Image Segmentation

Regions and Edges

• Ideally, regions are bounded by closed contours
  - We could “fill” closed contours to obtain regions
  - We could “trace” regions to obtain edges
• Unfortunately, these procedures rarely produce satisfactory results.

• Edges are found based on DIFFERENCES between values of adjacent pixels.
• Regions are found based on SIMILARITIES between values of adjacent pixels.
• Goal associate some higher level – more meaningful units with the regions of the image.

• Regions should be homogeneous with respect to some characteristic
• (gray level, texture, color)
• Interiors should be simple – no holes
• Adjacent regions should be significantly different
• Boundaries should be smooth – not ragged

J. Kosecka
Region Segmentation

- A segmentation is a partition $R_1, R_2, \ldots, R_n$ s.t.:
  $\bigcup_{i=1}^{n} R_i = I$
  $R_i \cap R_j = 0$ for $i \neq j$
  $\text{Pred}(R_i) = \text{true} \ \forall i$
  $\text{Pred}(R_i \cup R_j) = \text{false} \ \forall R_i$ adjacent $R_j$

- Where "Pred" is a function that evaluates similarities of the pixels in the region

Histogram-based Segmentation

- Previously - thresholding determine the best threshold given a histogram of intensities
- Automatic thresholding
  - P-tile method
  - Mode method
  - Peakiness detection
  - Iterative algorithm

Limitations of histogram methods:

- Use **GLOBAL** information
- Ignore **SPATIAL** relationships among pixels.

Clustering Methods

- Pattern recognition - process of partitioning a set of 'patterns' into clusters - find subsets of points which are close together
- Examples
  - Cluster pixels based on intensity values
  - Color properties
  - Motion/optical flow properties
  - Texture measurements etc.

Input - set of measurements $x_1, x_2, \ldots, x_m$
Output - set of clusters and their centers
Clustering

• Find set of clusters such that the least squares Error is minimized

\[ E = \sum_{k=1}^{K} \sum_{x_i \in C_i} \| x_i - m_k \|^2 \]

Iterative K-means clustering algorithm
1. set \( \text{iter} = 1 \)
2. Choose randomly K-means \( m_1, \ldots, m_k \)
3. For each data point \( x_i \), compute distance to each of the means and assign the point to the cluster with the nearest mean
4. iter = iter + 1
5. Recompute the means based on the new assignments of points to clusters
6. Repeat 3-5 until the cluster centers do not change much

Soft k-means

- assign membership of points to clusters probabilistically
- Recomputed means as weighted average of the points (weights are the membership probabilities)

SPLIT & MERGE Algorithms

• Simple intensity algorithms usually result in too many regions.
  - Reasons:
    • high frequency noise
    • Gradual transitions between regions
  - After segmentation, regions might need refinement:
    - Interactively or automatically
    - May use domain and or image process knowledge

Merging Algorithm

• Merge ADJACENT, SIMILAR regions
• What does "similar" mean?
  - Examples:
    • "similar" average values: \( |\mu_i - \mu_j| < T \)
    • "small" spread of gray values: \( |g_{\max} - g_{\min}| < T \)
      - \( g_{\max} = \max\{g(x,y) \mid (x,y) \text{ from union of } R_i \text{ and } R_j\} \)
      - \( g_{\min} = \min\{g(x,y) \mid (x,y) \text{ from union of } R_i \text{ and } R_j\} \)
  - Note:
    • A similar to B, and B similar to C does not imply that A is similar to C.
Merging Algorithm

- Start with an initial segmentation
  - Ex:
    - By thresholding,
    - nxn (5x5, 7x7, etc) regions
    - manually selected
  - Each region has a unique "label"

Merging Algorithm

- Form the Region Adjacency Graph
  - Regions are the nodes
  - Adjacency relations are the links

Merging Algorithm

- For each region in the image do:
  - Consider its adjacent regions and test if they are similar
  - If they are similar, merge them and update the RAG

Merging Algorithm

- Repeat the previous step until there are no more merges.
Splitting Algorithms

- When are regions split?
  - Split a region if:
    - A property is not "constant"
    - A predicate is not TRUE
  - Deciding to split is fairly straight-forward.

- Where to split?
  - This is a difficult problem.
  - Some approaches:
    - Divide it into equal parts along image dimensions.
    - Look for strong edges to create boundaries.

Split and Merge Algorithms

- Split and merge are often used together:
  - Start with the initial image and a "predicate"
  - Test the image with the predicate:
    - If it doesn't satisfy it, split image into quarters;
    - Repeat for each sub-region until there are no more splits.
  - Test adjacent regions with the predicate:
    - If they satisfy it: merge them.
Quadtree Representation

- Quadtrees:
  - Trees where nodes have 4 children
- Build quadtree:
  - Nodes represent regions
  - Every time a region is split, it's node gives birth to 4 children
  - Leaves are nodes for uniform regions
- Merging:
  - Siblings that are "similar" can be merged.

Normalization cut approach - J. Shi, J. Malik (see textbook)

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

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

Texture
\[ \text{aff}(x, y) = \exp \left\{ -\frac{1}{2\sigma^2} \left( \|x - c(y)\|^2 \right) \right\} \]
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
  \[ a^T A a \]
  \[ a^T a = 1 \]

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
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?

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)
    - 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
**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.

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

J. Kosecka

---

K-means clustering using intensity alone and color alone

J. Kosecka

---

K-means using color alone, 11 segments

J. Kosecka

---

K-means using color alone, 11 segments.

J. Kosecka
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
  - Likelihood
    - assume
    \[
    I(x, y, t) = I(x + v_x, y + v_y, t + 1) + \text{noise}
    \]
- Straightforward missing variable problem, rest is calculation

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

Three frames from the MPEG “flower garden” sequence

Grey level shows region no. with highest probability

Segments and motion fields associated with them