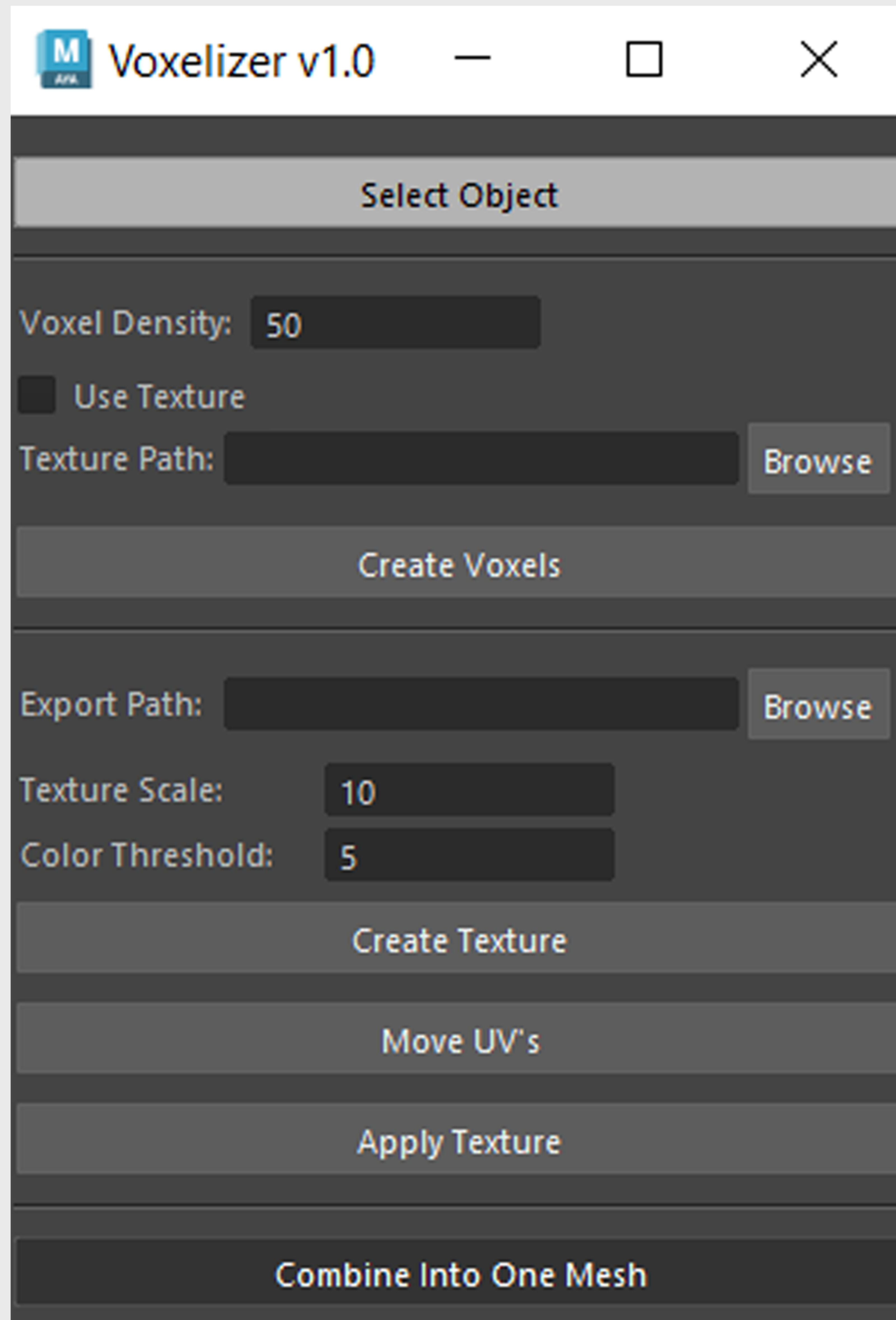


Voxelizer

Pablo Debusschere



Selection

1. Model Creation

2. Texture Creation

3. Merge

Important:

This tool uses Pillow, a fork of the Python Image Library (PIL), to create and save images. PIL must be installed on your device for the script to work. To install it, go to:

<https://pillow.readthedocs.io/en/stable/installation.html>

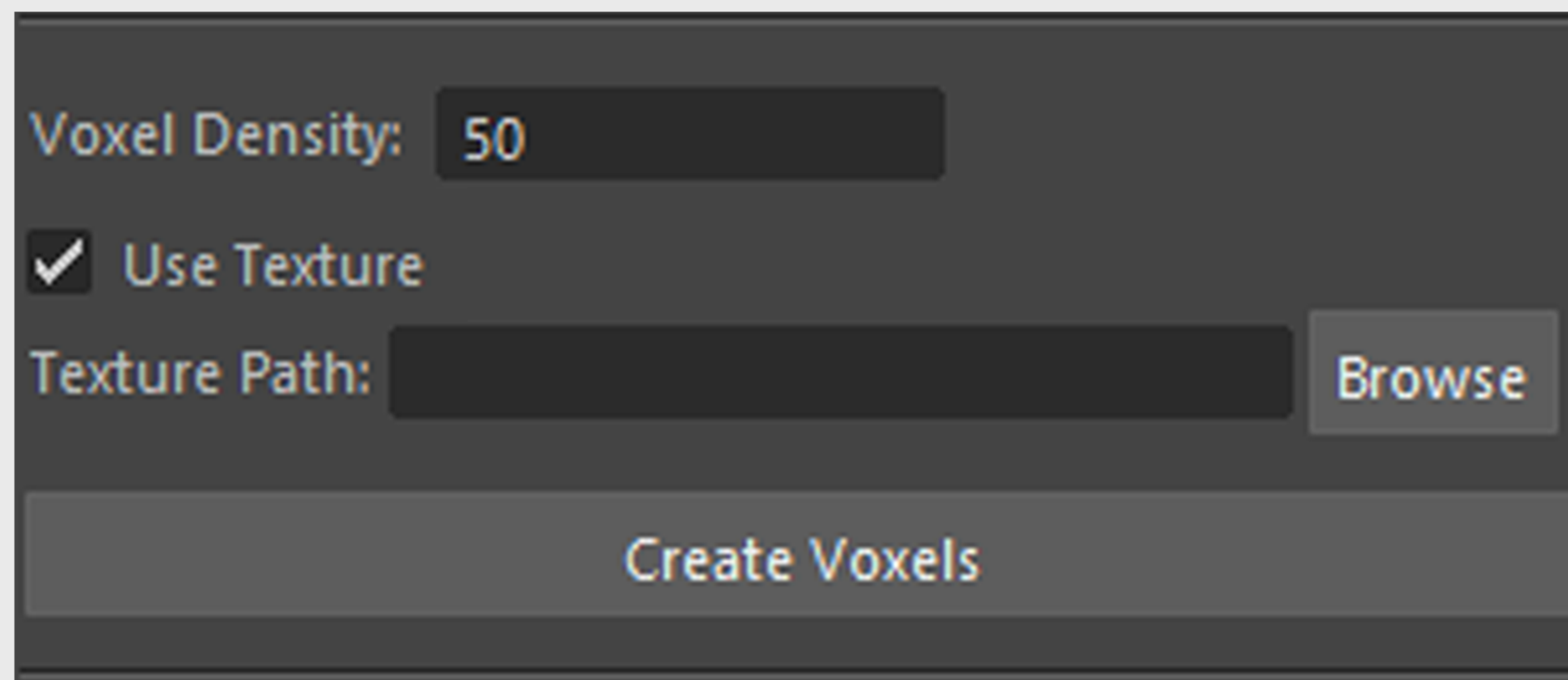


The script is comprised of 3 steps:

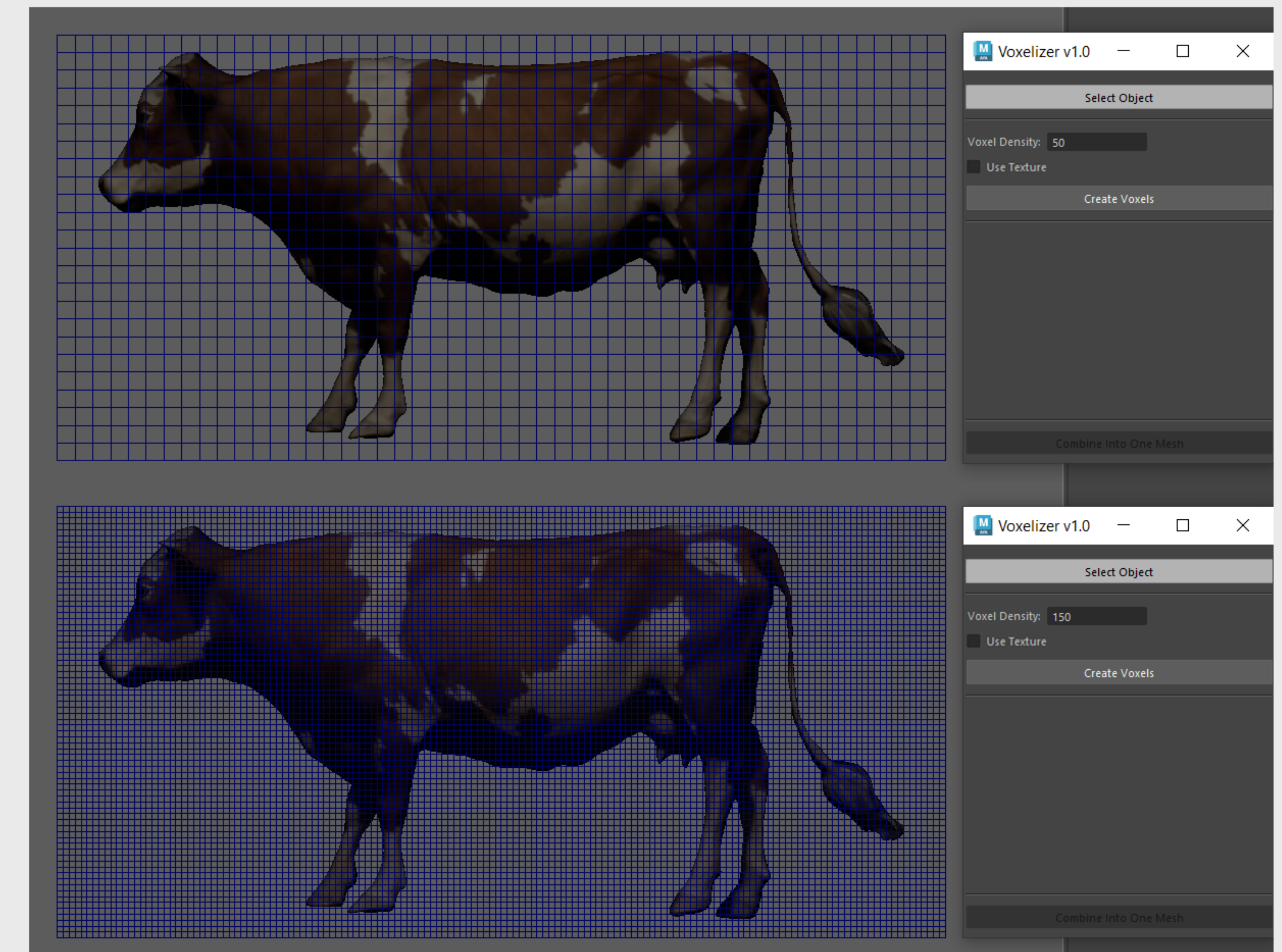
- First, the user must select the object they wish to voxelize at the top. Only one mesh can be processed at a time, and only one texture can be used. If an object has several textures/materials, the object can be split in several meshes, each with their unique texture. Each mesh can then be voxelized separately, and then merged together afterwards.

model used in examples: 'Cow' by alitural on Sketchfab

1. Model Creation



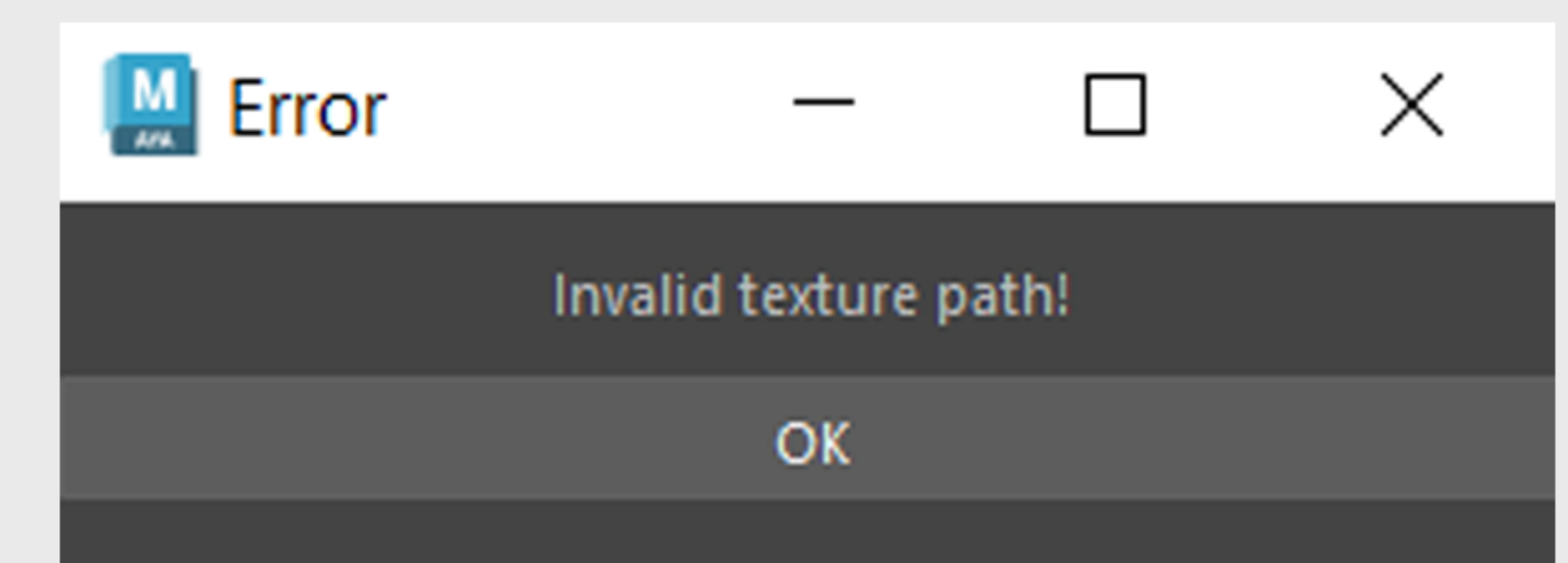
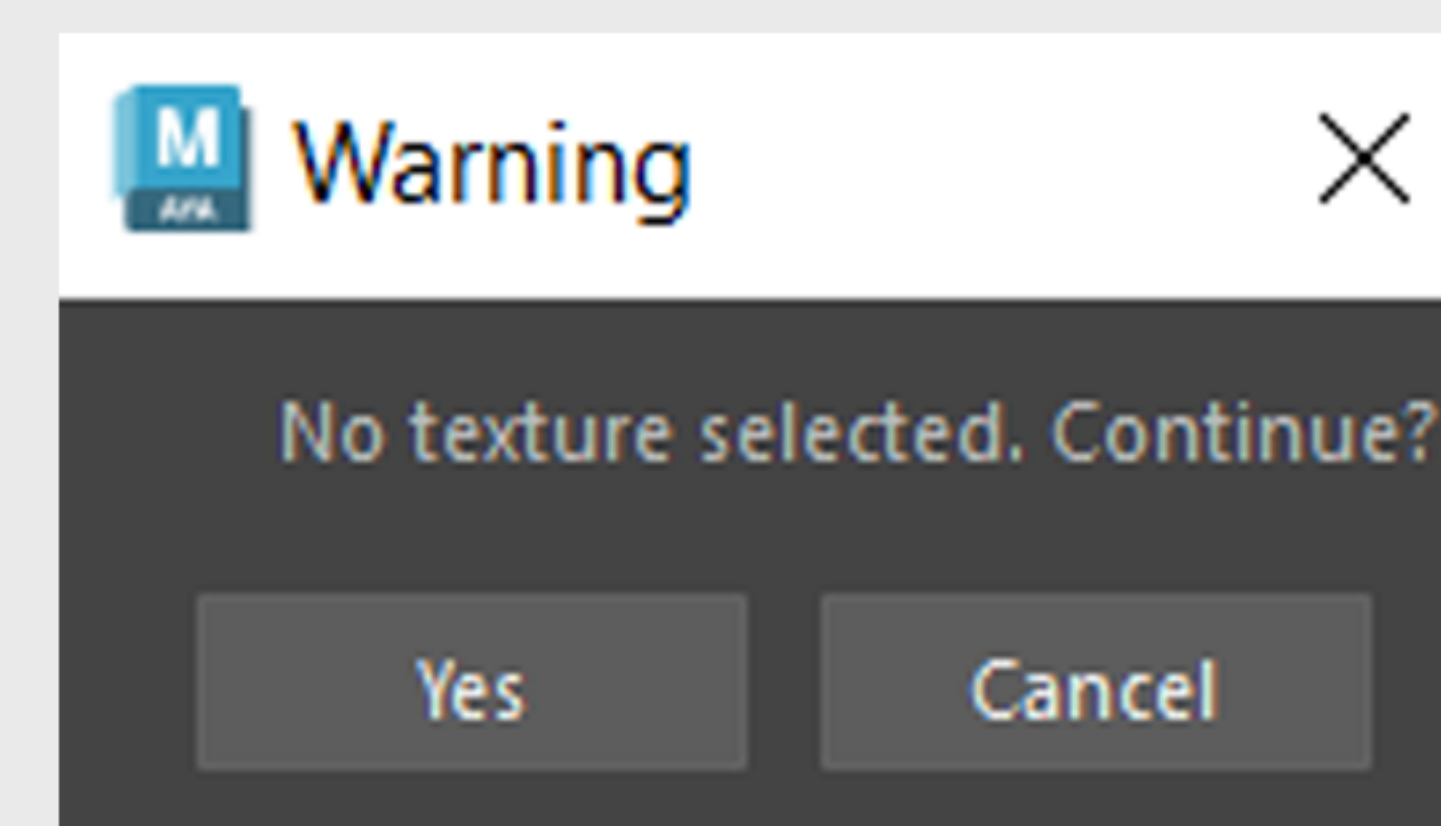
the model creation section



comparison of a 50 density grid and a 150 density grid

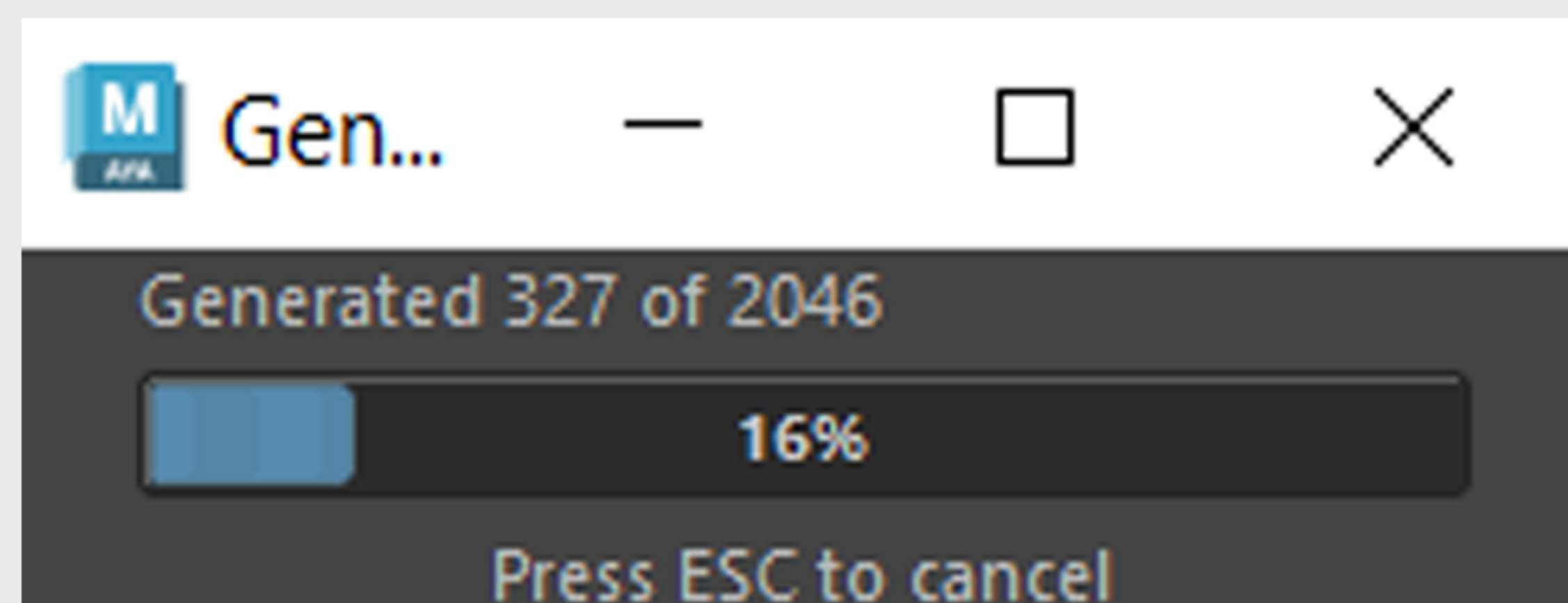
- When an object is selected, a grid box will spawn around it, and the parameters of the model creation section unlock.
- ‘Voxel Density’ controls the voxel size and count: the number equates to how many voxels fit inside the longest side of the object’s bounding box. This means that the size of the object in the scene does not matter; the voxel size is based on a percentage of the object’s size rather than fixed measurements.
- Changing the voxel density will update the box, giving an idea of what the resulted voxel mesh will look like. Setting the view mode to side or front view is recommended to make it more readable.

- The ‘Use Texture’ checkbox determines whether a texture should be generated for the mesh; if left unchecked, step 3 will be skipped.
- If checked on, the user may select a path to the texture that it used on the mesh, either by typing it directly, or by clicking ‘browse’ and selecting the file in the file dialog.
- Not entering a path or entering a path that does not exist will give a respective warning or error.



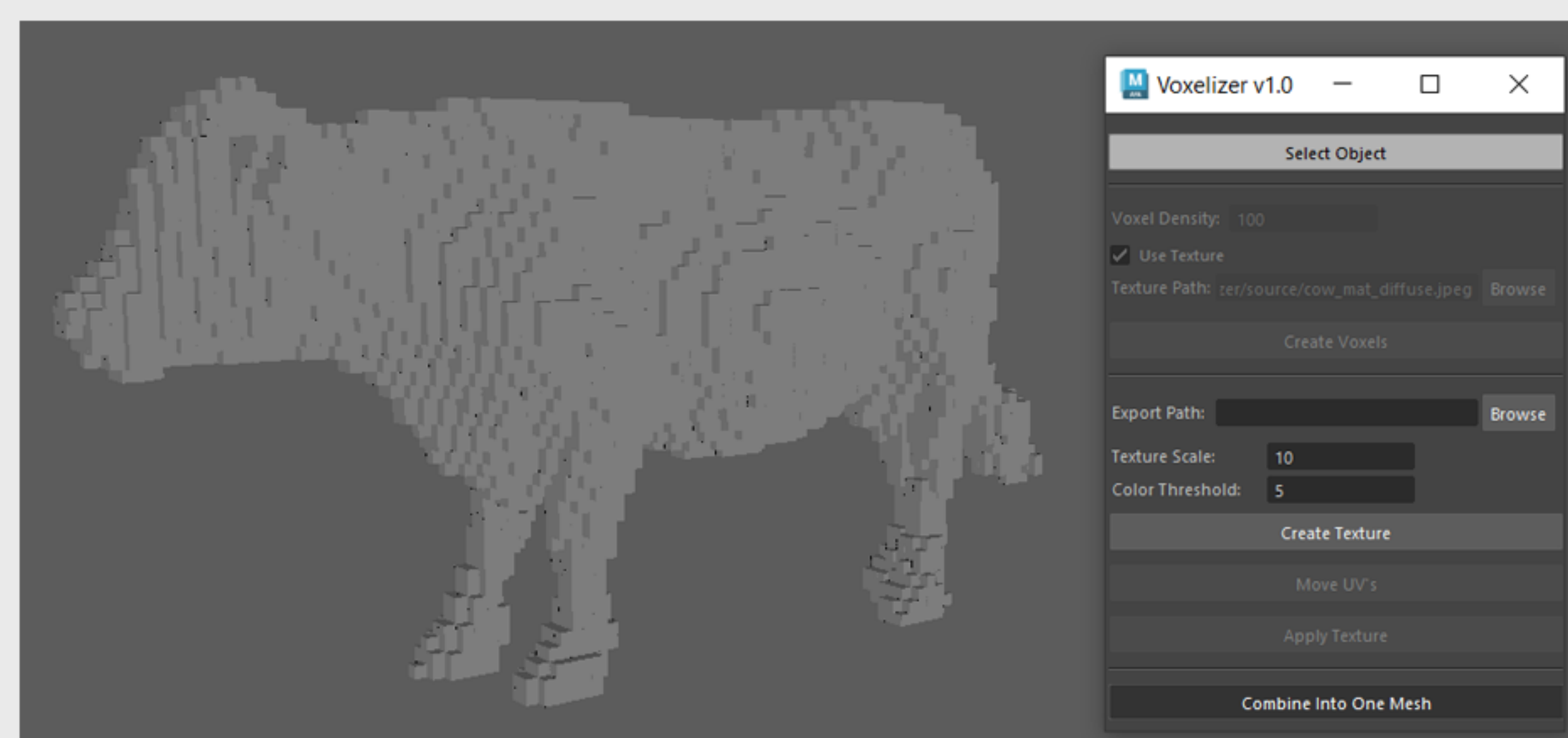
2. Texture Creation

- Once 'Create Voxels' is pressed, a short fluid simulation will run, and the voxels will be generated. A small new window opens that shows the progress in real time.
- Pressing the escape key will stop the generation. Any voxels created so far will remain. To run the operation again, undo and press the button again, or restart the tool.

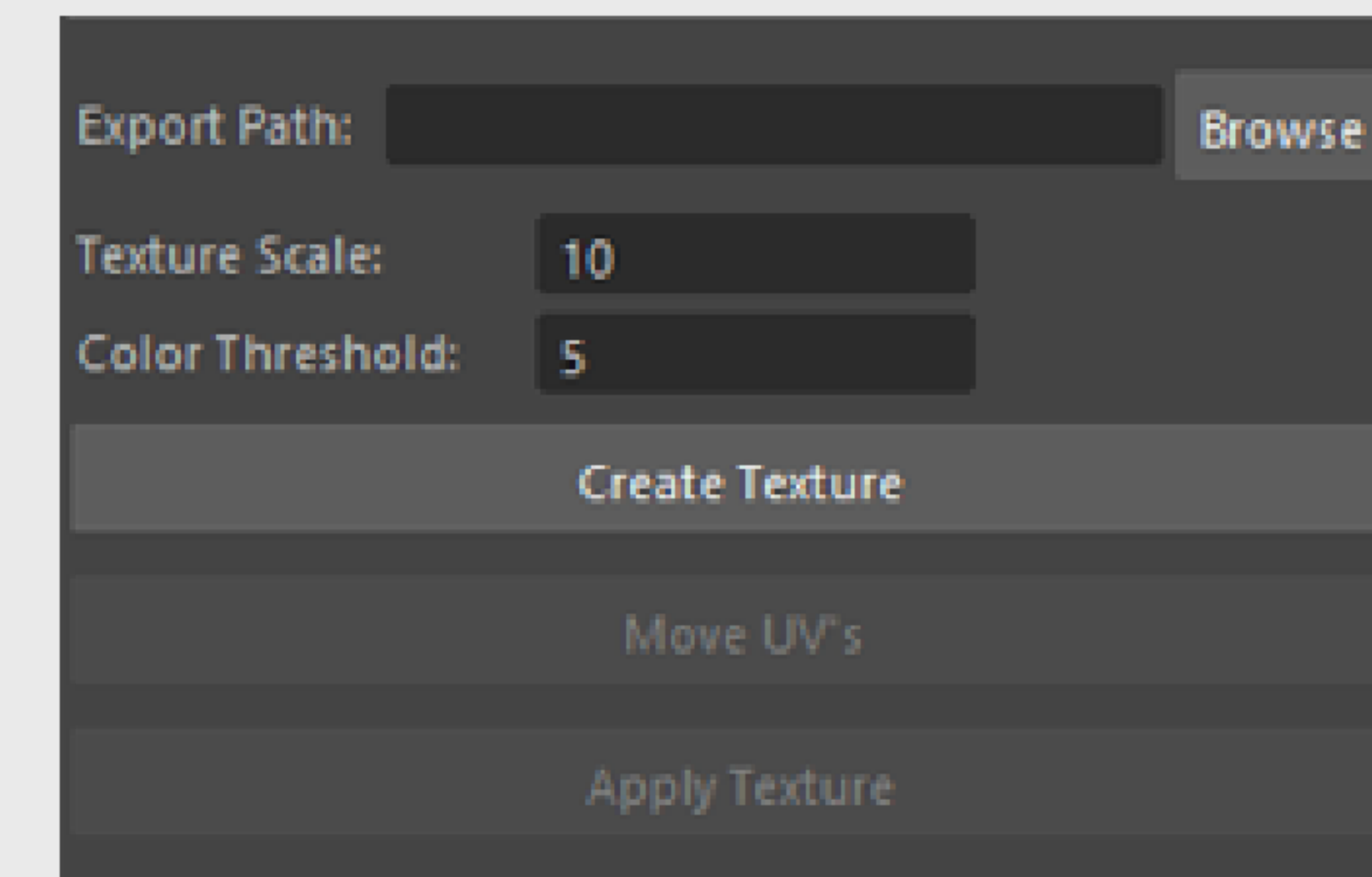


the voxel generation progress bar

- Once the operation is done, the model creation section will lock, and if the 'Use Texture' checkbox was ticked on, the texture creation parameters become available.



a 100 density mesh is generated;
the texture section is unlocked



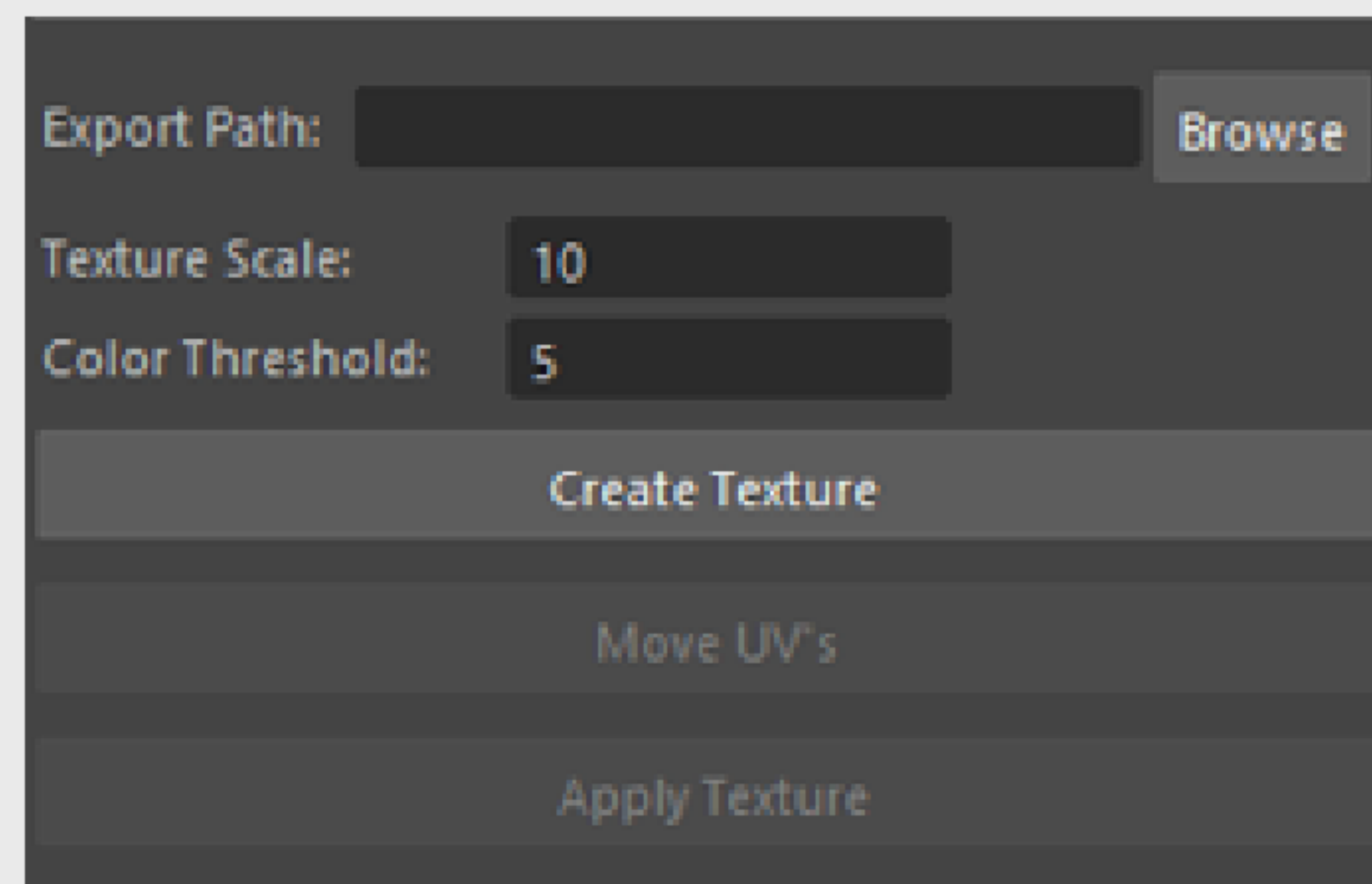
the texture creation section

- First, the user must select a folder to export the generated texture to. Like the texture import, both manually typing the path and selecting it via the file dialog is possible.
- The 'Texture Scale' parameter controls the scale of the outputted image. The texture is a tile grid of colors, and in theory every color only needs 1 pixel. However, some software can have problems with very small images, so this parameter allows for upscaling to bigger measurements. Setting the scale to 10 means that a single color gets a tile of 10x10 pixels.



the same texture with texture scales 1, 10 and 20

2. Texture Creation



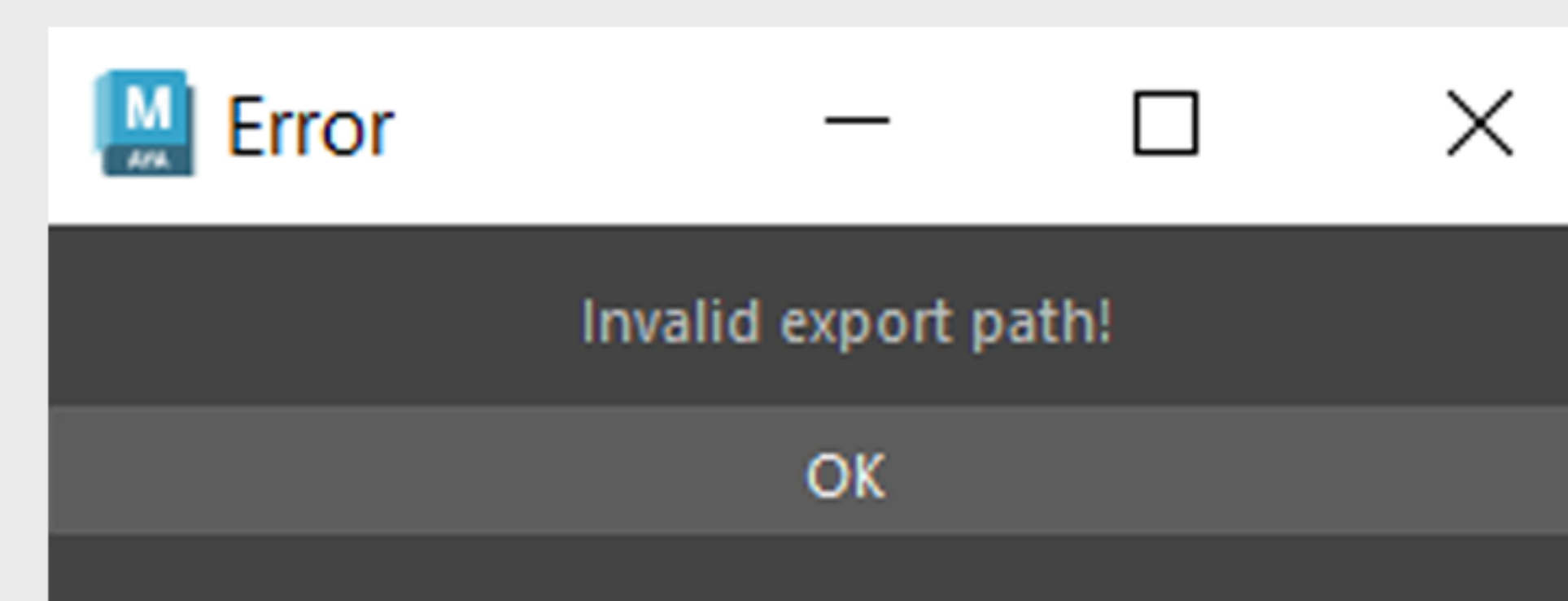
the texture creation section

- The 'Color Threshold' parameter controls how close colors need to be for them to be assigned to the same color tile. The number represents the size the difference in RGB values (expressed in [255,255,255]) needs to be for the color to be considered new and get its own new tile.
- e.g. [76,104,198] is compared to [75, 110,212]. The highest difference is 14, so if the threshold is set to 14 or higher, they get assigned the same color; a lower threshold will give both their own tile.
- Setting the threshold to 0 gives each voxel its own tile, even if some are the exact same color.

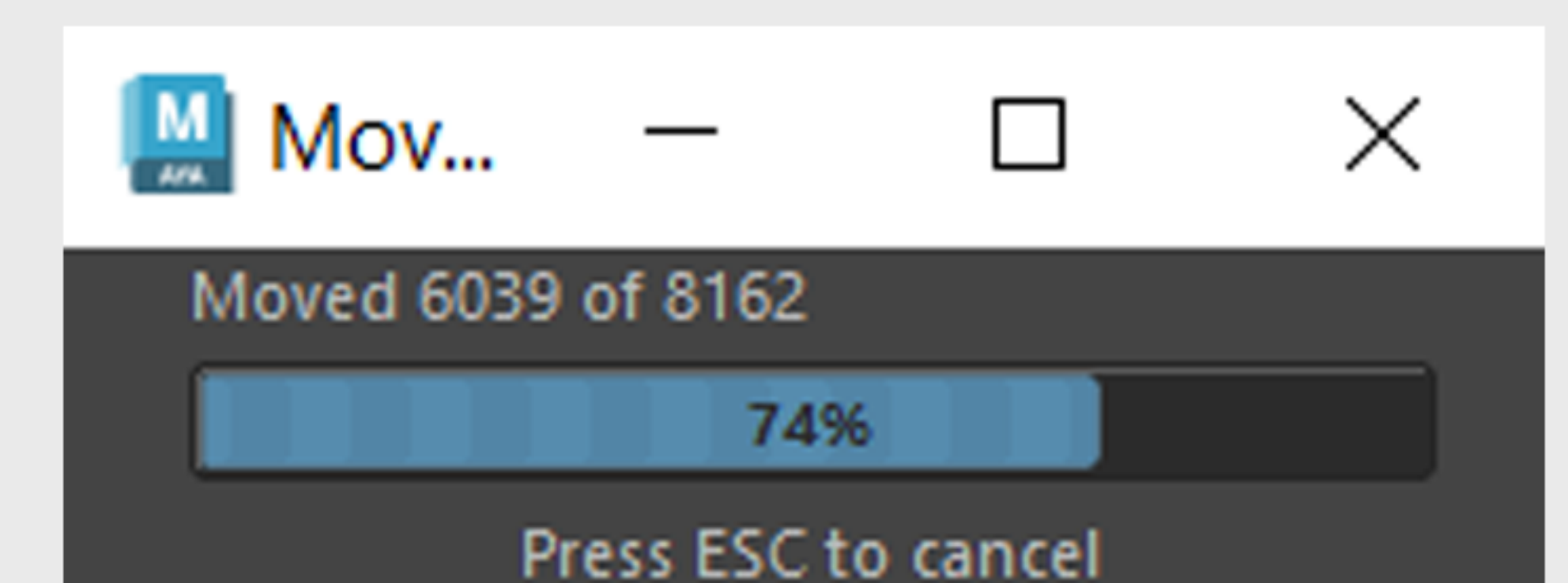


the same texture with thresholds 0, 5 and 20

- The 'Create Texture' button will create the image and save it in the specified folder. Entering a non-existent path or leaving the space empty gives an error message.
- Upon a successful export a confirm window is displayed. Should the user not be happy with the result, they can change the parameters and create a new one. Important: the texture output path is saved here, and in the following steps, **the texture that was generated last will be used**. The texture should not be renamed until all steps are completed.
- Creating the texture unlocks the 'Move UV's' button. This button will move the UV vertices of each voxel to their corresponding tile on the texture. As this step can take a while, A window with a progress bar tracks the amount of voxels that still need to be processed.

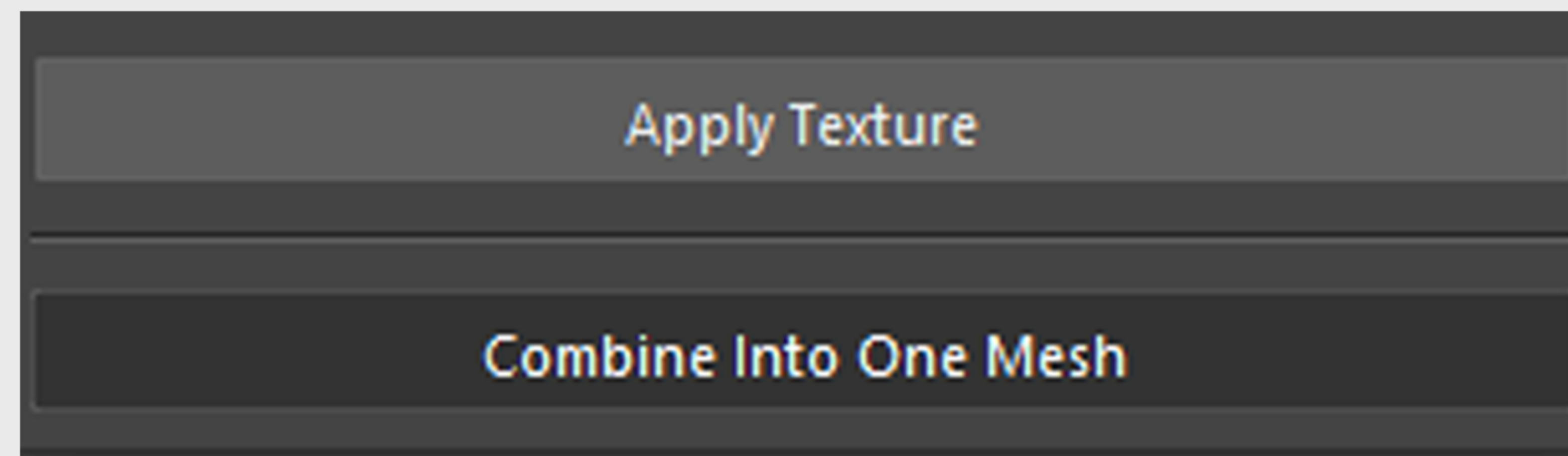


error on exporting the texture with an empty or invalid path



the move UV's progress bar

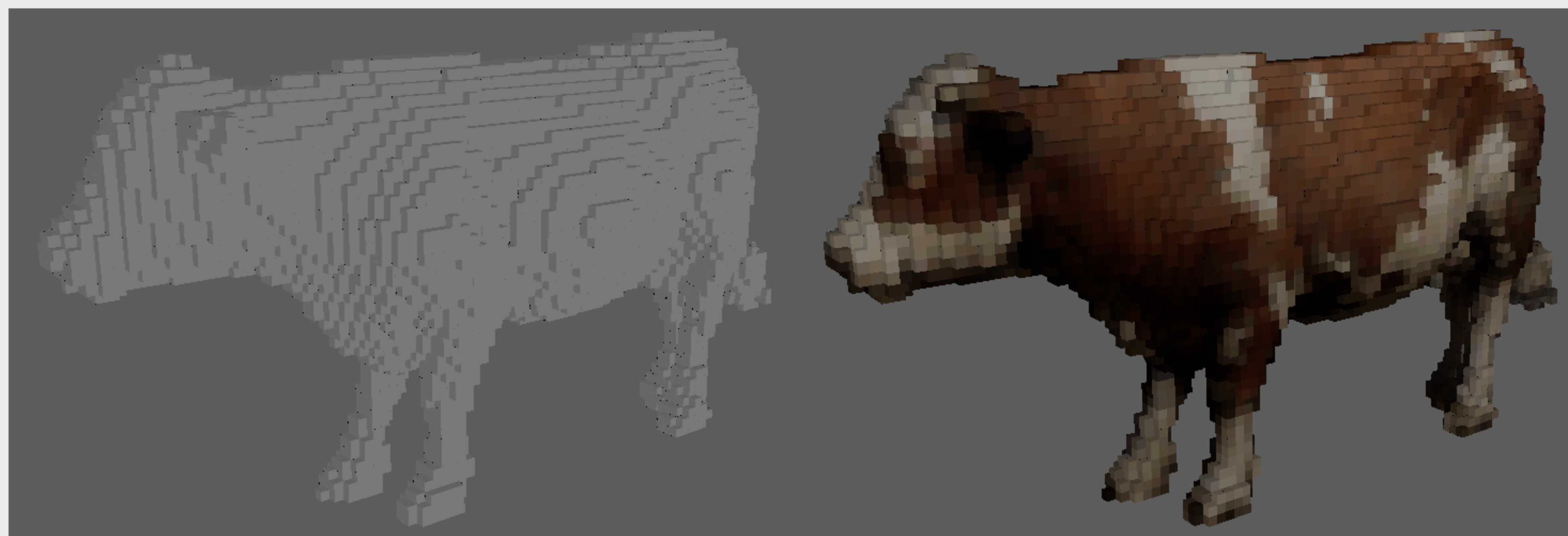
3. Apply & Merge



the optional apply texture and combine buttons

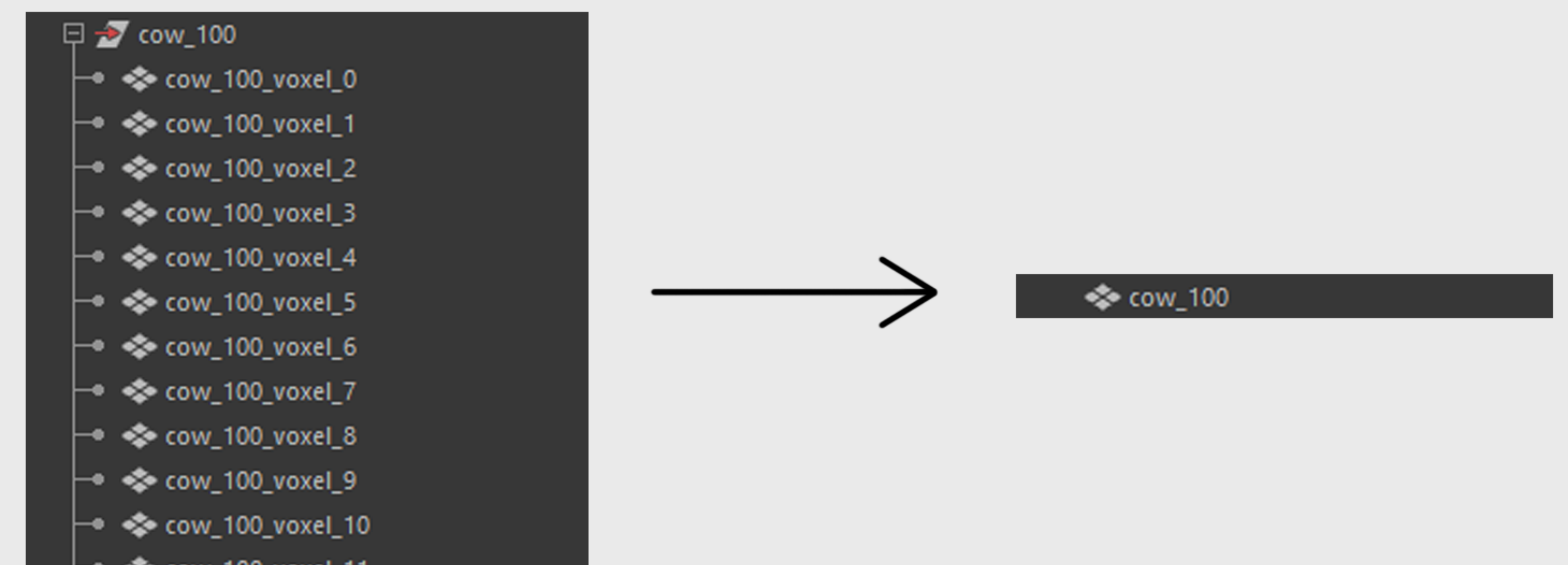
- The final two buttons are optional:

- 'Apply Texture' creates a new material in the scene that uses the exported texture as the base color, and assigns it to all voxels. While this is not necessary for exporting the mesh, it does visualise the end result directly inside of Maya.



the texture applied to the voxels

- 'Combine Into One Mesh' does as it states: all voxels are combined into one. It does **not** fuse the voxels: they are still cubes with overlapping sides and have faces on the inside of the mesh. While not required, this step results in the mesh running smoother inside of maya, a faster export, and a lower fbx file size.
- This step can take quite long, and there is no progress bar, as tracking the progress would require the meshes to be merged one by one, instead of all at once. This would massively increase the loading time.
- This button locks the rest of the script, as merging all voxels makes it impossible to individually edit them anymore.



the voxels being merged into one object