Ideally, partition an image into regions corresponding to real world objects.
Goals of segmentation

1. want *meaningful* segments of image
2. many objects differ in *appearance* from others
3. *changes the representation* Pixels $\rightarrow$ Regions
4. general algorithms collect pixels into regions by
   • by *adjacency*
   • by *similarity* (distance) in color-texture space
   • algorithms can also create boundaries using *differences*
5. general algorithms can be tested quickly in applications
6. general algorithms cannot be expected to work well in all cases
7. Can there be segmentation without recognition?
Segments formed by K-means
Segmentation attempted via contour/boundary detection
Clustering versus region-growing

- Clustering (similarity by features is major control, adjacency secondary)
  1. each pixel represented by an n-dimensional vector (color-texture space)
  2. clustering partitions all $M \times N$ image pixels into $K$ classes (How?)
  3. connected components performed to apply the adjacency requirement
Clustering versus region-growing

- Region-Growing (adjacency is major control, similarity secondary)

  1. must start at some location[s] (*seeds*) (How?)
  2. only add adjacent pixels that are similar (∼*painting*)
  3. can grow multiple regions in parallel and competitively (How?)

- *Above concepts also apply to edge segments!*

  1. *boundary following* uses adjacency as major control
  2. *Hough transform* uses spatial similarity as major control
K-means clustering as before: vectors can contain color+texture

Form K-means clusters from a set of n-dimensional vectors.

1. Set $ic$ (iteration count) to 1.

2. Choose randomly a set of $K$ means $m_1(1), m_2(1), \ldots, m_K(1)$.

3. For each vector $x_i$ compute $D(x_i, m_k(ic))$ for each $k = 1, \ldots, K$ and assign $x_i$ to the cluster $C_j$ with the nearest mean.

4. Increment $ic$ by 1 and update the means to get a new set $m_1(ic), m_2(ic), \ldots, m_K(ic)$.

5. Repeat steps 3 and 4 until $C_k(ic) = C_k(ic + 1)$ for all $k$. 
K-means

- assume $K$ clusters $C_1, C_2, \ldots, C_K$ with means $m_1, m_2, \ldots, m_K$.
- least squares error measure measures how close the data are to their assigned clusters

$$D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - m_k\|^2.$$ 

- could consider all possible partitions into $K$ clusters and select the one that minimizes $D$ – computationally infeasible
- is $K$ known in advance?
Histograms can show modes

a) original image   b) pixels below 93   c) pixels above 93
Otsu’s method assumes $K=2$. It searches for the threshold $t$ that optimizes the intra class variance.

Optimize on: $\sigma_W^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$
where $q_1(t)$ is the number of pixels with property $< t$, and $q_2(t)$ is the number of pixels with property $\geq t$, 
Ohlander bifurcated the histogram recursively.
Recursive histogram-controlled segmentation

- Recursive histogram-directed spatial-clustering scheme.

- Original image has four regions: grass, sky, and two trees.

- Mask (binary labeled image) defines set of pixels

- Current mask (shown at upper left) identifies the region containing the sky and the trees

- Clustering its histogram leads to two clusters in color space, one for the sky and one for the trees.

- The sky cluster yields one connected component, while the tree cluster yields two.

- Each of the three connected components become masks that are pushed onto the mask stack for possible further segmentation (one mask for each component).
URL’s of other work

- Tutorial on graph cut method and assessment of segmentation http://www.cis.upenn.edu/~jshi/GraphTutorial/
Segmentation via region-growing (aggregation)

Pixels, or patches, at the lowest level are combined when similar in a hierarchical fashion.
Decision: combine neighbors?

Region: a population of similar pixels with mean $\bar{X}$ and scatter $S^2$
Aggregation decision

- region is a population of pixels with similar stats
- region has mean $\overline{X}$ and scatter $S^2$

$$\overline{X} = \frac{1}{N} \sum_{[r,c] \in R} I[r, c] \quad (1)$$

$$S^2 = \sum_{[r,c] \in R} (I[r, c] - \overline{X})^2. \quad (2)$$

- use a statistical test to see if border pixel $N_1$ should be added to the region
Representation of regions

- **overlay or mask**: binary image for each region or cluster
- **labeled image**: integer code for each region or cluster
- **boundary coding**: use perimeter set or chain code
- **quad tree**: hierarchical spatial partition with white, black, grey nodes
- **property table**: property or feature vector
Chain codes for boundaries

original curve

chain code links

chain code representation

100076543532

encoding scheme

CSE 803 Fall 2015
Quad trees divide into quadrants

M=mixed; E=empty; F=full

image region

quad tree representation
Can segment 3D images also

- Oct trees subdivide into 8 octants
- Same coding: M, E, F used
- Software available for doing 3D image processing and differential equations using octree representation.
- Can achieve large compression factor.
More URLs of other work