Homework Assignment 1  
Mai Dabas (315026294) & Adi Green (313472417)  
Computer Vision

**Part A: Homography computation**

1. As seen in the lecture, the system of equations for **projective** transformation is:

Each point in the source coordinate system matches a point in the destination system.

To get the conversion matrix, we build the following equation system,

where:

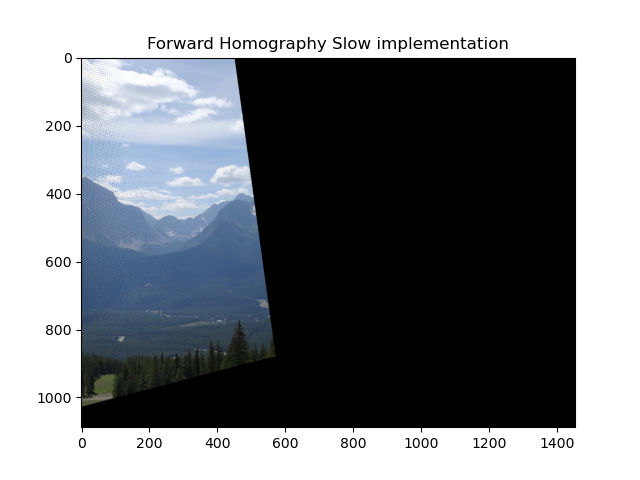
From these two equations, we can build this equation:

And find H such that AH=0, while adding an extra constraint |H|=1 to avoid the obvious solution of H being all zeros. To estimate H, we need at least 4 points (eight equations from the four points, and since H is up to scale, the last equation is a normalization constraint). If there are more than 4 points (as in our case), we can just keep plugging them as new rows in matrix A (to new rows per point). Then we can use SVD (, and select the smallest singular vector of  as the solution to H. To get the matrix form of 3x3 we reshape this vector.

1. In the code.
2. The results:

**Part A2: Forward mapping slow and fast:**

1. In the code.



1. In the code.

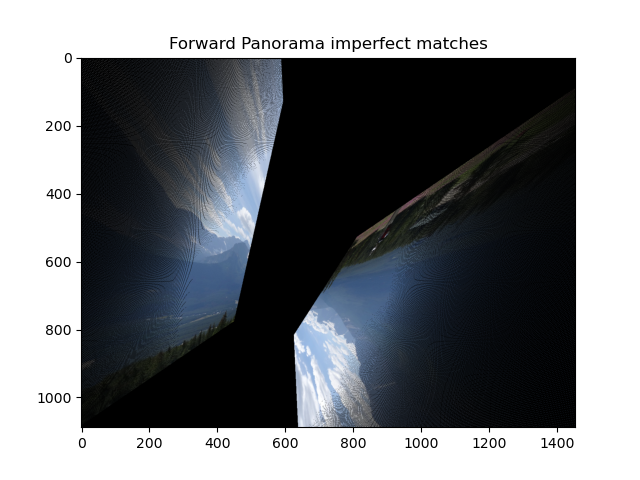
Graphical user interface

Description automatically generated

1. The forward mapping suffers from 2 issues:
   1. Integer coordinates can be mapped into non-integer coordinates.
   2. Leaving black pixels in the target as there is no guarantee that the mapping will fill all pixels in the target image.

In our images, we rounded the pixels to deal with problem a, but you can see in the images above black dots in some areas of the image which is cause by problem b.

1. The resulting image:

****

In this case, the image is not as we expected. Due to some matching points being wrong the result is distorted. In (3) we calculated the homography matrix when the matching point are perfect, and we can see that in this case where the matching points has outliers the homography matrix is also different.

**Part B: Dealing with outliers**

1. In the code.
2. In the code.
3. In class, we saw the following formula to find k –no. of iterations required to fit the model:

when:

As for a) :

Meaning we need ~5 iterations (rounding up to avoid getting less than the desired success rate).

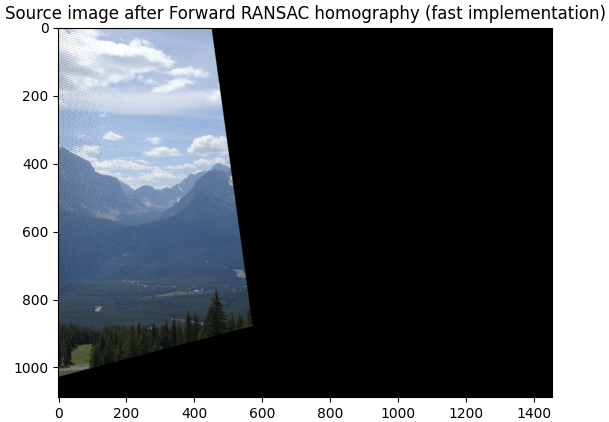
As for b) :

Meaning we need ~9 iterations (rounding up to avoid getting less than the desired success rate).

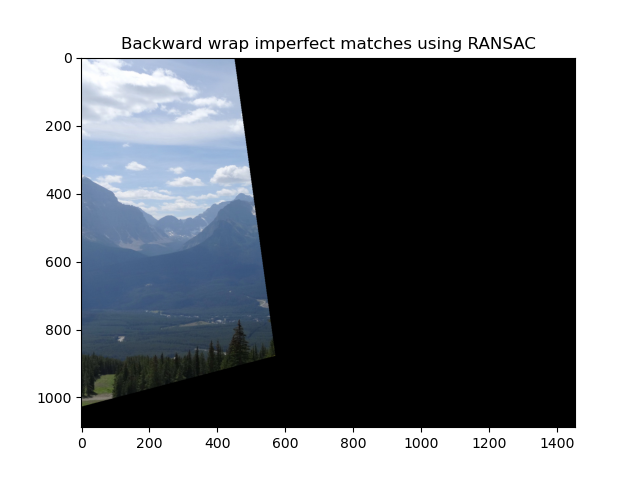
In order to cover all options, given 30 match points, assuming we don’t choose the same group of four points more than once, iterations are needed.

1. In the code.
2. RANSAC homography:

The source image after projective transform using forward mapping (fast implementation):



The homagraphy transformation is identical (with negative sign) to the one presented in section 3 where we used perfect match points, so the purpose of the RANSAC algorithm to remove outliers and gain a cleaner result has worked. The image is also identical to the image gained when using the perfect matches (section 5) as opposed to the image gained when there were outliers in the match points. (section 7) – also implying the RANSAC has worked.



Using interpolation and backward wrap has filled the black “holes” that we can see in section 12 – with is one of the main issues that forward mapping has.

1. In the code.
2. In the code.