Image Segmentation

J. Kosecka, CS 482

Some slides
From Computer Vision book D. Forsythe, J. Ponce

Segmentation as Graph Partitioning

- (Shi & Malik "97)
- Idea – each pixel in the image is a node in the graph
- Arcs represent similarities between adjacent pixels
- 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.

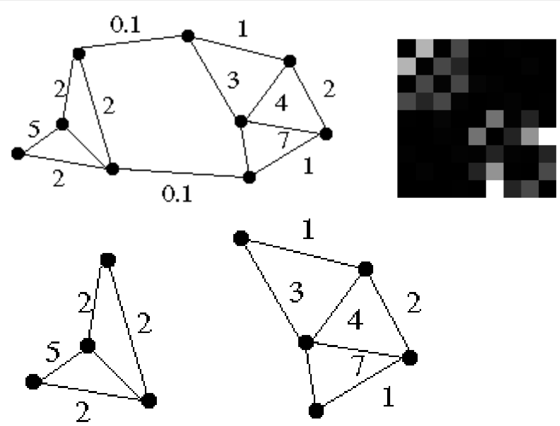
- See textbook for detailed description the algorithm.

J. Kosecka, cs482

Graph theoretic clustering

- Represent tokens using a weighted graph.
  - affinity matrix
- Cut up this graph to get subgraphs with strong interior links

J. Kosecka, cs482



J. Kosecka, cs482

Measuring Affinity

Intensity

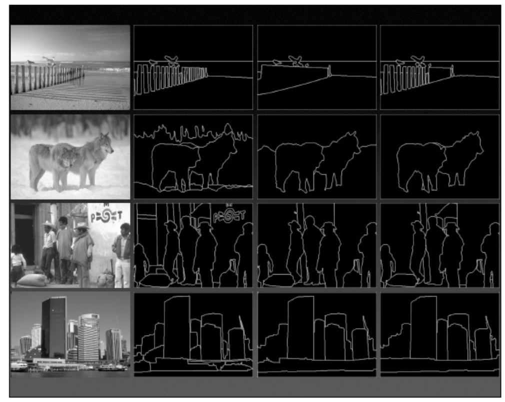
$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_i^2}\right)\left(\|I(x)-I(y)\|^2\right)\right\}$$

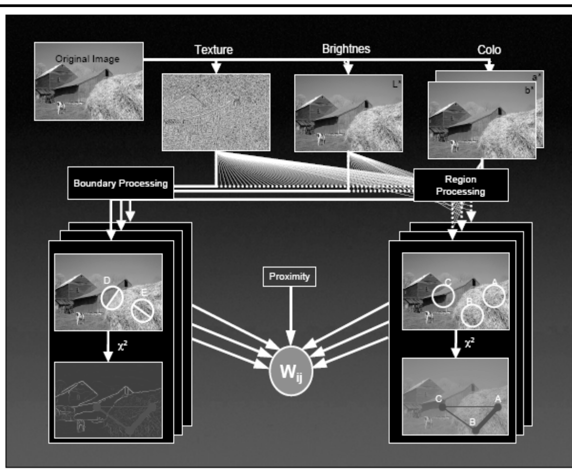
Distance

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

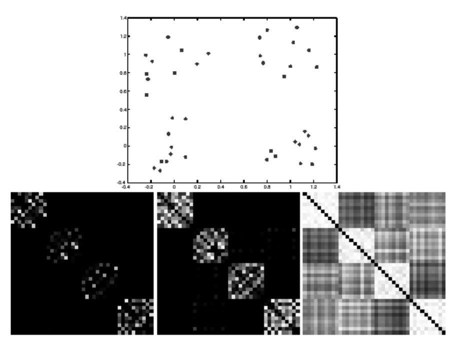
Texture

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





Scale affects affinity



Eigenvectors and cuts

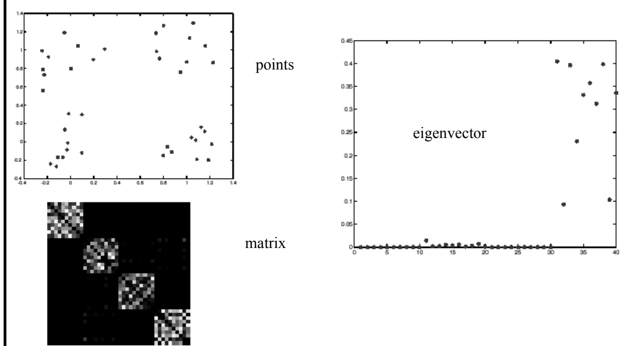
- Simplest idea: we want a vector a giving the association between each element and a cluster
- We want elements within this cluster to, on the whole, have strong affinity with one another
- We could maximize $w^t A w$

(association of element i with cluster n) X (affinity between i and j) X (association of element j with cluster n)

- But need the constraint $w^t w = 1$
- This is an eigenvalue problem - choose the eigenvector of A with largest eigenvalue

$$w^T A w + \lambda(w^t w - 1) \qquad Aw = \lambda w$$

Example eigenvector



More than two segments

- Two options
  - Recursively split each side to get a tree, continuing till the eigenvalues are too small
  - Use the other eigenvectors

  - Idea – eigenvectors of block-diagonal matrices Have non-zero entries for the elements along diagonal – rest padded with zeros –
  - If there are c-significant clusters, then c eigenvectors corresponding to them represent each cluster

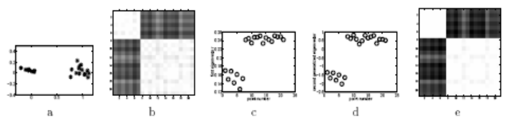
Problems

- When the eigenvalues are similar the eigenvectors will not split the clusters (if there are repeated eigenvalues – any vector which is a linear combination of the two eigenvectors is also eigenvector)

Alternative method – Normalized cut

Segmentation

Normalized cut approach – J. Shi, J. Malik (see textbook)

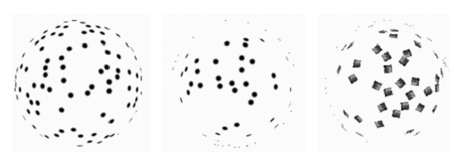


Segmentation and Grouping

- Motivation: not information is evidence
- Obtain a compact representation from an image/motion sequence/set of tokens
- Should support application
- Broad theory is absent at present
- Grouping (or clustering)
  - collect together tokens that "belong together"
- Fitting
  - associate a model with tokens
  - issues
    - which model?
    - which token goes to which element?
    - how many elements in the model?

General ideas

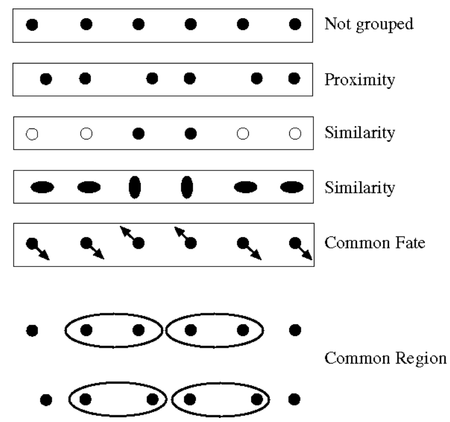
- tokens
  - whatever we need to group (pixels, points, surface elements, etc., etc.)
- top down segmentation
  - tokens belong together because they lie on the same object
- bottom up segmentation
  - tokens belong together because they are locally coherent
- These two are not mutually exclusive

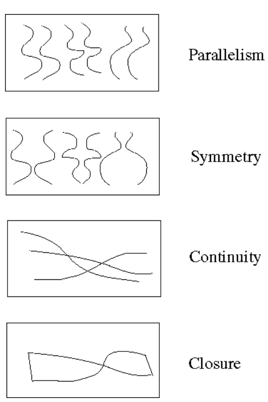


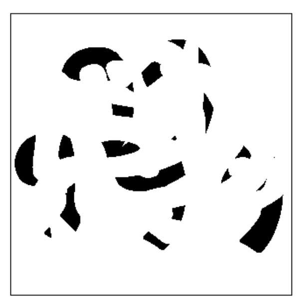
Why do these tokens belong together?

Basic ideas of 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)
    - gestaltqualitat
  - A series of factors affect whether elements should be grouped together
    - Gestalt factors







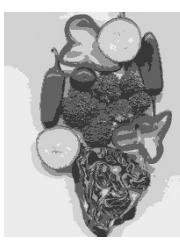
Segmentation by K-Means

- Choose a fixed number of clusters
- Choose cluster centers and point-cluster allocations to minimize error
- can't do this by search, because there are too many possible allocations.
- Algorithm
  - fix cluster centers; allocate points to closest cluster
  - fix allocation; compute best cluster centers
- x could be any set of features for which we can compute a distance (careful about scaling)

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

Image

Clusters on intensity



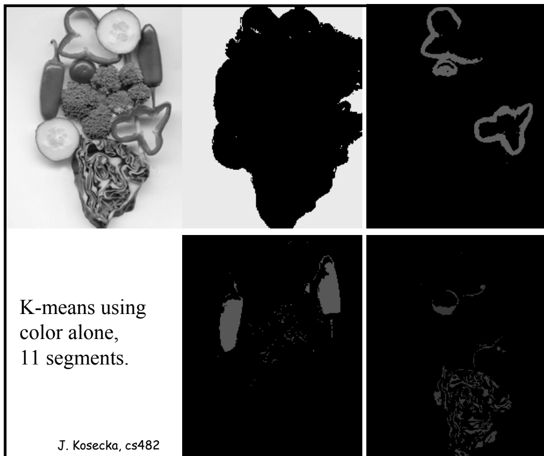
Clusters on color

K-means clustering using intensity alone and color alone

Image

Clusters on color

K-means using color alone, 11 segments



K-means using
color alone,
11 segments.

K-means using colour and
position, 20 segments

Segmentation with EM

- There are n – pixels and g groups – compute how likely
  is a pixel belonging to group and also what are the parameters of the group
- Probabilistic K-means clustering
- E.g. Use of texture and color cues

Segmentation with EM

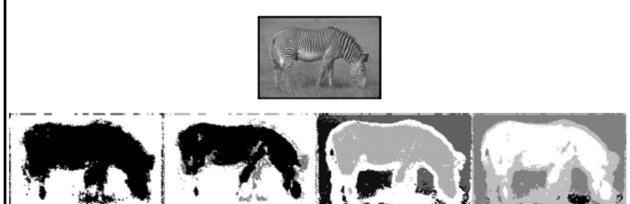


Figure from “Color and Texture Based Image Segmentation Using EM and Its Application to Content Based Image Retrieval”,S.J. Belongie et al., Proc. Int. Conf. Computer Vision, 1998, c1998, IEEE

Motion segmentation with EM

- Model image pair (or video sequence) as consisting of regions of parametric motion
  - affine motion is popular

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

- Now we need to
  - determine which pixels belong to which region
  - estimate parameters
- Likelihood
  - assume

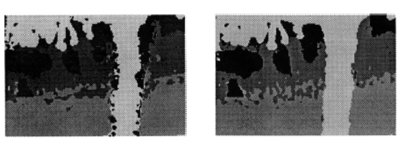
$$I(x,y,t) = I\left(x + v_x, y + v_y, t+1\right) + noise$$

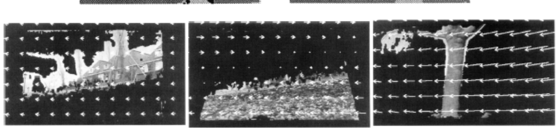
- Straightforward missing variable problem, rest is calculation

Three frames from the MPEG "flower garden" sequence

Figure from "Representing Images with layers,", by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE

Grey level shows region no. with highest probability





Segments and motion fields associated with them

Figure from "Representing Images with layers,", by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE

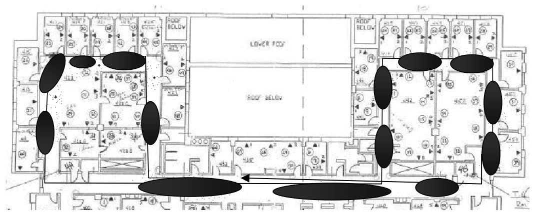
Segmentation of Time Varying Images (and tracking)

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

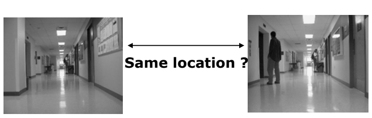
J. Kosecka, cs482

Vision Based Localization
Buidling Model of the environment from video



Global Topology and Local Geometry of the Environment

- Impose some discrete structure on the space of continuous visual observations
- Develop methods applicable to large scale environments
- Associate semantic labels with individual locations (corridor, hallway, office)

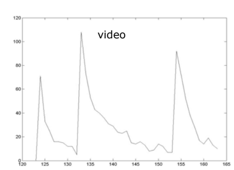


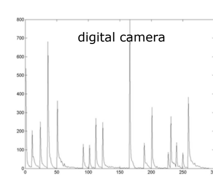
Issues for Vision Based Localization
- Representation of individual locations
- Learning neighborhood relationships between locations

J. Kosecka, cs482

Partitioning the video sequence

- Transitions between individual locations determined during exploration
- Location sub-sequence across which frames can be matched successfully
- (matching cri
- Location Representation - set of representative views





J. Kosecka, cs482

# of frames vs. # of matched features 1st – i-th view

Technique: Background Subtraction

- If we know what the background looks like, it is easy to identify “interesting bits”
- Applications
  - Person in an office
  - Tracking cars on a road
  - surveillance
- Approach:
  - use a moving average to estimate background image
  - subtract from current frame
  - large absolute values are interesting pixels
    - trick: use morphological operations to clean up pixels

Image Differencing

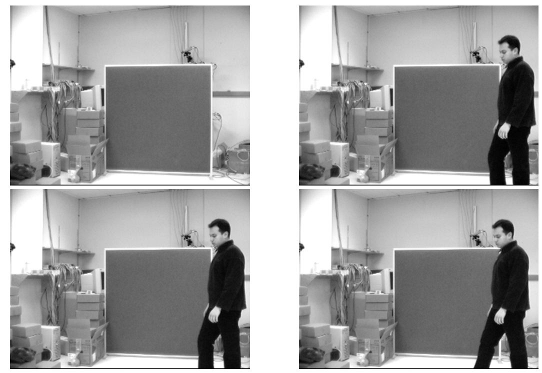
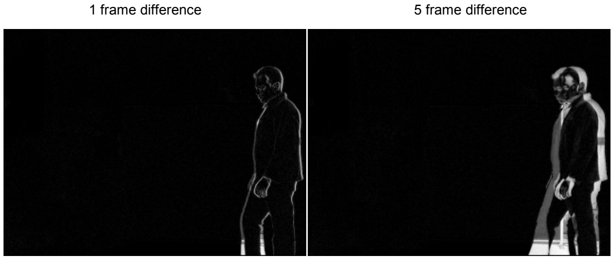


Image Differencing: Results

1 frame difference

5 frame difference



Motion detection

- Background subtraction
  - create an image of the stationary background by averaging a long sequence
    - for any pixel, most measurements will be from the background
    - computing the median measurements, for example, at each pixel, will with high probability assign that pixel the true background intensity - fixed threshold on differencing used to find “foreground” pixels
    - can also compute a distribution of background pixels by fitting a mixture of Gaussians to set of intensities and assuming large population is the background - adaptive thresholding to find foreground pixels

A 300-Frame Sequence with a “Busy” Background

click to start movie

J. Kosecka, cs482

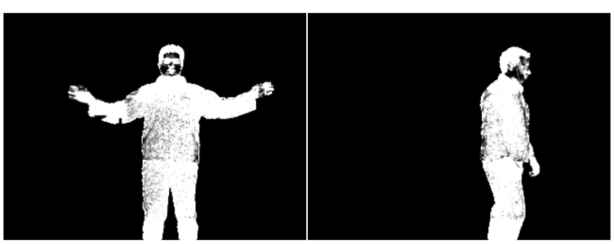
Motion Detection

- difference a frame from the known background frame
  - even for interior points of homogeneous objects, likely to detect a difference
  - this will also detect objects that are stationary but different from the background
  - typical algorithm used in surveillance systems
- Motion detection algorithms such as these only work if the camera is stationary and objects are moving against a fixed background

J. Kosecka, cs482

Background Subtraction: Results

Confidence corresponds to gray-level value.
High confidence – bright pixels, low confidence – dark pixels.



J. Kosecka, cs482

Background modeling: color-based

- At each pixel model colors *(r,g,b)* or gray-level values *g*. The following equations are used to recursively estimate the mean and the variance at each pixel:

$$\mu_{t+1} = \alpha\mu_t + (1-\alpha)z_{t+1}$$

$$\sigma_{t+1}^2 = \alpha(\sigma_t^2 + (\mu_{t+1} - \mu_t)^2) + (1-\alpha)(z_{t+1} - \mu_{t+1})^2$$

where z_{t+1} is the current measurement. The mean μ and the variance σ can both be time varying. The constant α is set empirically to control the rate of adaptation ($0<\alpha<1$).

- A pixel is marked as foreground if given red value *r* (or for any other measurement, say *g* or *b*) we have

$$|r - \mu_t| > 3\max(\sigma_r, \sigma_{rcam})$$

J. Kosecka, cs482

Background model

- σ_{rcam} is the variance of the camera noise, can be estimated from image differences of any two frames.
- If we compute differences for all channels, we can set a pixel as foreground if any of the differences is above the preset threshold.
- Noise can be cleaned using connected component analysis and ignoring small components.
- Similarly we can model the chromaticity values rc, gc and use them for background subtraction:

  $r_c=r/(r+g+b),\ g_c=g/(r+g+b)$

Background model: edge-based

- Model edges in the image. This can be done two different ways:
  - Compute models for edges in a the average background image
  - Subtract the background (model) image and the new frame; compute edges in the subtraction image; mark all edges that are above a threshold.
    - The threshold can be learned from examples
    - The edges can be combined (color edges) or computed separately for all three color channels

Foreground model

- Use either color histograms (4-bit per color), texture features, edge histograms to model the foreground
- Matching the foreground objects between frames: **tracking**
- Can compare foreground regions directly: shift and subtract. SSD or correlation: *M, N* are two foreground regions.

$$SSD = \sum_{i=1}^{n}\sum_{j=1}^{n}[M(i,j)-N(i,j)]^2$$

$$C = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} M(i,j)N(i,j)}{[\sum_{i=1}^{n}\sum_{j=1}^{n} M(i,j)^2 \sum_{i=1}^{n}\sum_{j=1}^{n} N(i,j)^2]^{1/2}}$$

Histogram Matching

- Histogram Intersection

$$I(h_c,h_b) = \frac{\sum_i \min\{h_c(i),h_b(i)\}}{\sum_i \max\{h_c(i),h_b(i)\}}$$

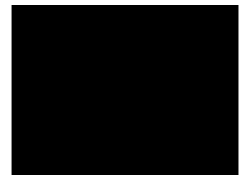
- Chi Squared Formula

$$\chi^2(h_c,h_b) = \sum_i 2\frac{(h_c(i)-h_b(i))^2}{h_c(i)+h_b(i)}$$

Surveillance: Interacting people

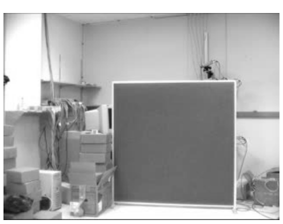
J. Kosecka, cs482

Background Subtraction



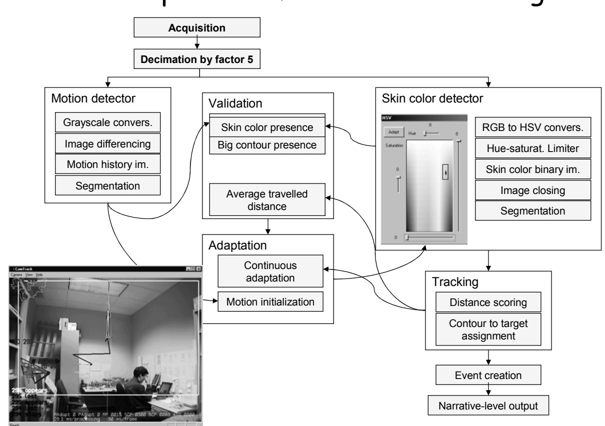
J. Kosecka, cs482

Background Subtraction



J. Kosecka, cs482

Adaptive Human-Motion Tracking



Adaptive Human-Motion Tracking