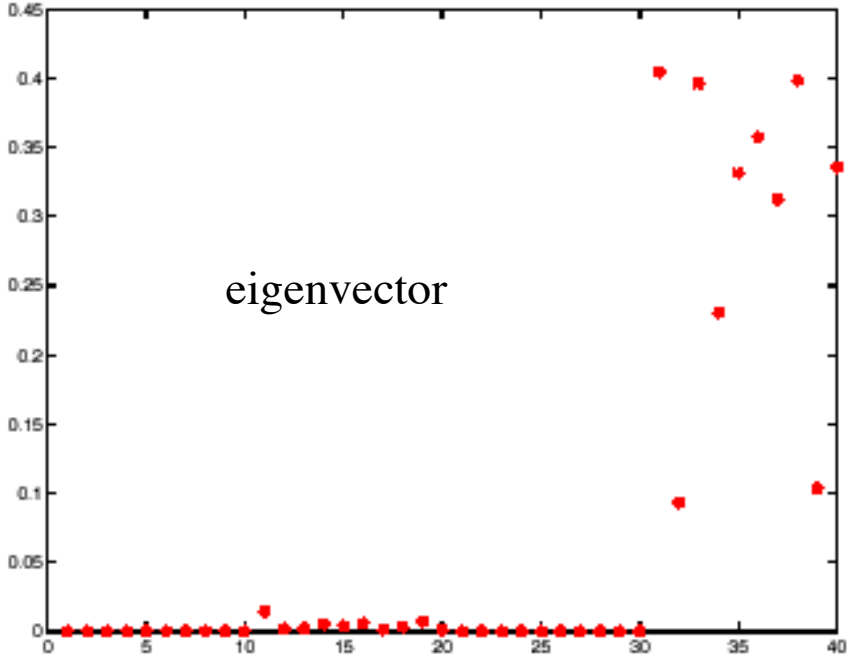
Example eigenvector



points



matrix



eigenvector

- Entries associated with the largest cluster are high
- Searching for additional clusters – look at
- Eigenvectors associated with second largest Eigenvalue etc

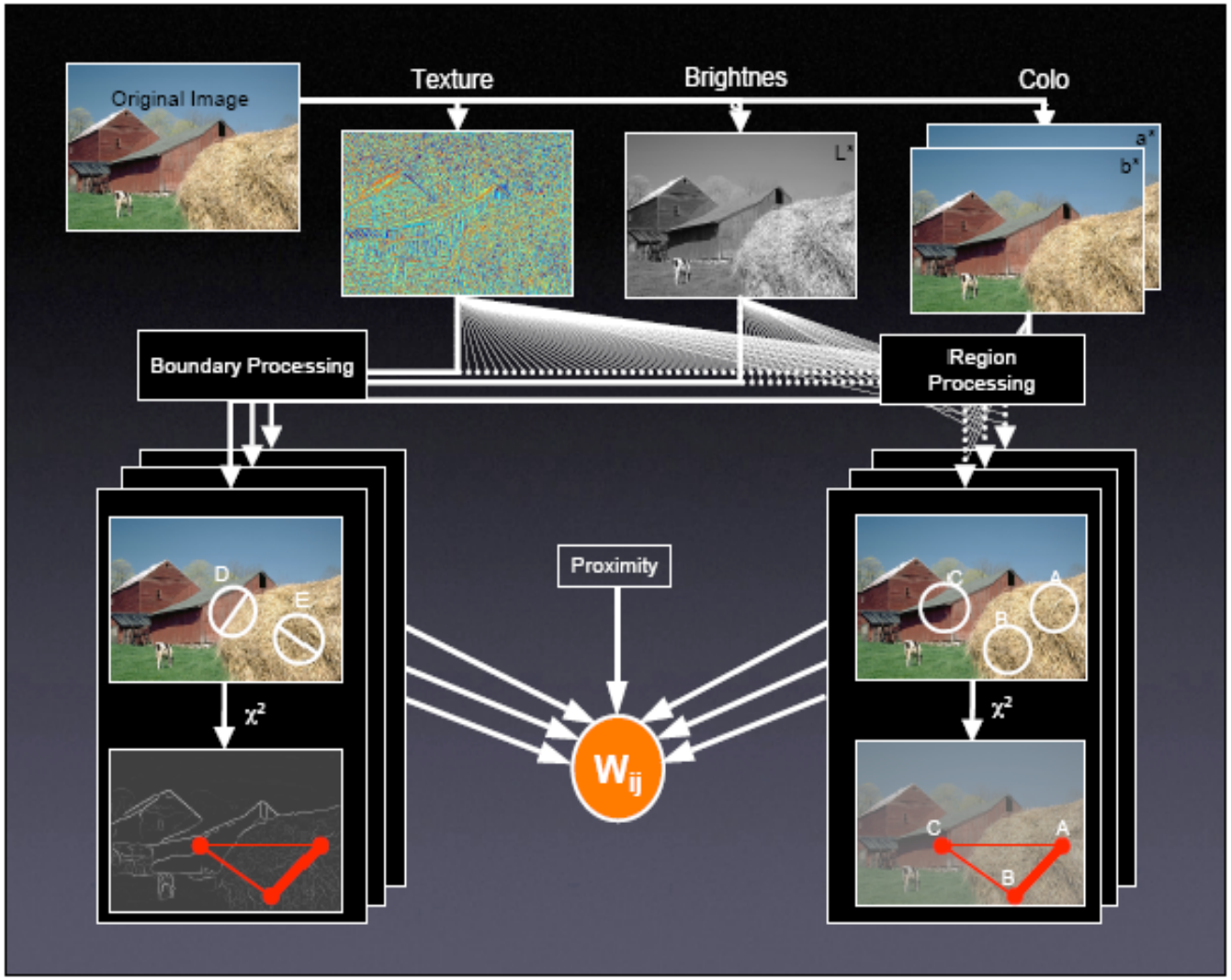
More than two segments

- Reasoning about other eigenvectors - consider that affinity matrix is block diagonal.
- Until there are sufficient clusters pick eigenvector associated with the largest eigenvalue, zero the elements which were clustered, threshold elements with large association weights - those will form a new cluster
- Keep going until there is sufficient number of clusters and all elements have been accounted for
- Spectral Clustering Techniques (A. Ng and M. Jordan)

- Problems - if the eigenvalues are similar - eigenvectors do not reveal the clusters

- Normalized cut - graph cut - alternative optimization criterion J. Shi and J. Malik – derivation on the board

Cue Combination



J. Kosecka

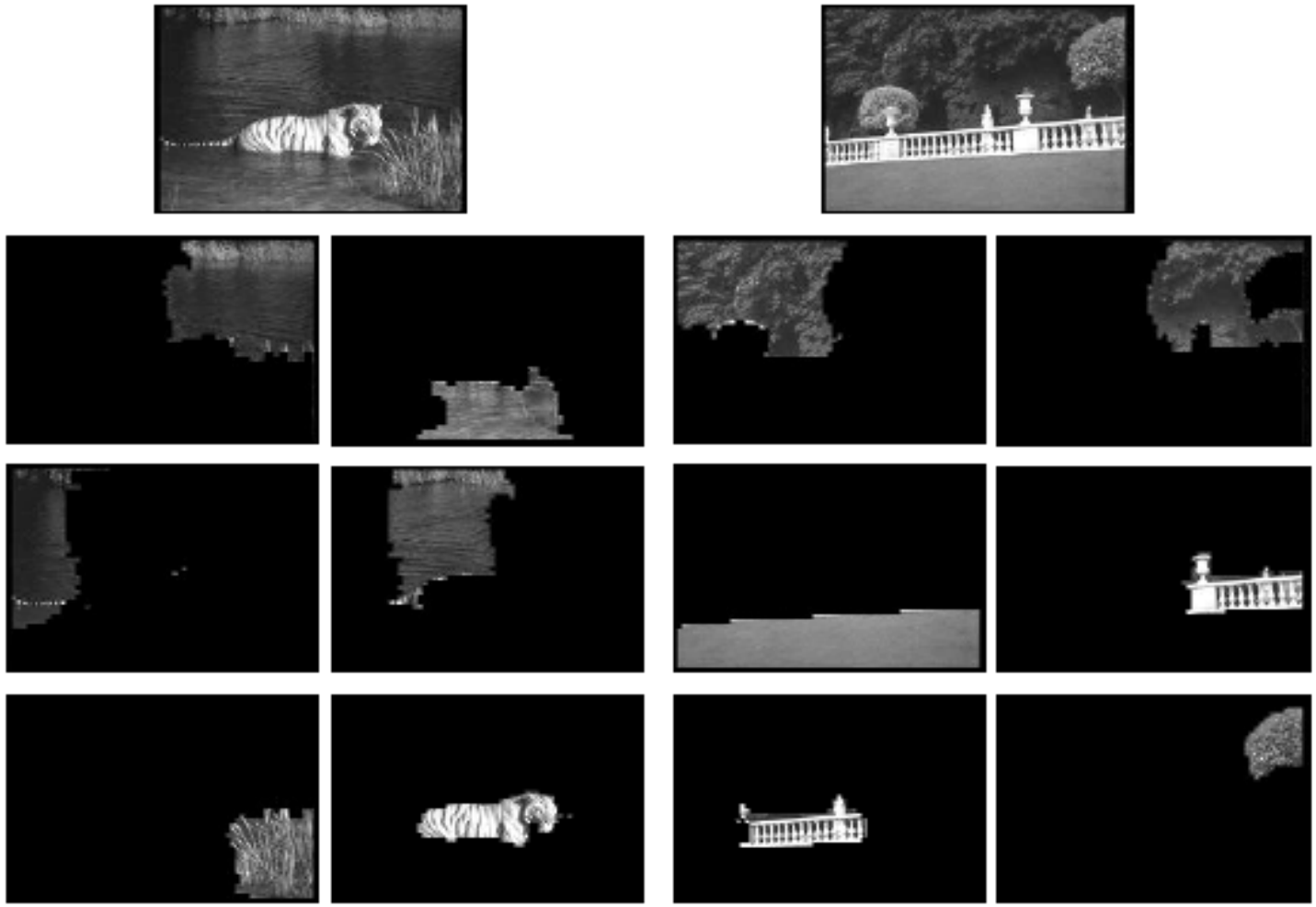


Figure from “Image and video segmentation: the normalised cut framework”, by Shi and Malik, copyright IEEE, 1998

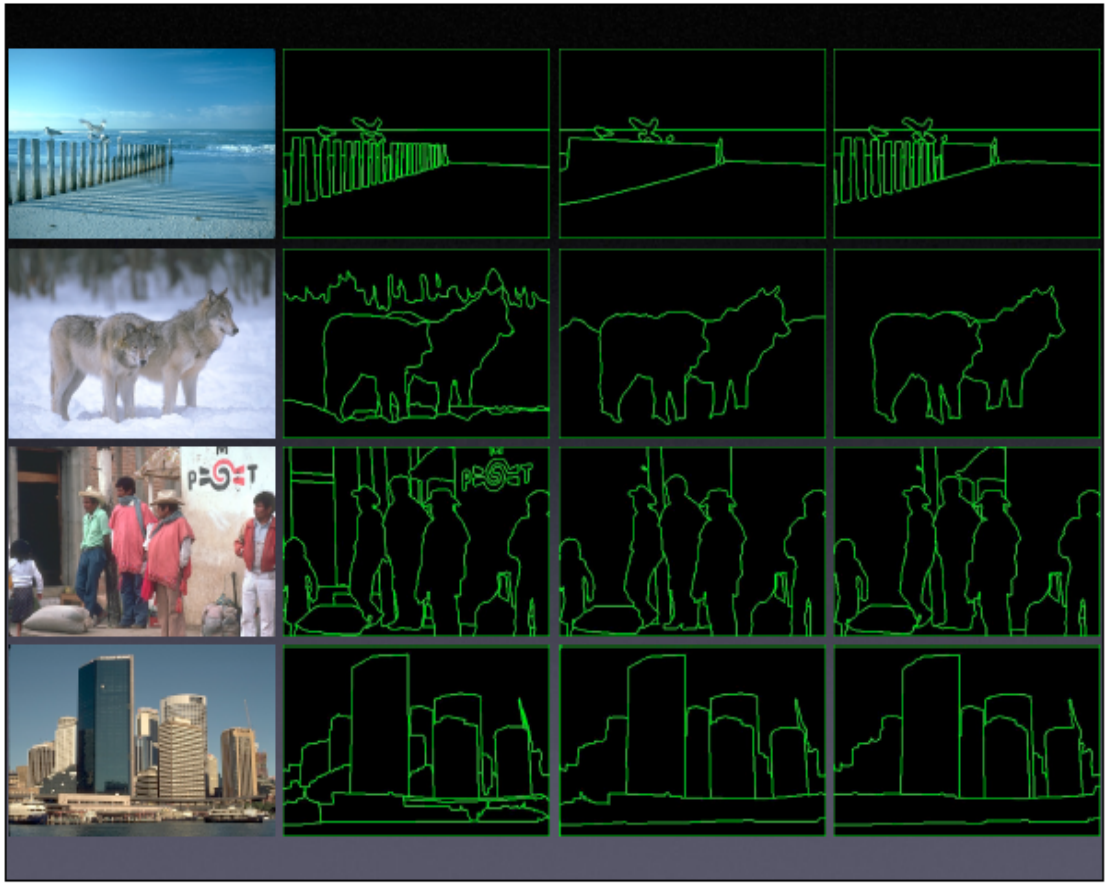
Example results

Results: Berkeley Segmentation Engine

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

Evaluation of Segmentation Algs.

- Comparison with human segmentations
- Berkeley Segmentation Datasets
- See details for metrics



Normalized cuts: pros and cons

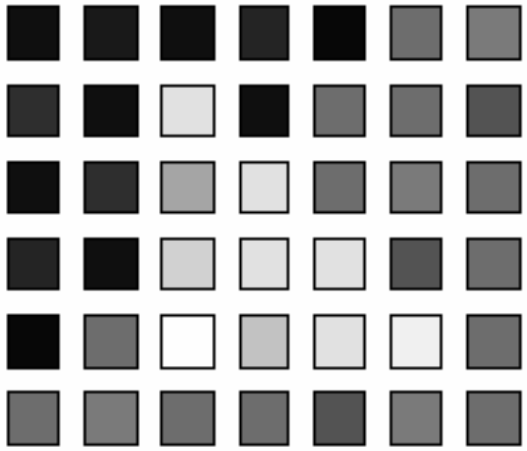
Pros:

- Generic framework, flexible to choice of function that computes weights ("affinities") between nodes
- Does not require model of the data distribution

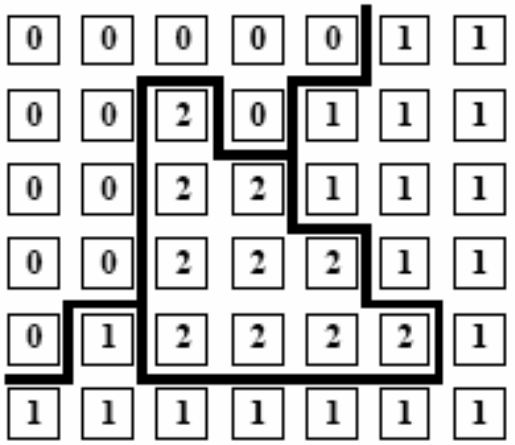
Cons:

- Time complexity can be high
  - Dense, highly connected graphs → many affinity computations
  - Solving eigenvalue problem
- Preference for balanced partitions

Graph Cuts, MRF's



(a) An image



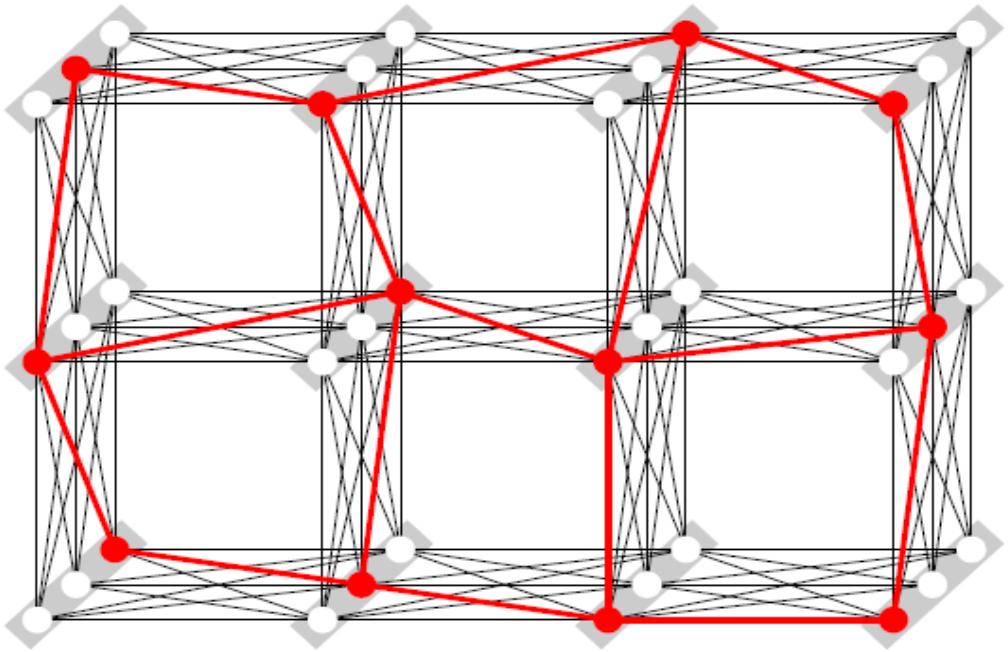
(b) A labeling

Pixel labeling problem - find such labels that the objective function is minimized
labels (stereo, object/no-object, original intensities (denoising)

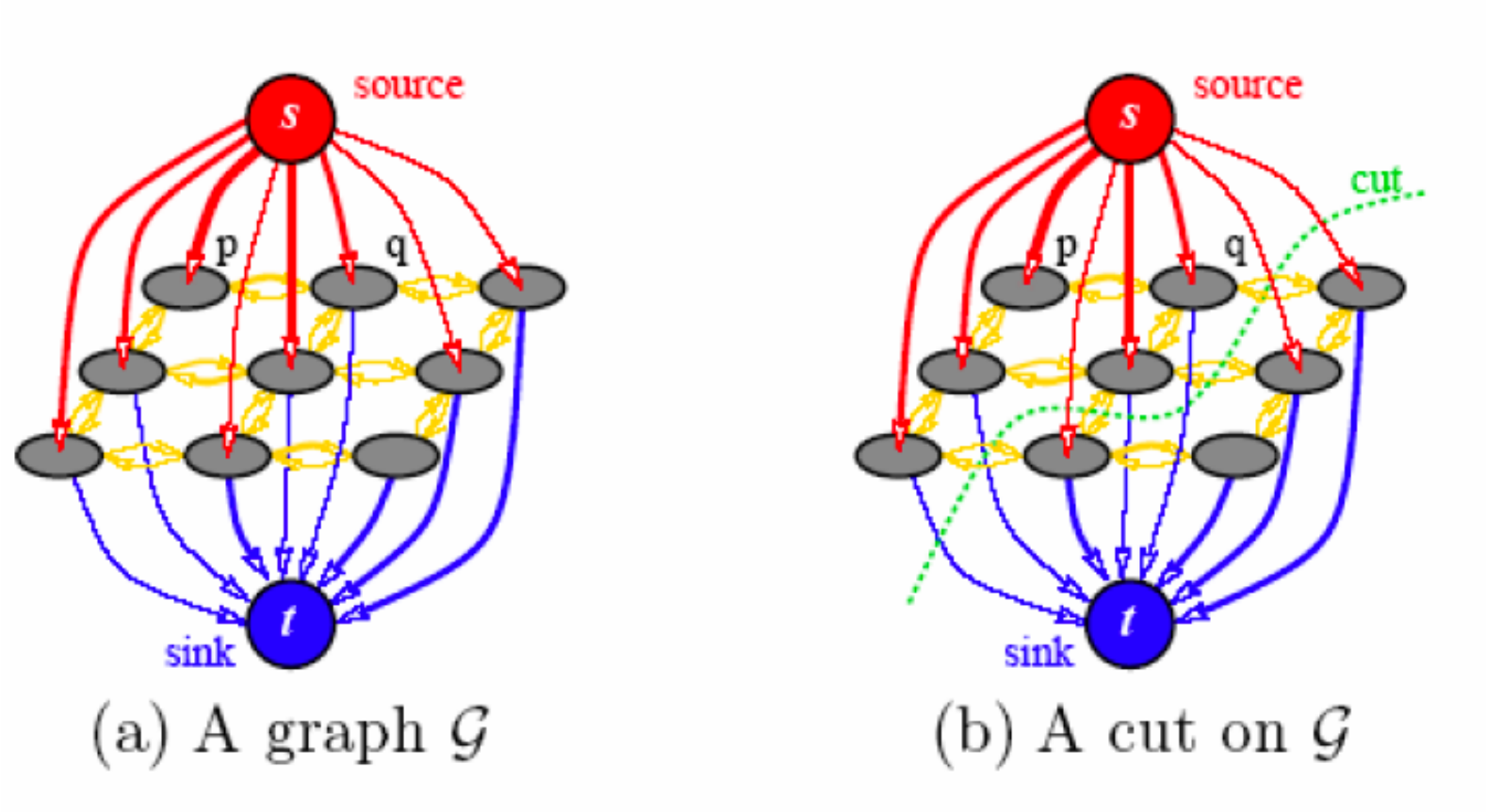
$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in N} V_{p,q}(f_p, f_q)$$

Toy example

- Graph 12 nodes; each node has 3 possible labels



Graph Cut

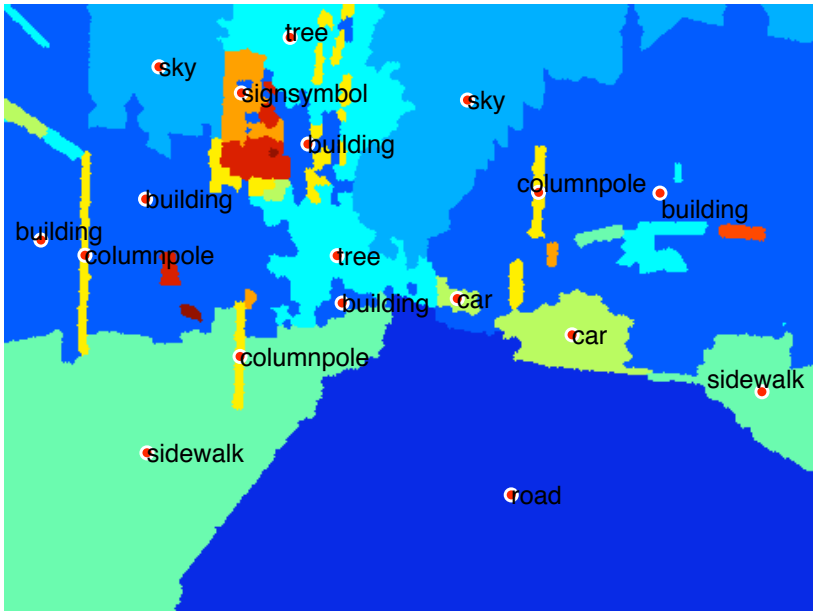


(a) A graph $\mathcal{G}$ (b) A cut on $\mathcal{G}$

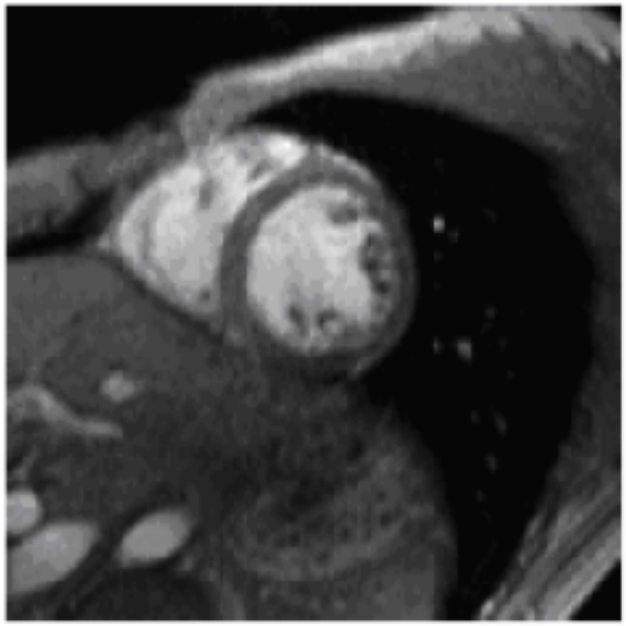
Formulated as minimum cost flow -
Network flow problem from Graph Theory
Kolmogorov, Boykov (et al)
efficient solvers available

Semantic Labelling problem

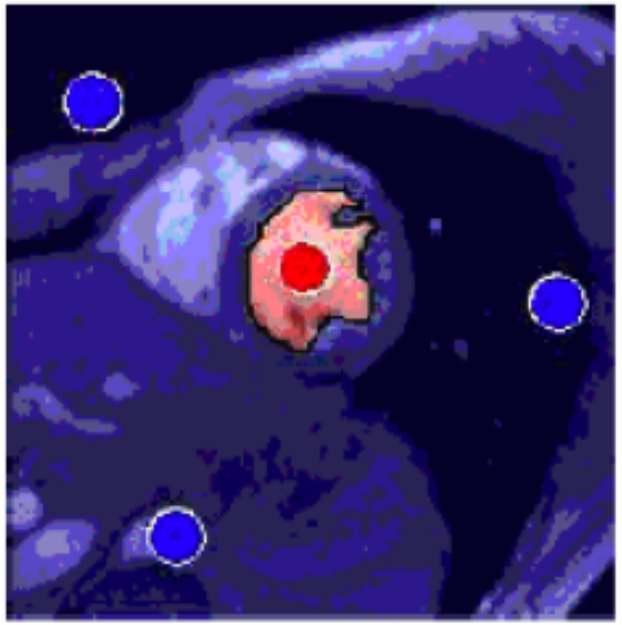
- Optimal semantic labelling problem
- Labels are hypotheses (object classes, disparities, flows)



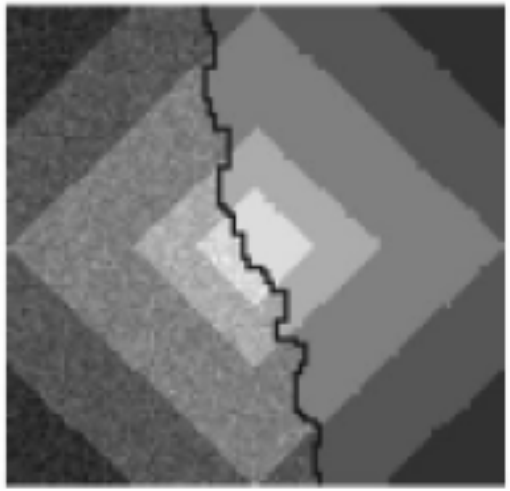
✓ available solvers: Kolmogorov PAMI'06, Werner PAMI'07



Original image



A minimum cut

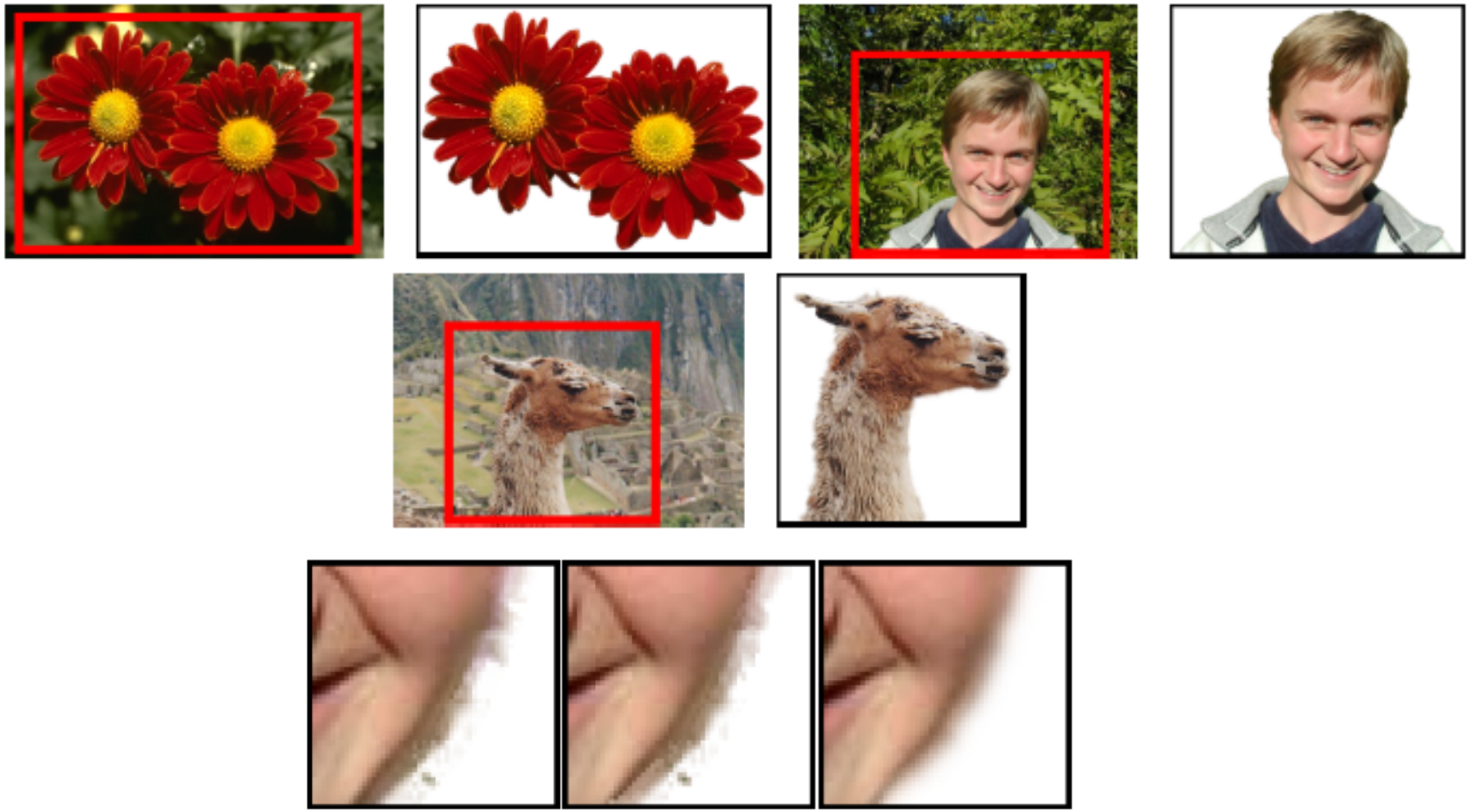


(a) *Diamond* restoration

(b) Original *Bell Quad*

(c) "Restored" *Bell Quad*

Interactive Foreground Segmentation



Interaction Foreground Segmentation: Grab Cuts
Rother, Kolmogorov, Blake, SIGRAPH 2005

Graph Based Segmentation

- Given representation of an image as a graph G(V,E)
- Partition the graph into C components, such that all the nodes within a component are similar
- Minimum weight spanning tree algorithm

1. Start with pixels as vertices, edge as similarity between neigbours, gradualy build connected components of the graph
2. Pick the lowest cost edge, the decision whether to join the two components or keep them apart depends on similarity within the components is smaller then similarity between the components. If the difference between the components is small then join them, else do nothing
3. Stopping criterion number of desired segments

Ref. P. Felzenswalb and D. Huttenlocher Efficient Graph Based Image Segmentation

MWST algorithm

- Idea – merge regions in decreasing order of the edges separating them
- Regions connected by weak edges merged first
- Each region characterized by internal difference

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

- Region difference – minimum weight connecting two regions

$$Diff(R_1, R_2) = \min_{\{v1 \in R1, v2 \in R2\}} w(e)$$

- Merge any two regions whos difference is smaller then the minimum internal difference of these regions

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

MWST graph based segmentation

- Example results

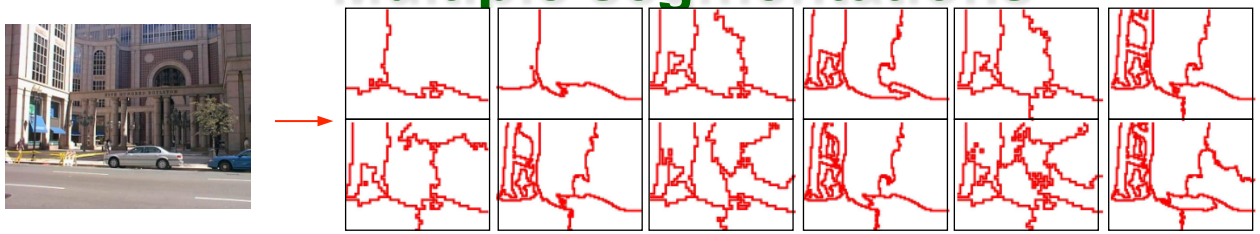



Advanced topics

- Output of segmentation – partitioning of the images into regions
- Intermediate representation for detection, recognition

Segments as primitives for recognition

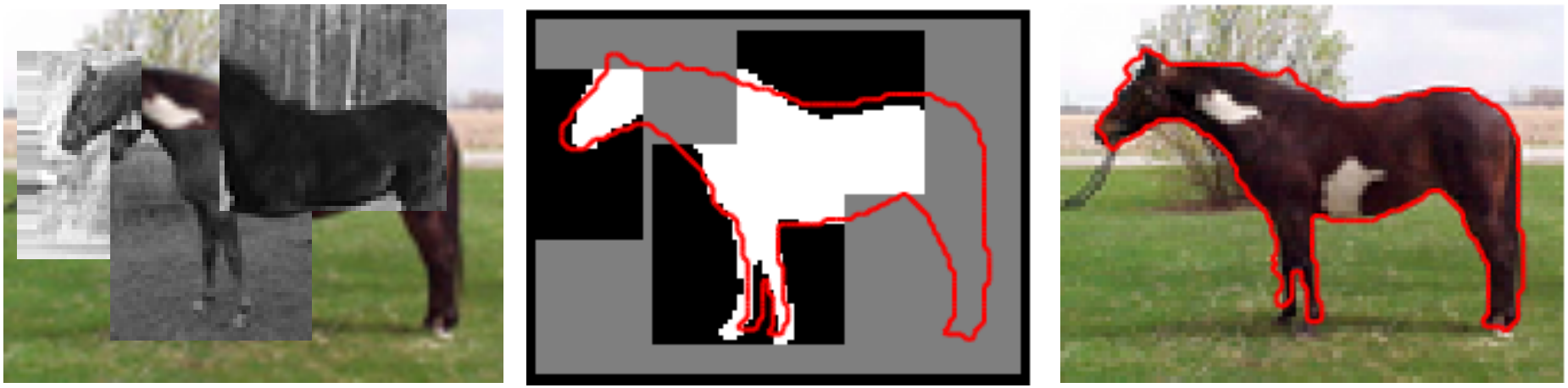
Multiple segmentations



- B. Russell et al., "Using Multiple Segmentations to Discover Objects and their Extent in Image Collections," CVPR 2006

Slide credit: Lana Lazebnik

Top-down segmentation

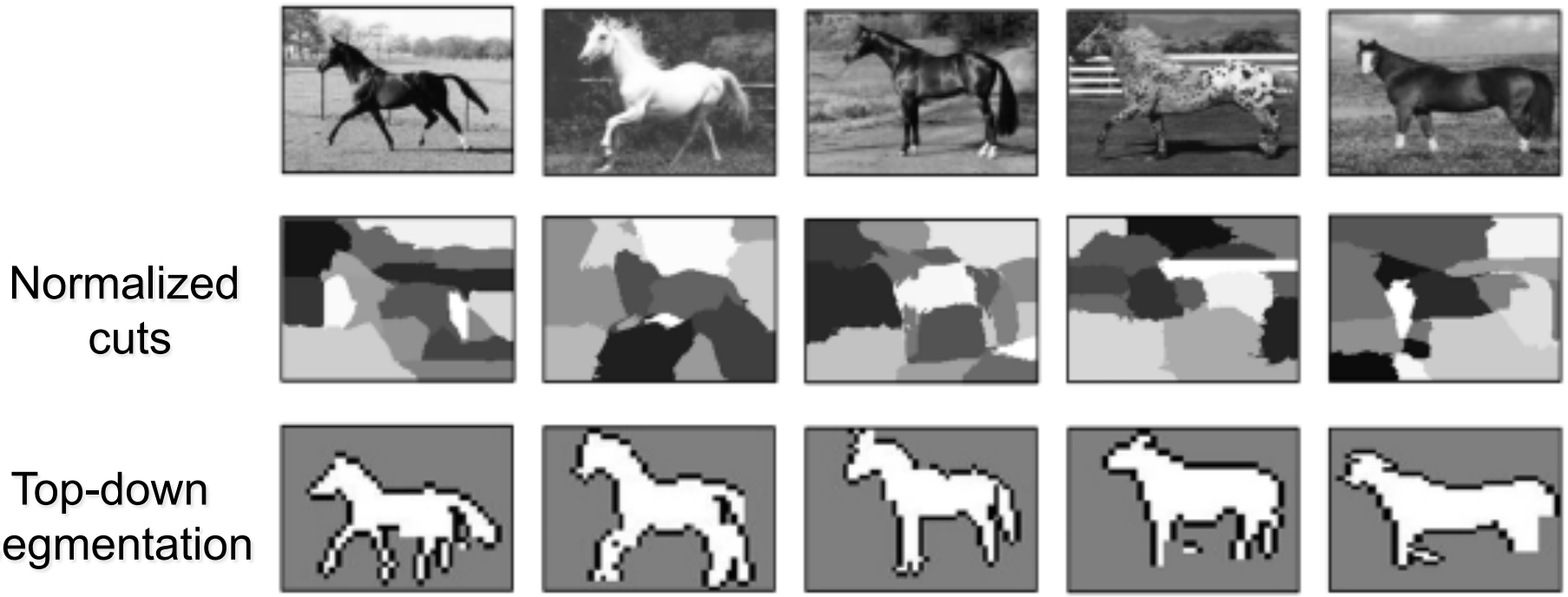


E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," ECCV 2002

A. Levin and Y. Weiss, "Learning to Combine Bottom-Up and Top-Down Segmentation," ECCV 2006.

Slide credit: Lana Lazebr

Top-down segmentation



- E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," ECCV 2002
- A. Levin and Y. Weiss, "Learning to Combine Bottom-Up and Top-Down Segmentation," ECCV 2006.

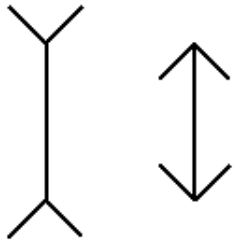
Slide credit: Lana Lazebnik

Technique: Shot Boundary Detection

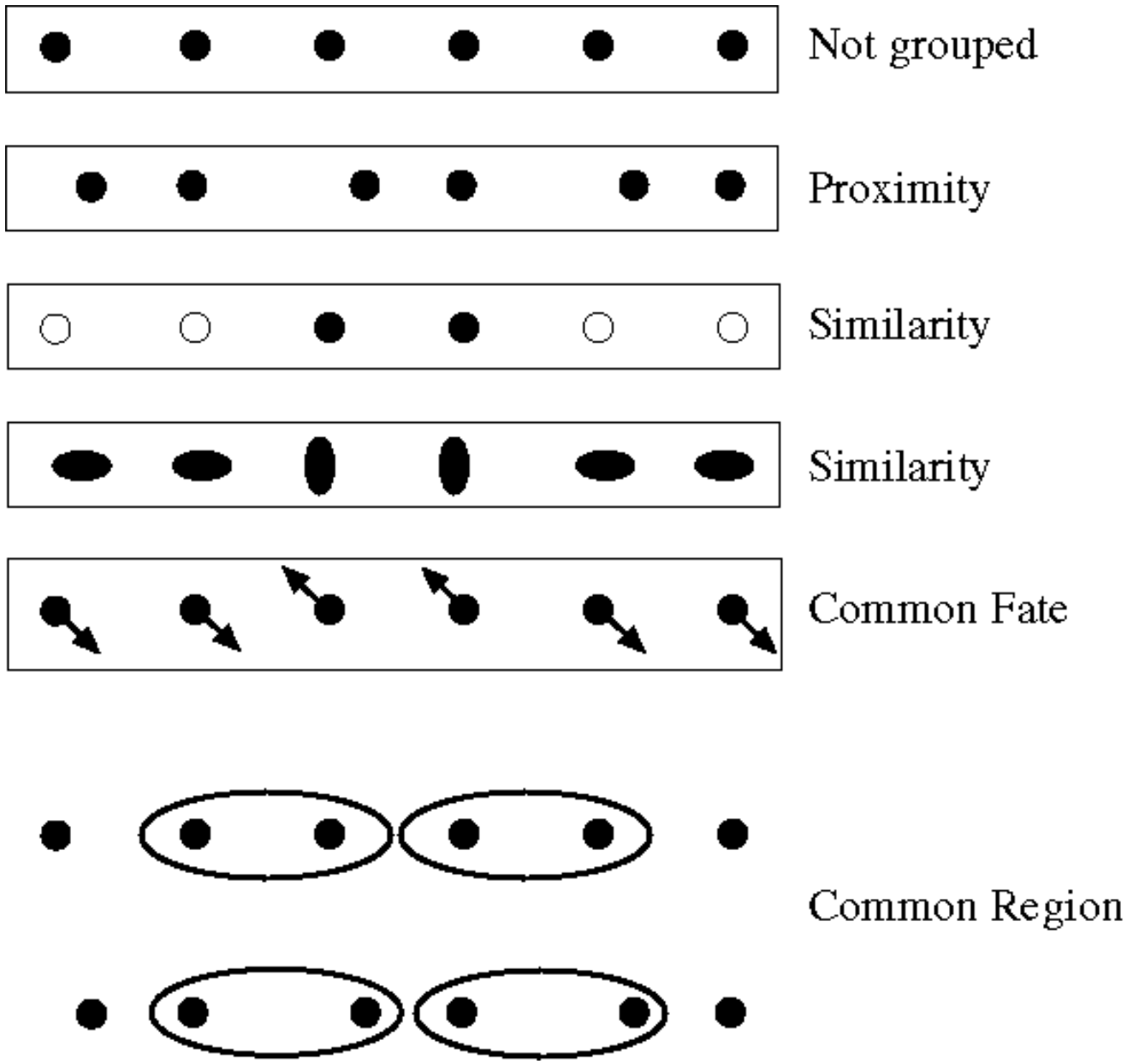
- Find the shots in a sequence of video
  - shot boundaries usually result in big differences between succeeding frames
- Strategy:
  - compute interframe distances
  - declare a boundary where these are big
- Possible distances
  - frame differences
  - histogram differences
  - block comparisons
  - edge differences
- Applications:
  - representation for movies, or video sequences
    - find shot boundaries
    - obtain “most representative” frame
  - supports search

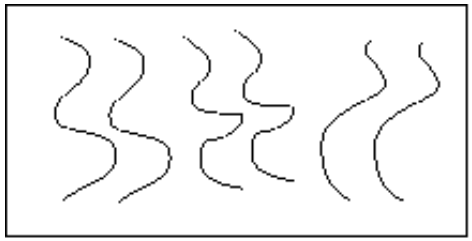
Grouping in humans

- Figure-ground discrimination
  - grouping can be seen in terms of allocating some elements to a figure, some to ground
  - impoverished theory

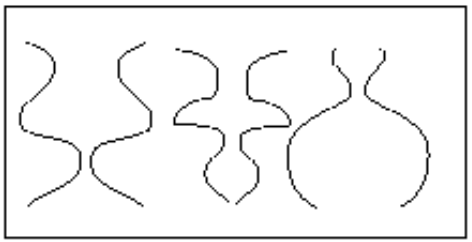


- Gestalt properties
  - elements in a collection of elements can have properties that result from relationships (Muller-Lyer effect)
    - Gestalt-qualitat
  - A series of factors affect whether elements should be grouped together
    - Gestalt factors

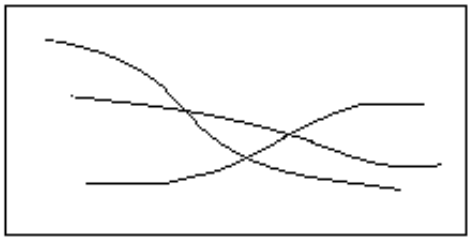

Not grouped
Proximity
Similarity
Similarity
Common Fate
Common Region



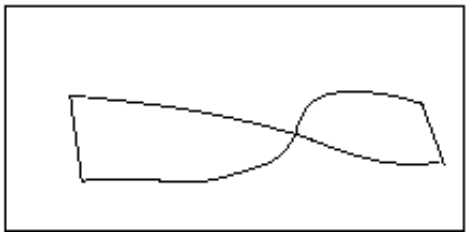
Parallelism



Symmetry



Continuity



Closure