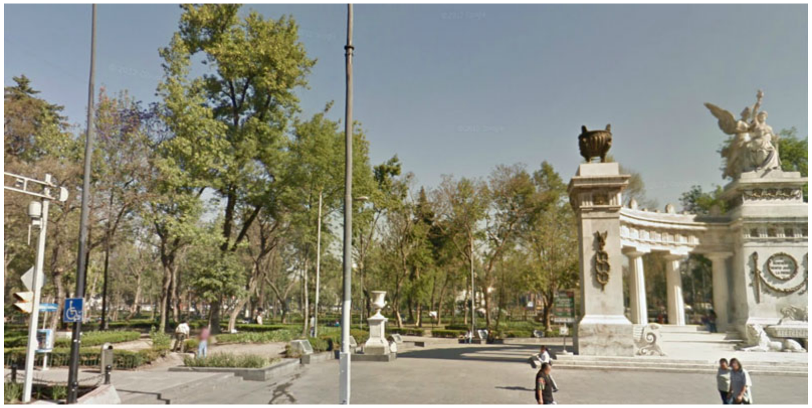
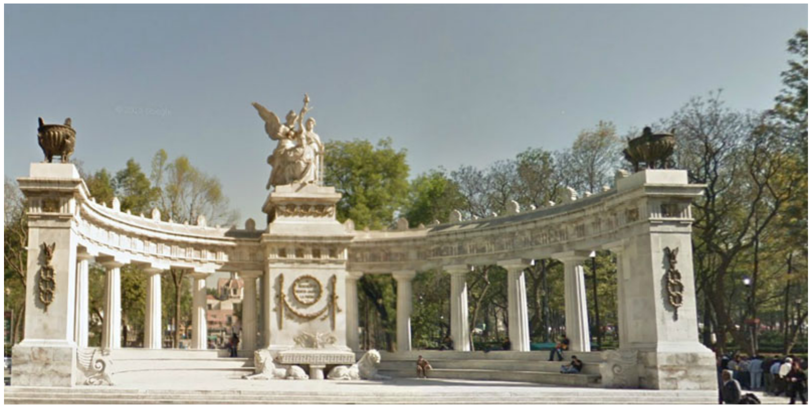
CS 682

HW3: Robust Estimation and panorama stiching

Stitching pairs of images

The first step is to write code to stitch together a single pair of images. For this part, you will be working with the following pair (click on the images to download the high-resolution versions):





Part 1 (10 pts)

1. Load both images, convert to double and to grayscale.

2. Detect feature points in both images. Use the OpenCV library to extract keypoints through the function `cv2.xfeatures2D.SIFT_create().detect` and compute descripters through the function `cv2.xfeatures2D.SIFT_create().compute`. This tutorial provides details about using SIFT in OpenCV.

3. Compute distances between every descriptor in one image and every descriptor in the other image. In Python, you can use `scipy.spatial.distance.cdist(X,Y,'sqeuclidean')` for fast computation of Euclidean distance. If you are not using SIFT descriptors, you should experiment with computing normalized correlation, or Euclidean distance after normalizing all descriptors to have zero mean and unit standard deviation.

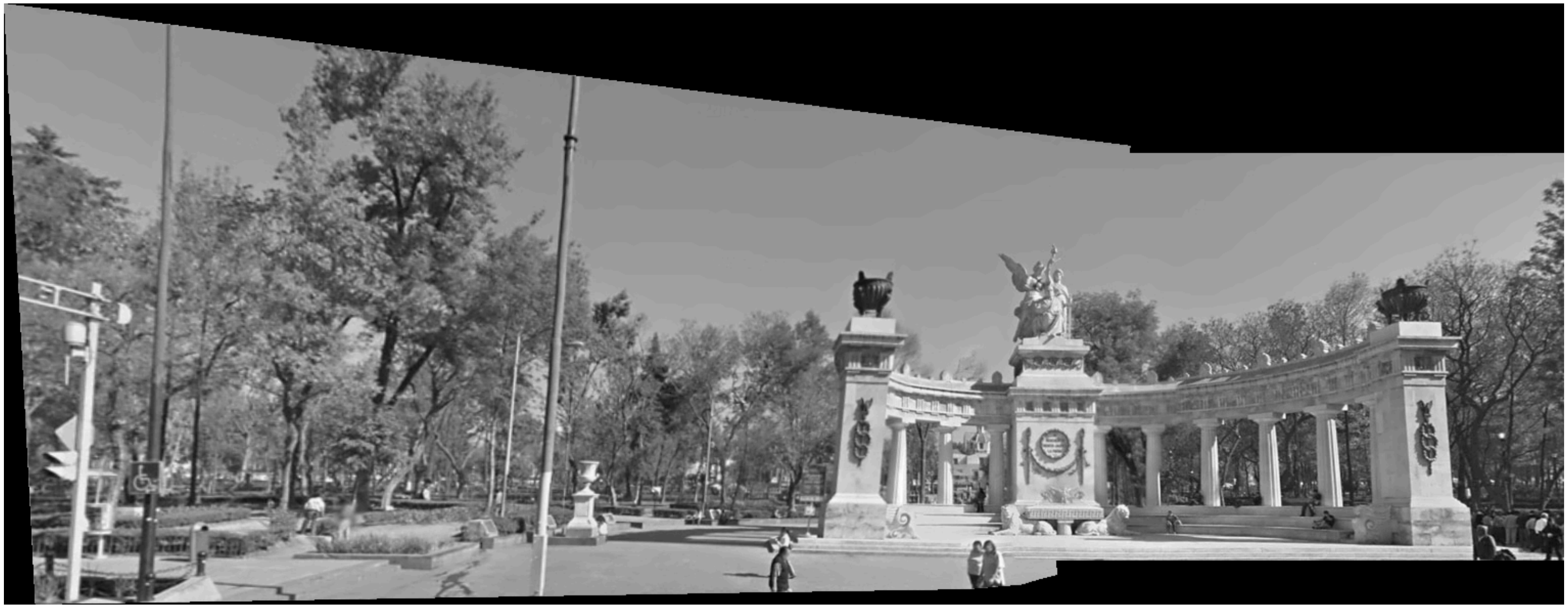
4. Select putative matches based on the matrix of pairwise descriptor distances obtained above. You can select all pairs whose descriptor distances are below a specified threshold, or select the top few hundred descriptor pairs with the smallest pairwise

distances.

5. Implement RANSAC to estimate a homography mapping one image onto the other. Report the number of inliers and the average residual for the inliers (squared distance between the point coordinates in one image and the transformed coordinates of the matching point in the other image). Also, display the locations of inlier matches in both images.

Part 2 (5 pts extra credit)

Warp one image onto the other using the estimated transformation. In Python, use `skimage.transform.ProjectiveTransform` and `skimage.transform.warp`. Create a new image big enough to hold the panorama and composite the two images into it. You can composite by averaging the pixel values where the two images overlap, or by using the pixel values from one of the images. Your result should look something like this:



You should create color panoramas by applying the same compositing step to each of the color channels separately (for estimating the transformation, it is sufficient to use grayscale images).

Tips and Details

- For RANSAC, a very simple implementation is sufficient. Use four matches to initialize the homography in each iteration. You should output a single transformation that gets the most inliers in the course of all the iterations. For the various RANSAC parameters (number of iterations, inlier threshold), play around with a few "reasonable" values and pick the ones that work best.

- Homography fitting calls for homogeneous least squares. The solution to the homogeneous least squares system AX=0 is obtained from the SVD of A by the singular vector corresponding to the smallest singular value. In Python, `U, s, V = numpy.linalg.svd(A)` performs the singular value decomposition and `V[len(V)-1]` gives the smallest singular value.

Grading checklist

Be sure to include the following in your report:

1. Homography estimation:
   a. Describe your solution, including any interesting parameters or implementation choices for feature extraction, putative matching, RANSAC, etc.
   b. For the image pair provided, report the number of homography inliers and the average residual for the inliers (squared distance between the point coordinates in one image and the transformed coordinates of the matching point in the other image). Also, display the locations of inlier matches in both images.
   c. Display the final result of your stitching (optional for extra credit).

Instructions for turning in the assignment

You must upload the files to GMU Blackboard.

1. All your code **in a SINGLE zipped file**. The filename should be **netid_hw3_code.zip**. There is no need for PDFs of any ipython notebook output, just make sure you include the notebooks themselves in the zip file and show any required outputs in the report.
2. A single report for all three parts in PDF format. The filename should be **netid_hw3_report.pdf**.