COSCH Training School
Lab session, Day 2

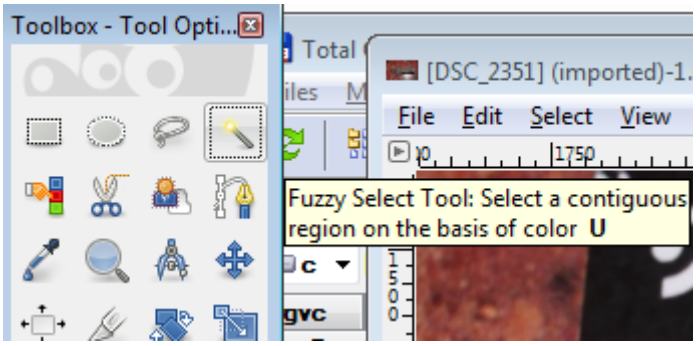
Basic steps to follow during the Lab session on *Correspondence-less fusion of colour images and 3D surfaces*

Preparing the 2D input data:

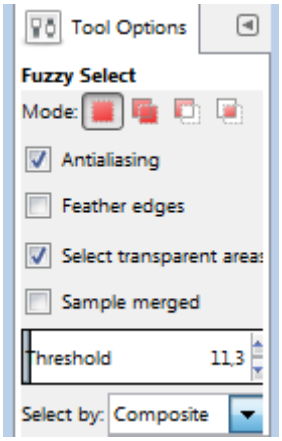
At this step, you need to prepare a "good" segmentation of your 2D image (a good segmentation means that you should be able to segment out the same region on the 3D data). You will need black&white segmentation masks, at the same resolution as the camera was calibrated, and the observation images were taken.

For this you could use any segmentation method, like the ones presented on the previous day's session implemented in PCL, or you can use some simple image manipulator software, like Gimp (or Photoshop). We will use Gimp here.

Open up your observation image in Gimp, and try to use the Fuzzy Select Tool (similar to the *Magic Wand* tool in Photoshop).



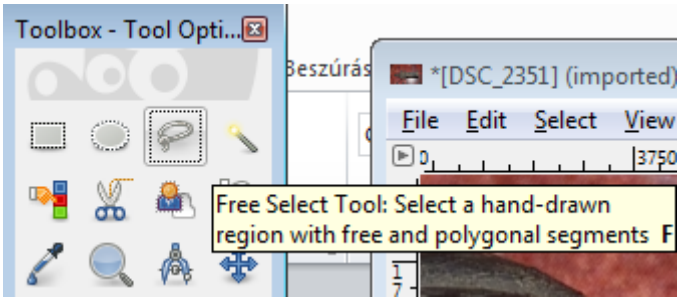
By clicking on any point of the image, you start an intelligent region growing segmentation that will result in the selection of a region around that point. You can add multiple such regions, by clicking while holding the SHIFT button. Eventually you can even subtract wrongly included regions, by clicking inside them while holding the CTRL key. If at any given step you want to go back one step, just press CTRL+Z (Undo). You can play around with the parameters of the tool you are using in the Toolbox panel:



At first steps you will see something like this:

You should try to define a region by a precise selection, without holes in it, and without branches hanging out on the sides.

If the automatic method does not enable you to do this, you can still refine your result by using the *free selection tool* (similar to *Lasso tool* in Photoshop).

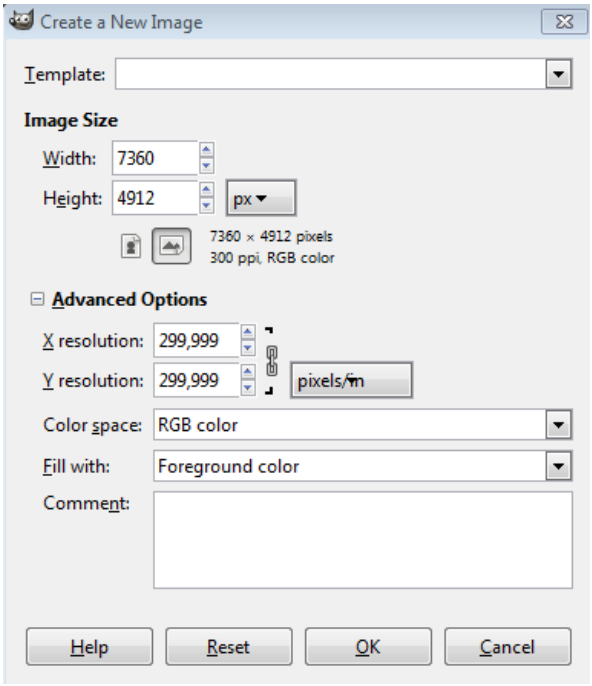


You can again press SHIFT to add regions to your selection, or CTRL to subtract. You just have to define a region by placing arbitrary number of points, and closing a loop by them, or by simply drawing a region.

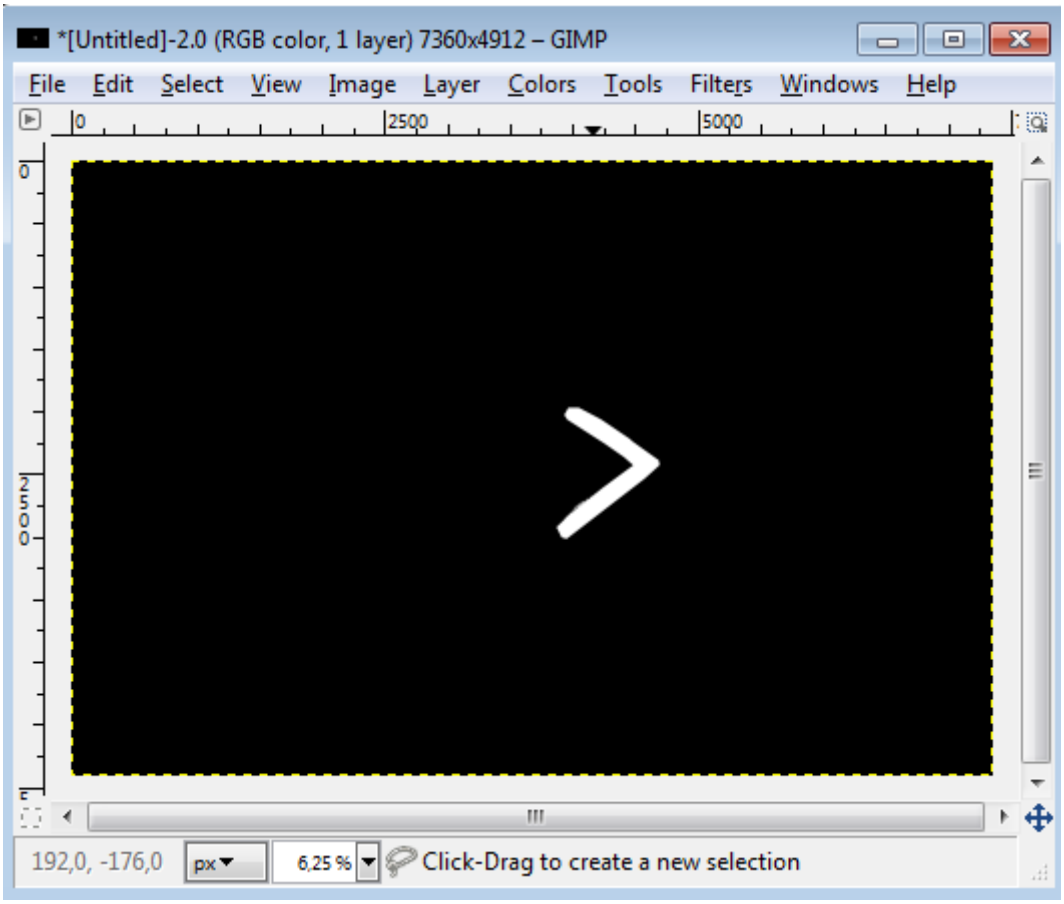
You should get to a result similar to this (if your desired region is this V shape region on the warrior statue):

You can export this selection into a 2D mask, by going into the menus, and clicking *Select / Save to Channel*, and then clicking *Select / All* (CTRL+A), and then copying the selected channel by pressing CTRL+C.

Then create a new black image, by clicking *File / New*. Make sure that the new file will have the same resolution as your original image, and select the fill color to be black (Foreground color if not changed):



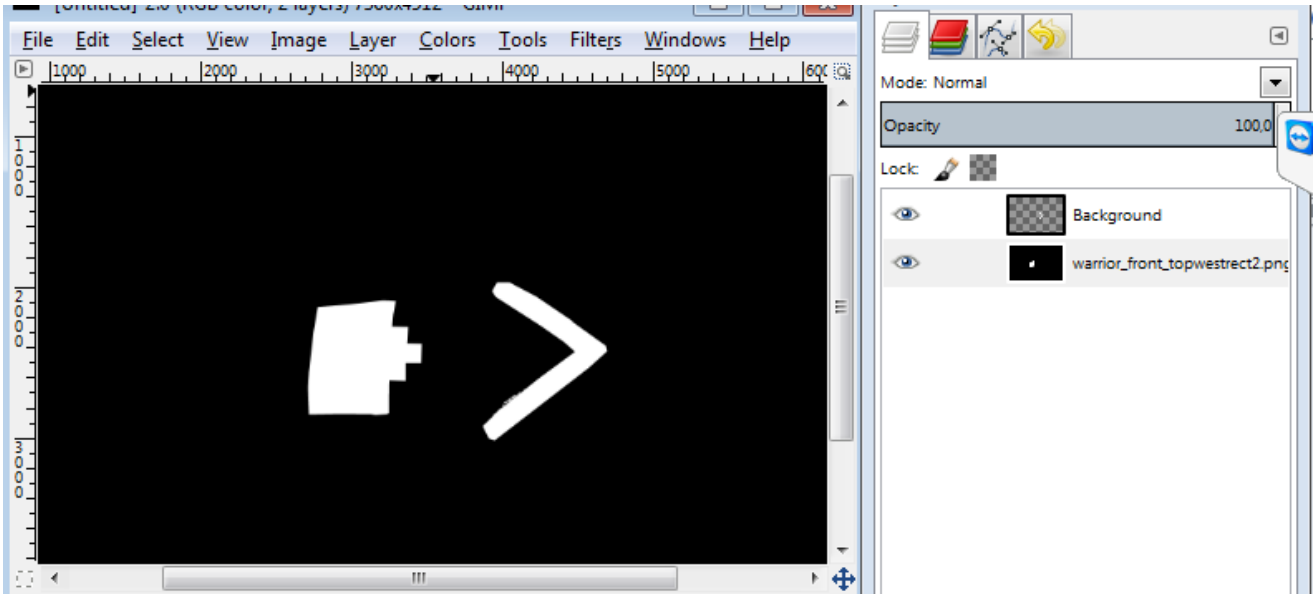
If done, just simply paste your copied layer into this window, it should shop up as a white mask like this:



You have successfully created a 2D segmentation mask. Just export it using *File / Export,* selecting your folder, and saving in **PNG** file format.

Adding up multiple regions:

If you have more such regions segmented, you can generate different combinations of them, by simply opening one of them in Gimp, and then going to *File / Open as Layers*, and loading the next region. You will see in the *Layers – Brushes* panel the new Layer appeared. Right now it's placed on top of the other Layer, so that is not visible. Resolve this by simply selecting the top layer in the Layers panel, and going to the menu *Colors / Color to Alpha*, and set the black color to be transparent. The resuld should look like:

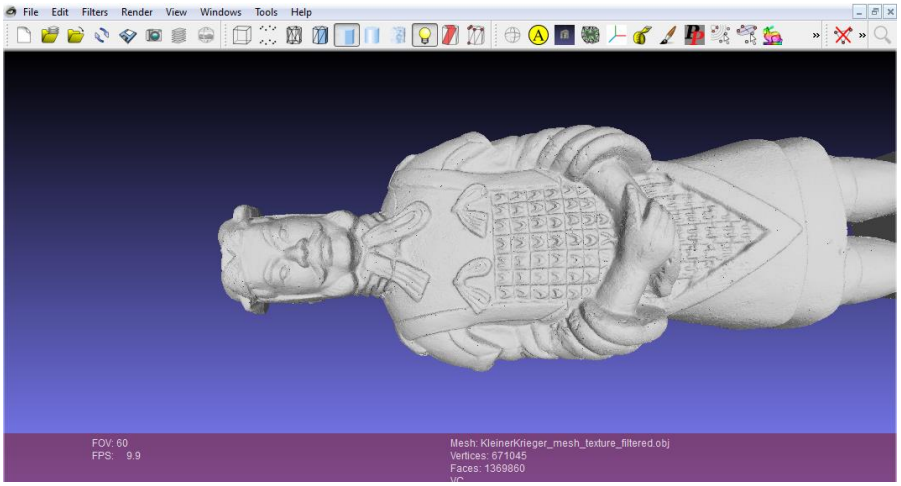


You can again simply Export this as a **PNG** file.

Preparing the 3D input data:

As you may suspect already, the main challenge here will be, to segment out the same regions in the 3D data domain. For this we will use the free Meshlab software.

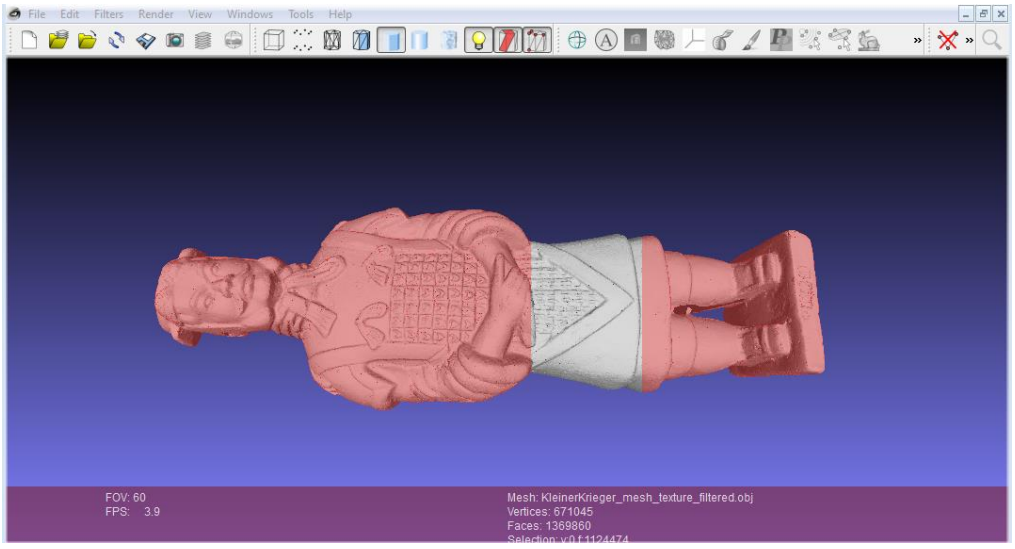
1. Open one of the 3D meshes from the dataset (use the .**obj** files) using Meshlab, eg. KleinerKrieger_mesh_texture_filtered.obj is already a filtered version, with reduced number of points, so you can manipulate it more easily (600k points instead of 2Million).



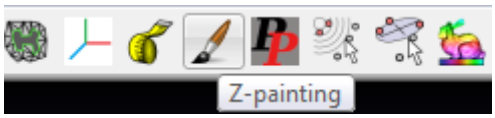
2. First you should compute the normal vectors for your point set. To do this, go to *Filters / Normals,Curvature...* or alternatively *Filters / Point set*, look for *Compute normals for point sets*. This information will be needed at the initialization step of the algorithm.
3. To provide the simple ASCII point list of the pointcloud we want to apply the fusion on (we want to visualize the resulting backprojection on), you should export the points list in an ASCII file format. Click on *File / Export Mesh As* and select .**xyz** Point Cloud file type, and on the next screen uncheck everything. We only need the point coordinates in this file.
   This step can also be performed after some filtering is done.
4. To segment out the desired region from the triangulated 3D pointcloud, you can use two simple tools.

   First the *Select faces* tool:

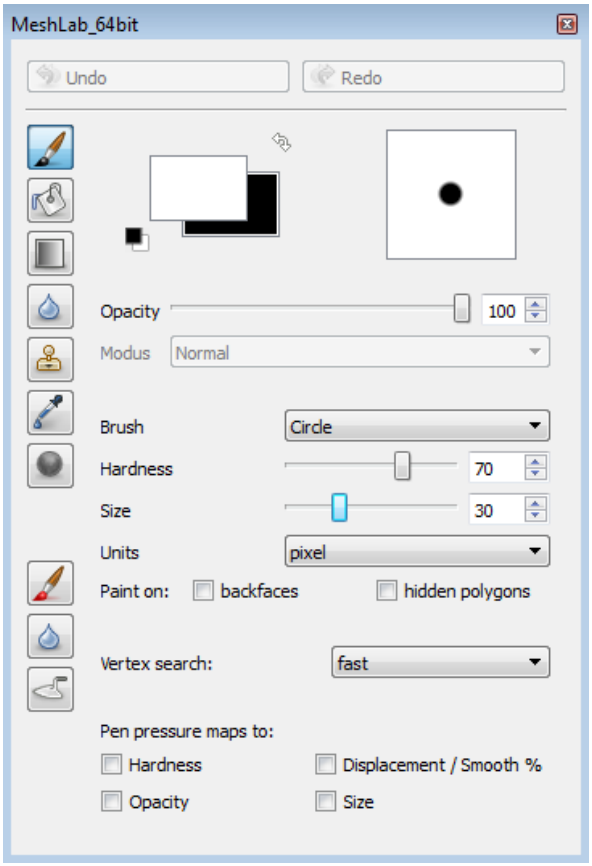
   Tip: Use this tool to select unwanted regions by drawing a rectangle around them. The selected points/faces will be shown in red. You can remove these unwanted regions by simply pushing SHIFT+DEL, or clicking on *Filters / Selection / Delete selected faces and vertices*. By doing so you also reduce the size of the data you work on, resulting a more fluid experience on slower machines.



   Second, for a more precise selection in the more complex parts, you can also use the *Z-painting* tool:

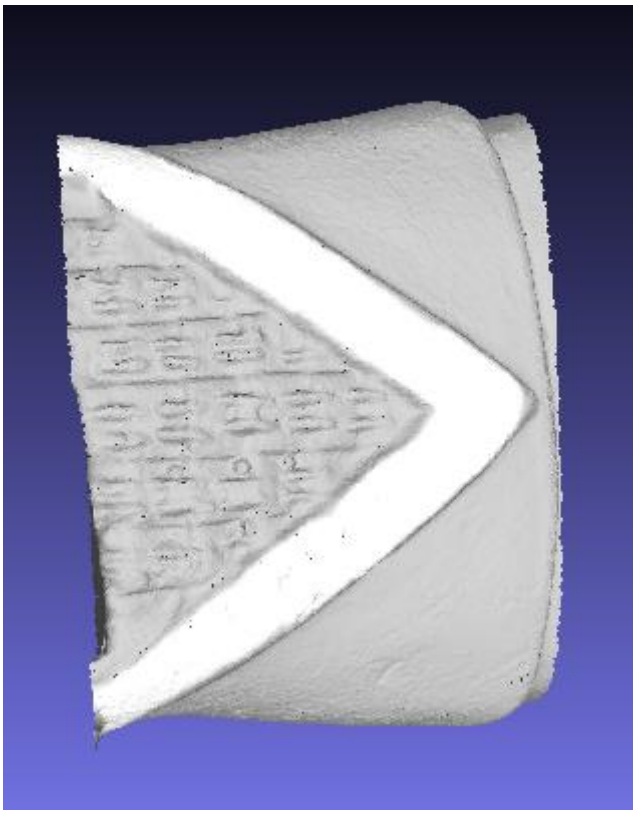


   First select the Brush tool, you can change the shape of the Brush, the Size and Hardness and other parameters as shown below.



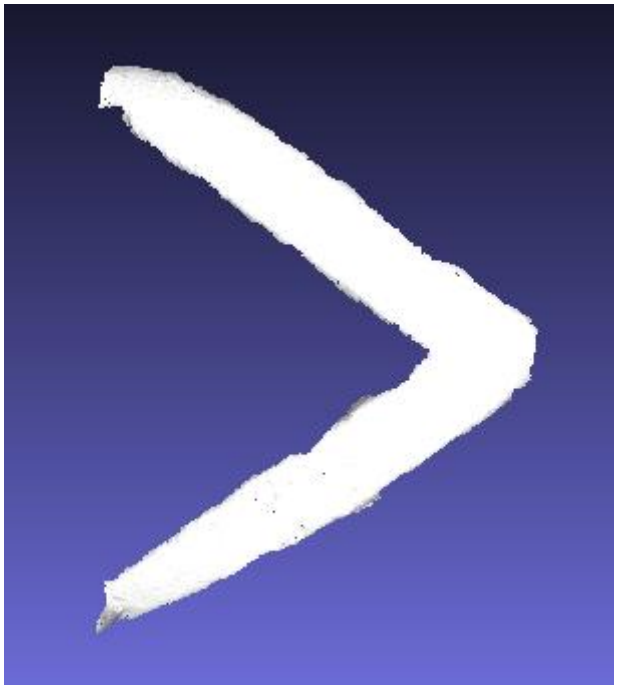
If done, you can start selecting your region.

There are two approaches here, you may want to paint over the desired region, like shown below, or you may also paint over the unwanted faces/points.

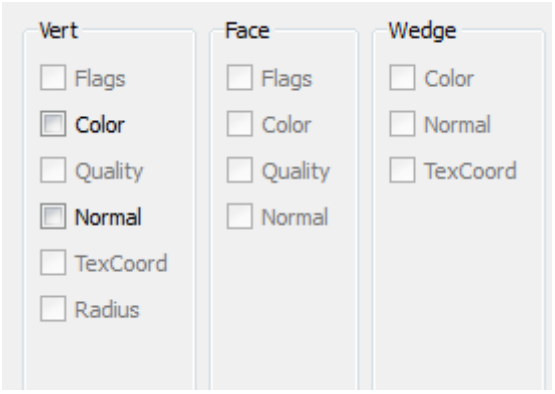


You can grab the painted region, by selecting it based on its painted color. Use the *Filters / Selection / Select faces by color* menu, select the same color you used with the Z-paint tool, then click apply.

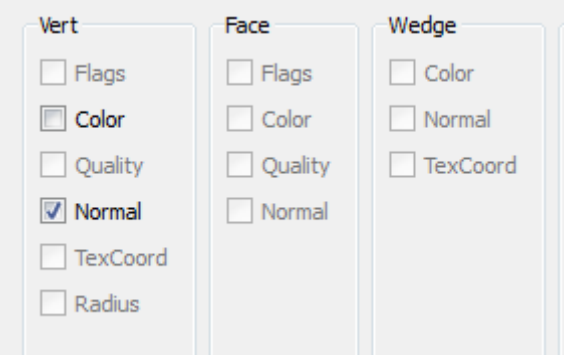
Now based on what you have painted, you may delete the selection with SHIFT+DEL, or you may invert the selection, by going to the *Filters / Selection / Invert selection* menu, then deleting the regions in red highlight.
You should end up with something similar:



5. Now use the *File / Export Mesh As* menu to export your data in .**obj** file. Make sure to **unselect** the Color and Normal checkboxes.

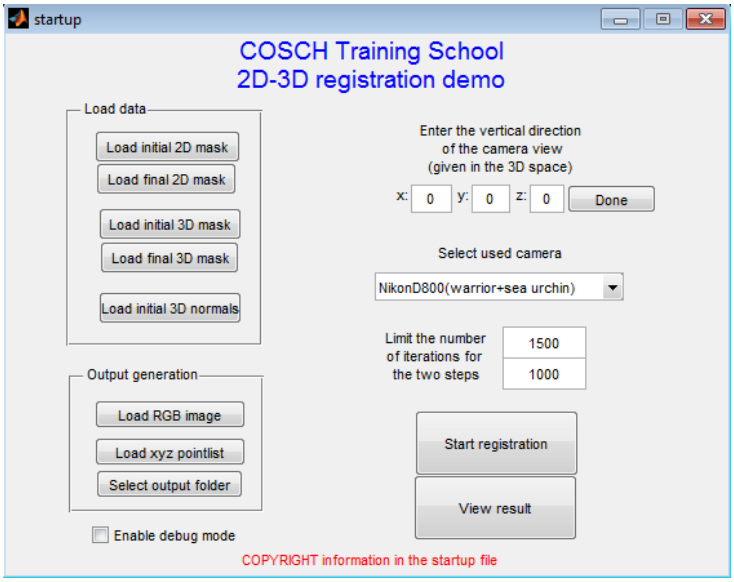


6. Our method will also need the Normal information calculated previously, for this, since this is already calculated for the segmented region, we should export it, in a separate file. Go to the *File / Export Mesh As* menu, give a descriptive filename, select .**obj** file type, and on the next window, **check** the Normal checkbox.



2D - 3D Registration:

1. First open the *startup.m* file from the downloaded package with Matlab and start it using the Green arrow in the matlab toolbar, or by pressing F5. You should see this User Interface:

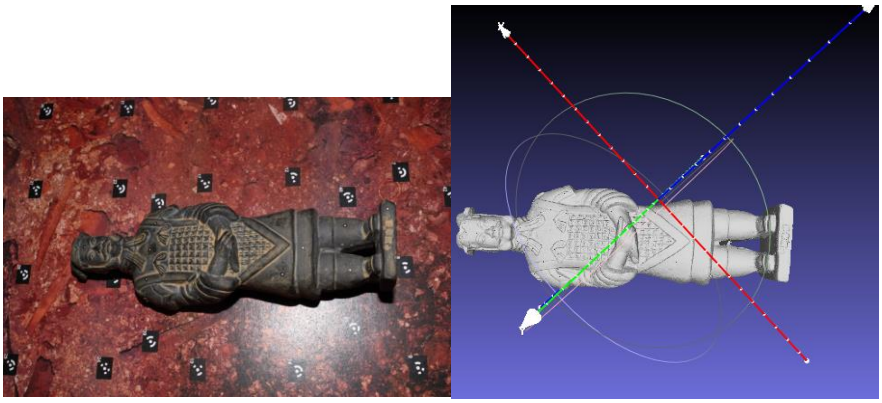


Here you have separate buttons for **loading the previously created 2D and 3D masks**. The initial masks are those that will be used for an **initial alignment**, this usually should contain only one region, while the final masks may have more regions on them. If you only wish to use a single mask for a simple registration, that's also possible, just simply load the same file for the initial and final mask, both in the 2D part and in the 3D part. Please use all four of them!

2. Pushing the *Load initial 3D normal* button, you will have to select the file generated at **point 6** above. Be careful, this region should **correspond** to the initial regions that you are using for the registration, selected above.

3. Next you should specify a rough **estimation** of the *vertical direction* of the camera, as it was looking when capturing the photo, relative to the coordinate frame of the 3D point cloud. This information could be easily obtained from an IMU sensor, but since we don't have this information stored, we need some user interaction at this step.

For this open up your original point cloud again in Meshlab, and try to visualize it from a point of view similar to the one that the camera was placed when taking the photo. Click on the *Render / Show Axis* menu to see the coordinate axis. Now try to estimate the direction of an up-looking vector (relative to this camera position) in this 3D space.



In this case it would be something looking in the positive X and Z direction just input an estimated vector like 0.5, 0, 0.5) and hit Done. (If you are unsure, just enter anything, you will see the results)

4. You have to select the camera calibration file that you want to use, based on the test case you are working on. Be careful as the coin test case also has portrait images, that need different calibration parameters.

5. At the output generation section, you also have to specify the RGB image's name, the .xyz pointlist that you have exported at point **3** in the previous section, and also an output folder to save the results to.

6. Finally you can check the Debug Mode checkbox, to see some step-by-step debug information and plots while the algorithm is running. In this case the program will occasionally wait for user input (hit ENTER) to continue to the next step.

Finally, if the registration was successful, you can push the view result button, to open up the fused pointcloud.