3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

ABSTRACT

This project exposes how to compute an homography between two images applying normalised DLT algorith matching SIFT points for further build a mosaic with two or more images . As optional parts, it will be done a plannar camera calibration and logo detection and insertion.

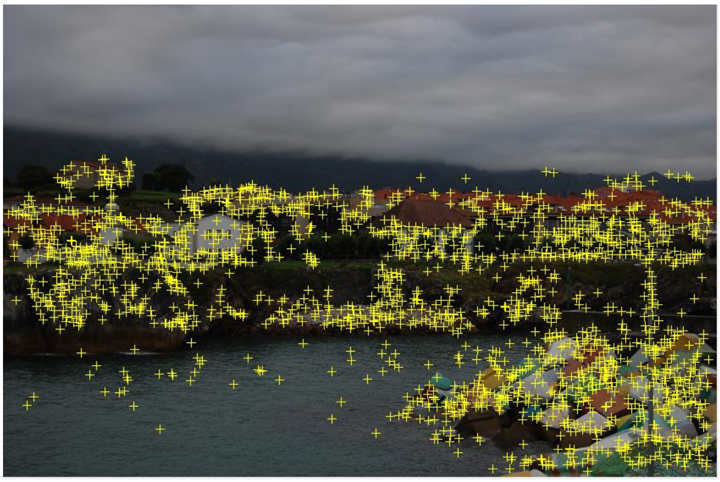
A base code and functions are provided. All the code is attached parallel to this document.

Code split in 3 files!! LAB2, PLANNARPATTERN, LOGOREPLACE

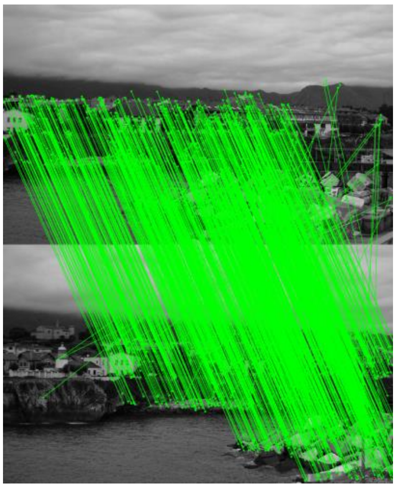
IMAGE MOSAICKING

Once the images are loaded, SIFT points are extracted and matched among three images, the matches between a and b and between b and c.

-Sift points from image 'b':



-Sift matched points between a and b:



Now there are some reference points to start playing with. In this case, it is desirable to get the homography matrix H that relates 2 images using the matched SIFT points. To do so, the following function is used:

- function 'ransac_homography_adaptative_loop'
  - ✓ **Input**: matched points from image a, matched points from image b, threshold to posteriorly compute inliers and an initial number of maximum iterations.
  - ✓ **Output**: points that are considered inliers of the two imges and the homography matrix H that relates these two images.

This function provides the homography matrix using RANSAC-loop algorith, that basically picks pairs of random matched points and compute the 2D homography between them, to later examinate them to check whether they are the best inliers so far or not, until a maximum number of iterations.

To calculate the 2D homography 2D a new function is created:

3D RECOVERING OF URBAN SCENES LAB 2

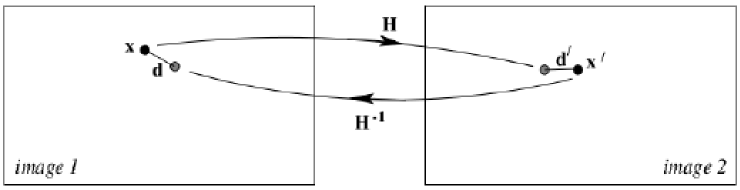
Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

- function 'homography2d'
  - ✓ **Input**: 4 pairs of matched points and
  - ✓ **Output**: homography matrix H that relates these two images.
  - ✓ How has this function been designed? Using the Normalized DLT algorithm [*Hartley and Zisserman].* To diminish the effect of arbitrary selection of origin and scale in the coordinate frame of the image, normalization transformation:
    1. Normalization of the points of both images: from x and x' → ẋ and ẋ' by computing a similarity transformation T and T' such that the centroid of the points ẋ is the coordinate origin and their average distance from the origin is $\sqrt{2}$.
    2. Application of the DLT algorithm that given A_i extracts the H matrix as a vector of the last column of the SVD decomposition.
    3. Denormalization of the H matrix such that $H = T'^{-1}\widetilde{H}T$

Once the 2D homography function, go back to RANSAC-loop function and get into detail with it. RANSAC-loop algorith is an iterative procedure for checking what are the 4 points which have more inliers to choose be chosen as best points to compute the homography.

To estimate the inliers symmetric geometric error is considered from the next function:

- function 'compute_inliers'
  - ✓ **Input**: All the matched points from image a and b, the case study H and a threshold.
  - ✓ **Output**: points that are inliers using the H matrix as an homography
  - ✓ How was this function designed? Symmetric geometric error is calculated from each one of the points:



And then selected those points that had an error smaller than the set threshold.

Up to now, matched pairs of points between image a and b have been defined, being able to run the RANSAC-loop algorith to get the best homography that relates both images using the function homography2d and computing inliers.

In this way, we can plot the image that shows us the matched points chosen as inliers.

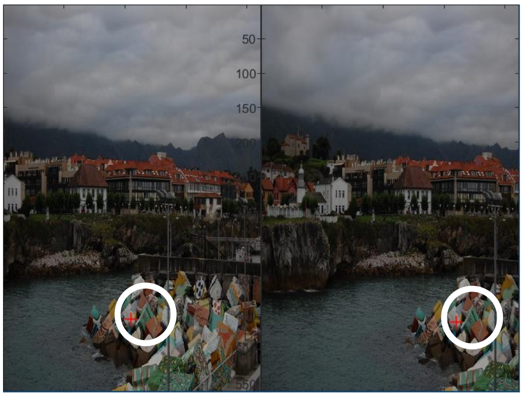
3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

[1]

To check how the homography matrix behaves, it is proposed a montage between both images in order to visually check the homography by clicking on the left image and observing the homologous transformed point on the right. It can be seen that the homography works correctly using 2 different points:

[2]



[3]

Same verifiying process is done for images b and c to compute the homography that relates those two images. Once computed the transformation of the two images it is possible to draw the mosaic using the original images and the two different homography matrices. To build the mosaic, set the image b as a reference and apply an homography as shown below:

```
iwb = apply_H_v2(imbrgb, eye(3)    , corners);
iwa = apply_H_v2(imargb, Hab       , corners);
iwc = apply_H_v2(imcrgb, inv(Hbc)  , corners);
```

[1] Image of the inliers matches points between image a and b
[2] Image showing the projected poitnts, to make sure we got a good homography matrix.
[3] Image showing the projected poitnts, to make sure we got a good homography matrix.

3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

There are different data for trying the mosaics:

1. **Llanes**: the mosaic works well because there is only rotation on the camera, no translation between the 3 different images:

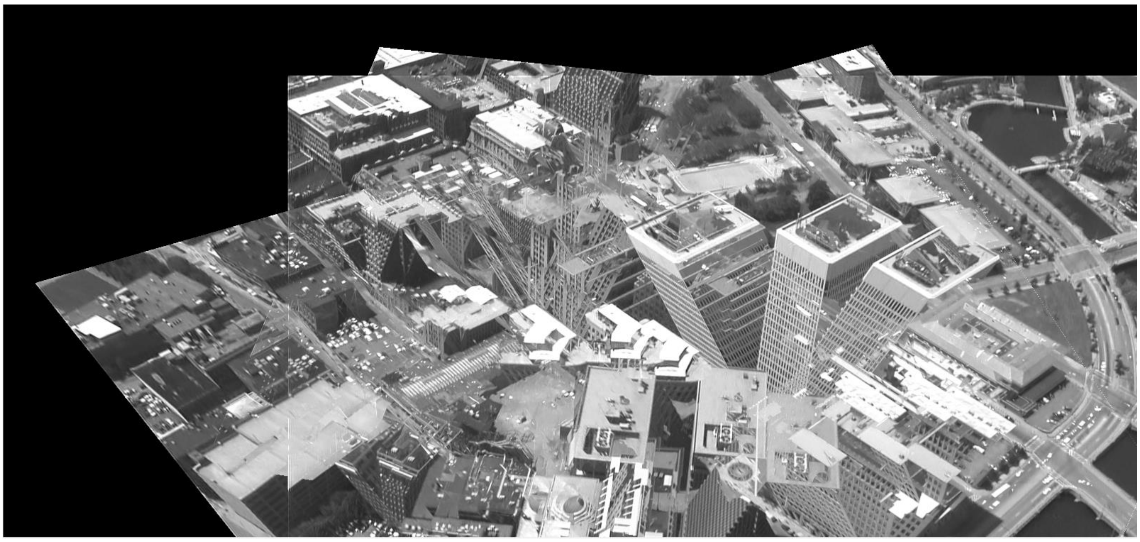
2. **castle_int images:** the mosaic does not work well because there is rotation and translation on the camera between the 3 different images:

3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

3. **aerial images set 13**: the mosaic works well because there is only rotation on the camera, no translation between the 3 different images:

4. **Aerial images set 22:** the mosaic does not work well because there is rotation and translation on the camera between the 3 different images:



3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

GOLD STANDARD ALGORITHM TO REFINE THE HOMOGRAPHY

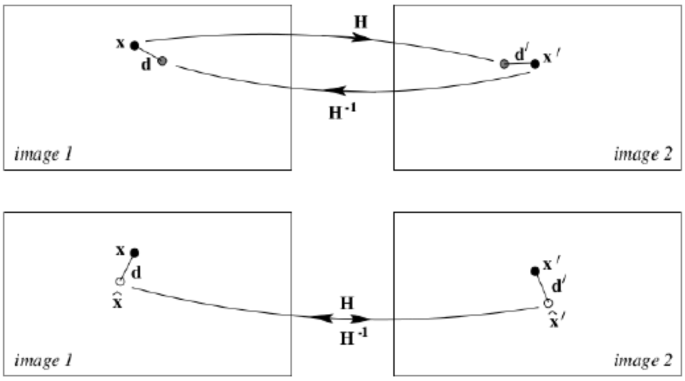
In this part, there is applied the Gold Standard algorithm to refine the homography applied before by using the best inliers points and distances between the projected and normalized points.

To do so, 1st calling the points:

```
x  = points_a(1:2, matches_ab(1,inliers_ab));
xp = points_b(1:2, matches_ab(2,inliers_ab));
```

After this, create a function to compute the distance between the real points and those projections:

- function 'gs_errfunction'
  - ✓ **Input**: Observed values and H provisional matrix.
  - ✓ **Output**: Euclidean distance of the projection and the normalization errors.
  - ✓ How was this function designed? We calculated the symmetric geometric and the reprojection error:



Afterwards, applying a matlab function called 'lsqnonlin' which finds the minimum of the sum of squares of the two different errors by changing the parameters on H. In this way we will get a new and definitive H to apply the projection with.

Once done this last step and got the definitive H we can point out that the sum of the errors did slow down by more than 35%. An image can show the small changes on the projection matrix H that the Gold Standrad Algorithm is doing on our points:

3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

CAMERA CALIBRATION

Plannar-pattern camera calibration was proposed by *[Z. Zhang], "A flexible new technique for camera calibration*", in 00's. An image point x=(u,v,1) can be obtained through pinhole camera model as x=sK[R t] X. Being X=[X, Y, Z, 1] any 3D point in homogeneus coordinates and s an scaling parameter.

Fig. CC1: Proposed pattern

Without losing of generality it is assumed the patter plane is placed at Z=0, therefore the homogeneus transformation can be treated as an homography: **K[r1 r2 t] = λ[h1 h2 h3].**

Due to orthogonality of rotation matrices there are two constrains:

$$\left.\begin{array}{r} \mathbf{r}_1^T\mathbf{r}_2 = 0 \\ \mathbf{r}_1^T\mathbf{r}_1 = \mathbf{r}_2^T\mathbf{r}_2 = 1 \end{array}\right\}$$

Isolating [r1 r2 t]= λK^{-1} [h1 h2 h3] the image of absolut conic is obtained:

$$\left.\begin{array}{r} \mathbf{h}_1^T K^{-T}K^{-1}\mathbf{h}_2 = 0 \\ \mathbf{h}_1^T K^{-T}K^{-1}\mathbf{h}_1 = \mathbf{h}_2^T K^{-T}K^{-1}\mathbf{h}_2 \end{array}\right\}$$

Where $w=K^{-T}K^{-1}$, which provides 2 constrains (equations) therefore it is necessary 3 different views of the plannar pattern to build a system of equations to calculate w. Since orthogonal matrices are unitary and due to the constrains applied, the solution is the nullspace:

$$V\Omega = \mathbf{0}$$

$$\min_{\Omega} \|V\Omega\|_2$$

$$\text{such that } \|\Omega\|_2 = 1$$

Where Ω=(w11,w12…w33). This solution is up to scale, therefore size of images may vary.

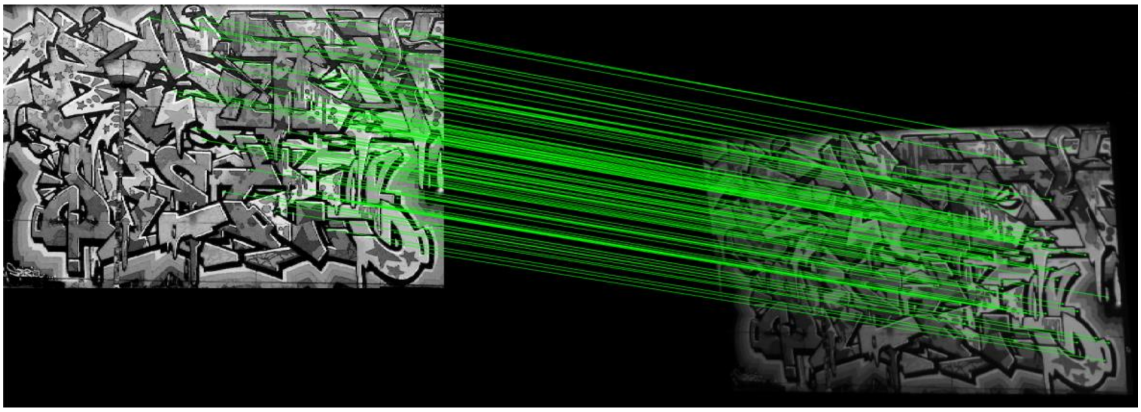


Fig. CC2: SIFT matching example.

Solving the external parameters is just replacing the formula [r1 r2 t]= λK^{-1} [h1 h2 h3], where λ allows to fit the unitary constrain of R, which dismiss the scale factor ambiguity defining the optical center. Since it is the Point which projects the principal point on the image (this is the points whose coords [u,v]=[0,0] the Center of Projection must be the nullspace of P, such that K[R t]*[Xcp, Ycp, Zcp, 1]'=[0,0,0].

$$\mathbf{r}_1 = \frac{K^{-1}\mathbf{h}_1}{\|K^{-1}\mathbf{h}_1\|}, \quad \mathbf{r}_2 = \frac{K^{-1}\mathbf{h}_2}{\|K^{-1}\mathbf{h}_2\|}, \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{t} = \frac{K^{-1}\mathbf{h}_3}{\|K^{-1}\mathbf{h}_1\|}$$

It is interesting to highlight both interpretations an homography can have. It could be considered as a camera rotating around the pattern (static), or the pattern rotating around a static camera.

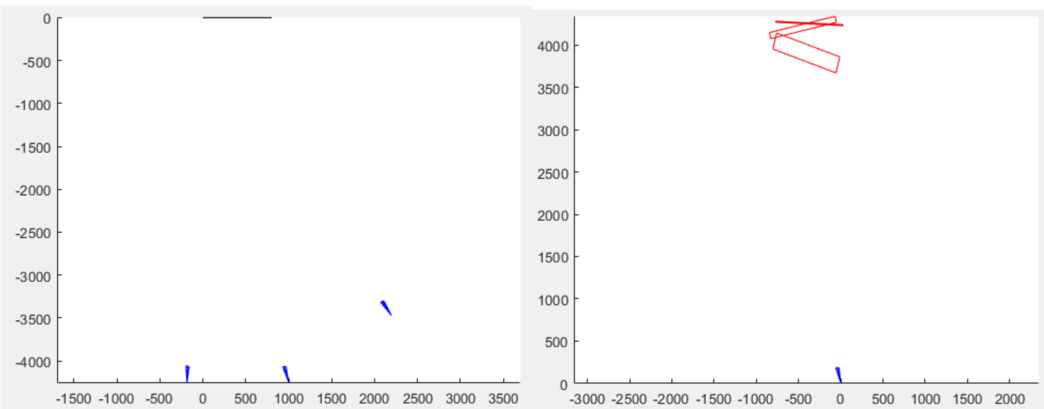


Fig.CC3: Static pattern (left) vs static camera (right) relative moveme

Once fundamental matrix is obtaned a set of possibilities is reachable, such as augmented reality.

3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

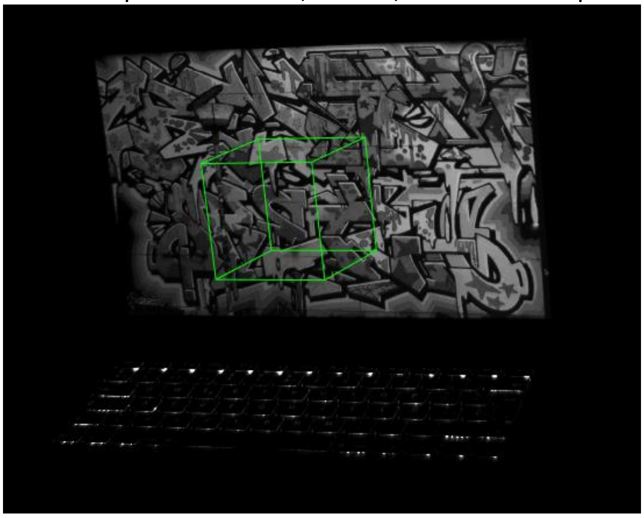


Fig.CC4: Augmented Reality example

3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

LOGO DETECTION AND INSERTION

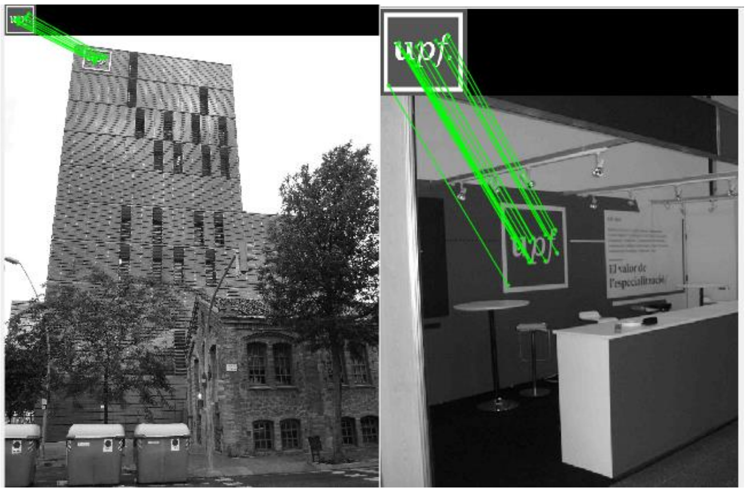
As final requirement we were supposed to replace UPF logo by Master CV logo in an image of UPF's building using DLT algorthm.

The way of proccedure has been the following (fully automatic):

1. Computing SIFT points and matching SIFT points into a UPF pattern logo and the target image.
2. Computing the relating homography using DLT algorithm + RANSAC (no RANSAC-loop algortihm has been applied in building case. The poor amount of SIFT points in the patter provokes worse results)
3. Resize MasterCV logo to UPF logo to obtain a good transformation.
4. Create a black background image with transformed MasterCV logo within (in the proper location).
5. Overwriting logo by creating a ground-truth matrix (A):
```
ED2(:,:,i)=CVC2(:,:,i)+uint8(~A).*ED(:,:,i);
```

3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer



SIFT matching at left, UFP building; at right UPF stand.

3D RECOVERING OF URBAN SCENES LAB 2

Juan Felipe Montesinos, Yi Xiao, Ferran Carrasquer

CONCLUSIONS

Once done this project, we can extract some points to consider:

- The way the points of interest are extracted is critical. In this case, extracting a good sift points and matching them among several images is something that really needs to be done with a lot of precision.
- The mosaics only work when there is only rotation of the camera, no translation. In this way, you may rotation the camera over a view, not translate the camera over some object.
- Using Gold Standard Algorithm we can refine the homography by some important index.
- Plannar pattern calibration is a interesting method to obtain the camera parameters. Applying some assumtions it is possible to do it with a single image!
- 3D vision balance goodness of algebra and geomtry in a beautiful way.