<u>Homework</u> Image Analysis and Computer Vision

Cortese Pasquale Enrico (903147)

pasqualeenrico.cortese@mail.polimi.it

Politecnico di Milano

A.Y. 2019/2020

January 27, 2020

CONTENTS

1. Introduction

The image given for this homework is taken from Sciarra gallery's internal courtyard situated in Rome. The building was built between 1885 and 1888 as the extreme courtyard of the Sciarra Colonna di Carbognano Palace, during the renovation and modernization of the central districts of Rome. The central room is rich in architectural scores, while the Art Nouveau style decoration develops the iconographic theme of the "Glorification of Woman", illustrating models of feminine virtues and representing scenes from everyday bourgeois life. The vaulted roof is made of iron and glass.

The architectural scores and all the decorations represented an issue to deal with in feature detection; also the meteorological condition in which the photo has been take, a sunny day probably around noon, created a high contrast between areas illuminated by sunlight, that penetrates through the glass roof, and the areas that are in the dark, another matter to tackle. Last but not least another facts to deal with was the assumption on the width of the windows, needed for metric reconstruction, the orthogonal structure of the facades and the assumption of skew symmetry (but not of natural camera), exploited in the calculation of the calibration camera matrix.

FIGURE 1 - IMAGE OF MAIN FACADE FROM ANOTHER POINT OF VIEW.

2. Image features extraction

The goal is to extract peculiar features from the image that will be useful in the subsequent step of the work; these features are edges, corners and line.
Colors are an information not needed, so the image has to be converted in grayscale. This operation is disturbed by the presence of dark halos in the image, so a procedure had to be followed:

FIGURE 2 - GRAY IMAGE

The image has been converted in the HSV color space, "Hue Saturation Value", in order to extrapolate information about the brightness of the pixels and from that enhance the image, obtaining the result in Figure 2: the right part and the low left corner area would have been so darker that no features could have been extrapolated from there.
After this preliminary step, we execute edge detection, which is an image processing technique for finding

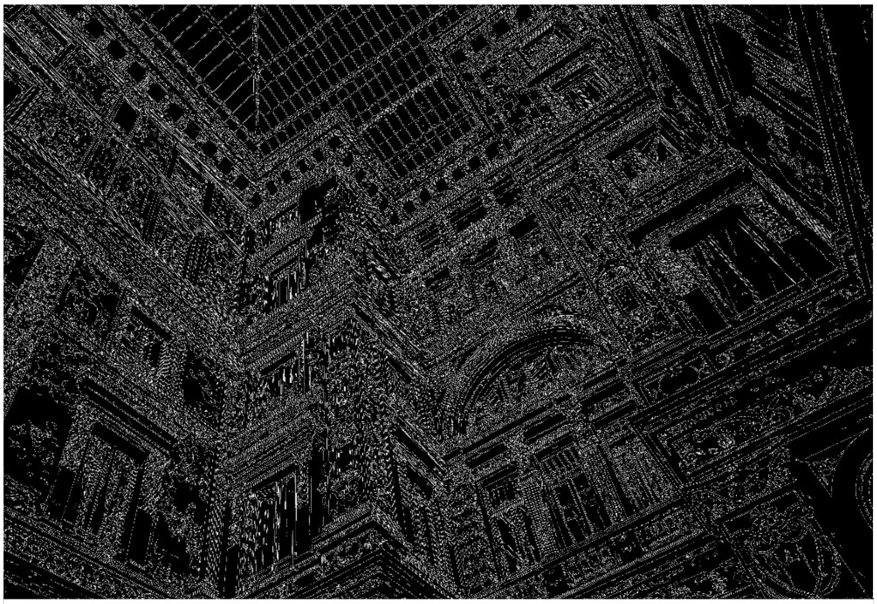

FIGURE 3 - EDGE DETECTION

the boundaries of objects within images, working by detecting discontinuities in brightness; between the many options, after some testing it has been chosen to adopt the Canny method, that revealed to be the most accurate.
After edge detection, the second step is to run corner detection. To do that, the Harris measure has been calculated from the enhanced gray image: the plotted corners correspond to those points being local

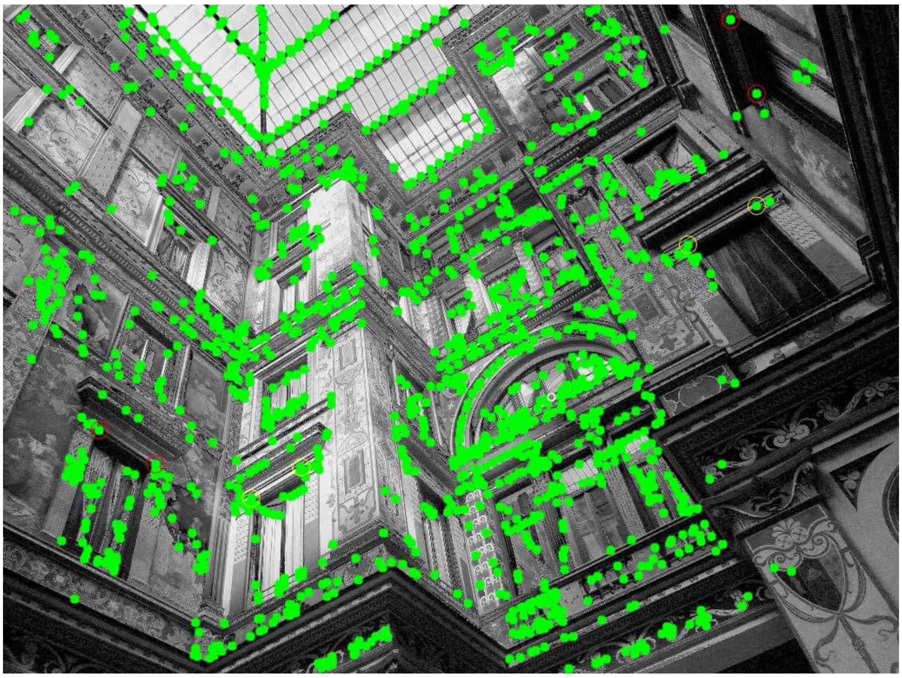

FIGURE 4 - CORNER DETECTION

maxima of Harris measure.

In Figure 4 it can be seen that eight corners, four reds and four yellows, in correspondence of the high corners of windows in facades 1,2,6 and 7 have been selected as features to use in the following.
Eventually, line detection has been executed. In order to do this, it has been used the Hough transform

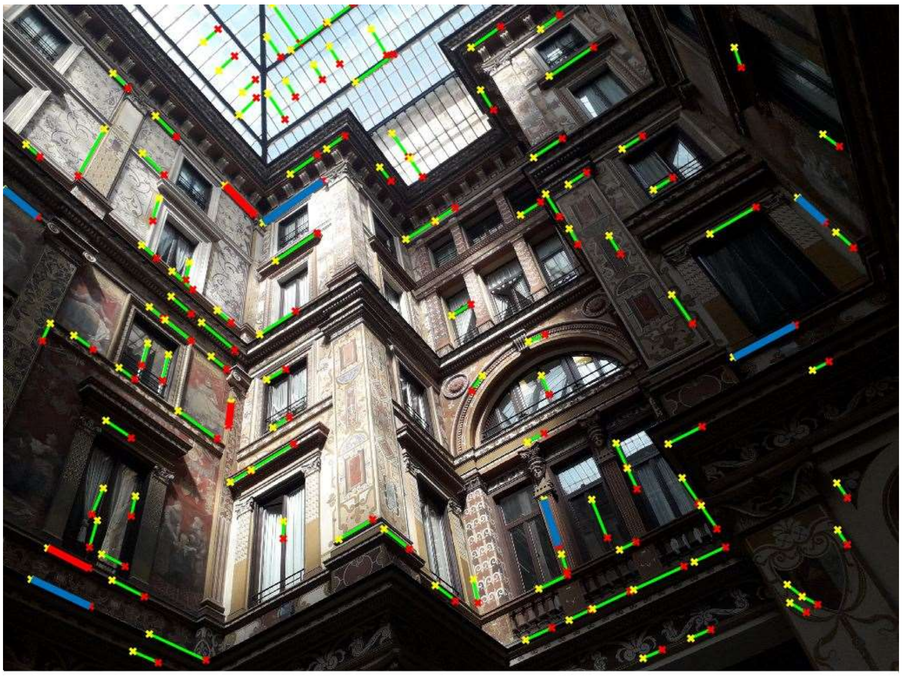

FIGURE 5 - LINE DETECTION

function on the result of the edge detection; also, here there's been tuning of the parameters with trials and errors and the result can be seen in the Figure 5 - Line Detection.

It's possible to observe in Figure 5 that nine lines has been chosen, six blues and three reds: the blue lines have been selected in order to calculate the vanishing points, while the red ones will be in the following. Crossing the three pairs of parallel lines depicted in blue, three Vanishing Points have been calculated: two horizontals, the first one left to the image and the second on the right, and one vertical, as shown in Figure 6.

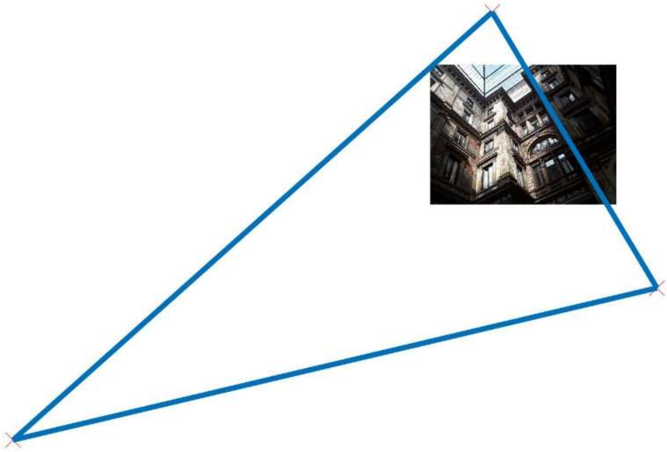

FIGURE 6 - VANISHING POINTS

3. Metric Reconstruction

An affine transformation is a non-singular linear transformation followed by a translation. Since an affine transformation includes non-isotropic scaling, the similarity invariants of length ratios and angles between lines are not preserved under affinity. Affinity invariance instead are parallelism between lines, the length ratio between parallel segments and the consequent ratio between areas [1].
The affine transformation required to metrically rectify an affine image may be computed from imaged orthogonal lines, thus four points have been selected depicting a plane parallel to the ground plane XZ. The mentioned four points are the ones circled in red in- Corner Detection (Figure 4) , except for one, which

FIGURE 7 - METRIC RECONSTRUCTION. A) ON THE LEFT THE ORIGINAL IMAGE WITH THE SELECTED PLANE, DEPICTED IN BLUE; B) ON THE RIGHT THE METRIC RECONSTRUCTED IMAGE, WITH THE TARGET LINE AND POINTS.

position has been improved using the vanishing points in order to achieve a better parallelism between the created plane and XZ; the image coordinates of the points are then mapped in the coordinates of the metric reconstructed image, relative to a selected point of the four (the high left corner), imposing the desired width of window; thus the transformation matrix is obtained.
In Figure 7b it's possible to observe the target line on facades 2,4 and 6 is perfectly perpendicular with itself on facades 1 and 7, so it was possible to measure the distance between the target points.

name	**value**
Width of window #1	100.00 cm
From windows #1 to facade's corner 1-2	104.06 cm
From facade's corner 1-2 to windows #2	49.53 cm
Width of window #2	100.24 cm
From window #2 to window #3	642.19 cm
Width of window #3	97.81 cm
From windows #3 to facade's corner 6-7	45.47 cm

TABLE 1 - RESULT OF METRIC RECONSTRUCTION

In Table 1 it's possible to observe that a not perfect result was achieved, due to the presence of errors in the order of centimeters. A better result could have been achieved probably if a less straightforward approach were adopted, first calculating the matrix for the affine property recover and then the one for the affine rectification, passing through the calculation of the conic at the infinity C^*_∞.

4. Estimation of Camera Calibration Matrix K

The information about orthogonality on lines gave the transformation matrix H in the last chapter. This information has been exploited as constraints in the calculation of the image of absolute conic ω, together with the property of the camera of being skew-symmetric and the vanishing points.
Since all the above constraints are described algebraically as linear equations on ω, it is a simple matter to combine them as rows of a constraint matrix; in a straightforward way one unknown has been taken away from the skew symmetric property.

condition	constraint	# of constraint
vanishing points v_i, v_j corresponding to orthogonal lines	$v_i^T \omega v_j^T = 0$	3
metric plane imaged with known homography H = [h1, h2, h3]	$h1_i^T \omega h2_j^T = 0$ $h1_i^T \omega h1_j^T = h2_i^T \omega h2_j^T$	2

TABLE 2 - CONSTRAINTS FOR THE CALCULATION OF Ω

All constraints may be collected together so that for five constraints the system of equations may be written as Aw = 0, where A is a 5 × 5 matrix and w is a 5-vector containing the six distinct homogeneous entries of ω, thus the solution is found looking for the null space solution of A, and finally ω is assembled.
The matrix K is obtained from

$$\omega = (KK^T)^{-1}$$

by Cholesky factorization of ω, followed by inversion.
As a reference, in the following we have a confront between the obtained K matrix and the one we would have obtained with the hypothesis of natural camera being valid, K_{nat}.

$$K = \begin{bmatrix} 3354.1 & 0 & 2029.4 \\ 0 & 3350.8 & 1596.7 \\ 0 & 0 & 1 \end{bmatrix} \qquad K_{nat} = \begin{bmatrix} 3349 & 0 & 2025.4 \\ 0 & 3349 & 1585.6 \\ 0 & 0 & 1 \end{bmatrix}$$

5. Vertical facade rectification using K

The goal of this part was to achieve the rectification of the frontal of the building, composed by facades 2,4 and 6. Starting from the intrinsic camera parameters - which are fx, fy, x0 and y0, corresponding to K_{11},K_{22}, K_{13} e K_{23} - a matrix $H_{support}$ was built to perform this operation, exploiting those to map the original image to a rectified one over the XY plane. The matrix $H_{support}$ is such:

$$H_{support} = \begin{bmatrix} 0 & 0 \\ K_{11} & 0 \\ 0 & K_{22} \\ K_{11} & K_{22} \end{bmatrix}$$

The intrinsic parameters were also used to obtain more desirable points to map on the original image, starting from the corners of the target facades; finally, a scale parameter was used to reduce computation time and space requirements that otherwise would have overtaken computer requirements.

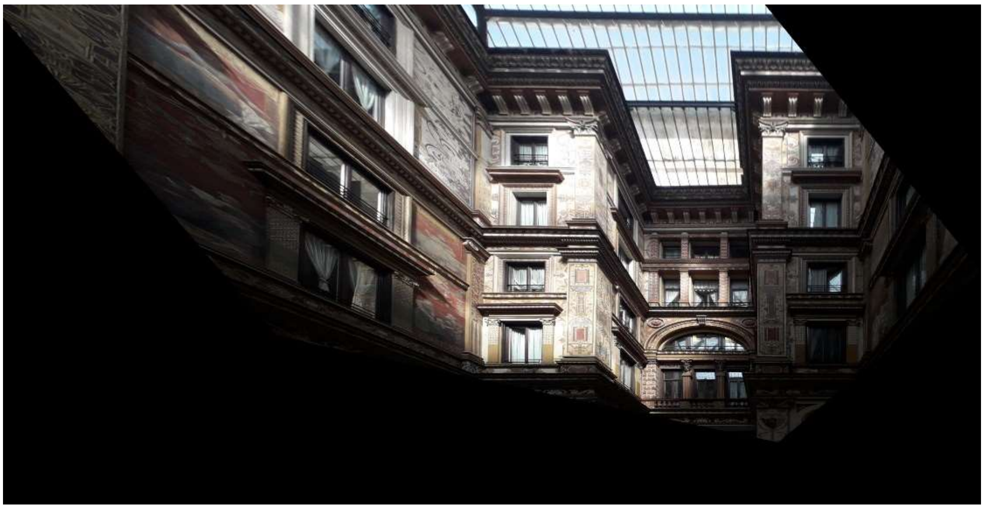

FIGURE 8 - RECTIFIED IMAGE USING K

The image reported in Figure 8 - Rectified image using kis only a small cropped part of the top right corner of the result of the Matlab Script: the picture being taken from the ground and the high inclination of the view point, resulting in one of the three vanishing point really far from the image, caused the increasing of the dimension of the output image, stretching it to a corner.

6. Camera Pose Estimation

The last point was to estimate the camera pose in 3D space; to do so, it has been chosen to tackle the problem with DLT approach [2].
The first goal was to estimate the matrix P, the matrix that maps points from world coordinates; for each set of coordinates we can write:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p1 & p2 & p3 & p4 \\ p5 & p6 & p7 & p8 \\ p9 & p10 & p11 & p12 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

which can be rewritten as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1^t \\ p_2^t \\ p_3^t \end{bmatrix} [X]$$

From this relationship we obtain a set of four constraint and thus a problem in the form:

$$AP = 0$$

where A is the matrix of constraints:

$$A = \begin{bmatrix} X_1^T & 0 & -x_1 X_1^T \\ 0 & X_1^T & -y_1 X_1^T \\ X_2^T & 0 & -x_2 X_2^T \\ 0 & X_2^T & -y_2 X_2^T \\ X_3^T & 0 & -x_3 X_3^T \\ 0 & X_3^T & -y_3 X_3^T \\ X_4^T & 0 & -x_4 X_4^T \\ 0 & X_4^T & -y_4 X_4^T \end{bmatrix}$$

The solution was found with Single Value Decomposition as the column of V corresponding to smallest singular value.
Once we have obtained P, the following relationship has been exploited to obtain separately the orientation matrix R and the translation vector T:

$$P = K\,[R \quad T]$$

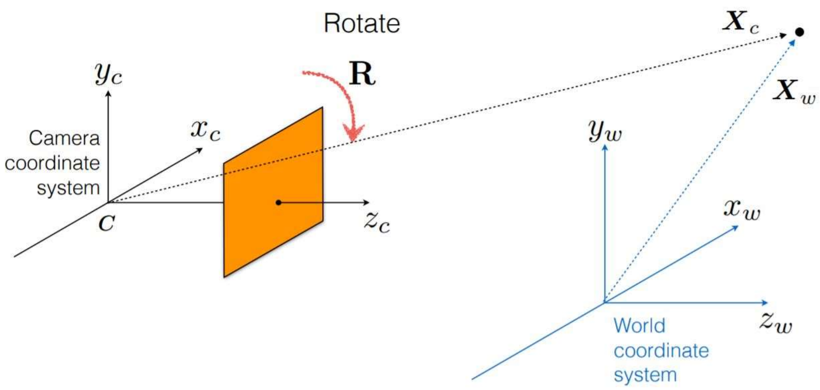

FIGURE 9 - CAMERA POSE

In the end the position vector of the camera C was obtained from the following equation:

$$T = -RC$$

Here follows the result obtained:

$$R = \begin{bmatrix} 1.7075 & 1.7414 & 0 \\ -1.4213 & 1.8794 & 0 \\ 02.3025 & 5.3280 & 1 \end{bmatrix} * 10^{-7} \qquad C = \begin{bmatrix} 1018.9 \\ 440.3 \\ -3.2761 * 10^{15} \end{bmatrix}$$

The reference world frame was considered fixed in the top left corner of the main facade, with axis XYZ directions following the left-hand norm usually adopt in computer vision.
The poor result obtained could be the outcome of the scaling needed to obtain the rectification.

7. References

[1] "Multiple View Geometry in Computer Vision", R. Hartley, and A. Zisserman. Cambridge University Press, New York, NY, USA, 2 edition, (2003).

[2] http://www.cs.cmu.edu/~16385/s17/Slides/11.3_Pose_Estimation.pdf

8. Index of figures