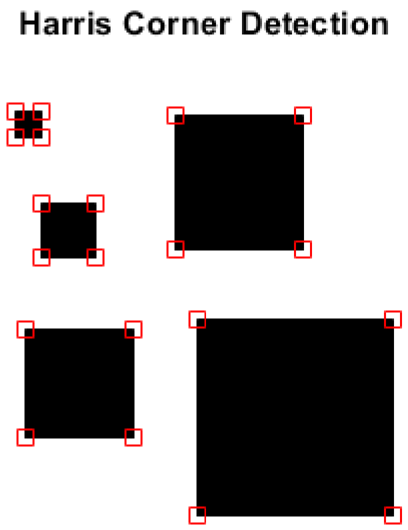
CMPE 465 ASSIGNMENT 2

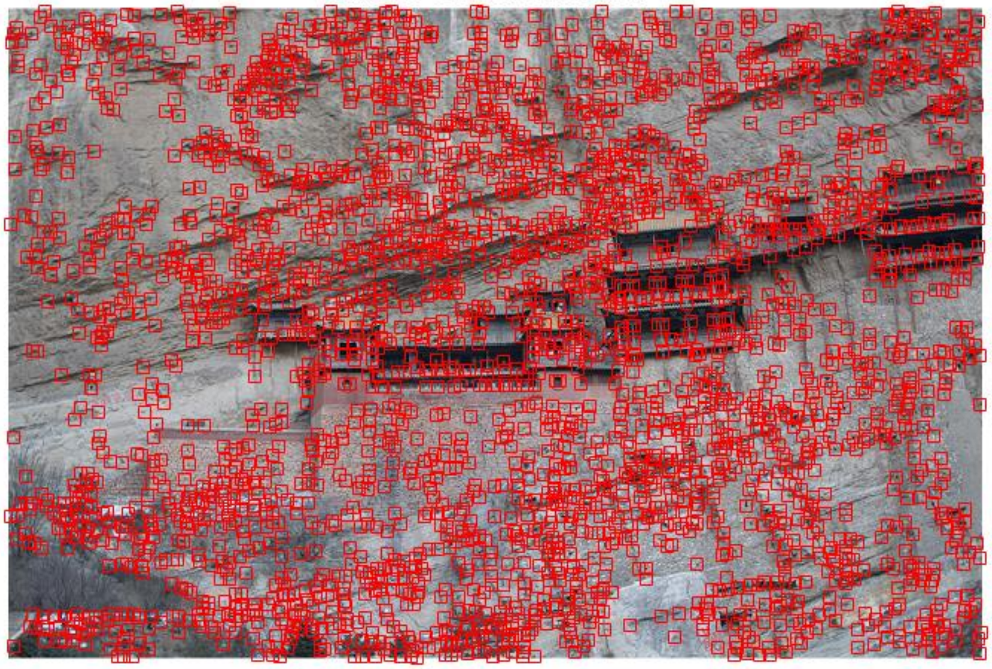
Berke Evrensevdi

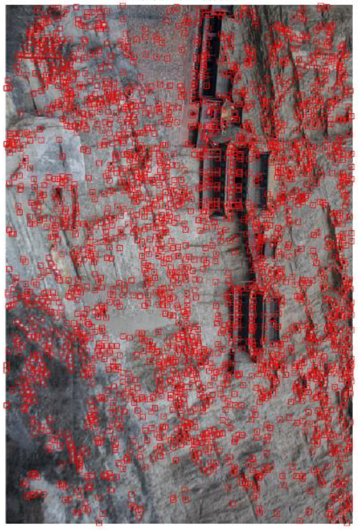
1- In part 1, I used imgradientxy function while computing x and y derivatives of the image. Imgradientxy function uses sobel method as default. Then from x and y derivatives, I obtained Ix^2, Iy^2, and IxIy. Then I smooth all of them with gaussian method. Afterward, I obtained response function from Ix^2, Iy^2, and IxIy. For finding local maxima of the response function, I used ordfilt2 function and it takes three parameters; (R function, order, window). I used 5x5 window which is ones(5) that comprises of 1's. And the order is 25, since in 5x5 windows there are 25 elements and we need to find the biggest number in it. To sum up, ordfilt2 will use 5x5 matrix and in this 5x5 matrix, it will find the biggest number, then put this number into the center of focused 5x5 part of R and it will apply this for all the 5x5 parts of R. After localMax is obtained, logical image is formed by comparing R function and localMaxima. If local max is already at the center of focused 5x5 part of R and also this local max is greater than threshold, then it will be 1 on the logical image which means that it is a corner. The results are shown below.



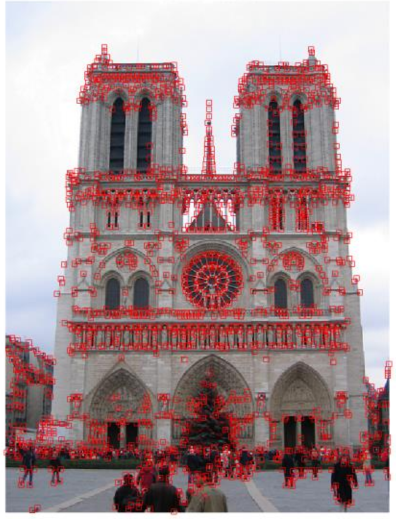
Harris Corner Detection

Harris Corner Detection



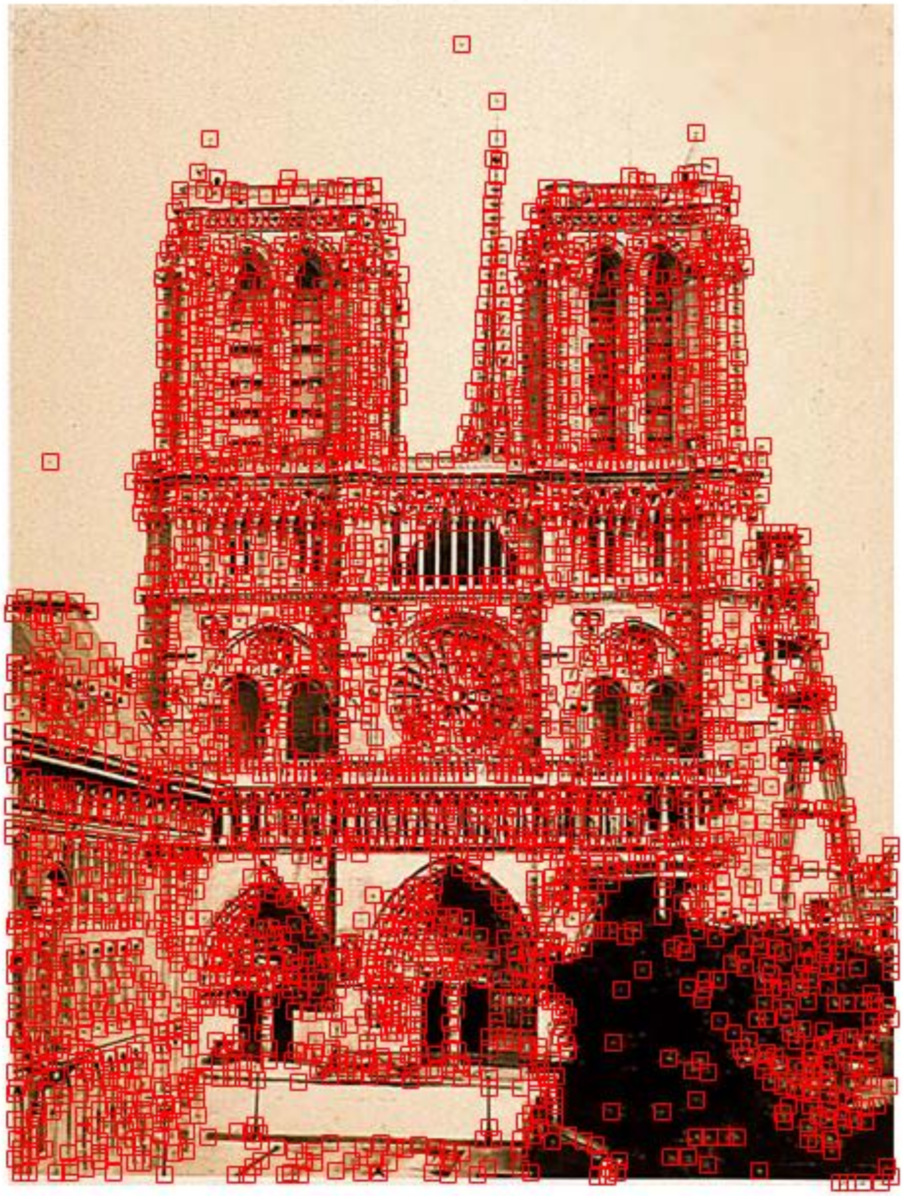


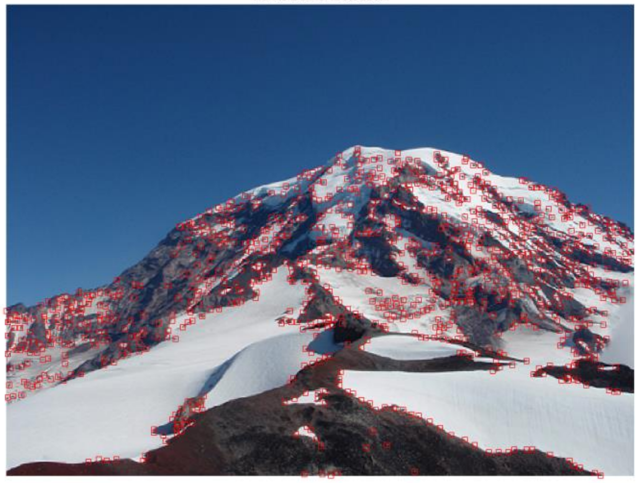
Harris Corner Detection



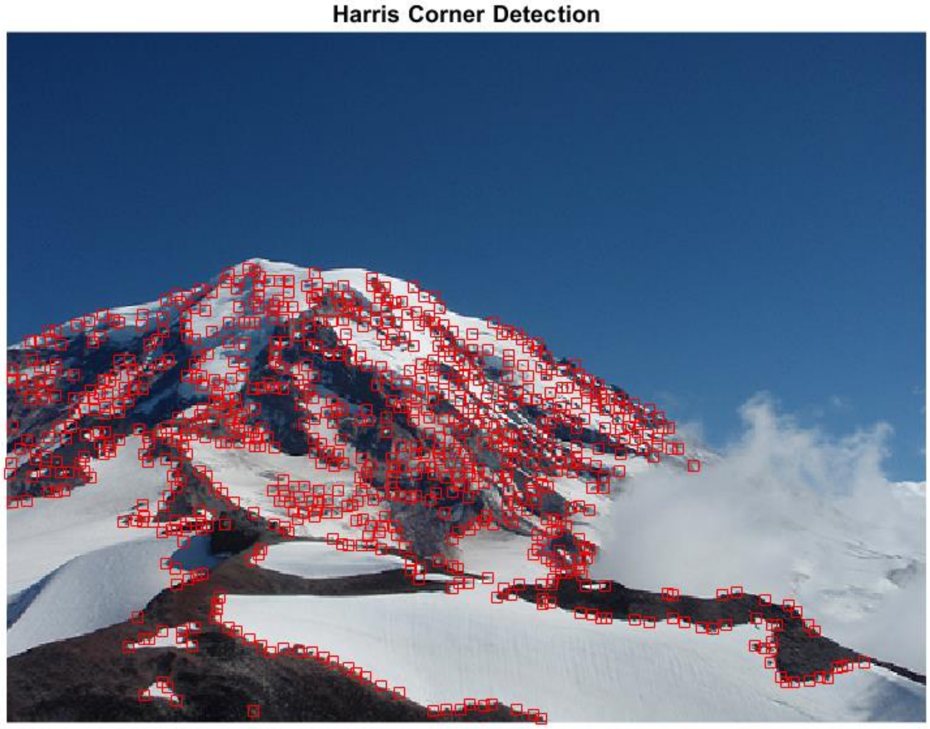
Harris Corner Detection

Harris Corner Detection





Harris Corner Detection



3- In part 3, Firstly I wrote a homography function to compute homography. The homography function takes two matched points and the matrix is filled with these points as 2n x 9. N is equal to the number of matched points. In my example, there are 141 matched points. Firstly among these 141 matchedPoints1 and 141 matchedPoints2, four couple will be selected randomly for the homography and it will be repeated 200 times to find homography with containing the biggest inliers in RANSAC function. For finding number of inliers, there is ComputeInlierCount function which takes homography, matched points, number of matches, and inlier threshold. It uses project function. Project function takes a point comprised of x1 y1 with homography, then applies affine transformation with these parameters. The obtained point is returned as projected point. So when ComputeInlierCount function uses project, it finds distance between x2 y2 and projected point of given x1 y1, then if this distance is less than inlier threshold, it will be counted as inlier. In brief, project function will be applied for each x1 y1 in matchedPoints1 and then distance will be computed between each x2 y2 in matchedPoints and each projected points of x1 y1. At the end distance with less than threshold will be counted as inlier. Afterwards, with the best homography, all inlier points will be detected. At this time, instead of sending four random points, all inlier points will be sent to homography function. Then resulted homography and inverse of it will be returned. When we do not use RANSAC, there are 141 matched points and some of them are outliers like in figure 1. When we do use RANSAC, there are approximately 80 matched points and all of them are inliers like in figure 2.

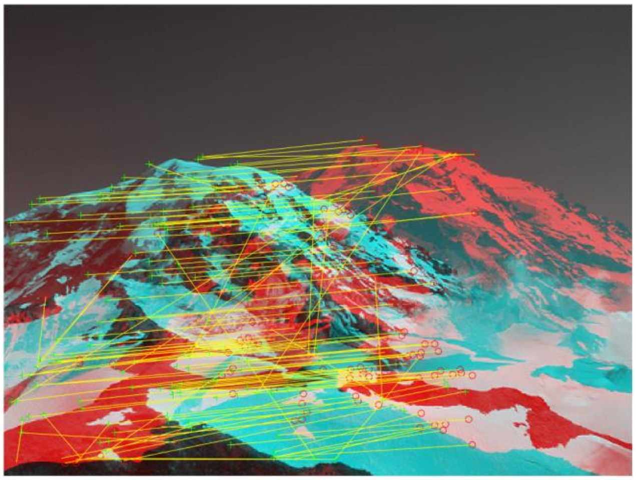


Figure 1

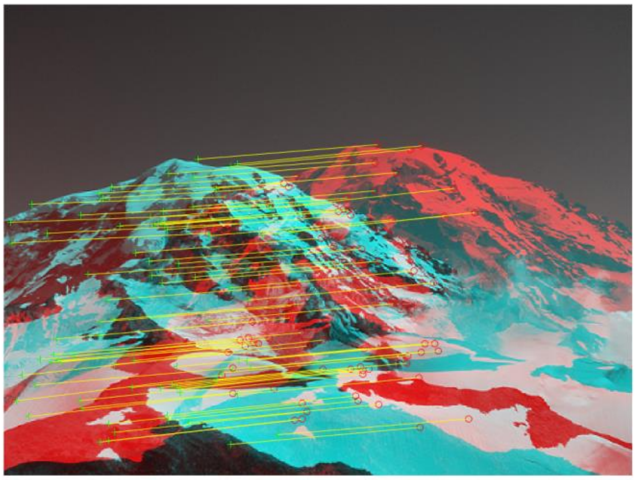


Figure 2