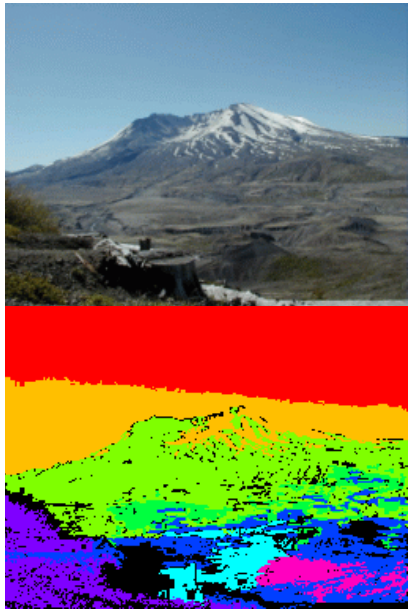
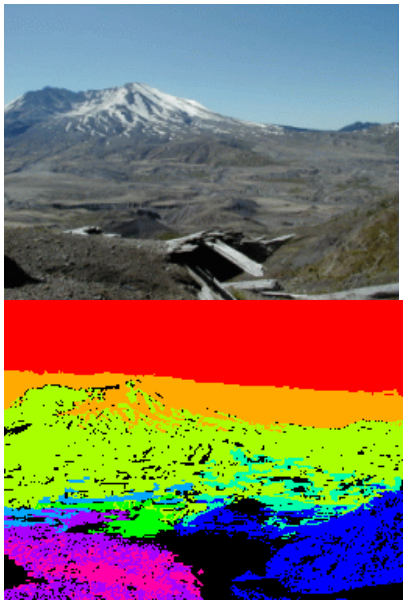




Assignment 3

Content-Based Image Retrieval

Steps

- Represent each image in the database we give you by a feature vector. (Preprocess)
- Design an image distance measure that can compare pairs of image.
- Retrieve, for each test image, the database images in ascending order of distance to the query. The query itself should have distance 0 and be first. Use the interface provided for retrieval.

Initial Processing

- First apply color clustering to the image to get a labeled image of multiple different cluster labels: 1, 2, 3, ...K.
- Then apply connected components (provided) to the labeled image to produce a second labeled image that labels each connected component of cluster labels: 1, 2, ... N. A single color cluster may break into more than one component.
- Possibly perform some noise cleaning to remove small regions. Don't vary parameters between images. You can get noise cleaning ideas or code from anywhere.

Features

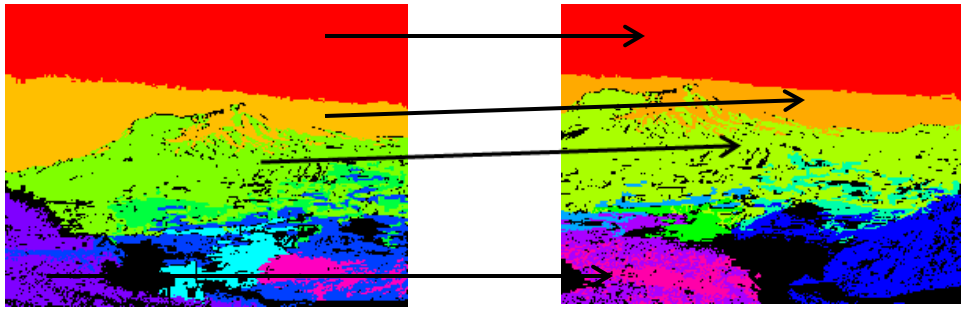
- For each major region (use a size threshold), compute *at least* the following features:
  - size (number of pixels) given
  - mean color, in RGB, or whatever space you like given
  - at least the following co-occurrence texture features using spatial relationship d=(1,1): energy, entropy, contrast.
  - centroid (row, column)
  - bounding box (or if you prefer, could be an ellipse)
- Store the features in the feature vector defined in the code.

Extra Credit Features

- Other region features you want to add
- RAG (region adjacency graph) including for each pair of adjacent regions:
  - above adjacency
  - below adjacency
  - left adjacency
  - right adjacency
  - other
- A fancier distance function to handle RAGs.

Distance Measure

- Dist(I_1,I_2) determines the distance from image I_1 to image I_2.
- Compute Dist from a correspondence you find from the regions of I_1 to those of I_2.
- Start with a greedy method: for each region of I_1, find the most similar region of I_2



- Do **not** ask me HOW to do this. That's for you.

More on Distance Measure

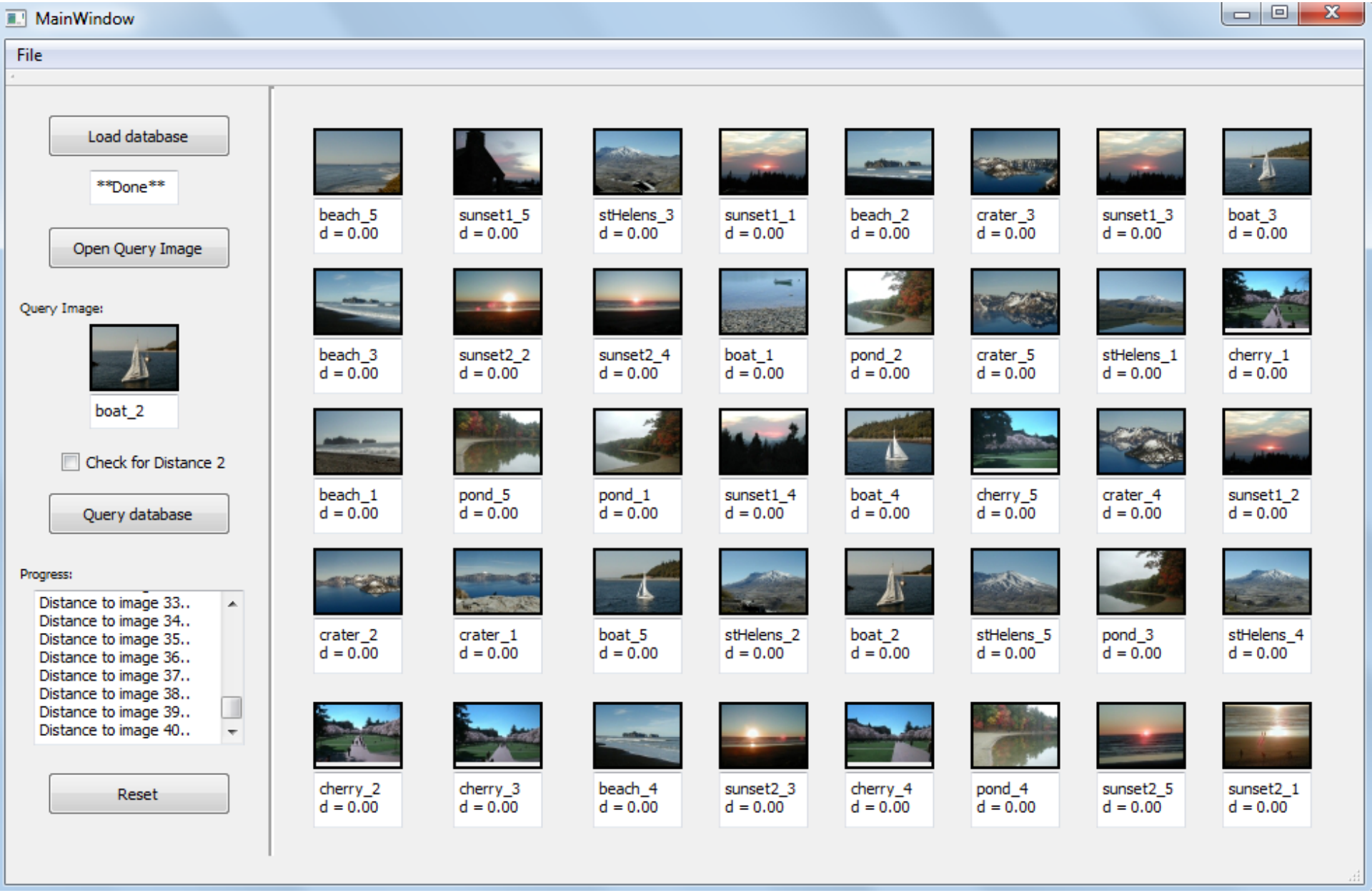
- You should try at least two difference distance measures. They can differ in:
  - attributes used, weights on attributes
  - the actual distance, ie. Euclidean vs. others

- If you do the graph structure, you need some kind of graph distance. See S&S Section 11.6 or make up your own.

Report

- Turn in a brief report in Word or PDF that describes:

1. the attributes you implemented
2. the distance measures you tried
3. the results of your tests including both the 8 screenshots and your comments.


MainWindow
File
Load database
Done
Open Query Image
Query Image:
boat_2
Check for Distance 2
Query database
Progress:
Distance to image 33..
Distance to image 34..
Distance to image 35..
Distance to image 36..
Distance to image 37..
Distance to image 38..
Distance to image 39..
Distance to image 40..
Reset
beach_5 d = 0.00
sunset1_5 d = 0.00
stHelens_3 d = 0.00
sunset1_1 d = 0.00
beach_2 d = 0.00
crater_3 d = 0.00
sunset1_3 d = 0.00
boat_3 d = 0.00
beach_3 d = 0.00
sunset2_2 d = 0.00
sunset2_4 d = 0.00
boat_1 d = 0.00
pond_2 d = 0.00
crater_5 d = 0.00
stHelens_1 d = 0.00
cherry_1 d = 0.00
beach_1 d = 0.00
pond_5 d = 0.00
pond_1 d = 0.00
sunset1_4 d = 0.00
boat_4 d = 0.00
cherry_5 d = 0.00
crater_4 d = 0.00
sunset1_2 d = 0.00
crater_2 d = 0.00
crater_1 d = 0.00
boat_5 d = 0.00
stHelens_2 d = 0.00
boat_2 d = 0.00
stHelens_5 d = 0.00
pond_3 d = 0.00
stHelens_4 d = 0.00
cherry_2 d = 0.00
cherry_3 d = 0.00
beach_4 d = 0.00
sunset2_3 d = 0.00
cherry_4 d = 0.00
pond_4 d = 0.00
sunset2_5 d = 0.00
sunset2_1 d = 0.00