3D Reconstruction

Hala DJEGHIM and Sarah SOUDANI

08/01/2022

1 Introduction

In this report, we present our work for the 3D reconstruction project. We present the Structure from Motion pipeline and its results at each step, the Multi-view surface reconstruction result and finally a discussion and conclusion to review our work.

2 Structure from Motion

SfM is the process of reconstructing 3D structure from its projections into a series of images taken from different viewpoints. The input is a set of overlapping images of the same object, taken from different viewpoints. The output is a 3-D reconstruction of the object, and the reconstructed intrinsic and extrinsic camera parameters of all images. [COL]

The SfM workflow is composed of 3 stages :

- Feature extraction
- Pose estimation
- 3D Triangulation

To solve our 3D Reconstruction problem, we choose to follow an Incremental SfM pipeline, which consists of two main phases, the first one is correspondences search between images, while the second one consists of iterative incremental reconstruction. The correspondence search phase is composed of three sequential steps: Feature Extraction, Feature Matching and Geometric Verification.

The iterative reconstruction phase is composed of an initialization step followed by three reconstruction steps: Image Registration, Triangulation and Bundle Adjustment.

Many implementation of the SfM pipeline are proposed, we can mention COLMAP, Theia, OpenMGV, VisualDFM, Bundler and MVE.
We choose to work with COLMAP [COL], since it is the one that gave the best average results in the performance evaluation presented in [BCM18]

3 SfM pipeline

3.1 Image capturing

We captured 46 images of a wood, in several angles. We tried to choose images with good texture, similar illumination conditions, high visual overlap and with different viewpoints.

3.2 Correspondences search

3.2.1 Feature extraction

A feature provides the meaningful information without the whole pixel. For each image Ii , SfM detects sets Fi = (xj ,fj) — j = 1...N of local features at location xj, represented by an appearance descriptor fj.

The feature extractor used in COLMAP is SIFT : Scale-invariant feature transform. [Lin12]

Major advantages of SIFT are [Lin12] :

- Features are local, it is robust to occlusion and clutter.
- Individual features can be matched to a large database of objects.
- Many features can be generated for small objects.
- Close to real time performance.

And the SIFT [Lin12] main step :

- Scale-space peak selection : potential location for finding features by using scale-space, blurring, Diffrence of Gaussian kernel and Laplacian of gaussian approximations.
- Keypoint localization : Accurately locating the feature keypoints, using Taylor series expansion.
- Orientation assignment : Assigning orientation to keypoints.
- Keypoint descriptor : Describing the keypoints as a high dimensional vector.

The features extracted can be visualized as shown in the figure 1

Figure 1: Features extracted

When processing every image, COLMAP gives the Name of the image, its dimensions, the camera model, the focal length, GPS measures and the number of features detected. 2

```
Processed file [1/51]
  Name:              1.jpg
  Dimensions:        4160 x 2080
  Camera:            #1 - SIMPLE_RADIAL
  Focal Length:      4992.00px
  GPS:               LAT=48.852, LON=2.517, ALT=2.000
  Features:          10629
```

Figure 2: Features output

3.2.2 Features matching

The goal of this step is to search for features correspondences by finding the most similar feature in an image Ia for every feature in an image Ib, using a similarity metric comparing the appearance of the feature. The output of this step is a set of potentially overlapping images. COLMAP allows us to choose between many matching modes : Exhaustive Matching, Sequential Matching, Vocabulary Tree Matching, Spatial Matching, Transitive Matching and Custom Matching. We choose the Exhaustive matching mode since the number of images of our dataset are relatively low, so every image can be matched against every other image.

3.2.3 Geometric verification

This stage verifies the potentially overlapping image pairs. Since the previous phase is based on the appearance, so the algorithm here tries to estimate a transformation that maps feature points between images using projective geometry. COLMAP uses Hamography, Relative Pose and F-matrix. The output of this stage is a set geometrically verified image pairs.

In COLMAP those two steps are combined in a single command called "Feature matching", here is a pair of a matched images example 3

Figure 3: Features matching

3.3 Incremental reconstruction

3.3.1 Initialization

The reconstruction starts from an initial image pair. To obtain a good reconstruction it is preferable to start from a dense region of the scene graph so that the redundancy of the correspondences provides a solid base for the reconstruction.

Figure 4: Image initialization

3.3.2 Image registration

The goal of this step is to solve the Perspective-n-Point problem, and to estimate the pose, and intrinsic parameters if the camera is uncalibrated.
The Perspective-n-Point is the problem of estimating the pose of a calibrated camera given a set of n 3D points in the world and their corresponding 2D projections in the image. In our case, we didn't calibrate the camera before starting our reconstruction process. COLMAP uses P3P the perspective-three-point problem [GHTC03] and EPnP An accurate o (n) solution to the pnp problem [LMNF08], as algorithms. [BCM18]

3.4 Triangulation

A newly registered image may observe new points as well as existing reconstructed 3D points. Such new points can be added to the 3D reconstruction through triangulation if they are observed by at least one previously registered image. This process makes the 3D point cloud more dense, and increases the stability of the model through redundancy. COLMAP uses sampling-based DLT triangulation. [BCM18]

3.4.1 Bundle Adjustment

Bundle adjustment is the joint non-linear refinement of camera parameters and point parameters that minimizes the reprojection error. It can be executed locally or globally on all images. It is executed globally on all images only when the rebuilt point cloud has grown by at least a certain percentage since the last time global BA was made. The software that we used adopts Multicore BA and Ceres Solver. [BCM18]

3.4.2 Results

The previous steps are combined in one command, "start reconstruction". At each iteration, we have Pose refinement report, Bundle Adjustment report, Retriangulation result and Global bundle adjustment. We could also visualize the images in real-time. 6

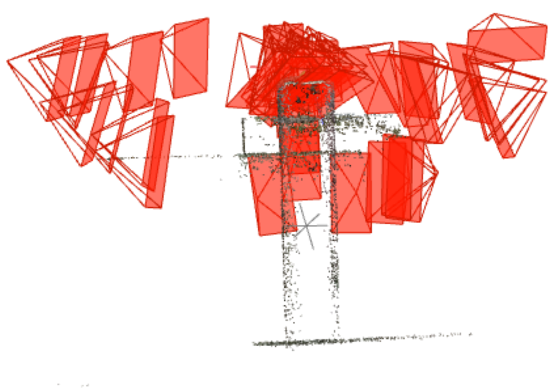
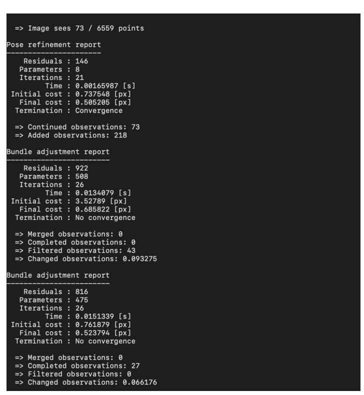


Figure 5: Pointscloud



```
 => Image sees 73 / 6559 points

Pose refinement report
----------------------
    Residuals : 146
   Parameters : 8
   Iterations : 21
         Time : 0.00165987 [s]
 Initial cost : 0.737548 [px]
   Final cost : 0.505205 [px]
  Termination : Convergence

  => Continued observations: 73
  => Added observations: 218

Bundle adjustment report
------------------------
    Residuals : 922
   Parameters : 508
   Iterations : 26
         Time : 0.0134079 [s]
 Initial cost : 3.52789 [px]
   Final cost : 0.685822 [px]
  Termination : No convergence

  => Merged observations: 0
  => Completed observations: 0
  => Filtered observations: 43
  => Changed observations: 0.093275

Bundle adjustment report
------------------------
    Residuals : 816
   Parameters : 475
   Iterations : 26
         Time : 0.0151339 [s]
 Initial cost : 0.761879 [px]
   Final cost : 0.523794 [px]
  Termination : No convergence

  => Merged observations: 0
  => Completed observations: 27
  => Filtered observations: 0
  => Changed observations: 0.066176
```

Figure 6: Reconstruction steps

COLMAP exports three text files for reconstructed model : [COL]

- **cameras.txt :** contains the intrinsic parameters, cameraID, Model, Width, Height, and 3 parameters : the focal length and principal point location. We choose the radial mode since we didn't calibrate our camera.
- **images.txt :** contains the pose and keypoints of all reconstructed images in the dataset using two lines per image. The reconstructed pose of an image is specified as the projection from world to the camera coordinate system of an image using a quaternion (QW, QX, QY, QZ) and a translation vector (TX, TY, TZ)
- **points3D.txt :** contains the information of all reconstructed 3D points in the dataset using one line per point.

4 Processing and editing the sparse pointclouds

After our SfM pipeline we used MashLab to editing, cleaning, rendering, texturing and converting our result. We obtained the result shown in 7

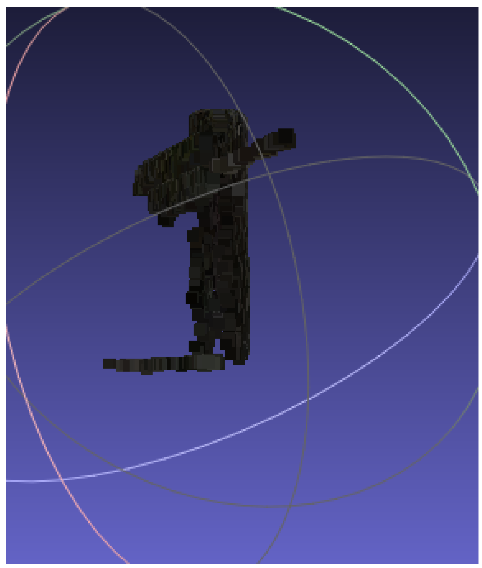


Figure 7: Processing and cleaning result

5 Multi-view surface reconstruction

Multi-View Stereo (MVS) takes the output of SfM to compute depth and/or normal information for every pixel in an image. Fusion of the depth and normal maps of multiple images in 3D then produces a dense point cloud of the scene. Since we don't have any GPUs on our systems, COLMAP dosen't allow us to use its dense reconstruction tool. As an alternative we used AgiSoft software, the free version, that helped us have the result shown, but didn't give us many possibilities. The result is shown in 8

Figure 8: Dense reconstruction result

6 Discussion, experiments and Conclusion

Before choosing our dataset, we experimented many datasets, and we conclueded that the texture, background and illumunation count.

6.1 Robot dataset

We took several images of a white robot with a white background, the pointclouds reconstructed were empty in the inside. And that was caused by the choice of the background and the object color.
As shown in 10 the features extracted and matched are mainly outside the robot.

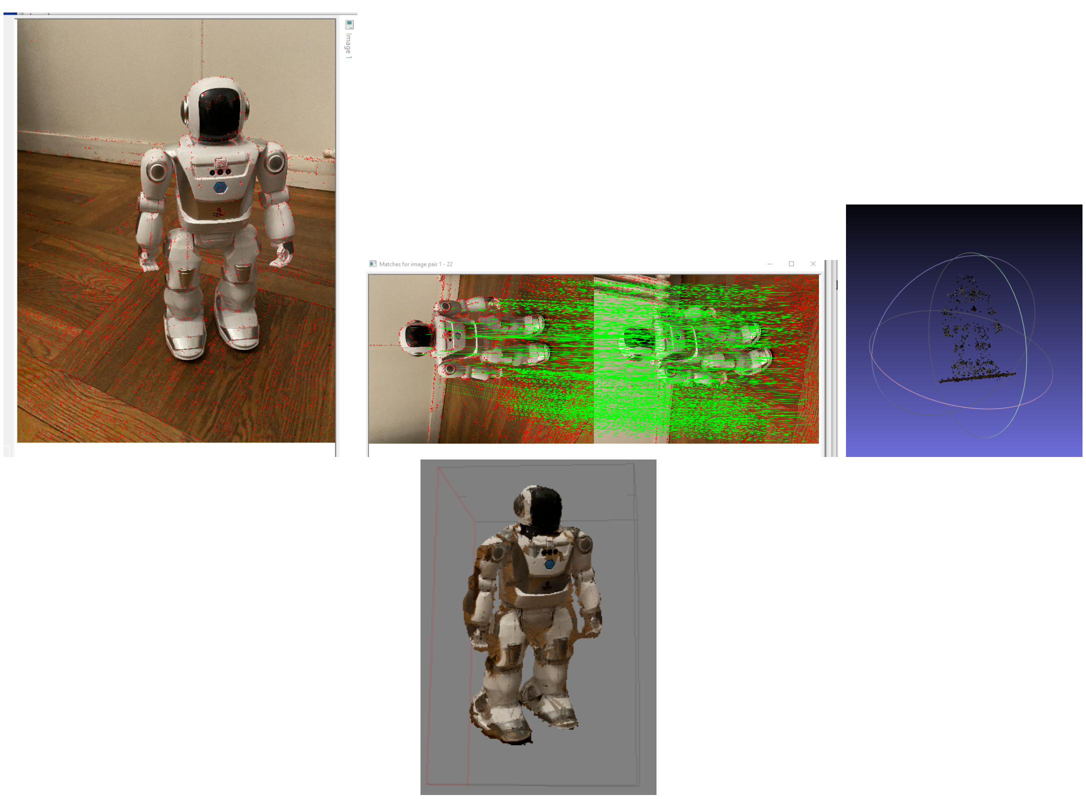


Figure 9: Robot results

6.2 Rock dataset

We also took pictures of a rock. The result was a much better of the robot, it because of the texture of the rock, it contains many details.

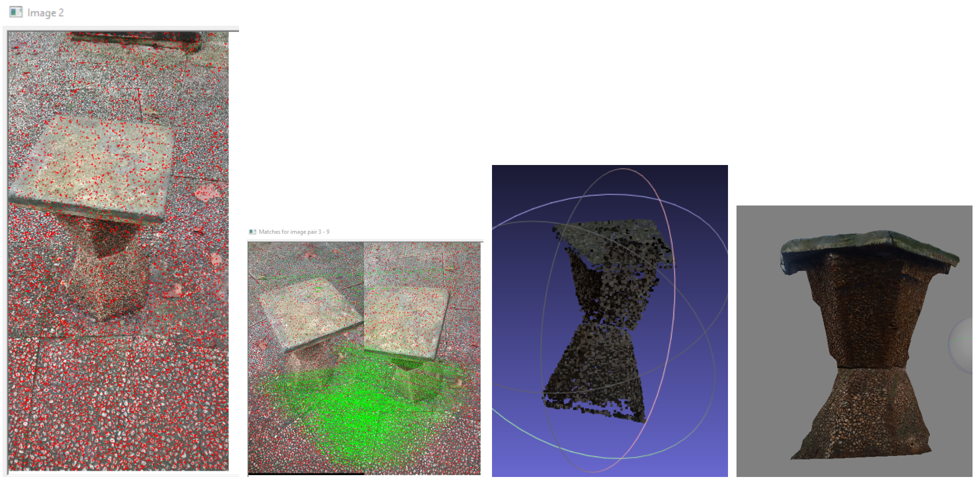


Figure 10: Rock results

6.3 Conclusion

During this project, we were able to learn many lessons, first the choice of the right software and its pipline. We were able to understand why an SfM pipeline is different from another and what all the 3D reconstruction softwares may offer. We learned how to choose the right dataset of 3D reconstruction problems, and we then could understand the result of each step, and interpret each value and output, until the final reconstruction.

References

[BCM18] Simone Bianco, Gianluigi Ciocca, and Davide Marelli. Evaluating the performance of structure from motion pipelines. *Journal of Imaging*, 4:98, 08 2018.

[COL] COLMAP.

[GHTC03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

[Lin12] Tony Lindeberg. *Scale Invariant Feature Transform*, volume 7. 05 2012.

[LMNF08] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem, 2008.