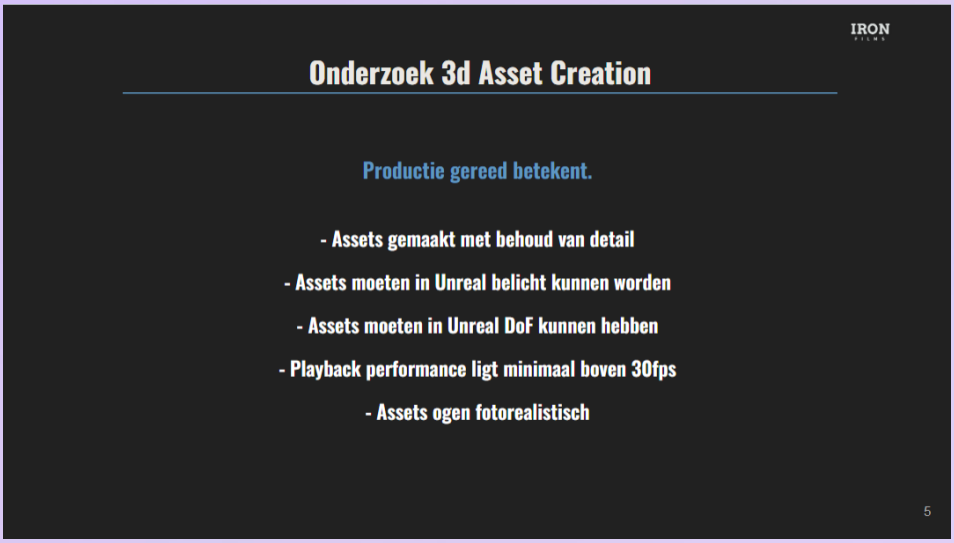
4. Client feedback influence

Intro

Getting close to the prototype showcase day, I felt like in the last couple of weeks even though the results started to get better once we applied our client's feedback to find new locations and work with those new environments, my results were still limited by the fact that they were still based on photogrammetry. I kept revisiting the first presentation the client did and looked very well through the requirements and what expectations they had for our project. Our target was to create an efficient workflow to generate production-ready assets. Photogrammetry didn't seem to be the best way of getting virtual production assets or real-time rendering quality in engines. So, I started looking into NeRF and Gaussian Splatting.



Process & Feedback

Throughout my research I confirmed my beliefs, summarising my findings into a simple Photogrammetry vs NeRF vs Gaussian overview:

Photogrammetry result:
Mesh + texture
Best for:
• real-world measurements
• game-ready assets
• accurate, editable, relatively light (if you have very good photos)

NeRF result:
Volumetric radiance field
Best for:
• virtual production
• camera flythroughs
• novel-view synthesis
• full-scene reconstructions
(Not measurable, pretty heavy)

Gaussian Splatting result:
Real-time splat cloud
Best for:
• real-time rendering in engines
• fast, sharp, Unity/Unreal-friendly
(Not a mesh, not measurable)

Conclusion:
Photogrammetry is excellent for accuracy and measurement.
NeRF and Gaussian Splatting provide much more immersive and flexible visual results for virtual production, with Gaussian Splatting offering extremely fast performance compared to classic NeRF.

Gaussian Splatting research:

Gaussian Splatting research - document

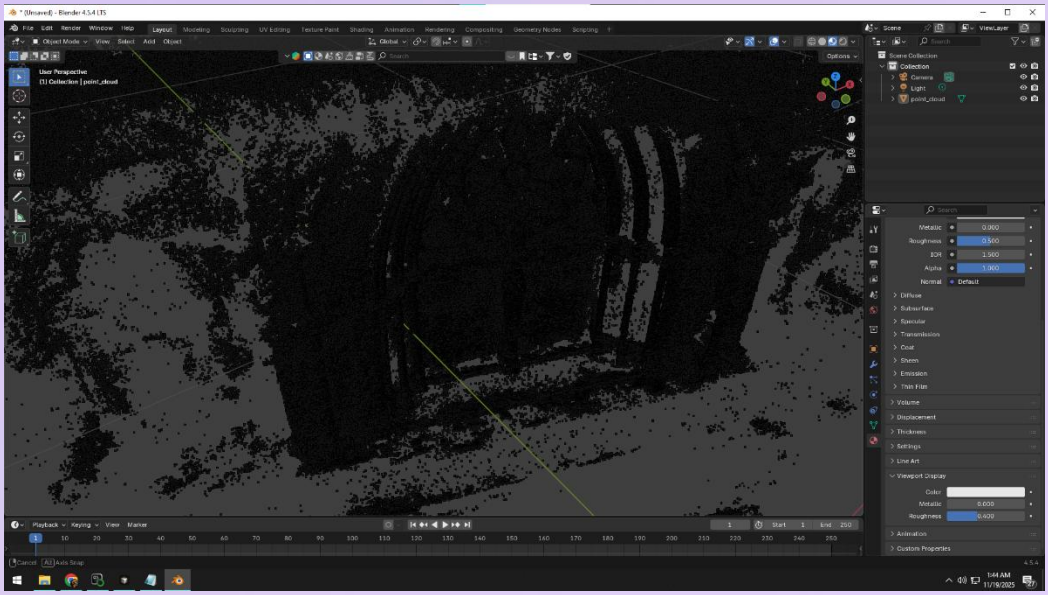
Since the client expected real-time-ready assets for virtual production, Gaussian Splatting seemed the closest to meeting that requirement. So it was time to explore these new technologies.

During my first try with NeRF Studio, it ended up failing to produce a Gaussian Splat because my hardware didn't meet the CUDA compute requirement (+7.0). My GTX 1070 only has compute capability 6.2, so even though I was able to get a NeRF render, my goal was to get Gaussian Splatting working. Linux with NVIDIA drivers provided a workaround, so I borrowed a USB stick and an external HDD from ISSD, installed Ubuntu on it, and started building everything from scratch to get a Gaussian Splat result running.

NeRFStudio try out:

NeRFStudio on windows – document

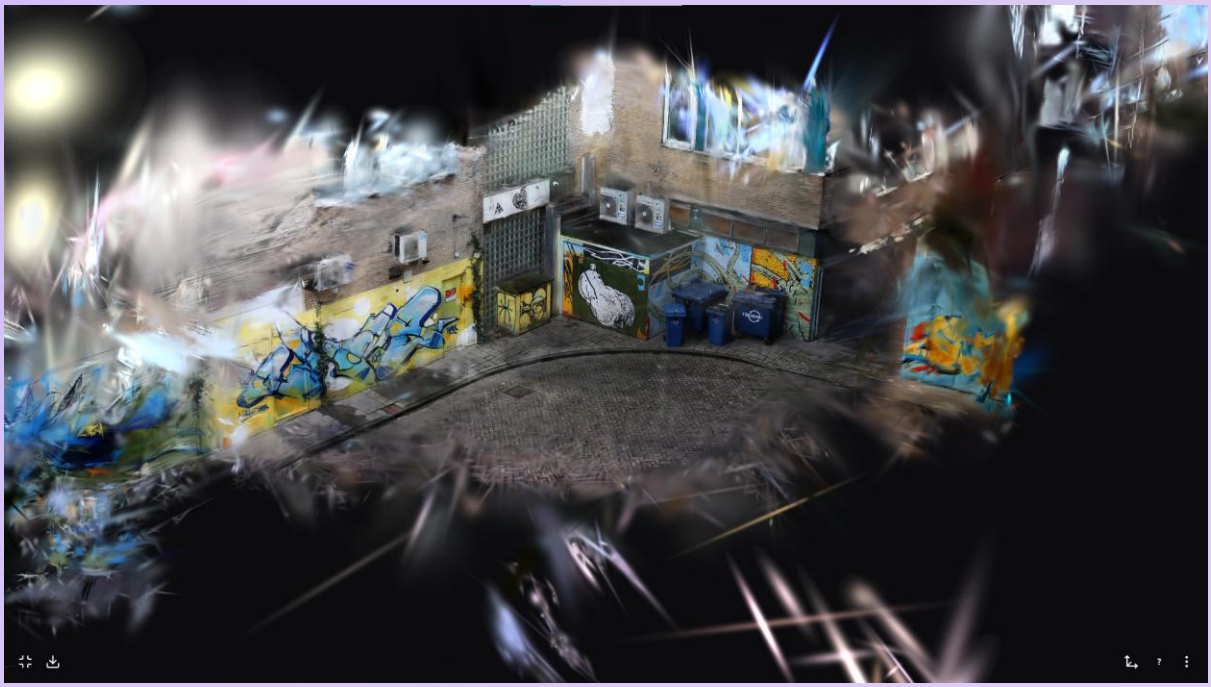
During this whole process I documented all the steps: installing NVIDIA drivers, the CUDA 12.4 toolkit, creating the conda environment, installing PyTorch, setting up the ffmpeg + COLMAP combo to process the dataset, then finally using the official 3DGS repository that included the converter from photos to a sparse cloud plus the train.py that performs the training on the dataset to produce the splat cloud and visualise the final Gaussian render.

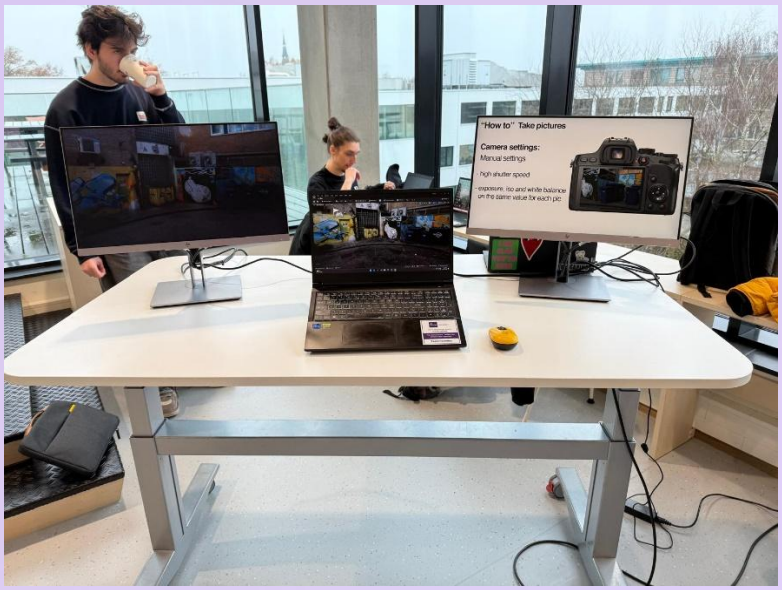


Guide to Gaussian from scratch:

GuideToGaussian - document

Feedback came in strong and clear during the prototype day. It was all positive, and a few teachers even said the results were impressive and unexpected. Most of the questions were focused on what the next steps are now that we managed to achieve this quality with both photogrammetry and Gaussian Splatting.





Reflection

Moving forward from this stage, the next steps would focus on making the workflow accessible and usable for others. One idea is creating a simple website with two main sections: “Collection” and “Documentation.” The Collection page would contain our models, and the Documentation page would explain the workflow, requirements, tools used, and troubleshooting steps. This would solve the question of how the information and results could be transferred to clients or other teams in the future.

Another important direction is to test Gaussian Splatting directly inside the Pixel Playground. Loading a Gaussian Splat in that environment would be a first-time attempt, and it would show how these scans behave in a proper virtual production setup. It would also help evaluate performance, lighting behaviour, and whether the workflow scales in a studio-level context.

Overall, this whole process helped me understand how to justify technical decisions more clearly, how to respond to client expectations, and how to use feedback to redirect the project toward better tools and outcomes. Instead of just improving the photogrammetry workflow, the feedback pushed me to rethink the entire pipeline and adopt a technology that aligns better with real-time rendering and virtual production goals.

“How to” video: howto-v1.mp4

Cinematics video: loop-cinematic.mp4