

# M6 3D RECOVERY OF URBAN SCENES - SESSION 2

AXEL BARROSO & SERGIO CASTRO & SERGIO SANCHO

## GOAL

The goal of this session is to learn and compute homographies between different images of the same scene, being able to make mosaics out of them. For doing so, We implement the DLT algorithm and Gold Standart algorithm. In addition, as a optional part, we were asked to perform camera calibration and add a logos to an images.

## 1 INTRODUCTION

The homography allows us to relate two images (or more) that are:

- In the same plane in a 3D scene
- Taken with a camera rotating about its center
- Taken with the same static camera varying the focal length
- Far away from the camera

In order to apply the homography to the three different images of the same picture, we will use different algorithms:

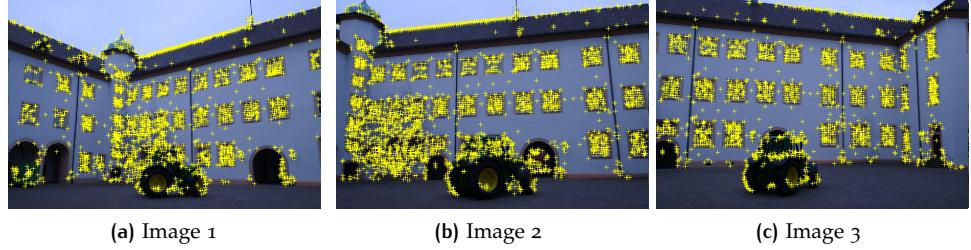
- Normalized DLT Algorithm
- Gold-Standard Algorithm
- Camera Calibration

These algorithms are applied in the computation of mosaics, augmented reality or logo insertions in images.

## 2 DLT ALGORITHM

### 2.1 Explanation

In order to compute the DLT Algorithm, we must find the different keypoints in the images that are related. This Keypoint detection is made with



**Figure 1:** Keypoints Detected

the SIFT Detector/Descriptor provided by the lecturers. In the figure 1 we can see the different keypoints found at each image of the same picture.

In the DLT Algorithm we will be able to compute the homography that relates the keypoints found with the SIFT detector.

The main steps of the DLT algorithm are:

1. Normalize the points of the first image
2. Normalize the points of the 2nd image
3. Apply the DLT Algorithm
  - a) Choose n equal/more to 4 correspondences to estimate H.
  - b) Obtain h as the last column of matrix V, obtained as the SVD of A.
4. Denormalize

The formulas of the previous mention steps are found in the figure 2.

$$\begin{aligned}
 1. \quad & \mathbf{x}_i^n = T \cdot \mathbf{x}_i \\
 2. \quad & \mathbf{x}'^n_i = T' \cdot \mathbf{x}'_i \\
 3. \quad & A_i \cdot \mathbf{h}^n = 0 \\
 & A_i = \begin{bmatrix} \mathbf{0} & -w'_i \cdot \mathbf{x}_i^T & y'_i \cdot \mathbf{x}_i^T \\ w'_i \cdot \mathbf{x}_i^T & \mathbf{0} & -x'_i \cdot \mathbf{x}_i^T \end{bmatrix} \\
 & A = (A_1, A_2, A_3, A_4) \\
 4. \quad & H = (T')^{-1} \cdot H^n \cdot T
 \end{aligned}$$

**Figure 2:** Steps of the DLT Algorithm

Along with the DLT algorithm, it is necessary to play with the different values of the Threshold set in the matlab file *homography2d.m*, as an incorrect Threshold would derive in a bad correspondence of the keypoints.

An example of the correspondence of keypoints between the previous images of figure 1 are found in figure 3

## 2.2 RANSAC

In order to compute correctly the mosaic of the images as demanded, it is necessary to apply the RANSAC function that will:

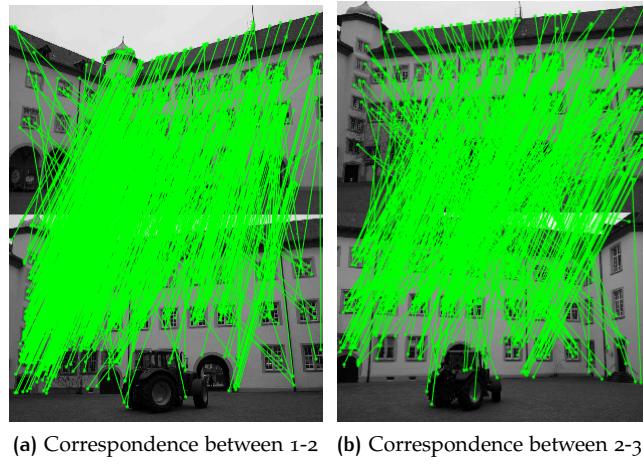


Figure 3: Correspondences between KeyPoints

1. For each sample  $N$ , compute the 4 random correspondences and its fundamental matrix  $H$  with the DLT algorithm.
2. Calculate the distance  $D_1$  for each correspondence
3. Compute the number of inliers in  $H$  where the distance  $D_1$  is lower than a threshold  $T$
4. Repeat and choose the matrix  $H$  that has the maximum number of inliers.

Thanks to RANSAC, we can pass from the correspondances of figure 3 to the ones showed in figure 4 that are clearly better than the previous obtained.

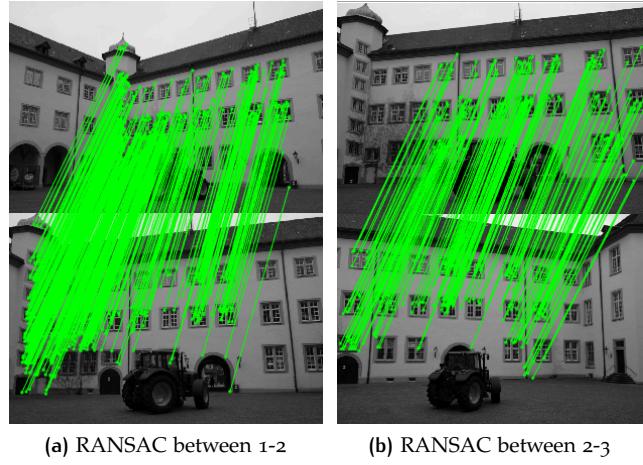


Figure 4: Correspondences after RANSAC

### 2.3 Results

In order to compute correctly the mosaic, we need to fix one of the three images as the center Image. This Image won't receive any Homography as

it is our reference Image. The other images of the same scene will be related to this Reference Image with the Homography Matrix  $H$  (mentioned before). In every threesome of images that we have, the reference image has been set to the 2nd Image.

### 2.3.1 Castle Images

In the figure 5 we can see the final result obtained in the Castle sequence. As we can see, the building has been correctly reconstructed, but the Truck that can be seen in the sequence at figure 6 is not correctly reconstructed.

This can be due to the inability of the SIFT Detector to compute correctly the keypoints of the green truck. This can be seen in figure 4 shown before. It seems that the algorithm is not robust to the different views of the position of the truck that can be found along the Castle sequence. One reason for this bad-reconstruction could be that relative movement of the camera with respect the Castle, where in the figure 6 can be noticed that not only the camera has rotated with respect its center point, but had also a translation.



**Figure 5:** Mosaic of the Castle Sequence



**Figure 6:** Castle Sequence

### 2.3.2 Llanes Sequence

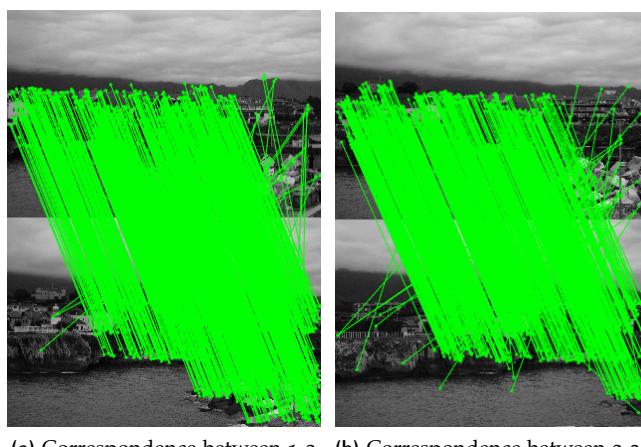
In this sequence we are making a mosaic out a landscape that can be seen in figure 7. It should be mentioned that in this case, it seems that the camera has remained static and has just rotated with respect its center point.

In figure , the different correspondences can be seen before the RANSAC algorithm has settled the optimum ones (with respect the threshold). It can



Figure 7: Llanes Sequence

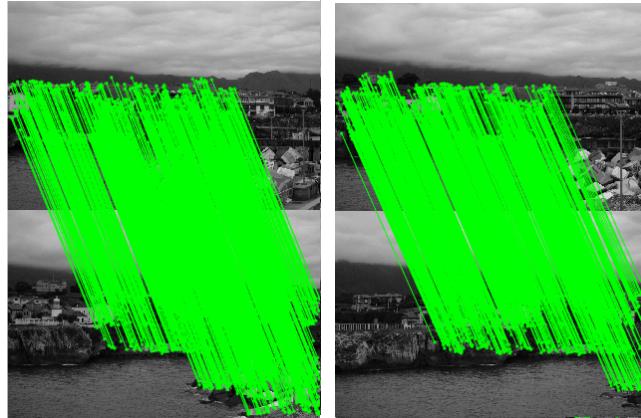
be seen that the majority of them have already a fixed orientation.



(a) Correspondence between 1-2 (b) Correspondence between 2-3

Figure 8: Correspondences between KeyPoints in Llanes

Comparing the figure 8 with figure 9 , it can be seen that in this case RANSAC has not delete many correspondences as almost all of them were inliers.



(a) Correspondence between 1-2 (b) Correspondence between 2-3

Figure 9: Correspondences between KeyPoints in Llanes after RANSAC

Finally, in figure 10 we can see the resulting Mosaic. As can be seen, in this sequence the reconstructions has been perfectly done. It appears

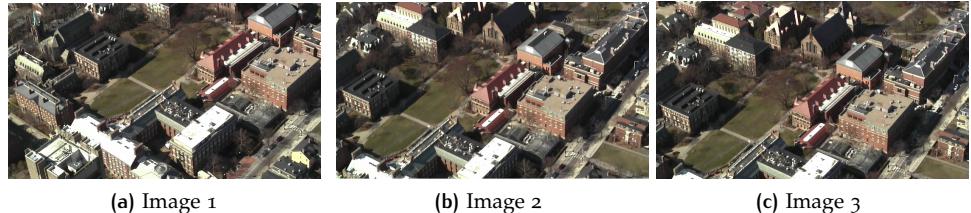
that the fact that the camera has only rotated with respect its center has given a possibility for the algorithm to reconstruct it better, as the previous Castle Sequence had also a translation that aggregated more inefficacy. The more similar point of view between the different images of the sequence has allowed us to reconstruct the mosaic better than the previous example.



**Figure 10:** Mosaic of the Llanes Sequence

### 2.3.3 Site13 Sequence

In this Sequence we are Dealing with an aerial view of a city. The different images of the sequence can be seen in the figure 11.



**Figure 11:** Site13 Sequence

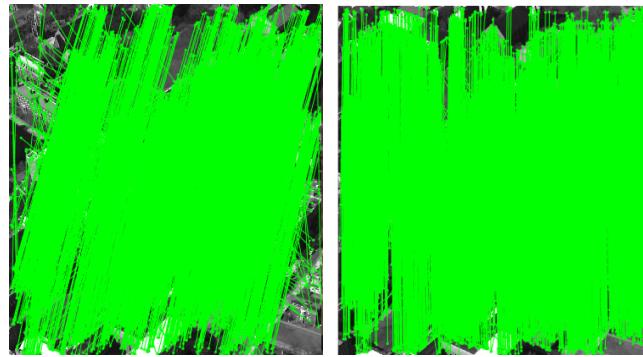
In this case, it seems that the camera is having a little translation between the different Images of the sequence. In the figure 12 the different correspondences between the two images with respect the referral image can be seen.

In this case, it also seems that all the correspondences follow the same pattern and that RANSAC would not delete many of them. In figure 13 the correspondences after applying RANSAC can be seen.

Finally, in the mosaic of figure 14 we can see that the result is good but not as good as in the Llanes sequence. In this case we can find a good reconstruction but there is also a little bit of glitter in the image. This can be due to the translation of the camera, even if it was little. Although, the result obtained is a good result.

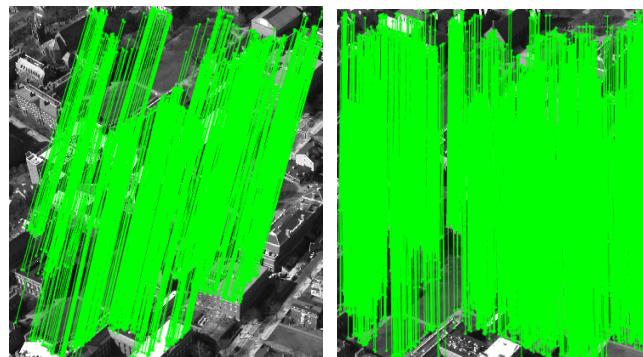
### 2.3.4 Site22 Sequence

As in the previous subsection, in this sequence we are dealing with an aerial view of a city but instead of being color images, they are gray-scale ones as can be seen in figure 15.



(a) Correspondence between 1-2 (b) Correspondence between 2-3

Figure 12: Correspondences between KeyPoints in Site13



(a) Correspondence between 1-2 (b) Correspondence between 2-3

Figure 13: Correspondences between KeyPoints in Site13 after RANSAC



Figure 14: Mosaic of the Site13 Sequence

As we can see in this sequence, we are having not only a translation but also a rotation of the view of the camera with respect its center.

Looking at the correspondences before and after RANSAC, we can conclude that in this case we are having a troublesome correspondence between the Image 1 and 2, while between 2 and 3 we can see that the correspon-

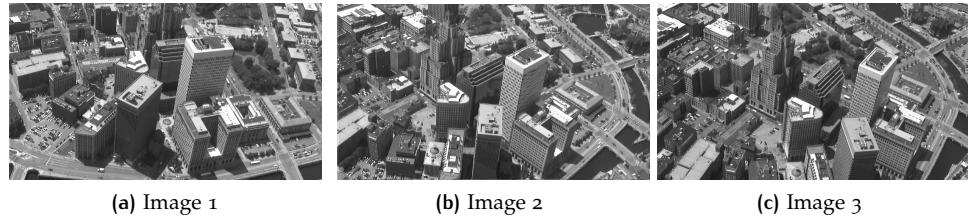


Figure 15: Site22 Sequence

dences are smoother and follow a similar pattern. This can be seen in image [16](#).

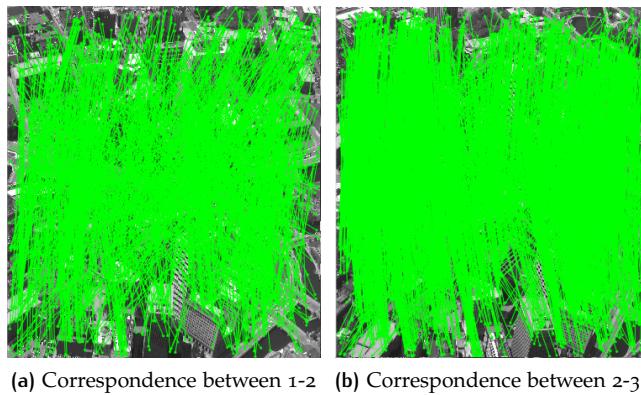


Figure 16: Correspondences between KeyPoints in Site22

As mentioned before, due to the chaotic correspondences between images [1-2](#), once we have applied RANSAC it can be seen that there are few Keypoints that can be settled in the same correspondence pattern. In the image [17](#) this phenomenon is shown.

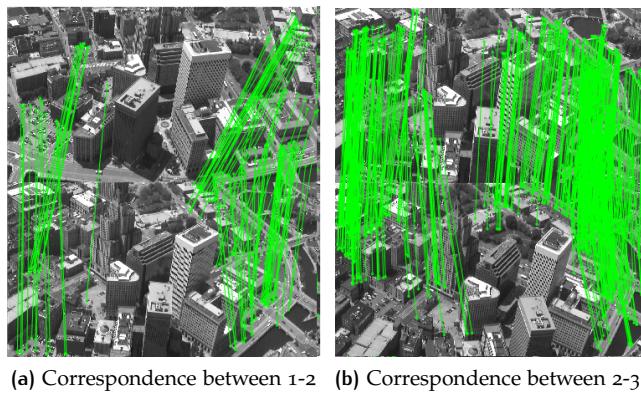


Figure 17: Correspondences between KeyPoints in Site22 after RANSAC

Finally, in the mosaic of figure [18](#) we can see that the result is a bad one. There have been not a real reconstruction of the image and it has no sense. This is due as the poor correspondence between keypoints as mentioned before, whose cause is the different points of view between the three images, that might be taken from different cameras and causing a totally different point of view.

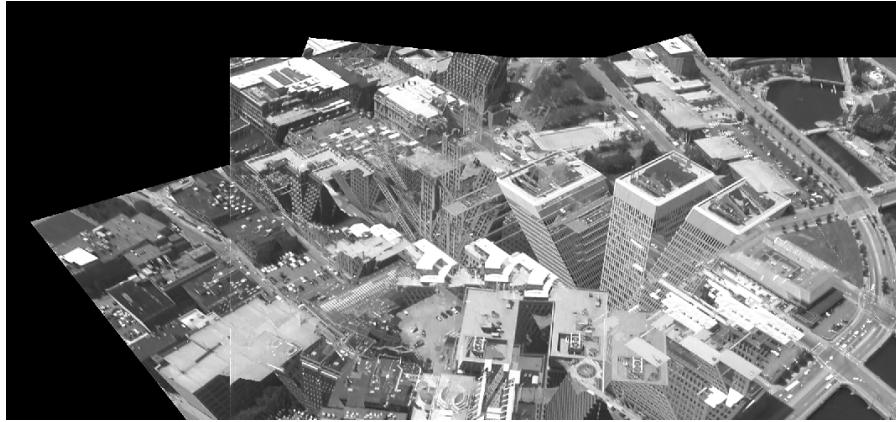


Figure 18: Mosaic of the Site22 Sequence

### 3 GOLD-STANDARD ALGORITHM

The Gold-Standard Algorithm is used in order to reduce the reprojection error, solving a non-constraint minimization problem as shown in image 19.

$$\min_{\hat{H}, \hat{x}_i} \sum_i d([x_i], [\hat{x}_i])^2 + d([x'_i], [\hat{H}\hat{x}'_i])^2$$

Figure 19: Minimization Problem

In the equation, we are reducing the error between  $\hat{x}$  and  $x'$ , which are the new pair of inlier points, with the minimization of least square algorithm.

In figure 20 and 21 we can see the different corrected Keypoints of the original images of the Castle. The images are shown in pairs between Image 1-2 and Image 2-3. In yellow there are marked the Keypoints found previously with the SIFT detector and in blue the Keypoints corrected with the Gold-Standard Algorithm.

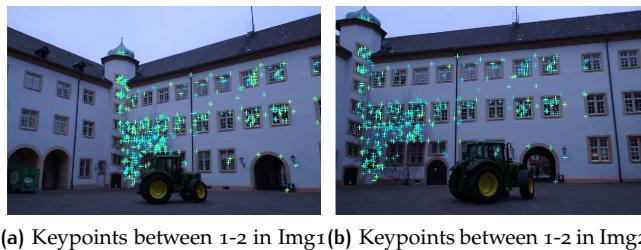


Figure 20: KeyPoints correction in Castle sequence after Gold-Standard

At the end the different mosaic reconstructions for each of the sequences are shown in figures 22, 23, 24 and 25 .

As can be seen, the results obtained are quite similar to the ones obtained in the previous mosaics with the DLT algorithm. So we can conclude that the optimization made with the Gold-Standard algorithm does not provide a better mosaic reconstruction compared with the DLT algorithm in the 4 sequences of images that we have.



(a) Keypoints between 2-3 in Img2 (b) Keypoints between 2-3 in Img3

Figure 21: KeyPoints correction in Castle sequence after Gold-Standard



Figure 22: Mosaic of the Castle Sequence with Gold-Standard Algorithm

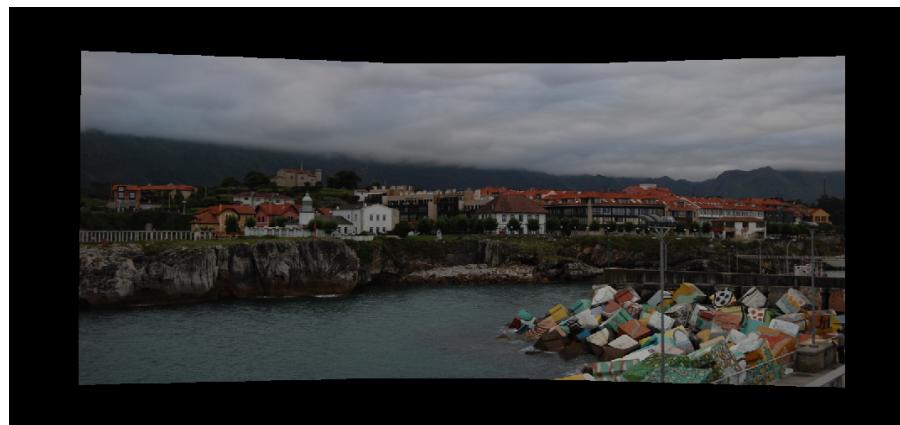


Figure 23: Mosaic of the Llanes Sequence with Gold-Standard Algorithm



Figure 24: Mosaic of the Site13 Sequence with Gold-Standard Algorithm



Figure 25: Mosaic of the Site22 Sequence with Gold-Standard Algorithm

## 4 OPTIONALS

### 4.1 Camera Calibration

In this optional task we have defined a Camera Calibration, whose steps are seen in figure 26. In this image there are shown not only the different steps of the Camera Calibration but also the definitions of the Matrix K and W.

$$\begin{aligned}
 P &= H = K \cdot [R|T] \\
 P &\sim K \cdot [r_1 \ r_2|T] \\
 r_1^T \cdot r_2 &= 0 \longrightarrow h_1^T \cdot K^{-T} \cdot K^{-1} \cdot h_2 = 0 \\
 r_1^T \cdot r_1 &= r_2^T \cdot r_2 = 1 \longrightarrow h_1^T \cdot K^{-T} \cdot K^{-1} \cdot h_1 = h_2^T \cdot K^{-T} \cdot K^{-1} \cdot h_2 \\
 w &= K^{-T} \cdot K^{-1} \quad \longrightarrow \quad w = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{12} & w_{22} & w_{23} \\ w_{13} & w_{23} & w_{33} \end{bmatrix}, \Omega = (w_{11}, w_{12}, w_{13}, w_{22}, w_{23}, w_{33}) \\
 r_1 &= \frac{K^{-1} \cdot h_1}{\| K^{-1} \cdot h_1 \|}; r_2 = \frac{K^{-1} \cdot h_2}{\| K^{-1} \cdot h_2 \|}; r_3 = r_1 \times r_2; t = \frac{K^{-1} \cdot h_3}{\| K^{-1} \cdot h_3 \|} \quad K = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

**Figure 26:** Camera Calibration

P is the projection Matrix, K are the intrinsic parameters of the camera and R and T the external ones. In order to approximate P as in figure , we must suppose that the planar plane Z=O.

Having in mind the orthogonality of the rotation matrix, we can arrive to the image of the absolute conic which is w, being w a symmetric matrix with 6 unknowns obtained by SVD. In order to obtain the parameters of K, we need to use the Cholesky decomposition. From the parameters of K (alphas), we can compute the real size of the image as they represent the focal length multiplied by the size of the pixel, and as we know the size of the image in pixels, we can compute the total size. Once we have K we can compute the rotational and translational parameters as shown in the last equation.

#### How is the optical center computed?

It is computed by the theoretical relation of  $t=-RC$ , where C is the center of the camera and t and R are the parameters computed in the last equation of figure 26.

The different views of the cameras with respect the planar pattern can be seen in figure 27 . Also, an example of how are the SIFT points computed is shown in figure 29.

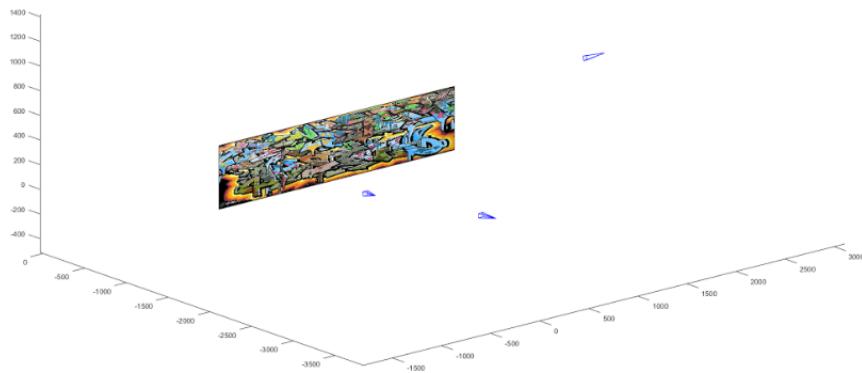


Figure 27: Different views from the camera

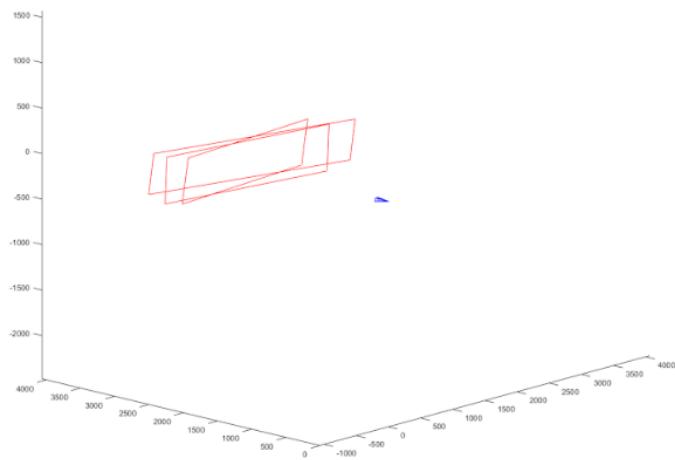


Figure 28: Different Projections

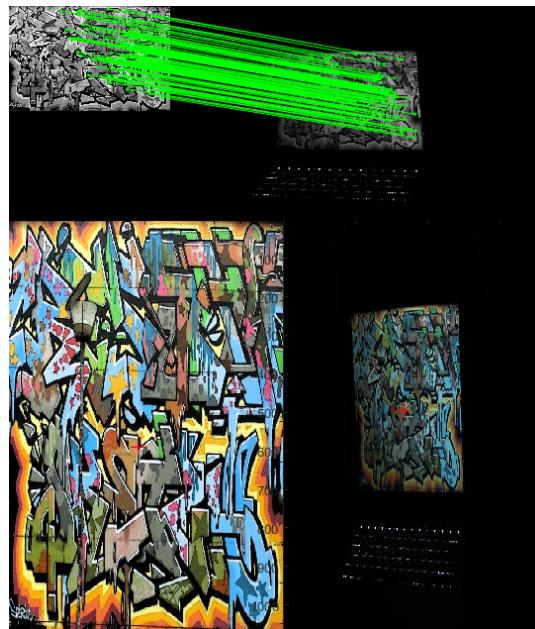


Figure 29: SIFT Points and Result

## 4.2 Augmented Reality

With camera calibration we can also include objects inside images in order to apply Augmented Reality.

In figure 30 it can be seen that we have implemented a pyramid inside the example image.

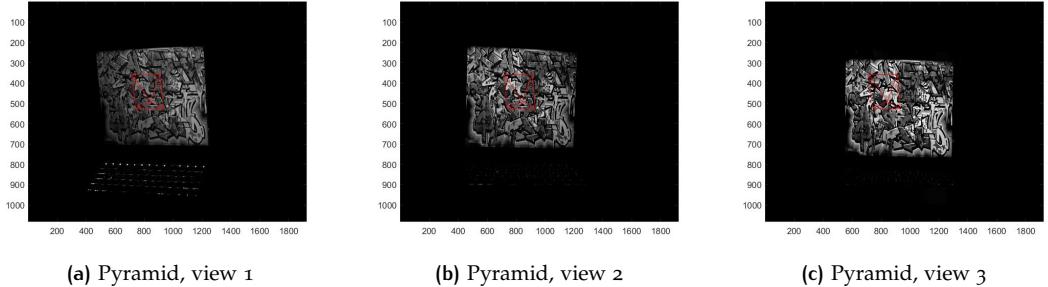


Figure 30: Pyramid Views

## 4.3 Adding a Logo

In this optional, we will include the image shown in figure 31 as a logo in the image of figure 32.



Figure 31: Image to be inserted as logo

In this case, we will select manually the different points of the image 32 that will contain the image 31. After, with homography we will convert the Logo image with a transformation in order to fit the characteristics of the receiving image.

The final Result that we have obtained is shown in figure 33.



Figure 32: Image receiver of the logo



Figure 33: Final Result

## REFERENCES

- [1] R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.