Segmentation as Graph Partitioning

- (Shi & Malik ‘97)
- Idea – each pixel in the image is a node in the graph
- Arrows 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.

Graph theoretic clustering

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

Intensity

\[ \text{aff}(x,y) = \exp\left\{ \frac{1}{2\sigma_i^2} \left( I(x) - I(y)^T \right) \right\} \]

Distance

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

Texture

\[ \text{aff}(x,y) = \exp\left\{ \frac{1}{2\sigma_t^2} \left( c(x) - c(y)^T \right) \right\} \]

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
- This is an eigenvalue problem - choose the eigenvector of $A$ with largest eigenvalue

$$w^TAw = \lambda w^T w - 1$$

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

Parallelism
Symmetry
Continuity
Closure

Not grouped
Proximity
Similarity
Similarity
Common Fate
Common Region
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

\[ \sum_{i \in \text{clusters}} \left( \sum_{j \in \text{elements of } i\text{'th cluster}} \sqrt{x_j - \mu_i^2} \right) \]

K-means clustering using intensity alone and color alone

K-means using color alone, 11 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

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
  \[
  I(x, y, t) = I(x + v_x, y + v_y, t + 1)
  \]

- Straightforward missing variable problem, rest is calculation

Three frames from the MPEG “flower garden” sequence

Segmentation of Time Varying Images (and tracking)

Grey level shows region no. with highest probability

Segments and motion fields associated with them
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

Vision Based Localization
Building 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

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
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

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

Image Differencing: Results

1 frame difference

5 frame difference
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

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

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:
\[
\begin{align*}
\mu_{z_t} &= \alpha \mu_{z_t} + (1 - \alpha) z_{t+1} \\
\sigma_{z_t}^2 &= \alpha (\sigma_{z_t}^2 + (\mu_{z_t} - \mu_{z_t})^2) + (1 - \alpha) (z_{t+1} - \mu_{z_t})^2
\end{align*}
\]
where \(z_{t+1}\) is the current measurement. The mean \(\mu\) and the variance \(\sigma\) can both be time varying. The constant \(\alpha\) 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
  \[|r - \mu_z| > 3 \max(\mu, \sigma_{\text{max}})\]
Background model

- $\sigma_{rcam}$ 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 $r_c, g_c$ and use them for background subtraction:
  $$r_c = r/(r+g+b), g_c = g/(r+g+b)$$

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.

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

$$C = \frac{\sum_{i,j} M(i,j)N(i,j)}{\left(\sum_{i,j} M(i,j)^2 \sum_{i,j} N(i,j)^2\right)^{1/2}}$$

Histogram Matching

- Histogram Intersection
  $$I(h_s, h_b) = \frac{\sum \min\{h_s(i), h_b(i)\}}{\sum \max\{h_s(i), h_b(i)\}}$$

- Chi-Squared Formula
  $$\chi^2 = \sum_i \frac{(h_s(i) - h_b(i))^2}{h_s(i) + h_b(i)}$$
Surveillance: Interacting people

Background Subtraction

Background Subtraction

Adaptive Human-Motion Tracking

Acquisition
Decimation by factor 2

Motion detector
Skin color detector

Segmentation
Motion history im.
Continuous adaptation

Validation
Skin color presence
Big contour presence
Average travelled distance

Motion initialization
Skin color presence

Adaptation
Distance scoring
Contour to target assignment

Tracking
Skin color binary im.
Image closing
Image differencing
Motion history im.
Continuous adaptation

EventClassifier
Narrative-level output
Adaptive Human-Motion Tracking