M4 - 3D Vision

3D recovery of Urban Scenes

Ana Caballero - ana.caballeroc@e-campus.uab.cat
Arnau Vallvé - arnau.vallve@e-campus.uab.cat
Claudia Baca - claudiabaca.perez@e-campus.uab.cat
Joaquim Comas - joaquim.comas@e-campus.uab.cat

INTRODUCTION

The goal of this project is to learn the basic concepts and techniques to reconstruct a real world scene given several images (points of view) of it, not necessarily previously calibrated. In this project we focus on 3D recovery of Urban Scenes using images of different datasets, namely images of facades and aerial images of cities.

WEEK 2

Goal: compute the homography that relates to images

Mandatory:

- Estimate homography with the normalized DLT algorithm.
- Complete RANSAC function and the lab2.m file.
- Compute mosaic with four different sets of data.
- Estimate homography with the Gold-Standard algorithm.

Optional:

- Camera calibration using a planar pattern.
- Detect a logo.
- Replace a logo.

1. Compute image correspondences

We start with three images of a landscape containing overlapping parts whose keypoints will be computed using SIFT descriptors as seen below

Image 1: Compute SIFT key points for the images.
From left to right: llanes_a.jpg , llanes_b.jpg, llanes_c.jpg

and matched together between image pairs.

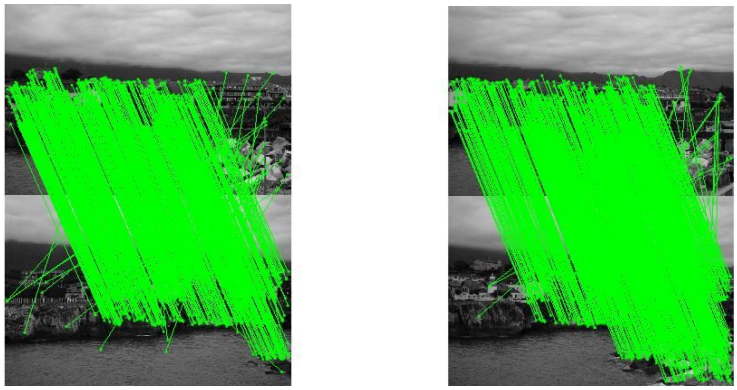

Image 2: Match SIFT key points between a pair of images.
Left: llanes_a.jpg and llanes_b.jpg, right: llanes_b.jpg and llanes_c.jpg

As we can qualitatively see from the pair correspondances, the displacement from the landscape between each of the image pairs is matched by the position of the keypoints as expected.

2. Normalized DLT Algorithm. Homography between image pairs

The goal of the DLT is to obtain a 2D homography in our case given a sufficient set of point correspondences. Inside the function ransac_homography_adaptive_loop.m the function homography2d.m is created in order to obtain the 2D homography matrix H given the matched pair values. To make the algorithm more stable, we can use the Normalized version. For the Normalized DLT we need first to do a similarity transformation to the data so that we have a new set of points that are centered on the centroids with an average distance from the origin of $\sqrt{2}$. This will be also applied to the correspondant points of the second image from the correspondent image pair.

After normalizing the data, using the found SIFT matches, the matrices A_i are computed for each of the correspondences (set of points x_i are the points that correspond to x_i').

$$\begin{pmatrix} \mathbf{0}^T & -w_i'\mathbf{x}_i^T & y_i'\mathbf{x}_i^T \\ w_i'\mathbf{x}_i^T & \mathbf{0}^T & -x_i'\mathbf{x}_i^T \end{pmatrix} \qquad \mathbf{x}_i = (x_i, y_i, w_i),\ \mathbf{x}_i' = (x_i', y_i', w_i')$$

and assembled forming a matrix A. With the SVD decomposition of A we will obtain the values of the homography matrix H as follows: For an A = UDV' as the SVD decomposition of A from the last column of V we will obtain the h values of the homography matrix. For now, we have been working with the normalized points, so we actually haven't computed H denormalized. To obtain it a denormalization step is used, where T are the similarity transformations used to normalize the points on the first place.

$$H = \mathcal{T}'^{-1}\widetilde{H}\mathcal{T}$$

For the RANSAC robust estimation we will be using the symmetric geometric error.

$$\min_{H,\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_i'} \sum_{i=1}^{n} \|[\mathbf{x}_i] - [\hat{\mathbf{x}}_i]\|^2 + \|[\mathbf{x}_i'] - [\hat{\mathbf{x}}_i']\|^2 \quad \text{such that } \hat{\mathbf{x}}_i' = H\hat{\mathbf{x}}_i,\ \forall i$$

After computing the homography with the RANSAC using Normalized DLT Algorithm, here we show some of the results when playing with the homography. Qualitatively, the correspondances seem to properly correspond for this images in the regions that include both images, and no correspondences are found in regions of one image that do not have correspondance on the image pair such as bounding regions, as expected (see below).

Image 3, 4: Examples of point correspondences between image pairs that match, as they are present on both images (i.e. overlapping regions)

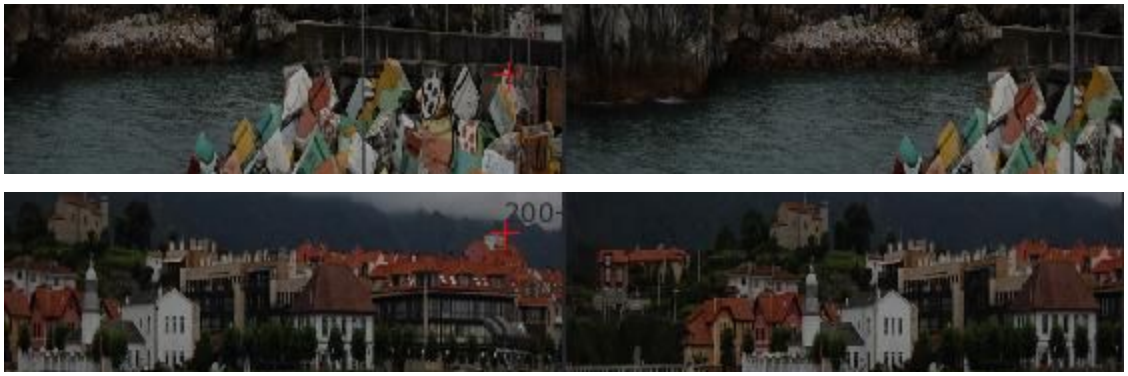

Image 5, 6: Examples of point correspondences between image pairs that have no matches, as they are not present on both images (i.e. non-overlapping regions)

3. Build the mosaic for every image

Using the computed homographies, we can now proceed to build the mosaics for the provided images. We build three different homography matrix and applied it over the corresponding image, obtaining a transformed image. The final image image is the result of the maximum value of the pixel (x,y) for every transformed image.

Mosaic “llanes”

For the “llanes” images, visually, the mosaic seems good. As the images show objects which are far away and images are overlapping there is no issues when computing the correspondences between points.

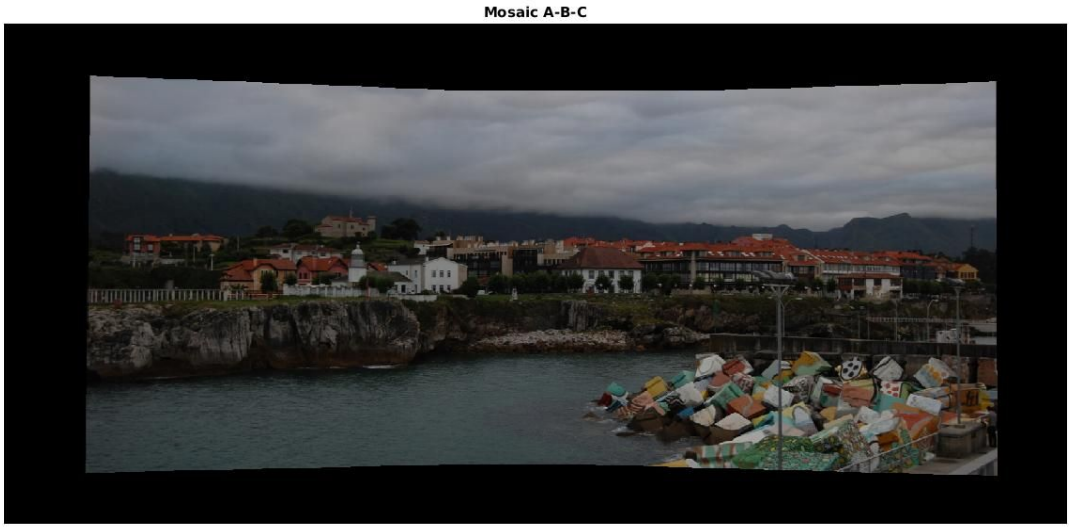

Image 7: Mosaic A-B-C images using “llanes” data.

Mosaic “castle_int”

In the case of the “castle_int” mosaic, we can see some evident issues. Even though the building seems to correspond, the truck appears as a ghostly image. So we have correspondences in the background castle but not with the truck. To see why this happens, we can analyze the homography correspondence images.

Image 8: Mosaic A-B-C images using “castle_int” data

We can see below that as we have an object close to the camera, the camera movement produces perspective changes that translate in the vehicle occluding different parts of the background castle depending on the perspective, due to the 3D nature of reality. Looking at the images below, it's clear that the the red cross pointing at a window on an image is being correctly corresponded to the one on the other image, but the truck isn't. If we have objects close to the camera and objects far from the camera as for now the perspective changes will accentuate and produce this issues and will produce undesired results when, for example, making a mosaic as we have seen.

Image 9. 10: Example visualization of point correspondences

Mosaic "aerial_13"

Using the "aerial_13" images containing far away images taken from an aerial perspective, visually, the mosaic appears to be correctly made as the images fit together nicely.

Image 11: Mosaic A-B-C images using "aerial_13" data

Mosaic "aerial_22"

Similarly to the case seen with the "castle_int" images, because of the perspective changes produced by camera movement, the 3D nature of reality as captured by the different images will produce points of view which cannot be put into the mosaic together by just overlapping, without doing non-linear transformations of the image to fit the different perspectives.

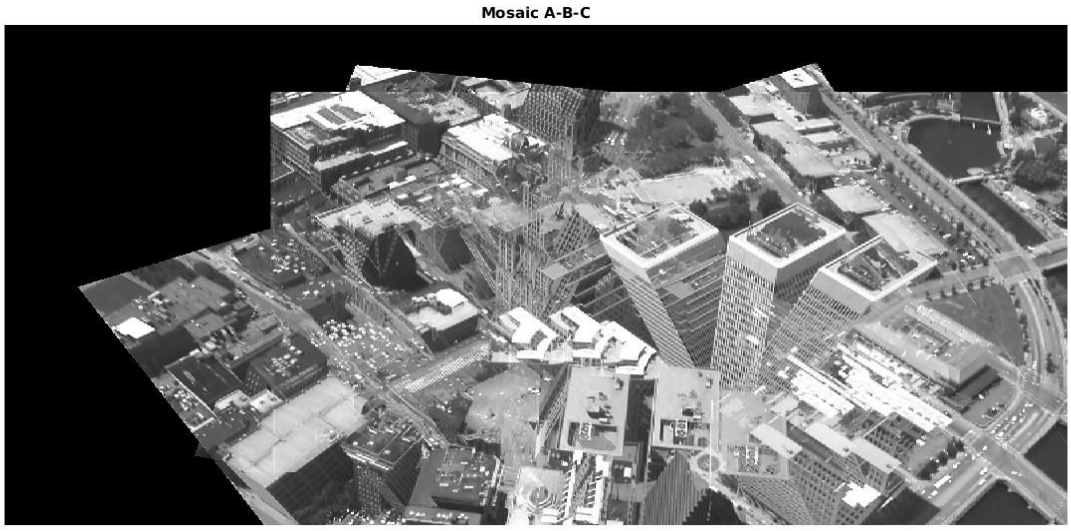

Image 12: Mosaic A-B-C images using "aerial_22" data

4. Estimation of the homography with the Gold-Standard algorithm

We set the non-homogeneous point coordinates and their correspondences. Then, apply the Gold Standard algorithm (+ info https://bit.ly/2SGiJRt) and plot the geometric error before and after the refinement. With this new keypoints, we are able to build the mosaic.

Image 13, 14, 15, 16: Keypoint locations (yellow = before ; cyan = G.S. refined)

Numerical results

From the obtained results, we can see how the reprojection error was reduced comparing before and after refinement, numerically proving a success in refining the homography for the evaluated cases.

"Llanes" data:

- Refined homography H_ab:
  Gold standard reproj error initial 605.155982, final 145.238230
- Refined homography H_bc:
  Gold standard reproj error initial 241.004381, final 63.184092

"Castle_int" data:

- Refined homography H_ab:
  Gold standard reproj error initial 10877.367333, final 1622.997281
- Refined homography H_bc:
  Gold standard reproj error initial 1125.476578, final 322.405245

"Aerial_13" data:

- Refined homography H_ab:
  Gold standard reproj error initial 126122.746854, final 30054.849841
- Refined homography H_bc:
  Gold standard reproj error initial 76884.267176, final 19286.845705

"Aerial_22" data:

- Refined homography H_ab:
  Gold standard reproj error initial 4220.386758, final 1422.233809
- Refined homography H_bc:
  Gold standard reproj error initial 41104.486113, final 8950.336734

Mosaics refined Gold Standard

Image 17, 18, 19, 20: Mosaics refined GS

5. OPTIONAL: Calibration with a planar pattern

The camera calibration with a planar pattern or in other words the calibration using the image of absolute conic, consists in computing the IAC (w), image of the absolute conic, and with that extract the K, the camera calibration matrix , knowing that w=(K*K^-1).

First, as we have seen, the IAC(w) is computed, in order to achieve that we compute the matching between the images and the pattern, having then the 3 homographies that relates each image with the pattern(x=H*x'). Then knowing that w=K*K^-1(*) (so hi*w*hj=0) , w can be computed as a solution this system of equation:

$$\mathbf{h}_i^T \omega \mathbf{h}_j = \begin{pmatrix} h_{1i} & h_{2i} & h_{3i} \end{pmatrix} \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{pmatrix} \begin{pmatrix} h_{1j} \\ h_{2j} \\ h_{3j} \end{pmatrix} = A * w$$

$$where \quad A = (h_{1i}h_{1j},\ h_{1i}h_{2j} + h_{2i}h_{1j},\ h_{1i}h_{3j} + h_{3i}h_{1j},\ h_{2i}h_{2j},\ h_{2i}h_{3j} + h_{3i}h_{2j},\ h_{3i}h_{3j})$$

(*)

Demo: this can be demonstrated by assuming the plane Z=0 (because it is a planar pattern) so the projection can express the next relation:

$$x = K \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad \begin{cases} \begin{pmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{pmatrix} \sim \begin{pmatrix} K\mathbf{r}_1 & K\mathbf{r}_2 & K\mathbf{t} \end{pmatrix} \\ \begin{pmatrix} K^{-1}\mathbf{h}_1 & K^{-1}\mathbf{h}_2 & K^{-1}\mathbf{h}_3 \end{pmatrix} \sim \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{pmatrix} \end{cases}$$

And then we know that r1 and r2 (the first columns of the rotation matrix of the camera) are unit vectors and orthogonal to each other, r1*r2=0 and r1*r1=1=r2*r2, so we get the next equation:

$$\mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_2 = 0$$

That is by definition the $h_1^t \, w \, h_2 = 0$ IAC , with that we arrive to the solution w=K*K^-1.

Taking in account the previous explanation the IAC(w) is computed by solving the next equation A*w=0, been A the matrix containing 6 equations provided by the 3 homographies between the 3 imatges and the template. We need 3 imatges because we want know 6 unknowns, Ω or w= [w_{11}, w_{12}, w_{13}, w_{22}, w_{23}, w_{33}].

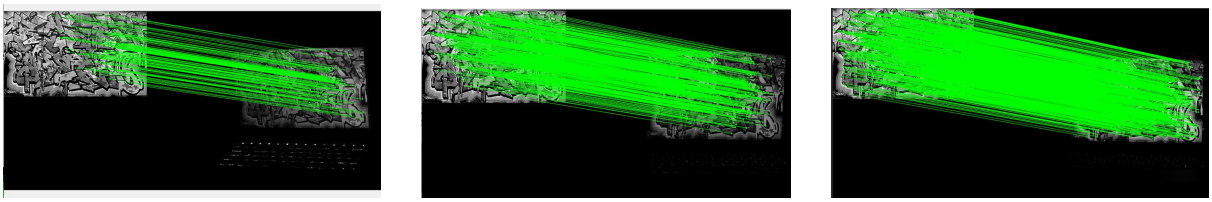

Image 21, 22, 23: SIFT matching in our 3 images to know the 6 unkows

Once A is defined, we can compute W using SVD, where the last column of V is the vector that contains Ω or w = [w_{11}, w_{12}, w_{13}, w_{22}, w_{23}, w_{33}], then we can compute the symmetric matrix W as :

$$\omega = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{pmatrix}$$

As we have seen w=K^t*K^-1, so we can use *Cholesky* to extract the camera calibration K, on the right we can see the resulting K, as expected the skew parameter is high because the photographies are a photo of a photo and regarding the size of the images the previous solution Ω it is up to scale so it can change.

```
K =

   1.0e+03 *

    3.3179   -0.0082    0.9684
         0    3.3172    0.5789
         0         0    0.0011
```

In the next step we were asked to compute the external parameters of the camera, r1, r2 and t, which can be express in the next formulation by the previous demo:

$$\mathbf{r}_1 = \frac{K^{-1}\mathbf{h}_1}{\|K^{-1}\mathbf{h}_1\|_2} \qquad \mathbf{r}_2 = \frac{K^{-1}\mathbf{h}_2}{\|K^{-1}\mathbf{h}_2\|_2} \qquad \mathbf{t} = \frac{K^{-1}\mathbf{h}_3}{\|K^{-1}\mathbf{h}_1\|_2}$$

Once we have the external parameters, we are able to extract the position of each of the camera by doing the each rotation and the translation to the optical center. In the next images we can see the position of each camera:

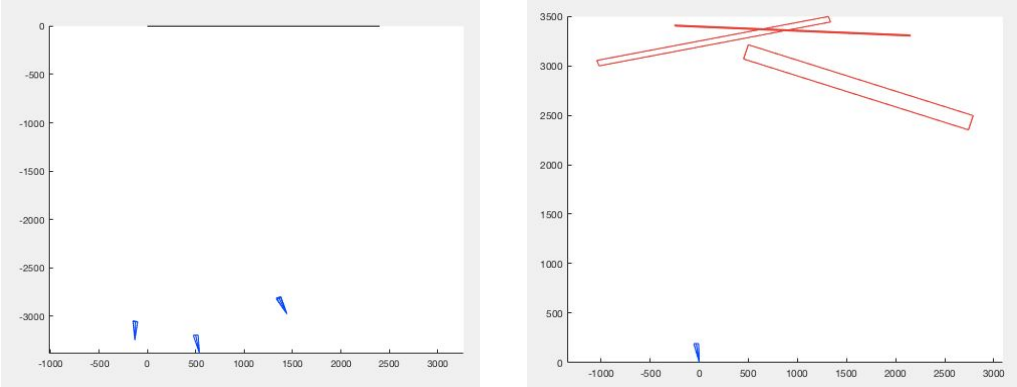

Image 24, 25: Comparison between static pattern moving camera and static camera moving the pattern

Finally, knowing the position of each camera we are to perform Augmented Reality by printing 3D points respect the actual position of the camera.

Image 26: Augmented Reality using calibration with planar pattern

6. OPTIONAL: Logo detection

The goal in this next step is use SIFT and the DLT algorithm to logo detection. The process to logo detection is very similar to the previous steps. First of all we have computed the SIFT descriptors for the desire logo which we want to find it and the same for the target image. Next, using a keypoint matching and computing the homography between them we could correlate and find the position of our logo in the target image as we can see in the next example:

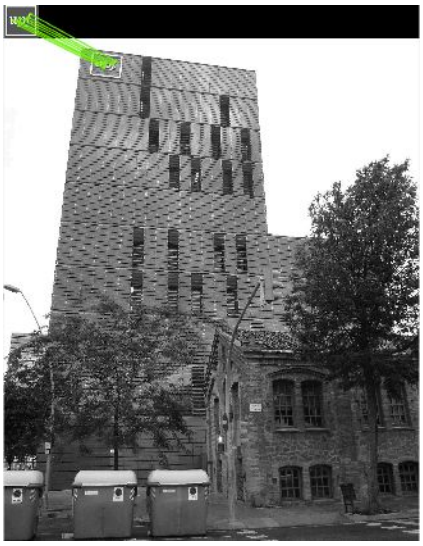

Image 27: SIFT keypoints matching between logo and target image

Finally, we have added a bounding box in the specific region as we can appreciate in the following images where we can see how our logo detection algorithm works :

Image 28, 29: Results of our logo detection algorithm

7. OPTIONAL: Logo replacement

The last optional task consists on an extension of the previous task, the logo detection. In this case we followed the same process as the logo detection adding an extra step to insert another logo in the same position where we find the original logo. To achieve our logo replacement we have used an homography to fit our new logo in the region that previously we have detected:

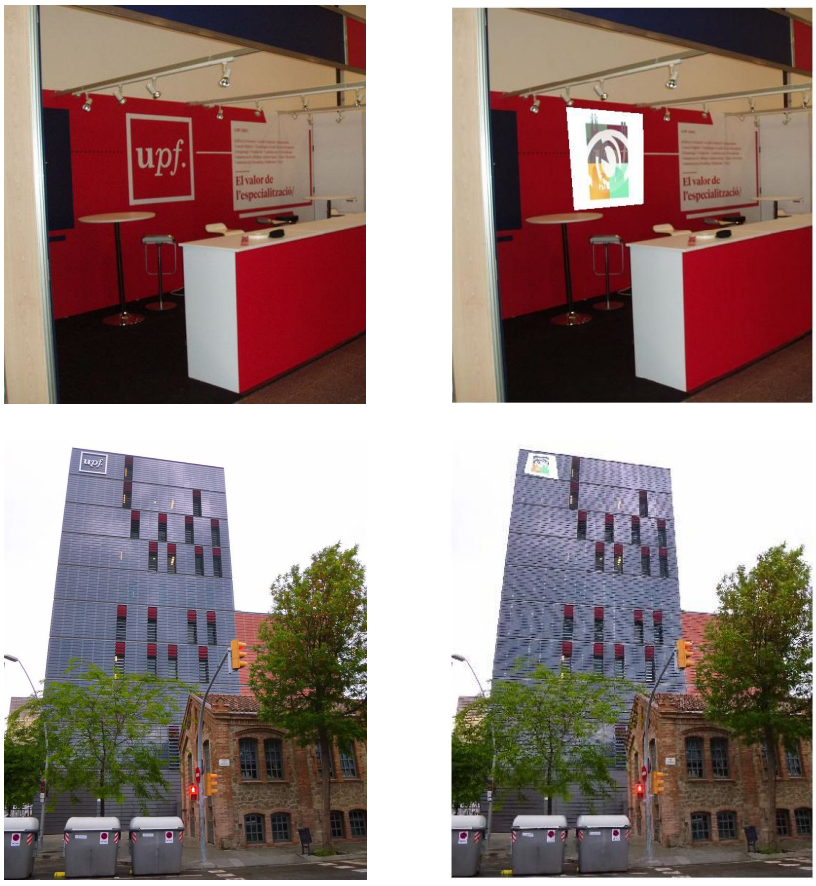

Image 30, 31, 32, 33: Examples of logo replacement, from UPF logo to CV master's logo before(left) and after (right)

Conclusions:

· Building panoramas with Robust normalized DLT algorithm (algebraic method) and even after refining it with the Gold-Standard algorithm (geometric method) is not always successful, as the issue of the 3D translation into the 2D world cannot always produce an adequate keypoint matching as we require homographies that properly relate the image pairs. So, as seen in the lectures, in order for an homography to relate two images, we need:

- · Images of the same plane in the 3D scene
- · Images taken with a camera rotating about its centre
- · Images taken with the same static camera varying its focal length
- · Or the whole scene is far away from the camera.

which was not always the case in some of the images we were given, which translated in visually unrealistic results of the panoramas.

· However, in all the evaluated cases, using the Gold-Standard algorithm always showed to reduce the reprojection error, even though the issue commented above was not solvable.

· Using a planar pattern to camera calibration (exercice 5) we can extract the camera parameters. This method can be useful for different applications such as lens distortion correction, measure the size of an object in world units or in our case for augmented reality. Also, this problem can be solved in another way, by using less images, assuming then restrictions and treating the problem as an minimization (Zhang's method).

· In the last optionals (logo exercises 6, 7), we saw how the Robust normalized DLT algorithm can have lots of different uses with successful results besides image mosaicing such as detection or replacement of image parts (logos, in our specific case) using the techniques and concepts learned in the main exercises.