Grouping and segmentation

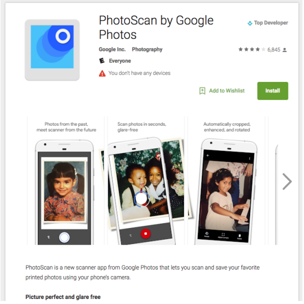
Subhransu Maji

CMPSCI 670: Computer Vision

November 17, 2016

Overview

- Grouping and segmentation
  - Goals of segmentation
  - Clustering using k-means
  - Choice of representation
  - Two techniques:
    - Mean shift algorithm
    - Graph cuts algorithm
  - Interactive segmentation



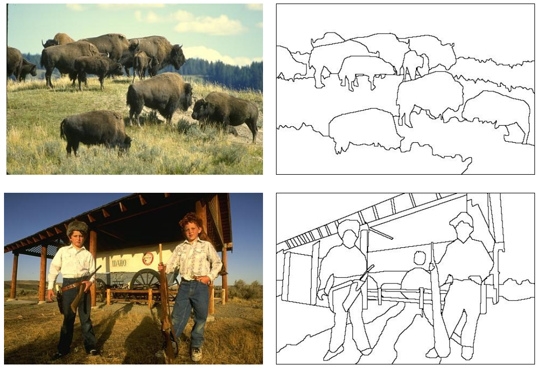
Photoscan by Google

- Take 4 photos — stitch them together + post-processing (remove glare, crop along the boundary, remove skew)

https://www.engadget.com/2016/11/15/google-photos-photoscan-app-editing-tools/

The goals of segmentation

- Separate image into coherent "objects"



- Another way of thinking about boundary detection

Source: Lana Lazebnik

Segmentation as clustering

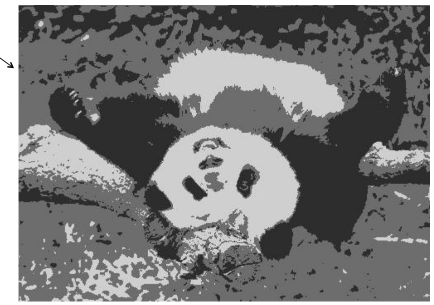
Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **intensity** similarity

Feature space: intensity value (1-d)

K=2

K=3



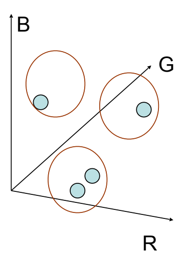
quantization of the feature space; segmentation label map

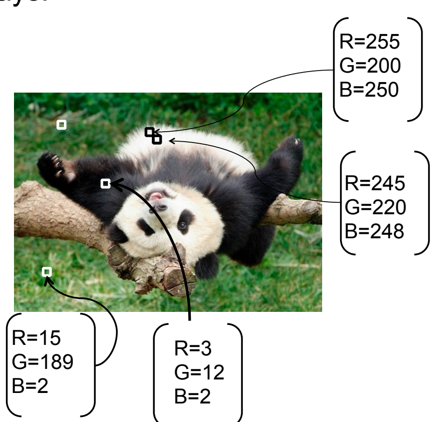
clustering using **kmeans**

Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **color** similarity





Feature space: color value (3-d)

Kristen Grauman

Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **intensity** similarity

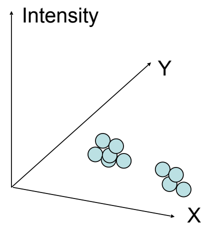
Clusters based on intensity similarity don't have to be spatially coherent.

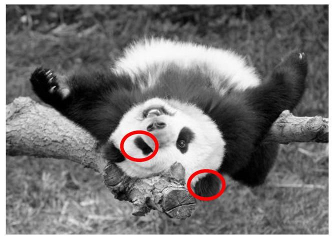
Kristen Grauman

Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **intensity+position** similarity



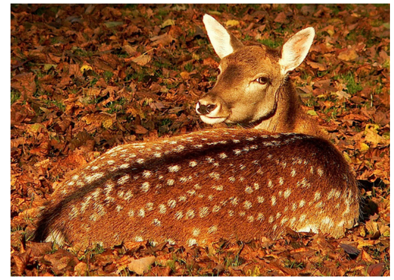


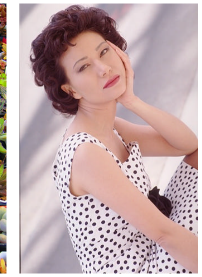
Both regions are black, but if we also include **position (x,y),** then we could group the two into distinct segments; way to encode both similarity & proximity.

Kristen Grauman

Segmentation as clustering

- Color, brightness, position alone are not enough to distinguish all regions…

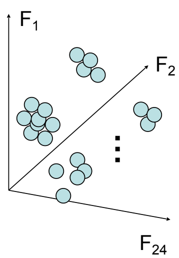


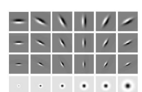


Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **filter response (texture)** similarity

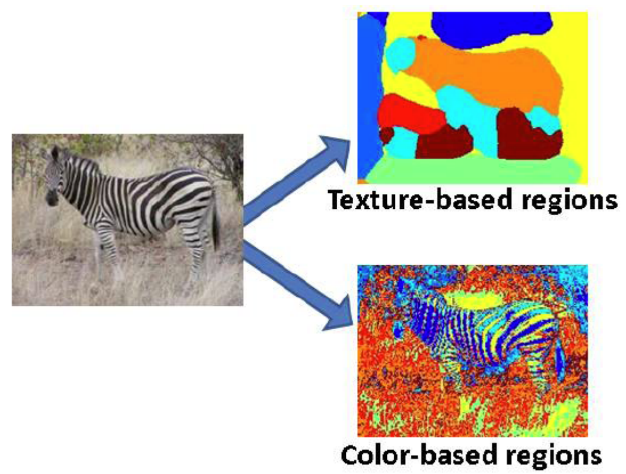




Filter bank of 24 filters

Feature space: filter bank responses (e.g., 24-d)

Image segmentation example

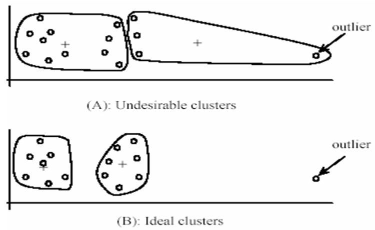


Kristen Grauman

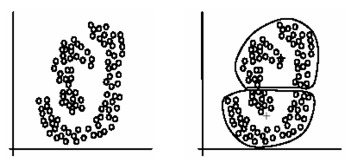
K-means: pros and cons

Pros
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error

Cons/issues
- Setting k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed



(A): Undesirable clusters
(B): Ideal clusters

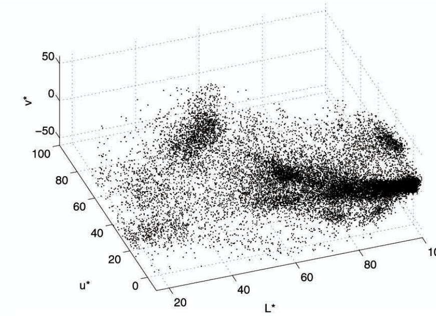

(A): Two natural clusters
(B): *k*-means clusters

Mean shift algorithm

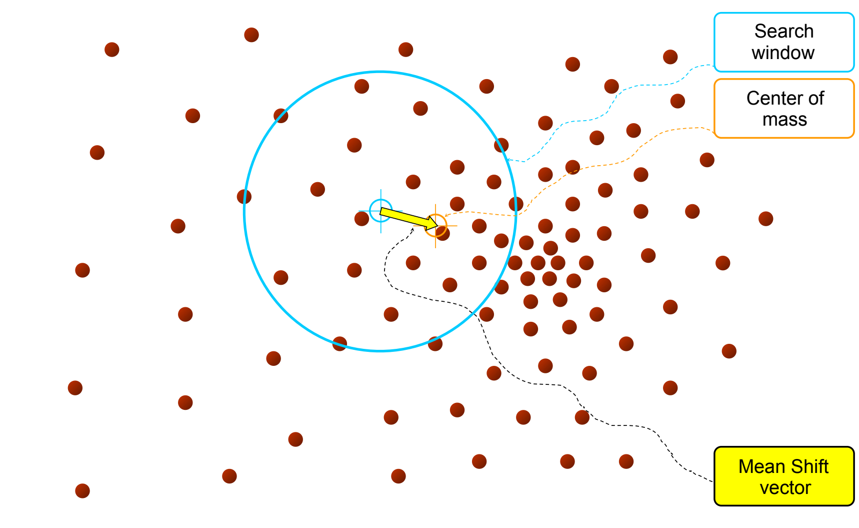
- The mean shift algorithm seeks *modes* or local maxima of density in the feature space

image

Feature space (L*u*v* color values)

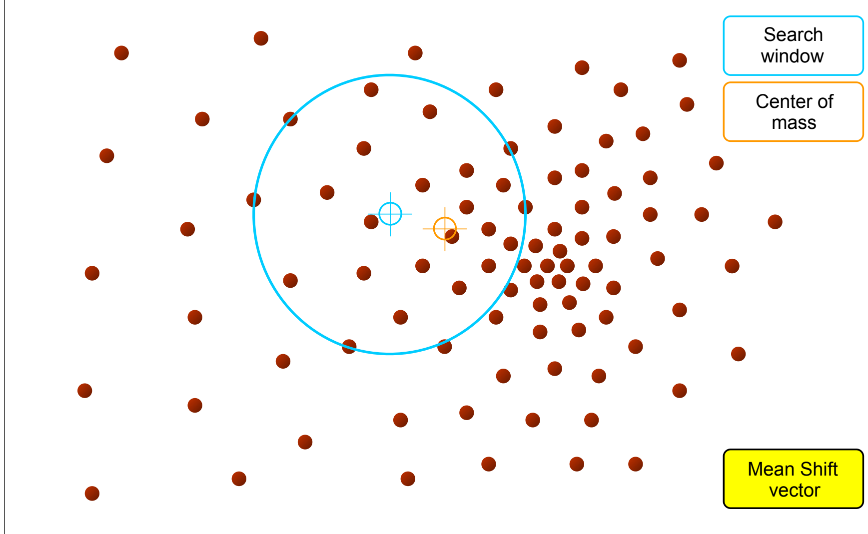


Mean shift



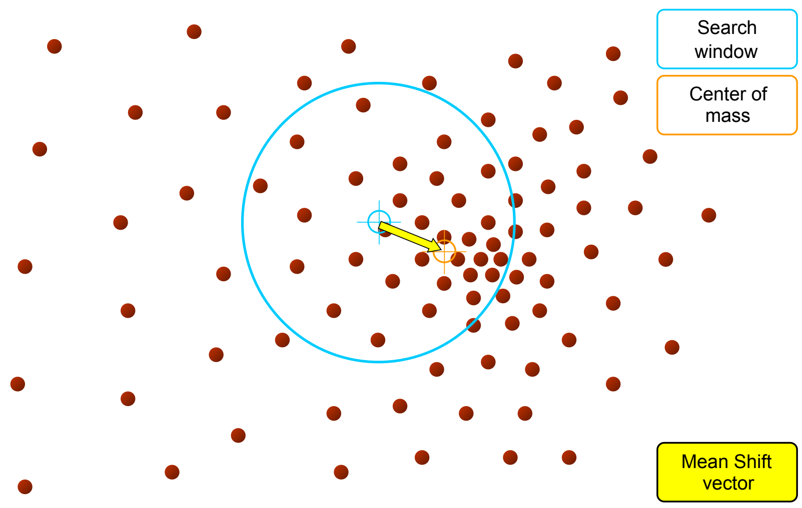
Slide by Y. Ukrainitz & B. Sarel

Mean shift



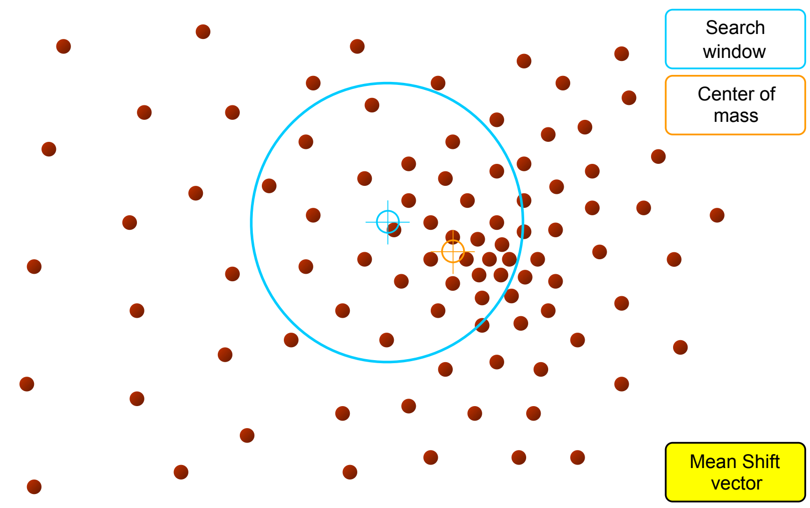
Slide by Y. Ukrainitz & B. Sarel

Mean shift



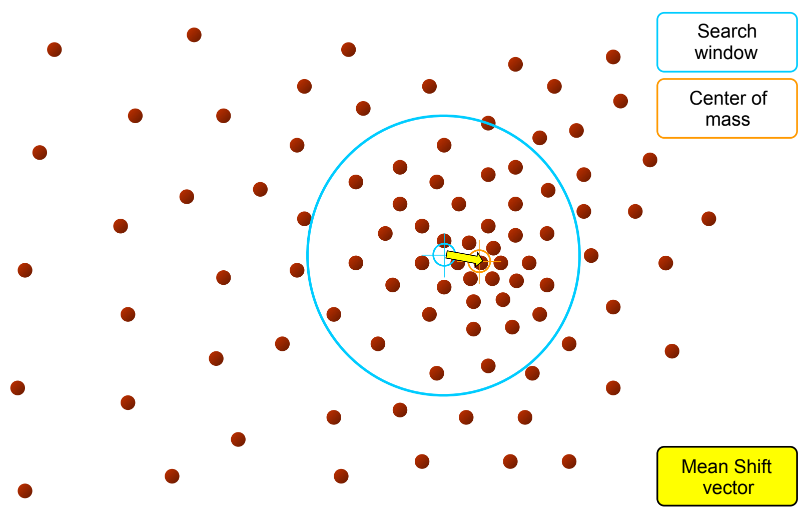
Slide by Y. Ukrainitz & B. Sarel

Mean shift



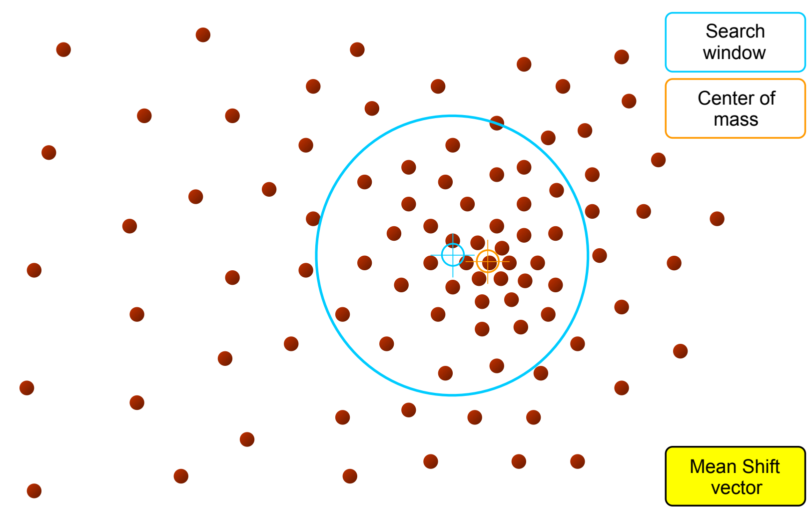
Slide by Y. Ukrainitz & B. Sarel

Mean shift



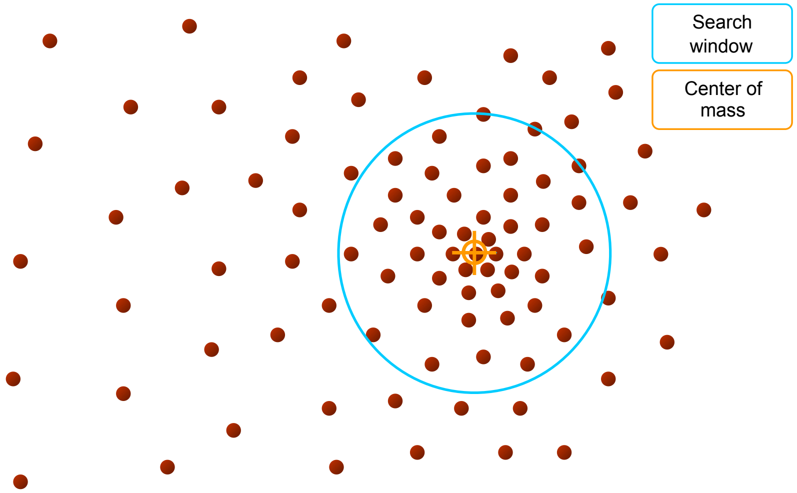
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



Computing the Mean Shift

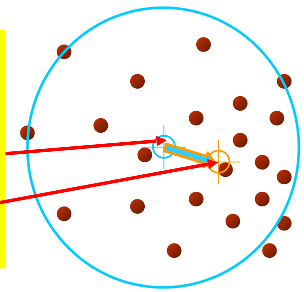
<u>Mean Shift procedure</u>:

For each point, repeat till convergence:

- Compute mean shift vector
- Translate the Kernel window by **m(x)**

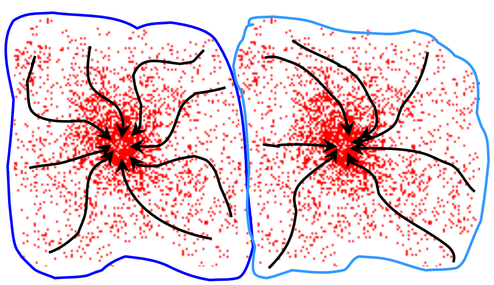
$$\exp\left(-\frac{||\mathbf{x}-\mathbf{x}_i||^2}{h}\right)$$

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)} - \boxed{\mathbf{x}}\right]$$



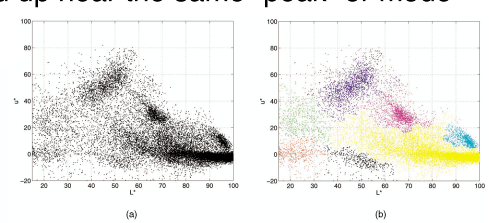
Mean shift clustering

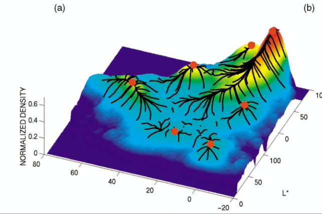
- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode



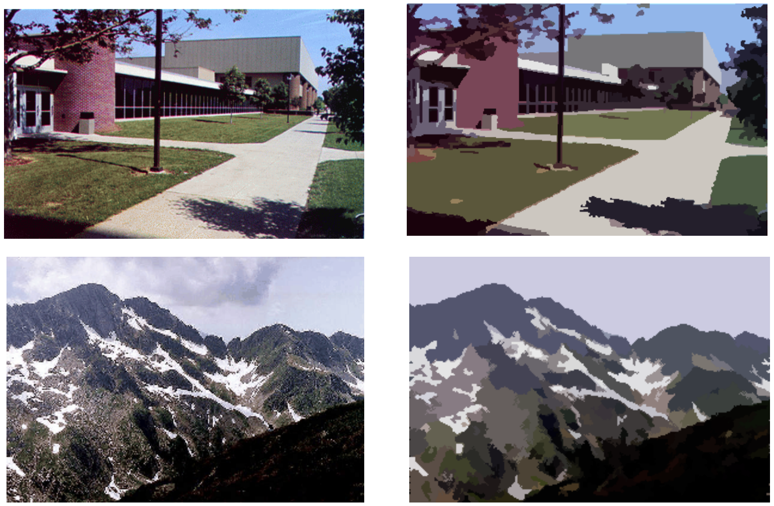
Mean shift clustering/segmentation

- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode





Mean shift segmentation results



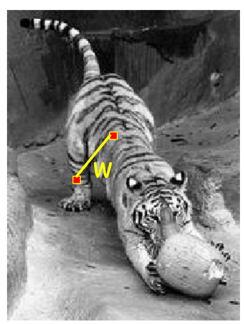
http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

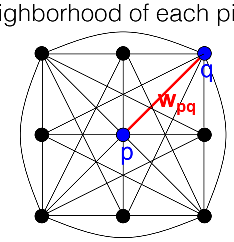
Mean shift clustering results

Images as graphs

Image graph

- node (vertex) for every pixel
- link between 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…)
- In practice only connect nodes within a neighborhood of each pixel



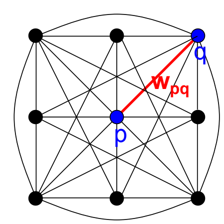


Source: Steve Seitz

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

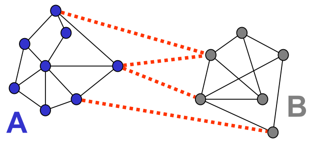


Source: Steve Seitz

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 (max flow/min cut algorithms)
- faster implementations exist that exploit the grid-structure of the graph (e.g., Boykov and Jolly 2001)

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.

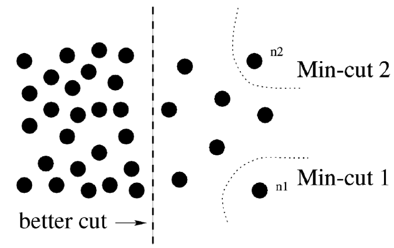
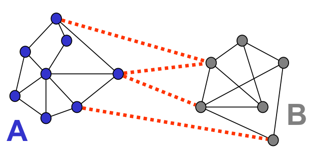


Fig. 1. A case where minimum cut gives a bad partition.

[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 is small when we get two clusters with many edges with high weights, and few edges of low weight between them
- NP-hard to compute, but 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

Example results

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

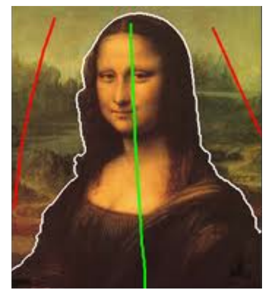
Cons:

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

Kristen Grauman

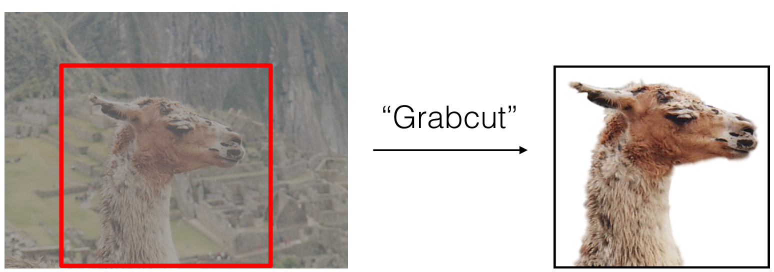
Image segmentation with priors

- Often we want to incorporate prior information
  - User input in interactive applications
  - Shape priors, e.g., we want a round object

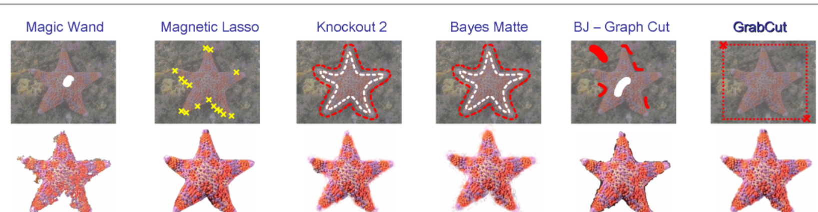


Constrains the set of possible segmentations

Image segmentation with priors

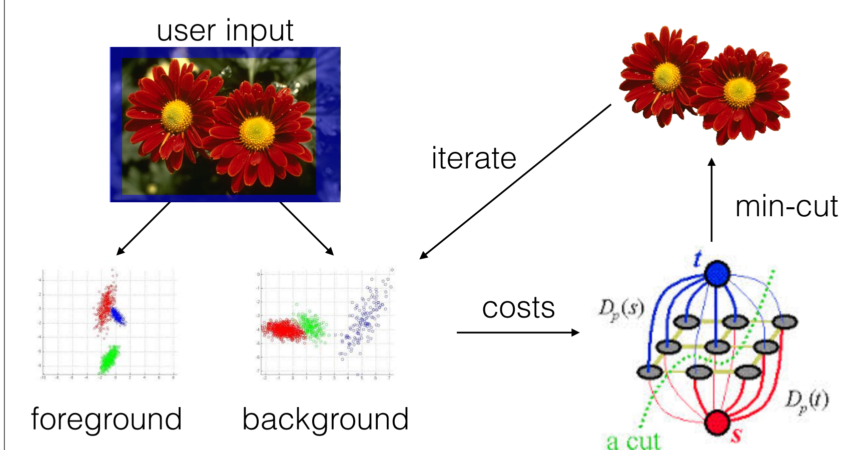


C. Rother, V. Kolmogorov, A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Transactions on Graphics (SIGGRAPH'04), 2004*



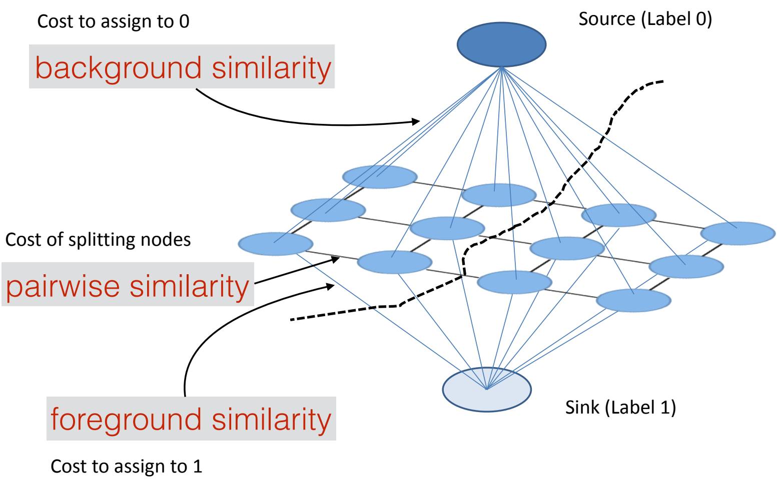
“Grabcut" algorithm

Construct a color model of foreground and background

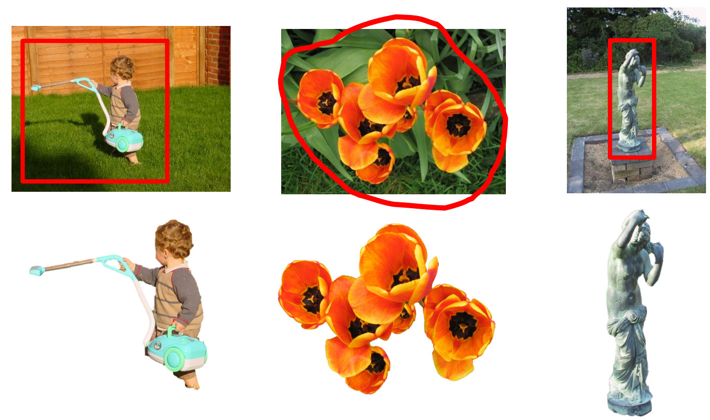


Gaussian mixture model (5-8 components)
(probabilistic version of k-means)

Solution using min-cut



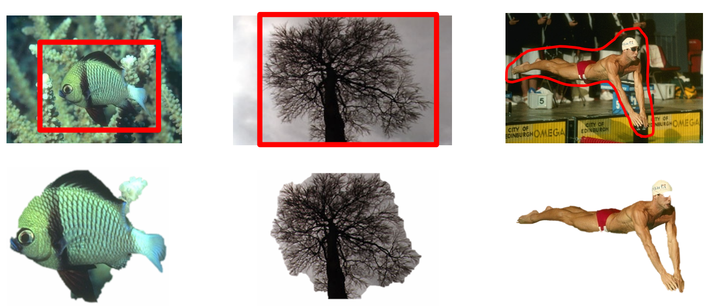
Moderately straightforward examples



… GrabCut completes automatically

Difficult examples

Camouflage & Low Contrast | **Fine structure** | **Harder Case**



Grabcut algorithm

- Pros
  - Globally optimal solution using min-cut/max-flow algorithms
  - Fast algorithms exist for grid-graphs
  - Works well in many cases
- Cons
  - Color similarity does not work when contrast is low, or when the image has fine-structures

Further thoughts and readings ..

- Chapter 5, Richard Szeliski's book
- **Berkeley segmentation database and benchmark**
  - Also read about the Berkeley boundary detector
- **http://www.cis.upenn.edu/~jshi/GraphTutorial/**
- Image segmentation via. graph cuts
  - Boykov and Jolly, **Interactive graph cuts for optimal boundary & region segmentation of objects in ND images**, ICCV 2001
- Normalized cuts for image segmentation (Shi and Malik)
  - **http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf**
- Biased normalized cuts
  - **http://people.cs.umass.edu/~smaji/projects/biasedNcuts/index.html**