

# Computer Vision assignment 1

Adir Eliza 308124890

Oran Ridet

1. For Projective  $H = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$  to be found, we need to find 8 parameters (because of the in variation to multiplication). Therefore 4 points are needed to solve it:

$$x_1 \leftrightarrow x'_1$$

$$x_2 \leftrightarrow x'_2$$

$$x_3 \leftrightarrow x'_3$$

$$x_4 \leftrightarrow x'_4$$

For each set of points  $x_i$  and  $x'_i$ , we have  $x'_i = Hx_i$ , Where  $x_i = \begin{pmatrix} v_i \\ u_i \end{pmatrix}$ ,  $x'_i = \begin{pmatrix} v'_i \\ u'_i \end{pmatrix}$ . We get:

$$\begin{cases} \frac{u'_i}{1} = \frac{H_2x_i}{H_3x_i} \\ \frac{v'_i}{1} = \frac{H_1x_i}{H_3x_i} \end{cases} \rightarrow \begin{cases} u'_iH_3x_i - H_2x_i = 0 \\ v'_iH_3x_i - H_1x_i = 0 \end{cases}$$

These are the 2 constrains we have from one pair of points. If we write it as a system of equations, we get:

$$\begin{bmatrix} x_1^T & 0 & -v'_1x_1^T \\ 0 & x_1^T & -u'_1x_1^T \\ x_2^T & 0 & -v'_2x_2^T \\ 0 & x_2^T & -u'_2x_2^T \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} H_1^T \\ H_2^T \\ H_3^T \end{bmatrix} = 0$$

(where  $H_i$  is the i-th row of H).

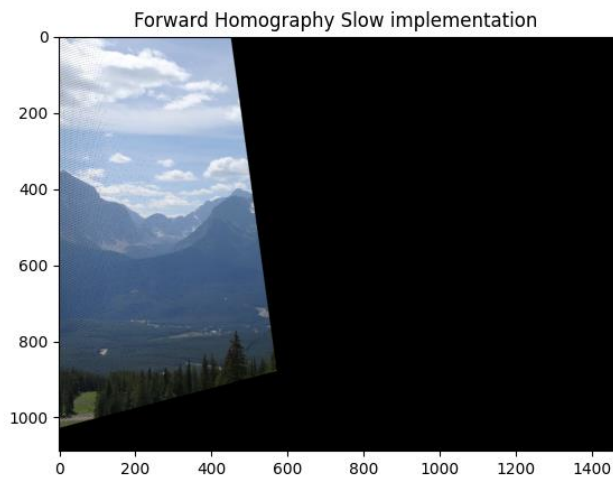
3. These are the coefficients for Naive Homography:

[[ 1.12313764e-03 1.64757802e-04 -9.99919579e-01]

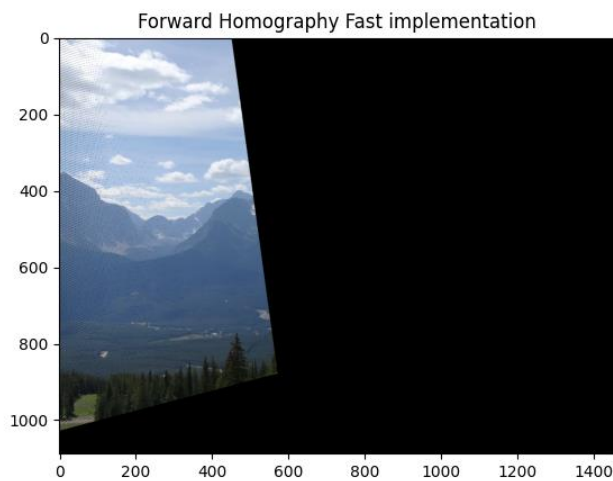
[ 1.05119761e-05 1.05462486e-03 -1.25626989e-02]

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

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

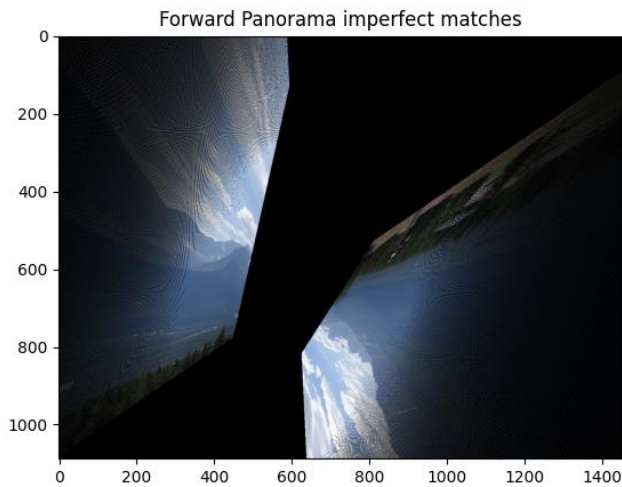


5. The image we got for the fast forward mapping:



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.



As can be 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:

$$k = \frac{\log(1 - p)}{\log(1 - w^n)}$$

We got the following parameters:

$w = 0.8$

$n = 4$

$$k(p = 0.9) = 4.36 \rightarrow 5$$

$$k(p = 0.99) = 8.73 \rightarrow 9$$

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

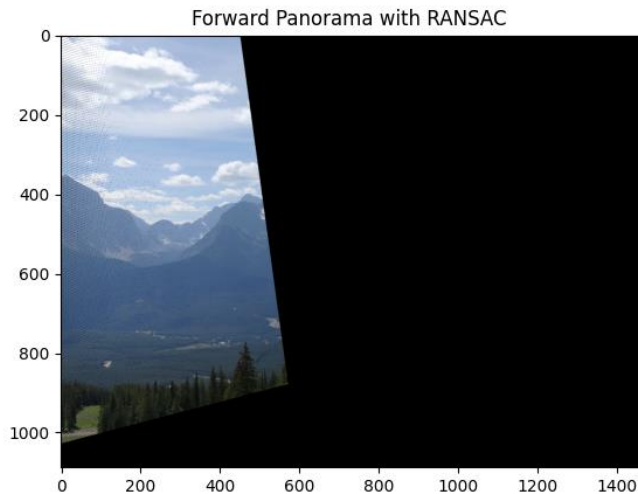
$$\binom{30}{4} = 27405 \text{ combinations}$$

**12.** These are the coefficients we got with RANSAC:

[[ 1.12681441e-03 1.56712736e-04 -9.99903291e-01]

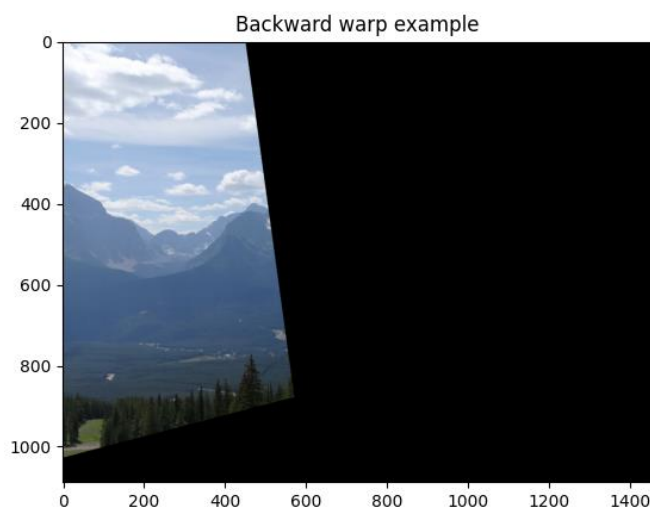
[ 1.70119655e-05 1.04170502e-03 -1.37993949e-02]

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



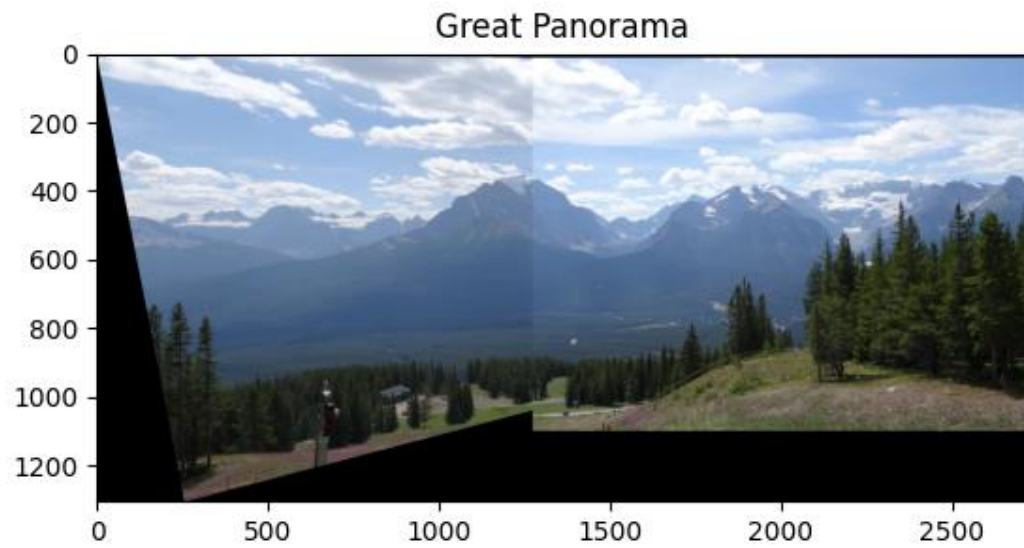
As seen in the picture we got pretty good results and a lot of the pixels seems to be at the same place as we got earlier, but you can also notice that there are black dots the appearing in the up-left side of the transformed pixels. This is due to the fact that we are using RANSAC for dealing with outliers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

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



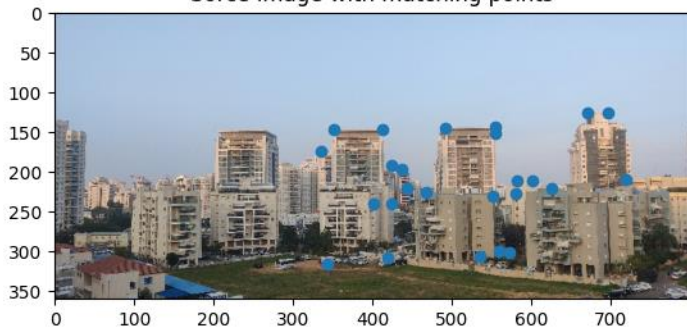
As can be seen in the picture we get better results for backward mapping. We don't see the black dots that appeared in section 12 with forward mapping.

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

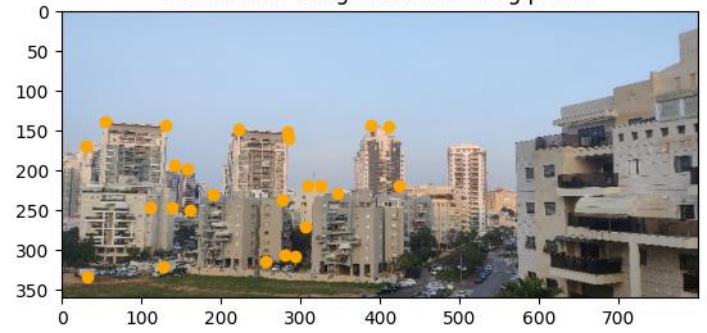


17. These are the results we got for two images of our own choice:

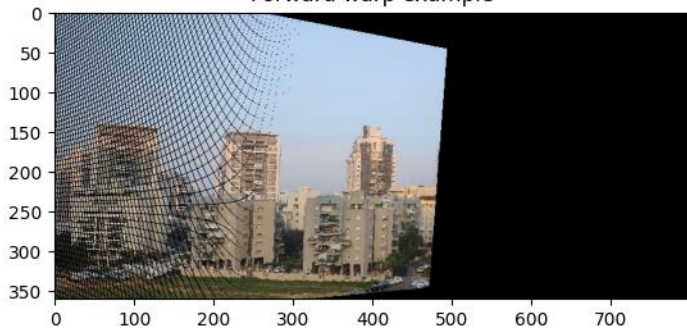
Source image with matching points



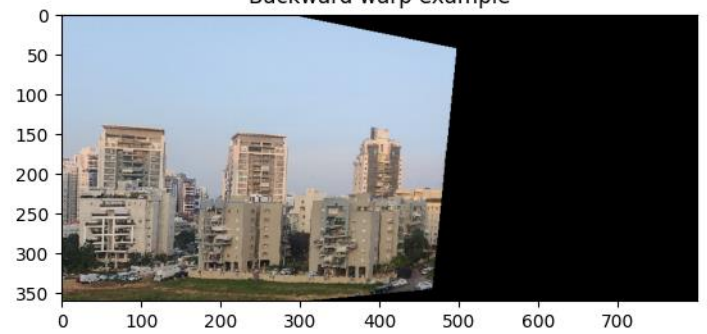
Destination image with matching points



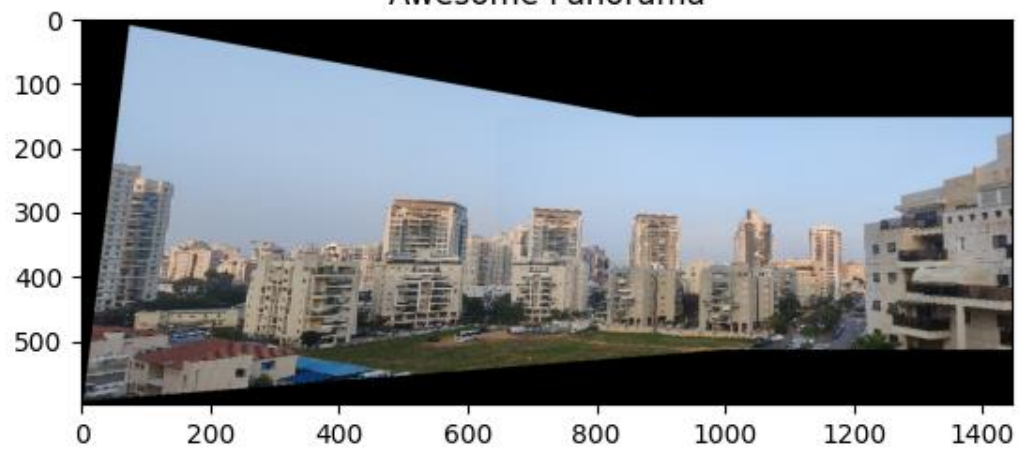
Forward warp example



Backward warp example



Awesome Panorama



Reversed Awesome Panorama

