MP3 - Report

Course: Computer Vision (CS543/ECE549)

Credit: 4

Instructor: Prof. Svetlana Lazebnik

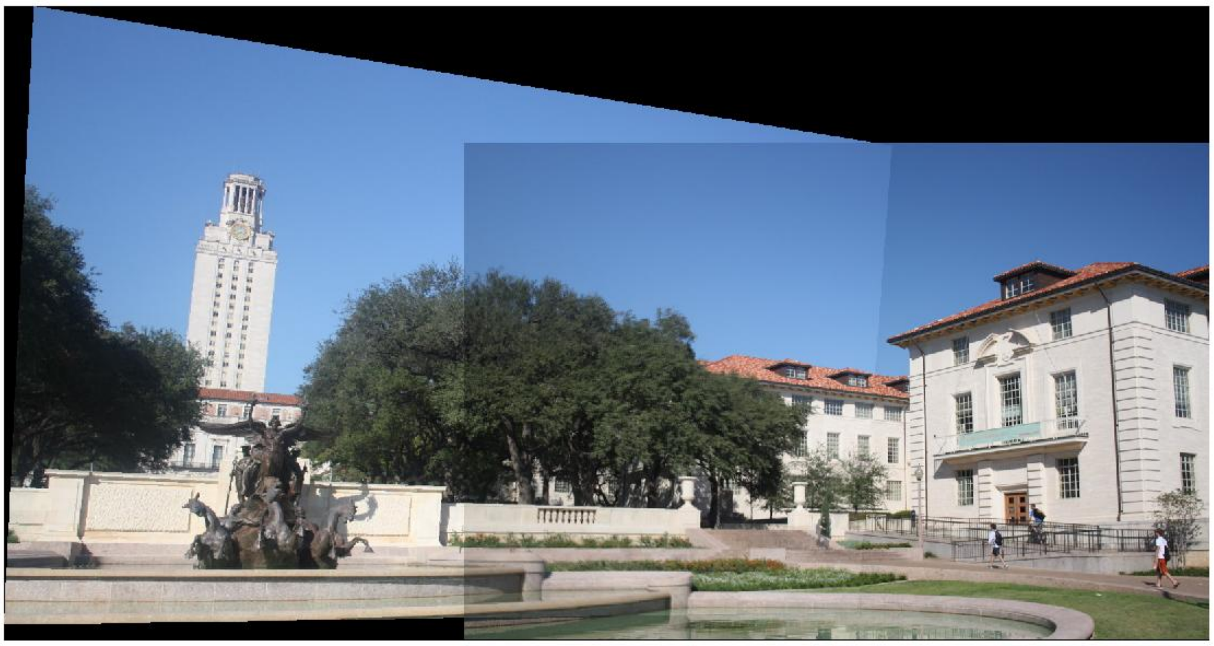
TA: Arun Mallya, Zhicheng Yan

Name: Bo-Yu Chiang

UIN: 677629729

email: bchiang3@illinois.edu

Pairwise Stitching



Solutions:

1. The two input images are fed into Harris corner detector, get two set of corners.
2. Build descriptors for every corners with size = (2*neighbor_size+1)* (2*neighbor_size+1)
3. Build distance matrix for Euclidean distances of every possible pair of two descriptors
4. Find top 200 descriptor pairs with the smallest distance
5. Matched descriptors fed into RANSAC

Parameters in harris code:

sigma: 3

thresh: 0.05

radious: 3

Parameters in RANSAC iteration: 200

Descriptors size: I tried 3*3, 5*5, 7*7, 9*9, 11*11, 13*13, 11*11 works best

There are two possible ways to describe descriptor:

1. 11*11 one-dimensional vector with original intensity.
2. 11*11 one-dimensional vector with zscore (zero mean and unit standard deviation)

There are two possible ways to compute matched of descriptor:

1. Euclidean distance of two descriptors
2. Normalized correlation of two descriptors

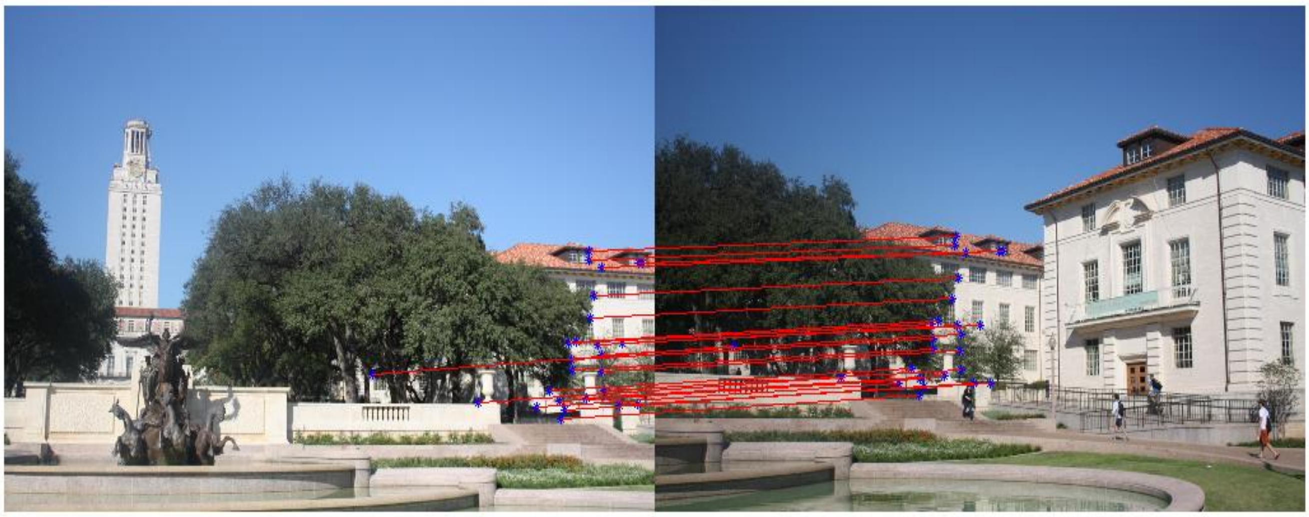
   Note: The best matched descriptor not computed by min value (like previous approach), but by max value because all the normalized correlation are between -1 and 1 and the larger the value, the closer the two descriptor.

Descriptors with original intensity and Euclidean distance

Number of inliers: 26

Average residual: 0.3034

Inliers matches:

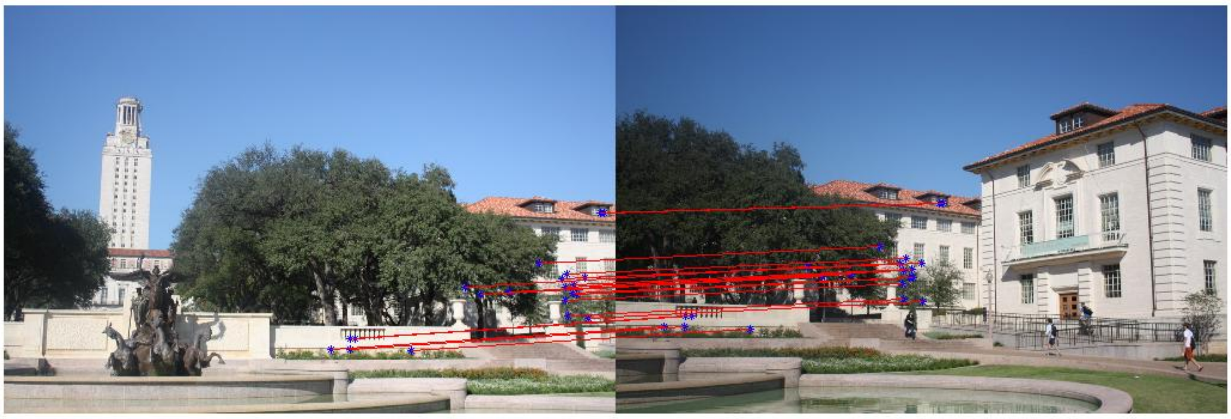


Descriptors with original intensity and Normalized correlation:

Number of inliers: 21

Average residual: 0.3170

Inliers matches:

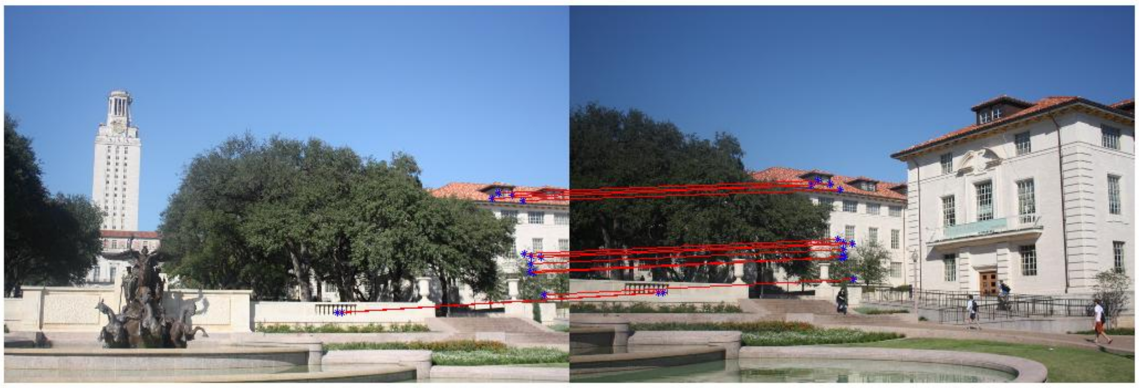


Descriptors with zscore and Euclidean distance:

Number of inliers: 13

Average residual: 0.3181

Inliers matches:

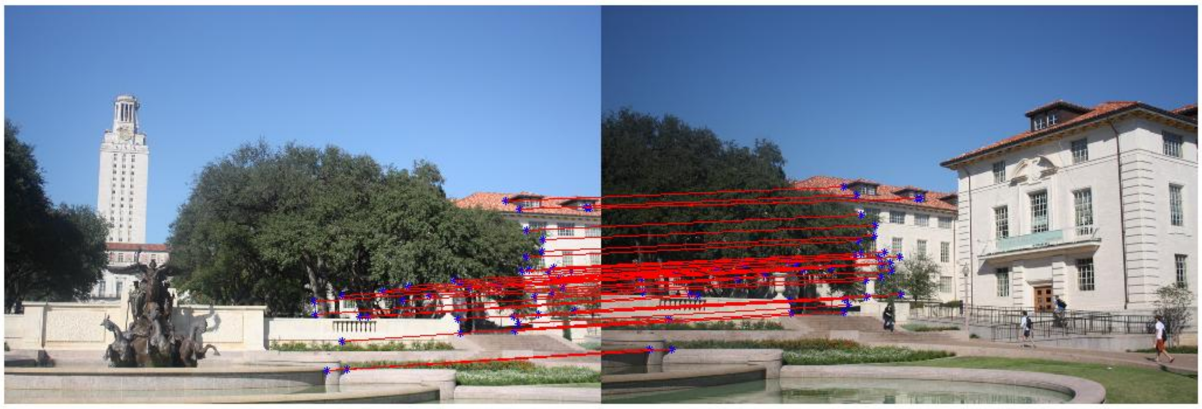


Descriptors with zscore and Normalized correlation:

Number of inliers: 41

Average residual: 0.3527

Inliers matches:



Discussion:

Descriptor with normalized correlation is a good pre-process because the descriptors are invariant to affine intensity change. I got more inliers and some of the matches lie on the tree, which is very rare in first approach.

Stitching multiple images

Strategy for ordering:

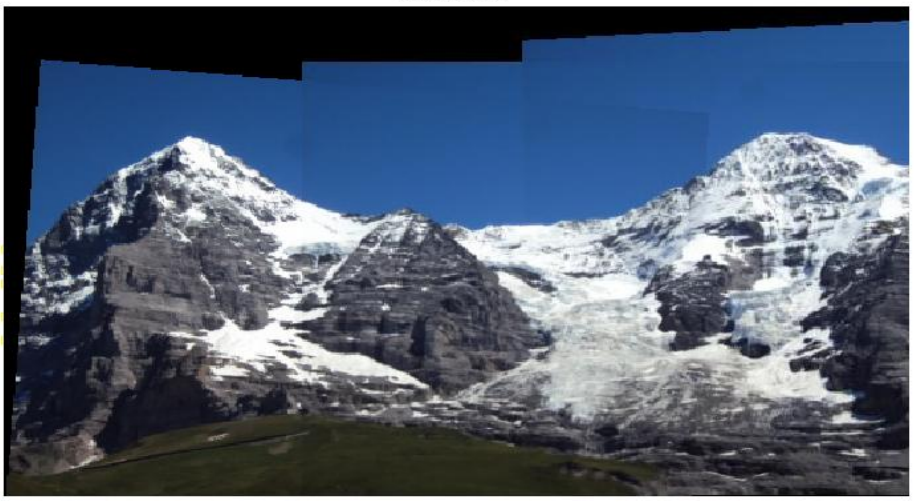
1. Decide which image is middle image
   1) Compute number of inliers for every pair (inliers_12, inliers_23, inliers_31)
   2) The common input for the largest number of inliers and second largest number of inliers is the middle input. (inliers_12: 30, inliers_23: 25, inliers_31: 10, then input2 is the middle image)
2. Decide which one is left, which one is right
   3) Since we know which one is middle image, compute the avarege x-coordinate of inliers (say, middle image is input2, compute the x-coordinate of inliers_12)
   4) If the average x-coordinate is on the left part of the image, then it is the right image compare to the middle image and vice versa.
3. After deciding left, middle and right images, first stitch the left image and middle image into an output image. Then, stitch the right image into final output image. By doing so, the middle image will have to distortion or affine effect, which I think is better for three images stitching.

Note: Given the algorithm above, No matter what is the input sequence, I can correctly judge the left, middle and right image and merge them:

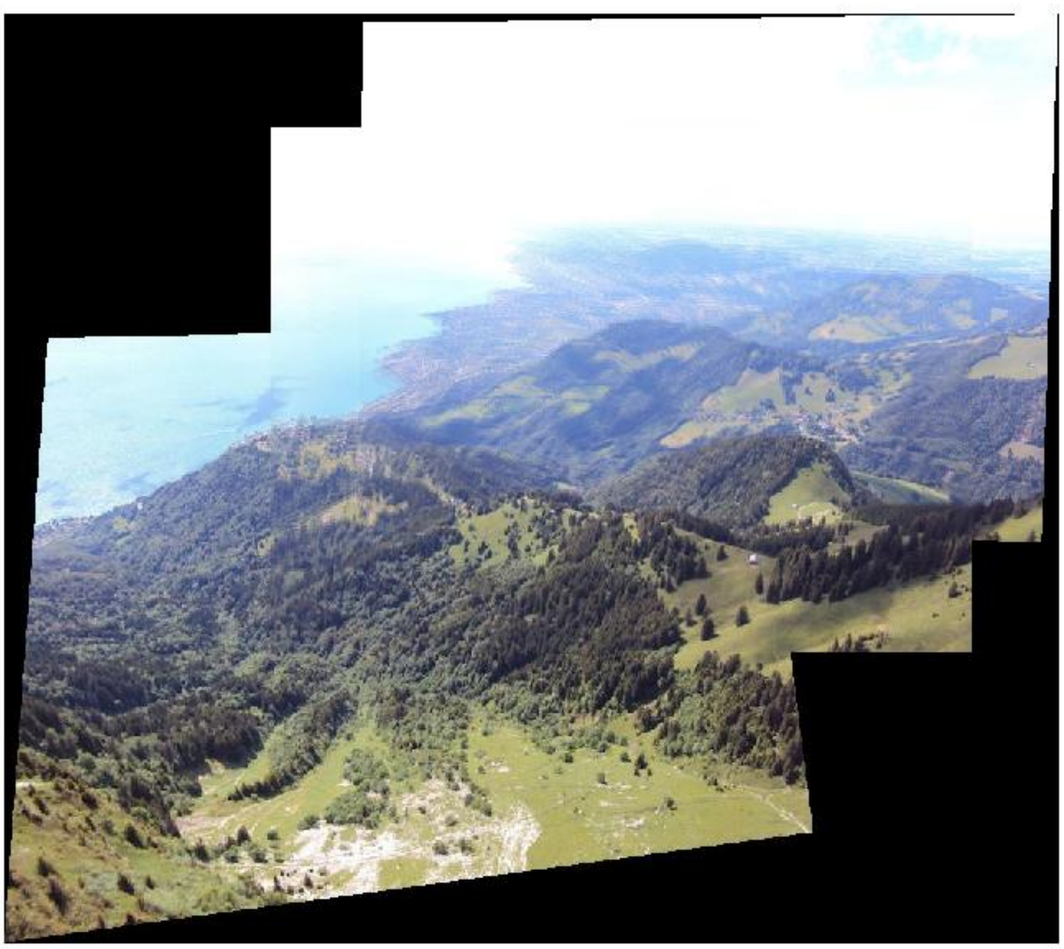
Output = merge(right, merge(left, middle))

So the output will be the same no matter the order of given input images. I only show one below for every inputs.

Hill:



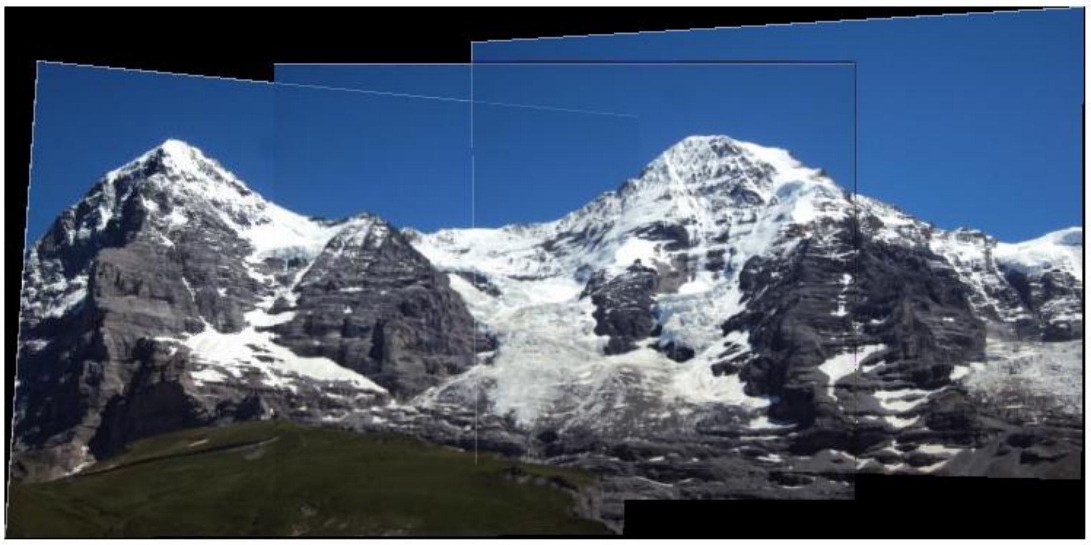
Ledge:



Pier

Artifacts applied:

In hill’s example, there are white border around the image, which may cause a bad output as below:



I filter the remove the white border before I stitching, and the output looks better as below: