Image analysis and computer vision

Homework

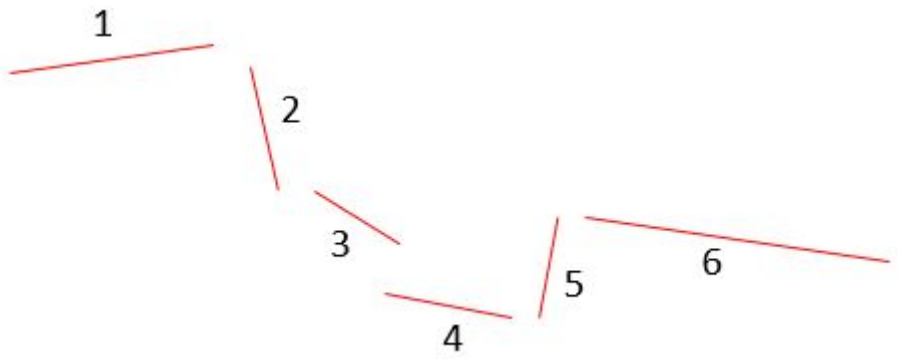
2020/2021

Camnasio Samuele

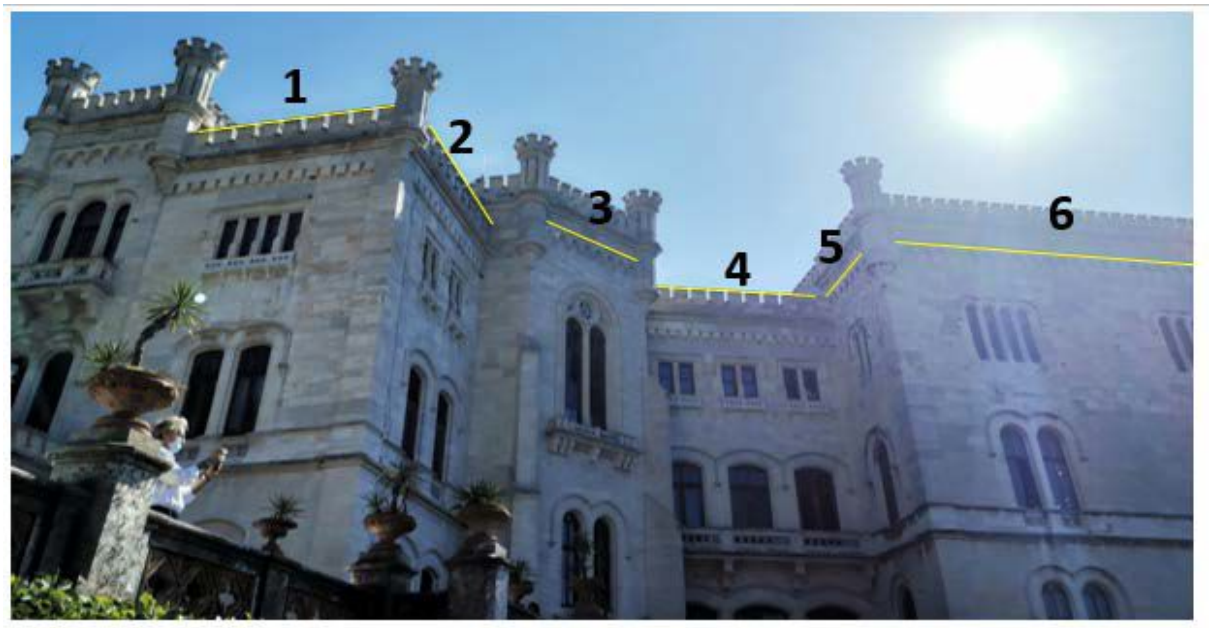
Contents

1 Introduction to the problem

The given image is a picture of the Castello di Miramare, Trento, Italy, taken with a digital camera, with a skew factor assumed to be null and with unknown aspect ratio, principal point and focal distance.



Known constraints on an horizontal plane Π on the real castle (i.e. lines 1 and 2 are orthogonal, as well as lines 4 and 5, and lines 5 and 6)



The plane Π on the given image

Given the constraints shown on the pictures above, a series of operations have to be applied to the image, as will be shown in the next chapters

2 Image processing

F1 Feature extraction

In this section is described the process of extraction of useful features from the image, in particular edges, straight lines and corners.
Before starting some preprocessing is made on the image, in particular, the image is resized due to computational complexity, in fact unfortunately it was hard to work on matlab with a full size image with the available hardware, especially for the computation of the transformations in the rectification part; secondly, to extract some features it's been useful to increase the contrast on the black and white image, the following are the images before and after adjustment.

Black and white image

Black and white image with contrast adjustment

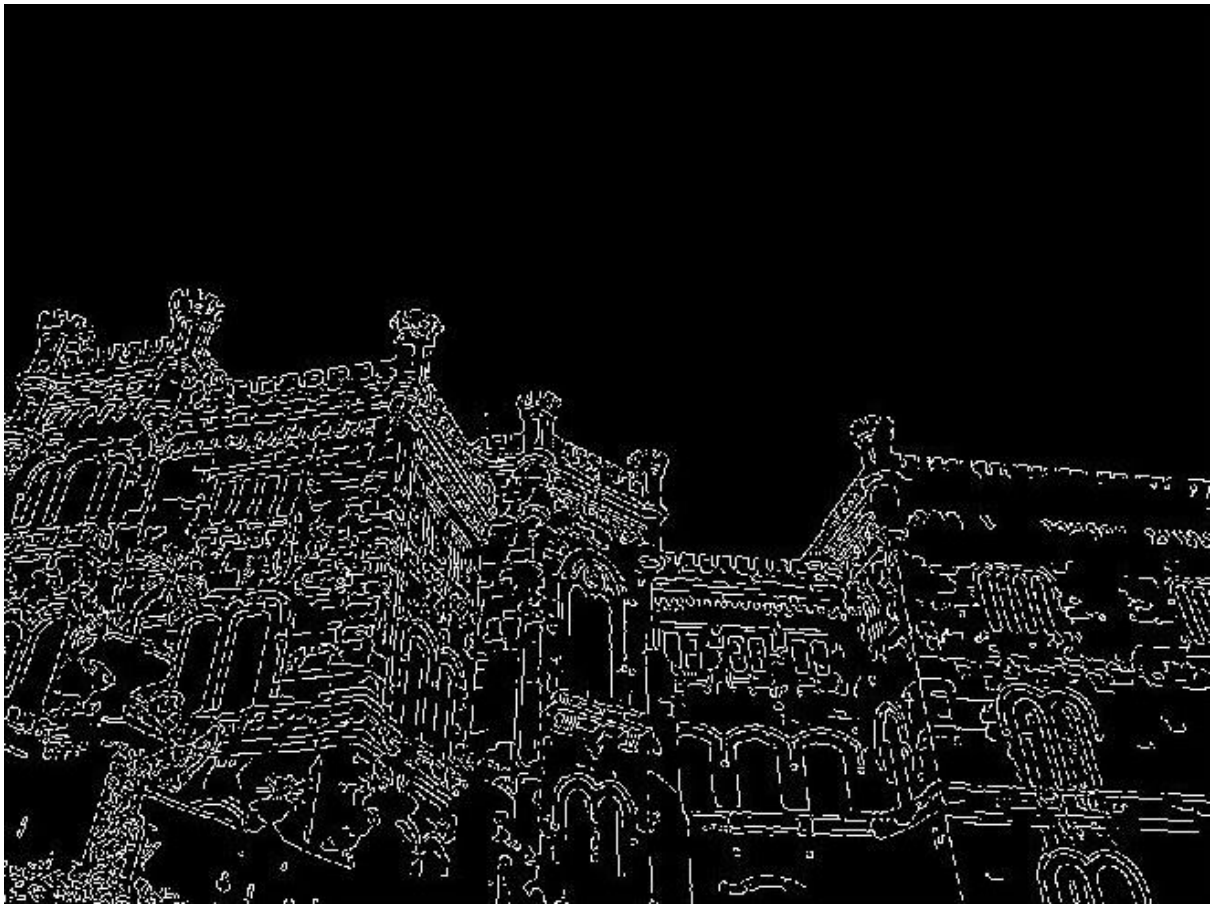
F1.1 Edge detection

Edges are extracted from the black and white adjusted image with the method ‘canny’:

```
edges = edge(I, 'canny',threshold);
```

where the `threshold` parameter ignores all edges that are not stronger than `threshold`.

The following is the result of the canny method

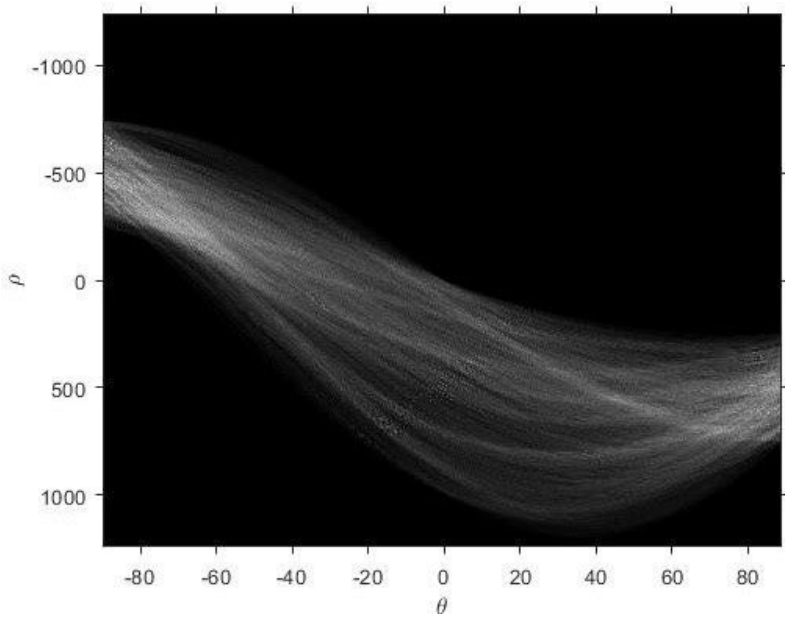


Edges found by canny method

F1.2 Identification of straight lines

Line segments in the image are extracted with the Hough Transformation, starting from result of canny edges, the first step is creating the hough transform:

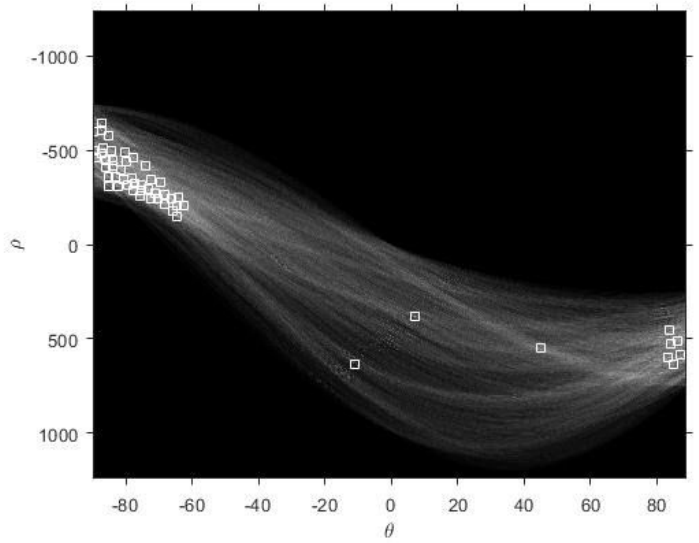
```
[H,T,R]=hough(edges,'RhoResolution',1,'Theta',-90:0.1
:89);
```



Hough transform

Then the peaks are extracted from the Hough transform:

```
P=houghpeaks(H,numPeaks,'Threshold',0.7*max(H(:)),
'Theta', -90:0.1:89);
```



Hough peaks

And finally line segments are found:

```
lines=houghlines(edges,T,R,P,'FillGap',10,'MinLength'
,30);
```

Due to the light exposure of the image and the large number of lines on it, it was not easy to extract all the meaningful lines with a fixed set of parameters in the Hough method, the following image is the result of the Hough procedure with given `Threshold`, `Fillgap` and `MinLength` parameters , playing with different values of these parameters it is possible to obtain all the desired feature, a set of useful features extracted is provided in the next chapters

Result of the Hough procedure

F1.3 Corner detection

Corner features are extracted with Harris–Stephens algorithm

```
corners = detectHarrisFeatures(I,'MinQuality', 0.0001);
```

`'MinQuality'` attribute is set to a low value due to the difficulties to detect some corner in the image.

Corners extracted with Harris algorithm

F1.4 Combination of results

Combining the previous methods the most meaningful features are selected, the following is a representation of a set of features that will be useful during the next steps.

Subset of image features

3 Geometry

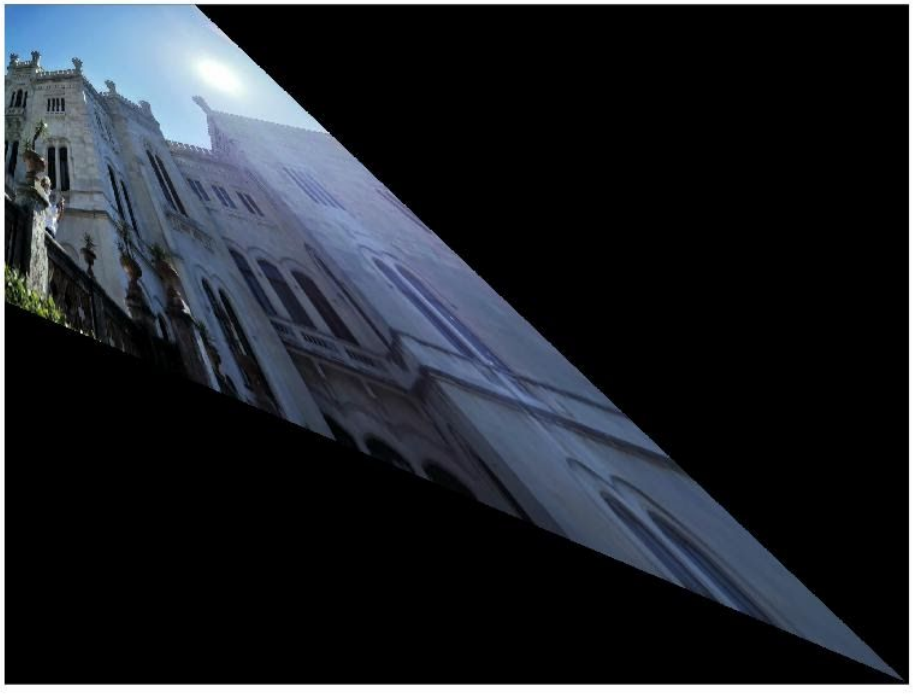
G1 2D reconstruction

Reconstruction of the plane Π is made in a stratified way, first with affine and then with metric reconstruction

G1.1 Affine reconstruction

The first phase is the affine reconstruction, which uses 2 pairs of parallel lines on the plane Π or on a plane parallel to it. After picking the desired couples of images of parallel lines, from each couple is calculated the corresponding vanishing point, as the cross product between the 2 lines. From the 2 vanishing points is calculated the image of the line at infinity `I`, as the cross product of them.
The image of the line at infinity can be used to calculate the rectification matrix `H` and from it the image can be affinely rectified. `H` obtained as:

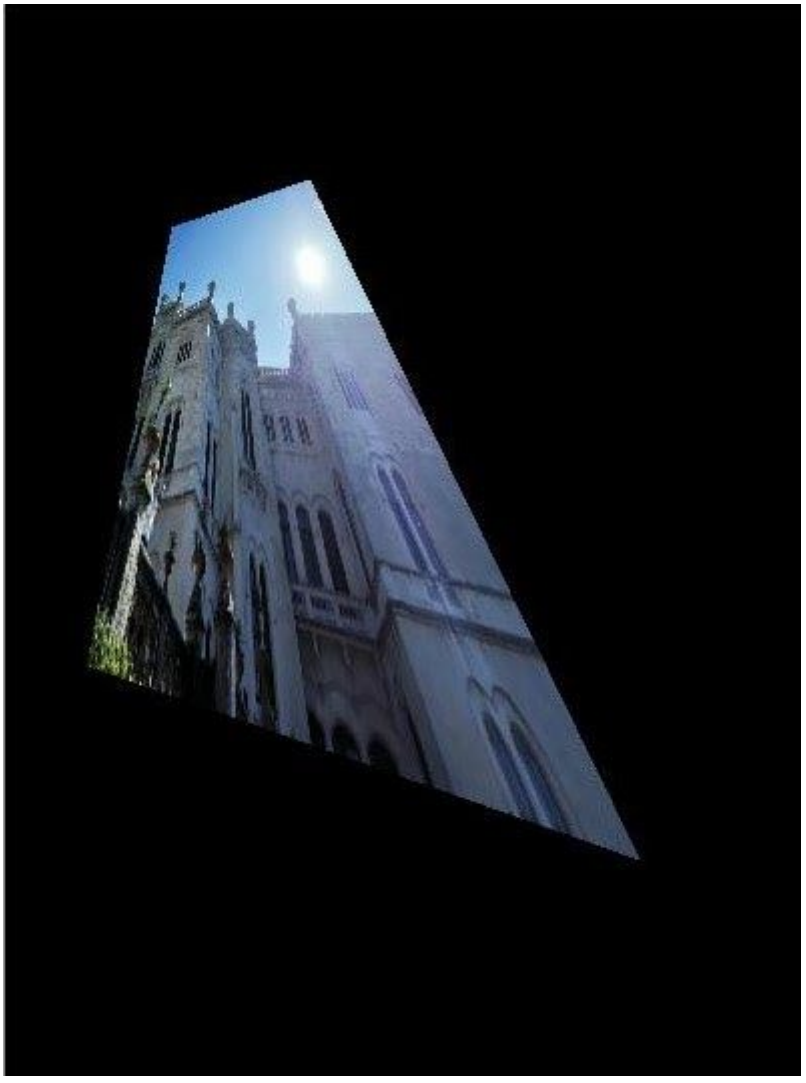
```
H=[ 1      0     0
    0      1     0
  I[1] I[2] I[3]]
```



Result of affine rectification

G1.2 Metric reconstruction

The second phase is the metric reconstruction, using pairs of orthogonal lines respecting the orthogonality constraints given by the problem(i.e. lines 1 and 2 are orthogonal, as well as lines 4 and 5, and lines 5 and 6). After picking 2 pairs of images of orthogonal lines, use them to build a matrix with the desired constraints, from which we can deriva the rectifying homography matrix H and from it can be computed the image rectification of the image



Result of metric rectification

G2 Calibration

Calibration matrix K containing intrinsic parameters of the camera can be calculated starting from a planar object of known reconstructed shape(our plane Π) and a vanishing point of the normal to the plane.
4 vanishing points are needed, the first 2 can be computed as the same lines used for the affine rectification (h2 and h3), and from them is found again the image of the line at infinity, 2 more lines are picked in such a way that each of them is orthogonal to one of the previous couple of lines, and with intersection of these 2 new lines with the image of the line at infinity 2 new vanishing points are obtained(h1 and h4). At last a vanishing point obtained from 2 lines orthogonal to the plane Π is needed(v1). With all these elements it is now possible to solve the following system of equations to find the image of the absolute conic ω:

$$
\begin{aligned}
h1'*\omega*h2=0\\
h3'*\omega*h4=0\\
v1'*\omega*h1=0\\
v1'*\omega*h2=0
\end{aligned}
$$

Since $\omega=inv(K*K')=\begin{bmatrix} a^2 & 0 & -u*a^2 \\ 0 & 1 & -v \\ -u*a^2 & -v & fy^2+a^2*u^2+v^2 \end{bmatrix}$,
from ω it is possible to estimate K and all the intrinsic parameters of the camera as:

```
a = sqrt(ω(1, 1));
u = ω(1, 3)/ω(1, 1);
v = ω(2, 3);
fy = sqrt(ω(3, 3) - ω(1, 1)*u^2 - v^2);
fx = fy/a;

K = [fx,  0, u;
      0, fy, v;
      0,  0, 1];
```

Where `a` is the aspect ratio of the image, `u` and `v` are the principal point, `fx` and `fy` are the focal distances.

The following are the analytical results found for `K` and the intrinsic parameters:

```
a = 0.5249586;
u = 208.6523;
v = 569.3237;
fy = 659.6928;
fx = 1256.657;

K = [1256.657,  0,        208.6523;
     0,      659.6928,    569.3237;
     0,          0,       1];
```

G3 Localization

The relative pose(i.e. position and orientation) between the reference attached to the plane Π and the camera reference can be determined starting from a calibrated image and a reconstructed image, the procedure is to obtain the homography H from the reconstructed image to the original one, this can be done by picking 4 points on the first image and their corresponding on the other image, from these can be derived H, given H previously calculated K our plane can be localized as:

```
[i j o]= inv(K)*H
```

Additionally, rotation and translation vectors of the camera can be computed as:

```
R=[i j cross(i,j)]
t=-R'*o
```

R has actually been approximated with SVD, since due to numerical error it might not be a rotational matrix.

The analytical results found are:

```
i=[  0.8357732
    -0.1691801
    -0.5223613]
j=[  0.0349438
     0.5243465
     0.6951671]
o=[ -593.8621
    -1500.617
     941.092]
R=[  0.9315174, 0.3003643,  0.2050773
    -0.0259272, 0.6172765, -0.7863189
    -0.3627715, 0.7271527,  0.5827914]
```

```
t=[   855.6875
       420.353
     -1606.636]
```

G4 Reconstruction

To rectify also a vertical facade given the calibration matrix `K`, 2 pairs of parallel lines on the desired facade are needed, from them are calculated 2 vanishing points, and the corresponding image of the line at infinity, the image of the circular points `I` and `J` are then calculated as the solution of a system which represents the intersection points between the image of the line at infinity and the image of the absolute conic. Finally the image of the conic dual to the circular points `C` is obtained as:

```
C=I*J'+J*I'
```

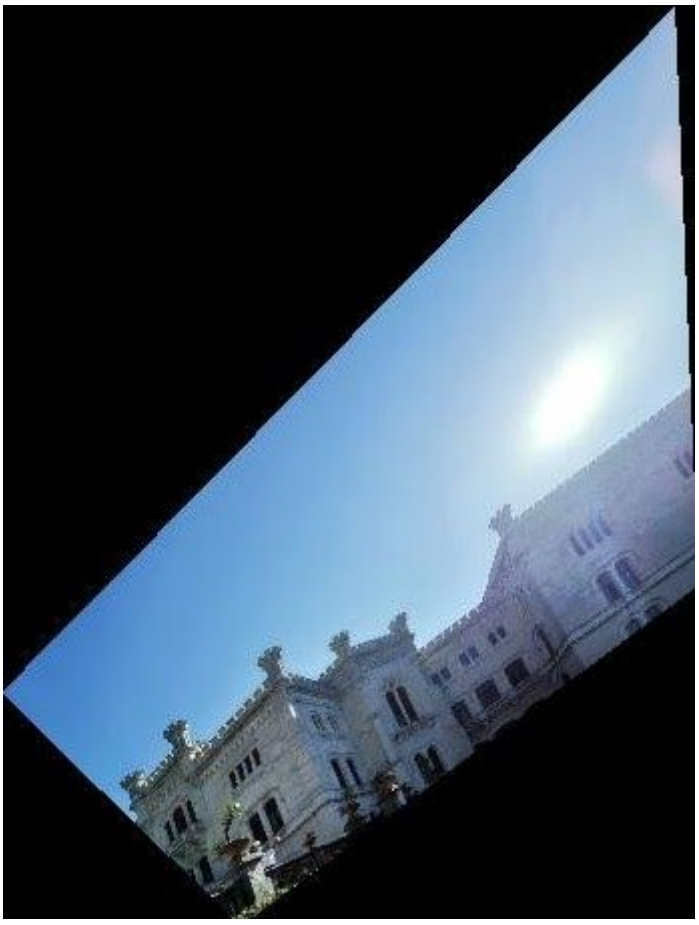
Applying single value decomposition:

```
svd(C)=U*|1 0 0| *U'= inv(H)*C*inv(H')
         |0 1 0|
         |0 0 0|
```

The rectifying Homography is then:

```
H=U'
```

The resulting reconstruction of a vertical face (facade containing line 6 of plane Π) is the following:



Result of vertical face reconstruction

The image above has been manually rotated of 180° for visualization purpose, since the reconstruction provided an upside down image. Probably due to some lack of precision, the image doesn't seem to be perfectly rectified, the top-left corner of the reconstructed facade looks a bit larger that 90°.

4 References

-Theoretical notions from course slides and lessons by professor Vincenzo Caglioti

-Matlab code for rectification examples by professor Luca Magri
-https://boracchi.faculty.polimi.it/teaching/IAS/Rectification/demo_affine_rectification.html
-https://boracchi.faculty.polimi.it/teaching/IAS/Rectification/demo_metric_rectification_stratified.html

-Matlab code for finding homography that matches an image into another by professor Giacomo Boracchi:
-https://boracchi.faculty.polimi.it/teaching/IAS/Stitching/stitch.html

-Mathwork documentation about application of Canny edges, Hough transform and Harris algorithm :
-https://it.mathworks.com/help/images/ref/edge.html
-https://it.mathworks.com/help/images/ref/houghlines.html
-https://it.mathworks.com/help/vision/ref/detectharrisfeatures.html