**Guide to make use of Nvidia Instant NGP**

Heavily inspired from repo below:

* The original repository link – <https://github.com/NVlabs/instant-ngp>
* Better one with more detailed installation guide - <https://github.com/bycloudai/instant-ngp-Windows>

**Installation:**

Note- Make sure no directory name has a space in between i.e., whenever you create a folder, don’t add space in between the name.

1. Install anaconda
2. Using conda in anaconda prompt, we will clone the repo and install the library
3. If working on windows, we need to reset the filename limit by following command:  
     
   git config --global core.longpaths true  
     
   Else continue to next step
4. Create a folder where you want to install **instant\_ngp** and open it in your terminal and then run:  
   git clone --recursive https://github.com/nvlabs/instant-ngp
5. Install cmake if it’s not installed or present already from - <https://cmake.org/download/>



Download the zip and just unzip it in a location. Then add this folder’s “bin” location to the Environment path, similar to below example- Z:\amit\_amola\Augmented\softwares\cmake-3.26.3-windows-x86\_64\bin

Instructions to perform this are at - <https://learn.microsoft.com/en-us/previous-versions/office/developer/sharepoint-2010/ee537574(v=office.14)>

1. Now we will perform some builds using cmake. Make sure you didn’t create any folder with spaces in it. Go into the main directory and type following commands:

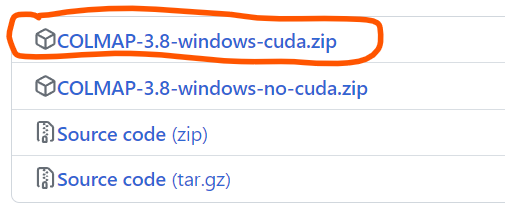
cmake . -B build

cmake --build build --config RelWithDebInfo -j 16

There might be some errors, if that’s the case then this might not work properly.

1. Now we will install Colmap that will create the transform.json that we will use later:

Go to the link - <https://github.com/colmap/colmap/releases/tag/3.8>

At the end of the page, download the cuda.zip version

Unzip this file in a location and add the location of the folder which has Batch file inside. Like below:

Z:\amit\_amola\Augmented\softwares\COLMAP-3.8-windows-cuda

1. Create a new virtual environment using conda. Steps are provided in the appendix.

conda create -n nv\_ngp python=3.9

conda activate ngp

pip install -r requirements.txt

1. We have to separately install pyexr. Try running pip install pyexr

In case it doesn’t work, we can install it via whl file. Download [OpenEXR‑1.3.2‑cp39‑cp39‑win\_amd64.whl](https://www.lfd.uci.edu/~gohlke/pythonlibs/#openexr) and move it to your main folder. Then you can run:

pip install OpenEXR-1.3.2-cp39-cp39-win\_amd64.whl

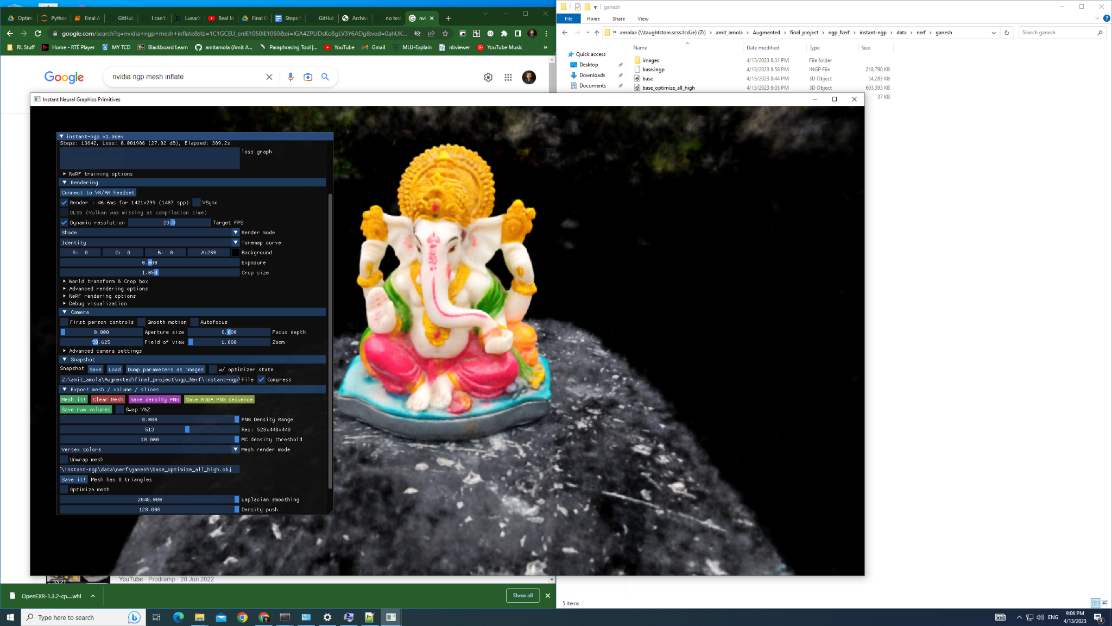
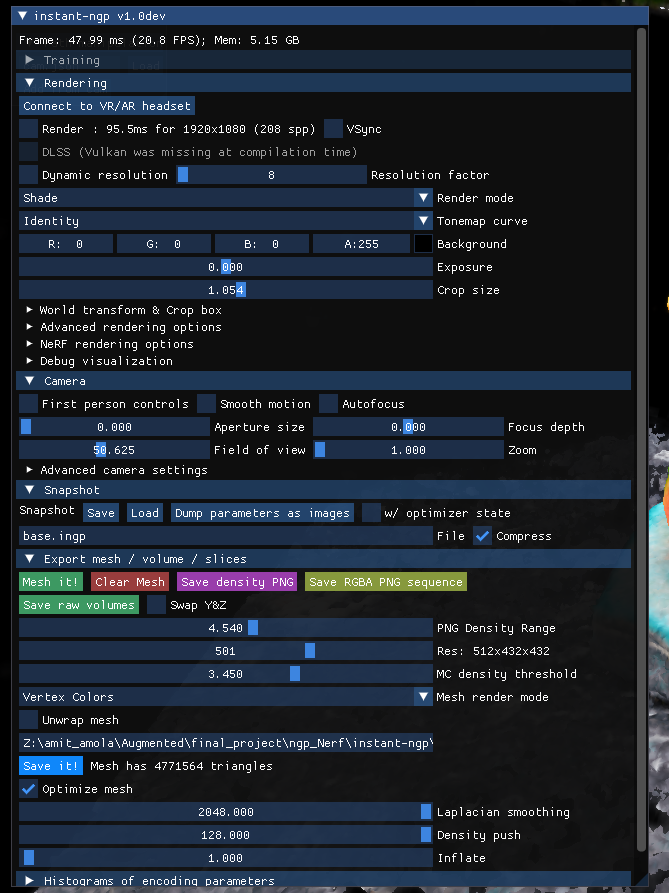
1. Place your custom image set under data/<project\_name>/images

Get transform.json from the following command. Insert your path to your images at <image/path>

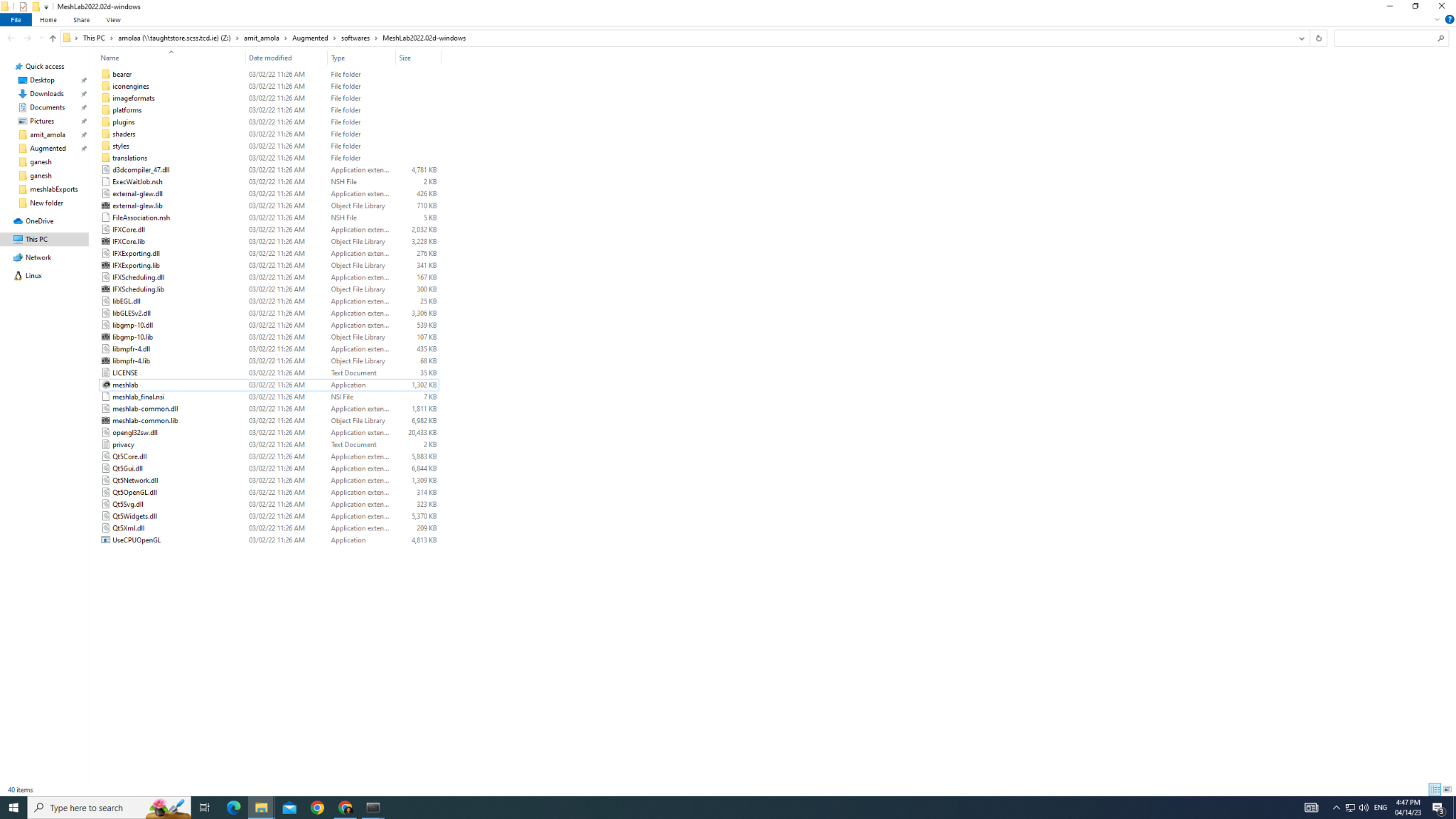
python scripts/colmap2nerf.py --colmap\_matcher exhaustive --run\_colmap --aabb\_scale 16 --images <image/path>

There might be some issue with COLMAP at this moment. So close the terminal, add the COLMAP to the path again. It might happen in case of windows. So when you run the above command once, it downloads the compatible COLMAP. It will be one directory behind your current one inside the folder - external. Find the path where the Batch(.BAT) file is and add it to the path like you did before.

And now open the conda terminal again, go to your main working directory and run that above command again. It should work well this time.

1. Once it’s done running, it will produce transform.json file. Copy this file and paste it in data/<project\_name>/
2. Now set transform param to low ‘aabb\_scale’ if the object is small. Less clutter. I set it to 8.
3. Now try to keep resolution small enough. Larger the resolution, more memory it will consume and the application might crash. So, keep it below 400-500. Moreover, MC density threshold value is something to play with. Lower the value, more triangles it forms but messes up the mesh creation process. Higher the value will decrease the triangles but it won’t be that precise. Usually, a value between 4 -5 is good.

Click on Optimize mesh and then click on Save it!

1. Since the mesh creation happens using colouring the vertices and not using a texture map, importing this object in Blender won’t work well. So we are going to make use of meshlab. Go to - <https://github.com/cnr-isti-vclab/meshlab/releases/tag/MeshLab-2022.02> and download the windows zip version. And similarly as above, unzip it and we can now use the **meshlab** application:
2. Once open, drag and drop your file on the app. Now we can perform cleaning the extra noise around our object, the same way we did in Blender using meshroom object. Here are your two tools to perform this:  
     
   The tool in the green circle is for selecting and the red one then deletes. You can also perform selection and then do invert selection using Ctrl+Shift+I and then perform deletion like we did in Blender.
3. One can also perform subdivision of the object to make it look more detailed, but it will lead to increase in size of the object file.

Conclusion - the output isn’t that amazing.

**Best NGP Nerf params:**

* "Aabb\_scale" parameter - This parameter specifies the extent of the scene, defaulting to 1; that is, the scene is scaled such that the camera positions are at an average distance of 1 unit from the origin. For small synthetic scenes such as the original NeRF dataset, the default aabb\_scale of 1 is ideal and leads to fastest training. The NeRF model makes the assumption that the training images can entirely be explained by a scene contained within this bounding box. However if you were looking to make a larger scene area like the Train station example, like displayed in 0:26 in the video- <https://www.youtube.com/watch?v=fvXOjV7EHbk>, then you will use a larger value. It can at max take a value of 128. More details available at: <https://github.com/NVlabs/instant-ngp/blob/master/docs/nerf_dataset_tips.md#:~:text=The%20aabb_scale%20parameter%20is,scripts/colmap2nerf.py%20script>.