Lab Assignment 1: Homography estimation and interpretation

Homography computation

1. **Exercise: homography defined by four pairs of transformed points**
   I used the maketform with 'projective' for both methods.

   The applied homographies can be seen in the following figures. They are not equal but proportional to each other.

tform_1.tdata.T

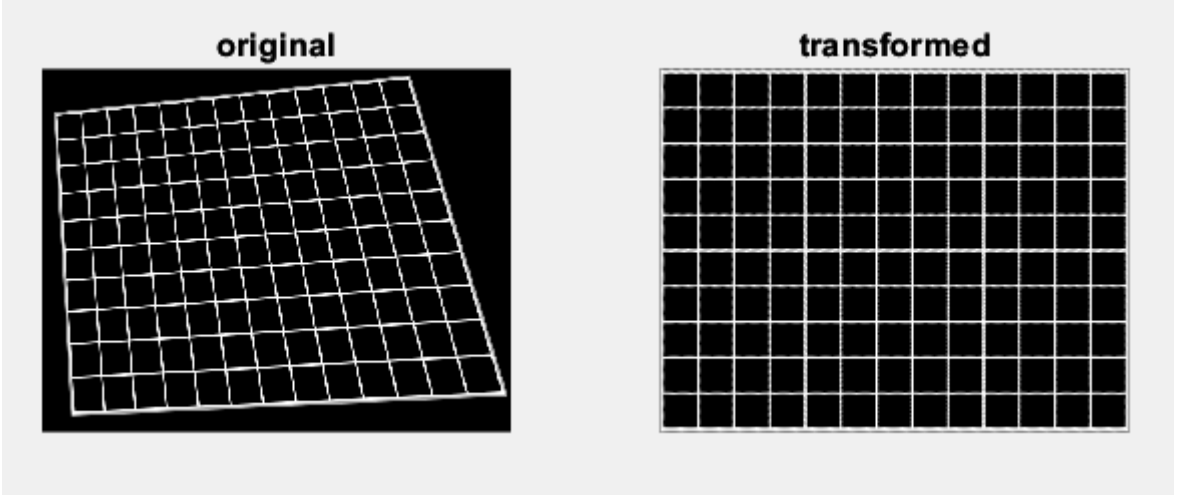
	1	2	3
1	1.4064	0.4004	8.6289e-05
2	-0.1307	1.8918	6.0428e-04
3	-6.9449	-224.0339	1

tform_2.tdata.T

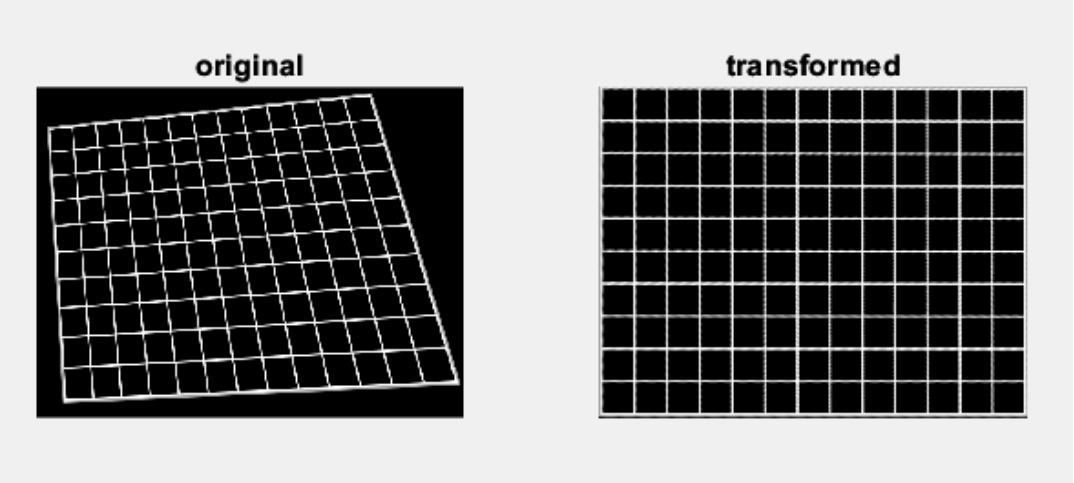
	1	2	3
1	0.0063	0.0018	3.8495e-07
2	-5.8313e-04	0.0084	2.6958e-06
3	-0.0310	-0.9995	0.0045

   I calculated the energy difference and for both transformations the result was 7.2005e+03.

2. **Exercise: problems in the definition of a homography from four pairs of transformed points.**
   I loaded the objective image', and calculated the error between the obtained image and this objective image with get_error_energy_vmmc. The calculated error was null and the images are shown in the following picture.

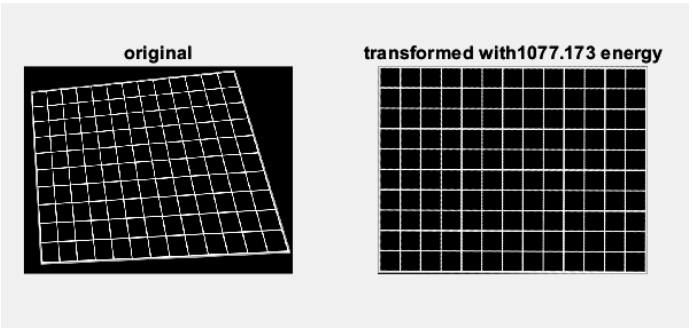


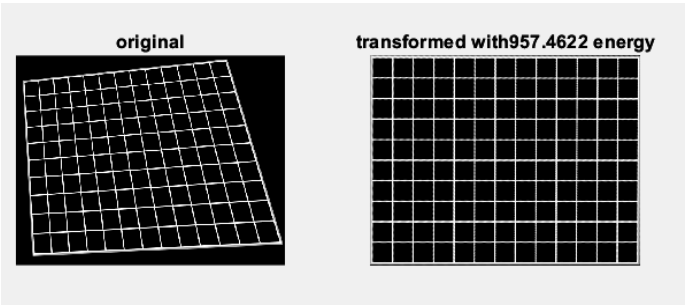
After that I manually selected the corners using the get_user_points_vmmc function. I was not able to even get close to zero energy. The picture below shows a result with energy of 3.4504e+03.

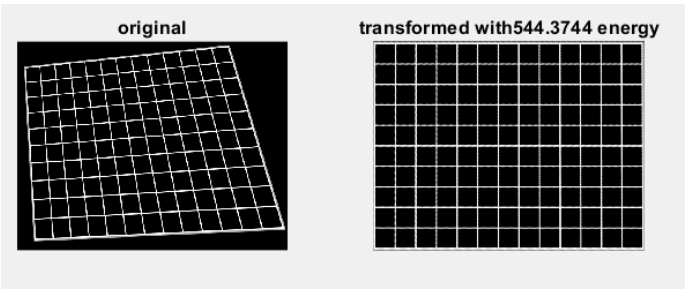


3. **Exercise: homography that best fits more than four pairs of transformed points**

Nº points:	**4 points**	**8 points**	**12 points**
Error:	**1.077173e+03**	**9.574622e+02**	**5.443744e+02**







It is clearly visible from the table that the more points are marked, the better.

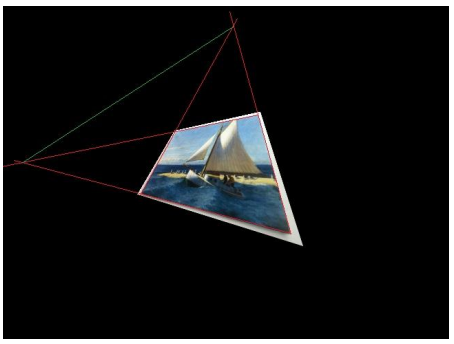
For the second part of the exercise.

Nº realization	1	2	3	4	5
Error:	43.2489	58.6847	34.2679	51.4144	64.6063
Nº used points:	1021	1021	1021	1021	1021
Nº iterations:	5056	5214	2472	2214	1639
Nº inliers	258	176	222	218	235

Homography interpretation

4. **Exercise: vanishing points and the line at infinity**

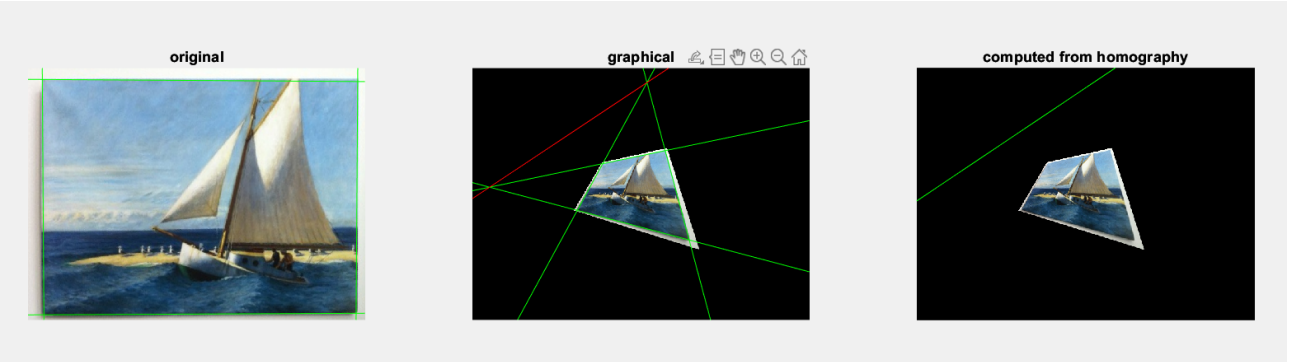
The result picture of the graphical



The result of the process obtained by calculating for the graphical points shown in the middle and the one which was computed from homography is on the right. I've got the homography matrix from the homography_solve_vmmc function. As an example I used the points from the reference image selected by the get_user_points_vmmc function manually and inserted lines for the four quarter point. To get the lines I created the x as a linspace from 0 to the second dimension of the image. To create the lines I calculated from the equation $y = m * x + c$.

The slope (m) calculated from $m = rise/run = (y_2 - y_1)/(x_2 - x_1)$, where (x1, y1) and (x1, x2) are two corner points of the picture. Then the c is calculated from $y_2 - m * x_2$. The plotted lines are seen in the picture on the left. After I've made the same calculation for the modified picture (hopper_vanishing.bmp). From the intersections of the lines I calculated and visualized the connecting line, which can be seen in the picture on the right with red. To

calculate the intersection of the lines I used the function lineintersect. Source of the function https://es.mathworks.com/matlabcentral/fileexchange/30502-find-intersection-of-two-lines
For the image on the right I used the following equation, where H is the homography matrix: $y = (H(3,1) * x + H(3,3))/(-H(3,2))$. The line that I got graphically and the line from the calculation are quite the same.

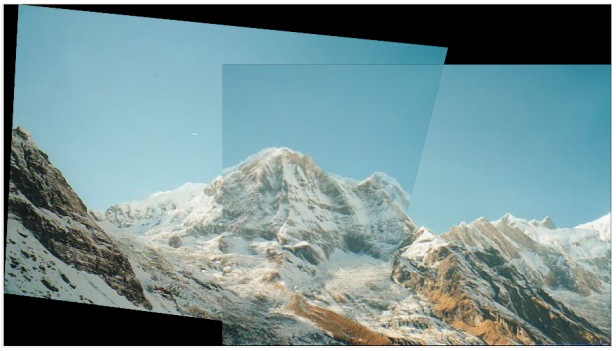


Generation of panoramic images

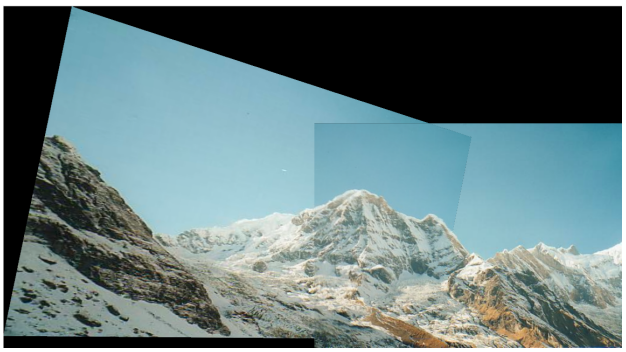
5. **Exercise: combining two images in a panorama**

	pan_demo_1 - pan_demo_2		**mountain1 -mountain2**	
	Nº Points	Quality	Nº Points	Quality
th=0.01	725 (best matches 0)	no picture	1638 (best matches 137)	medium
th=0.05	725 (best matches 112)	low	1638 (best matches 401)	medium
th=0.1	725 (best matches 346)	low	1638 (best matches 1019)	medium
th=0.5	725 (best matches 725)	medium	1638 (best matches 1638)	high
th=1	725 (best matches 725)	medium	1638 (best matches 1638)	high

Medium **High**



Low

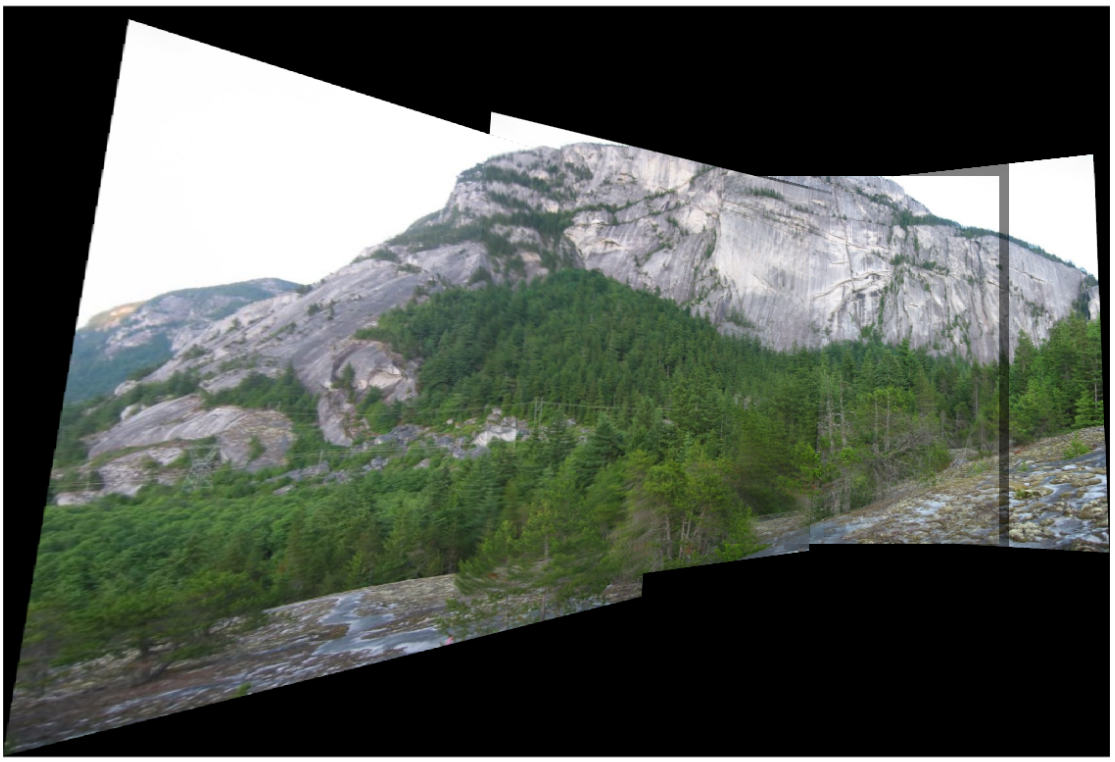


Medium

6. Exercise: combining several images in a panorama

For the threshold I used 0.5 for both the 3 and 0.1 for the 4 picture panoramas. It surprised me how long calculation time was needed but I was satisfied with the overall result. The code was able to find the feathers and connect them.

Panorama with 3 pictures
Number of points = 3280 (best matches 3247)



Panorama with 4 pictures
Number of points =3237 (best matches 3192)