Final Project CS/BIOEN 6640 Fall 2014

Final Project Due: 12/16/2014, 11:59 midnight (official deadline of final exam date for this class).

Concept: The final project is a kind of "double project", i.e. there are less guidelines but just a layout of ideas where students can be creative to develop solutions. As in all projects, there is coding, generation of experiments and results, and a detailed report on the concept, methodologies and evaluation with sufficient level of details for readers to fully understand what is being done and how to interpret the results.

Students can choose between 3 major themes centered about methodologies which were discussed in class.

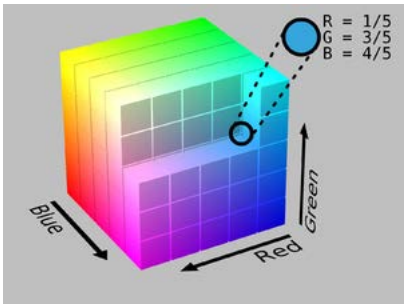
Theme 1: Classification of color images using color histogram methods.

Theme 2: Generation of Panorama images from sets of images.

Theme 3: Hough transform to automatically find objects in images

Theme1: Classification of color images using color histogram methods.

We used the histogram for thresholding and segmentation of scalar black-and-white image data. Color images have three channels RGB, and the histogram is now being seen as a histogram in the 3-dimensional RGB space.



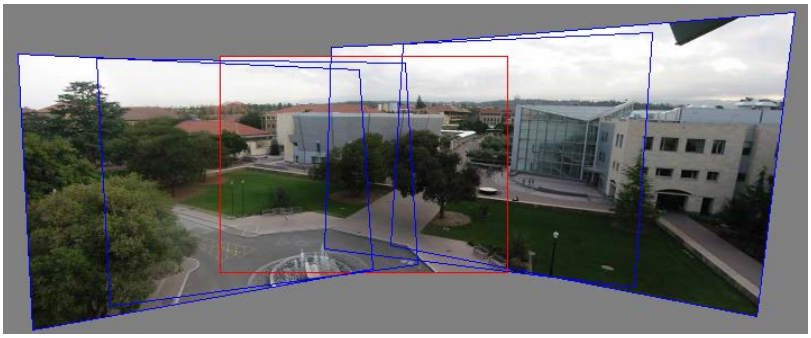
Let us think about color images of objects that have a distinct, primary color. Your task may be to take a picture, create a simple descriptor, and classify new images into the respective categories.

Sketching a procedure:

- Choose an application area which you like, the fruits above are just one example, but these could be cars, toys etc.. Objects need to have very large uniformly colored regions without background clusters. You may choose an application with a relatively small number of categories, e.g. around five.
- The color histograms can be very coarsely binned (e.g. 2-4 bins per color, resulting in 2*2*2=8 up to 4*4*4=64 cubic bins). The histograms can be seen as 1-D vectors with these 8 to 64 bins.
- The RBG images can be mapped into these histograms to fill the bins (just remember the histogramming project).
- **Classification**:
    - Choose a typical template per category and create a color histogram or 1-D color vector, respectively.
    - Choose new images containing such an object, and calculate the color histogram for the new data.
    - Calculate the **distance** between the new color histogram vector and all the category templates. A possible distance is the sum of squares difference between the vectors, others may be the dot product etc.
    - Choose the object which smallest distance as the result category.

Theme 2: Generation of Panorama images from sets of images.

This theme follows instructions as you find in the lecture slides on image stitching/mosaicking. Given a set of images taken from a single viewpoint by rotating the camera, generate a panorama.



The project should have two parts, the first using manually selected landmarks in overlapping regions and calculation of the transformation. The second should use a set of image templates to find the best location by template matching.

A) Manual landmarking:

4 correspondences

Try larger numbers of landmarks and test (e.g. blending, overlap, showing landmarks in panorama) if more landmarks would give a better alignment.

B) Landmarking by template matching: Select a few small regions with good detailed texture in the left image. Find the best matching region for each template in the second image. Use these pairs to calculate the transformation.

Theme 3: Hough transform to automatically find objects in images

This theme follows the lecture on Hough transform, but using the HT for circles. The image below shows a typical example of an object detection problem, i.e. to automatically classify coins. However, you may choose your favorite application where circular objects are to be found and classified.

The procedure can be described as follows:

- Choose your favorite image(s) containing circular objects (best are images where object and background are well separated). It is also suggested to use an example where you have a limited number of radii, i.e. objects from a small number of categories of circle sizes.
- Apply Canny edge detection and thresholding to find the boundaries. Besides the boundary, you may also be interested in using he gradient as the normal to the edges.
- Implement the HT for circles as described in the course slides. You may implement versions with/without using the edge gradient. Select a strategy for robust detection of peaks.
- Apply the HT to the images, detect peaks, and overlay the detected circles on the image for verification.
- Extend the HT for a single radius to multiple radii by running the HT repeatedly, storing the accumulators for the set of radii, and select peaks for circles with different sizes.