Module: 3D Vision
Project: 3D recovery of urban scenes (Session 4)

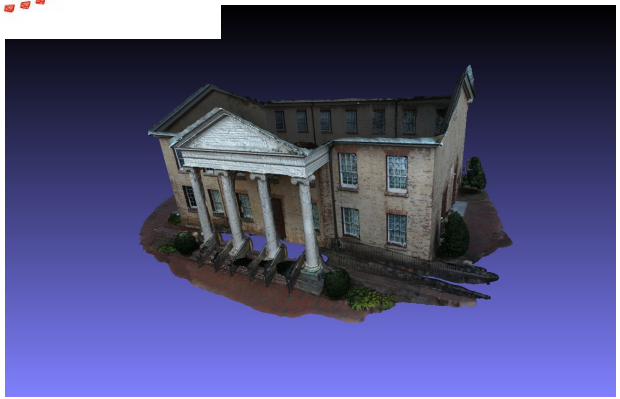
Original Lab: Marc Perez (marc.perez.quintana@upc.edu)

Goal

3D Reconstruction from uncalibrated images with a stratified method, by applying a Structure from Motion (SfM) pipeline in order to achieve a 3D sparse reconstruction. Dense 3D Reconstruction with Multi-View Stereo.

Sparse Reconstruction (Point cloud)

Images to 3d



Dense reconstruction (mesh)

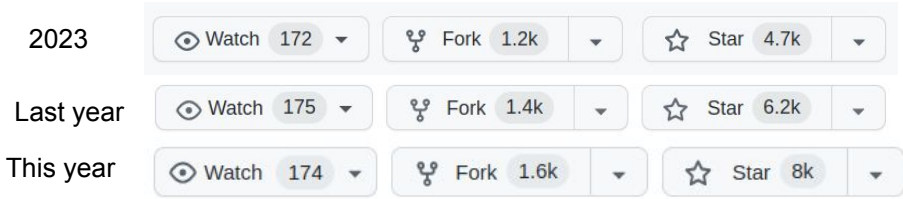
Colmap

https://github.com/colmap/colmap

- Opensource (BSD license)
- Pre-built binaries for Ubuntu/Windows/Mac
- Widely used

- Documented
  https://colmap.github.io/
  https://demuc.de/tutorials/cvpr2017/
- Maintained

- Code in C/C++

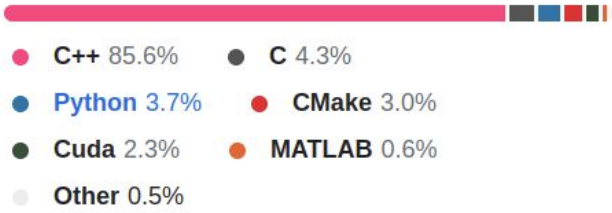


Cited by 6491 (last year: 4396, 2023: 2892)

```
@inproceedings{schoenberger2016sfm,
    author={Sch\"{o}nberger, Johannes Lutz and Frahm, Jan-Michael},
    title={Structure-from-Motion Revisited},
    booktitle={Conference on Computer Vision and Pattern Recognition (
    year={2016},
}

@inproceedings{schoenberger2016mvs,
    author={Sch\"{o}nberger, Johannes Lutz and Zheng, Enliang and Pollefeys, Marc and Frahm,
    title={Pixelwise View Selection for Unstructured Multi-View Stereo},
    booktitle={European Conference on Computer Vision (ECCV)},
    year={2016},
}
```

Cited by 2548 (last year: 1777, 2023: 1139)



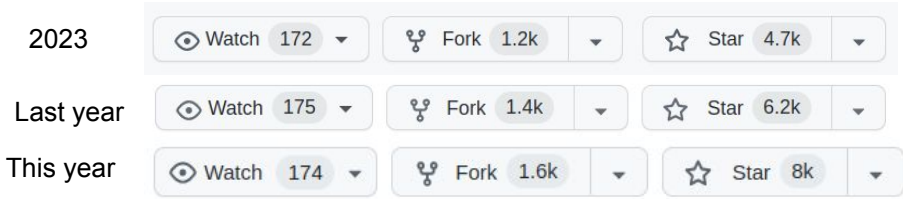
Colmap

https://github.com/colmap/colmap

- Opensource (BSD license)
- Pre-built binaries for Ubuntu/Windows/Mac
- Widely used

- Documented
  https://colmap.github.io/
  https://demuc.de/tutorials/cvpr2017/
- Maintained

- Code in python

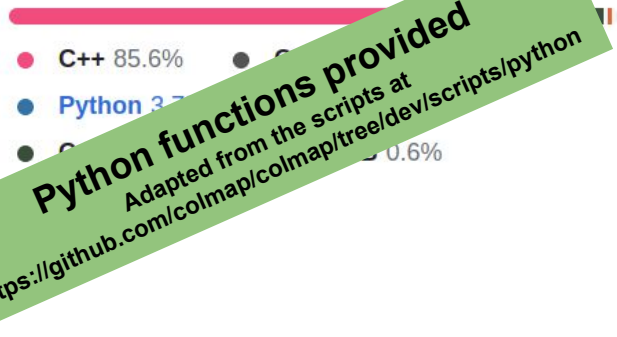


Cited by 6491
(last year: 4396,
2023: 2892)

Cited by 2548
(last year: 1
2023: 1139)

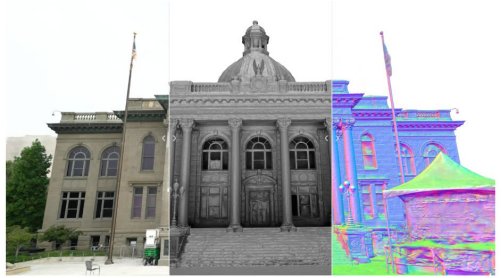
```
@inproceedings{schoenberger2016sfm,
    author={Sch\"{o}nberger, Johannes Lutz and Frahm, Jan-Michael},
    title={Structure-from-Motion Revisited},
    booktitle={Conference on Computer Vision and Pattern Recognition (
    year={2016},
}

@inproceedings{schoenberger2016mvs,
    author={Sch\"{o}nberger, Johannes Lutz and Zheng, Enliang and Pollefeys, Marc and Frahm
    title={Pixelwise View Selection for Unstructured Multi-View Stereo},
    booktitle={European Conference on Computer Vision (ECCV)},
    year={2016},
}
```



Colmap

- Feedback from 2023
  - “Don't ever use colmap again **there’s better alternatives**”



Code: https://github.com/NVlabs/neuralangelo
Project: https://research.nvidia.com/labs/dir/neuralangelo/
Paper: https://research.nvidia.com/labs/dir/neuralangelo/paper.pdf

Neuralangelo: High-Fidelity Neural Surface Reconstruction

Zhaoshuo Li[1,2] Thomas Müller[1] Alex Evans[1] Russell H. Taylor[2]
Mathias Unberath[2] Ming-Yu Liu[1] Chen-Hsuan Lin[1]

[1]NVIDIA Corporation [2]Johns Hopkins University

IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), 2023

TIME's Best Inventions of 2023

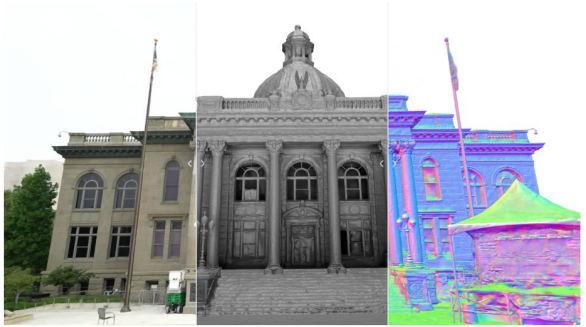
neuralangelo / **third_party** /

Add file ···

Max Li move colmap to tag 3.8 3464428 · 4 months ago History

Name	Last commit message	Last commit date
..		
colmap @ 43de802	move colmap to tag 3.8	4 months ago

2. **Q:** The reconstruction of my custom dataset is bad. What can I do?
   A: It is worth looking into the following:
   - The camera poses recovered by COLMAP may be off. We have implemented tools (using Blender or Jupyter notebook) to inspect the COLMAP results.
   - The computed bounding regions may be off and/or too small/large. Please refer to data preprocessing on how to adjust the bounding regions manually.
   - The video capture sequence may contain significant motion blur or out-of-focus frames. Higher shutter speed (reducing motion blur) and smaller aperture (increasing focus range) are very helpful.



graphdeco-inria / gaussian-splatting

Type / to search

Code | Issues 541 | Pull requests 29 | Actions | Projects | Security | Insights

Files

main

Go to file t

- SIBR_viewers
- arguments
- assets
- gaussian_renderer
- lpipsPyTorch
- scene
  - __init__.py
  - cameras.py
  - colmap_loader.py
  - dataset_readers.py
  - gaussian_model.py
- submodules
- utils
- .gitignore
- .gitmodules
- LICENSE.md

gaussian-splatting / scene / colmap_loader.py

bkerbl Bump rasterizer, faster text loading — ea68bdf · 2 years ago — History

Code | Blame | 294 lines (269 loc) · 11.6 KB — Raw

```
1   #
2   # Copyright (C) 2023, Inria
3   # GRAPHDECO research group, https://team.inria.fr/graphdeco
4   # All rights reserved.
5   #
6   # This software is free for non-commercial, research and evaluation use
7   # under the terms of the LICENSE.md file.
8   #
9   # For inquiries contact  george.drettakis@inria.fr
10  #
11
12  import numpy as np
13  import collections
14  import struct
15
16  CameraModel = collections.namedtuple(
17      "CameraModel", ["model_id", "model_name", "num_params"])
18  Camera = collections.namedtuple(
19      "Camera", ["id", "model", "width", "height", "params"])
20  BaseImage = collections.namedtuple(
21      "Image", ["id", "qvec", "tvec", "camera_id", "name", "xys", "point3D_ids"])
22  Point3D = collections.namedtuple(
23      "Point3D", ["id", "xyz", "rgb", "error", "image_ids", "point2D_idxs"])
```

Symbols

Find definitions and references for functions and other symbols in this file by clicking a symbol below or in the code.

Filter symbols r

- const CameraModel
- const Camera
- const BaseImage
- const Point3D
- const CAMERA_MODELS
- const CAMERA_MODEL_IDS
- const CAMERA_MODEL_NAMES
- func qvec2rotmat
- func rotmat2qvec
- class Image
  - func qvec2rotmat
- func read_next_bytes

Colmap

- Feedback from last year
  - “3d reconstruction with colmap seems **too much research oriented**. A pipeline for the industry, with some techniques that could be run in scripting for dense 3d could be awesome.”
  - “We had to spend some time **installing the software**, It'd [be] good if the students can download a virtual machine with all they need on it.”

Mandatory Tasks

- 1. 3D mesh reconstruction from a set of images from the Gerrard Hall dataset. **(1.5)**
- 2. Step-by-step 3D mesh reconstruction on the CASTLE dataset (lab 4) following the notebook provided, explain each step. **(6.0)**
- 3. Configure the reconstruction to improve the results. **(1.0)**
- 4. Reconstruct a 3D mesh from images captured by you. **(1.5)**

Optional Tasks

- 5. Use Neuralangelo or Gaussian Splatting to reconstruct a 3D mesh from images captured by you **(+2.0)**
  - https://research.nvidia.com/labs/dir/neuralangelo/
  - https://github.com/graphdeco-inria/gaussian-splatting/tree/main

Assignment

- Code is provided in python in a jupyter notebook.
- Deliver before 9 AM of Monday, January 27. (You have some extra time this week)

Deliverables

- **Jupyter notebook:** ready to run.
  - Document your code and decisions on markdown. All code in the notebook! **Not aux/utils files!**
  - Be clear of what information is assumed/required for each algorithm/operation.
  - Understand the equations do not just reproduce them from the slides.

- **Report:**
  - Short report with in depth analysis, insights, problems and comments.
  - Do not paste code in report. I am interested in analysis and justification.
  - You can use the notebook as a report **IF, AND ONLY IF,** you format the notebook appropriately.
  -  **UPLOAD THE NOTEBOOK EXECUTED** and just in case upload it as well as PDF or HTML.

 **Meshes:**

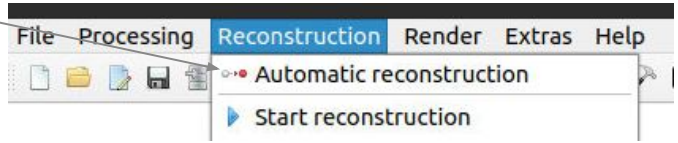
- The **3D meshes** generated in all exercises or videos showing the 3D structure of the mesh. (link to Google Drive for example, if they take too much space)

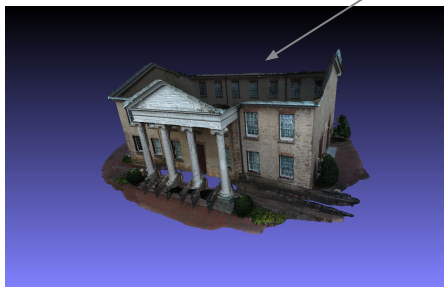
 **Naming conventions:**

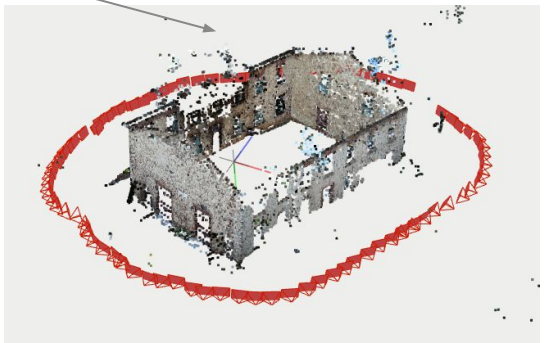
- All the files (zip, notebook, report) should be named [<GROUP_ID>]_lab4 (e.g. [1]_lab4.ipynb)

1. 3D mesh reconstruction from a set of images from the Gerrard Hall dataset **(1.5)**

- Install Colmap: https://colmap.github.io/install.html
- Read the documentation: https://colmap.github.io/, https://demuc.de/tutorials/cvpr2017/
- Download the Gerrard Hall dataset: https://demuc.de/colmap/datasets/
- Run the automatic reconstruction on the Gerrard Hall dataset
- Visualize the sparse reconstruction on Colmap
  (for example on Meshlab: https://www.meshlab.net/)





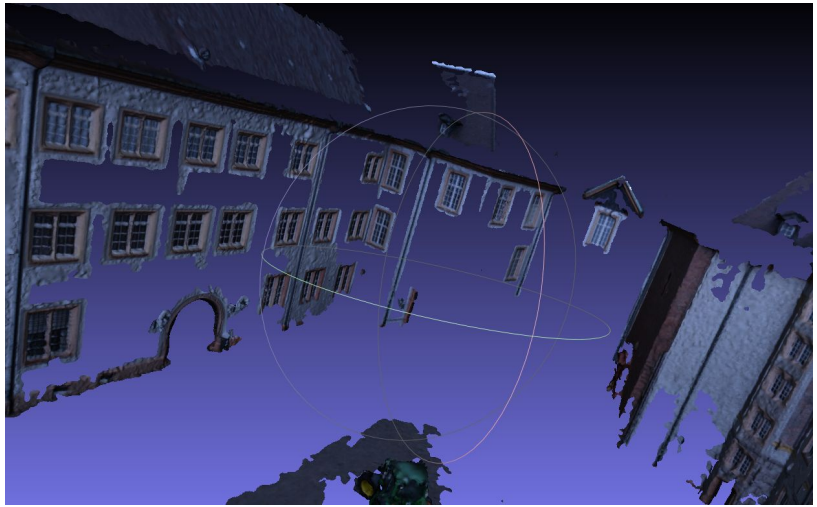


2. Analyze reconstructions using python. **(6.0)**

- 2.1 Run the notebook, using the Gerrard Hall reconstruction **(0.5)**
- 2.2 Plot the 3D points coloured according to the number of images and error. **(0.5)**
- 2.3 Plot the 3D points that originated from a keypoint in the first image. Also plot the image with the keypoints **(1.0)**
- 2.4 Create a visualization for the number of matches between all images. **(1.0)**
  - For example: https://seaborn.pydata.org/generated/seaborn.heatmap.html
- 2.5 Visualize the keypoints and matches between the two images used in lab 4 using Colmap, how it compares to the results from lab 4? **(1.0)**
- 2.6 Triangulate and visualize the 3D points from the keypoints extracted using Colmap on the two images used in lab 4, how it compares to the results from lab 4? **(1.0)**
- 2.7 Visualize the sparse reconstruction using the 2 images from lab 4, and the complete CASTLE dataset. Comment on the differences between techniques and number of images used. **(1.0)**
  - https://documents.epfl.ch/groups/c/cv/cvlab-unit/www/data/multiview/denseMVS.html (Castle P-30)

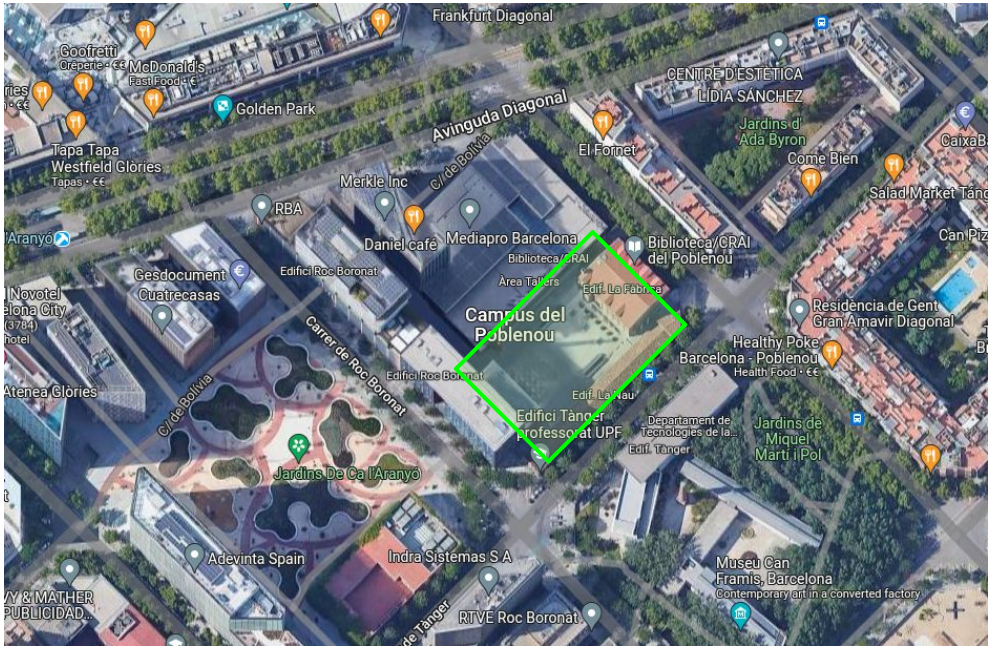
3. Configure the reconstruction to improve the results. **(1.0)**

- Configure the reconstruction to get the best possible mesh, explain the parameters changed and why you decided to change them. Remove noisy points with Meshlab or others.



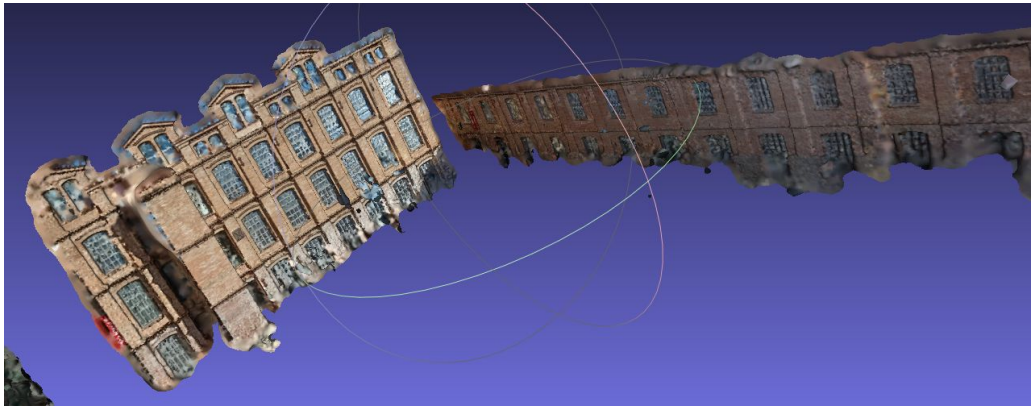
4. Reconstruct a 3D mesh from images captured by you. **(1.5)**

- Ideally from the same environment, to be able to compare between groups: UPF campus!



4. Reconstruct a 3D mesh from images captured by you. **(1.5)**

- Ideally from the same environment, to be able to compare between groups: UPF campus!
- Image capturing is important to get good results.



4. Reconstruct a 3D mesh from images captured by you. **(1.5)**

- What are the requirements for image capturing to get good 3D reconstructions?
- You might need to change the configuration to get a good reconstruction, comment on the changes you make.
- You can apply it to other scenes or objects.
- You can try with different cameras.
- You can try with images from the web.

5. Use **Neuralangelo** or Gaussian splatting to reconstruct a 3D mesh from images captured by you (Optional **2.0**)

https://research.nvidia.com/labs/dir/neuralangelo/

https://github.com/graphdeco-inria/gaussian-splatting/tree/main

Image from the paper:
https://research.nvidia.com/labs/dir/neuralangelo/paper.pdf

5. Use Neuralangelo or **Gaussian splatting** to reconstruct a 3D mesh from images captured by you (Optional **2.0**)

https://research.nvidia.com/labs/dir/neuralangelo/

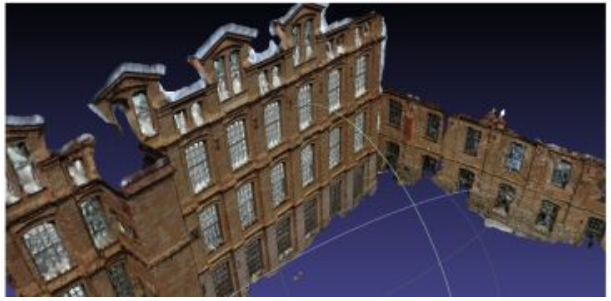
https://github.com/graphdeco-inria/gaussian-splatting/tree/main

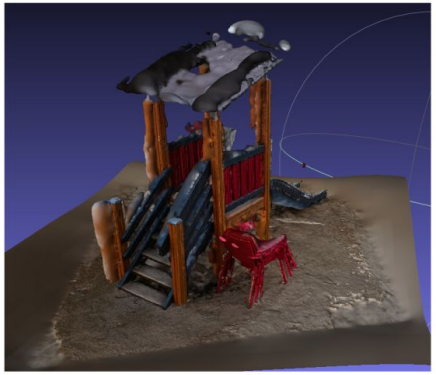
Image from the paper:
https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/3d_gaussian_splatting_high.pdf

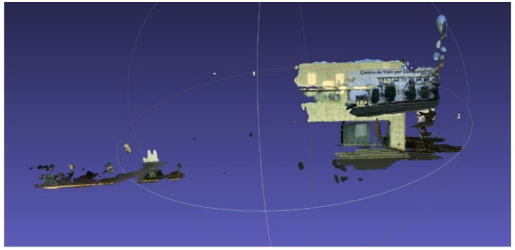
Every group can access a GPU?

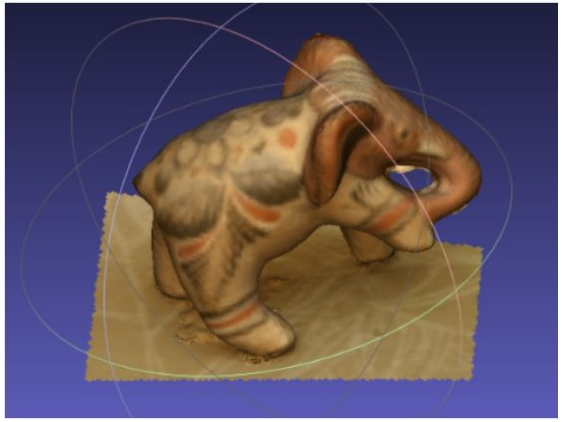
There is an option to use a cluster, but it has not been tested.
If any group wants to try let me know.

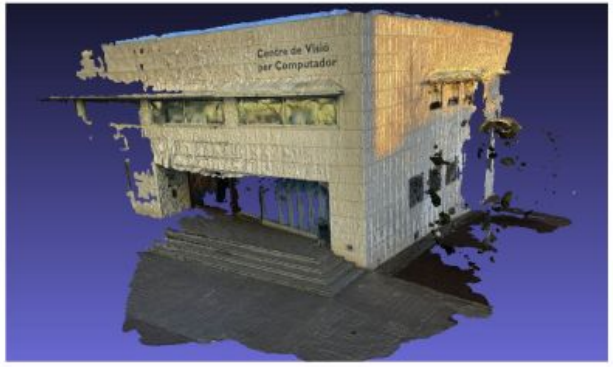
Some 3D reconstructions from 2023





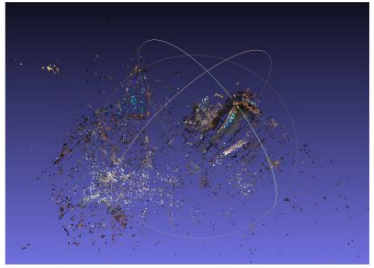




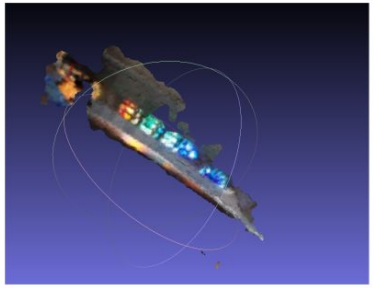


Some 3D reconstructions from 2023

(a) Sample image for the generated dataset of Sagrada Familia from Google StreetView

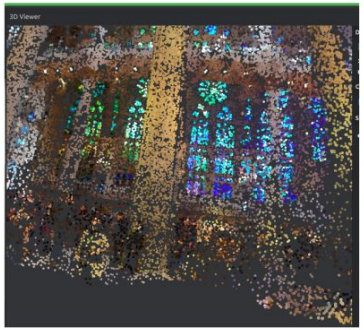


(b) Sparse representation of the data of Sagrada Familia

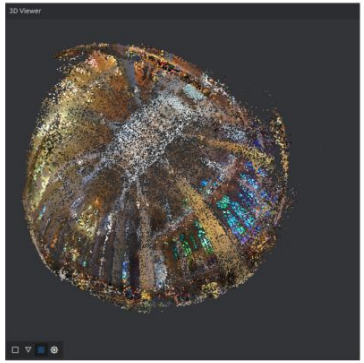


(c) Dense representation for the data of Sagrada Familia

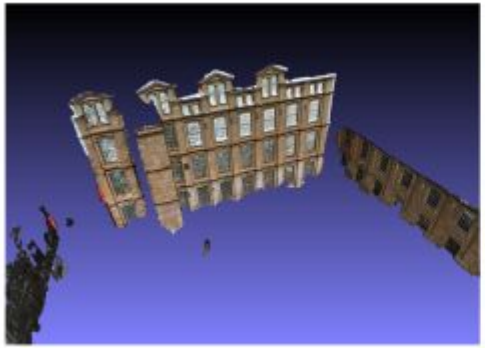
(a) Example of working with `Meshroom`



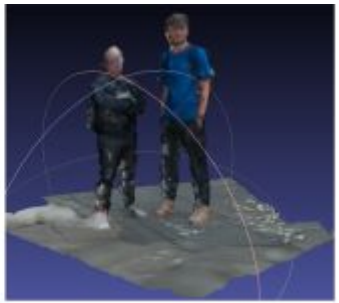
(b) Inner view of Sagrada Familia reconstruction by `Meshroom`.

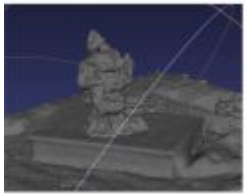


(c) Bottom view of Sagrada Familia reconstruction.



Some 3D reconstructions from last year





Neuralangelo

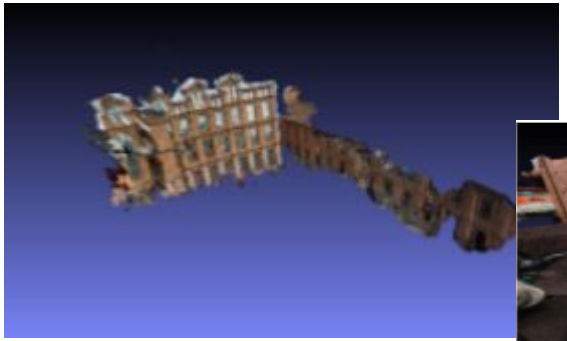
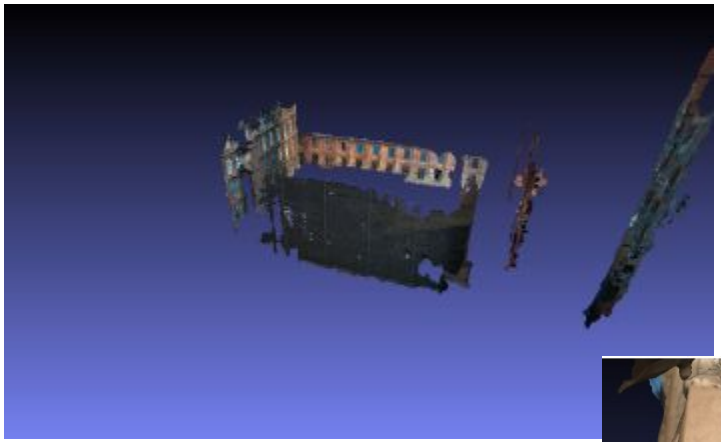


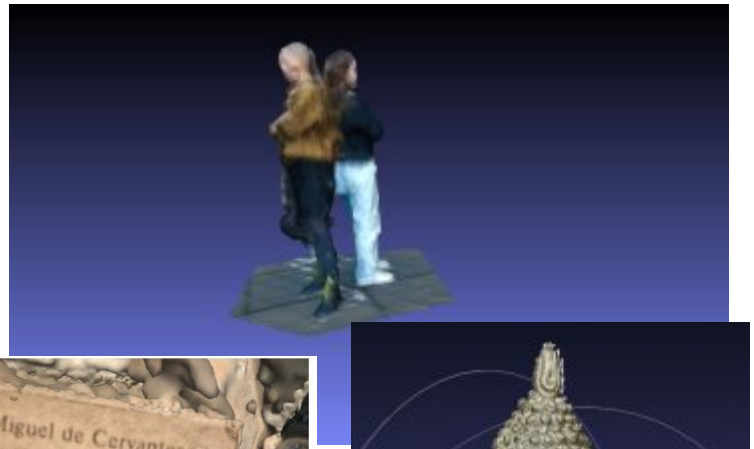
Fig. 22: Neuralangelo reconstruction from 100 epochs

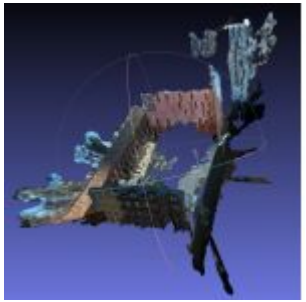
Fig. 23: Neuralangelo reconstruction from 44k epochs

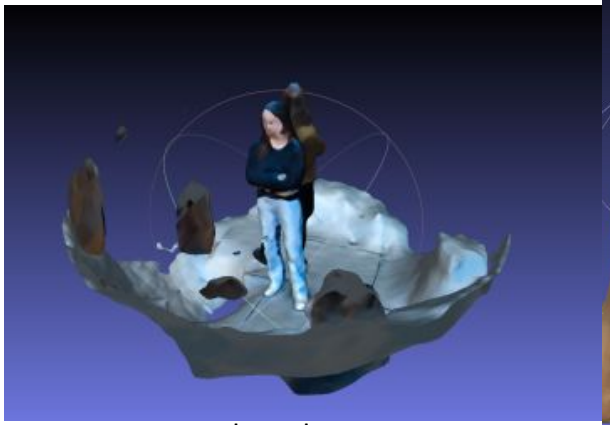
(Provided dataset)

Some 3D reconstructions from last year









Neuralangelo

Some 3D reconstructions from last year