Homework Image Analysis and Computer Vision

Sofisti Giorgio (10500171)

giorgio.sofisti@mail.polimi.it

Politecnico di Milano

A.Y. 2019/2020

January 27, 2020

**Contents**

1. **Iintroduction**
2. **Features Extraction**
3. **Shape Reconstruction**
4. **Metric Reconstruction**
5. **Camera Calibration**
6. **Vertical façade Reconstruction**
7. **Camera Localization**
8. **Analitical Details**
9. **Computed Results**
10. **References**

**1 Introduction**

Images provided for this homework are taken inside Sciarra palace’s internal courtyard situated in Trevi neighborhood, Rome. The palace built between 1885 and 1888 is a great representation of Liberty architecture with complex, flower inspired, decoration over all the facades. The inside courtyard is covered by a glass roof that let the sunlight penetrate inside this connecting area between two streets. The two images provided shows one internal façade from the bottom of the court. Since the images are taken during a sunny day (probably around noon) there is a high contrast between areas illuminated by sunlight, that penetrates through the glass roof, and the areas that are in the dark, a point to consider during edge detection. Another point that introduce complexity for features detection are for sure the complex decorations that are present on the facades.

Last but not least we have to consider the assumption given in the homework description about metric dimension of windows (fundamental for metric reconstruction step), orthogonality of adjacent façades and skew symmetry assumption (but not natural camera assumption) fundamental when we need to look for K calibration matrix.

**2 Features Extraction**

In order to extract relevant features form the image we have to perform a series of steps:

1. In features extraction colors are useless information to carry on that complicates and worsen the performance so we can apply rgb2gray(image) to transform the image form RGB to Gray Scale.
2. As mentioned in the introduction, since the image presents areas with different exposition we need to perform a pre-processing phase to improve accuracy and obtain better result in next phases. To do that we tested different normalization algorithms and evaluated that MATLAB function adapthisteq(image)applied twice on the image returns the best result.
3. Now we’re ready to apply to the image edge extraction algorithm. After some testing of different option (sobel, log, roberts, canny) we opted for Canny method that is the most accurate. Canny is accurate but the threshold imposed is the result of many trial and error and is fundamental for good performance of the next step.
4. Now we are ready to use the learned method of Hough transform to extrapolate from canny image relevant edges. After some parameter tuning the result in the picture below.

**3 Shape reconstruction**

Now that we have obtained some relevant image features we are ready for image reconstruction. The process involves the application of two subsequent transformation one after the other that maps the original image to the affine one ad finally to the Euclidean.

**3.0 Vanishing Points and Lines**

After the selection of relevant features for each direction (x, y, z) I’ve computed the vanishing lines and found three best approximation of vanishing points as weighted average of the vanishing points given by the intersection of vanishing lines in the respective directions. An important factor to be consider to obtain reliable and good results is to choose as many features as possible for each direction (at least 3-4 edges, definitely 2 are not enough) so that the approximation is more accurate. Below is shown the result obtained during the computation.

**3.1 Affine property recovery**

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.

In order to identify the correct transformation matrix for recovering affinity property I’ve found the line at the infinity passing through two of the three found vanishing points found at the previous step (I have to choose the two vanishing point based on which façade I want to rectify, in my case I have chosen the two bottom ones to rectify the image in the plane parallel to ground, for better understanding of next steps I’ll call it XZ). [pag. 49 book]

Where the last row is built using the tree values of the found line at the infinity.

As you can see now the lines on XZ direction are represented also in the image as parallel lines that intersect in the point at infinity instead of meeting in the vanishing points (main façade, direction Y (vertical direction) is not rectified since we’re not interested for now to work on that).

**3.2 Euclidean rectification**

Now that I’ve recovered affinity property the next step is to impose orthogonality between façade. In order to perform this operation we have to build another transformation matrix imposing two conditions [pag. 54 book]:

I’ve basically found two couple of lines, chosen form the edge detection step, in X and Z direction that are orthogonal in reality () and imposed that since should be 90° apart. I’ve computed the solution of the linear system finding the conic at the infinity and then using single value decomposition I’ve found the three matrices and used them to build that maps the affinity to an Euclidean reconstructed image.

Then we decompose S as:

And we obtain:

Finally we find:

**3.3 Additional transformation applied**

For a better visualization I’ve applied a rotation to the image of 92° in order to have X axis parallel to the image and I’ve scaled the image to reduce computation time and space requirements that otherwise would have overtaken computer specs.

**3.4 Metric property recovery**

Based on information provided on window size I’ve imposed that the dimension of coplanar top window size should measure the same length in pixel and applied a non-isomorphic scaling. This basically means that:

The composition of all the transformation matrix applied maps the original image to the final metric reconstructed one.

**4 Camera calibration**

In this step we want to estimate the camera parameters knowing that the camera is skew-symmetric. Based on the theory [pag. 224 book] we need tree vanishing points and at least one more constraint since I cannot assume natural camera. I’ve used the constraint derived from the H matrix since we are in the case of metric plane with known homography (computed in previous steps).

K matrix assume the form of:

K follow the relation:

In order to compute K I’ve imposed 4 constraint plus skew-symmetry constraint to , image of the absolute conic [pag. 226 book]. In our case K will assume the form of the symmetric matrix:

We can now apply the reconstructive transformation [pag. 211 book] on the horizontal façade. The three steps are:

1. For each square compute the homography H that maps its corner points, to their imaged points.
2. Compute the imaged circular points for the plane of the square as. Writing , the imaged circular points are .
3. Fit a conic to the six imaged circular points. The constraint that the imaged circular points lies on ma be rewritten as two real constraints.

If lies on then and the imaginary and real parts give respectively:

Which are equation linear in The conic is determined up to a scale form five or more such equations.

1. Compute the calibration matrix K from using Cholesky factorization.

Just for reference during the computation I’ve also computed the K matrix imposing natural camera and skew-symmetry constraint. We are pretty close to the natural camera case.

**5 Main façade reconstruction using K**

Then the homework requires to use the knowledge of K to reconstruct the main façade. To perform this operation, I’ve built a matrix from K capable of leveraging on the information fitted inside to map the original image to a rectified one over the XY plane. The matrix has the form of:

Using this matrix, I’ve built the transformation matrix that projects the façade points and rectify them. The resulting image is an Euclidean reconstruction of the central façade itself and has been scaled down significantly for reducing computation time.

**6 Camera pose estimation**

Last point of the homework requires to estimate the approximative camera position in 3d space.