REPORT LAB 2:
Structure From Motion

Alberto Zerbinati	alberto.zerbinati@studenti.unipd.it	N.Matricola:	2053861
Davide Roana	davide.roana@studenti.unipd.it	N.Matricola:	2058341

1. INTRODUCTION

This assignment regards structure from motion estimation, that is the process of estimating a 3D structure of a scene from a sequence of images with some field-of-view overlaps. With this work, we explored Homography and Essential Matrix estimation, plus we learned to use Ceres Solver to solve Bundle Adjustment iterations. We then tested the implementation on some datasets.

2. IMPLEMENTATION

In the provided code, left to complete were the following functions:

For the file **features_matcher.cpp**:

A. Salient point extraction (extractFeatures())
B. Descriptors matching (exhaustiveMatching())

For the file **basic_sfm.cpp**:

C. Rigid body transformation estimation and setting (solve())
D. Triangulation of 3D points (solve())
E. Auto-differentiable cost function for Ceres Solver (ReprojectionError())
F. Residual Block addition (solve())

A. Salient Point Extraction (done by Davide)

The implementation of the salient point extraction was pretty straightforward, just using the detector to extract keypoints and descriptors, and storing them in the provided vectors. Using the 2D point associated with each keypoint it was then possible to obtain the color information from the image.

B. Descriptors Matching (done by Davide)

We used a Brute-Force Matcher with Hamming Norm and then, through computation of both Homography and Essential Matrix using the RANSAC algorithm, we computed the inlier points.

We set the threshold to 5 and, by checking the inliers size for both matrices, we kept the best one in terms of the number of inliers. If the threshold wasn't surpassed by either of the two matrices, we left the `inlier_matches` empty.

C. Rigid Body Transformation Setting (done by Alberto and Davide)

In this function, through computation of both the Homography and Essential matrices (as before, using the RANSAC algorithm), we compared the number of inliers. If the number of inliers for the Essential Matrix was higher than the one for the Homography matrix we recovered the pose from the corresponding matrix.

This was set as optimal if the transformation was mainly given by a sideward motion instead of a forward one (by checking that the translation on the x-axis was greater than on the z-axis). When setting the transformation we kept only the inliers points for later computation.

D. Triangulation of 3D points (done by Alberto and Davide)

In this function, we used the provided Camera Blocks to obtain the needed projection matrices thanks to the Rodrigues Formula. We then triangulated the Points and checked the cheirality constraint (meaning the point layed in front of the image plane). If the cheirality constraint was satisfied, we added the 3D point to the optimization vector.

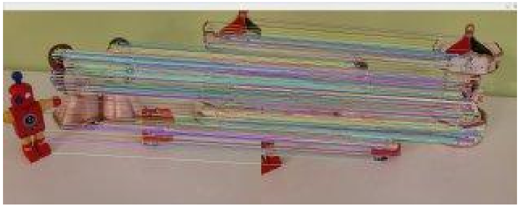
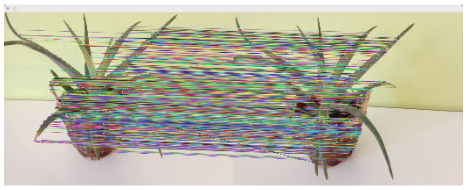
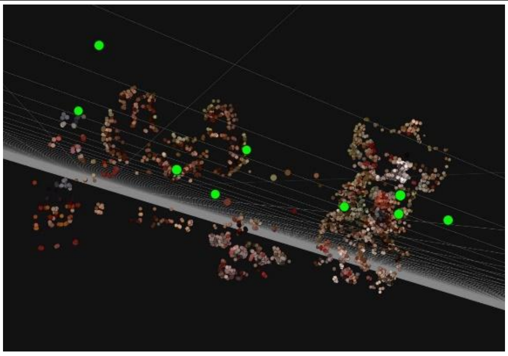
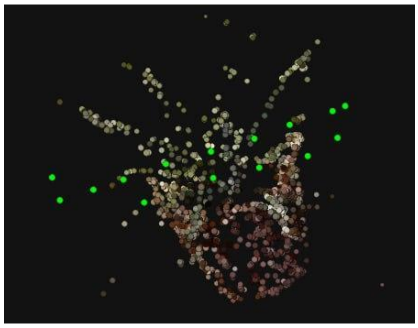
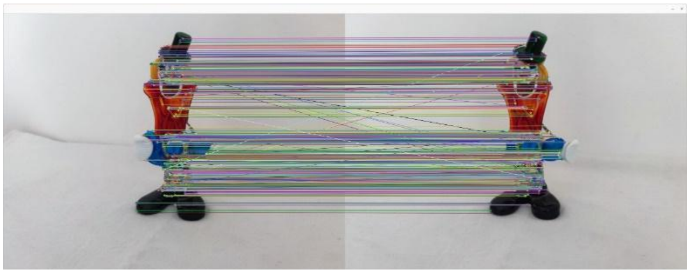
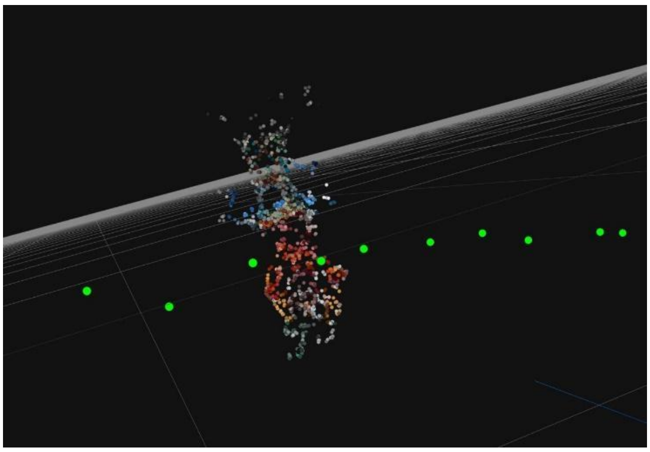
E. Auto-differentiable cost function (done by Alberto)

The implementation of the cost function was done by setting the residuals as equal to the difference between the predicted point (obtained considering the camera and the associated transformation) and the observed one.

F. Residual Block Addition (done by Alberto)

The residual block is added to the optimization problem by using the Cauchy Loss function (to better deal with potential outliers) and providing the needed camera and point parameters.

3. RESULTS AND COMMENTS

<table>
<tr><td>DATASET</td><td>ONE</td><td>TWO</td></tr>
<tr><td>FEATURE MATCHING (example)</td><td></td><td></td></tr>
<tr><td>POINT CLOUD</td><td></td><td></td></tr>
<tr><td>DATASET</td><td colspan="2">PERSONAL</td></tr>
<tr><td>FEATURE MATCHING (example)</td><td colspan="2"></td></tr>
<tr><td>POINT CLOUD</td><td colspan="2"></td></tr>
</table>

NOTES:

To obtain the results depicted in the table some tuning was needed. In fact, we used for the Feature Matching part a threshold of 2.0 for RANSAC.

Other thresholds made different issues arise, such as having No Seed Pair Found error when running the SFM part or generating wrong point clouds.

Due to the stochastic nature of the algorithm used by Ceres Solver, we also needed some runs before obtaining the desired results, as different runs with the same feature points led to different results.

To obtain good results with our own dataset we needed to capture different scenes and try different combinations of images, but in the end, the result was representative of what was pictured.

N.B.: *the images in the table can be found in the folder named "report_imgs" and the final results can be found in the "results" folder.*