3D Gaussian Splatting Documentation

Aiman Anuar | anuaraimanimran@gmail.com
Yale Peabody Museum
PI: Nelson Rios

Introduction

Hello, this documentation covers nearly all of the research I've done into 3D Gaussian Splatting so that future investigators may be able to sift through all of this easily and get started quickly. It contains information on how to use the 3D Gaussian Splatting workflow from a video input to mesh output.

Relevant Links

- This document can be viewed on Google Docs at:
  3D Gaussian Splatting Documentation - Aiman Anuar
- This project's GitHub repo can be found at:
  GitHub Repo
- This project's relevant datasets can be found at:
  Postshot Datasets (1/2)
  Postshot Datasets (2/2)
  SuGaR Datasets (1/2)
  SuGaR Datasets (2/2)
- The download page for Jawset Postshot can be found at:
  Jawset Postshot
- The SuperSplat viewer can be found at:
  SuperSplat
- The video demonstration of the pipeline can be found at:
  Insert link here

Relevant Software

- FFMPEG
- COLMAP
- CUDA
- Conda
- Windows VS Studio
- Postshot
- Supersplat
- Blender

Contact

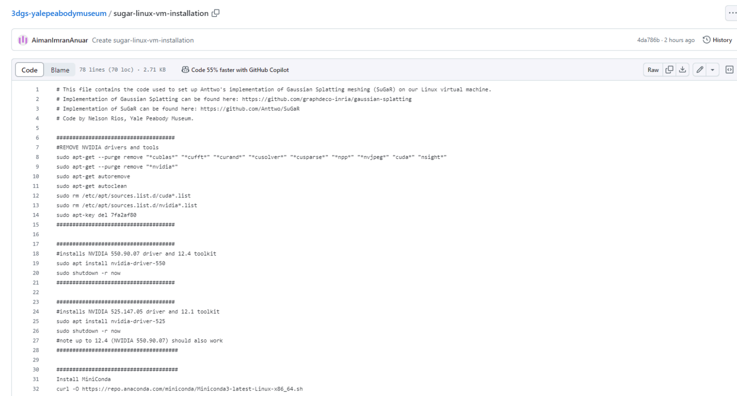
If you need to contact me, email me at anuaraimanimran@gmail.com.

3D Gaussian Splatting Pipeline Steps

The below describes the pipeline I used. Note that this entire thing can be done in Linux alone, but I did not want to set up FFMPEG or COLMAP on the virtual machine so as not to introduce possible issues. Thanks to Jon Stephens for his code & tutorial on how to do this all. His tutorial can be found [here](#).

-1. SET-UP (INITIAL)

The first step that only needs to be done once is to install the necessary drivers and packages and add the necessary variables to the path. Follow the steps in Jon Stephen's guide for more detailed guidance. Commands for Linux VM driver installation can be found on the GitHub repo.

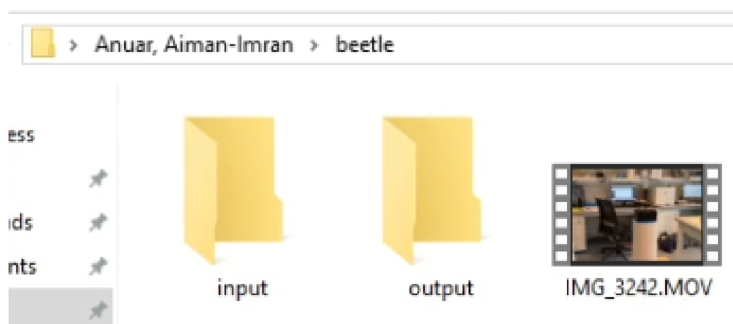


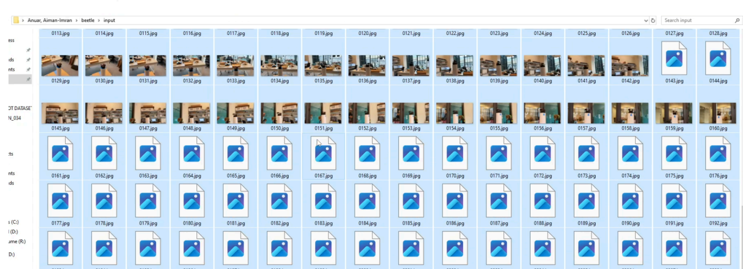
0. IMAGE ACQUISITION (INITIAL)

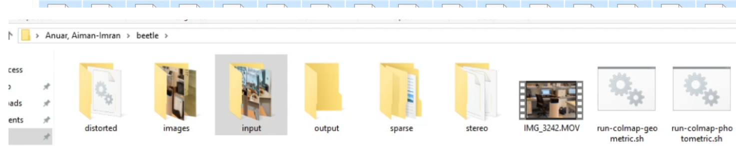
Acquire a video or pictures of your object, following photogrammetry best practice. Make sure it is not blurry, that you are able to capture distinct features and multiple angles, and try to use a camera that records EXIF (positional) data. Upload it to your computer.

1. IMAGE PROCESSING (WINDOWS)

In File Explorer, navigate to your *User* folder and then into the *gaussian-splatting* folder, and then create a new folder to store your data (**data_folder**). Then create two folders within that one, titled (**input**) and (**output**) without the parentheses. *Input MUST be named "input".* If you have pictures of your object, simply put them in **input**. If you have a video, convert it to an image set using FFMPEG.







2. STRUCTURE FROM MOTION (WINDOWS)

Open Anaconda Prompt and navigate to your gaussian splatting directory and activate the Conda environment for Gaussian splatting.

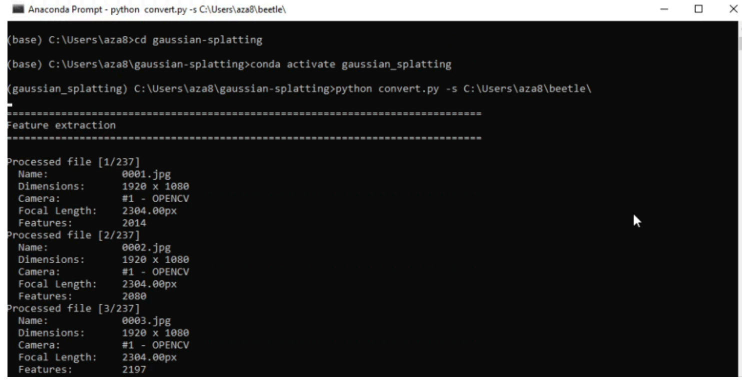
conda activate gaussian_splatting

Staying in that directory (as it needs the script), apply structure-from-motion conversion.

python convert.py -s ***data_folder***

Then zip your data folder and move it to the SuGaR directory of your Linux VM.

scp ***data_zip.zip*** *~/SuGaR*



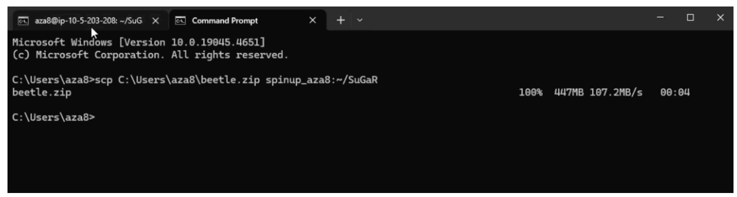
3. GAUSSIAN SPLATTING (LINUX)

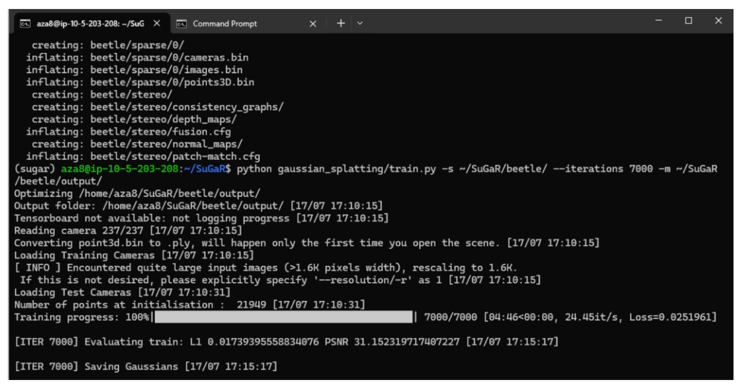
Navigate to your SuGaR directory, and unzip your data folder.

unzip ***data_zip.zip***

Then train your initial Gaussian splat. Make sure to include the backslashes at the end of files!

python gaussian_splatting/train.py -s ***data_folder/*** *--iterations 7000 -m* ***data_folder/output/***



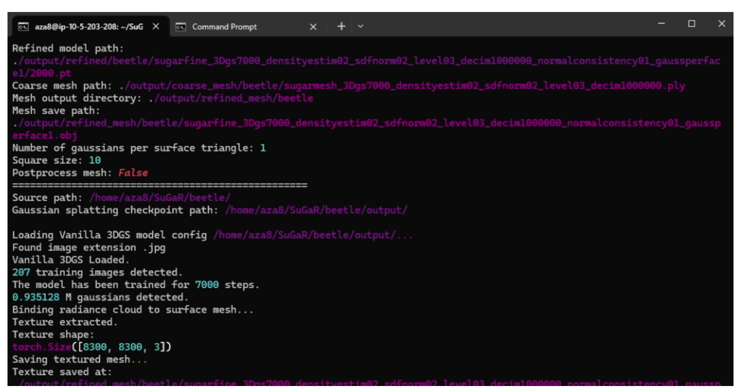


4. SURFACE GENERATION (LINUX)

If no errors occur, train the optimized Gaussian splat.

python train.py -s ***data_folder/*** *-c* ***data_folder/output/*** *-r* ***"density"/"sdf"*** *--refinement_time* ***"short"/"medium"/"long"***

The first parameter is how to refine: "density" for objects, "SDF" for scenes, and then you can choose how long to refine model.



5. MODEL VIEWING (WINDOWS)

Export the model's mesh and ply from the Linux machine to the Windows machine.

In Linux:

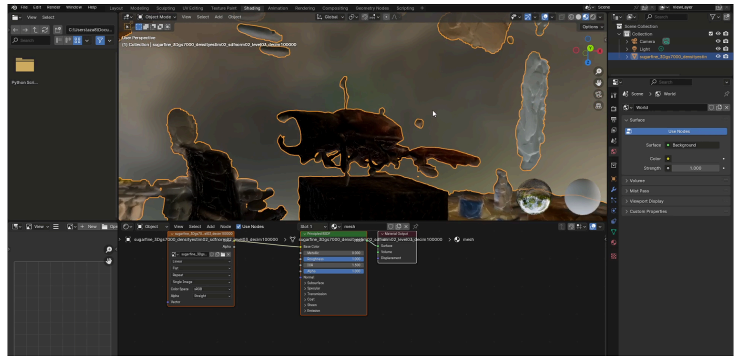
```
zip mesh_export.zip output/refined_mesh/data/ -r
```

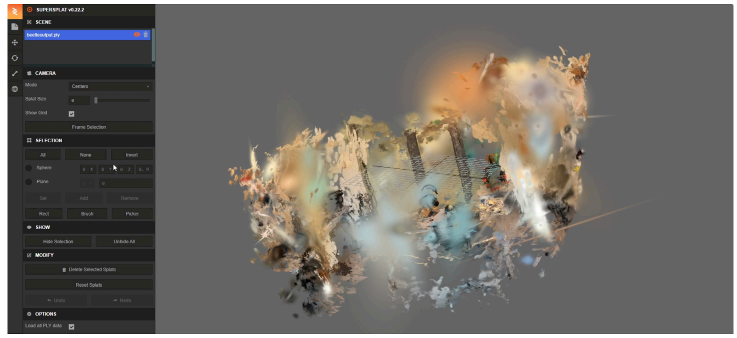
```
scp mesh_export.zip .
```

```
scp ply_export .
```

You can view and manipulate the mesh by inserting the .obj and .mtl file into Blender and selecting "Shading" to see it in color (you may have to zoom in). You can also crop it to the area you would like.

You can view and manipulate the ply file by inserting the file into SuperSplat and decreasing the parameter "Splat Size" to 0 to see the model better.





3D Gaussian Splatting Pipeline Code

FFMPEG –i **<VIDEO>** -qscale:v 1 -qmin 1 -vf fps=**<FRAME EXTRACTION RATE>** %04d.jpg

Convert video to image set using FFMPEG.

conda activate gaussian_splatting

Activate the gaussian-splatting environment on Anaconda Prompt.

python convert.py -s **<DATA_FOLDER>**

Run a Python script to run Structure-From-Motion on your data using COLMAP.

scp **<DATA_ZIP> . <LINUX_SUGAR_DIRECTORY>**

Move your zipped data folder to your Linux machine.

conda activate sugar

Activate the sugar environment on your Linux machine.

unzip **<DATA_ZIP>**

Unzip the zipped data folder.

python gaussian_splatting/train.py -s **<DATA_FOLDER>** –iterations 7000 -m **<OUTPUT>**

Train an initial Gaussian splat of your data.

python train.py -s **<DATA_FOLDEER>** -c **<OUTPUT>** -r **<“density”/“sdf”>** --refinement_time **<“short”/“medium”/“long”>**

Train a surface generation of your data.

zip **<REFINED_MESH_ZIP> <REFINED_MESH_FOLDER>** -r

Zip the output folder containing your mesh.

scp **<REFINED_MESH_ZIP>** .

Move your zipped output folder to your Windows machine.

scp **<REFINED_PLY>** .

Move your output .ply++ file to your Windows machine.

3D Gaussian Splatting Pipeline Code Example

FFMPEG –i **C:\Users\aza8\beetle\beetle.MOV** -qscale:v 1 -qmin 1 -vf fps=**1** %04d.jpg

Convert beetle video “beetle.MOV” at its path to image set using FFMPEG.

conda activate gaussian_splatting

Activate the gaussian-splatting environment on Anaconda Prompt.

python convert.py -s **C:\Users\aza8\beetle**

Run COLMAP on the “beetle” folder. Requires an “input” folder with images.

scp **C:\Users\aza8\beetle.zip . spinup_aza8:~/SuGaR**

Move beetle zip folder to the SuGaR directory on the Linux machine.

conda activate sugar

Activate the sugar environment on your Linux machine.

unzip **beetle.zip**

Unzip the beetle zip folder.

python gaussian_splatting/train.py -s **~/SuGaR/beetle/** –iterations 7000 -m **~/SuGaR/beetle/output**

Train an initial Gaussian splat of the beetle data, output it into the beetle output folder.

python train.py -s **~/SuGaR/beetle/** -c **~/SuGaR/beetle/output** -r **“density”** --refinement_time **“short”**

Train a short, density-based surface generation of the beetle data.

zip **beetleoutput.zip beetle** -r

From the refined_mesh folder, zip the beetle data.

scp **spinup_aza8:~/SuGaR/output/refined_mesh/beetleoutput.zip** .

Move the zipped output folder to the Windows machine.

scp **spinup_aza8:~/SuGaR/output/refined_ply/*long_file_name*** .

Move your output .ply++ file to your Windows machine.

Appendix