MVA 2008 – 2009

Object recognition and computer vision

Assignment 2: Stitching photo mosaics Report

Antoine Labatie

Parameters
Threshold on Laplacian: 20 for image in range 0:255
Threshold on matching point: Second Minimum>= 1.5* First Minimum
Number of RANSAC iterations: 500
Inlier Threshold in RANSAC: | [ax;ay]*H |² <= 0.0001

Algorithm

To detect the interest points I use the detector of the previous assignment which is called at the beginning with the function **features** on the three images nd returns the descriptors and the positions of the interest points. I also played on the Laplacian threshold in order to obtain a good number of points (I set it at 20 for images in range 0 to 255).

Then I call the function **match_desc** with the descriptors and their positions in two images as inputs. In the code I take the image 2 as the center image and as the reference. So every point descriptor in image 1 is matched with its nearest neighbour in image 2 if the condition on the minimums holds, and same for image 3. I tried to gain computation time here by computing at the same time the nearest and second nearest neighbours for each descriptor in image 1 but it did not seem to change a lot. I found this function may take a significant time to be run and that is why I restricted the number of points to 400 before calling it since 400 is enough.

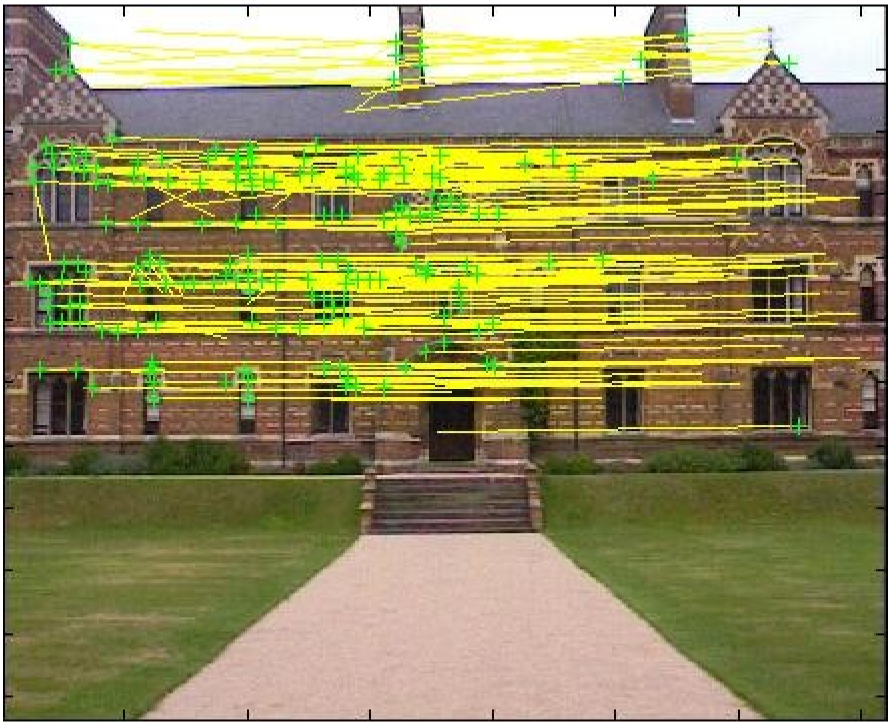
The function **RANSAC** takes as inputs the tentative correspondences stored arrays match12 and match 23 and iterates, finding the homography to minimize ||A*H|| with respect to H as in David Kriegman's paper. It stores at each iteration the number of inliers (defined by ||[ax;ay]*H||²<=0.0001 where [ax;ay] is the 2x9 matrix with the coordinates of the points in correspondence again as in David Kriegman's paper), and the index of the correspondences classified as inliers. It enables to compute at the end the homography to minimize ||A*H|| with A=[a1x;a1y;a2x;a2y;...;aNx;aNy] where the [aix;aiy] are the 2x9 matrices for the ith correspondence classified as an inlier.

Finally the images are warped as suggested with the function **vgg_warp_H.**

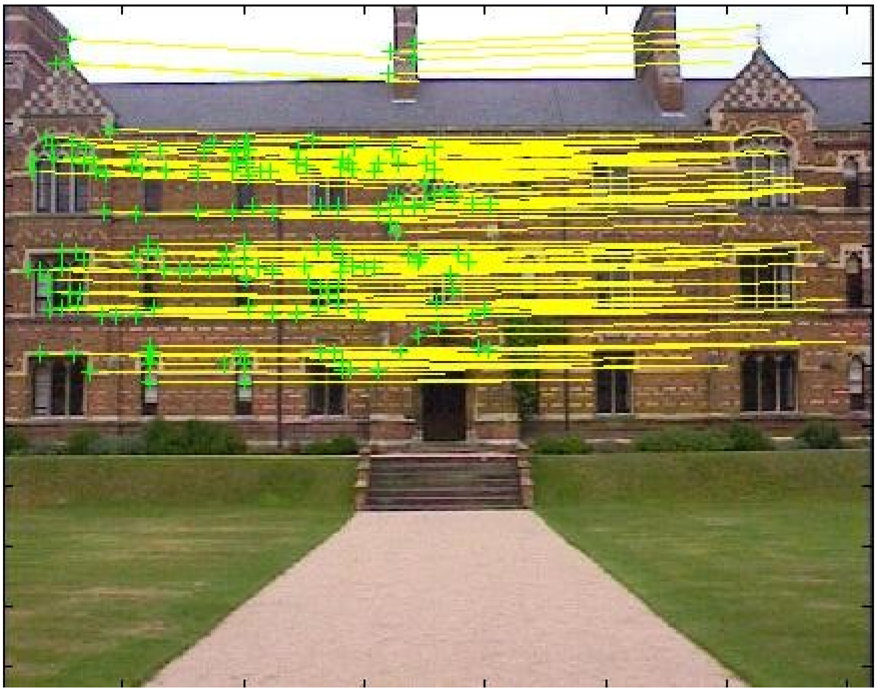
Results

Overall the results are good, espacially the number of correspondences and inliers stays high even in harder example like the one with the mountain in my test images. There is just a little problem on the Oxford Keble college since there seems to be an artefact at the cut, a vertical blue line, but I do not know where it comes from.

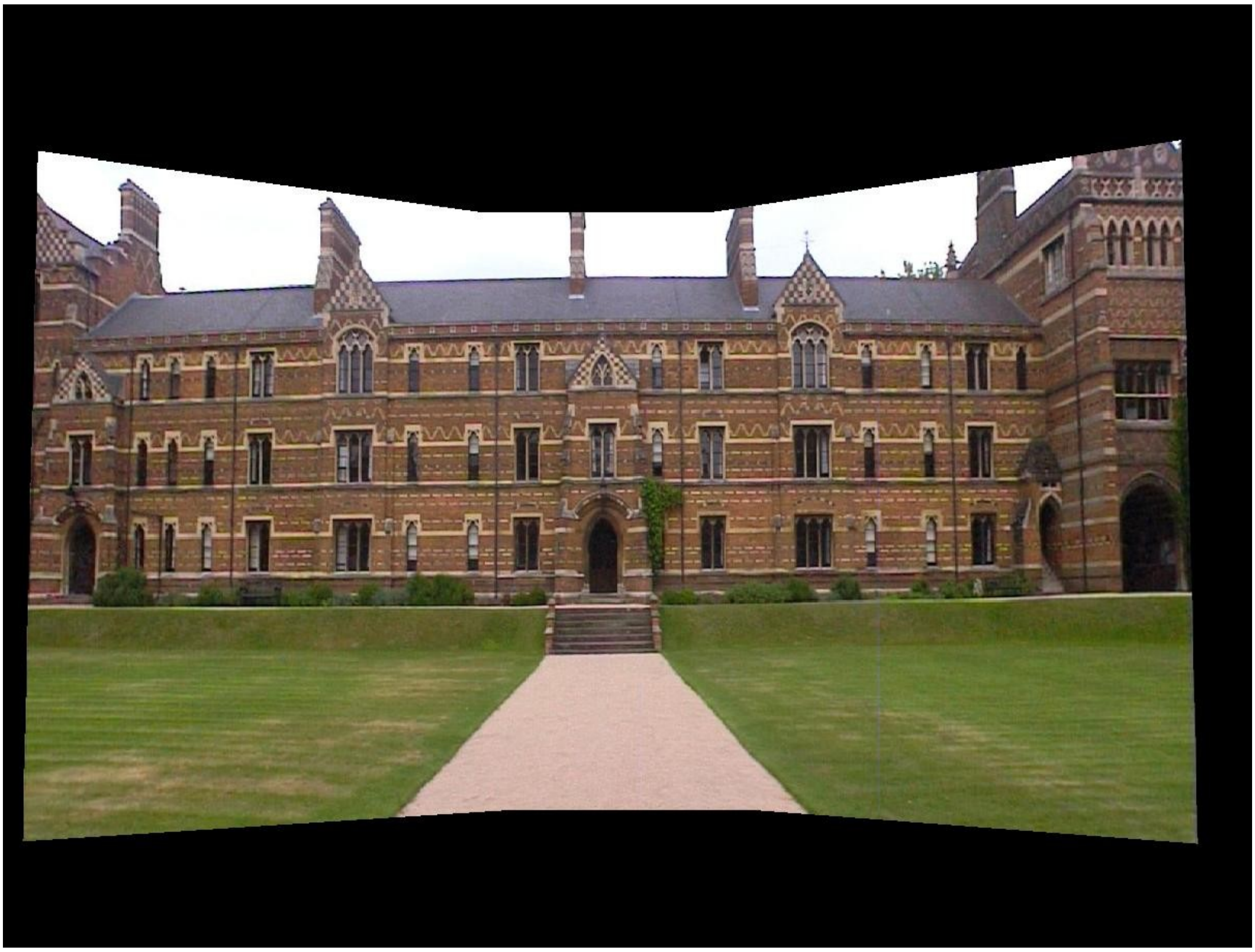
First example



Tentative correspondences

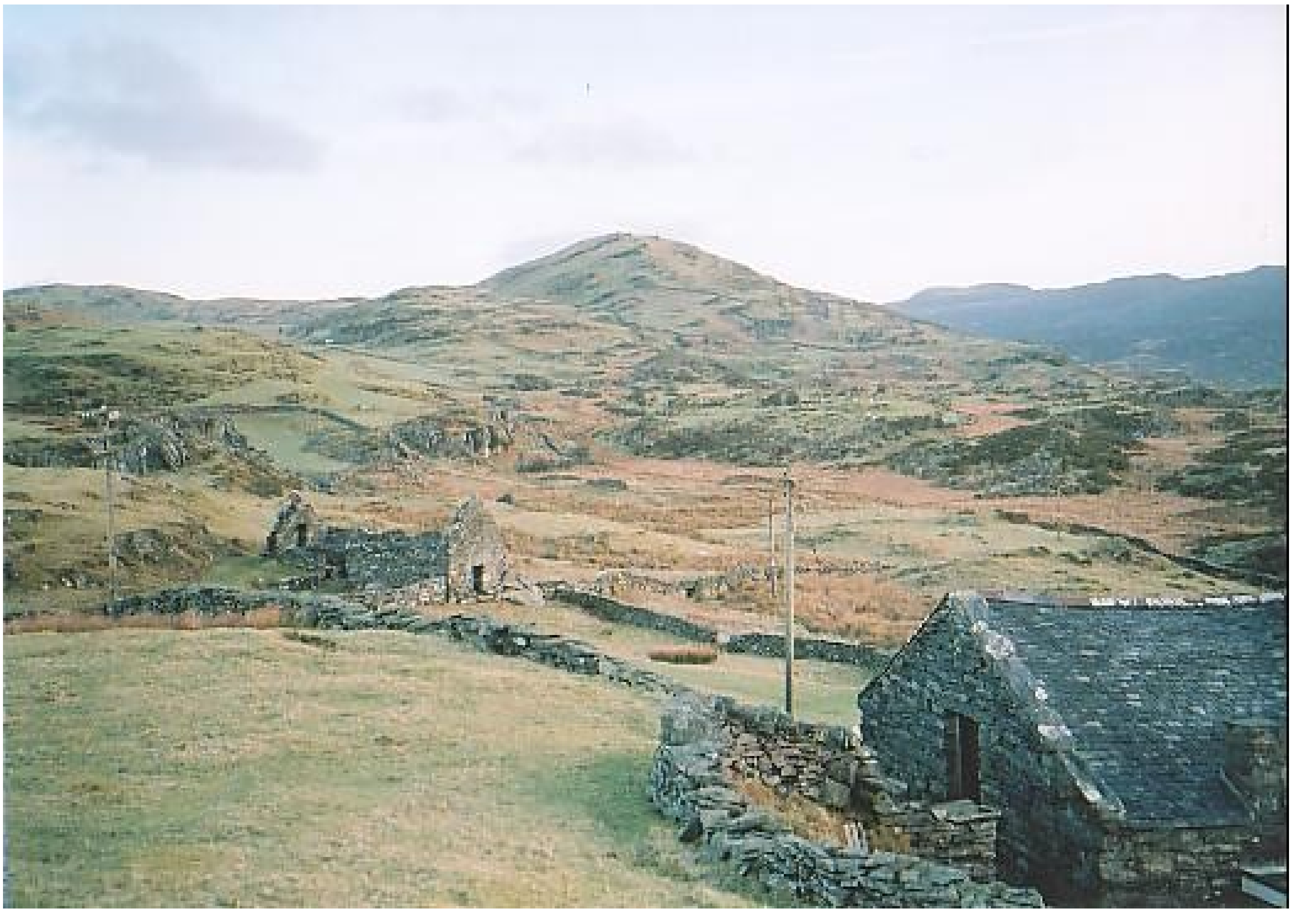


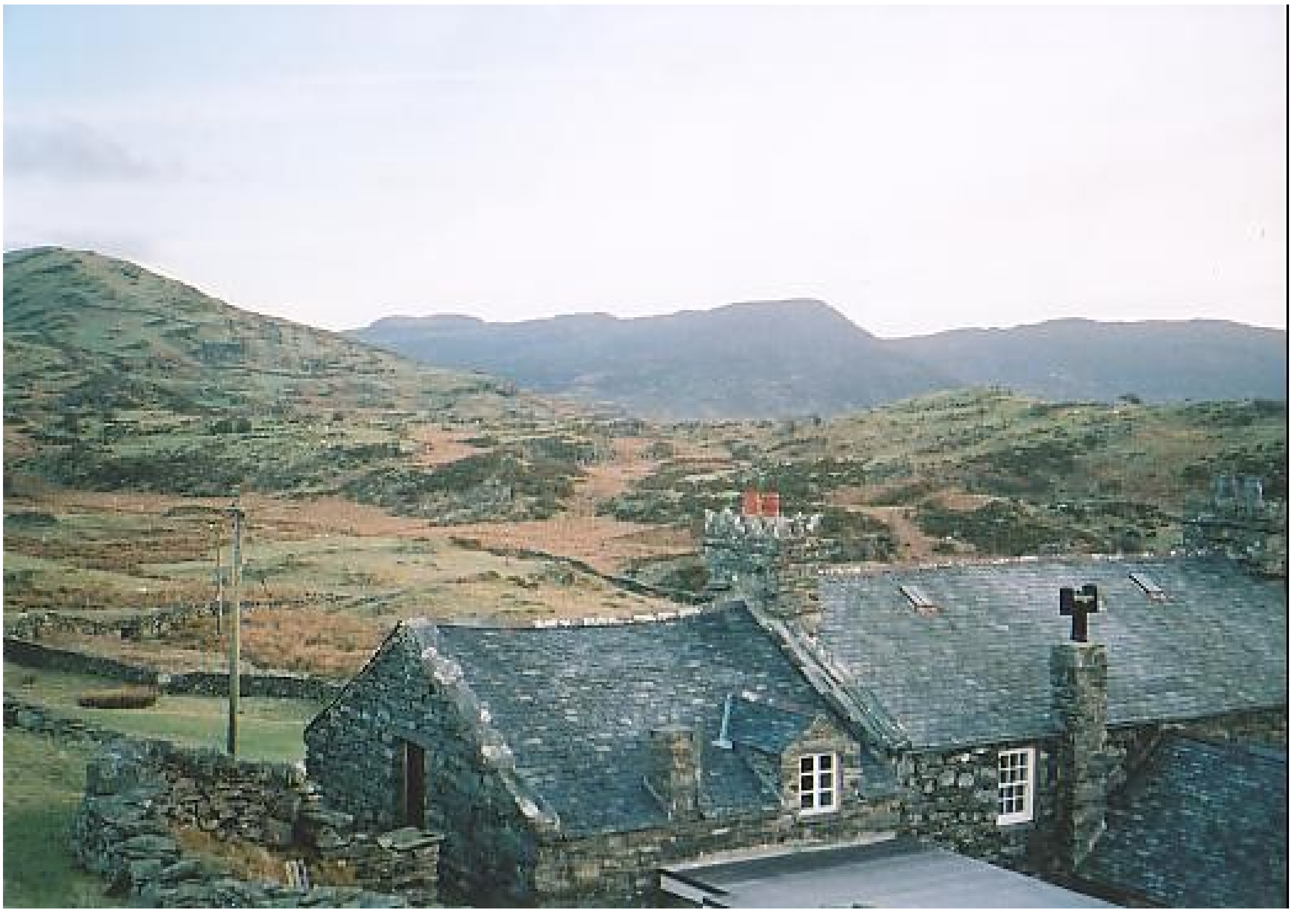
Inliers



Result

Second Example





Third example