Computer Vision **CS 543 / ECE 549**

Homework 3
Due April 5, 2011

Suggested Completion Dates: P1 by 3/22; P2, P3 by 3/29; P4 by 4/5

Answer the following questions and explain all solutions. Numbers in parentheses give maximum credit value. Remember that code is not an explanation.

1. Object Instance Recognition (45%)

This problem explores the Lowe-style object instance recognition. For the third part, it is strongly recommended to use the included Matlab source code and data.

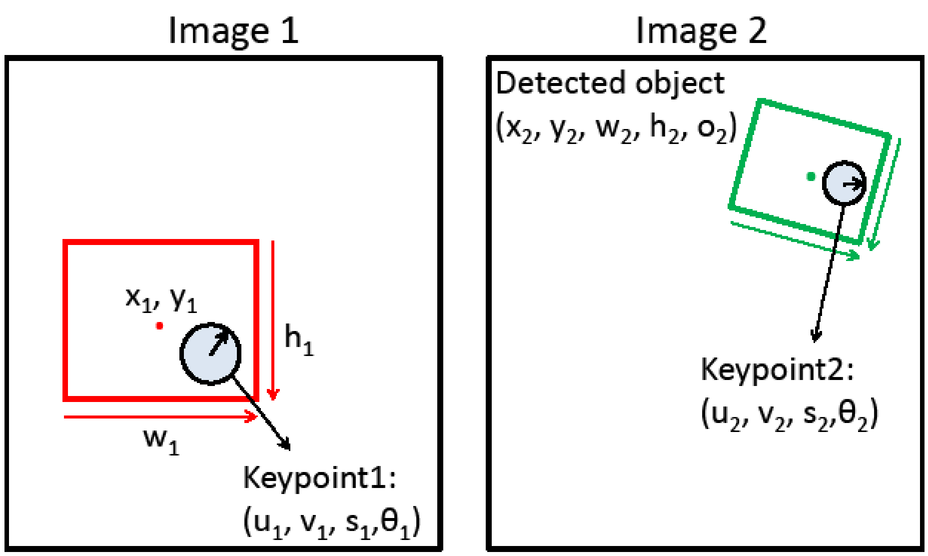


Figure for Problem 1B

A. Given a keypoint descriptor **g** from one image and a set of keypoint descriptors $\mathbf{f}_1 \ldots \mathbf{f}_n$ from a second image, write the algorithm and equations to determine which keypoint in $\mathbf{f}_1 \ldots \mathbf{f}_n$ (if any) matches **g**. (5%)

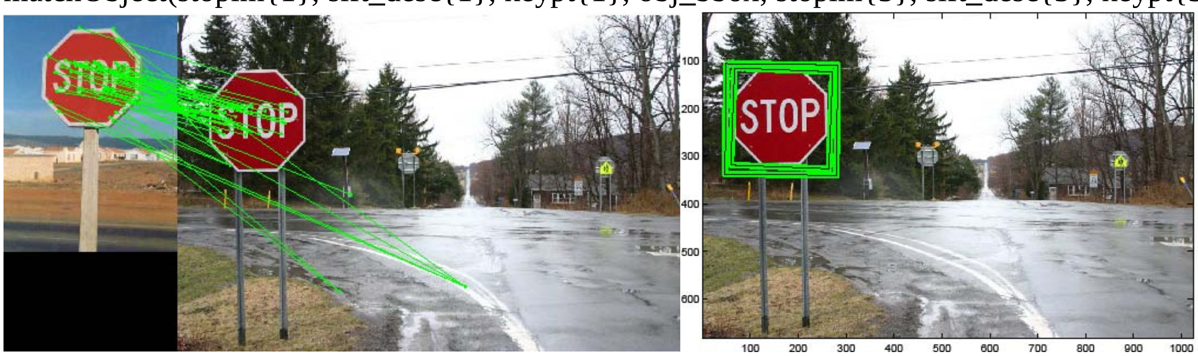
B. Suppose that you have matched a keypoint in the object region to a keypoint in a second image (see Figure 1B above). Given the object bounding box center x-y, width, and height (x1, y1, w1, h1) and the position, scale, and orientation of each keypoint (u1,v1,s1,theta1; u2,v2,s2,theta2), show how to compute the predicted center position, width, height, and relative orientation of the object in image 2. (15%)

C. Implementation. Use the stop sign in stop1.jpg (stopim1) with coordinates ([x1 y1 x2 y2] = [76 26 287 236]) as a training example. Match keypoints in the other four images and recover the position, scale, and orientation of the objects. A rough result is ok – it is not necessary to perform a subsequent geometric verification or refinement step (but you can, if you want). Describe your algorithms to match keypoints and to use the matched points to localize the objects. Where applicable, you may refer to your solutions in parts A and B. Explain any design decisions. Use the same code and parameters for all four images. If you are not able to localize the objects (this could happen in stopim2 and stopim5), explain what makes these cases difficult. *For each image, include figures that show the corresponding keypoints and the detected objects* (code is included to do this). If you turn in printed homework, please use a high-quality printer. (25%)

The supplemental file (hw3_prob1.zip) contains the following:

- Five images (stopim1.jpg, stopim2.jpg, stopim3.jpg, stopim4.jpg, stopim5.jpg)
- SIFT descriptors and keypoints for each image (hw3_prob1.mat). The SIFT keypoints were created using the code here: http://www.vlfeat.org/~vedaldi/code/sift.html. For each keypt, the four coordinates are x, y, scale, and orientation, in that order.
- A function (matchObject.m) that outlines the recognition process and includes code to display matches and results. You need to write the functions to match keypoints and recover likely object position/size/orientation.

Here is an example of the output for one of the images (stop3):
matchObject(stopim{1}, sift_desc{1}, keypt{1}, obj_bbox, stopim{3}, sift_desc{3}, keypt{3});



2. Principal Components Analysis (PCA) (10%)

Suppose you are provided with 1024 face images of size 128x128 pixels, each stored as a vector of intensity values: $\mathbf{x}_1..\mathbf{x}_{1024}$.

A. Suppose: (1) each pixel value requires one byte to store; (2) that each element of the principal component requires two bytes; (3) and that each component weight requires two bytes. What fraction of space is saved if 30 principal components are used to represent the data? Remember that the mean vector, the principal component vectors, and the component weights for each image must be stored. (5%)

B. How many principal components are required to store these face images with zero reconstruction error? How much space is required in that case? Would the reconstruction error be zero for a new face image? (5%)

3. Scene Categorization using Histograms (30%)

The supplemental data (hw3_prob3.zip) contains images from two classes: “coast” and “forest”. The included “.mat” file contains the training/test images and labels in a simple format. Using the training images (50 images for each class), train classifiers to differentiate between the two classes. Describe the feature representation and classification method used, including any equations and design decisions. Test your classifiers on the test images and report accuracy. Also show one example from each class that is misclassified (if accuracy is not perfect). (30%)

Note: You may use a simple method, such as histogram of color pixels with a nearest-neighbor classifier. You could also try texture or gradient features, or spatial pyramids, or other classifiers such as SVM. I achieved 92% accuracy using color histograms with 1-NN. You may achieve less or greater accuracy. Feel free to experiment with different features or classifiers and report results for each.

4. Object Category Detection (15%)

Statistical template sliding window detectors have been a mainstay of object detection since about 1995 ([Rowley Baluja Kanade 1996] and [Sung Poggio 1994]). Features are defined at each position within a coordinate system specified by the bounding box, and a classifier determines whether each detection window is an object of interest or not. Name *two object categories* that are difficult to detect using this approach. What makes each category difficult? Can you suggest an alternative approach that might work better?