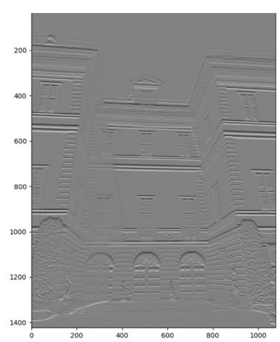
Computer Vision Report

Alessandro Ardenghi, Rocco Cristiano Giampetruzzi, Rolf Minardi

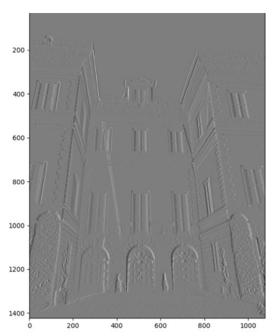
07 April 2024

1 Task1

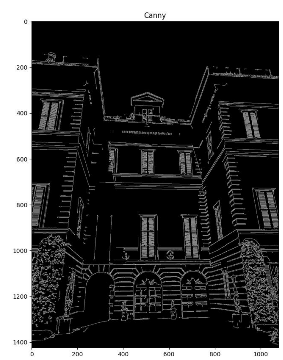
For our initial task, we employ several methods to extract features. We begin by performing edge detection, utilizing both Sobel and Canny edge detection techniques following image preprocessing with GaussianBlur for smoothing. Note that in this task, parameters such as kernel size for Sobel, or thresholds for Canny, have been chosen arbitrarily. Selection of ”best” parameters is carried out later, in Task3, where we actually use the output of the Canny edge detector. In order to find lines, we use the Hough transform. Note that instead of using the raw image as input, we fed the transform with the output of the Canny Edge detector. This indeed led to relevantly better results. Also in this case, selection of ”best” parameters is carried out in Task3, where we actually use the output of the Hough transform. Finally, for Corner Detection, we used the “Harris corner detection method”.



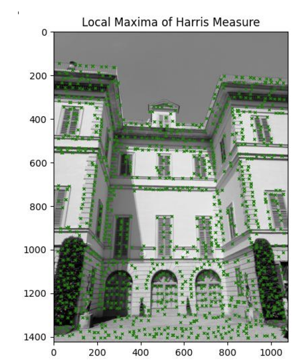
(a) Sobel Y Filter



(b) Sobel X Filter



(c) Canny Filter



(d) Harris Filter

2 Task2

In this task, we are asked to rectify the horizontal section of the ’villa.png’ picture, and to estimate the ratio between the lengths of facade 2 and 3.

2.1 Theoretical Framework

In this short subsection, we will go over the main theoretical results we used to carry out this task. What we will use next in this task is the angle that the ray of the sun creates with the facade 3 of the villa, so in this section we will provide an argument to compute it.
We set ourselves in the top view reference system described in the homework. Looking to the Figure (2) below, it will be enough to compute the ratio between the length of facade 2 and segment defined by the point P (which is intersection of the ray in orange with the y-axis) and the origin O.

First, we notice that the point at infinity of the sun $(3.9, -1, 0)$, defines all the parallel lines which intersect in this point, indeed, the slope of these lines is $-1/3.9$, and by changing the coefficient c, we can generate all the possible parallel lines. Among those lines we have also our optical ray in orange. The line passing through the origin and this point will be exactly $y = -1/3.9x$.

Now, consider the right-angled triangle given by the points : $\{(0, 0, 1), (0, -1, 1), (3.9, -1, 1)\}$, with the hypotenuse defined by the line described above. It is clear that the ratio between the two cathetes is 3.9.
This triangle is similar to the triangle in Figure 2 defined by points $\{P, A, O\}$, for this reason the ratio between their

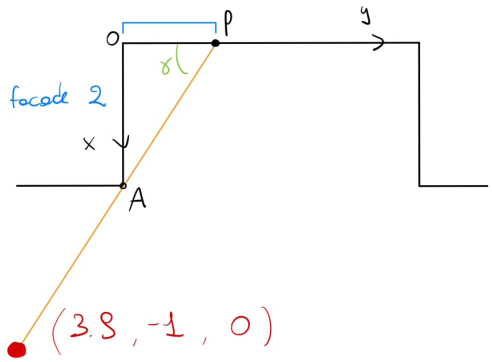


Figure 2: Top view reference system

sides is the same.

We can finally conclude that the ratio that we were looking for, between facade 2 and segment OP, is 3.9.

2.2 Metric Rectification

We can proceed by doing a stratified rectification of our original image 'villa.png'. In the first step we perform the affine rectification by choosing following 2 couples of parallel lines (Figure 3a) and the result that we get is the following in Figure (3b).

(a) Lines for the affine rectification.

(b) Affine rectification.

(c) Lines for the second step of rectification

(d) Final rectification, image up to a similarity.

At this point, for the second step of the rectification process, the information computed in section 2.1 comes into play. Indeed, we are able to find only one couple of orthogonal lines, so for the second constraint we will use the norm of the lines in Figure (3c) defined by the points $0-1$ and $1-2$ (name these lines l and m). Note that they are the images of the segments AO and OP in Figure (2). Thus, the new constraint is the following:

$$l^* C_\infty l = k^2 m^* C_\infty m$$

where $k = 3.9$ as we computed in section 2.1.
This last constraint allow us to define the system of equation to find the homography that defines the image up to similarity (Figure 3d).

Since in this image the original lengths are preserved up to a constant, then we can measure the length of facade 2 and 3 and compute the correct ratio, which ends up to be ≈ 0.62.

3 Task3

In this task, we are asked to use robust fitting to perform two tasks: finding the dominant lines and then divide them in different clusters of lines which are parallel to each other in the real world.

3.1 Theoretical Framework

The main theoretical results that we exploited in this section are duality theorems and dual nature of points and lines in homogeneous coordinates.
More specifically, we exploited the fact that given any two points $x = [x_1, x_2, x_3]^\intercal, y = [y_1, y_2, y_3]^\intercal$ and any two lines $l_1 = [a_1, b_1, c_1]^\intercal, l_2 = [a_2, b_2, c_2]^\intercal$ in the projective plane, they can be seen as the two lines $x^{dual} = [x_1, x_2, x_3]^\intercal, y^{dual} = [y_1, y_2, y_3]^\intercal$, and two points $l_1^{dual} = [a_1, b_1, c_1]^\intercal, l_2^{dual} = [a_2, b_2, c_2]^\intercal$ in the dual space.
This correspondence implies that using the RanSaC algorithm in the dual space to find the line with greatest consensus (i.e. that passes close to as many points as possible), means finding the point in the original projective space which is close to as many lines as possible.
Working in this framework, and assuming that a good portion of the lines in the real world view of the picture will be parallel to one of the three Carthesian axes, we expected the dual lines found by RanSaC (which represent intersecton points of lines in the original projective space) to be the intersection points of sets of parallel lines, and hence vanishing points.
In the next subsections, I will go into detail regarding the steps we took to reach the result.

3.2 Hough Transform

The first step we took was extracting from the picture a set of lines on which to perform the robust fitting. To do so, we exploited the Hough Transform. More specifically, we took the edge map generated in Task1 using Canny Edge Detector and applied the Hough Transform on it, obtaining a set of lines in return (Figure 4).

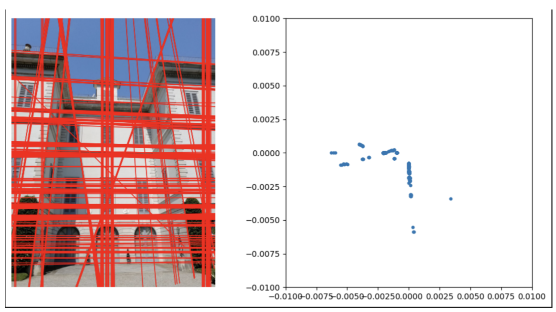


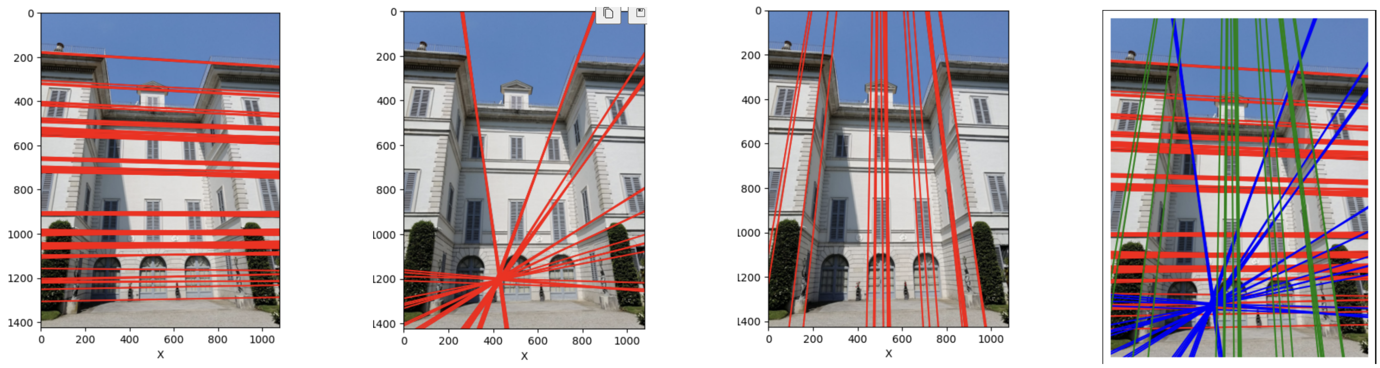
Figure 4: Left: Lines obtained through Hough Transform. Right: Representation of the lines as points in the dual plane

The lines returned by the Transform were expressed in polar coordinates (i.e. pairs of (ρ, θ) values), and we had to convert them to standard form (i.e. $ax + by + c = 0$, or more compactly $[a, b, c]^\intercal$) to then plot them on the dual plane. As can be seen from Figure 4, the dual points in the dual plane are aligned and seem to form some dual lines, which means that the lines representing these points intersect very close to each other in the original projective plane. Our task with RanSaC was to correctly estimate such dual lines.

3.3 Sequential RanSaC and Results

After having obtaines the lines and having transformed them in homogeneous notation, we implemented a RanSaC algorithm with the objective of fitting lines through the points displayed in Figure 4. In order to extract more clusters of inliers, we ran it sequentially, storing at each complete iteration of the algorithm the best dual line found and the set of inliers, removing such inliers from the plane and running again the algorithm, until we had the desired number of clusters.

The results of this sequential run are displayed in Figure 5, and we can see that they are very satisfactory. The algorithm found the following results: the inliers corresponding to the first dual line are the lines parallel to the y-axis in the real world, and the best-fitting dual line is their vanishing point; the inliers corresponding to the second cluster (Figure 5b) are lines parallel to the x-axis in the real world, and the ones belonging to the third cluster are the ones parallel to the z-axis.
We can therefore conclude that the dominant lines of the picture are the ones displayed in Figure 5d, and that the different clusters of dominant lines are the ones displayed in Figure 5a, 5b, 5c.



(a) 1st Cluster (b) 2nd Cluster (c) 3rd Cluster (d) Lines

Figure 5: Dominant Lines and Clusters

4 Task4

In order to calculate the intrinsic camera matrix $\mathbf{K}$, I first compute $\mathbf{W} = (\mathbf{K}\mathbf{K}^T)^{-1}$, and then use Cholesky decomposition to find the inverse of $\mathbf{K}$. I first represent W as a homogeneous vector $\mathbf{w} = (w_1, w_2, \ldots, w_6)^T$, where $\mathbf{w}$ represents $\mathbf{W} = \begin{bmatrix} w_1 & w_2 & w_4 \\ w_2 & w_3 & w_5 \\ w_4 & w_5 & w_6 \end{bmatrix}$ In order to compute $\mathbf{w}$, I use 5 constraints:

1. $v_i^T \mathbf{w} v_j = 0$ for all $i, j \in \{1, 2, 3\}$ with $i \neq j$, where the $\mathbf{v}$'s are the vanishing points in homogeneous coordinates computed in Task 3 (3 constraints).
2. $w_2 = 0$, since the skew equals 0.
3. $w_1 = w_3$, since the aspect ratio equals 1.

Practically, we represent these constraints in the form $\mathbf{A}\mathbf{w} = \mathbf{0}$, where $\mathbf{A}$ is a 5×6 matrix. We solve for $\mathbf{w}$ using Singular Value Decomposition (SVD), as explained in class. We then decompose $\mathbf{w}$ using Cholesky Decomposition, and invert the output to obtain the non-normalized Camera Matrix. Finally, we refine our result by setting to zero terms smaller than 1×10^{-12}, and normalize with respect to w_6 (which we can do because the matrix was computed up to a similarity).

5 Task5

In this last task, we have to extract and match relevant features from the 2 different views of "Villa Melzi d'Eril in Bellagio", and draw the epipolar lines.

5.1 Theoretical Approach

Before proceeding, we give a bit of background on what is the theoretical we are going to pursue. The leading question is: once we have the matching points between the 2 images, how do we compute the epipolar lines for image 2 (or viceversa)?

The epipolar lines are the projection of the optical ray passing through the points in image 1, into the second image. The way in which we can compute them is the Fundamental Matrix. Indeed, by multiplying any point in image 1 by the fundamental matrix, we get the corresponding epipolar line in image 2 :

$$l_2 = F \cdot x_1$$

Thus, once we get the fundamental matrix our task is almost done. We can compute the F by exploiting the feature matching between the 2 images. One possible algorithm is the 8-point algorithm, which uses as constraints the fact that, if a point belongs to a line then the dot product between the point and the coefficients of the line will be 0. Given a correspondence $\{x_1 \Leftrightarrow x_2\}$ in image1 and image2, then:

$$x_2^T \cdot F \cdot x_1 = 0$$

In conclusion, we need to find a set of correspondences in order to fit the fundamental matrix F and finally compute the epipolar lines corresponding to point in image 1 into image 2.

5.2 Automatic Feature Matching

We started by extracting the relevant features, by using SIFT algorithm, from both the images (we avoid to show again the picture since very close to what we get in task 1; if needed it is available in the notebook). In order to match the keypoints that we obtained as output of SIFT, we used FlannBasedMatcher which is the nearest neighbors approach to the matching feature problem and runs much faster than classic brute force search, and then apply the Lowe's Ratio Test to discard bad matches.

Figure 6: Matching between the 2 views of "Villa Melzi d'Eril in Bellagio" by using FLANN.

As it can be noticed from Figure (6), the matches do not correspond to reality, indeed, the peak of the mountain in the second view is matched with the angle of a door in the first view, or some windows in first image are matched with different windows in the second image and so on. Because of this we believe that estimating the fundamental matrix and computing the epipolar lines using the matches above would lead to incorrect estimates, since there are too many outliers and even by using Ransac it could lead to unexpected results.

5.3 Manual Feature Matching

To get around this issue we decided to select manually the set of correspondences. In particular, we tried to select couple of points which happens to be keypoints in both the views, according to SIFT.

Given the correspondences we used 'findFundamentalMat' method of OpenCV with the option FM_RANSAC, such that it keep only the inliers correspondences and fit F on them. The fundamental matrix that we got is :

$$F = \begin{bmatrix} -3.80861130e-07 & 1.68268568e-06 & -3.00939787e-03 \\ -2.13175955e-06 & 5.20488748e-07 & 1.95997446e-03 \\ 2.23856328e-03 & -1.99364479e-03 & 1.00000000e+00 \end{bmatrix}$$

Finally, we can apply F on the correspondences to compute the epipolar lines as required, and the final result is shown in Figure (7).

Figure 7: Epipolar lines on the 2 images, computed on the correspondences that we selected

From the pictures, we can almost identify the positions of the epipoles in the 2 images. They seem to appear in reasonable positions, given the 2 views of the "Villa" that we have.