I400/B457 Intro to Computer Vision - Programming assignment #6

This assignment heavily relies on lecture7_1_localfeats.ppt and lecture8_1_indexing.pptx. We will use the opencv library.

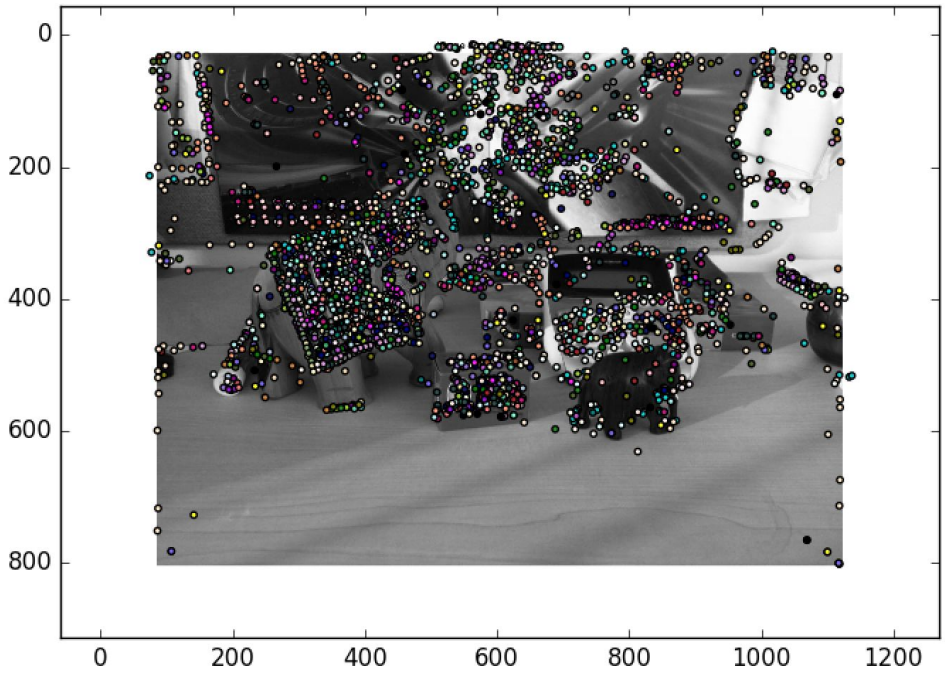
Problem 1: Bag-of-words matching with SIFT [50%]

0. Download the images from Canvas: elephant_model.png, staple_remover_model, and cluttered_desk.png

1. Extract SIFT features from the images using OpenCV.

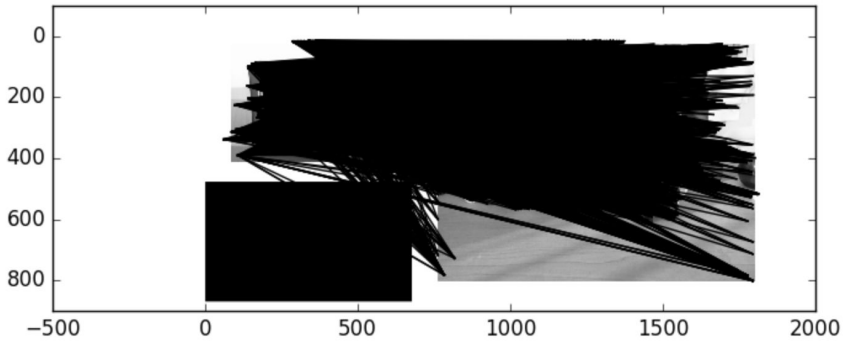
2. Implement bag-of-words. Concatenate all features extracted from all three images. Use kmeans library (from the programming assignment #3) to cluster them into k clusters. Try different k (e.g., k = 25, 50, 100, 200, ...). Visualize the clustering results on top of each image by coloring each SIFT feature based on its cluster ID (random color generation from the PA#3 will help). Save the results as elephant_model_bow.jpg, staple_remover_model_bow.jpg, and cluttered_desk_bow.jpg.

Example result:

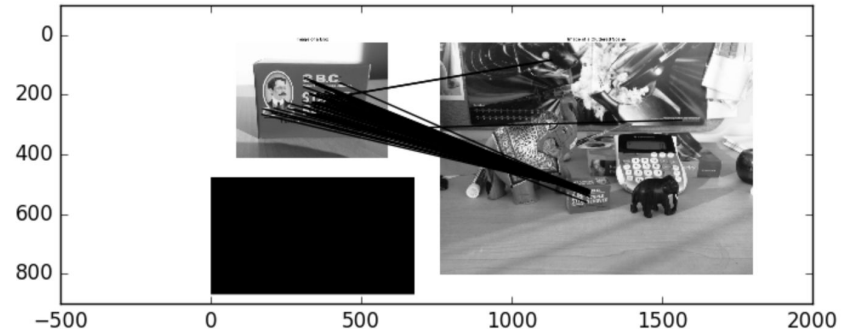


3. Match SIFT features (1) between elephant_model.png and cluttered_desk.png and (2) staple_remover_model.png and cluttered_desk.png, solely based on their cluster ids. If two features have the same cluster ID, there are matching. If not, there's no match. Visualize the results similar to what you did in PA #5. The matching can be noisy like below. There will be one-to-many matches per SIFT.
Save your results as match1.jpg and match2.jpg.

Example result:



4. Apply RANSAC identical to the problem 1 above. Use `cv2.findHomography` once more. Save the result as match1_RANSAC.jpg and match2_RANSAC.jpg.



Problem 2: Image retrieval using Bag-of-words [70%]

: For this problem, we will need to use the OpenCV SIFT feature extractor (from PA5), kmeans clustering (PA3 and the above problem1) for learning visual words, and bag-of-visual-words (BoW) histogram construction (slides 22-23, 30-33 of lecture8_1.pptx).

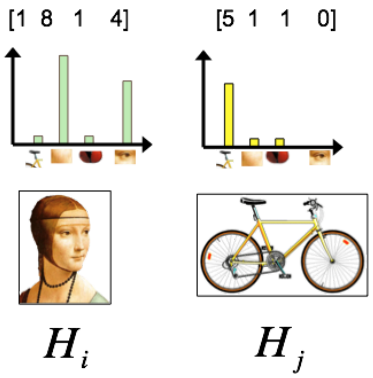
0. Download the Caltech101 dataset from either the original website:
http://www.vision.caltech.edu/Image_Datasets/Caltech101/ or a faster IU server:
http://homes.soic.indiana.edu/classes/spring2016/csci/b490-mryoo/101_ObjectCategories.tar.gz
http://homes.soic.indiana.edu/classes/spring2016/csci/b490-mryoo/101_ObjectCategories.zip

We will only be using a subset of the dataset. We will only consider the following 10 categories: {airplanes, camera, chair, crab, crocodile, elephant, headphone, pizza, soccer_ball, starfish}. For each of these categories, we will only use 10 image files: image_0001.jpg~image_0010.jpg. That is, we only use 100 images.

1. For each image, extract SIFT features.

2. Use k-means to find cluster centers (k=200) of 128-D SIFT descriptors. When doing this, only use SIFT features from “image_0001.jpg” of each category (i.e., we only use 10 images).

3. Based on the learned cluster centers, for each of the 100 images, construct a bag-of-visual-words histogram. This can be done by just assigning each SIFT feature to the nearest cluster. Since our k is 200, it will be a 200-D histogram.



It will be important to save the obtained BoW histograms in a hard-disk, since it will take a good amount of computational time to re-compute this every time you debug your code. Please try to save it as a file, and also submit the file when you upload it to Canvas.

4. Use “image_0001.jpg” of each category as the query image. Try to find 5 best matches per query. The match score is computed as the Euclidean distance between two BoW histograms: the lower distance implies the higher match score. Use `scipy.spatial.distance.euclidean` to compute Euclidean distance.

Save the retrieved images as “airplanes_query_result1.jpg”, …, “airplanes_query_result5.jpg”, “camera_query_result1.jpg”, ...