# Computer Vision Ex1 – Solution

## The Problem

Plotting the given interest points on the images:

## Part A

1. Homography is defined by:

We learned in class that to solve an homography we need at least 4 pairs of matching points between two images (that are either only rotated or the in which the subject is planar) in order to get the homography. In addition, no linear dependent rows can be present in the constructed matrix.

For each pair of points we get two linear equations:

Where:

is a col vector representing the point in the src image ; () is the coords of the matching point in the dst image and are the rows of the homography matrix .

Similarly, we can write it in a different way:

Then:

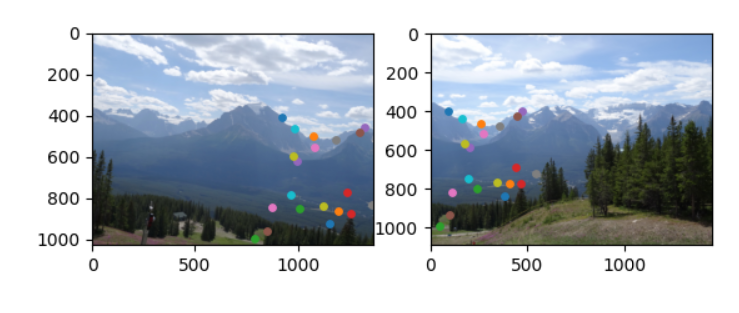
Put in matrix, we get the following eq:

As stated above, each matching pair will add two additional rows to the final matrix A.

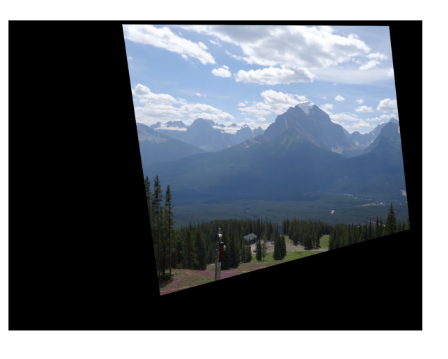
To get the conversion matrix from the equation system, we first need to calculate

Then we need to find the eigen vector that corresponds to the smallest eigen value, this can be done using SVD. The eigen vector that we got will be reordered to get the final homography matrix.

3. The “src” & “dst” images with the matching pointes



And after the forward mapping we got the mapped src image

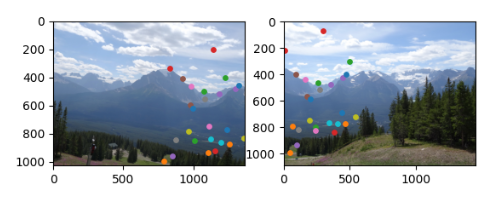


5. There are two problems with forward mapping:

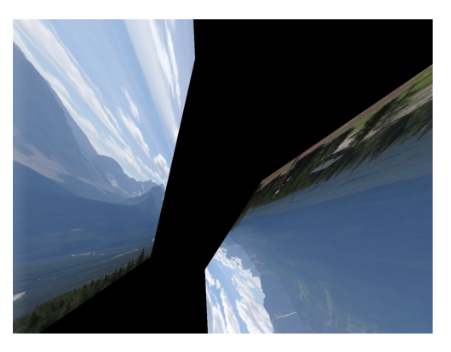
1. Fractional pixel coordinates - we are not going to necessarily fall on an integer pixel coordinates
2. Possible holes in the output – there is no guarantee that we are going to fill all the pixels in the target image since there may be two pixels in the src image that fall on the same pixel in the target image

6. In this section we assume a “naïve” approach and because of the outliers matching point in the “matches.mat” file we got an incorrect image transformation

* “matches.mat” - Matching pointes



* Src image transformation



9. Suppose there are 30 match points and it is known that 80% of them are correct.

The formula we developed in class is the following:

Where

* p – RANSAC success probability that we want (90%, 99%)
* k – number of iterations required to achieve RANSAC success probability of p
* w – probability to pick a correct matching (0.8) probability to choose a correct matching
* n – number of minimal matching needed to calculate an homography (4)

So we have for **90% confidence**:

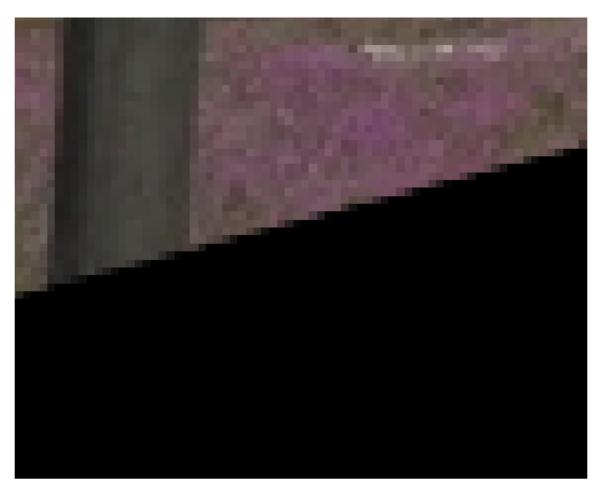
And for **99% confidence:**

10. we can see that in this section we got an image that is similar to the one we got after mapping with the perfect matching points. This tells us that RANSAC worked well at ignoring the outliers from the “contaminated” matching points file.

**But when we compare the result to section 4 when we had a perfect matching points we can see no difference and this is because that now the RANSAC does not eliminate any outlier.**

* **Src imaged mapped after RANSAC**

11. We can see in the two images below that the Backward Mapping transform, which uses Bi-linear interpolation has improve our resolution in compare to the forward mapping image from the previous sections

* Forward src imaged mapping zoomed  
  
* The Backward Mapping transform, which uses Bi-linear interpolation src mapped imaged  
  

12. The great panorama image



13. Our own Panorama images

* Src\_test   
  
* Dst\_test  
  
* Panorama - **add image**