

# CSCI 677: Advanced Computer Vision - Fall 2024

Instructor: Prof. Nevatia

## Assignment 3

**Due on October 14, 2024 before 13:00 PDT**

This is a programming assignment consisting of two parts: First is to apply COLMAP software for inferring structure from motion (and camera poses) from a given set of images and the second is to use the SFM output to train a Gaussian Splatting (GS) model for rendering novel views. Details of the two parts are given below. In both cases, the code is essentially made available so your effort is mostly in evaluating the results based on some variations and in describing the results. The GS code requires use of a GPU, if you do not have a computer with a suitable GPU, you may use Google Colab (instructions will be provided in class).

Please do not use language tools such as ChatGPT to write the code or the report for you.

### 1 SFM: COLMAP

In class, we studied COLMAP as one of the available SFM software packages that can take a set of images from cameras of unknown poses and infer these camera poses and also reconstruct 3D positions of a sparse set of points. Code for COLMAP is available at <https://github.com/colmap/colmap>. Code is available as binaries for Windows, Linux and MacOS environments or can be built from the available source files; use of binaries is acceptable for the assignment. The github page also provides good documentation for using the software package.

We provide a set of 20 images from the "South Building" dataset here. We ask you to process the data in two sets; one is to use just the first 10 images, the other is to use all 20. The software will output files containing camera poses and reconstructed 3D points. We suggest that for camera model, you choose "SIMPLE\_PINHOLE". For matching, try "exhaustive" and "sequential" options.

Visualize the results using available commands and evaluate qualitatively. The software offers several visualization tools such as showing the detected feature points, matched points and camera locations. Use your own steps to best show the quality of results. Besides qualitative evaluation, also consider the reprojection error (which is the quantity that the method optimizes); you can get this error from the GUI by following "Extras, Show model statistics". Compare the results using the two given matching modes and also with the two different sized datasets.

You may find it easier to your own computer for this part of the assignment though you may also use Colab if you prefer.

## 2 Gaussian Splatting

In this part, we ask you to learn a GS model for the images used in the SFM part above, using the camera poses and sparse reconstruction points and to use the learned model to infer images from other viewpoints. You may use the matching option that gives the best SFM results.

You can find the code and other useful information of the original GS implementation at <https://github.com/graphdeco-inria/gaussian-splatting>. The code requires a GPU to run efficiently. We do not expect students to have a GPU-enabled personal computer so recommend using Colab. Fortunately, <https://github.com/camenduru/gaussian-splatting-colab> provides us with directly usable code for Colab.

Once a GS model is trained, it can be used to render the scene from arbitrary viewpoints and we encourage you to do so and evaluate visually. For more precise evaluation, we can use the images in our dataset by separating them in training and test sets. Please use the following settings.

- Use the SFM models the first 10 images but exclude “image #5” from training set for GS model. Use camera pose for “image #5” to render using the GS model and compare with the actual image. You should also compute quantitative metrics; the software provides an option to compute these. Note that the test is a little bit imprecise because “image #5” will have been used in COLMAP to add to the set of sparse 3D points used to initialize.
- Repeat the above experiment but using 20 images this time with images #s 5, 10 and 15 used as test images (and excluded from the training set). Observe how the results change with more training images, if at all.
- Repeat both of above experiments with different number of iterations and note the dependence, if any, of the quality of rendering on this number.

The software package also offers options to change several other parameters. You may experiment with these for better understanding but are not required for the assignment and will not be given extra credit.

## What to Submit

As the code for this assignment is mostly provided, you should focus on displaying and analyzing results (both qualitatively and quantitatively where possible). Hints of what to include are in the descriptions above.

You should submit a single PDF file including your source code and report.