CS11: 3D Scanning of Large Spaces

1/29/2021 Meeting

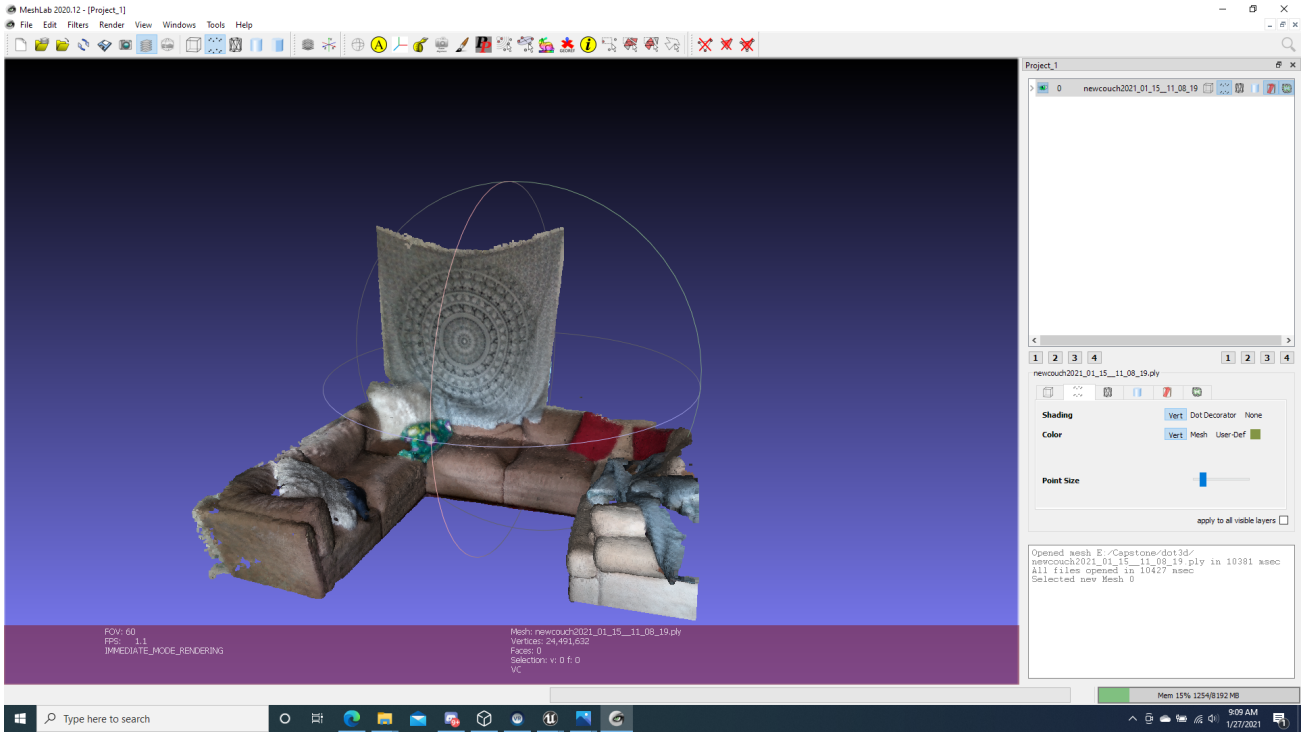
Brandon Withington, Seika Mahmud, Casey Boomer

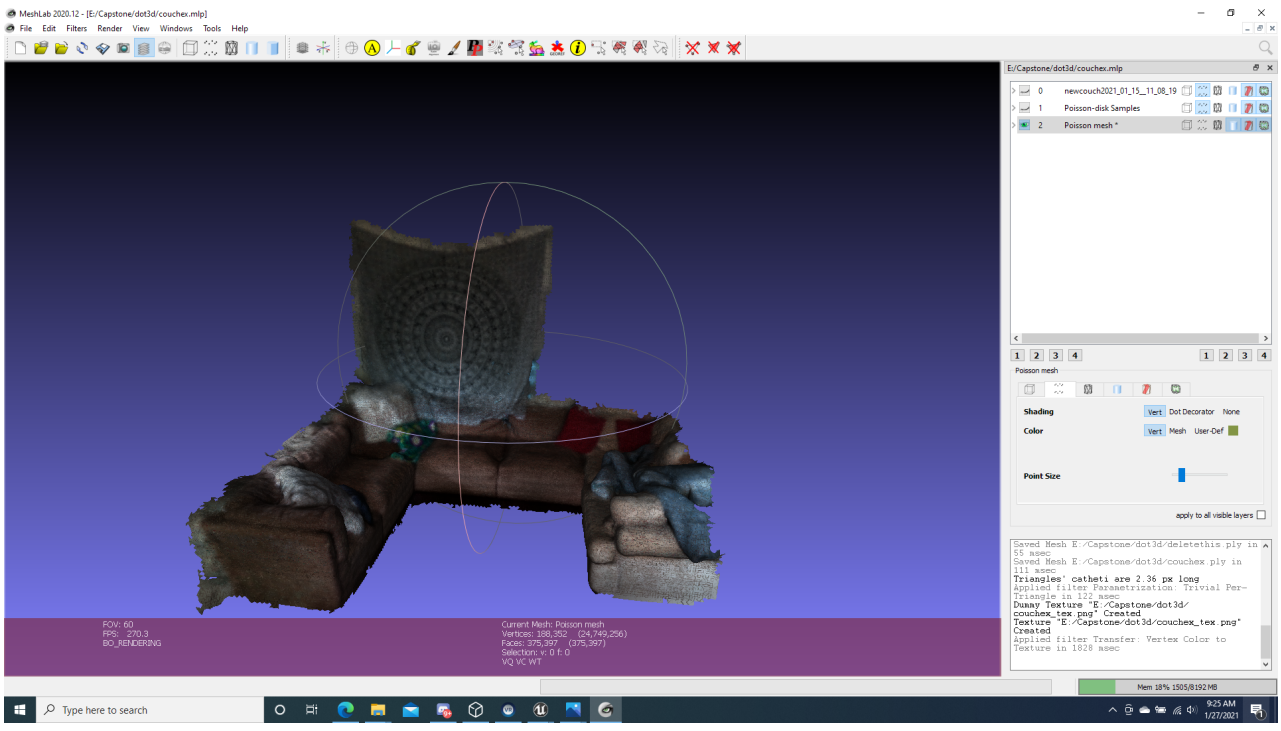
Cleaning up Scans :
How-To: A Simple Way to 3D Scan an Environment – Intel® RealSense™ Depth and Tracking Cameras

This tutorial shows how to scan a point cloud with an Intel RealSense camera using Dot3D Pro and convert it into a mesh with Meshlab. These instructions worked for the most part, but using ball pivoting surface reconstruction resulted in a “spiky,” disjointed mesh.

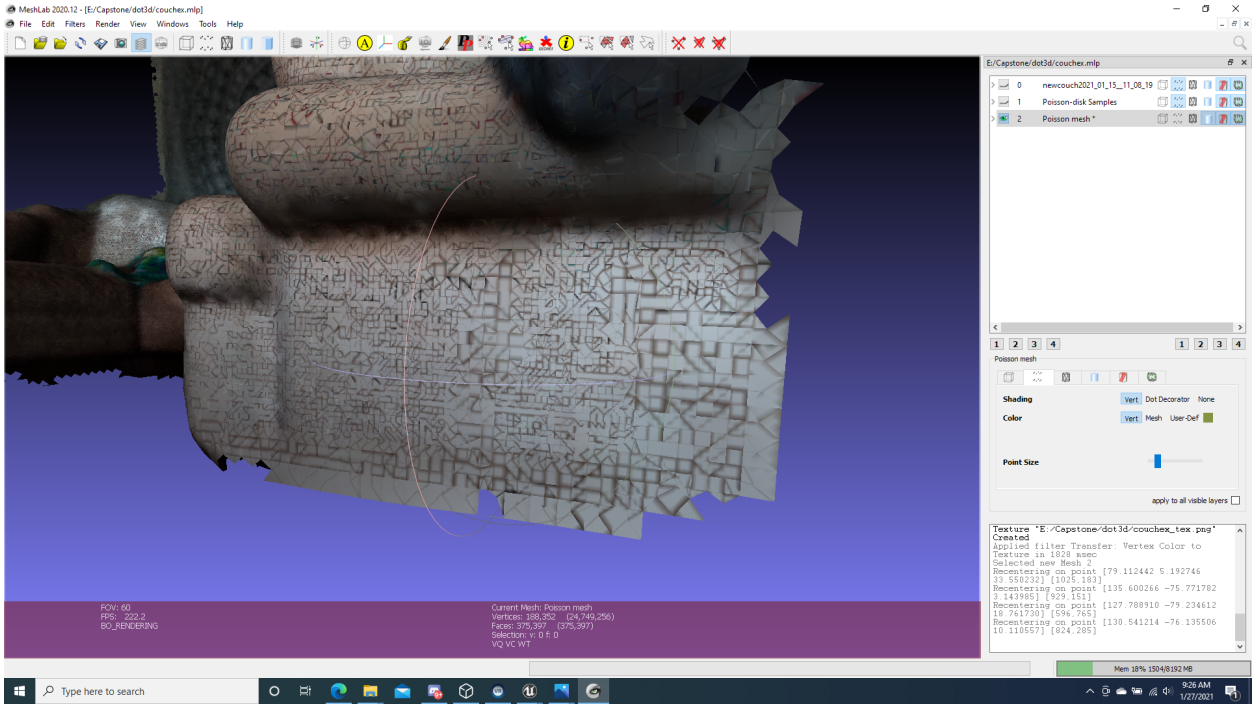
Steps to create textured mesh from point cloud using Meshlab · GitHub

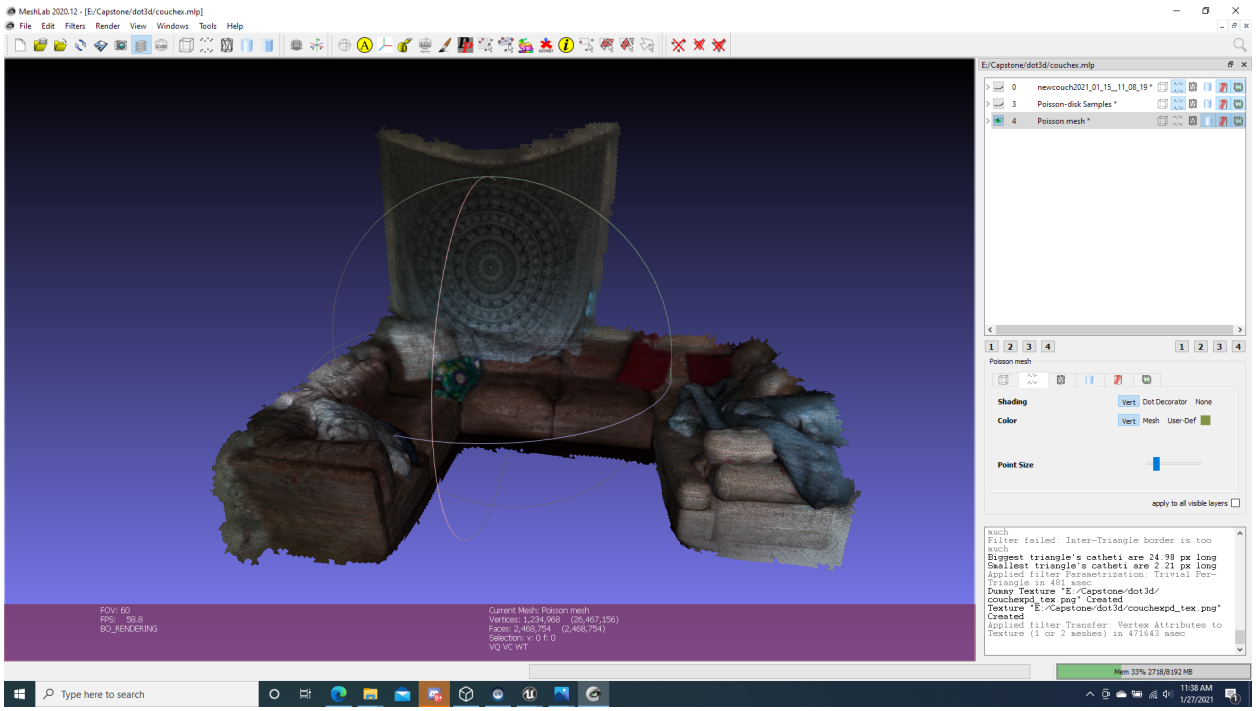
This tutorial uses screened poisson reconstruction to create a smooth, watertight mesh. While this type of mesh works better for our purposes, it also tends to lose some detail and requires more processing to generate and assign a texture. The textures are somewhat obscured by the outlines of the faces.



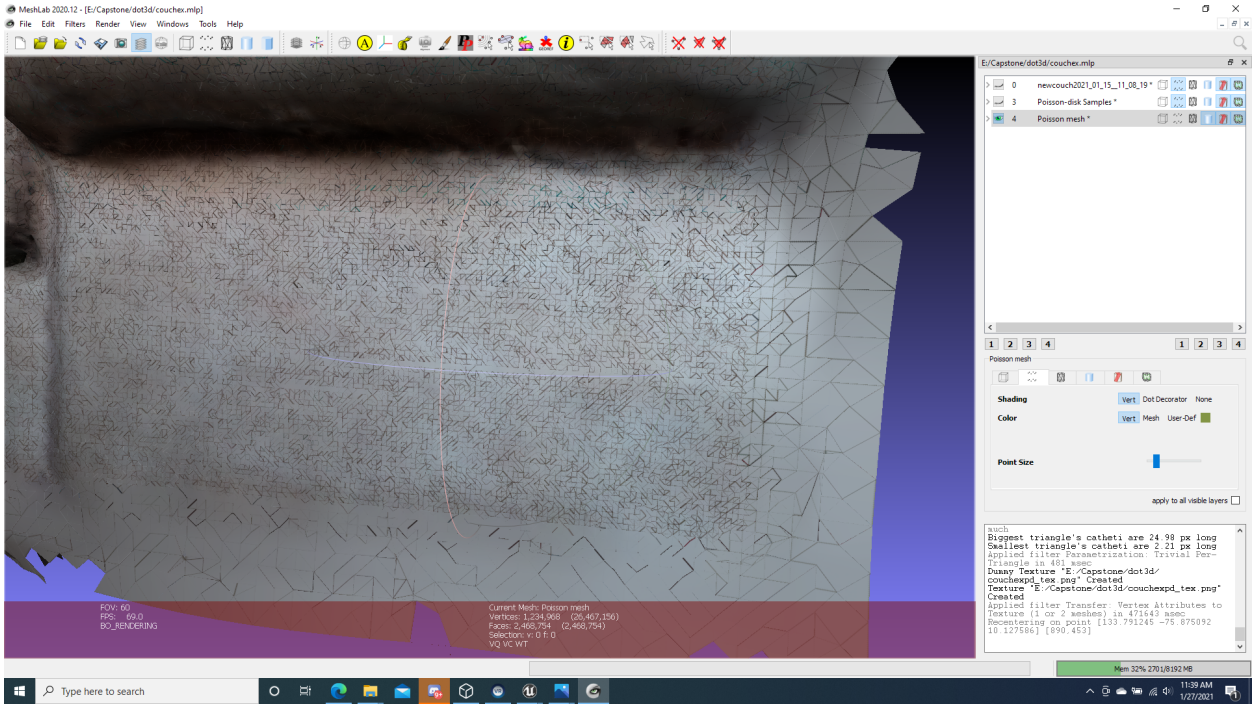


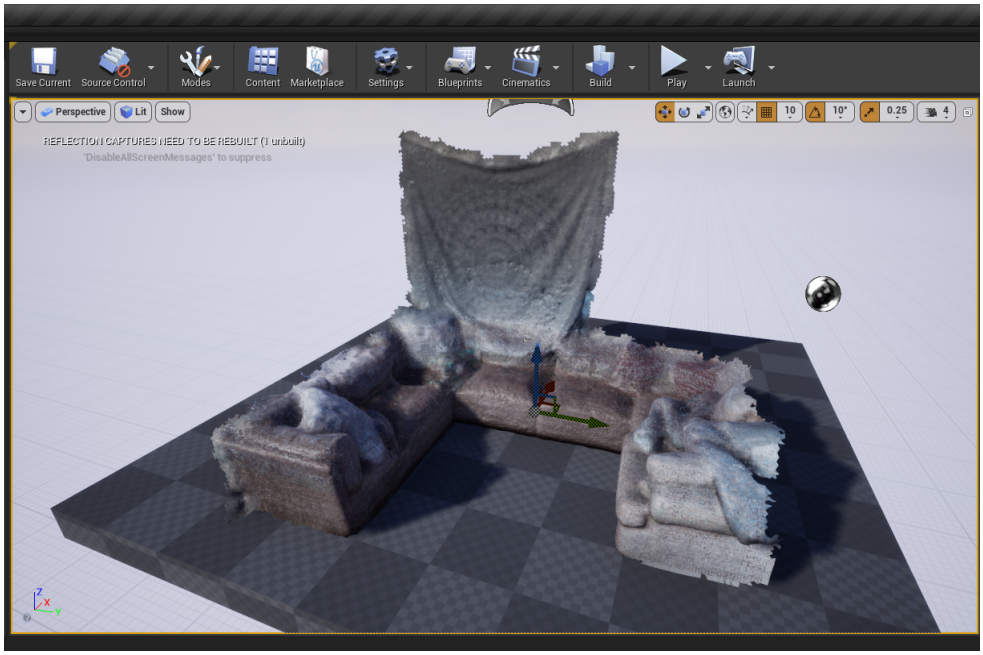
Textured poisson mesh: poisson-disk sampling with 100 thousand samples





Textured poisson mesh: poisson-disk sampling with 1 million samples

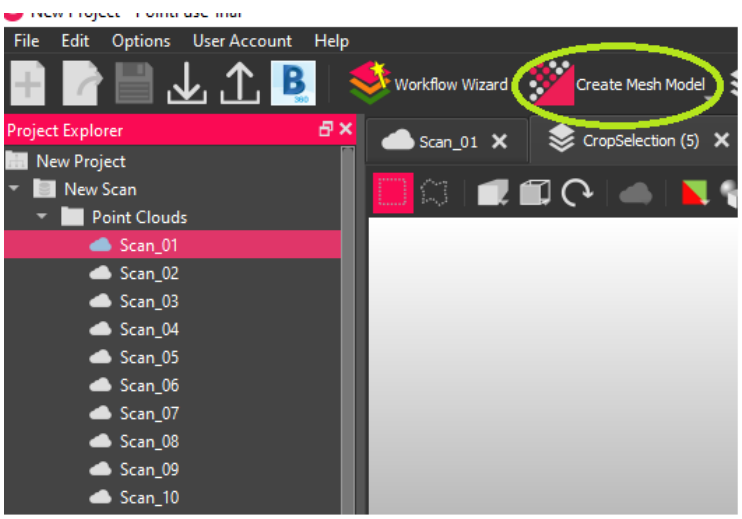


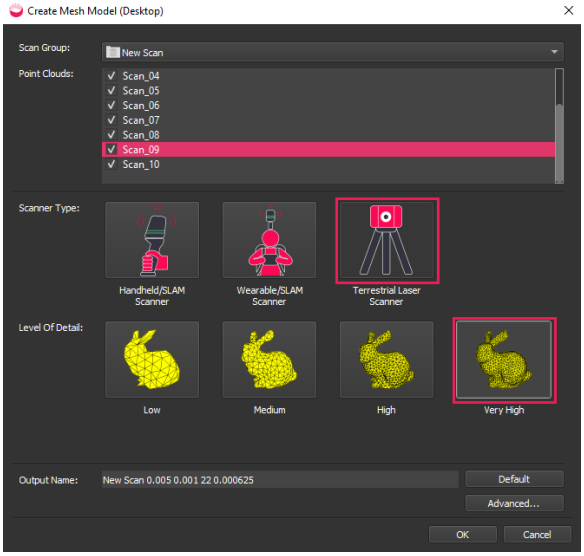


Utilizing PointFuse :

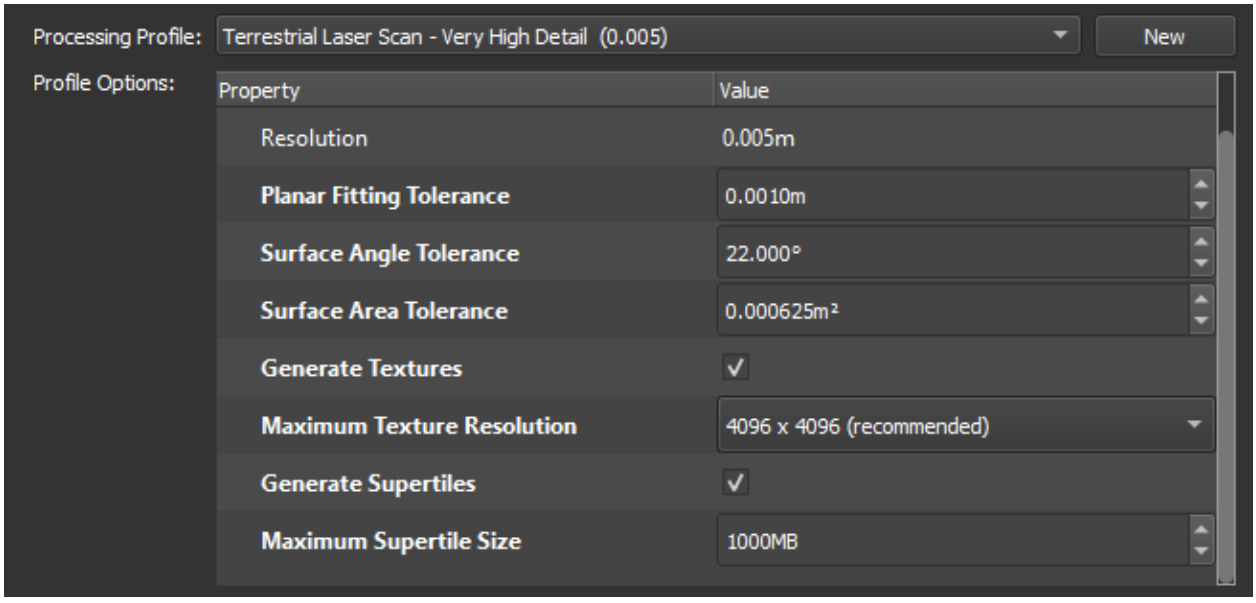
[Pointfuse for DP (dotproduct3d.com)]

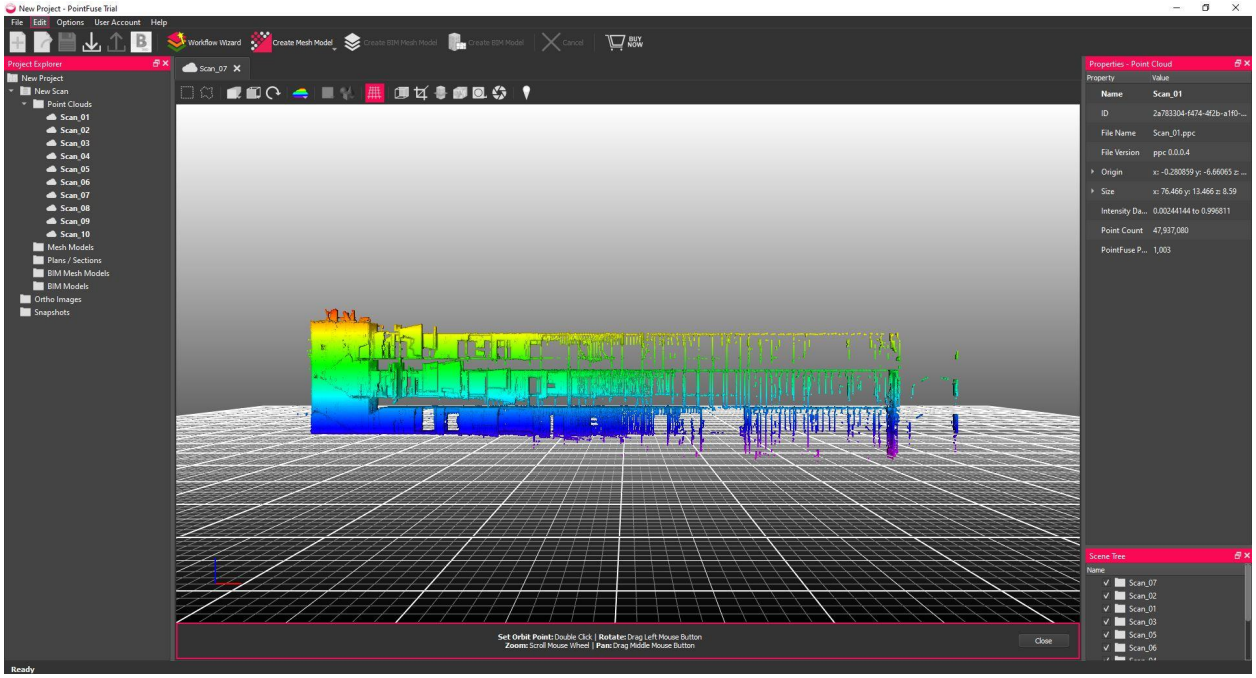
PointFuse is another mesh creation program that DotProduct is compatible with. It operates similar to meshlab in the way you can create meshes by importing LAS point clouds. With this software, we imported the example forestry scans given to us by Rafaelle. They have a very clear method of creating meshes from imported point cloud scans.



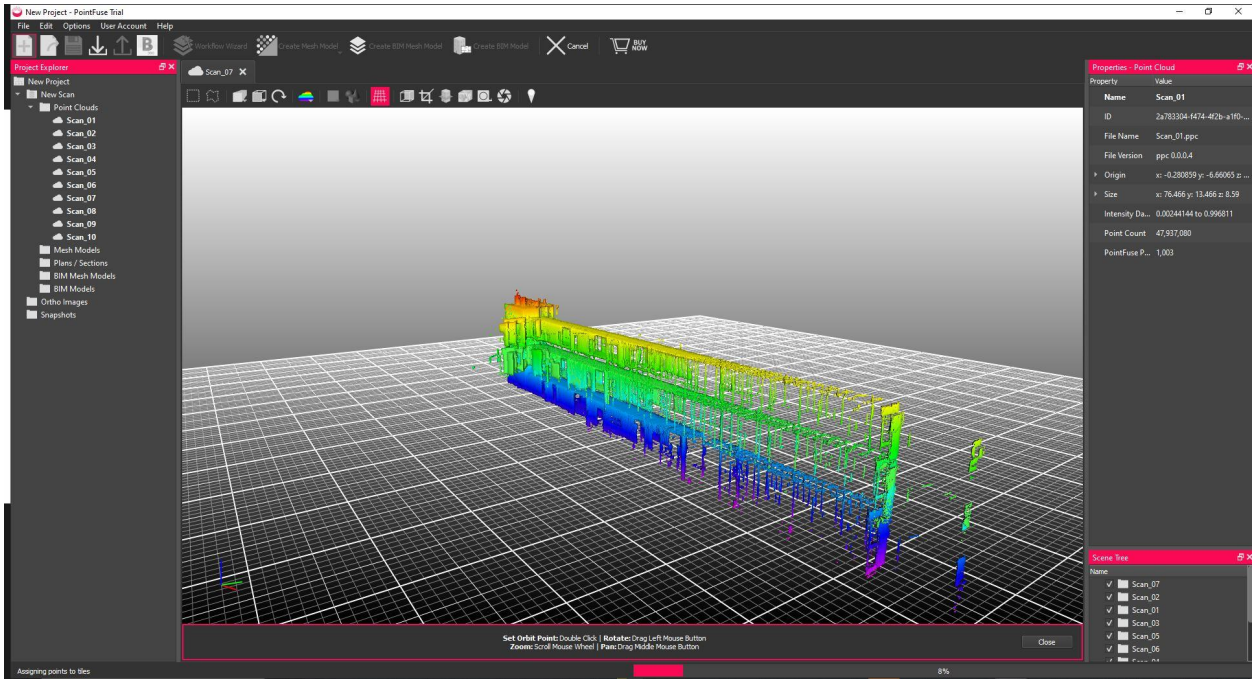


After importing the example scans, PointFuse has a very handy and very intuitive process for creating a mesh model from point clouds. If you want to change manuals of the mesh directly you have the option of going into the advanced menu :

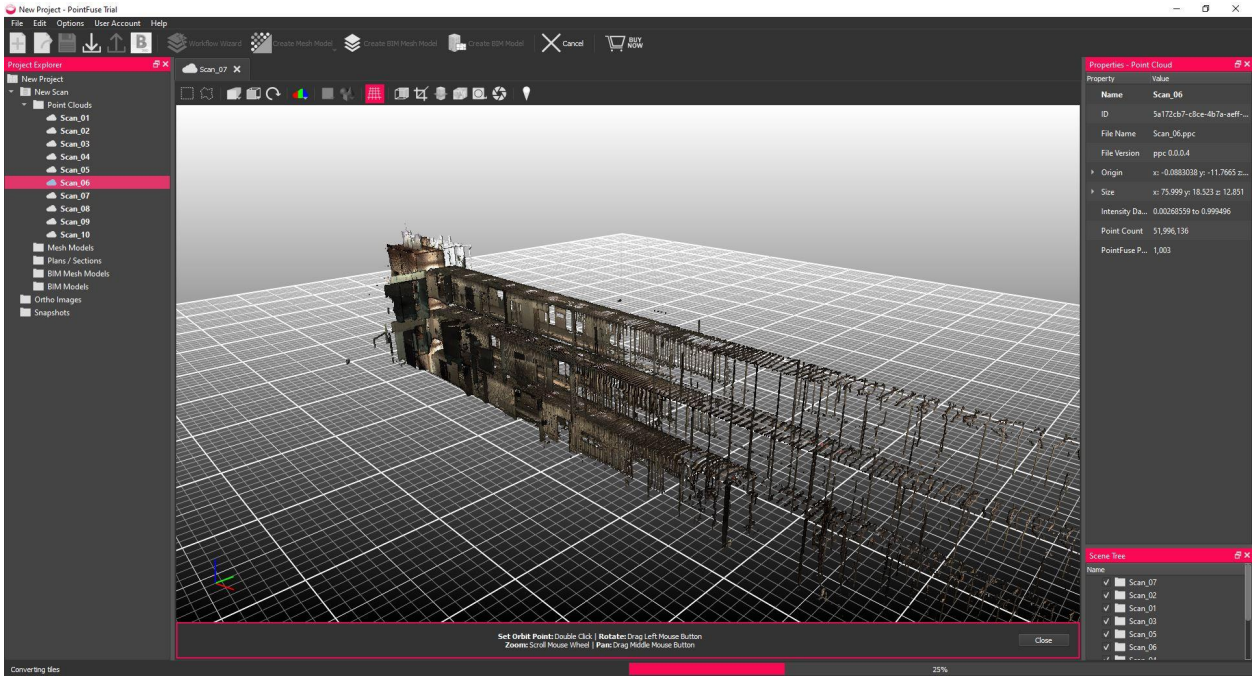




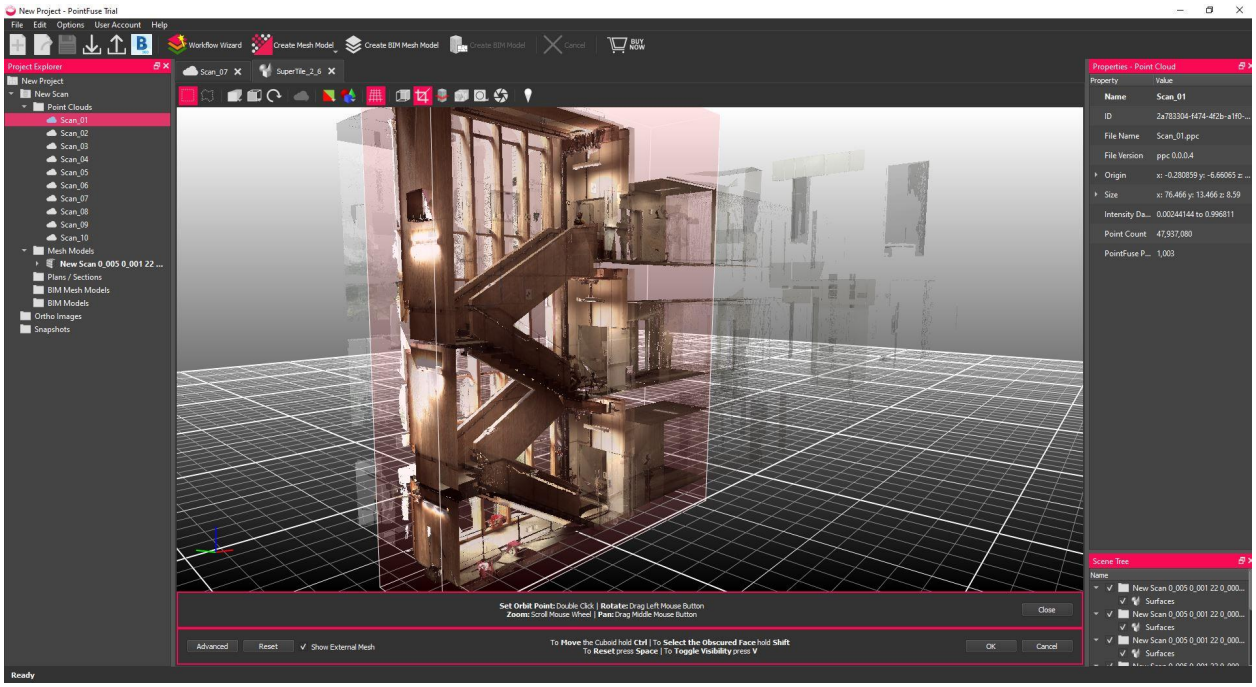
Importing each set of cloudpoint scans into PointFuse. This is the height map filter which colors the scan according to the height in relation to the top and bottom of the scanned object.



Secondary view.

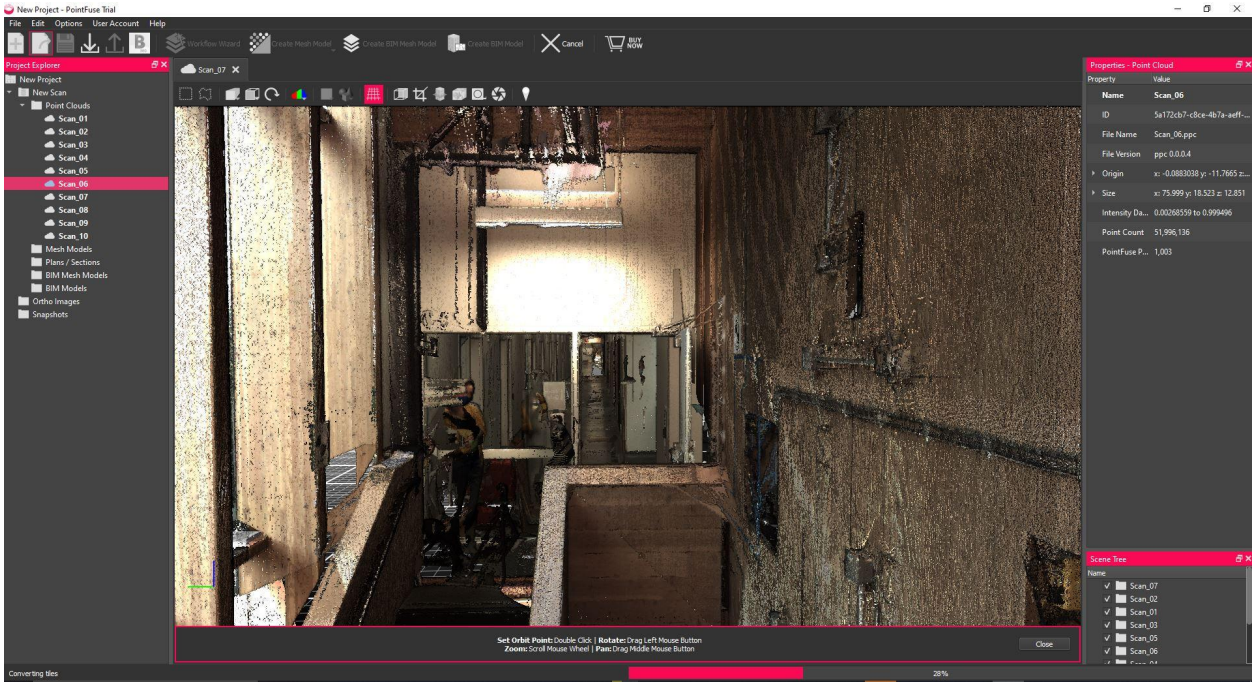


Colored view of the imported scan. Note that there are lots of points that stretch outward to the very right of the object. These points can be filtered out using the crop utility tool found in PointFuse.

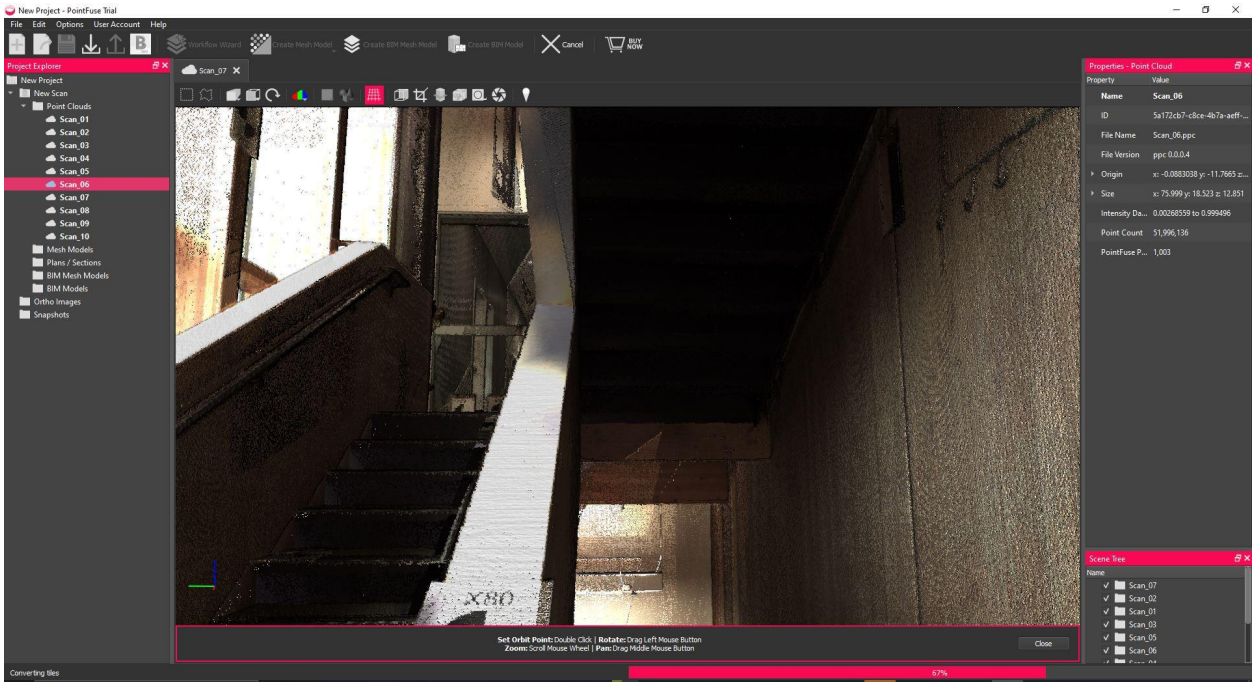


Applying crop to the forestry scan in PointFuse.

Here we can see how accurate and high quality the given scan is. The interior of the building is captured and can clearly be seen.

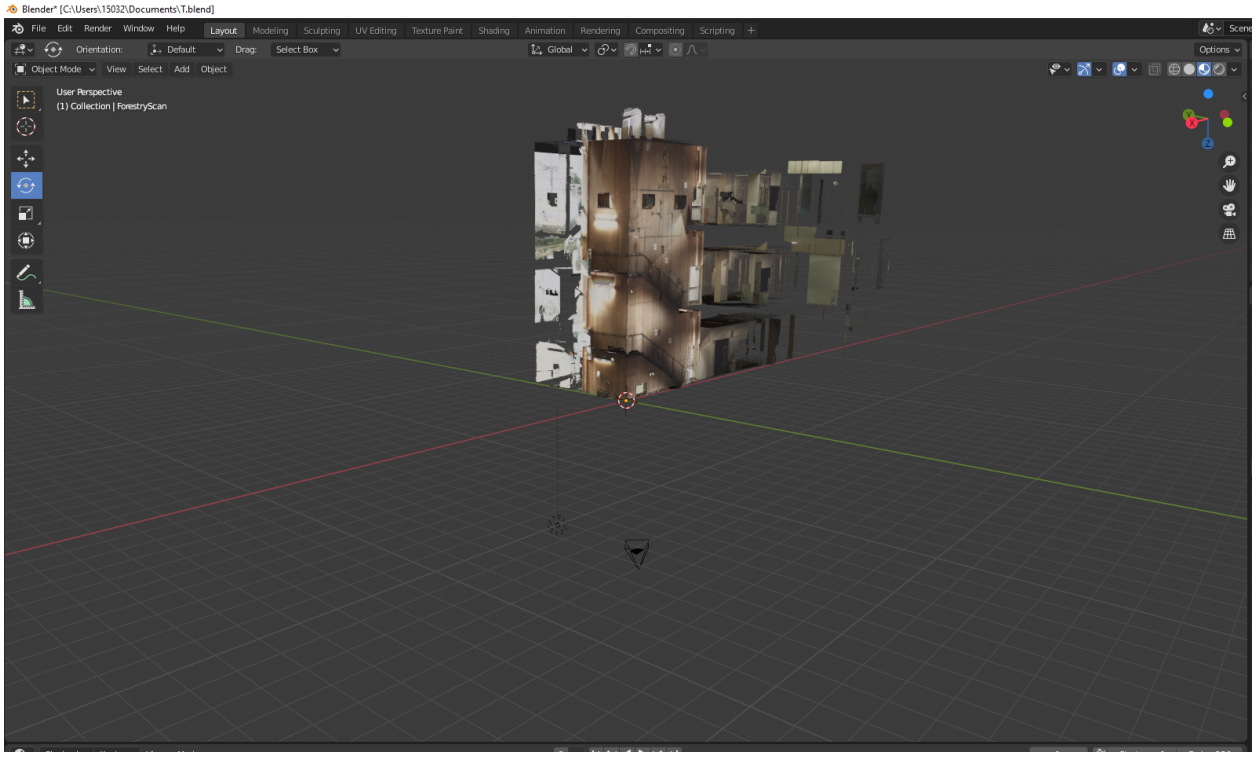


Interior of the scan in PointFuse

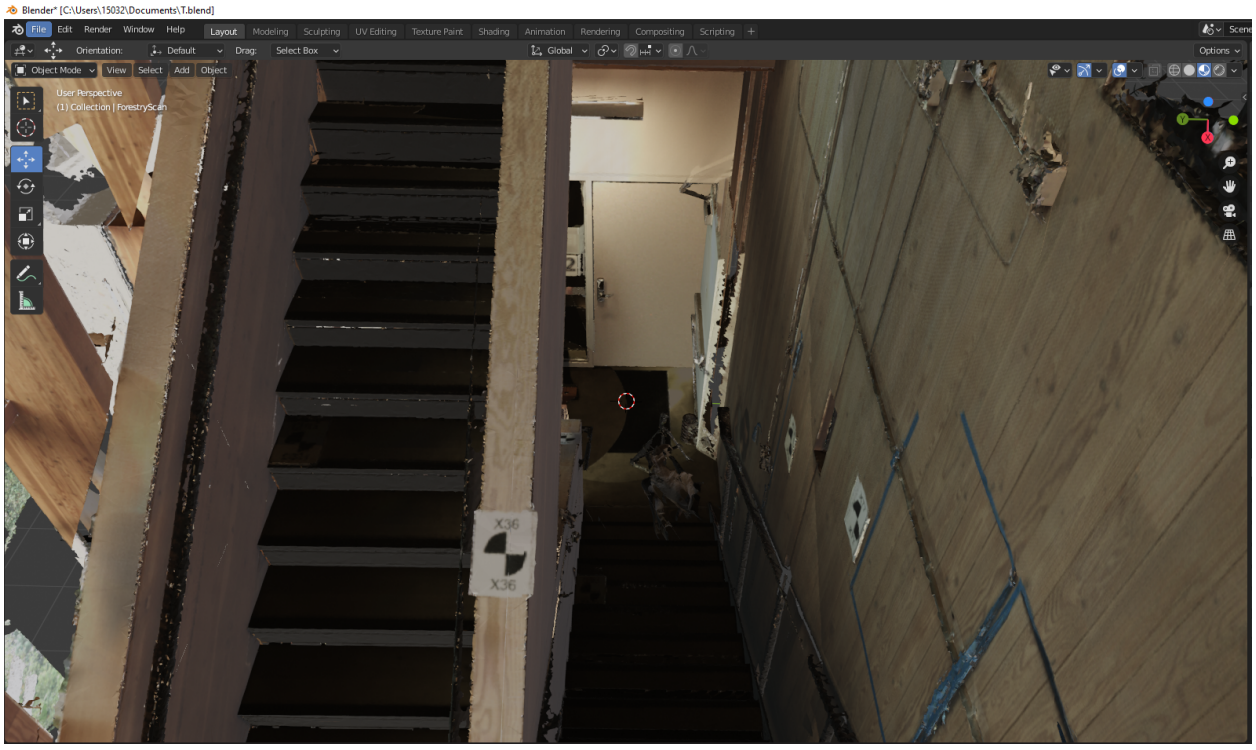


Secondary view on a floor below.

The free trial of PointFuse, in combination with it being timed, also restricts us to three free exports. We exported this scan into an OBJ file to then pass into Blender.



Exported into Blender



Interior view of the scan within Blender.

Videos used to import the LAS point cloud into PointFuse :

Pointfuse Tutorial 1 Importing and converting a point cloud - YouTube
Pointufse Tutorial 2 Classifying and exporting 3D Mesh Models mepx - YouTube
Pointfuse Tutorial 3 - Changing Parameters - YouTube
Pointfuse Quick Tips - Using the cropping tool - YouTube

Pros and cons of PointFuse :

- Pros :
    - Imports large LAS pointclouds very quickly compared to MeshLab.
    - Automatically combines separate but related pointclouds together, making the creation of meshes very easy.
    - Assigns the texture to the exported OBJ file automatically whereas with MeshLab that has to be done manually.
- Cons :
    - In addition to being on a timed free trial, it also restricts the amount of exports you can do to three.
    - Some of their tutorial videos are not up to date with the modern version of the product.

Importing to Unreal Engine :

There are still issues with exporting from blender and importing into the Unreal Engine.
The scan looks great in Blender, the interior of the scan has been retained and it looks pretty detailed. However, when you import the FBX from Blender into Unreal the result is very choppy. this :

Presentation Feedback :

This Friday, we will present our progress in our Capstone class. This 20-minute presentation will explain how our project has changed form milestone 2 to today. There will be time afterwards for questions and feedback from the other groups. We will present that right here:

- There wasn't very much feedback, but many people said they were impressed with the work we were doing and that it seemed like there was a lot of research involved
- It is hard to compare our progress versus the other projects we watched because most of the other projects were either started by a capstone group last year or are continuing to a capstone group next year. However, They seem to be at a similar level to us in terms of work that they have done.

Next sprint Planning :

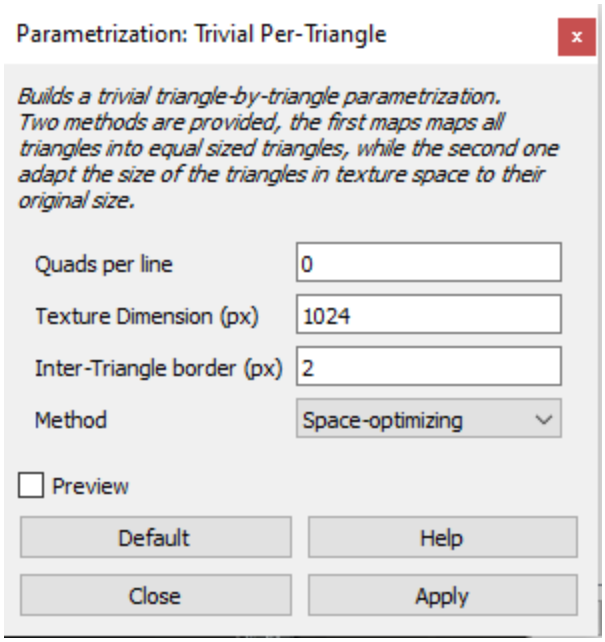
Though not perfect, the pipeline for creating a mesh from Dot3D is becoming more realized, and it is becoming time to find the next steps for our project. Here are a list of tasks that will be completed for the next sprint:

- Providing documentation for our pipeline
- Making the Pipeline User Friendly
    - Currently, the pipeline requires the user to change between multiple applications to finally walkthrough the scan in Unreal. We will come up with solutions to make this portion of the software more practical for a regular user
    - One solution is to create an Unreal plugin between PointFuse and Unreal, that automatically imports the mesh into an unreal file, and modifies some settings such as adding collision and adding a camera.
    - Another is to have some sort of api that can send the point cloud directly to MeshLab. This could be some kind of C++ code that has access to libraries that direct the newly created point cloud to meshlab. Ultimately, we can prompt the user to make any modifications that are required.
    - pointcloudplugin
- Create a design for the user interface
    - Before we can start programming the functionality of a user interface, we would have to create a design that is efficient for the user to operate. This could be inspired by DotProduct's simple layout, with added functionality for the stages after creating the point cloud.
    - Look into displaying the point cloud directly into our user interface. There are tools such as Qt3D that could work with some additional research
- Scanning the forestry lab
    - Schedule a time to officially scan the forestry lab, now that we have had more progress on creating more practical scans.
- Compile licenses that are used for this software
    - Now that we have a pretty solid grasp of the software components we will use, we need to look at what components require licenses to use.
    - Figure out the logistics of a person using this software: What kind of licenses they will need and how much will it cost. Is there a way to have the user subscribe to all of the licenses in one easy purchase?

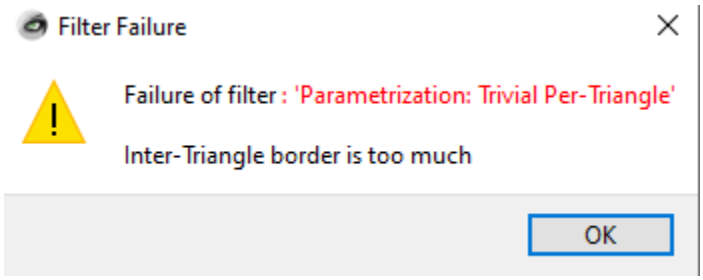
Tests with self-scanned environment and Meshlab :

# of Vertices in Scan	# of faces generated	Able to set parameterization per triangle	Can texture colors be mapped	Face creation method used	Texture outcome
54,718	101,729	Yes	Yes	Ball Pivot	Spikey and inaccurate. Texture is not applied to the scan and cannot be viewed in other programs.
54,718	126,602	Yes	Yes	Poisson Reconstruction	Textures are accurately mapped onto the 3d model however texture is blurry.
389,175	715,393	Yes (Takes a very long time)	Yes	Ball Pivot	The same outcome as the first Ball pivot : spikey and inaccurate texture mapping.
1,279,652	2,186,398	No	No	None	No outcome.
2,215,564	3,885,134	No	No	None	No outcome.
61,688,364	Meshlab Crashes before faces can be made.	No	No	None	No outcome.

- Ball pivot surface reconstruction starts with the normals of the point cloud and uses a triangle based algorithm to reconstruct the desecrated scan.
- Screen Poisson surface reconstruction creates “watertight” surfaces from the original point set. This generally tends to create smoother surfaces and textures.



This sets up the texture for the 3d object. It is based solely on how many triangles there are between points within the point cloud that were generated from either of the surface reconstruction methods used. If there are too many faces then this issue comes up :



Our resource : [How-To: A Simple Way to 3D Scan an Environment – Intel® RealSense™ Depth and Tracking Cameras]

Tells us to set the inter-triangle border to 1px. However, in the case of larger scans it will not run. Changing the texture dimension by factors of two does allow a texture to be mapped to the 3d object meshlab is generating. However, It is not accurate texture generation. The outcome :