

IACV Homework A.A. 2021-2022

Marzia Favaro - 10749161

## Contents

<b>1 F1 - Features extraction</b>	<b>2</b>
1.1 Histogram optimizations . . . . .	2
1.1.1 Building features enhancement . . . . .	2
1.1.2 Shadows enhancement . . . . .	3
1.2 Thresholding . . . . .	4
1.2.1 Building features . . . . .	4
1.2.2 Shadows features . . . . .	5
1.3 Edges detection . . . . .	6
1.4 Lines extraction . . . . .	7
1.5 Corner detection . . . . .	8
<b>2 G1 - 2D reconstruction of a horizontal section</b>	<b>9</b>
2.1 Affine reconstruction . . . . .	9
2.1.1 Lines selection . . . . .	9
2.1.2 Vanishing points and horizon . . . . .	11
2.1.3 Results and tests . . . . .	11
2.2 Metric reconstruction . . . . .	13
2.2.1 Triangles selection . . . . .	13
2.2.2 Triangle mapping . . . . .	14
2.2.3 Metric rectification matrix from affine . . . . .	16
2.3 Composite transform . . . . .	16
2.3.1 Metric rectification matrix . . . . .	16
2.3.2 Results . . . . .	16
2.4 Walls length ratio . . . . .	17
<b>3 G2 - Calibration</b>	<b>18</b>
3.1 Constraint 1 . . . . .	18
3.2 Constraints 2 and 3 . . . . .	18
3.3 System and its solution . . . . .	19
3.3.1 Comments . . . . .	19
3.4 Corrections - Triangulation of vanishing points . . . . .	19
3.5 Results . . . . .	19
<b>4 G3 - 2D reconstruction of a vertical section</b>	<b>20</b>
4.1 Image of the vanishing line . . . . .	20
4.2 Image of the absolute conic . . . . .	20
4.3 Image of the conic dual to the circular points . . . . .	20
4.4 Rectifying matrix . . . . .	20
4.5 Results . . . . .	21
<b>5 G4 - Localization</b>	<b>22</b>
5.1 Theoretical solution . . . . .	22
5.2 Adjustments . . . . .	22
5.3 Results . . . . .	23

# 1 F1 - Features extraction

## 1.1 Histogram optimizations

### 1.1.1 Building features enhancement

To extract the features of the image, it is important that they are well visible. The image we are considering is the picture of a real object under natural light, therefore its lighting is not uniform (it is explicit from the shadows), and the color is slightly undersaturated. The first operation we are going to perform is therefore an equalization, and we can see from fig.1 that now the shadows are sharper, making also the edges of the building more evident. We can check how the procedure affected the histogram from fig.2



Figure 1: The original image (left) and the same after histogram equalization (right)

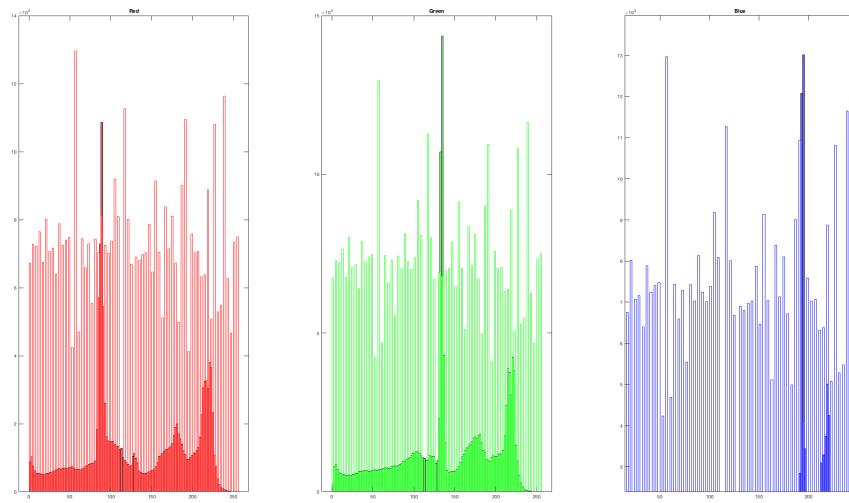


Figure 2: The hisogram before equalization (filled columns) and after (hollow) for each colour channel

However, we can also observe that the procedure introduced some disturbances because it increased the contrast created by the sun on areas that are supposed to be monochromatic (walls, sky). To help with this problem (and at the same time increase the contrast in the areas of interest) we use a Laplacian filter, that increases the contrast on the sharper edges (fig.3).



Figure 3: Before and after the application of a Laplacian filter

### 1.1.2 Shadows enhancement

The operations we performed up to now were aimed to highlight the features of the building, but we are also interested in the information provided by the shadows. We are then going to work separately on two images: the one analyzed before, with the building features, and one with the features of the shadow.

To enhance the shadows we are going to perform a stretch of the histogram, that will darken the darker regions and brighten the lighter. We can see the effect in fig.5. From fig.4 we observe that we created two high peaks in correspondence of 0 and 255, and that the peak around 100 is now smoothed. The procedure is very similar to thresholding (especially for the high values of stretch used) but allows some uncertainty in the classification of some areas.

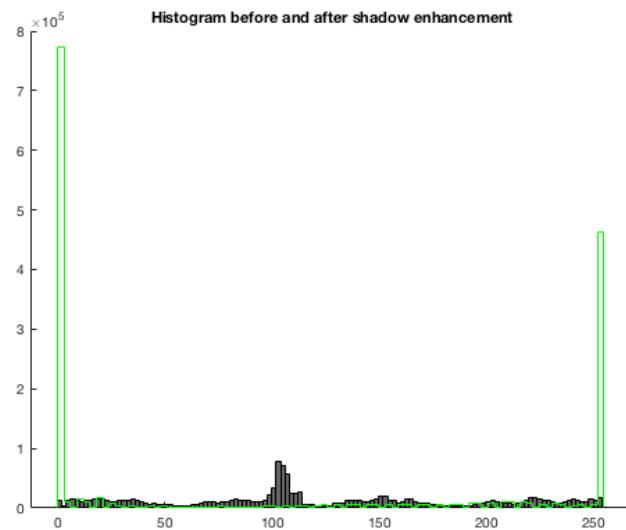


Figure 4: Histogram before (black fill) and after (green, hollow) the histogram stretch to enhance the shadows



Figure 5: Image before and after shadows enhancement

## 1.2 Thresholding

To help the future edge detection and control it better, we are now going to binarize the images.

### 1.2.1 Building features

As we expected, because the image has many colors and shadows, we can't identify two peaks for the threshold. Instead of performing a global operation, that would create problems due to the variability of the image, we perform local thresholding, which means selecting the threshold on smaller areas. We can see from fig.?? that thanks to this strategy we were able to recover information both in the shadow and in the lit areas, which would not have been possible with a global approach. To improve the result, we can also fill the remaining holes in the lines by using a minimum filter (erosion). We are using a small filter, so we are not going to apply a consequent maximum filter (dilation), because it doesn't impact much the lines we extracted and to avoid disconnecting again some weakly connected areas. The result is in fig.7.



Figure 6: Adaptive thresholding

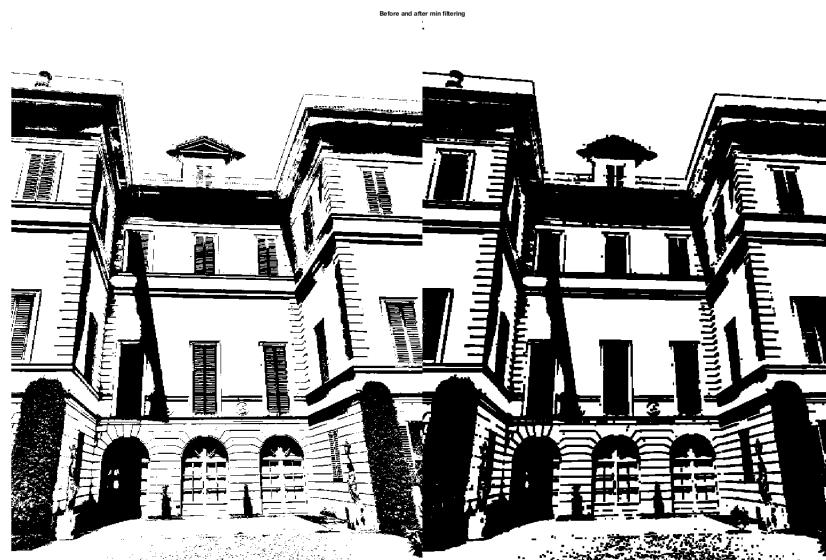


Figure 7: Holes filled with a min filter. Before (left) and after (right) its application

### 1.2.2 Shadows features

The shadow image is easier to threshold, and this time we are going to apply a global threshold. We can see (fig.??) that we have a sharp outline of the shadows but also the shadows on the windows. As these provide unwanted features, we are going to remove them with a maximum filter (dilation).



Figure 8: Global thresholding on the shadow image



Figure 9: Max filter to remove unwanted information. Before (left) and after (right) its application.

### 1.3 Edges detection

If we were approaching the black and white image directly, we would normally choose Canny edge detection because of its versatility and higher noise robustness. However, we are working on binary images, so the Gaussian filtering included in Canny could be a problem, and we could prefer a simpler algorithm such as Sobel, because the derivatives we encounter are very simple. After trying both the approaches, we can see in fig.10 that they are equivalent, but still not satisfactory, as the connectivity on the edges of our interest is poor. The best result is actually obtained using a very basic technique: as the image is binary, we can find the edges by performing the XOR operation on the image and its dilation. Comparing this technique with Canny (fig.11) we can observe that certain connections are improved also thanks to the dilation itself (Canny's lines are thinner, therefore less connected), but other segments are new.

The same masking operation is performed separately also on the shadows' image (fig.12).



Figure 10: Canny (left) and Sobel (right) edge detections

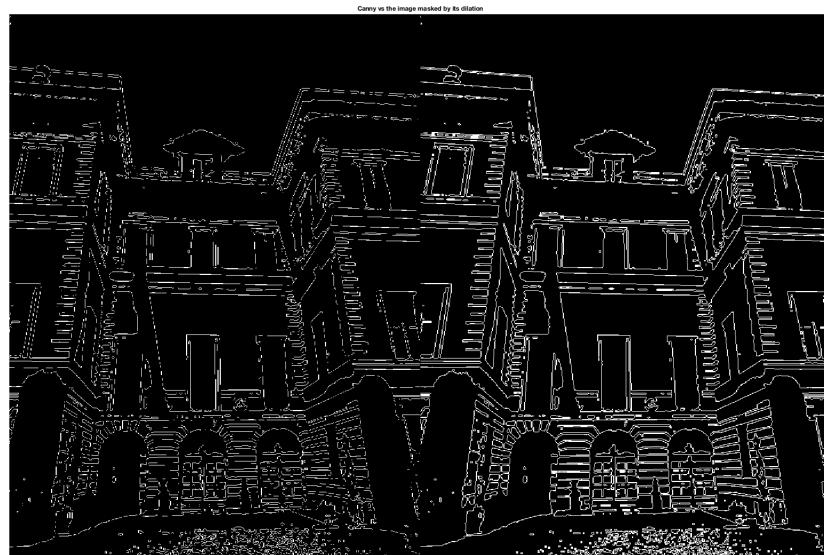


Figure 11: Canny (left) and masks subtraction method (right)



Figure 12: Masks subtraction method on shadow image

#### 1.4 Lines extraction

The images we obtained are now easily interpretable and we can extract the main lines with the Hough transform. The operation is performed on the two feature images separately, but the result is then combined in fig.13



Figure 13: Detected straight lines

### 1.5 Corner detection

To detect the corners in the image we can use Harris algorithm and take the strongest points. Because of the nature of the ground, however, this method would return most of the edges in the floor region, which is of no interest to us. To deal with this we can reduce the region that we analyze (or split the image into areas, to adjust the parameters similarly to how we did for the thresholding part), avoiding useful features to be hidden by uninteresting ones. The points detected are shown in fig.14.



Figure 14: Harris corner detection on the region of interest

for

## 2 G1 - 2D reconstruction of a horizontal section

To obtain the metric rectification of the image we are going to use a two-step (stratification) approach. Not only does this allow us to have more robust results, but also we don't have another reasonable choice: because we don't have any known ellipses or squares, the best alternative would be to find 5 pairs of independent orthogonal lines, which we don't have.

### 2.1 Affine reconstruction

The first step is to recreate an image affine to the original scene (with respect to the horizontal planes). To do this we are going to use the parallelism relationships between lines to find the horizon and consequently the homography.

In its general formulation, an homography is expressed as:

$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ v_1 & v_2 & v \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \quad (1)$$

Because we don't want (and we can't have) the precise reconstruction of the scene (an identity mapping between the original scene and our extracted image), we don't have all the unknowns of the matrix of eq.1, but less. In this first step we want to obtain any valid affinity, which then can assume the form:

$$H_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^T & v \end{bmatrix} \quad (2)$$

To find the unknowns, we observe that the mapping  $H_1$  will map the horizon line (relative to the plane considered) to the line at infinity. The horizon  $\mathbf{h} = [h_1 \ h_2 \ h_3]^T$  alone determines the homography, because:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ v_1 & v_2 & v \end{bmatrix}^{-T} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \iff v_1 = h_1, v_2 = h_2, v = h_3 \quad (3)$$

#### 2.1.1 Lines selection

We are going to select some parallel lines to calculate their vanishing points. The selection of the lines must respect the following rules:

- R1 There must be at least two pairs of parallel lines
- R2 There must be at least two independent pairs
- R3 The lines must concur to the same horizon

Rule R1 can be explained because, according to equation 2, we have 2 unknowns (3 but with a scale factor) and from each pair of lines, we can get two equations. Rule R2 is needed to have a not under-defined system.

Even if theoretically two lines are enough, using more than that allows us to increase precision and it is needed because of the lines selection's imprecision. In our work we selected pairs of lines along 3 directions (see fig.15):

- Lines parallel to the viewer (light pink)
- Lines orthogonal to the viewer (purple and magenta)
- Lines from shadows (yellow)

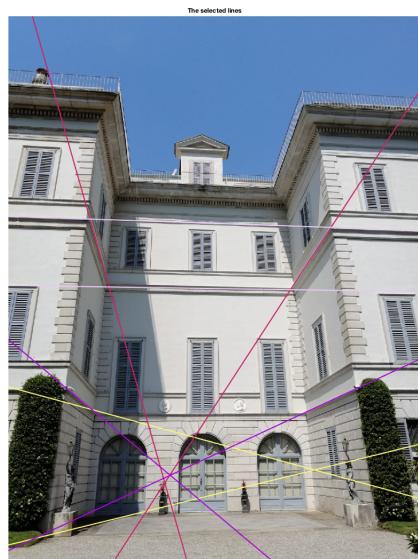


Figure 15: Selected lines

The first two are clearly visible choices: they follow the building's decorations, each pair lies on the same plane and the planes are parallel. While it's trivial to notice that we can work with pairs lying on parallel planes, because they all have the same horizon, it is less obvious why we can select the lines defined by the shadows (yellow), which are parallel on a plane orthogonal to the horizontal one.

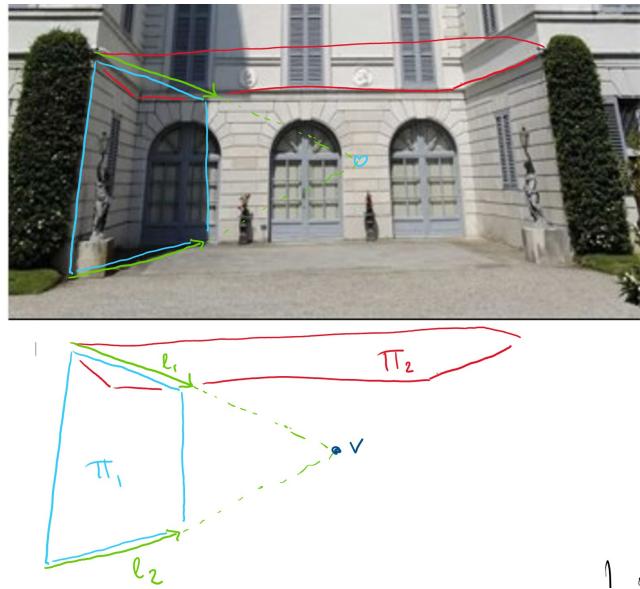


Figure 16: Highlight of the lines from shadows

Referring to fig.16, we observe that:

- $l_1$  and  $l_2$  are trivially parallel on  $\Pi_2$ , because the shadow of the edge of the building on the flat wall is vertical as the edge is
- For the same reason,  $\Pi_2$  is a plane orthogonal to the ground
- Because  $l_1$  and  $l_2$  are parallel, they converge at the vanishing point  $v$
- $\Pi_2$  is a horizontal plane
- $l_1$  belongs to  $\Pi_2$ , hence it will pass through its line at infinity
- $l_2$  belongs to the ground plane, hence it will pass through its line at infinity

- $\Pi_2$  and the ground have the same line at infinity because they are parallel

Then, the vanishing point of the two lines belongs to the horizontal plane at infinity.

While these lines are not strictly needed, it is good to include them to have a third (distinct) vanishing point.

### 2.1.2 Vanishing points and horizon

To get the vanishing points we simply use the cross product on the pairs  $(l, m)$  of lines:  $v = l \times m$ . To get the horizon we are going to perform a mean square error based line fitting, as we have more than two vanishing points (otherwise we would have used the cross product).

We obtained the following vanishing points:

$$\begin{bmatrix} 801.5 \\ 1189.9 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 409.7 \\ 1223.5 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 9490.3 \\ 850.6 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 423.8 \\ 1203.8 \\ 1 \end{bmatrix}$$

Table 1: Vanishing points

Consequently, the horizon is:

$$\mathbf{h} = \begin{bmatrix} 0 \\ 0.0008 \\ 1 \end{bmatrix} \quad (4)$$

### 2.1.3 Results and tests

From 3 and 4 we obtain:

$$H_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.0008 & 1 \end{bmatrix} \quad (5)$$

Some tests were run to check the result.

The most basic one is to check that the horizon is really mapped to the line at infinity:  $H_1^{-T}\mathbf{h} = [0 \ 0 \ 1]^T$   
Then, we can also visually check that the horizon has been estimated as expected in fig.17



Figure 17: The selected lines and their horizon

We then check what happened to the lines. In fig.18 we can observe what we expected: the lines that were parallel in the original scene, now are parallel also in the image. This confirms that we obtained an image affine to the scene.

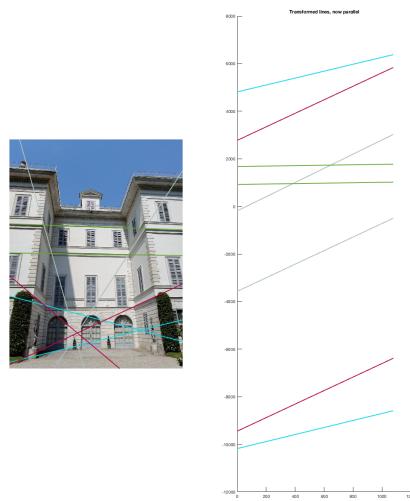


Figure 18: Lines parallel in the real scene are now parallel also in the image

The observation above is confirmed also by the image warped with  $H_1$  (fig.19 and fig.20).

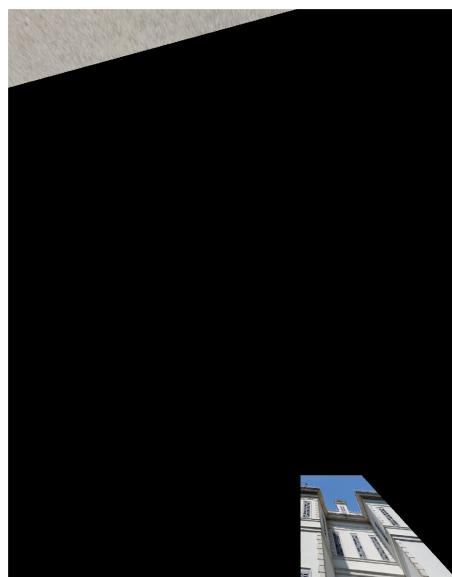


Figure 19: Warped image, affine to the original

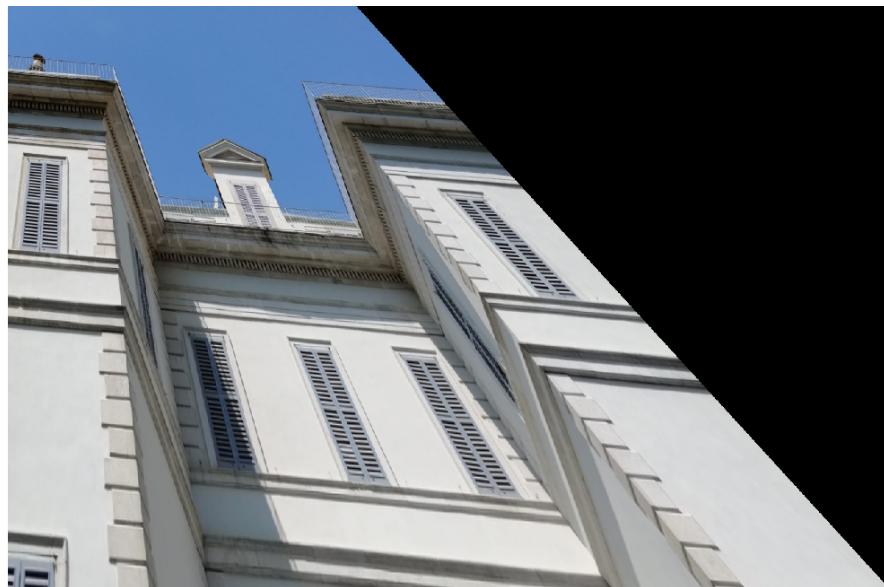


Figure 20: Zoom on fig.19

## 2.2 Metric reconstruction

Because of the data we have available, we can't use a "standard" method to perform this rectification. The approach we are going to use maps triangles whose proportions are known.

Now we are working on the affine rectified image. An implementation note: to reduce the imprecisions, the points are selected on the original image and then transformed as  $x \rightarrow H_1x$  instead of selecting them from the warped image.

### 2.2.1 Triangles selection

The first triangle selected is shown in fig. it21 and in fig.22 can be seen how the vertices have been named. It has a right angle on corner A and lies on a horizontal plane (as the catheti AB and BC do). It has a known angle  $\tan(\theta) = \frac{\text{sun } y \text{ coordinate}}{\text{sun } x \text{ coordinate}} \rightarrow \theta = 14.3814$  on corner B, because corner C is the intersection of the plane in which the catheti lie with the (vertical) shadow of the vertical wall edge. This is true under the assumption that the sun produces rays that we can consider parallel, which is true in a real and open-air context.



Figure 21: First triangle

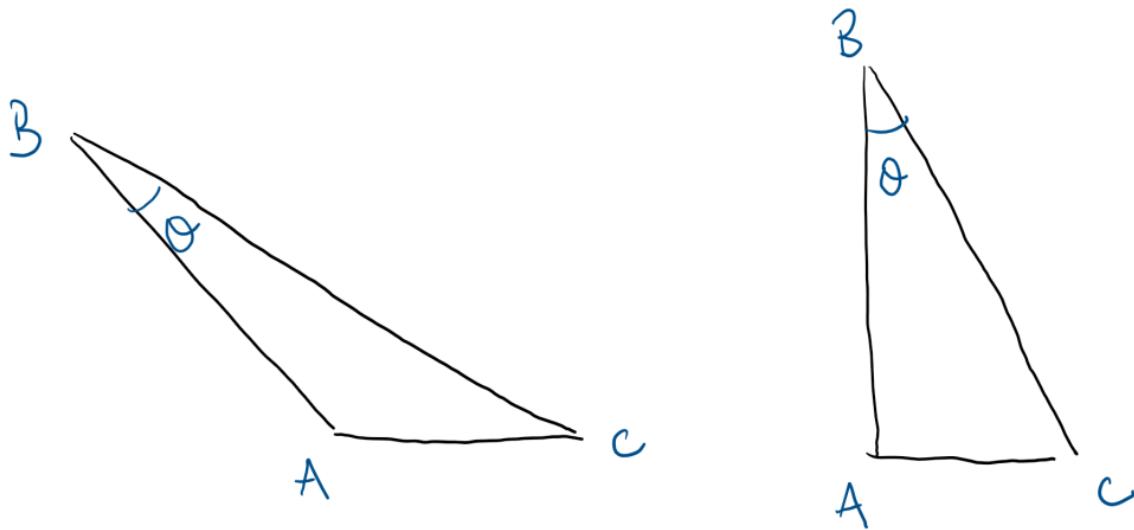


Figure 22: Triangle of fig.21 as seen on the image and from the top-view

The first triangle, as we will see, would be enough to perform the metric rectification, but because of the various imprecisions that we introduce selecting the points, it is better to use other information. We observe that the line in which the vertical shadow we considered before lies, is the same as the one of an edge of a window; this allows us to easily identify another triangle, symmetric to the first (thanks to the building symmetry). It can be seen in fig.23.



Figure 23: Second triangle

### 2.2.2 Triangle mapping

To find the homography of interest we are going to map the triangle corners to their "known" position (except for scale, rotation and translation, because our goal is to find a similarity of the original scene).

For the first triangle it is a trivial trigonometry problem: setting A on the origin, B on the x-axes, and AB of length 1, we have that the points A B and C are respectively mapped in:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \text{scale} \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ \text{scale} \cdot \tan \theta \\ 1 \end{bmatrix} \quad (6)$$

where the scale is relevant only to define the final image size and has been set to 1000. Because the tangent in our case is negative and we prefer working with positive coordinates, we are actually using its absolute value. The second triangle coordinates after rectification are a bit trickier: their relative position can be found in the

same way as before, but for the absolute position we are going to use the cross-ratio. The relationship between the original and metric-reconstructed image is an affinity (or, more in general, a homography), so the cross-ratio of four points is kept constant in both the images.

We are going to consider the points A, C, and their symmetric on the other triangle (that we call A' and C' respectively) on the original image (we could do the same on the affine rectified one, but we may lose precision). We get:

$$CR(A, C, C', A') = \frac{AC' \cdot CA'}{CC' \cdot AA'} = 1.0702 \quad (7)$$

Because the cross-ratio in the final image will be the same, and we know that  $AC = C'A'$ , inverting the equation above we obtain that  $A' = 1070.2$ . We can now define the target coordinates applying simple trigonometry and this information. The result is visualized in fig.25, and we can see that the proportions are compatible with what we would expect them to be.

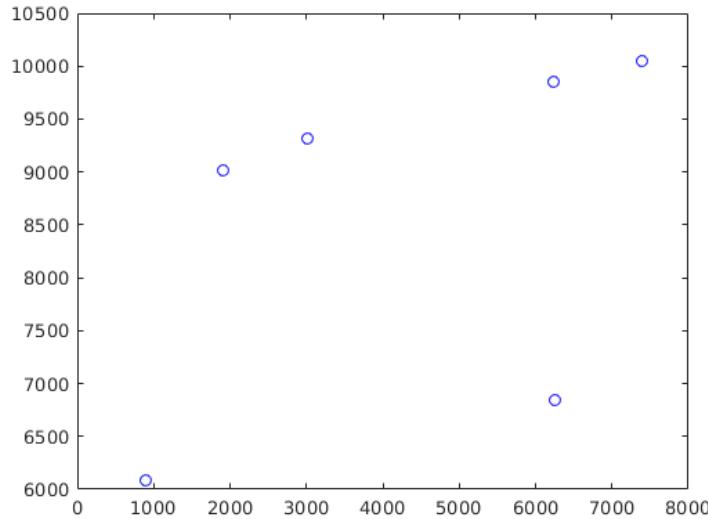


Figure 24: The selected points in the affine image

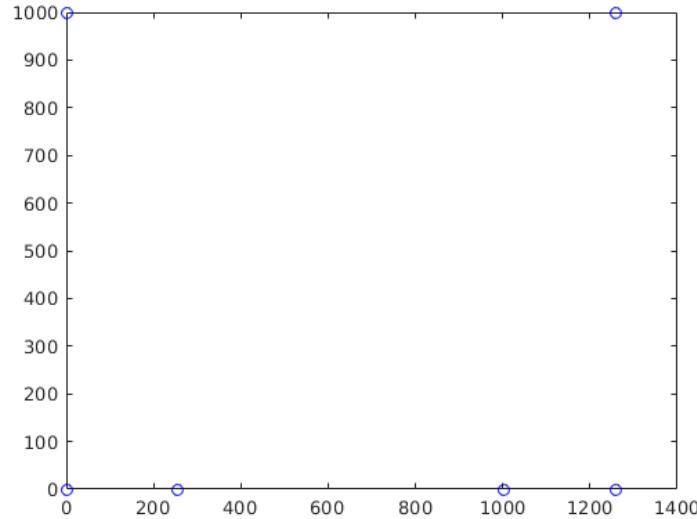


Figure 25: The goal position of the points in fig.24

### 2.2.3 Metric rectification matrix from affine

The matrix we are looking for is an affinity matrix between the affine rectified image and the metric rectified one, so its general formulation is:

$$\begin{bmatrix} a_1 & a_2 & t_1 \\ a_3 & a_4 & t_2 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

To find it we apply a simple mapping between the points  $X_{aff}$  and their goal position  $X_{rect}$ :  $X_{rect} = H_2 X_{aff}$ . Because the matrix has 6 unknowns and each point gives two independent equations, we need at least three points (one triangle). Because we are using 6 the system will be over-determined. We obtain:

$$H_2 = \begin{bmatrix} 0.2 & -0.1 & 297 \\ 0.1 & -0.3 & 3013.6 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

## 2.3 Composite transform

### 2.3.1 Metric rectification matrix

We can now compose the two transformations we obtained before:

$$H = H_2 * H_1 \quad (10)$$

The result we obtain is  $H \approx H_2$  because of the difference in the order of magnitude of the matrices, but what this product did is substitute the last row of  $H_1$  in  $H_2$ .

### 2.3.2 Results

To check the result we can analyze the difference (in pixels) between the expected and the obtained points coordinates (on an image of size 3456x1413):

4.6019	2.0801	7.4335	6.5794	2.0801	3.7479
30.0704	20.6615	9.1292	10.4403	20.6615	10.5009

The result is not a perfect match of our goal points, but it was expected, being the system we built over-determined.

We can visually check it in fig.26.



Figure 26: The rectified image, using  $H$  on the original (plus a rotation)



Figure 27: Zoom of fig.26

We select the first triangle on the newly obtained image and measure the angles:

Corner	Measured angle	Target
A	85.873	90
B	15.223	14.381
C	78.904	75.619

## 2.4 Walls length ratio

The ratio between the lengths of the walls can be easily found with the data we already have, indeed when calculating the target points for the second triangle we got the size of wall 3 (because  $A' = 1070.2$  and  $A = 0$ ), under the assumption  $AB = 1$ . Then, the ratio is 0.7859.

### 3 G2 - Calibration

The most direct method to find the camera calibration parameters uses:

- The rectifying homography  $H$  relative to the plane  $\Pi$ : we already have it
- The image of the line at infinity  $l'_\infty$  relative to  $\Pi$ : we found it during the affine rectification
- A vanishing point  $v$  in a direction orthogonal to  $\Pi$

The only element we are missing is  $v$ , which can be easily found thanks to the vertical edges (fig.30) of the building: it is the cross product of two of them.

We select the vertical lines on the original image and then apply the rectification to  $v$  as  $Hv$  to reduce errors.

We obtain  $\begin{bmatrix} -0.0013 \\ -0.0105 \\ 0 \end{bmatrix}$ .



Figure 28: Vertical lines selected to find the vanishing point

#### 3.1 Constraint 1

Let us call  $v_1$  and  $v_2$  any two vanishing points on  $\Pi$  and  $\omega$  the image of the absolute conic. We know that:

- $v$  and  $v_1$  ( $v_2$ ) are orthogonal, then  $v^T \omega v_1 = 0$  ( $v^T \omega v_2 = 0$ )
- $l'_\infty = v_1 \times v_2$

Then, by substituting the first in the second expression we obtain  $l'_\infty = \omega v$ , which provides two linear and independent equations

#### 3.2 Constraints 2 and 3

About the image of a circular point  $I'$  we know that:

- It can be obtained from  $H$  as  $I' = H^{-1}I = [h_1 \ h_2 \ h_3] \begin{bmatrix} 1 \\ i \\ h \end{bmatrix} = h_1 + ih_2$
- It belongs to the image of the absolute conic:  $I' = 0$

By substitution we obtain  $(h_1 + ih_2)^T \omega (h_1 + ih_2) = 0$ , whose solution is found by setting to zero both the real and the imaginary part:  $h_1^T \omega h_2$  and  $h_1^T \omega h_1 = h_2^T \omega h_2$ . We found two new equations, so we can solve two more constraints.

### 3.3 System and its solution

The constraints above lead to the system:

$$\begin{cases} l'_\infty = \omega v \\ h_1^T \omega h_2 \\ h_1^T \omega h_1 = h_2^T \omega h_2 \end{cases} \quad (11)$$

for a total of 4 equations. These are enough to determine  $\omega = \begin{bmatrix} a^2 & 0 & -u_0 a^2 \\ 0 & 1 & -v_0 \\ -u_0 a^2 & -v_0 & f_Y^2 + a^2 v_0^2 + v_0^2 \end{bmatrix}$ , as there are 4 unknowns:

- $(u_0, v_0)$ : the camera principal point
- $(f_x, f_y)$ : the focal distance
- $a$ : the aspect ratio  $f_y/f_x$

Once we have  $\omega$  we can get the camera calibration matrix  $K$  as the Cholesky factorization of  $\omega^{-1} = KK^T$ .

From here it is trivial to get the parameters of interest, as  $K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ .

We observe, however, that the result is not what we would expect, as the principal point is located at (416, 3260) pixels, which is outside of the image (which is of size (1424, 1078)). Because the principal point is the projection of the camera on the image, the obtained result is visibly wrong.

#### 3.3.1 Comments

- In practice, to avoid accumulating errors, the direct substitution of the parameters in both  $\omega$  and  $K$  has been adopted.
- The difference in the orders of magnitude of the elements of  $\omega$  makes it look full of zeros. By construction, however, it respects the structure presented before theoretically.

### 3.4 Corrections - Triangulation of vanishing points

Another option to find the principal point is via the triangulation of three vanishing points of orthogonal directions (we can take the main axis of the building, parallel to the edges of the walls). Thanks to their orthogonality we obtain three constraints on the location of the principal point that impose that it is located on the orthocenter of the triangle formed by the vanishing points.

This method is sound in theory, however, in practice, it is not so precise. Because of the imprecision on the selection of the lines and the fact that the vanishing points are very far, small mistakes are amplified and can move the result too much. To deal with this, we select two triangles and average their orthocenters.

### 3.5 Results

The newly found results are:

$$\begin{aligned} (u_0, v_0) &= (563, 560) \\ (f_x, f_y) &= (591, 419) \rightarrow a = 1.4 \\ \omega &= \begin{bmatrix} 1.9 & 0 & 1118 \\ 0 & 1 & 560 \\ 1118 & 560 & 1.12 * 10^6 \end{bmatrix} \\ K &= \begin{bmatrix} 592 & 0 & 562 \\ 0 & 420 & 560 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

The new principal point is better than the previous one, but we can still notice that it is not very compatible with our intuition.

visibly

## 4 G3 - 2D reconstruction of a vertical section

Because we now have a calibrated image we can use a more straightforward procedure with respect to the one we used for the horizontal rectification.

### 4.1 Image of the vanishing line

In this case, we are only missing the image of the vanishing line  $l'_\infty$  which belongs to the set of planes parallel to  $\Pi$ , that is the plane in which the facade 3 lies. We get  $l'_\infty$  from the cross product of two vanishing points  $v_1, v_2$  in  $\Pi$  (obtained from two pairs of parallel lines 29):

$$l'_\infty = v_1 \times v_2 \quad (12)$$

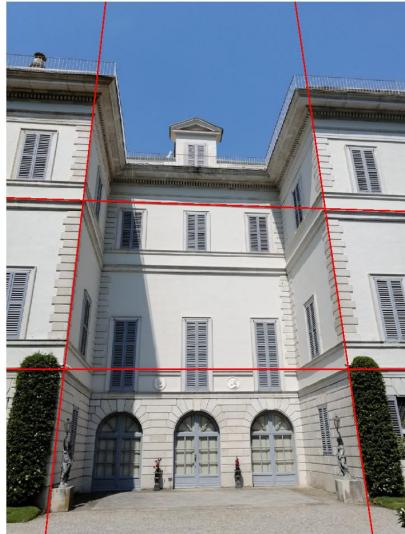


Figure 29: Parallel lines on a vertical facade

### 4.2 Image of the absolute conic

In the calibration process, we already obtained  $\omega$ , but in case we didn't, we could easily calculate it from  $K$ :  $\omega = (K * K^T)^{-1}$ .

### 4.3 Image of the conic dual to the circular points

Since we have the image of the absolute conic and the horizon for  $\Pi$ , we can find the image of the circular points  $I', J'$  as their intersection:

$$\begin{cases} \omega_{1,1}x^2 + 2\omega_{1,2}xy + \omega_{2,2}y^2 + 2\omega_{1,3}x + 2\omega_{2,3}y + \omega_{3,3} = 0 \\ l'_{\infty 1}x + l'_{\infty 2}y + l'_{\infty 3} = 0 \end{cases} \quad (13)$$

From the circular points, we then proceed to find the image of the conic dual to the circular points as:

$$C'^*_\infty = IJ^T + JI^T = \begin{bmatrix} -0.93 & -0.32 & 0.0001 \\ * & 0.63 & -0.0003 \\ * & * & 0 \end{bmatrix} \quad (14)$$

### 4.4 Rectifying matrix

Once we have  $C'^*_\infty$  we can find the rectifying matrix  $H_v$ :

$$C'^*_\infty = H_v^{-1} C^*_\infty H_v^{-T} = SVD(C'^*_\infty) \Rightarrow H_v = \begin{bmatrix} 0.98 & 0.19 & 0 \\ -0.2296 & -1.18 & 0.0006 \\ 0.0001 & -0.0005 & -1 \end{bmatrix} \quad (15)$$

## 4.5 Results

From fig.30 we can see that the reconstruction did not come out perfect: the walls look parallel but not very orthogonal.

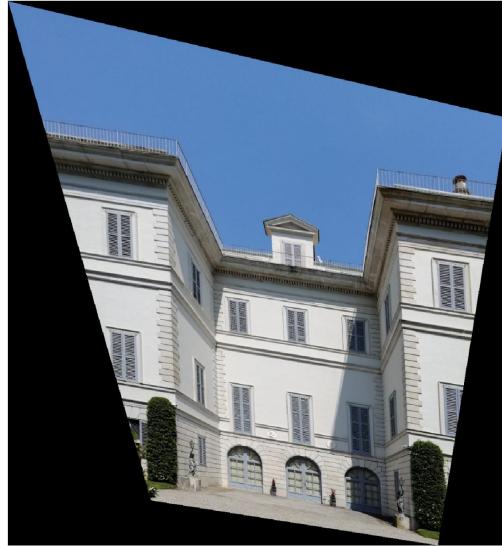


Figure 30: Vertical reconstruction

We can verify the quality of our estimate by checking if the lines we supposed being orthogonal are now indeed orthogonal. The angles between the orthogonal lines that we selected in fig.29 are both of  $\theta = 100$  degrees, and confirm that the image is not too accurate from this point of view.

We can also verify if the lines are now parallel by checking their cross products, which ideally should now be zero. We get that each component of the cross products is in the order of  $10^{-3}$  in the worst case.

## 5 G4 - Localization

### 5.1 Theoretical solution

To localize the camera with respect to wall 3, we can exploit the knowledge of the mapping between such wall and a similitude of such wall, and that the camera has been calibrated.

We are interested in the position of the camera in relation to the wall, however, it is easier to work indirectly by first finding the opposite. We impose that the world coordinates  $\{x, y, z\}$  coincide with the camera ones (so it has rotation  $R = I$  and translation  $t = 0$  with respect to the world), and we fix a coordinate system  $\{i_\pi, j_\pi, k_\pi\}$  with origin in  $o_\pi$  associated to wall 3 (see fig.31). We construct the reference systems such that a point  $X$  on the plane surface has homogeneous coordinates  $X_\pi = [x, y, 0, w]'$  in the plane reference, which can be written in terms of the world reference as:

$$X_w = \begin{bmatrix} i_\pi & j_\pi & k_\pi & o_\pi \\ 0 & 0 & 0 & 8 \end{bmatrix} X_\pi = \begin{bmatrix} i_\pi & j_\pi & o_\pi \\ 0 & 0 & 1 \end{bmatrix}, \quad x_\pi = \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (16)$$

If the camera reference is such that the image plane lies on  $z = 0$ , the projection of  $X$  can be written as:

$$u = P X_w = K \begin{bmatrix} i_\pi \\ j_\pi \\ o_\pi \end{bmatrix} x_\pi \quad (17)$$

because  $P = [KRK^T]$ , and as we said,  $R = I, t = 0$  by construction. Having the homography  $H$  between the plane in the space and in the image:

$$K \begin{bmatrix} i_\pi \\ j_\pi \\ o_\pi \end{bmatrix} x_\pi = H x_p \rightarrow \begin{bmatrix} i_\pi \\ j_\pi \\ o_\pi \end{bmatrix} = K^{-1} H \quad (18)$$

The location and rotation of the plane w.r.t. the camera is represented by

$$\begin{bmatrix} i_\pi & j_\pi & k_\pi & o_\pi \\ 0 & 0 & 0 & 8 \end{bmatrix}, \quad k_\pi = i_\pi \times j_\pi \quad (19)$$

and the reverse relation can be extrapolated by inverting that matrix.

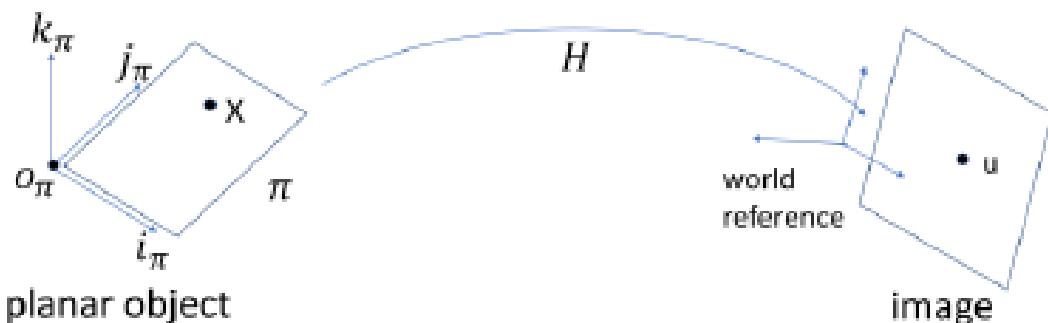


Figure 31: The reference systems

### 5.2 Adjustments

Because of the imprecision on the estimation of the rectifying homography, we can rebuild it to improve the result. To do this, we are going to map the corners of facade 3 on a rectangle whose aspect ratio is extracted by using the one obtained from the previous rectification (around  $a = 1.24$  considering the height of the wall as the distance between the ground and the topmost decorations/shadows). It is built by mapping the bottom-left corner of facade 3 to  $(0, 0)$  and the top-right on  $(1000, 1000 * a)$ .

To interpret the resulting matrix as a roto-translation, we also need to take into account the small quantities that pollute our numerical result by introducing unwanted scaling factors on the rotation part. To deal with this we can isolate the rotation component of the matrix (its up-left 3x3 sub-matrix) and apply SVD to consider only the rotation.

$$H = \begin{bmatrix} 2.6 & -0.1 & 0 \\ 0.2 & -1.9 & 0 \\ -825 & 2495 & 1 \end{bmatrix} \quad (20)$$



Figure 32: New vertical rectification

### 5.3 Results

The matrix describing the position of the camera with respect to the wall, is reported numerically in eq.21 and visually in fig.33. We read the result as the composition of a rotation and a translation:  $M_{wall \rightarrow camera} = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix}$

$$M_{wall \rightarrow camera} = \begin{bmatrix} 0.999 & 0.044 & -0.029 & 393 \\ 0.046 & -0.99 & 0.079 & 595 \\ -0.025 & -0.081 & -0.99 & 256 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

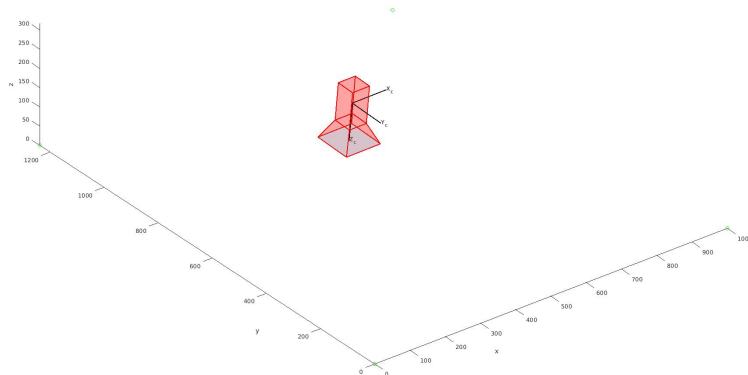


Figure 33: The camera in relation to facade 3 (whose corners are described by the green circles)

We can already observe that the result is not precise, as the camera looks too high and close to the facade. This suspect is confirmed also when translating the position coordinates from pixels to meters, indeed we obtain  $camera\ position = (0.803, 1.5, 0.457) \text{ meters}$ , which is too close to the wall and would imply that the building would be less than 2 meters high.