4. Iterations on scans

Intro

During our first weeks of planning, we had to look into what technologies to use to transform the pictures we were taking into 3D assets. I started exploring Polycam a bit, but to be fair it didn't provide much to work with unless I purchased a subscription, so the main objective became finding completely free alternatives. After taking a good look at the client's presentation, I ended up picking Meshroom since it seemed like a good option, being a free, open-source 3D reconstruction software based on the AliceVision framework.

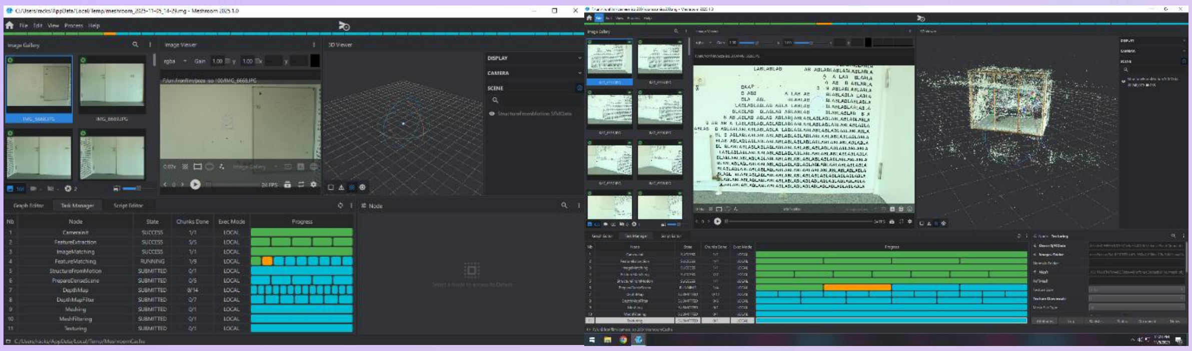
So I then researched how to actually take photogrammetry pictures. With my homework done, it was time to find a room. I thought it would be a great idea to start with a small room, finding my first target in the R3-4 building on campus. I took 166 pictures on fixed camera settings with two different ISO settings but keeping the rest the same. I wanted to test 100 ISO vs 200 ISO and also test different settings in Meshroom.

Report:

Defining the workflow + tests

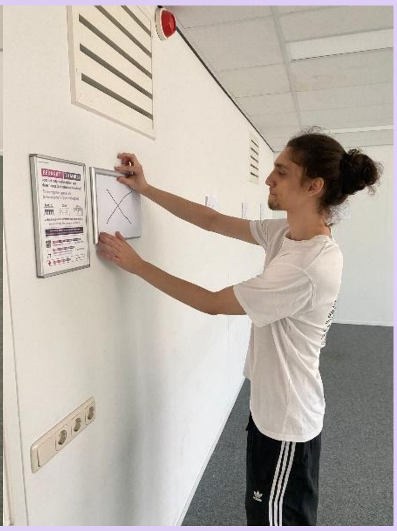
Process & Feedback

One of the main issues I had no idea I was going to encounter was the render time. It went from 4–5 hours to even 8–9 hours depending on what settings I was using. The most demanding parts were Feature extraction (where it depended on the Describer Density and Quality), the downscale in DepthMap, and the Input Points and Max Points in Meshing. Each of these processes alone could take several hours, while others would go as fast as a few minutes.



Nb	Node	State	Chunks Done	Exec Mode
1	CameraInit	SUCCESS	1/1	LOCAL
2	FeatureExtraction	SUCCESS	5/5	LOCAL
3	ImageMatching	SUCCESS	1/1	LOCAL
4	FeatureMatching	RUNNING	1/9	LOCAL
5	StructureFromMotion	SUBMITTED	0/1	LOCAL
6	PrepareDenseScene	SUBMITTED	0/5	LOCAL
7	DepthMap	SUBMITTED	0/14	LOCAL
8	DepthMapFilter	SUBMITTED	0/7	LOCAL
9	Meshing	SUBMITTED	0/1	LOCAL
10	MeshFiltering	SUBMITTED	0/1	LOCAL
11	Texturing	SUBMITTED	0/1	LOCAL

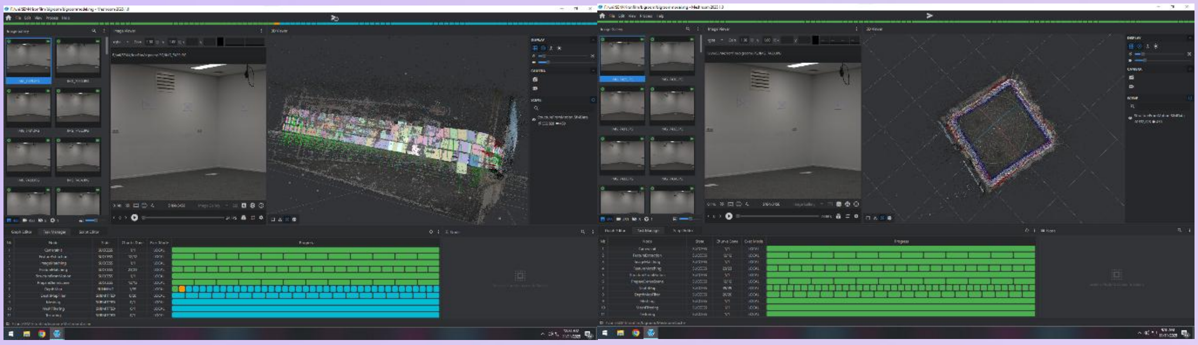
The lack of technique in taking pictures and the lack of a tripod to keep the camera more stable was easy to spot in the results. The model had big holes, and the big glass window in the room was a complete mess. So, it was time to search for another place. This time we found a room without glass walls, but it was quite big, almost 7x7 meters, which meant a lot more pictures.

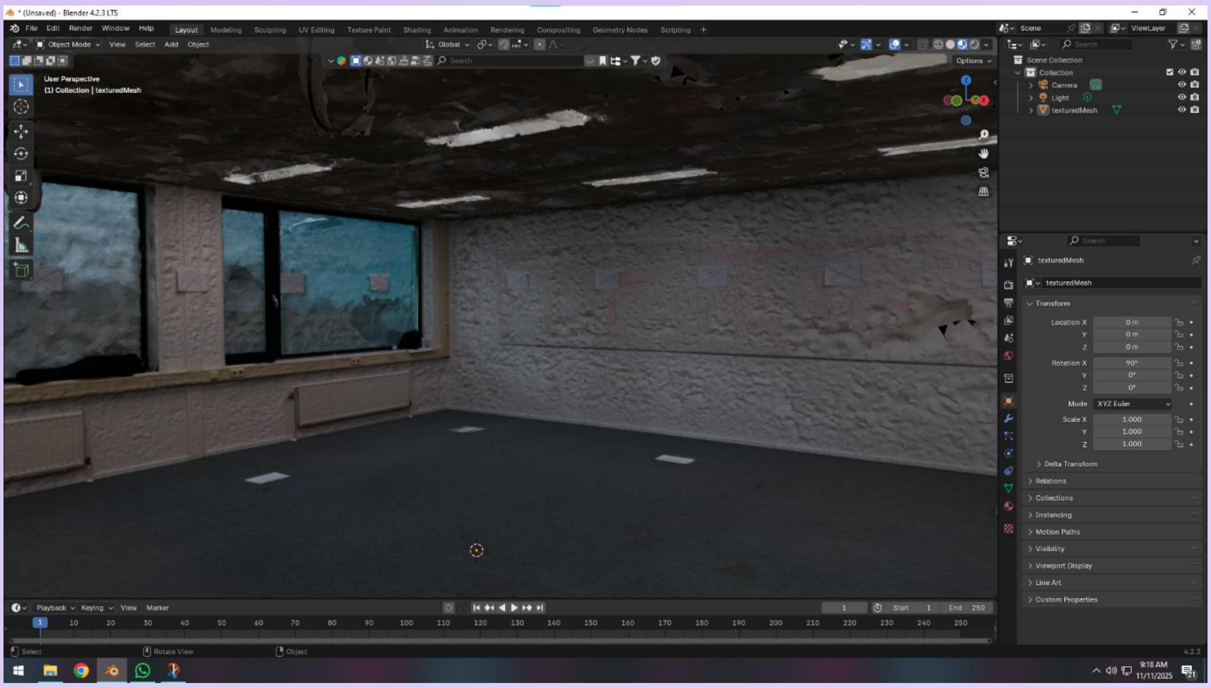


Following the orbital technique more carefully, I did two full perimeter passes plus ceiling and floor, resulting in four orbits around the room. This time I used LED lights, covered the windows, and added markers on walls so the software had help distinguishing wall/floor/ceiling. The results after Meshroom processing were already much better, with a really clear floor, but the issue remained with the white walls, which came out with a ton of noise. We ran another scan with higher ISO on daylight and pushed the Meshroom settings to the max, just to find out that the result came out even worse since the software was just pushing a bad data set as far as possible.

Report:

[Natural light vs LED light tests - document]



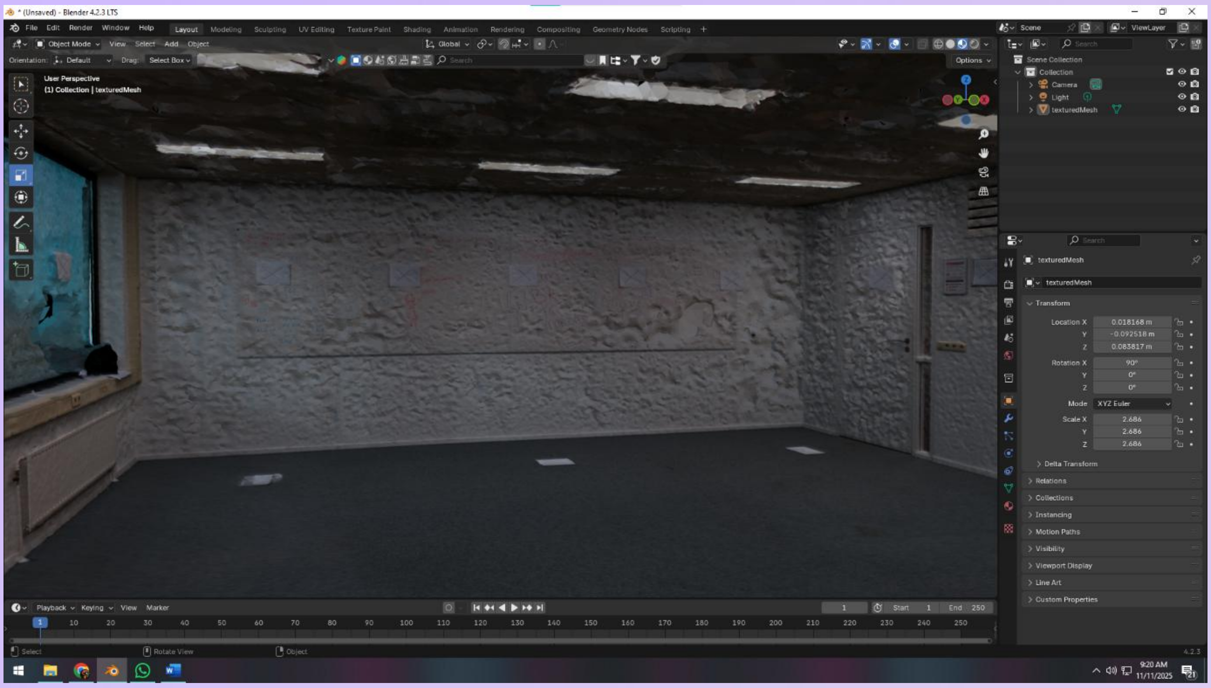

* (Unsaved) - Blender 4.2.3 LTS

User Perspective
(1) Collection | texturedMesh

Scene Collection
Collection
Camera
Light
texturedMesh

texturedMesh
Transform
Location X 0 m
Y 0 m
Z 0 m
Rotation X 90°
Y 0°
Z 0°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000

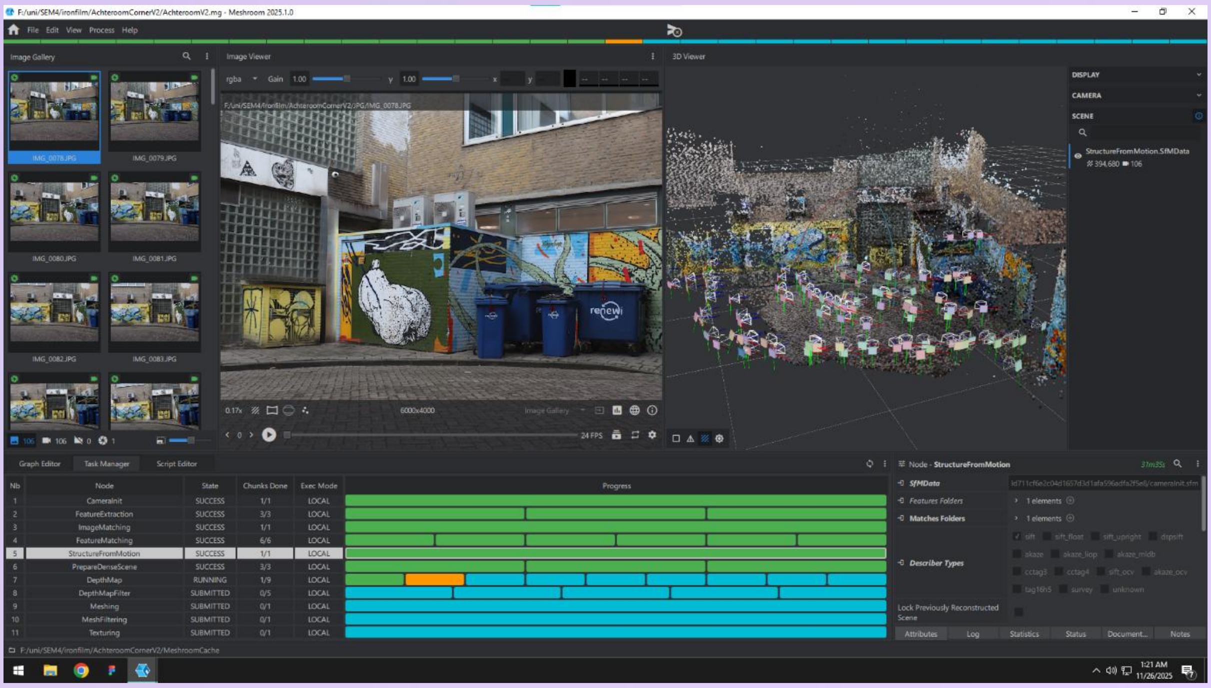
File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Geometry Nodes Scripting
Object Mode View Select Add Object
Global
Options
Delta Transform
Relations
Collections
Instancing
Motion Paths
Visibility
Viewport Display
Line Art
Custom Properties
Playback Keying View Marker
Start 1 End 250
Select Rotate View Object
9:18 AM 11/11/2025


texturedMesh

Transform

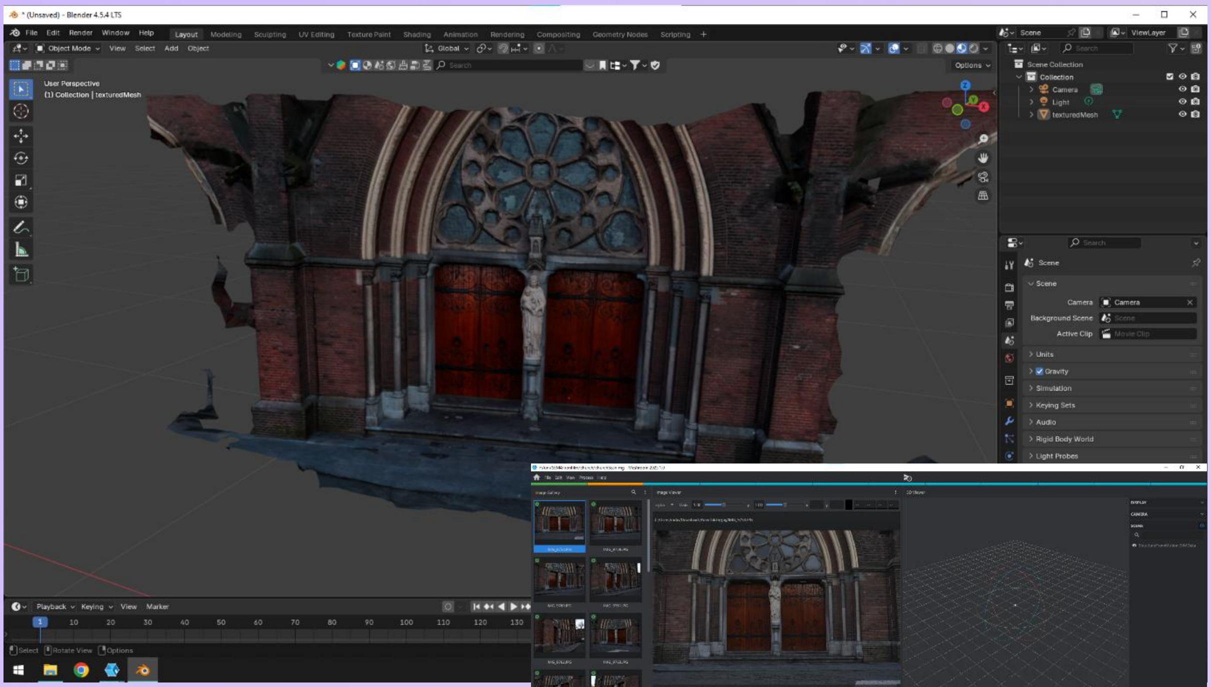
Location X	0.018168 m
Y	-0.092518 m
Z	0.083817 m
Rotation X	90°
Y	0°
Z	0°
Mode	XYZ Euler
Scale X	2.686
Y	2.686
Z	2.686

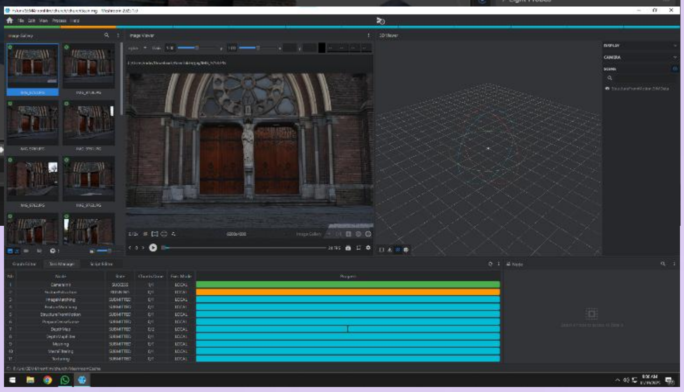
After presenting all our findings to the client and getting feedback from teachers, it was time to look for other spaces with more texture and more color. At first, since I was already used to the long waiting times, I wanted to input a much smaller data set — this time around 20 pictures, for faster tests. Things improved immediately. The new scans of the Eindhoven Center Church, Achterom Street or Bibliotheek Café showed a huge improvement in quality with high settings on Describer Density and Quality while keeping the rest on medium.


F:/uni/SEM4/ironfilm/AchteroomCornerV2/AchteroomV2.mg - Meshroom 2025.1.0

Nb	Node	State	Chunks Done	Exec Mode
1	CameraInit	SUCCESS	1/1	LOCAL
2	FeatureExtraction	SUCCESS	3/3	LOCAL
3	ImageMatching	SUCCESS	1/1	LOCAL
4	FeatureMatching	SUCCESS	6/6	LOCAL
5	StructureFromMotion	SUCCESS	1/1	LOCAL
6	PrepareDenseScene	SUCCESS	3/3	LOCAL
7	DepthMap	RUNNING	1/9	LOCAL
8	DepthMapFilter	SUBMITTED	0/5	LOCAL
9	Meshing	SUBMITTED	0/1	LOCAL
10	MeshFiltering	SUBMITTED	0/1	LOCAL
11	Texturing	SUBMITTED	0/1	LOCAL

F:/uni/SEM4/ironfilm/AchteroomCornerV2/MeshroomCache


* (Unsaved) - Blender 4.5.4 LTS
File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Geometry Nodes Scripting
Object Mode View Select Add Object
User Perspective
(1) Collection | texturedMesh
Scene Collection
Collection
Camera
Light
texturedMesh
Scene
Camera
Background Scene
Active Clip
Units
Gravity
Simulation
Keying Sets
Audio
Rigid Body World
Light Probes
Playback Keying View Marker



Examples of settings used:

Achterom street corner

- featureExtraction: sift, high, high
- DepthMap: downscale 1, nb neighbors 12
- depthMap filter: min 3, min bad 4
- meshing: max points 6,000,000, max input 50,000,000
- meshFiltering: smoothing iterations 3

Church

- Describer Types: sift
- Describer Density: high
- Describer Quality: high
- Max Input Points: 40,000,000
- Max Points: 4,000,000

The quality of the new scans was validated even more during the prototype day with a lot of positive feedback and appreciation for how much the results improved across these iterations.

Reflection

At first, I assumed the software should handle most of the work, like RealityCapture does, where even with a weaker data set the results can still look strong. But through all these iterations in Meshroom I started to understand how much the input photos determine the outcome, and that pushing the settings higher does not fix a bad data set, it makes the flaws even more visible.

Each iteration taught me something concrete: the first scan showed the impact of unstable photos, the second scan showed how lighting and white walls affect noise, and the next scans proved how textures, color variation and better technique improve the reconstruction drastically. These connected steps made it clear why each decision mattered and how each round of feedback changed the next attempt.

Overall, this iterative process helped me develop a more methodical way of testing: adjusting one variable at a time, documenting the changes, comparing results, and using both technical findings and client feedback to improve the next version. It made the whole workflow more intentional instead of just trial and error.

Now it was time to learn how to properly take pictures for Gaussian Splatting

Research:

Gaussian Photo methods research - document