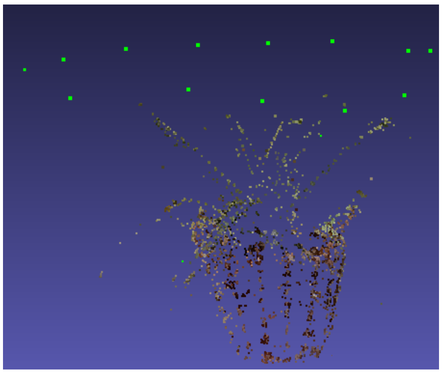
*3D DATA PROCESSING - LAB 2 (**GROUP ASSIGNMENT**)*



Topic: Structure from Motion
Goal: Estimate the 3D structure of a small scene taken by your smartphone from a sequence of images with some field of view overlaps.
Groups: Students should organize in groups of two and do the assignment together. Only one submission per group is sufficient (just clearly indicate the names and the “matricola” of the components in the report)

Introduction

For this assignment, it is required to complete some code portions of a simple Structure from Motion pipeline. Such pipeline includes two main modules:

- A feature extractor and matcher module, by means of the `matcher` app defined in `matcher_app.cpp`. This application requires as input the calibration parameters of the camera used to acquire the images, a folder containing the images used for the 3D reconstruction, and a file to store the data that will be used in the subsequent reconstruction module.
- An incremental structure from motion module, by means of the `basic_sfm` app defined in `sfm_sfm.cpp` that requires as input the dataset produced by the matcher app and a `.ply` file to save the 3D reconstruction

Groups are required to complete two chunks of code within the source file `features_matcher.cpp`, namely:

1) Extract salient points, descriptors, and features colors from images;
2) Match descriptors between image performing Essential + Homography geometric verification.

and three chunks of code within the source file `basic_sfm.cpp`, namely:

1) Extract both Essential matrix E and Homograph matrix H and check the number of inliers for both models to extract the seed pair. Recover from the selected pair the initial rigid body transformation;
2) Add a residual block inside a Ceres Solver problem;
3) Implement an auto-differentiable cost function for the Ceres Solver problem.

Detailed descriptions of what is required is reported directly inside the code. Chunks of code to be completed are marked with the preamble:

```
/////////////////// Code to be completed (x/y) /////////////////
```

followed by the description of the required functionalities.

Please check the code carefully, reading all the comments and trying to understand the flow.

What you need to deliver

- Source code (without objects and executables)
- A short written report with:
  - A brief description of the work done;
  - Some qualitative results on at least one of the two provided datasets and at least one dataset acquired from the group with a smartphone.
- Your (small) dataset (Avoid high resolution images: rescale them)

The results can be shown through point-cloud screenshots (for example, you can use MeshLab to visualize the produced .ply file, see for instance the figure above): they have to be reasonably good.

HINTS

1) Open CV provides several utility functions for geometry and 3D reconstruction:

https://docs.opencv.org/4.2.0/d9/d0c/group__calib3d.html

Among others:

```
void cv::Rodrigues ()
```
convert to axis-angle rotation representation and vice-versa.

```
Mat cv::findEssentialMat()
Mat cv:: findHomography()
```
Estimate the Essential matrix and the Homography matrix from two sets of correspondences

```
int cv::recoverPose()
```
Recovers the relative camera rotation and the translation from an estimated essential matrix

2) Read carefully the Ceres Solver bundle adjustment tutorial:

http://ceres-solver.org/nnls_tutorial.html#bundle-adjustment