**Computer Vision assignment 1**

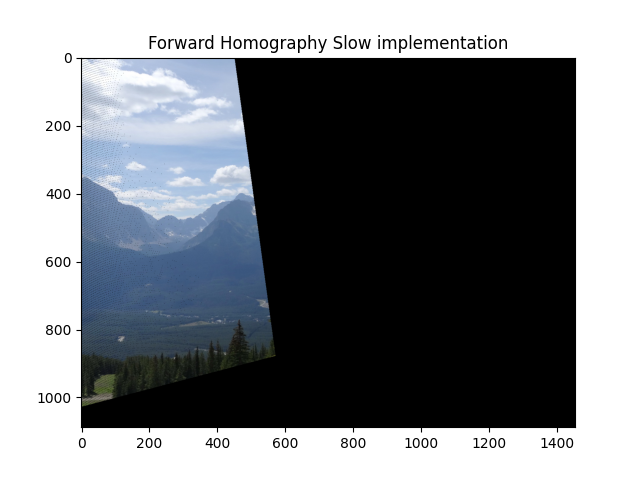
Adir Eliza 308124890  
Oran Ridel

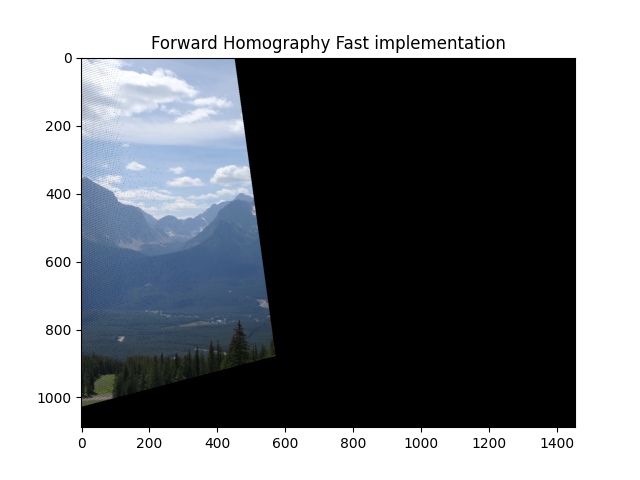
**1.** For Projective to be found, we need to find 8 parameters (because of the in variation to multiplication). Therefore 4 points are needed to solve it:  
For each set of points and , we have , Where . We get:  
These are the 2 constrains we have from one pair of points. If we write it as a system of equations, we get:  
(where is the i-th row of H).

**3.** These are the coefficients for Naive Homography:  
 [[ 1.12313764e-03 1.64757802e-04 -9.99919579e-01]

[ 1.05119761e-05 1.05462486e-03 -1.25626989e-02]

[ 2.96941051e-07 4.35711334e-08 7.82906877e-04]]

**4.** The image we got for the slow forward mapping:  


**5.** The image we got for the fast forward mapping:  


**6.** The problem is that the new pixel coordinates after multiplication with the Homography matrix will not always gives us an integer, therefore there might be a problem with the exact location we suppose to assign the pixel in the new image, also this can cause a problem when while rounding the float number to get an integer we can get that two pixels will be assigned to the same coordinate in the new image and will override one another.

**7.**   
A graph showing the sky and clouds

Description automatically generated with medium confidence  
As can seen above, we got a result for the second matches.mat matrix which contains mismatching points. We got that these points affected the final result because the Homography matrix was computed with false data and therefore gave us wrong results. This resulted in weird orientation and shape of the translated photos.

**10.** The equation for calculating the number of iterations needed in order to find all model’s parameters for RANSAC is given by:

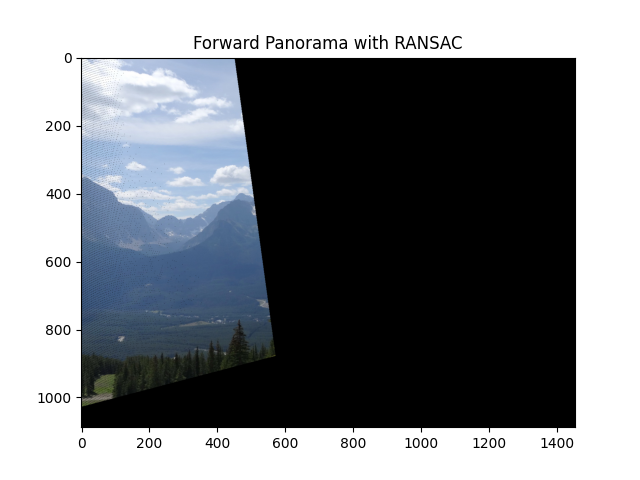
We got the following parameters:  
w = 0.8  
n = 4

The number of combinations we get from 30 match points when 4 is needed to configure a model is:

**12.** These are the coefficients we got with RANSAC:  
[[ 1.12681441e-03 1.56712736e-04 -9.99903291e-01]

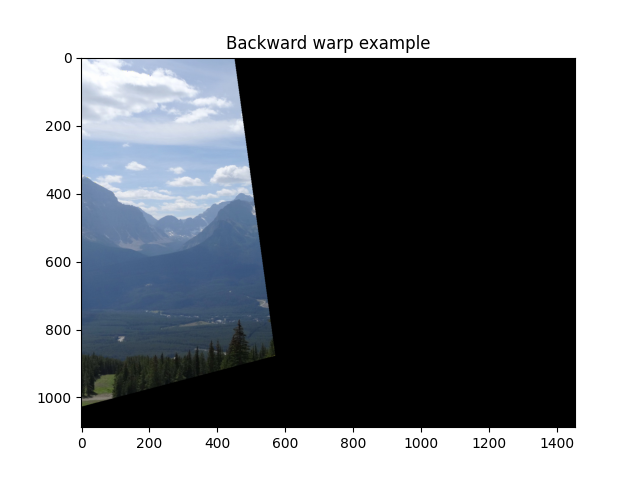
[ 1.70119655e-05 1.04170502e-03 -1.37993949e-02]

[ 3.07084612e-07 2.76208079e-08 7.78621993e-04]]

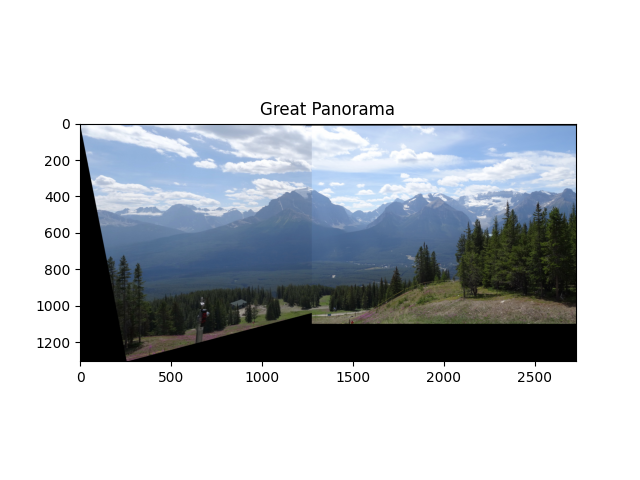


As seen in the picture we got pretty good results and a lot of the pixels seems to be at the same place as we got earlier, but you can also notice that there are black dots the appearing in the up-left side of the transformed pixels. This is due to the fact that we are using RANSAC for dealing with outliers while using the second matrix that contains bad matching points as well.

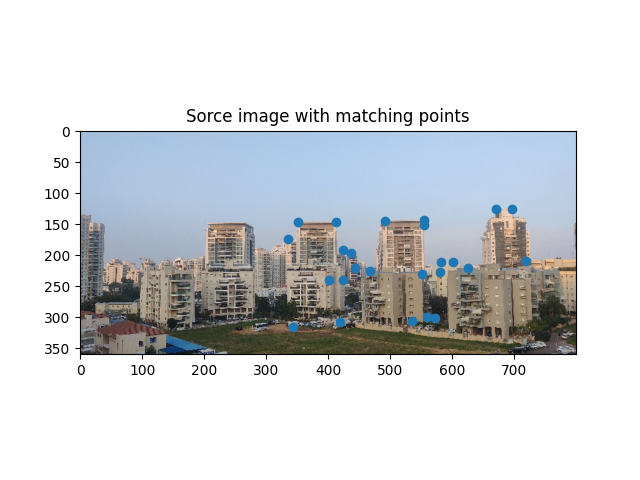
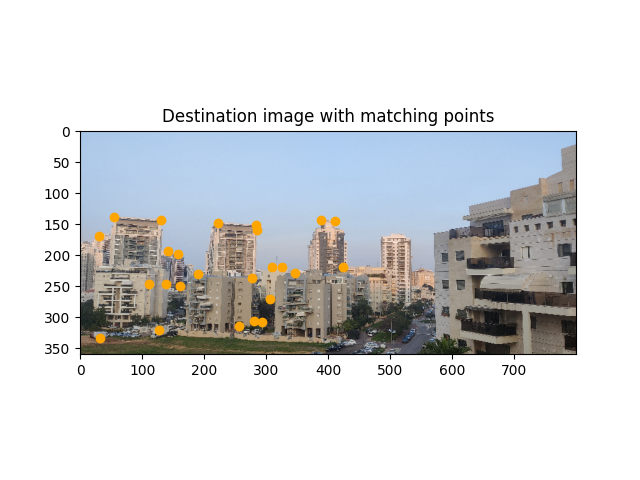
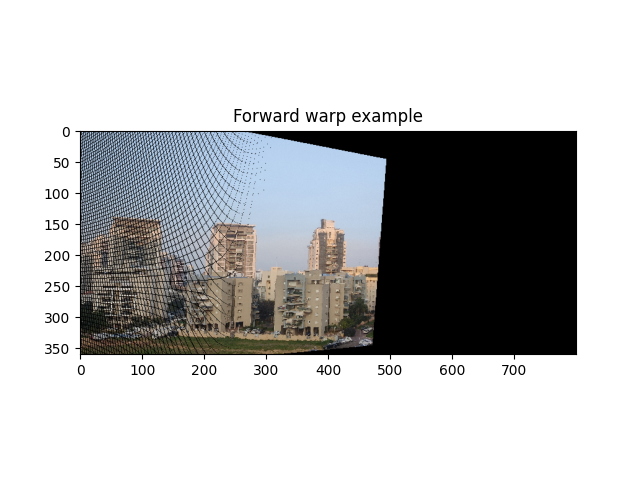
**13.** We got the following results after using backward mapping



As can be seen in the picture we get better results for backward mapping. We don’t see the black dots that appeared in section 12 with forward mapping.

**16.** This is the Panorama image we got using the following methods:  


A view of a city from a distance

Description automatically generated**17.** These are the results we got for two images of our own choice:

A view of a city from a window

Description automatically generatedA panoramic view of a city

Description automatically generated