Module 4 - Report week 5

Hicham El Muhandiz, Julia Ariadna Blanco Arnaus, Marcos Muñoz González, and Rosana Valero Martínez

Universitat Autònoma de Barcelona, Campus UAB, Edifici O, 08193 Cerdanyola del Vallès, Barcelona

1 Introduction

In this project, a 3D reconstruction from 2D images was performed utilizing the COLMAP software and datasets that were both provided and acquired by our own means. A comprehensive analysis of the various steps involved in the reconstruction process was conducted, and various visualizations and metrics of the resulting reconstruction were evaluated.

2 3D mesh reconstruction from a set of images from the Gerrard Hall dataset

We can generate a sparse and dense reconstruction given a set of images within COLMAP. The sparse representation can be seen in Figure 1. The camera and point sizes have been adjusted to have a better visualisation. As we can see, some of the points in the representation are outliers, such as the matches that are on the sky. On the right, we can see the dense reconstruction. The

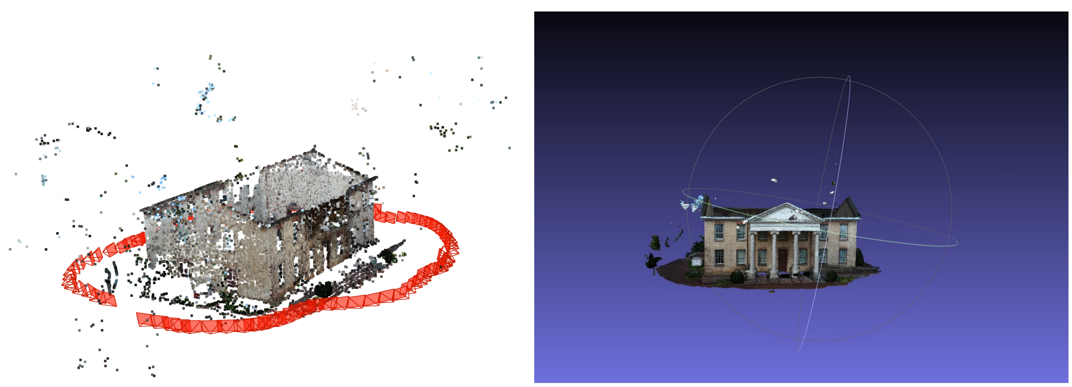


Fig. 1: Sparse representation visualised on COLMAP and dense representation visualised on Meshlab

outliers that we saw with the point representation are more visible on the mesh. In our case, we removed them on Meshlab by selecting all the facets with an edge threshold of 0.01 and removing them (Figure 2). However, this also removed regions that were not outliers, but rather partially reconstructed areas, since it is a threshold-based selection.

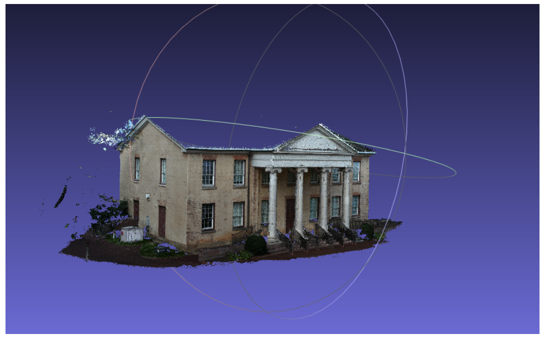


Fig. 2: Dense representation after removing the small facets

3 Implementation of the step-by-step 3D mesh reconstruction on the CASTLE dataset

3.1 Analyze using the Gerrard Hall reconstruction

Using the Gerrard Hall reconstruction, a point cloud was generated, which is visualized in Figure 3. A total of 3112449 keypoints were extracted from the database and used in the analysis. Of these keypoints, a total of 3243 three-dimensional (3D) points were obtained from a single keypoint present in the first image.

Fig. 3: Point cloud obtained from Gerrard Hall Reconstruction

3.2 Plot 3D points coloured according to the number of images and error

Both representations have been done with the information provided when loading the model. In particular, we have retrieved the point3D coordinates, the length of the amount of images for each point, and the respective error value. According to COLMAP documentation [4], the error value corresponds to the reprojection error in pixels of each point.
On the one hand, the edges of the building seem to be the points with the least amount of images used to generate them. This could be because those edges get occluded easily when moving around

the building. However, it could also be the fact that the inner parts of the facade have a brick, repetitive pattern. This might generate lots of matches between images that are taken nearby, as these patterns generate lots of edges, and, consequentially, key points that might get matched with key points of other bricks that are close-by. Similarly, the points on the sky also have a low count of images, which makes sense as they are outliers.
On the other hand, the error also increases in the inner regions of the facades (Figure 5). As mentioned earlier, the repetitive brick pattern might get easily mismatched by matching different bricks as the key points look really similar and are close to each other. In contrast, well-defined regions such as the edges of the windows have the lowest error. If we examine it closely (Figure 6), we can see that the error more inconsistent than the number of images, as it can vary greatly between close points.

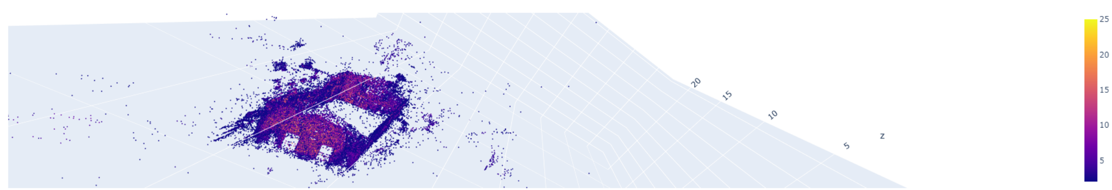


Fig. 4: 3d point representation coloured according to the number of images

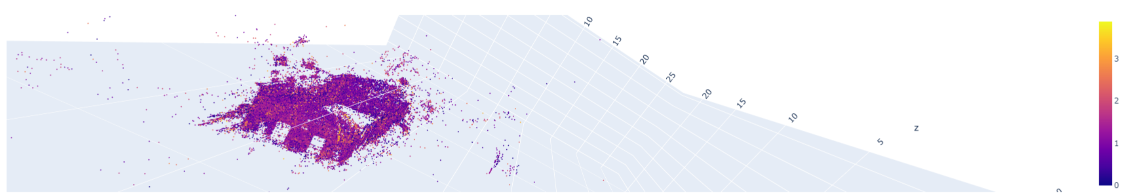


Fig. 5: 3d point representation coloured according to the error

3.3 Plot the 3D points that correspond to a key point in the first image

We can see the scatter plot of the 3D points that correspond to a key point in the first image and the key points in Figure 7. As expected, most of the points close to the facade that appears in the first image. The amount of points decreases as the distance from where the first image was taken from increases, as there will not be matches with images too far away and, consequentially,

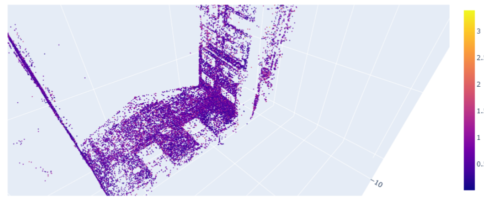
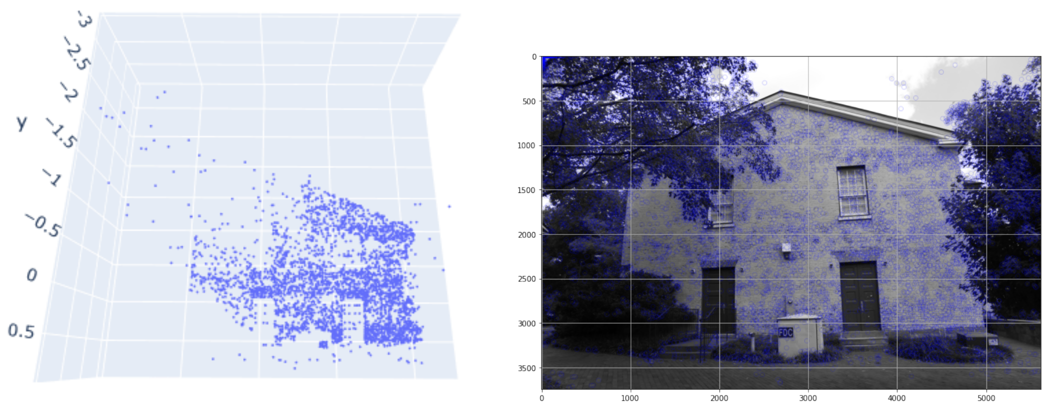


Fig. 6: Close-up of the 3d point representation coloured according to the error

3d points. In the image we can see that the key point outliers in places such as the clouds ended up creating 3d key points on the sky.



(a) (b)

Fig. 7: 3D point scatter plot generated using the first image (a) and key points (b)

3.4 Create a visualization for the number of matches between all images

The implementation of this section is a straightforward iterative approach where we go through the matches and increase a counter for each cell. The heat-map can be seen in Figure 8. As we can see, the matches between images that are close to each other sequentially, such as image 1 and image 2, have a higher amount of matches. This makes sense since the pictures of the database have been taken circumambulating the building in a sequential order. It also explains the fact that pictures close to the first one have matches with the ones taken last like the 100th one. Another observation we can extract from this result is that the database does not include redundant matches, in the sense that it will include the matches of the image 1 and image 2, but not the matches of image 2 and image 1, which would be the same.

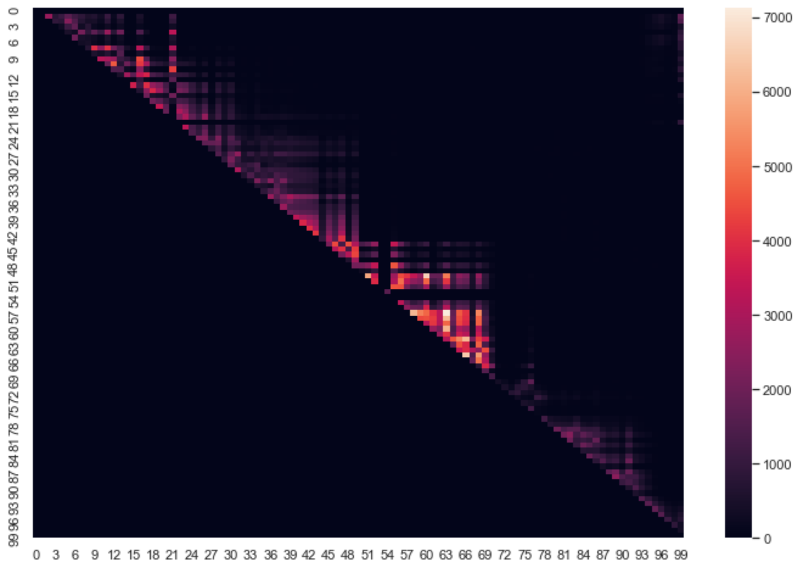


Fig. 8: Number of matches between images

3.5 Visualize the keypoints and matches between the two images used in lab 4 using COLMAP

On Figure 10 the keypoints obtained for each of the two images used in lab 4 are shown. These keypoints have been obtained with COLMAP. The number of keypoints obtained is larger with respect to the number of keypoints obtained in lab 4 due to the fact that COLMAP uses SIFT for feature extraction, while in the previous lab ORB was used. In Figure 10 the matches between the two images are shown.

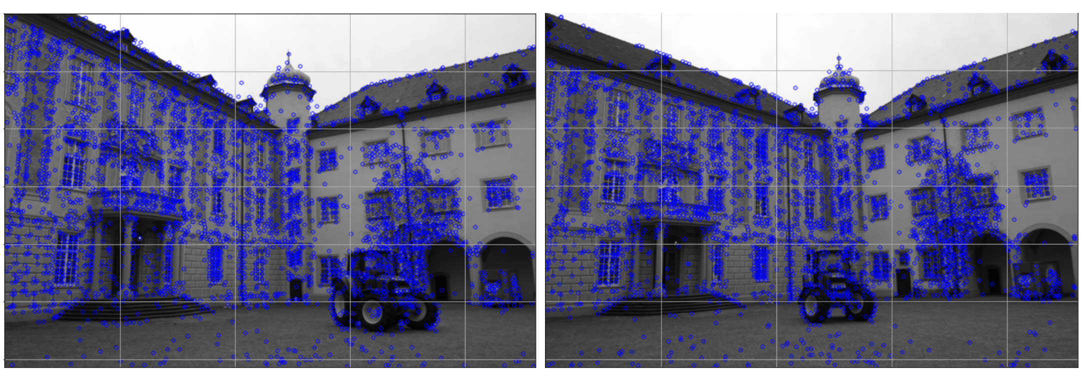


Fig. 9: Keypoints of the two images used in lab 4 obtained with COLMAP

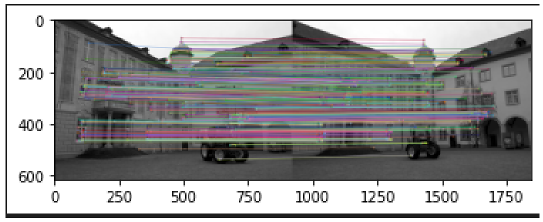


Fig. 10: Matches between the keypoints of the two images in lab 4 obtained with COLMAP

3.6 Triangulate and visualize the 3D points from the keypoints extracted using COLMAP on the two images used in lab 4

In this section, we use the triangulation with the DLT method that we have used in Lab4 to obtain the 3D points and compare it with the results we have obtained from Lab 4.

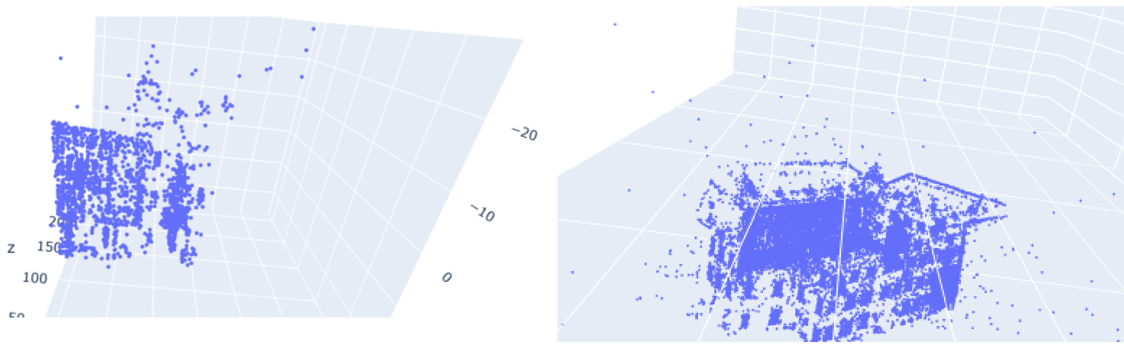
As mentioned in the previous section, the utilization of COLMAP in Lab 5 leads to an increase in the number of keypoints. As a consequence, it results in a rise in the quantity of 3D points, thus leading to a more precise reconstruction in comparison to the result from Lab 4, as we can observe in Figure 11. For instance, in Lab 5, we can begin to guess the shape of a tractor through the obtained 3D points.

(a) (b)

Fig. 11: Visualize the 3D points from the keypoints extracted using COLMAP for (a) Lab 4 and (b) Lab 5

3.7 Visualize the sparse reconstruction using the 2 images from lab 4, and the complete CASTLE dataset

As we can see in Figure 12, a low count of images gives a really limited view of the building. Using only two images allows us to get a good estimate of the 3d points of that region, but it is a rather limited representation compared to what we obtain using the 30 images set, as we have more information on the surroundings and not just one facet. Besides, it is likely that the 3d points of the later have less error, as more images have been used to generate the keypoints.



(a) (b)

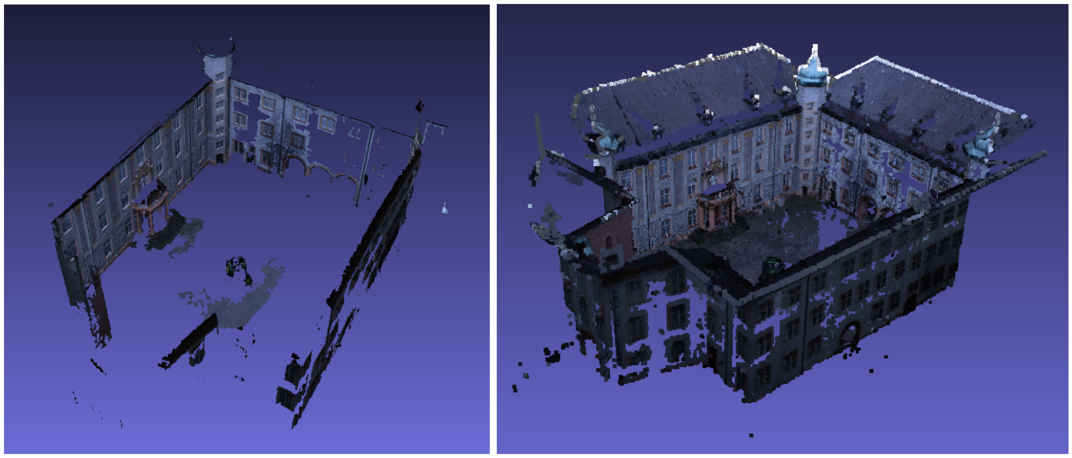
Fig. 12: COLMAP sparse representation using 2 (a) and 30 (b) images of the CASTLE dataset

4 Configure the reconstruction to improve the results

The changes were mostly in the key point detection and matching stage. In the key point detection we increased the amount of maximum features to have a denser key point representation. Furthermore, we checked the boxes for estimating affine shapes and domain size pooling. For the feature matching step we checked the box for guided matching, as, according to COLMAP documentation, it gives better results. The result with those parameters and the default one can be seen in Figure 13. Since we increased our amounts of keypoints, this led to an increase of 3d points, which helped generating a better mesh.

5 Reconstruct a 3D mesh from images captured by you

Now we try to reconstruct a 3D mesh from our own images. In order to take these images, it is not only important to consider the spatial location of them, but also to be sure that they come from the same environment and with similar circumstances on weather, camera and date among other things. In our case, we take the images with an iPhone 13 smartphone, and we do this on an environment free of people and only with what we want to convert into the 3D model (no random people or objects roaming). To make things interesting we decided to use a human model (one of the authors) and tried to convert it into a 3D reconstruction as we can see in Figure 14. This way we could explore the behavior of the 3D reconstruction on images with certain complexity (an human is not flat) and this could be used as a prop for other tasks (for instance a virtual world). Interestingly, some important characteristics of the image were preserved such as the textures (like on the backpack of Figure 15) or the shadows.



(a) (b)

Fig. 13: Mesh with the default parameters (a) and the updated ones (b)

(a) (b)

Fig. 14: On the left (a) our 3D reconstruction, on the right (b) some image of our dataset where the reconstruction was done

Fig. 15: The 3D reconstruction of the backpack of the 3D human. Notice how the textures are preserved

We also performed our own 3D reconstruction with the Computer Vision Center building images. We could see that the materials of the building and the text could be clearly reconstructed in Figure 16. Even though, some parts of the original images such as the tree were not reconstructed correctly.

6 Conclusions

In this work, we have generated sparse and dense reconstructions given a set of images. We have shown different ways to visualise the data and some of its short-comings.

Some problems that appeared during the implementation of the exposed techniques are:

- To create the Dense mesh COLMAP requires a CUDA version of it, else the .ply files are not generated when following the steps to perform an automatic reconstruction.

Overall this has been an interesting project, where we have seen how we can get mesh representations from point clouds generated from image correspondences. As a relevant bibliography, we have used the book Multiple View Geometry in Computer Vision by Richard Hartley and Andrew Zisserman [1].

7 Feedback on the lab

We had a lot of problems with the installation process. Despite our initial attempts, we couldn't succeed to install the python bindings that were required previously to the changes in the assignment. Fortunately, with the new version we were able to set all up and could proceed with the tasks.

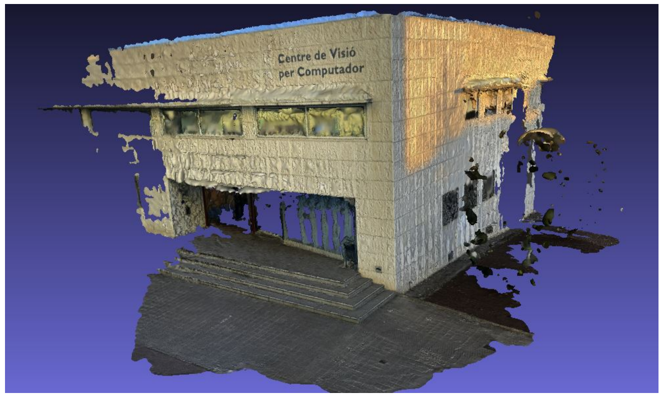


Fig. 16: The 3D reconstruction of the CVC building. Notice how the text can be clearly seen

On the other hand, we found some tasks of the lab a little bit confusing. However, we asked for a clearer explanation in some points and finally we were able to complete all the assignment. Overall, we consider that the points were distributed fairly and the difficulty level, although we found it a little bit challenging in the beginning, we consider it well balanced for this subject. Furthermore, it was interesting to use different kinds of software such as COLMAP and Meshlab, and they were helpful to understand the concepts of sparse and dense representations.

References

1. Hartley, R., Zisserman, A. (2004). Multiple View Geometry in Computer Vision (2nd ed.). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511811685
2. Stereo Matching, https://campar.in.tum.de/twiki/pub/Chair/TeachingWs11Cv2/3D_CV2_WS_2011_Stereo.pdf. Last accesed 02 Feb
3. MeshLab: Simple Mesh Editing, http://vcg.isti.cnr.it/corsi/G3D_InfoUma/Slides_2021/06_MESH_EDITING.pdf. Last accesed 06 Feb
4. Schoenberger, J.L. COLMAP, https://COLMAP.github.io/index.html