

---

# Meshroom

*Release 0.1*

Oct 17, 2019



<b>1</b>	<b>Manual</b>	<b>1</b>
<b>2</b>	<b>Install</b>	<b>3</b>
<b>3</b>	<b>The Graphical User Interface (GUI)</b>	<b>9</b>
<b>4</b>	<b>Simple import</b>	<b>11</b>
<b>5</b>	<b>3D Viewer</b>	<b>13</b>
<b>6</b>	<b>Advanced Node parameters</b>	<b>15</b>
<b>7</b>	<b>Start Reconstruction</b>	<b>17</b>
<b>8</b>	<b>Augment Reconstruction</b>	<b>19</b>
<b>9</b>	<b>Live Reconstruction</b>	<b>23</b>
<b>10</b>	<b>External Reconstruction</b>	<b>25</b>
<b>11</b>	<b>Import old Meshroom project</b>	<b>27</b>
<b>12</b>	<b>Test Meshroom</b>	<b>29</b>
<b>13</b>	<b>Reconstruction: How long does it take?</b>	<b>31</b>
<b>14</b>	<b>Connect Nodes</b>	<b>33</b>
<b>15</b>	<b>Complete Node List</b>	<b>35</b>
<b>16</b>	<b>Supported Formats</b>	<b>59</b>
<b>17</b>	<b>Tutorials</b>	<b>61</b>
<b>18</b>	<b>Capturing</b>	<b>75</b>
<b>19</b>	<b>More</b>	<b>77</b>
<b>20</b>	<b>FAQ from GH-Wiki</b>	<b>87</b>

<b>21</b>	<b>References</b>	<b>95</b>
<b>22</b>	<b>Glossary</b>	<b>97</b>
<b>23</b>	<b>About</b>	<b>99</b>
	<b>Index</b>	<b>103</b>

# CHAPTER 1

---

## Manual

---

March. 2019 v0.4.4



## 2.1 Requirements

**Hardware:** NVIDIA CUDA-enabled GPU (compute capability  $\geq 2.0$ ).

You can check your [CUDA Properties](#) here or on the [NVIDIA dev](#) page.

Install the [latest NVIDIA drivers](#).

**Note :** If you do not have a CUDA GPU, you can use the draft meshing option which uses the CPU for meshing.

**Minimum :** Ram: 8gb+, HDD: ~400MB Meshroom, 2gb+ for cache data and models

CPU: not too old (~3 years and newer should be ok), NVIDIA GPU

**Recommended :** CPU: i7/Ryzen 7 or better, 32gb+ Ram, HDD/SSD: 20GB+ , NVIDIA GTX1070+ (reconstruction time for this setup: 1 minute per image)

**Operating Systems:** Windows x64, Linux, OSX (some work is required)

## 2.2 Get Meshroom

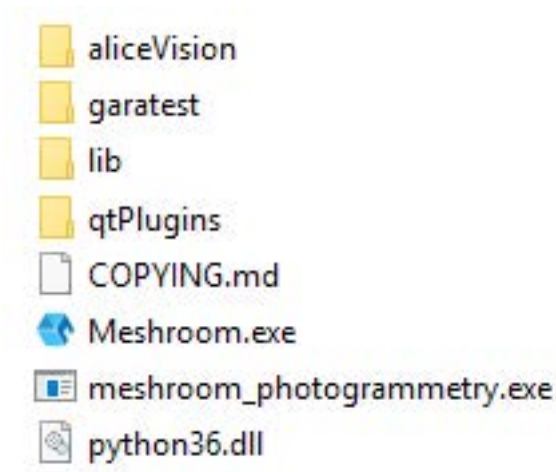
Meshroom Binaries can be downloaded from <https://alicevision.github.io/#meshroom>

Prebuilt binaries on this page are all-in-one packages including AliceVision and all required resources.

*Note:* The [pre-build Meshroom 2019.1.0](#) does not include all the features of the developer version. Some of those features will be included in the *next build release*.

## 2.3 Windows

1. Download Meshroom
2. extract ZIP to a folder of your choice



3. You can start Meshroom by clicking on the executable. No installation required.

---

**Note:** Do not run Meshroom as Admin. This will disable drag-and-drop.

---

## 2.4 Linux

### Get the project

See [INSTALL.md](#) to setup the project and prerequisites.

Get the source code and install runtime requirements:

```
git clone --recursive git://github.com/alicevision/meshroom
cd meshroom
pip install -r requirements.txt
```

### Start Meshroom

You need to have [AliceVision](#) installation in your **PATH** and **LD\_LIBRARY\_PATH**.

### Launch the User Interface

```
# Linux/macOS
PYTHONPATH=$PWD python meshroom/ui
```

### Note:

On some distributions (e.g Ubuntu), you may have conflicts between native drivers and mesa drivers, resulting in an empty black window. In that case, you need to force usage of native drivers by adding them to the

**LD\_LIBRARY\_PATH:**

```
LD_LIBRARY_PATH=/usr/lib/nvidia-340 ./Meshroom
```

You may need to adjust the folder

```
/usr/lib/nvidia-340
```

with the correct driver version.



On Ubuntu, you may have conflicts between native drivers and mesa drivers. In that case, you need to force usage of native drivers by adding them to the

`LD_LIBRARY_PATH:`

`LD_LIBRARY_PATH=/usr/lib/nvidia-340 PYTHONPATH=$PWD python meshroom/ui`

You may need to adjust the folder `/usr/lib/nvidia-340` with the correct driver version.

Do not use foreign characters in path names (latin characters only).

### Launch a 3D reconstruction in command line

**Linux:** `PYTHONPATH=$PWD`

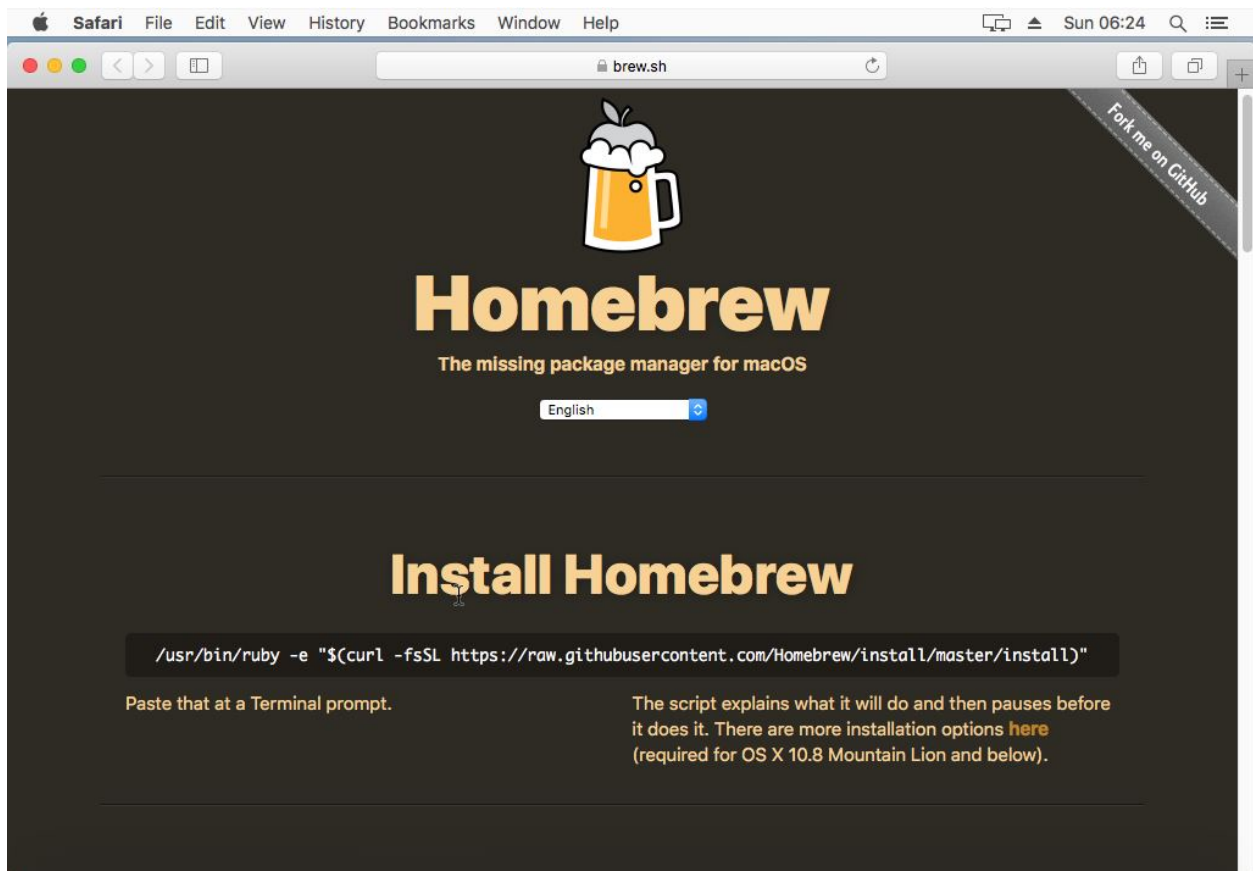
`python bin/meshroom_photogrammetry --input INPUT_IMAGES_FOLDER --output OUTPUT_FOLDER`

German: <http://paravel.org/blog/2018/12/10/how-to-photogrammetry-mit-meshroom-unter-ubuntu-16-04>

## 2.5 OSX

Originally published on 2018-08-17 by [Ryan Baumann](https://ryanfb.github.io) on [ryanfb.github.io](https://ryanfb.github.io)

[AliceVision](#) and its Meshroom program are an exciting new free and open-source pipeline for photogrammetry processing. Unfortunately, compiling and using either of these programs on Mac OS X is *not exactly straightforward*. As a result, I've compiled a [Homebrew](#) tap which includes the necessary formulae, and will use this post to outline how to use them to get up and running. *Note that this is intended as a first step for Mac users wishing to experiment with and improve the AliceVision/Meshroom software, and as a result these instructions may become outdated with time.*



The following instructions assume a working [Homebrew](#) install.

## 2.5.1 System Requirements

First off, your Mac will currently need an nVidia GPU with a CUDA compute capability of 2.0 or greater. This is probably a pretty small portion of all Macs sold, but you can check your GPU by looking in “About This Mac” from the Apple icon in the top left corner of the screen, under “Graphics”. If you have an nVidia GPU listed there, you can check its compute capability on [the nVidia CUDA GPUs page](#) .

Second, you’re going to need to install [the latest CUDA toolkit](#) . As of this writing, that’s CUDA 9.2, which is only officially compatible with OS X 10.13 (High Sierra), so you may also need to upgrade to the latest version of High Sierra if you haven’t already. Alongside this I would also suggest installing the latest nVidia CUDA GPU webdriver, which as of this writing is [387.10.10.10.40.105](#) .

Third, [CUDA 9.2](#) is only ‘ compatible with the version of [\\_ clang \\_](https://docs.nvidia.com/cuda/cuda-installation-guide-mac-os-x/index.html) distributed with Xcode 9.2 [\\_](https://docs.nvidia.com/cuda/cuda-installation-guide-mac-os-x/index.html) , and will refuse to compile against anything else. You may have an older or newer version of Xcode installed. As of this writing, if you fully update Xcode within a fully updated OS X install, you’ll have Xcode 9.4.1. To get back to Xcode 9.2, what you can do is go to [Apple’s Developer Downloads page](#) (for which you’ll need a free Apple developer account), then search for “Xcode 9.2”, then install the Command Line Tools for Xcode 9.2 package for your OS version. After installing, run

```
sudo xcode-select --switch /Library/Developer/CommandLineTools
```

 and then verify that

```
clang --version
```

 \* \* shows *Apple LLVM version 9.0.0*

Once you’ve done all this, you can verify a working CUDA install by going to `/Developer/NVIDIA/CUDA-9.2/samples/1_Uutilities/deviceQuery` and running `sudo make && ./deviceQuery` , which should output your GPU information. If it doesn’t build correctly, or deviceQuery errors or doesn’t list your GPU, you may need to look over the steps above and check that everything is up to date (you can also check the CUDA panel in System Preferences). ..  
image:: homebrew-inst.jpg

## 2.5.2 Installation

If you’ve followed all the above setup instructions and requirements, installing the AliceVision libraries/framework should be as easy as:

```
brew install ryanfb/alicevision/alicevision
```

## 2.5.3 Meshroom Installation & Usage

I haven’t yet created a Homebrew formula for the [Meshroom package itself](#) , as it’s all Python and doesn’t seem particularly difficult to install/use once AliceVision is installed and working correctly. Just follow the install instructions there (for my specific Python configuration/installation I used `pip`

3 instead of `pip` and `python3` instead of `python` ):

```
git clone --recursive git://github.com/alicevision/meshroom
```

```
cd meshroom
```

```
pip install -r requirements.txt
```

You can report an issue on <https://github.com/ryanfb/homebrew-alicevision/issues>

One gotcha I ran into is that the CUDA-linked AliceVision binaries invoked by Meshroom don’t automatically find the CUDA libraries on the `DYLD_LIBRARY_PATH` , and setting the `DYLD_LIBRARY_PATH` from the shell launching

Meshroom doesn't seem to get the variable passed into the shell environment Meshroom uses to spawn commands. Without this, you'll get an error like:

```
dyld: Library not loaded: @rpath/libcudart.9.2.dylib
```

```
Referenced from: /usr/local/bin/aliceVision_depthMapEstimation
```

```
Reason: image not found
```

In order to get around this, you can symlink the CUDA libraries into /usr/local/lib (most of the other workarounds I found for permanently modifying the DYLD\_LIBRARY\_PATH seemed more confusing or fragile than this simpler approach): 1

```
for i in /Developer/NVIDIA/CUDA-9.2/lib/.a /Developer/NVIDIA/CUDA-9.2/lib/.dylib; do ln -sv "$i"
"/usr/local/lib/${basename "$i"}"; done
```

You can undo/uninstall this with:

```
for i in /Developer/NVIDIA/CUDA-9.2/lib/.a /Developer/NVIDIA/CUDA-9.2/lib/.dylib; do rm -v
"/usr/local/lib/${basename "$i"}"; done
```

You may also want to download the voctree dataset:

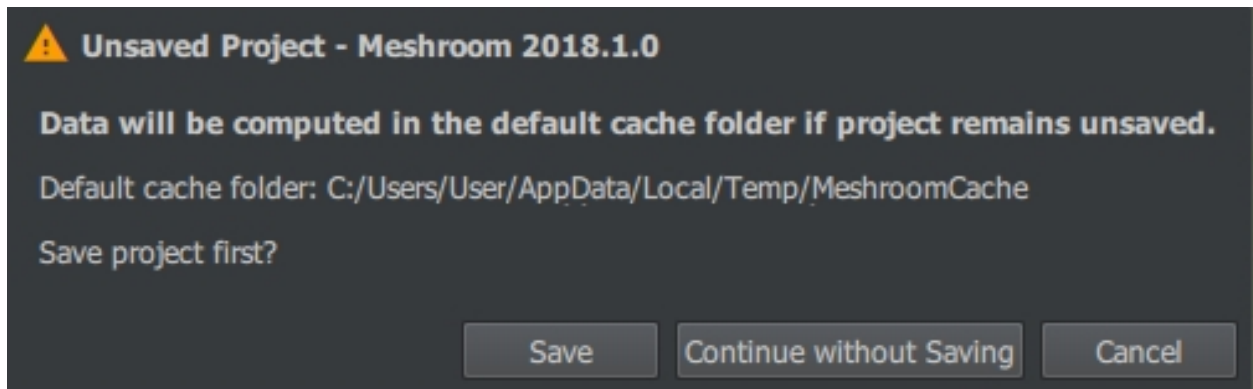
```
curl 'https://gitlab.com/alicevision/trainedVocabularyTreeData/raw/master/vlfeat_K80L3.SIFT.tree' -o
/usr/local/Cellar/alicevision/2.0.0/share/aliceVision/vlfeat_K80L3.SIFT.tree
```

Then launch with:

```
ALICEVISION_SENSOR_DB=/usr/local/Cellar/alicevision/2.0.0/share/aliceVision/cameraSensors.db ALICE-
VISION_VOCTREE=/usr/local/Cellar/alicevision/2.0.0/share/aliceVision/vlfeat_K80L3.SIFT.tree PYTHON-
PATH=$PWD python meshroom/ui
```

Import some photos, click "Start", wait a while, and hopefully you should end up with a reconstructed and textured mesh. By default, the output will be in MeshroomCache/Texturing/ (relative to where you saved the project file).

When you launch Meshroom *without* sudo, the temp path will be something like this:



When starting with sudo, it will be /tmp/MeshroomCache by default

#### Footnotes:

1. Previously, I suggested modifying meshroom/core/desc.py so that the return value at the end of the buildCommandLine 'method' <<https://github.com/alicevision/meshroom/blob/develop/meshroom/core/desc.py#L368>> '\_ instead reads:

```
return 'DYLD_LIBRARY_PATH="/Developer/NVIDIA/CUDA-9.2/lib" ' + cmdPrefix +
chunk.node.nodeDesc.commandLine.format(**chunk.node._cmdVars) + cmdSuffix
```

Baumann, Ryan. "AliceVision and Meshroom on Mac OS X." *Ryan Baumann - /etc* (blog), 17 Aug 2018, [https://ryanfb.github.io/etc/2018/08/17/alicevision\\_and\\_meshroom\\_on\\_mac\\_os\\_x.html](https://ryanfb.github.io/etc/2018/08/17/alicevision_and_meshroom_on_mac_os_x.html) (accessed 10 Oct 2018).

## 2.6 Docker

(WIP)

official: docker pull alicevision/meshroom <https://hub.docker.com/r/alicevision/meshroom>

Docker file on Github: <https://github.com/alicevision/meshroom/blob/master/Dockerfile>

Other:

<https://hub.docker.com/r/derfetzter/meshroom/> <https://hub.docker.com/r/fschwaiger/meshroom>

Link to CUDA and Docker <https://stackoverflow.com/questions/25185405/using-gpu-from-a-docker-container>

---

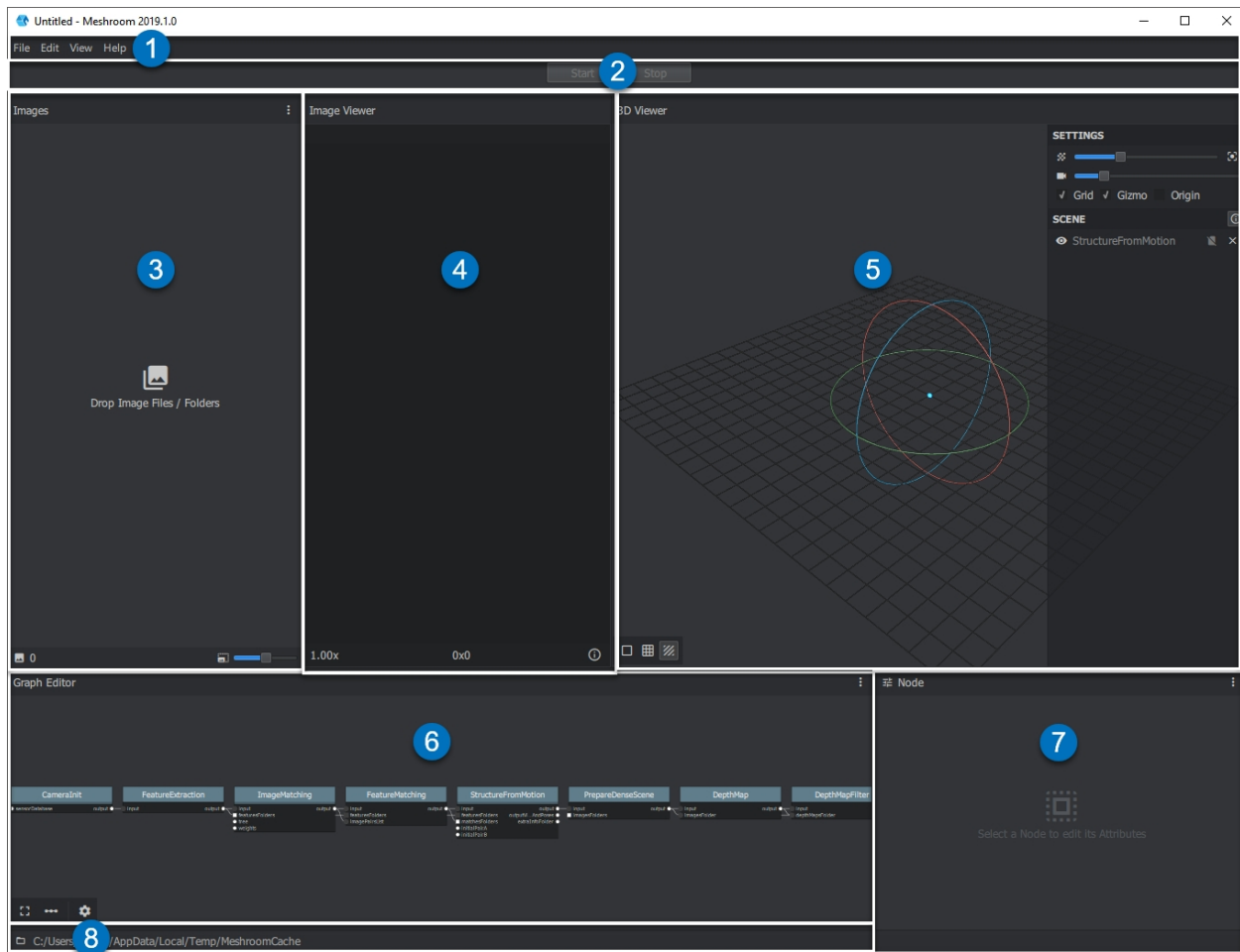
### The Graphical User Interface (GUI)


---

When you first start Meshroom, two windows open:

the main graphical user interface with different panes and in the background the Command-line interface window.

- 1 Menu bar: File / View / About
- 2 Start/Pause/Stop/(Submit) *processing with progress bar below*
- 3 Images *Pane*
- 4 Image Viewer *Pane*
- 5 3D Viewer *Pane*
- 6 Graph Editor *Pane*
- 7 Graph Editor Properties *Pane*
- 8 Cache Folder File Path (*where temp files and final results are stored*)



You can grab a Pane border  and move it to change the **pane** size.

## CHAPTER 4

---

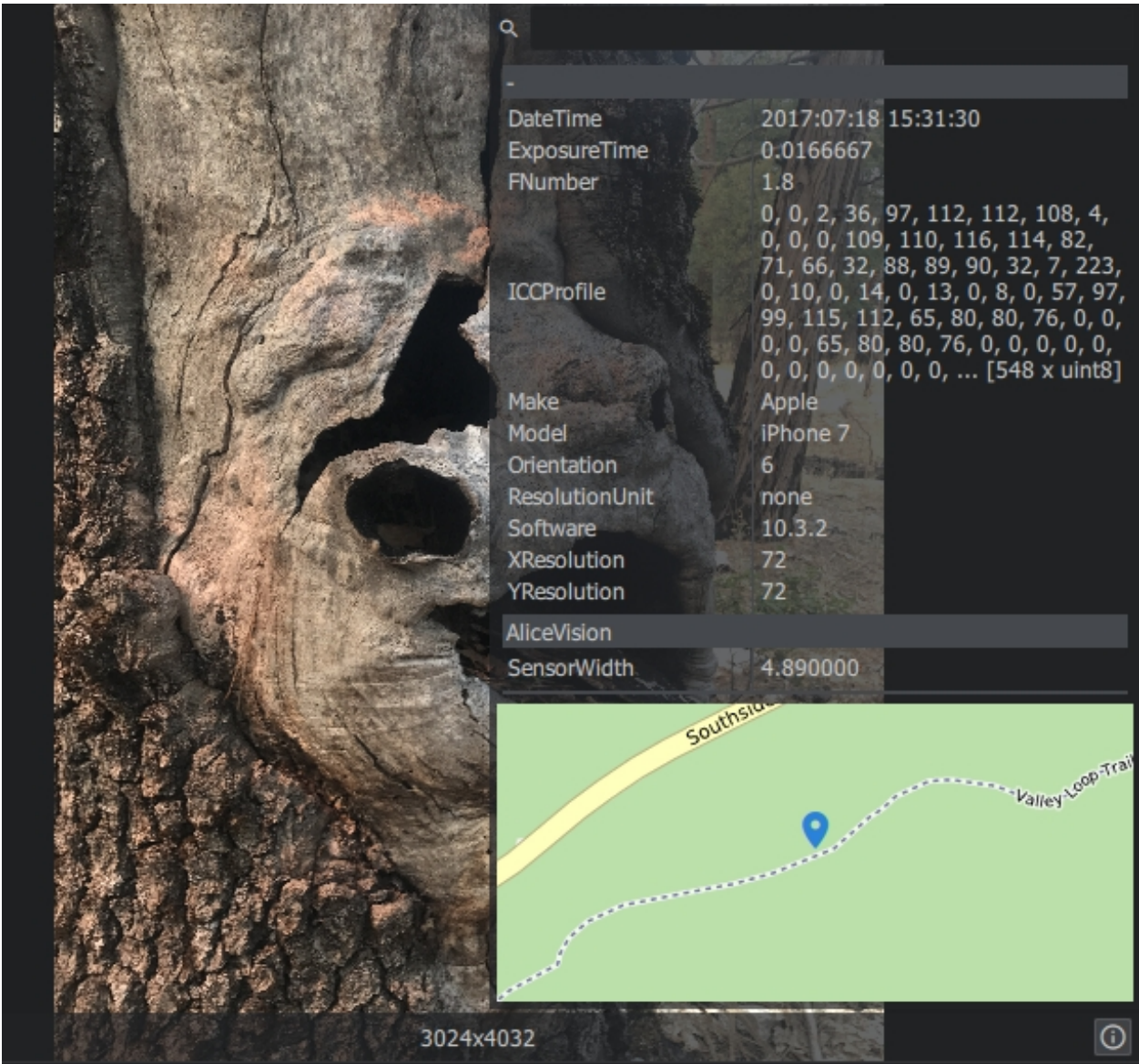
### Simple import

---

Drag-n-drop your images or your image folder into the **Images** pane on the left hand side.

You can preview the images in the **Image Viewer** pane. To display the image metadata click the (i) icon in the bottom right corner. For images with embedded GPS information an additional openstreetmap frame will be displayed.



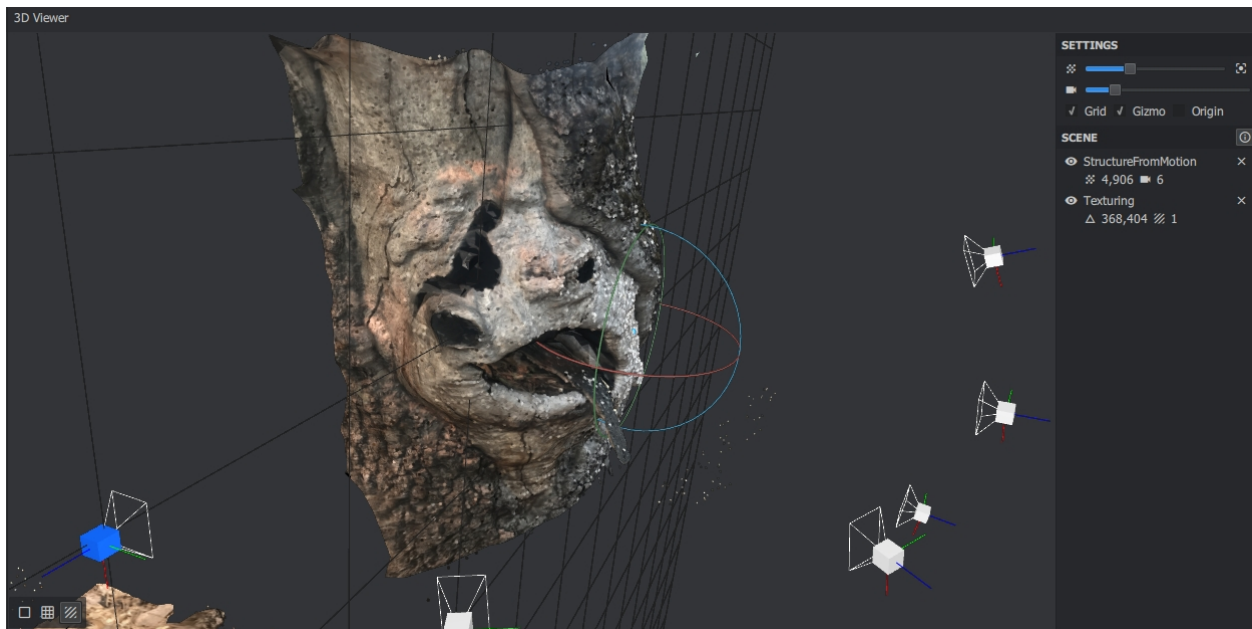


**Note:** If your images won't appear in the **Images** pane after you imported them, your camera was not recognized correctly. Make sure the EXIF data contains all relevant camera information. If the import still fails, your camera is not in the database or your image files are not valid.

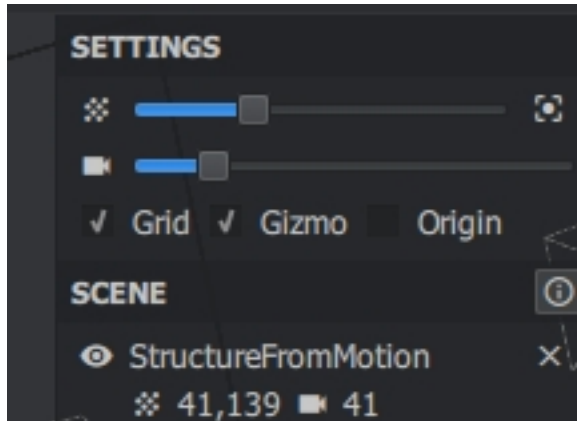


## CHAPTER 5

### 3D Viewer



The **3D Viewer** will preview the SfM Pointcloud, cameras and the Mesh preview. You can use your mouse or the rotate/scale toolbar on the left. You can hold **Shift** to pan. Press **F** to reset the view. **Double-click** to create a new rotation center for the Mesh. To display the final model, a button will appear on the bottom side to load the mesh (**Load model**). Uncheck the SfM layer for a better view. To refit the 3D-model to the new dimensions of the pane if you changed its size, right-click to display a menu with refitting options.



By default **StructureFromMotion** and **Texturing** results will be added to your **Scene** layers. You can add the outputs of other node variations to your Scene in the **3D Viewer** by double clicking on the nodes. Supported nodes: StructureFromMotion, Texturing, MeshDecimate, MeshDenoise, MeshResampling

**3D Model** The final 3D-Model will be saved in **Project Folder** → **MeshroomCache** → **Texturing**

By default it will be saved in the OBJ format. You can change it in the node settings.

---

**Note:** At the moment Meshroom does not support model realignment, so the model can be orientated upside down relative to the grid. You can change the orientation in another software like Meshlab.

---

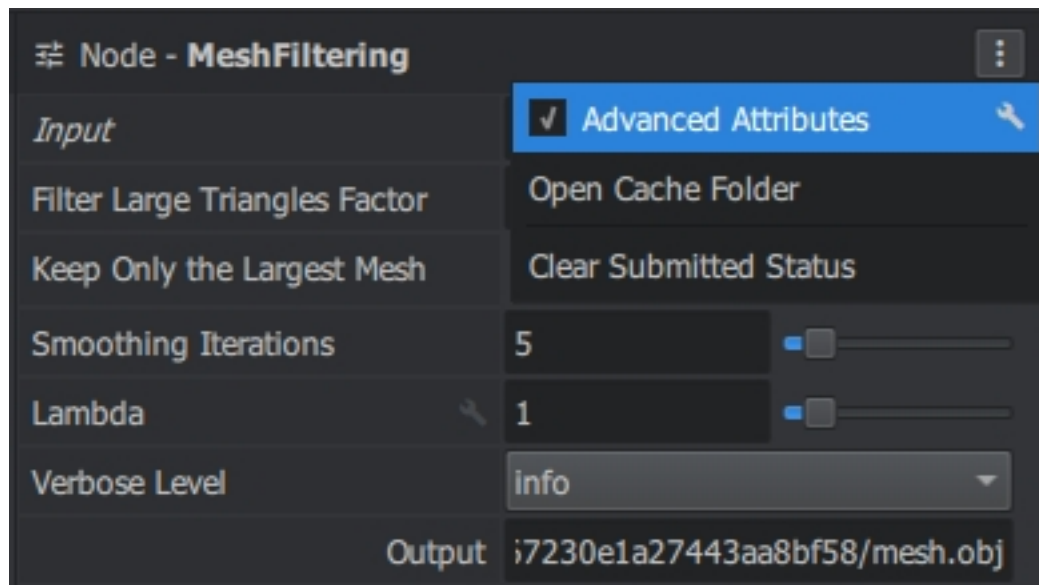
## CHAPTER 6

---

### Advanced Node parameters

---

This PR introduces the notion of “advanced” parameters on nodes. The goal is to separate experimental/debug/advanced from end-user attributes. On the UI side, the AttributeEditor has been redesigned and now provides an additional option to show/hide those advanced parameters.





## CHAPTER 7

---

### Start Reconstruction

---

Click the green **Start** button to start processing. To stop/pause click the **Stop** button. The progress will be kept.

There are two progress bars: the line below the **menu bar** indicating the overall progress and the other in the **Graph Editor** within the **nodes**. To get a detailed progress log, open the **CommandLine** window or click on the **node** you are interested in and go to the Log tab in the properties pane of the **Graph Editor**.

You can open the (Your-Project-Folder) -> **MeshroomCache** to see the output of each node. (Shortcut: Icon and path at the bottom left side of the main window)

A node folder contains the output of the node. By default Meshroom uses a unique id to name the output folders to prevent overwriting data and already computed results of the project can be reused.

**Example:** You are not satisfied with your first result and make changes to the **StructureFromMotion** node. The new output will be placed under a different name inside the **StructureFromMotion** Folder.

You can change the **name of the output folders** of your nodes by clicking on the node and changing the **Output Folder** name in the **Attributes tab** of the **Graph Editor** Properties pane.



---

### Augment Reconstruction

---

You can drag-n-drop additional images into the lower part of the **Images** Pane, called **Augment Reconstruction**. For each batch of images, a new **Group** will be created in the **Images** Pane. You can drop successive batches of N images in the **Images** Pane. for each batch of images the graph will branch.

You can use this method for complex scenes with multiple objects

---

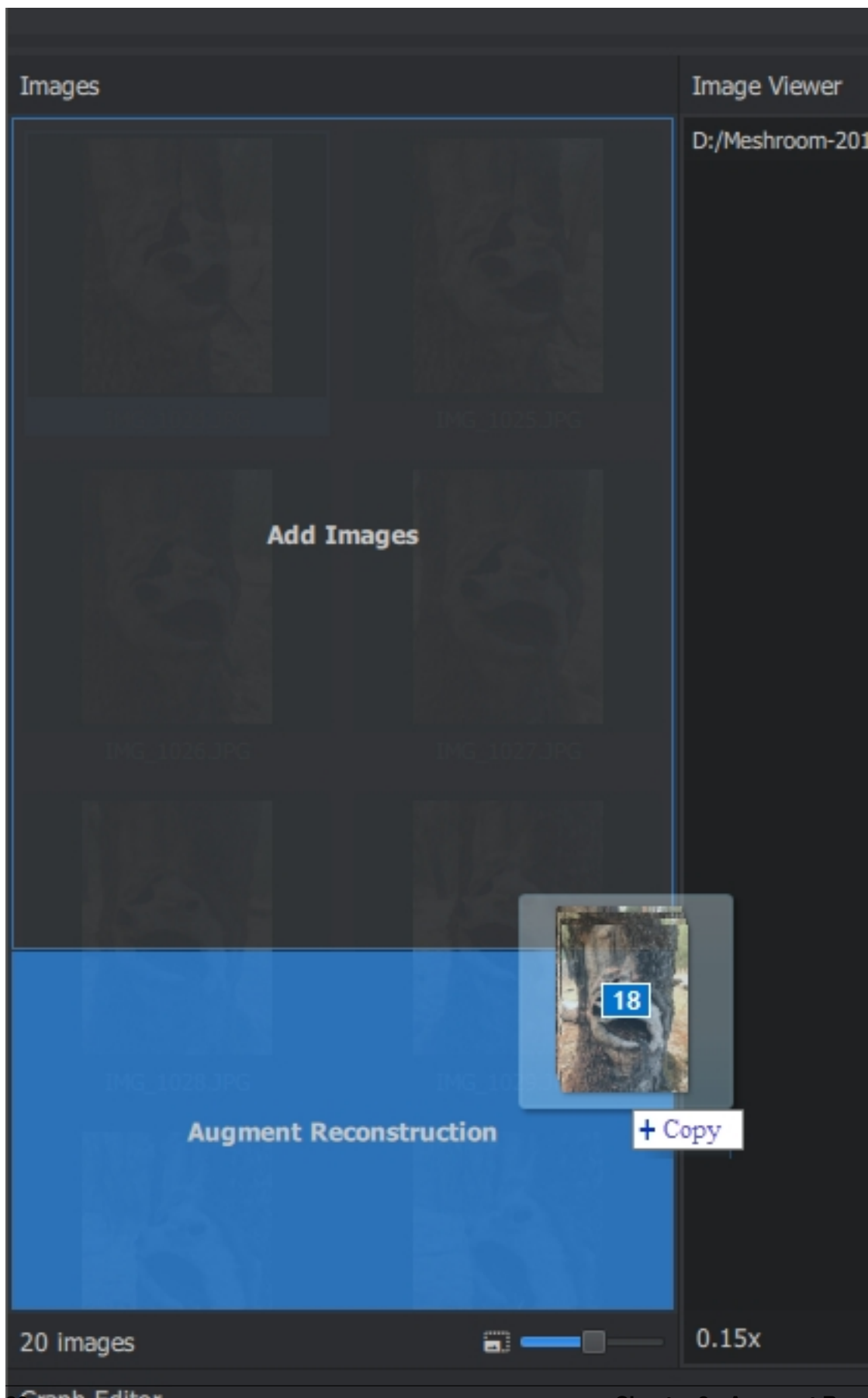
**Note:** Images can not be added while processing

---

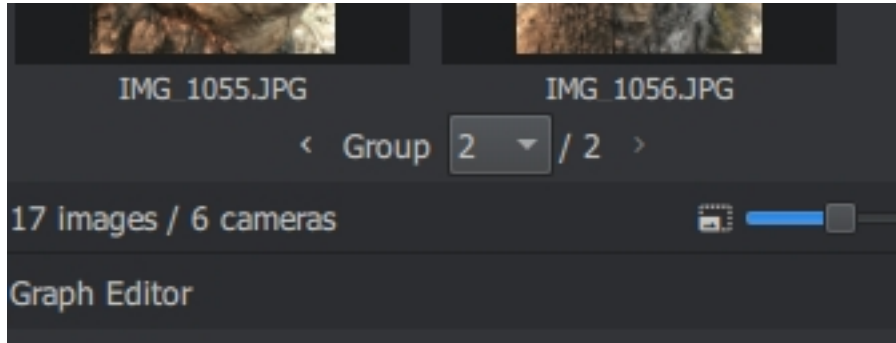
---

**Note:** The groups will be merged using the ImageMatchingMultiSfM node. Read the node description for details

---









## Live Reconstruction

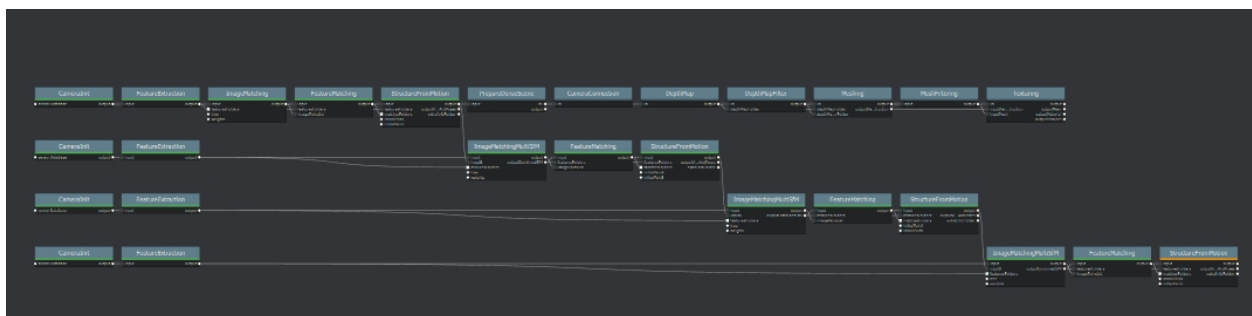
Live reconstruction is meant to be used along with a camera that can transfer images to a computer while shooting (using wifi, a wifi sd-card or Tethering). Meshroom can watch a folder for new images and successively augment previous SfM (point clouds + cameras) after each {Min. Images} per Step. This allows to get an iterative preview during shooting, e.g to see which areas of the dataset requires more coverage.

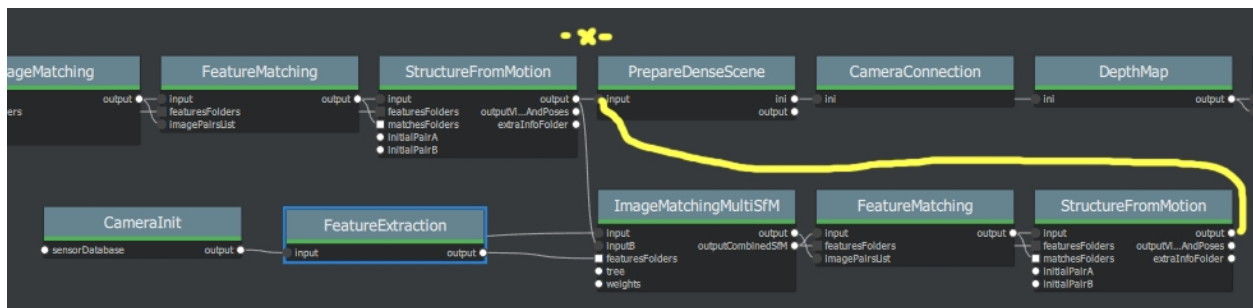
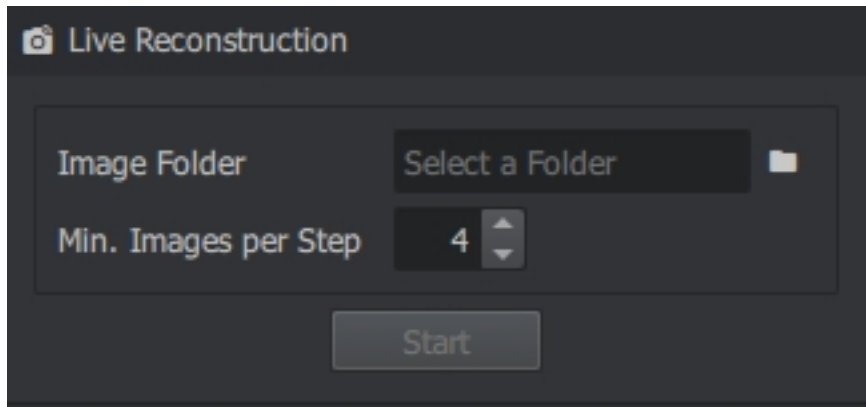
To enable **Live Reconstruction** go to the menu bar **View -> Live Reconstruction** A new **Live Reconstruction** pane will appear under the Images pane.

For each new import, a new **Image Group** inside the **Images** pane will be created. Also the **Graph Editor** updates the graph, adding nodes to process the newly added images and add them to the pipeline.

Select the **Image Folder** to watch and the minimum of new images folder to be imported per step. Click **Start** in the **Live Reconstruction** pane to start monitoring the selected folder for new files. You should then see in the graph one branch (from **CameraInit** to **StructureFromMotion**) for each batch of images. 1 The reconstruction process will stop at the last processed **StructureFromMotion** node and will not automatically go through the rest of the default pipeline. This is for practical reasons. The point cloud will update in real time with newly added images. Computing the mesh for every new image batch is not effective.

Once you complete the image capturing process, click **Stop** and disconnect the **PrepareDenseScene** node from the first **StructureFromMotion** node and connect it with the last **StructureFromMotion** node.





**Note:** The groups will be merged using the **ImageMatchingMultiSfM** node. Read the node description for details.

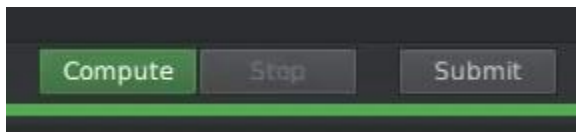
A demo video can be found here: [https://www.youtube.com/watch?v=DazLfZXU\\_Sk](https://www.youtube.com/watch?v=DazLfZXU_Sk)

## CHAPTER 10

---

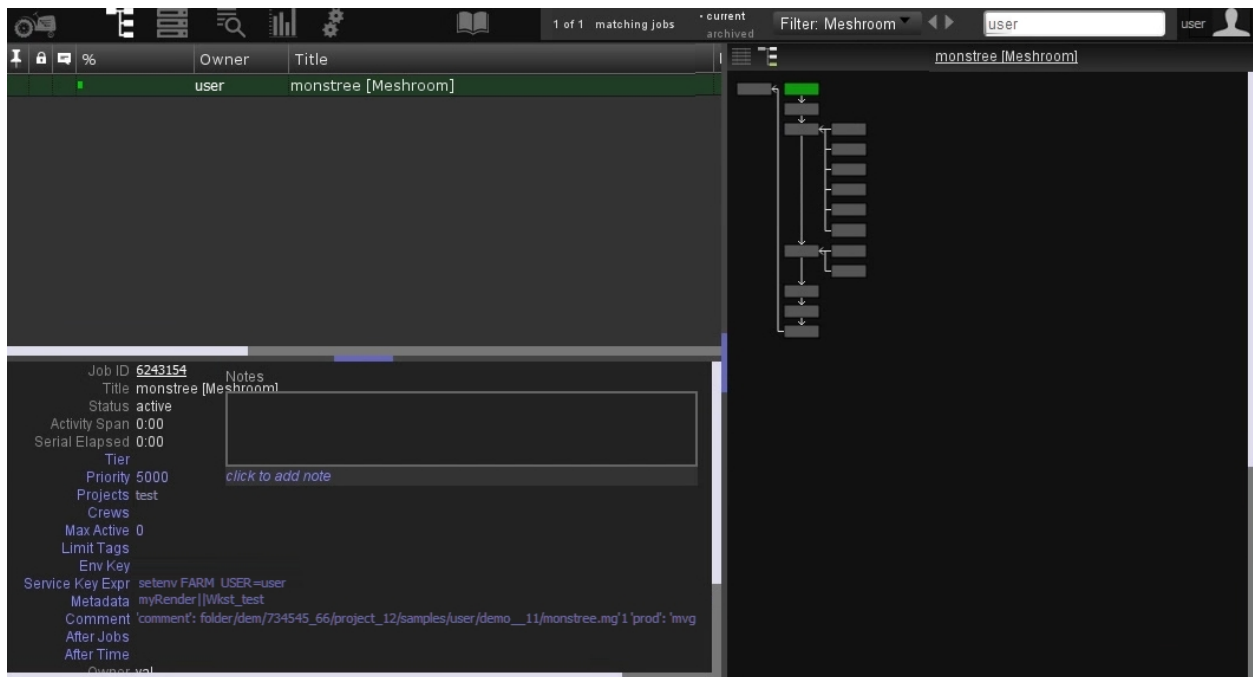
### External Reconstruction

---



Use this option when you compute externally after submission to a render farm from meshroom. (need to have access to a renderfarm and need the corresponding submitter).

This way, you can make use of external computing power. If you can not compute GPU nodes locally (no cuda) you can still submit them.



Available submitters:

- Pixar Renderman Tractor
- Fireworks (<https://materialsproject.github.io/fireworks/>)

WIP

# CHAPTER 11


---

## Import old Meshroom project


---

Projects created in an older version of Meshroom can be imported.

- CameraConnection node has been removed in v2019.1. You need to reconnect the neighboring nodes.
- With a new release of Meshroom, some nodes might require an update to the new version.
- Projects created in a newer version may become incompatible with an older version.



Compatibility issues detected - Meshroom 2018.2.0



**This project contains 8 node(s) incompatible with the current version of Meshroom.**

Project was created with Meshroom 2018.1.0.

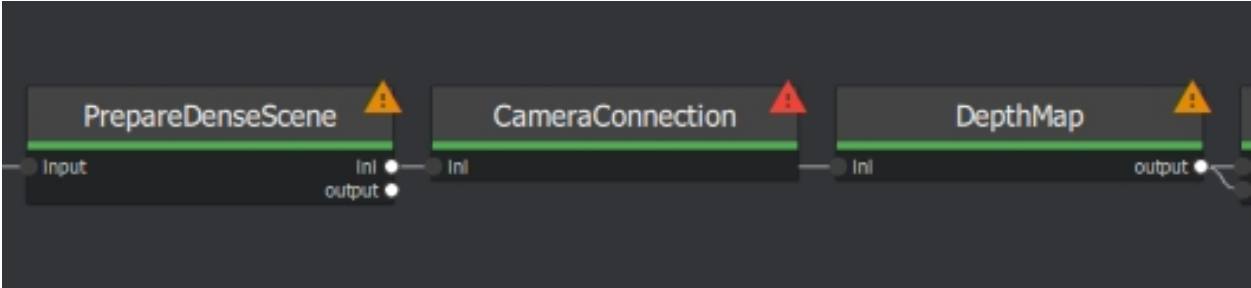
7 node(s) can be upgraded but this might invalidate already computed data.  
This operation is undoable and can also be done manually in the Graph Editor.

Node	Issue	Upgradable
CameraInit	Node version '1.0' conflicts with current version '2.0'.	✓
Meshing	Node version '1.0' conflicts with current version '2.0'.	✓
Texturing	Node attributes don't match node description.	✓
StructureFromMotion	Node version '1.0' conflicts with current version '2.0'.	✓
DepthMap	Node attributes don't match node description.	✓
DepthMapFilter	Node attributes don't match node description.	✓
CameraConnection	Unknown node type: CameraConnection.	✗
PrepareDenseScene	Node version '1.0' conflicts with current version '2.0'.	✓

Upgrade all possible nodes to current version ?

Yes

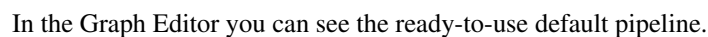
No





# Test Meshroom

Import the images in Meshroom by dropping them in the Images pane. There are different folders in the Monstree dataset: full (all images), mini6 (6 images) and mini3 (3 images) to test out.



The Graph Editor contains the processing nodes of your pipeline. For this project you do not need to change anything! In fact, for many projects the default pipeline delivers good results

You can zoom in or restructure the nodes. You can hold Shift to pan using the mouse. To insert new nodes right-click in the Graph Editor pane. For the Graph Editor use the buttons on the bottom left side of the pane to (re)order.

Before you start the reconstruction, save the project to the Monstree folder. (File ? Save as) (The HDD should have enough free space.)

---

### Reconstruction: How long does it take?

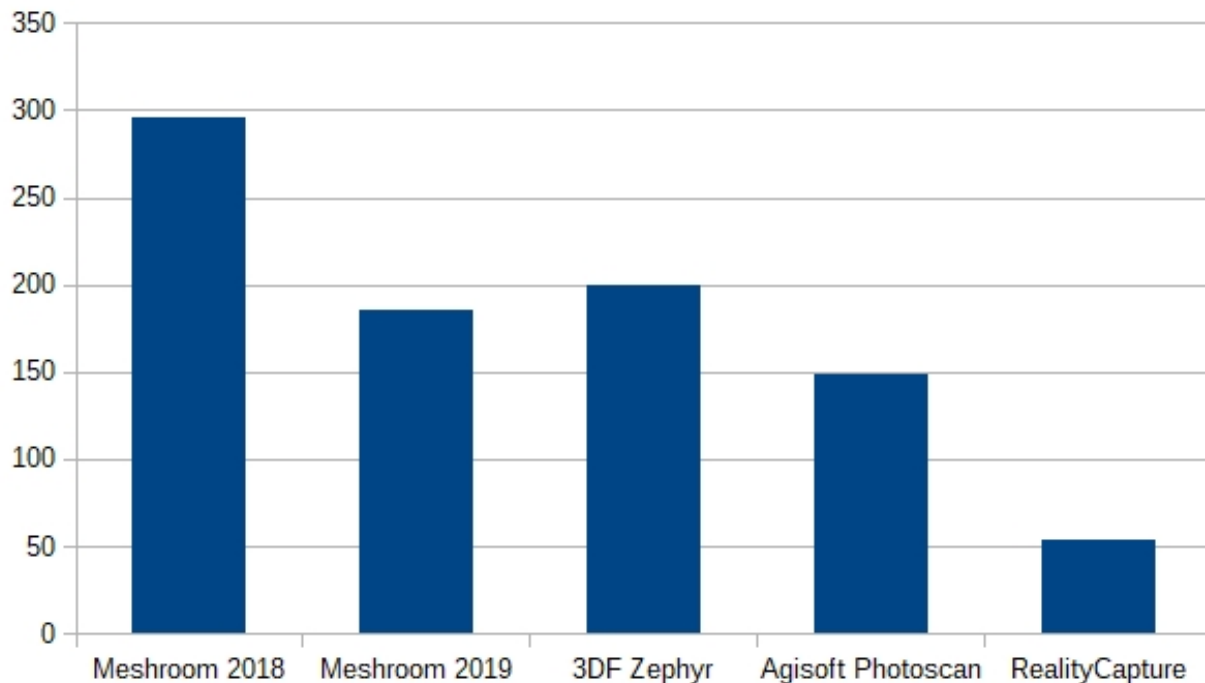
---

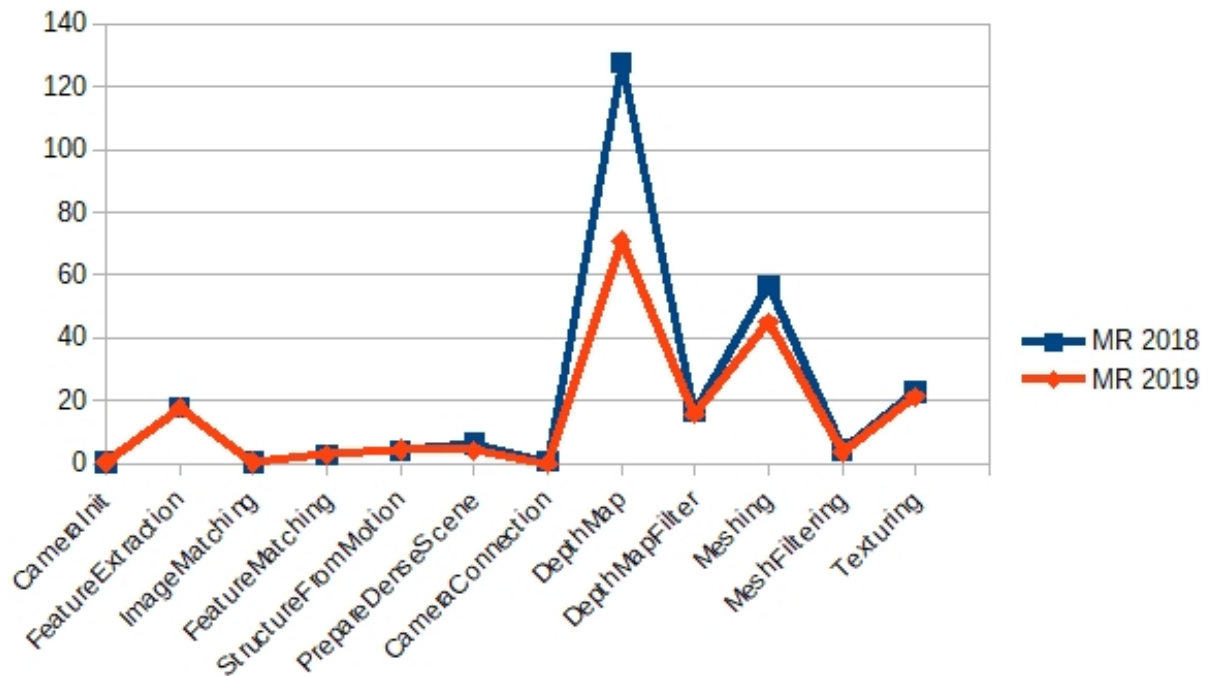
You can calculate with 30sec. per image on a computer with i7@2,9GHz, GTX1070 8GB, 32GB Ram.

Performance: % of overall processing time with default pipeline: ~38% DepthMap / ~24% Meshing

With the 2019.1.0 release, the reconstruction time has been reduced by ~30% compared to the 2018.1 release. The Cache Folder file size has been reduced by 20% Tested with the **Monstree Dataset**: (comparing only computing time, not quality) Computing time in seconds: (total MR2018 260s / MR2019 185s)

<https://web.archive.org/web/20181010161448/https://scanbox.xyz/blog/alicevision-opensource-photogrammetry/>





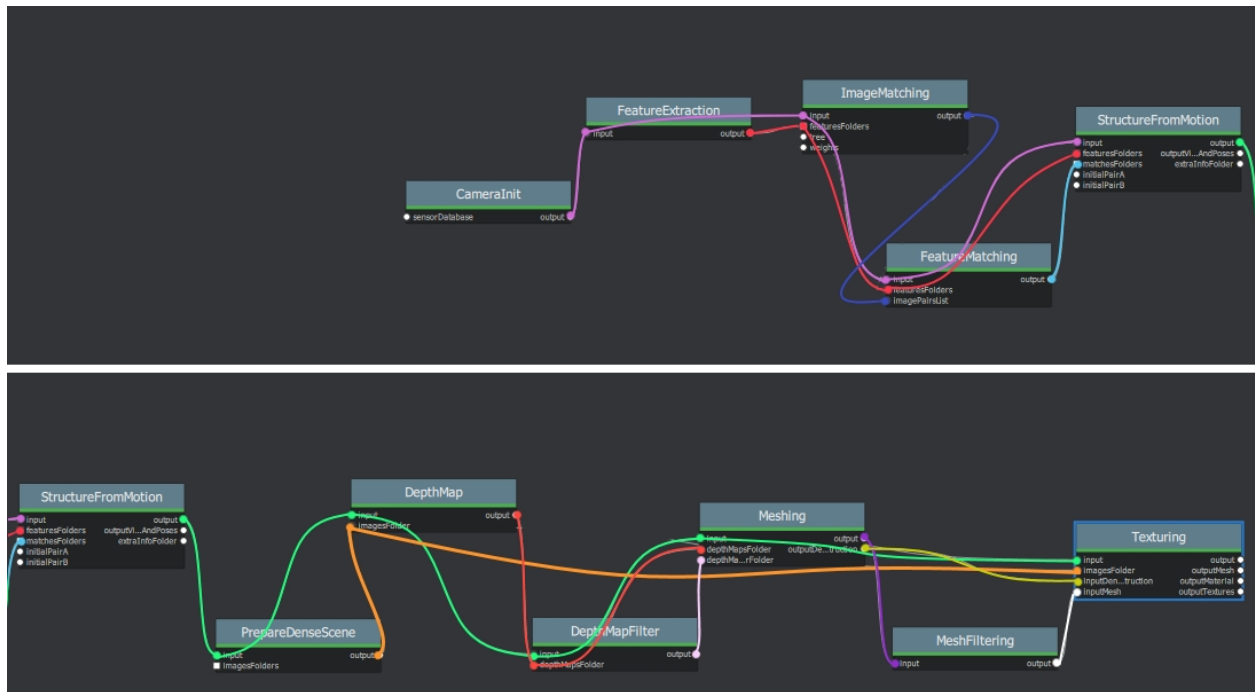
For a Full Pipeline Evaluation including the “Tanks and Temples” evaluation benchmark read **D5.4: Deliver 3D reconstruction benchmarks with dataset** available on <https://cordis.europa.eu/project/rcn/205980/results/en> in Documents, reports.

# CHAPTER 14

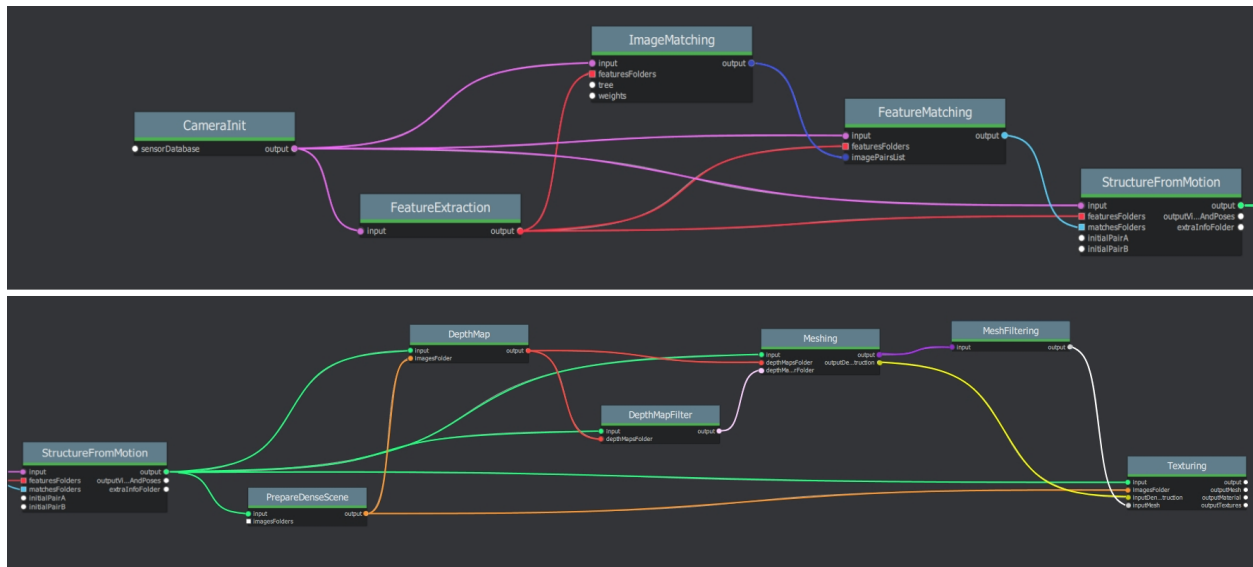
## Connect Nodes

### 14.1 Default graph

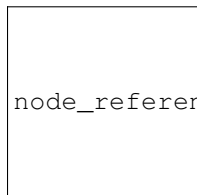
The node connections of the **default Graph** can be difficult to understand. The following images illustrate how the nodes are connected.



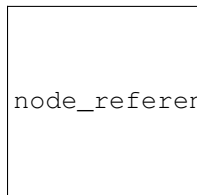
This image illustrates the default graph with node connections on the origin nodes:



## 14.2 Draft Meshing



node\_reference/connect-nodes/draft-meshingnode-graph.jpg



node\_reference/connect-nodes/draft-meshingnode-graph-color.jpg

Note: Some parameters are exposed for development purposes.

Nodes/features marked with an # are not supported/implemented in the current release

\* in default pipeline \*\*tested and working ? not tested

## 15.1 CameraCalibration (#)

### Description

Note: This node requires AliceVision compiled with opencv. Not included in the MR 2019.1 binary.

The internal camera parameters can be calibrated from multiple views of a checkerboard. This allows to retrieve focal length, principal point and distortion parameters. A detailed explanation is presented in [opencvCameraCalibration].

[opencvCameraCalibration]      [http://docs.opencv.org/3.0-beta/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](http://docs.opencv.org/3.0-beta/doc/tutorials/calib3d/camera_calibration/camera_calibration.html)

Table 1: settings

Name	Description
Input	Input images in one of the following form: – folder containing images – image sequence like “/path/to/seq. @.jpg” – video file
Pattern	Type of pattern (camera calibration patterns) - CHESSBOARD - CIRCLES - ASYMMETRIC_CIRCLES - ASYMMETRIC_CCTAG
Size	(Size of the Pattern) - Number of inner corners per one of board dimension like Width Height
Square Size	Size of the grid’s square cells (0-100mm)
Nb Distortion Coef	Number of distortion coefficient (0-5)
Max Frames	Maximal number of frames to extract from the video file (0-5)
Calib Grid Size	Define the number of cells per edge (0-50)
Max Calib Frames	Maximal number of frames to use to calibrate from the selected frames (0-1000)
Min Input Frames	Minimal number of frames to limit the refinement loop (0-100)
Max Total Average Error	Max Total Average Error (0-1)
Debug Rejected Img Folder	Folder to export delete images during the refinement loop
Debug Selected Img Folder	Folder to export debug images
Output	Output filename for intrinsic [and extrinsic] parameters (default filename cameraCalibration.cal)

## 15.2 CameraInit

### Description

-load image metadata and sensor information You can mix multiple cameras and focal lengths. The CameraInit will create groups of intrinsics based on the images metadata. It is still good to have multiple images with the same camera and same focal lengths as it adds constraints on the internal cameras parameters. But you can combine multiple groups of images, it will not decrease the quality of the final model.

Note: In some cases, some image(s) have no serial number to identify the camera/lens device. This makes it impossible to correctly group the images by device if you have used multiple identical (same model) camera devices. The reconstruction will assume that only one device has been used, so if 2 images share the same focal length approximation they will share the same internal camera parameters. If you want to use multiple cameras, add a corresponding serialnumber to the EXIF data.



Table 2: settings

Name	Description
Viewpoints Input viewpoints	<b>(1 Element for each loaded image)</b> ID Pose ID Image Path Intrinsic: Internal Camera Parameters (Intrinsic ID) Rig (-1 - 200) Rig Sub-Pose: Rig Sub-Pose Parameters (-1 - 200) Image Metadata: (list of metadata elements)
Intrinsic Camera Intrinsics	(1 Element for each loaded image) ID Initial Focal Length: Initial Guess on the Focal Length Focal Length: Known/Calibrated Focal Length Camera Type: pin- hole', 'radial1', 'radial3', 'brown', 'fisheye4' #Make: Camera Make (not included in this build, commented out) #Model: Camera Model #Sensor Width: Camera Sensor Width Width: Image Width (0-10000) Height: Image Height (0-10000) Serial Number: Device Serial Number (camera and lens combined) Principal Point: X (0-10000) Y(0-10000) DistortionParams: Distortion Parameters Locked(True/False): If the camera has been calibrated, the internal camera parameters (intrinsics) can be locked. It should improve robustness and speedup the reconstruction.
Sensor Database	Camera sensor width database path
Default Field Of View	Empirical value for the field of view in degree 45° (0°-180°)
Verbose Level	verbosity level (fatal, error, warning, info, debug, trace)
Output SfMData File	.../cameraInit.sfm

## Notes

Issue: structure from motion reconstruction appears distorted, and has failed to aligned some groups of cameras when loading images without focallength

Solution: Keep the "Focal Length" init value but set the "Initial Focal Length" to -1 if you are not sure of the value.

<https://github.com/alicevision/meshroom/issues/434>

## 15.3 CameraLocalization (?)

### Description

Based on the SfM results, we can perform camera localization and retrieve the motion of an animated camera in the scene of the 3D reconstruction. This is very useful for doing texture reprojection in other software as part of a texture clean up pipeline. Could also be used to leverage Meshroom as a 3D camera tracker as part of a VFX pipeline

<https://alicevision.github.io/#photogrammetry/localization>

Table 3: settings

Name	Description
SfM Data	The sfm_data.json kind of file generated by AliceVision
Media File	The folder path or the filename for the media to track
Visual Debug Folder	If a folder is provided it enables visual debug and saves all the debugging info in that folder
Descriptor Path	Folder containing the descriptors for all the images (ie the .desc.)
Match Desc Types	Descriptor types to use for the matching: 'sift', 'sift_float', 'sift_upright', 'akaze', 'akaze_liop', 'akaze_mldb', 'cctag3', 'cctag4', 'sift_ocv', 'akaze_ocv
Preset	Preset for the feature extractor when localizing a new image (low, medium, normal, high, ultra)
Resection Estimator	The type of /sac framework to use for resection (acransac, loransac)
Matching Estimator	The type of /sac framework to use for matching (acransac, loransac)
Calibration	Calibration file
Refine Intrinsics	Enable/Disable camera intrinsics refinement for each localized image
Reprojection Error	Maximum reprojection error (in pixels) allowed for resectioning. If set to 0 it lets the ACRansac select an optimal value (0.1 - 50)
Nb Image Match	[votree] Number of images to retrieve in database (1 - 1000)
Max Results	[votree] For algorithm AllResults, it stops the image matching when this number of matched images is reached. If 0 it is ignored (1 - 100)
Commonviews	[votree] Number of minimum images in which a point must be seen to be used in cluster tracking (2 - 50)
Votree	[votree] Filename for the vocabulary tree
Votree Weights	[votree] Filename for the vocabulary tree weights
Algorithm	[votree] Algorithm type: (FirstBest, AllResults)
Matching Error	[votree] Maximum matching error (in pixels) allowed for image matching with geometric verification. If set to 0 it lets the ACRansac select an optimal value (0 - 50)
Nb Frame Buffer Matching	[votree] Number of previous frame of the sequence to use for matching (0 = Disable) (0 - 100)
Robust Matching	[votree] Enable/Disable the robust matching between query and database images, all putative matches will be considered
N Nearest Key Frames	[cctag] Number of images to retrieve in the database Parameters specific for final (optional) bundle adjustment optimization of the sequence: (1-100)
Global Bundle	[bundle adjustment] If <code>-refineIntrinsics</code> is not set, this option allows to run a final global bundle adjustment to refine the scene
No Distortion	[bundle adjustment] It does not take into account distortion during the BA, it consider the distortion coefficients all equal to 0
No BA Refine Intrinsics	[bundle adjustment] It does not refine intrinsics during BA
Min Point Visibility	[bundle adjustment] Minimum number of observation that a point must have in order to be considered for bundle adjustment (2-50)
Output Alembic	Filename for the SfMData export file (where camera poses will be stored) desc.Node.internalFolder + 'trackedCameras.abc
Output JSON	Filename for the localization results as .json desc.Node.internalFolder + 'trackedCameras.json

## 15.4 CameraRigCalibration (?)

### Description

If a rig of cameras is used, we can perform the rig calibration. We localize cameras individually on the whole sequence. Then we use all valid poses to compute the relative poses between cameras of the rig and choose the more stable value across the images. Then we initialize the rig relative pose with this value and perform a global Bundle Adjustment on all the cameras of the rig. When the rig is calibrated, we can use it to directly localize the rig pose from the synchronized multi-cameras system with [Kneip2014] approaches.

..The rig calibration find the relative poses between all cameras used. It takes a point cloud as input and can use both CCTag and SIFT features for localization. The implication is that all cameras must see features (either SIFT or CCTag) that are part of the point cloud, but they do not have to observe overlapping regions. (See:POPART: Previz for Onset Production Adaptive Realtime Tracking)

“Given the position of the tracked reference frame relative to the motion capture system and the optical reference frames it is possible to retrieve the transformation between the tracked and the optical reference frames”<sup>1</sup> “In practice, it is particularly difficult to make the tracked frame coincident with the camera optical frame, thus a calibration procedure is needed to estimate this transformation and achieve the millimetric accuracy” [Chiodini et al. 2018]

[Chiodini et al. 2018] Chiodini, Sebastiano & Pertile, Marco & Giubilato, Riccardo & Salvioli, Federico & Barera, Marco & Franceschetti, Paola & Debei, Stefano. (2018). Camera Rig Extrinsic Calibration Using a Motion Capture System. 10.1109/MetroAeroSpace.2018.8453603. [https://www.researchgate.net/publication/327513182\\_Camera\\_Rig\\_Extrinsic\\_Calibration\\_Using\\_a\\_Motion\\_Capture\\_System](https://www.researchgate.net/publication/327513182_Camera_Rig_Extrinsic_Calibration_Using_a_Motion_Capture_System)

<https://alicevision.github.io/#photogrammetry/localization>

[Kneip2011] A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation. L. Kneip, D. Scaramuzza, R. Siegwart. June 2011

[Kneip2013] Using Multi-Camera Systems in Robotics: Efficient Solutions to the NPnP Problem L. Kneip, P. Furgale, R. Siegwart. May 2013

[Kneip2014] OpenGV: A unified and generalized approach to real-time calibrated geometric vision, L. Kneip, P. Furgale. May 2014.

[Kneip2014] Efficient Computation of Relative Pose for Multi-Camera Systems. L. Kneip, H. Li. June 2014

Table 4: settings

Name	Description									
SfM Data	The sfmData file									
Media Path	The path to the video file	the folder of the image sequence or a text file (one image path per line) for each camera of the rig (eg. -media-path /path/to/cam1.mov /path/to/cam2.mov)								
Camera Intrinsic	The intrinsics calibration file for each camera of the rig. (eg. -cameraIntrinsics /path/to/calib1.txt /path/to/calib2.txt)									
Export	Filename for the alembic file containing the rig poses with the 3D points. It also saves a file for each camera named 'filename.cam##.abc (trackedcameras.abc)									
Descriptor Path	Folder containing the .desc									
Match Descriptor Types	The descriptor types to use for the matching “sift”	'sift_float'	'sift_upright'	'sift_upright'	'akaze' 'akaze' 'cc-tag3' 'cc-tag4'	'sift_ocv'	'akaze_ocv'			
Preset	Preset for the feature extractor when localizing a new image (low	medium	normal	high	ultra)					
Resection Estimator	The type of /sac framework to use for resection (acransac	loransac)								
Matching Estimator	The type of /sac framework to use for matching (acransac	loransac)								
Refine Intrinsic	Enable/Disable camera intrinsics refinement for each localized image									
Reprojection Error	Maximum reprojection error (in pixels) allowed for resectioning. If set to 0 it lets the ACRansac select an optimal value. (0 - 10)									
Max	Maximum number of									

**Voctree Weights:** <http://www.ipol.im/pub/art/2018/199/> voctree (optional): For larger datasets (>200 images), greatly improves image matching performances. It can be downloaded here. <https://github.com/fragofer/voctree> You need to specify the path to vlfeat\_K80L3.SIFT.tree in **Voctree**.

## 15.5 CameraRigLocalization (?)

### Description

This node retrieves the transformation between the tracked and the optical reference frames.(?) <https://alicevision.github.io/#photogrammetry/localization>

Table 5: settings

Name	Description									
SfM Data	The sfmData file									
Media Path	The path to the video file	the folder of the image sequence or a text file (one image path per line) for each camera of the rig (eg. -media-path /path/to/cam1.mov /path/to/cam2.mov)								
Rig Calibration File	The file containing the calibration data for the rig (subposes)									
Camera Intrinsic	The intrinsics calibration file for each camera of the rig. (eg. -cameraIntrinsics /path/to/calib1.txt /path/to/calib2.txt)									
Descriptor Path	Folder containing the .desc									
Match Descriptor Types	The descriptor types to use for the matching “(sift’	‘sift_float’	‘sift_upright’	‘sift_upright’	‘akaze’	‘akaze’	‘akaze’	‘cc-tag3’	‘cc-tag4’	‘sift_ocv’
Preset	Preset for the feature extractor when localizing a new image “(low	medium	normal	high	ultra)	“				
Resection Estimator	The type of /sac framework to use for resection “(acransac	loransac)								
Matching Estimator	The type of /sac framework to use for matching “(acransac	loransac)								
Refine Intrinsic	Enable/Disable camera intrinsics refinement for each localized image									
Reprojection Error	Maximum reprojection error (in pixels) allowed for resectioning. If set to 0 it lets the ACRansac select an optimal value (0 - 10)									
Use Lo	Enable/Disable the naive method for rig localiza-									

## 15.6 ConvertSfMFormat

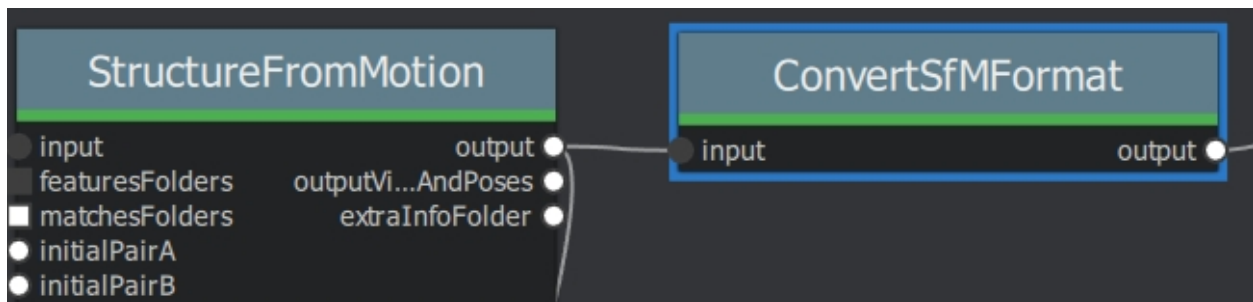
### Description

- creates abc', 'sfm', 'json', 'ply', 'baf' SfM File from SfMData file

Table 6: settings

Name	Description									
Input	SfMData file									
SfM File Format	SfM File Format “(output file extension: abc’	‘sfm’	‘json’	‘ply’	‘baf’	“				
Descriptor Types	Descriptor types to keep. “‘sift’	‘sift_flow’	‘sift_upright’	‘hough’	‘akaze’	‘laplace’	‘nndb’	‘cc-tag3’	‘cc-tag4’	‘sift_ocv’
Image id	Image id									
Image White List	image white list (uids or image paths).									
Views	Export views									
Intrinsics	Export intrinsics									
Extrinsics	Export extrinsics									
Structure	Export structure									
Observations	Export observations									
Verbose Level	verbosity level “(fatal	error	warning	info	debug	trace’	“			
Output	Path to the output SfM Data file. (desc.Node.internalFolder + ‘sfm.{fileExtension}’)									

Input nodes: StructureFromMotion:output->input:ConvertSfMFormat



### Can I convert between Openmvj and alicevision SfM formats?

OpenMVG and AliceVision json formats are very similar in the structure but not compatible right away as openmvj is a data serialization file among other things. <https://github.com/alicevision/AliceVision/issues/600>

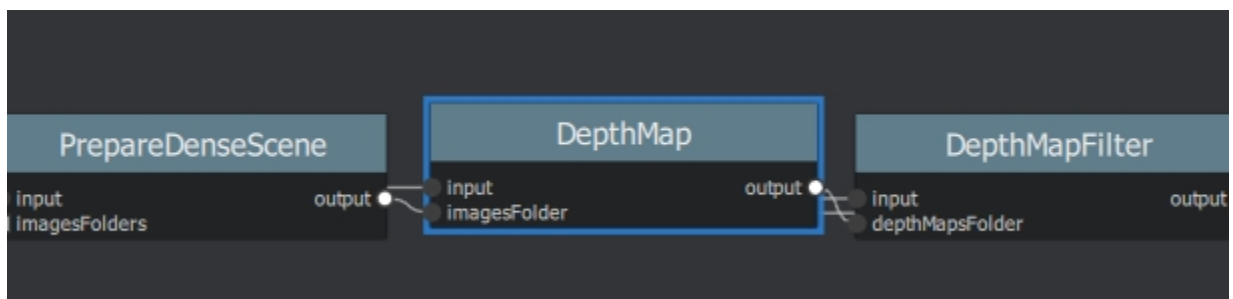
## 15.7 DepthMap

### Description

Table 7: settings

Name	Description					
MVS Configuration File:	SfMData file.					
Images Folder	Use images from a specific folder instead of those specify in the SfMData file. Filename should be the image uid.					
Downscale	Image downscale factor “(1	2	4	8	16)”	
Min View Angle	Minimum angle between two views. “(0.0	10.0	0.1)”			
Max View Angle	Maximum angle between two views. “(10.0	120.0	1)”			
SGM: Nb Neighbour Cameras	Semi Global Matching: Number of neighbour cameras (1 - 100)					
SGM: WSH: Semi Global Matching	Half-size of the patch used to compute the similarity (1 - 20)					
SGM: GammaC	Semi Global Matching: GammaC Threshold (0 - 30)					
SGM: GammaP	Semi Global Matching: GammaP Threshold (0 - 30)					
Refine: Number of samples	(1 - 500)					
Refine: Number of Depths	(1 - 100)					
Refine: Number of Iterations	(1 - 500)					
Refine: Nb Neighbour Cameras	Refine: Number of neighbour cameras. (1 - 20)					
Refine: WSH	Refine: Half-size of the patch used to compute the similarity. (1 - 20)					
Refine: Sigma	Refine: Sigma Threshold (0 - 30)					
Refine: GammaC	Refine: GammaC Threshold. (0 - 30)					
Refine: GammaP	Refine: GammaP threshold. (0 - 30)					
Refine: Tc or Rc pixel size	Use minimum pixel size of neighbour cameras (Tc) or current camera pixel size (Rc)					
Verbose Level	verbosity level (fatal	error	warning	info	debug	trace)
Output	Output folder for generated depth maps					

default:





## 15.8 DepthMapFilter

### Description

The original depth maps will not be entirely consistent. Certain depth maps will claim to see areas that are occluded by other depth maps. The DepthMapFilter step isolates these areas and forces depth consistency.

Table 8: settings

Name	Description					
Input	SfMData file					
Depth Map Folder	Input depth map folder					
Number of Nearest Cameras	Number of nearest cameras used for filtering 10 (0 - 20)					
Min Consistent Cameras	Min Number of Consistent Cameras 3 (0 - 10)					
Min Consistent Cameras Bad Similarity	Min Number of Consistent Cameras for pixels with weak similarity value 4 (0 - 10)					
Filtering Size in Pixels	Filtering size in Pixels (0 - 10)					
Filtering Size in Pixels Bad Similarity	Filtering size in pixels (0 - 10)					
Verbose Level	verbosity level (fatal	er- ror	warn- ing	info	de- bug	trace)
Output	Output folder for generated depth maps					

**Min Consistent Cameras** lower this value if the Meshing node has 0 depth samples input

**View Output** open output folder and view EXR files

## 15.9 ExportAnimatedCamera

### Description

creates an Alembic animatedCamera.abc file from SfMData (e.g. for use in 3D Compositing software)

Table 9: settings

Name	Description					
Input SfMData	SfMData file containing a complete SfM					
SfMData Filter	A SfMData file use as filter					
Export Undistorted Images	Export Undistorted Images value=True					
Undistort Image Format	Image file format to use for undistorted images “(*.jpg	*.jpg	*.tif	*.exr (half))“		
Verbose Level	Verbosity level “(fatal	er- ror	warn- ing	info	de- bug	trace)”
Output filepath	Output filepath for the alembic animated camera					
Output Camera Filepath	Output filename for the alembic animated camera internalFolder + ‘camera.abc’					

SfM->ExportAnimatedCamera .. details <https://www.youtube.com/watch?v=ldhdEmGLZhY>

## 15.10 ExportMaya

### Description

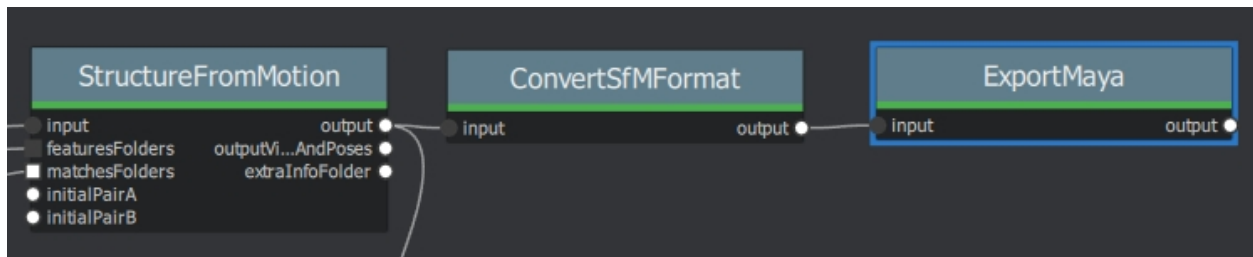
Mode for use with MeshroomMaya plugin.

The node “ExportMaya” exports the undistorted images. This node has nothing dedicated to Maya but was used to import the data into our MeshroomMaya plugin. You can use the same to export to Blender.

Table 10: settings

Name	Description
Input SfM Data	sfm.sfm or sfm.abc
Output Folder	Folder for MeshroomMaya output: undistorted images and thumbnails

ExportMaya: requires .sfm or .abc as input from ConvertSfMFormat



## 15.11 FeatureExtraction

### Description

---

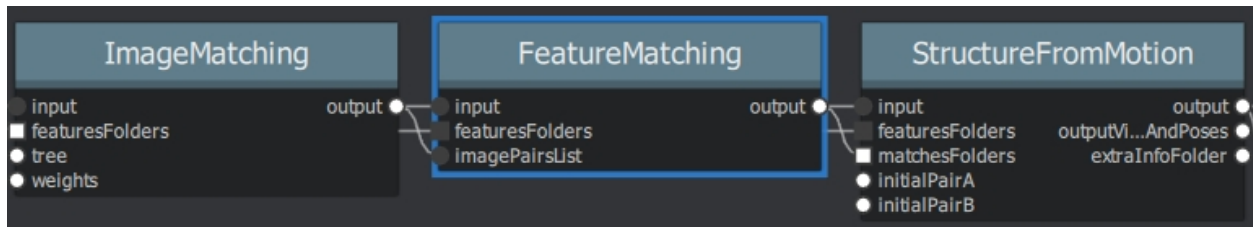
## 15.12 FeatureMatching

### Description

---

Table 11: settings

Name	Description				
In-put	SfMData file				
Fea- tures Folder					
Fea- tures Fold- ers	Folder(s) containing the extracted features and descriptors				
Im- age Pairs List	Path to a file which contains the list of image pairs to match				
De- scriber Types	Describer types used to describe an image <code>**sift**</code> / <code>'sift_float'</code> / <code>'sift_upright'</code> / <code>'akaze'</code> / <code>'akaze_liop'</code> / <code>'akaze_mldb'</code> / <code>'cctag3'</code> / <code>'cctag4'</code> / <code>'sift_ocv'</code> / <code>'akaze_ocv'</code>				
Pho- to- met- ric Match- ing Method	For Scalar based regions descriptor ' * BRUTE_FORCE_L2: L2 BruteForce matching' ' * ANN_L2: L2 Approximate Nearest Neighbor matching ' * CASCADE_HASHING_L2: L2 Cascade Hashing matching ' * FAST_CASCADE_HASHING_L2: L2 Cascade Hashing with precomputed hashed regions (faster than CASCADE_HASHING_L2 but use more memory) 'For Binary based descriptor ' * BRUTE_FORCE_HAMMING: BruteForce Hamming matching'				
Ge- o- met- ric Es- ti- ma- tor	Geometric estimator: (acransac: A-Contrario Ransac // loransac: LO-Ransac (only available for fundamental_matrix model)				
Ge- o- met- ric Fil- ter Type	Geometric validation method to filter features matches: <code>**fundamental_matrix**</code> // <code>essential_matrix</code> / <code>homography_matrix</code> /// <code>homography_growing</code> // <code>no_filtering</code>				
Dis- tance Ra- tio	Distance ratio to discard non meaningful matches 0.8 (0.0 - 1)				
Max It- era- tion	Maximum number of iterations allowed in ransac step 2048 (1 - 20000)				
Max Matches	Maximum number of matches to keep (0 - 10000)				
Save Pu- tative Matches	putative matches (True/False)				
15.12. Guided Match-	<b>FeatureMatching</b> the found model to improve the pairwise correspondences (True/False)				47

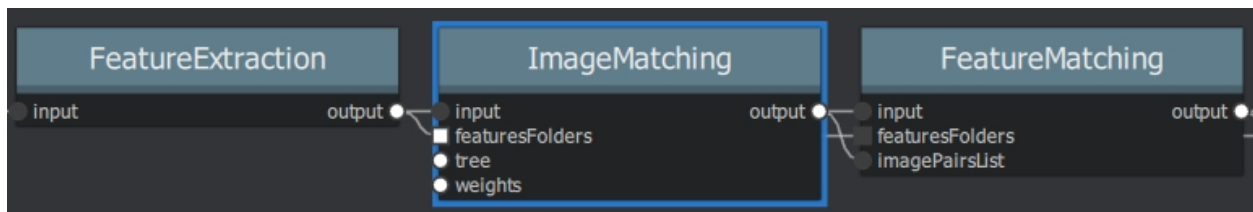


## 15.13 ImageMatching

### Description

Table 12: settings

Name	Description					
Image	SfMData file					
Features Folders	Folder(s) containing the extracted features and descriptors					
Tree	Input name for the vocabulary tree file ALICEVISION_VOCTREE					
Weights	Input name for the weight file	if not provided the weights will be computed on the database built with the provided set				
Minimal Number of Images	Minimal number of images to use the vocabulary tree. If we have less features than this threshold	we will compute all matching combinations				
Max De-descriptors	Limit the number of descriptors you load per image. Zero means no limit					
Nb Matches	The number of matches to retrieve for each image (If 0 it will retrieve all the matches) 50 (0-1000)					
Verbose Level	verbosity level (fatal	error	warn- ing	info	de- bug	trace)
Output List File	Filepath to the output file with the list of selected image pairs					



## 15.14 ImageMatchingMultiSfM

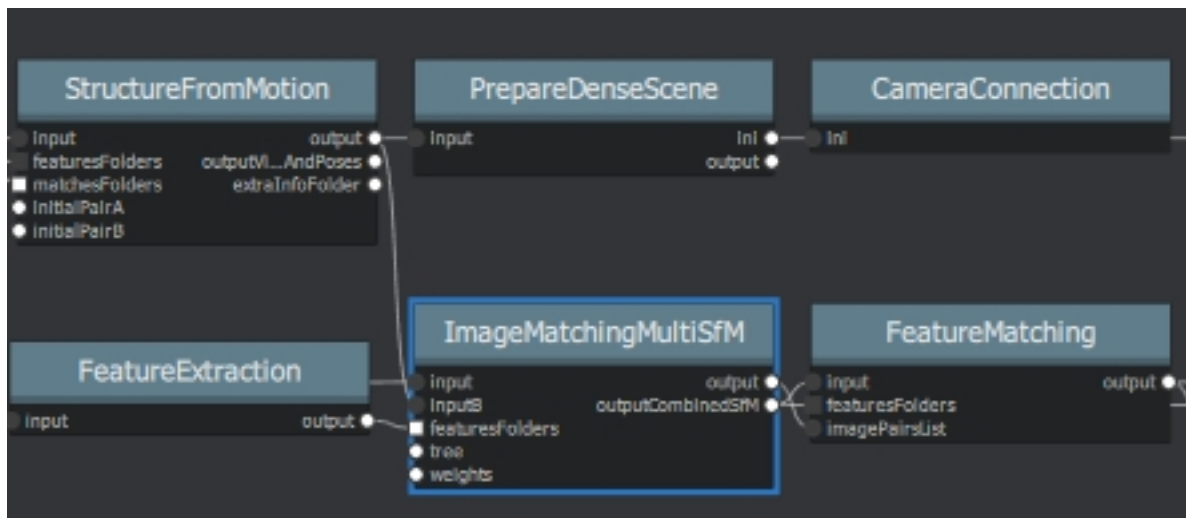
### Description

This node can combine image matching between two input SfMData.

Used for **Live Reconstructin** and **Augmentation**

Table 13: settings

Name	Description
Input A	SfMData file
Input B	SfMData file
Features Folders	Folder(s) containing the extracted features and descriptors
Tree	Input name for the vocabulary tree file ALICEVISION_VOCTREE
Weights	Input name for the weight file if not provided the weights will be computed on the database built with the provided set
Matching Mode	The mode to combine image matching between the input SfMData A and B: $a/a+a/b$ for A with A + A with B. $a/ab$ [ $'a/a+a/b'$ // $'a/ab'$ // $'a/b'$ ]
Minimal Number of Images	Minimal number of images to use the vocabulary tree. If we have less features than this threshold we will compute all matching combinations
Max Descriptors	Limit the number of descriptors you load per image. Zero means no limit 500 (0-100000)
Nb Matches	The number of matches to retrieve for each image (If 0 it will retrieve all the matches) 50 (0-1000)
Verbose Level	verbosity level (fatal // error // warning // info // debug // trace)
Output List File	Filepath to the output file with the list of selected image pairs
Output Combined SfM	Path for the combined SfMData file internalFolder + 'combineSfM.sfm'



## 15.15 KeyframeSelection

**Description** Note: This is an experimental node for keyframe selection in a video, which removes too similar or too blurry images. This node is not yet provided in the binaries as it introduces many dependencies. So if you built it by yourself, you can test the KeyframeSelection node. It is not yet fully integrated into Meshroom, so you have to manually drag&drop the exported frames to launch the reconstruction (instead of just adding a connection in the graph) <https://github.com/alicevision/meshroom/issues/232>

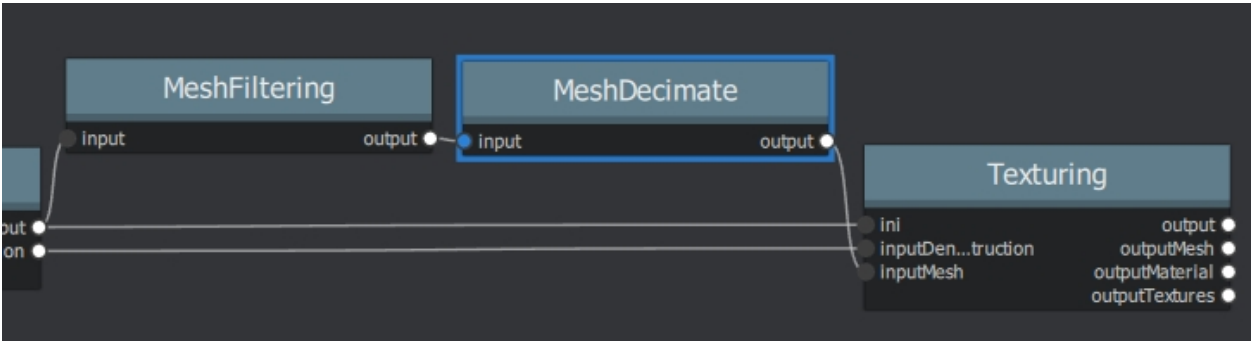
## 15.16 MeshDecimate

**Description**

Simplify your mesh to reduce mesh size without changing visual appearance of the model.

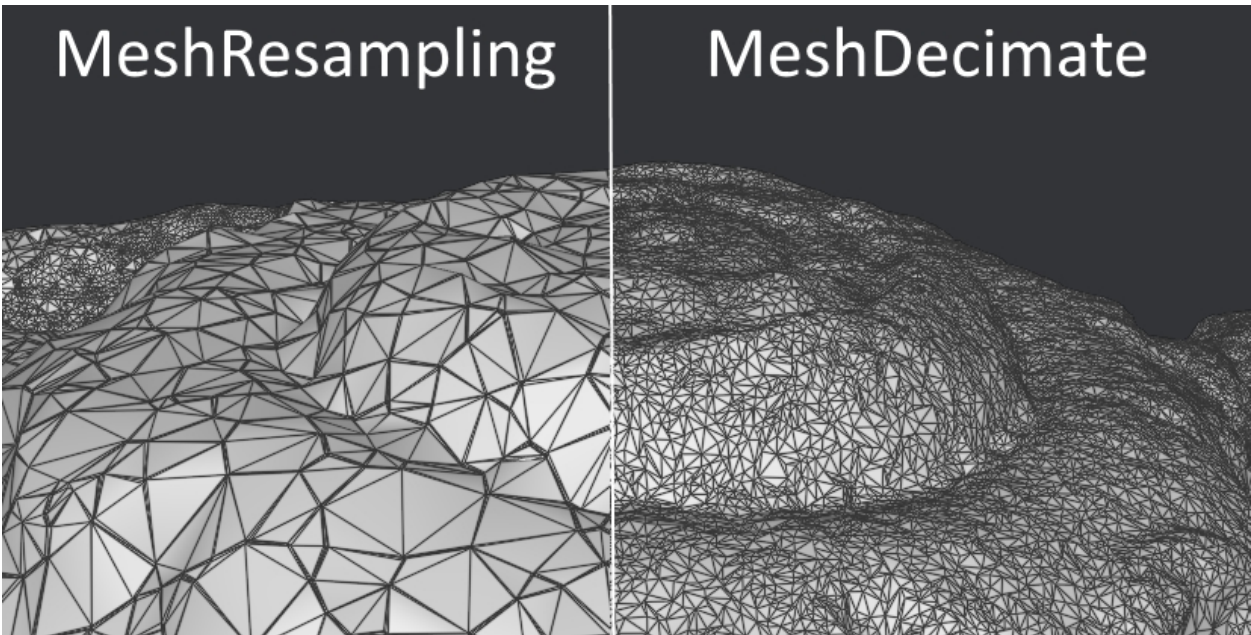
Table 14: settings

Name	Description
Input Mesh (OBJ file format)	
Simplification factor	Simplification factor 0.5 (0 - 1)
Fixed Number of Vertice	Fixed number of output vertices 0 (0 - 1 000 000)
Min Vertices	Min number of output vertices 0 (0 - 1 000 000)
Max Vertices	Max number of output vertices 0 (0 - 1 000 000)
Flip Normals	Option to flip face normals 'It can be needed as it depends on the vertices order in triangles and the convention change from one software to another. (True/False)
Verbose Level	verbosity level (fatal // error // warning // info // debug // trace)
Output mesh	Output mesh (OBJ file format) internalFolder + 'mesh.obj

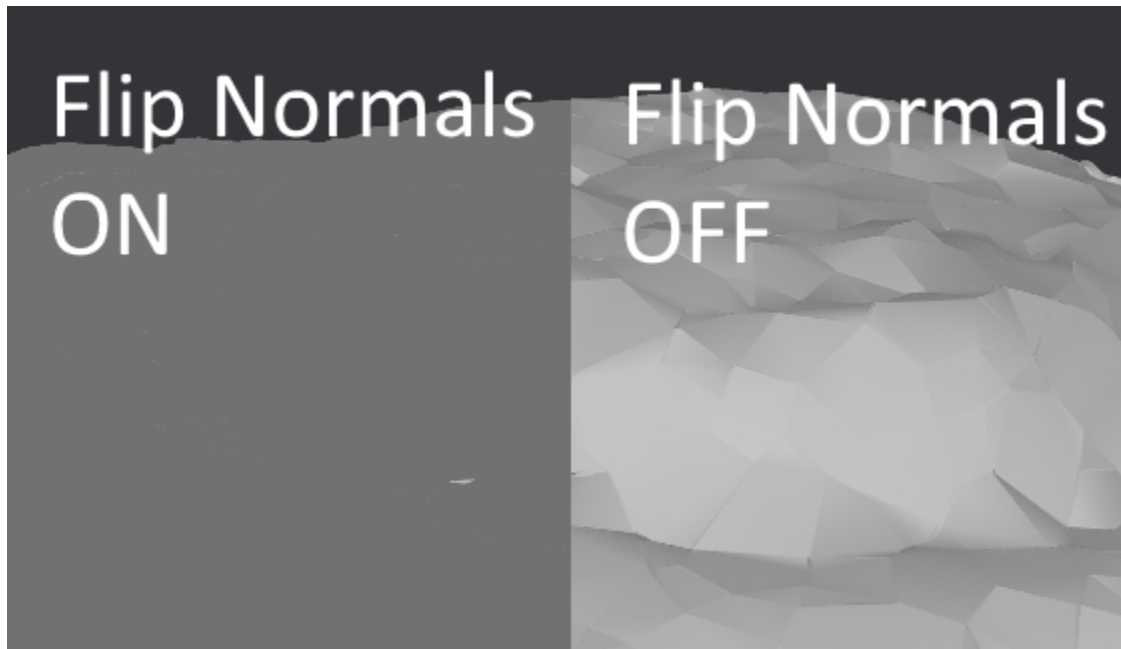


or Meshing->MeshDecimate->MeshFiltering?

Comparison MeshDecimate and MeshResampling



Flip Normals



## 15.17 MeshDenoising

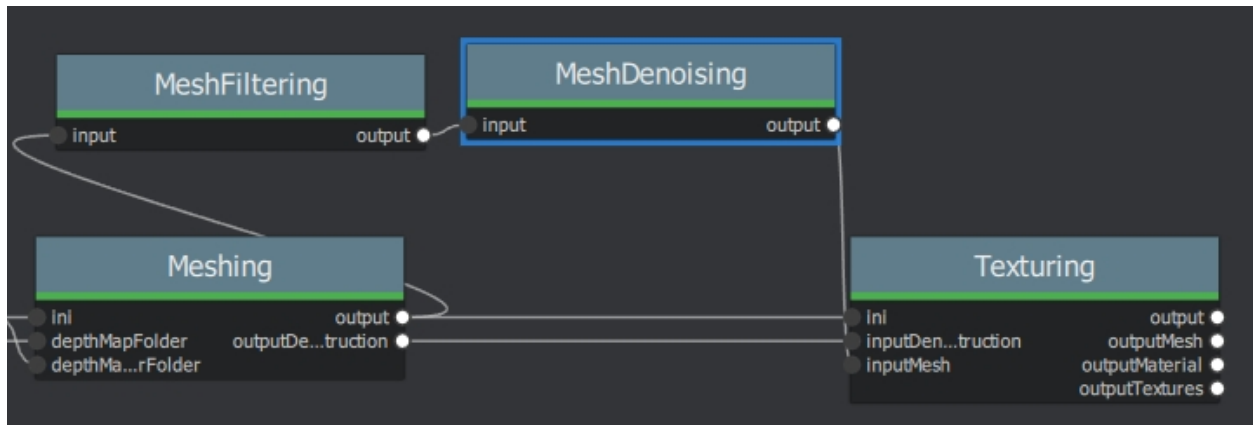
### Description

Denoise your mesh Mesh models generated by 3D scanner always contain noise. It is necessary to remove the noise from the meshes. Mesh denoising: remove noises, feature-preserving [https://www.cs.cf.ac.uk/meshfiltering/index\\_files/Doc/Random%20Walks%20for%20Mesh%20Denoising.ppt](https://www.cs.cf.ac.uk/meshfiltering/index_files/Doc/Random%20Walks%20for%20Mesh%20Denoising.ppt)

Table 15: settings

Name	Description					
input	Input Mesh (OBJ file format)					
Denoising Iterations	Number of denoising iterations “(0	30	1) 5“			
Mesh Update Closeness Weight	“Closeness weight for mesh update	must be positive “(0.0	0.1	0.001) 0.001“		
Lambda	Regularization weight. (0.0 // 10.0 // 0.01) 2					
Eta	“Gaussian standard deviation for spatial weight	scaled by the average distance between adjacent face centroids. Must be positive.(0.0	20.0	0.01) 1.5“		
Mu	“Gaussian standard deviation for guidance weight (0.0	10.0	0.01) 1.5“			
Nu	“Gaussian standard deviation for signal weight. (0.0	5.0	0.01) 0.3“			
Mesh Update Method	“Mesh Update Method * ITERATIVE_UPDATE (default): ShapeUp styled iterative solver * POISSON_UPDATE: Poisson-based update from [Want et al. 2015] (0	1)“				
Verbose Level	“[‘fatal’	‘error’	‘warning’	‘info’	‘debug’	‘trace’]“
Output	Output mesh (OBJ file format).					





Mesh      Update      Method      [https://www.researchgate.net/publication/275104101\\_Poisson-driven\\_seamless\\_completion\\_of\\_triangular\\_meshes](https://www.researchgate.net/publication/275104101_Poisson-driven_seamless_completion_of_triangular_meshes)

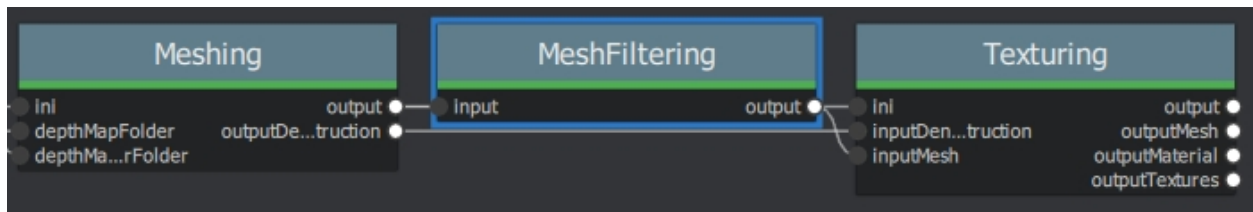
## 15.18 MeshFiltering

### Description

Filter out unwanted elements of your mesh

Table 16: settings

Name	Description					
Input	Input Mesh (OBJ file format)					
Filter Large Triangles Factor	Remove all large triangles. We consider a triangle as large if one edge is bigger than N times the average edge length. Put zero to disable it. 60 (1 - 100)					
Keep Only the Largest Mesh	Keep only the largest connected triangles group (True/False)					
Nb Iterations	5 (0 - 50)					
Lambda	1 (0-10)					
Verbose Level						
Verbose Level	“[‘fatal’	‘error’	‘warning’	‘info’	‘debug’	‘trace’]“
Output mesh	Output mesh (OBJ file format) internalFolder + ‘mesh.obj					



Note: “Keep Only The Largest Mesh”. This is disabled by default in the 2019.1.0 release to avoid that the environment is being meshed, but not the object of interest. The largest Mesh is in some cases the reconstructed background. When

the object of interest is not connected to the large background mesh it will be removed. You should place your object of interest on a well structured non transparent or reflecting surface (e.g. a newspaper).

## 15.19 MeshResampling

### Description

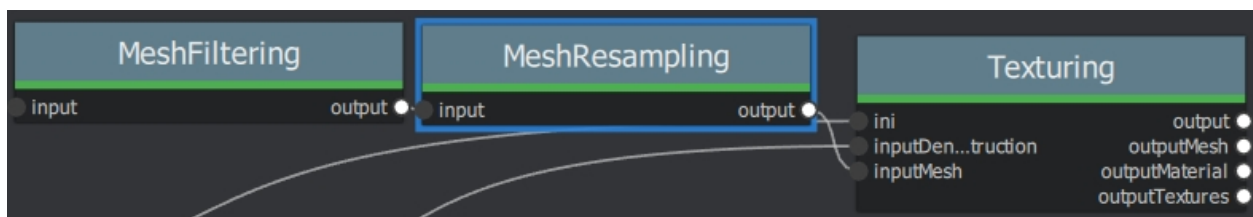
Reducing number of faces while trying to keep overall shape, volume and boundaries You can specify a fixed, min, max Vertices number.

This is different from MeshDecimate!

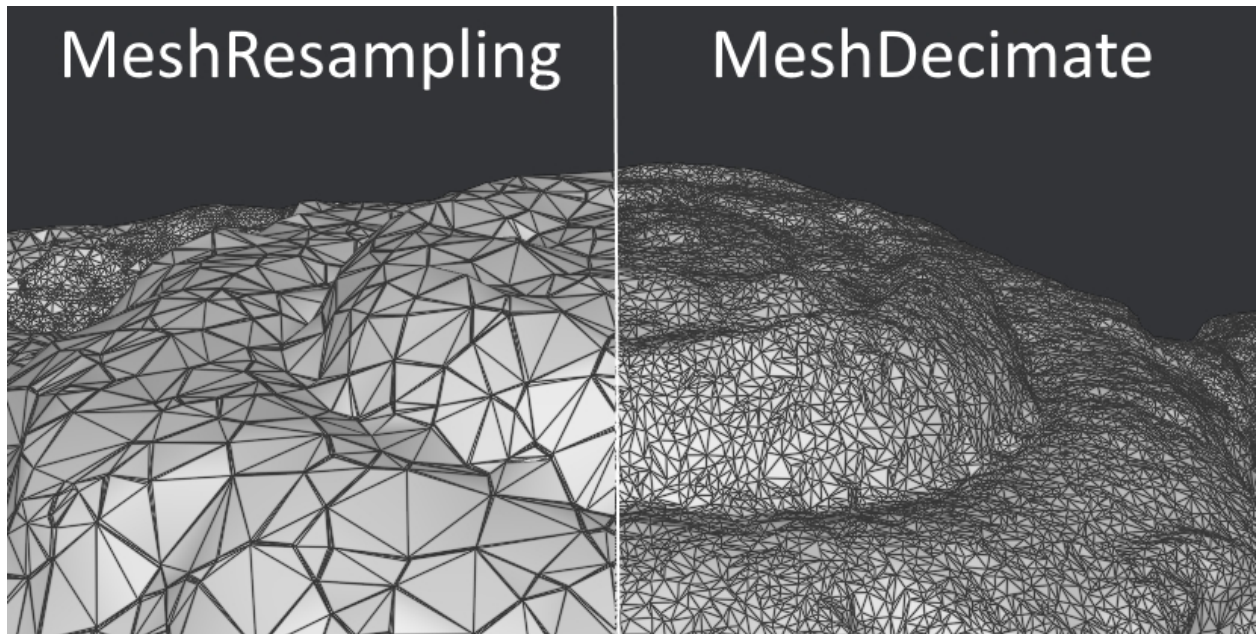
Resampling <https://users.cg.tuwien.ac.at/stef/seminar/MeshResamplingMerge1901.pdf>

Table 17: settings

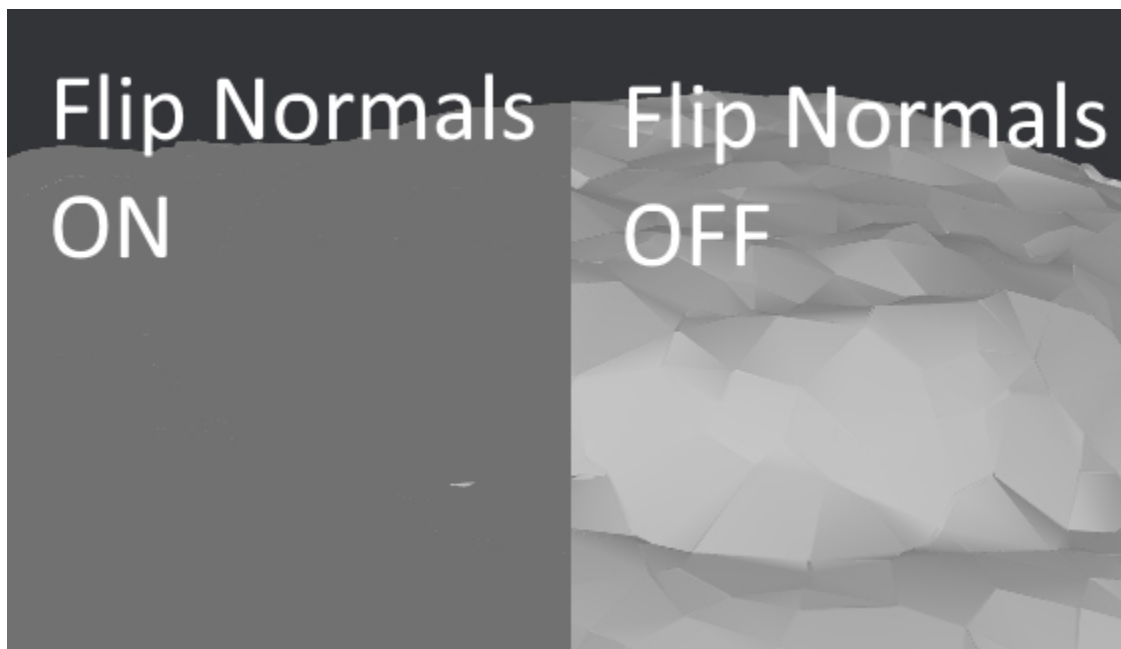
Name	Description				
Input	Input Mesh (OBJ file format)				
Simplification factor	Simplification factor 0.5 (0 - 1)				
Fixed Number of Vertice	Fixed number of output vertices 0 (0 - 1 000 000)				
Min Vertices	Min number of output vertices 0 (0 - 1 000 000)				
Max Vertices	Max number of output vertices 0 (0 - 1 000 000)				
Number of Pre-Smoothing Iteration	Number of iterations for Lloyd pre-smoothing 40 (0 - 100)				
Flip Normals	“Option to flip face normals It can be needed as it depends on the vertices order in triangles and the convention change from one software to another. (True/False)“				
Verbose Level	“[‘fatal’	‘error’	‘warning’	‘info’	‘debug’
Output mesh	Output mesh (OBJ file format) internalFolder + mesh.obj				



## Comparison MeshDecimate and MeshResampling



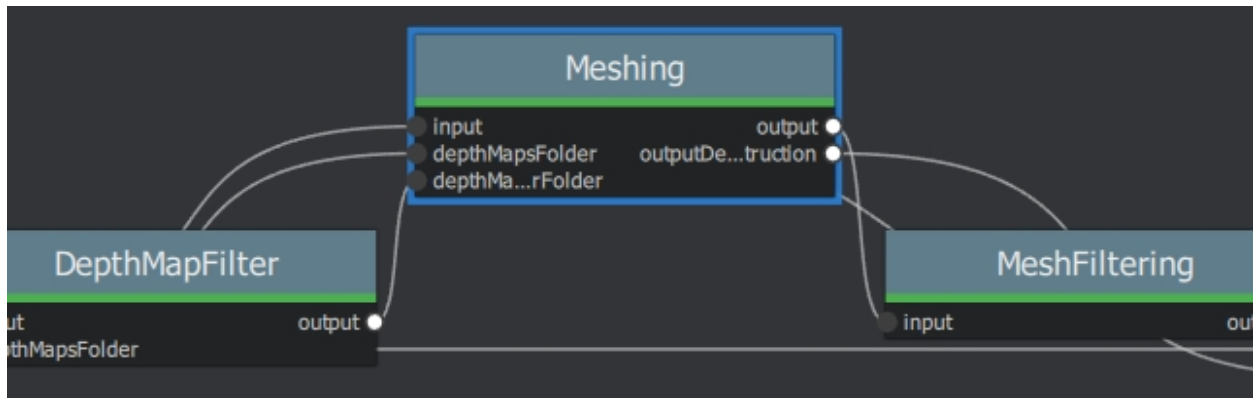
## Flip Normals



## 15.20 Meshing

### Description

none



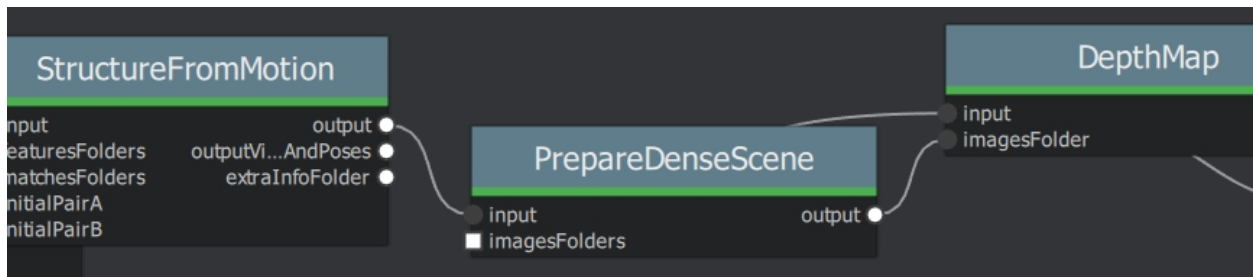
## 15.21 PrepareDenseScene

### Description

- This node undistorts the images and generates EXR images

Table 18: settings

Name	Description
Input	SfMData file
Verbose Level	['fatal' // 'error' // 'warning' // 'info' // 'debug' // 'trace']
Output	MVS Configuration file (desc.Node.internalFolder + 'mvs.ini')



## 15.22 Publish

### Description

- A copy of the Input files are placed in the Output Folder

Can be used to save SfM, Mesh or textured Model to a specific folder

Table 19: settings

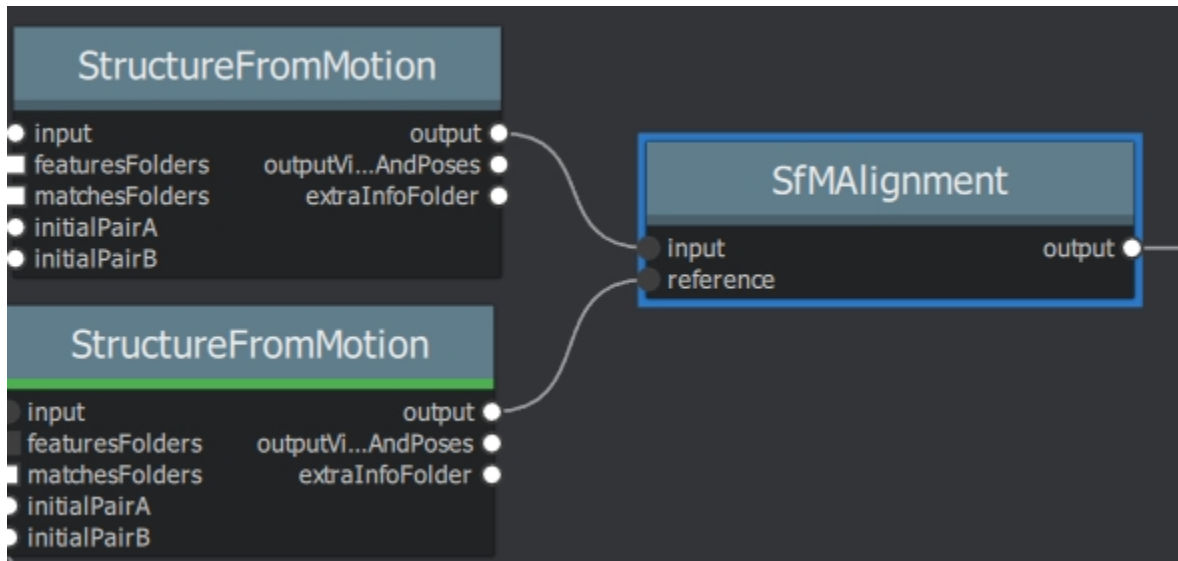
Name	Description
Input Files	Input Files to publish
Output Folder	Folder to publish files to

## 15.23 SfMAlingnment

**Description** align SfM file to a scene

Table 20: settings

Name	Description
Input	SfMData file
Reference	Path to the scene used as the reference coordinate system
Verbose Level	['fatal' // 'error' // 'warning' // 'info' // 'debug' // 'trace']
Output	Aligned SfMData file internalFolder + 'alignedSfM.abc'



## 15.24 SfMTransform

**Description**

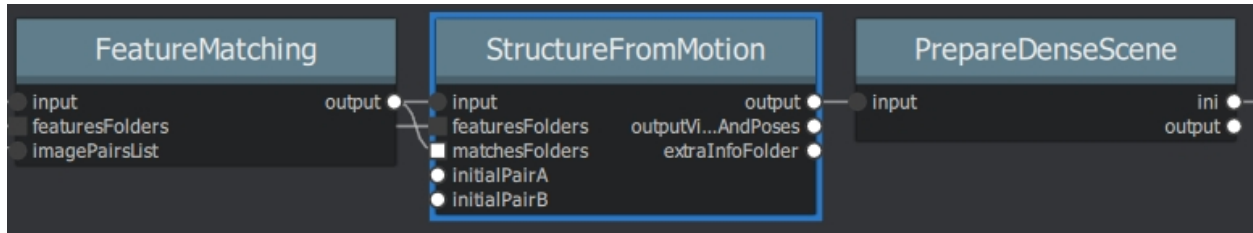
Apply a given transformation camera as the origin of the coordinate system with the SfMTransform node. You can rescale the scene based on the bounding box of CCTAG markers.



## 15.25 StructureFromMotion

### Description

none



## 15.26 Texturing

### Description

Texturing creates UVs and projects the textures change quality and size/ file type of texture



### 16.1 Image File formats

#### **Supported file extensions of Images / Image Viewer:**

All image formats supported by the [OIIO library](#) such as:

‘.jpg’, ‘.jpeg’, ‘.tif’, ‘.tiff’, ‘.png’, ‘.exr’, ‘.rw2’, ‘.cr2’, ‘.nef’, ‘.arw’.

can be imported to Meshroom. However there might be some unexpected behaviour when using RAW images.

## 16.2 Video File formats

## 16.3 3D File formats

Name	Reference	Description		
Alembic (.abc)	<a href="#">Alembic</a>	cloud_and_poses Alembic is a format for storing information about animated scenes after programmatic elements have been applied.		
OBJ		OBJ is a very strict ASCII format for encoding vertices	points	faces and textures first introduced by Wavefront Technologies.
PLY	<a href="#">PLY</a>	The Polygon File Format (or Stanford Triangle Format) has an ASCII representation and a binary representation. It is inspired by the OBJ format that allows the definition of arbitrary properties for every point. This allows an implementation to add arbitrary information to points including accuracy information, but not in any backward-compatible way. Camera information could be included in comments.		
SfM				

FBX support (paused) <https://github.com/alicevision/AliceVision/pull/174>

Alembic is the preferred choice for intermediate storage of points clouds, because it is the only format that is already supported by all of the major 3d software packages.

## 16.4 Other file formats

**.bin** denseReconstruction: The bin format is only useful to get the visibility information of each vertex (no color information)

**.cal** calibration file

**.desc** describer file

**.EXR** OpenEXR image format: for depth map images

**.txt** text file list to describer image parameters **.ini** A configuration file

**.json** describes the used image dataset

**.baf (sfm)** Bundle Adjustment File Export SfM data (Intrinsics/Poses/Landmarks)



### 17.1 Turntable

As mentioned in chapter X, it is possible to use a turntable.

Have you ever heard about masking? Currently, meshroom does not support masking but you can see #188 for a decent workaround.

Essentially, the software is detecting features on both the foreground and background. On a turntable, the subject is moving but the background is not. This confuses it.

So you have 2 choices: make the background completely white and same lighting so that no features can be extracted from this region, or mask your images - that is basically covering the background artificially to stop the region being used in the pipeline, or both.

*Another approach entirely would be to just keep the scene the same but you move the camera instead, which is usually the best way to go about things anyway, this what I would most recommend.*

- without masking, the object on the turntable will become blurry/only partially reconstructed and the background will be reconstructed fine
- we use a blank background to easily mask it

simply using your white wallpaper will not work as it has too many recognizable features you should use a clean and smooth background that will not allow any feature detection use the “Scale for Small-Object Photogrammetry” by Samantha Porter ‘<<http://www.stporter.com/resources/>>’\_ <http://www.stporter.com/resources/> ‘<<https://conservancy.umn.edu/handle/11299/172480?show=full>>’\_ <https://conservancy.umn.edu/handle/11299/172480?show=full>

or create your own

### 17.2 Tutorial: Meshroom for Beginners

<https://sketchfab.com/blogs/community/tutorial-meshroom-for-beginners>

## 17.3 Goal

In this tutorial, we will explain how to use Meshroom to automatically create 3D models from a set of photographs. After specifying system requirements and installation, we will begin with some advice on image acquisition for photogrammetry. We will then give an overview of Meshroom UI and cover the basics by creating a project and starting the 3D reconstruction process. After that, we will see how the resulting mesh can be post-processed directly within Meshroom by applying an automatic decimation operation, and go on to learn how to retexture a modified mesh. We will sum up by showing how to use all this to work iteratively in Meshroom.

Finally, we will give some tips about uploading your 3D models to Sketchfab and conclude with useful links for further information.

## 17.4 Step 0 – System requirements and installation

Meshroom software releases are self-contained portable packages. They are uploaded on the [project's GitHub page](#). To use Meshroom on your computer, simply download the proper release for your OS (Windows and Linux are supported), extract the archive and launch Meshroom executable.

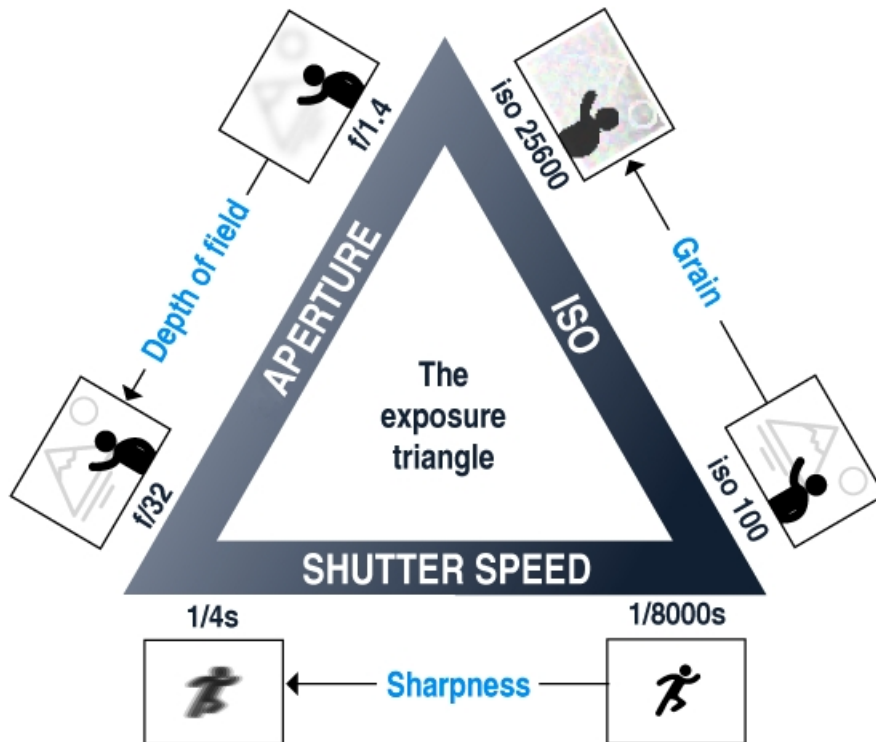
Regarding hardware, an Nvidia GPU is required (with [Compute Capability](#) of at least 2.0) for the dense high quality mesh generation. 32GB of RAM is recommended for the meshing, but you can adjust parameters if you don't meet this requirement.

Meshroom is released in open source under the permissive [MPLv2 license](#), see [Meshroom COPYING](#) for more information.

## 17.5 Step 1 – Image acquisition

The shooting quality is the most important and challenging part of the process. It has dramatic impacts on the quality of the final mesh.

The shooting is always a compromise to accommodate to the project's goals and constraints: scene size, material properties, quality of the textures, shooting time, amount of light, varying light or objects, camera device's quality and settings.



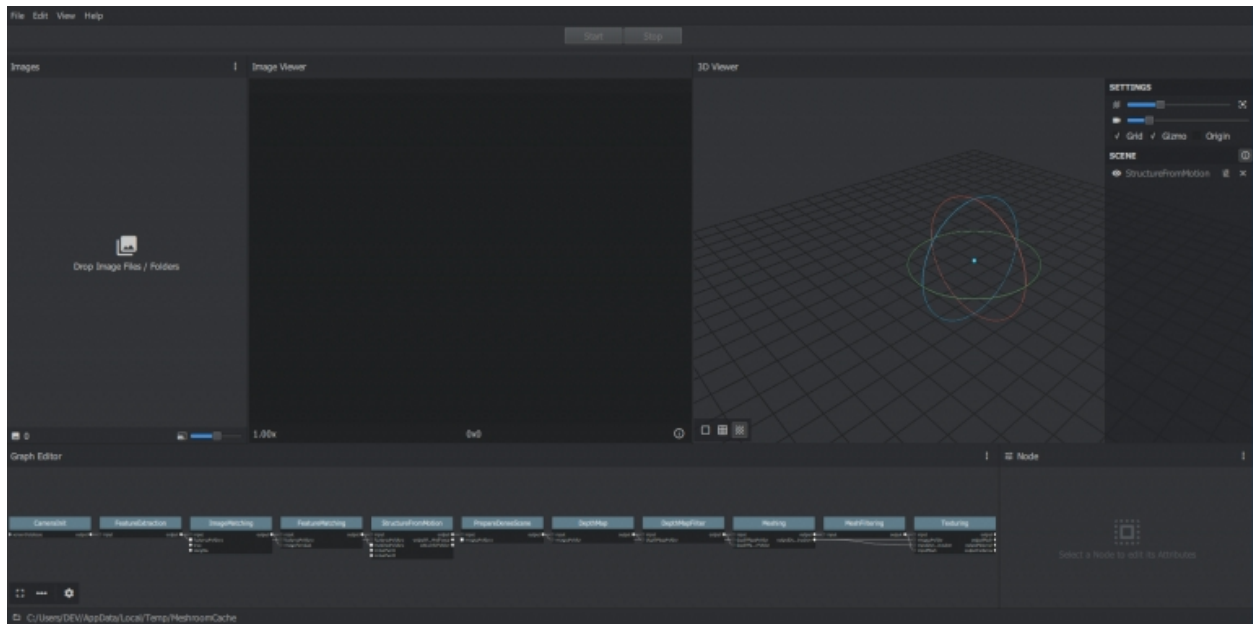
The main goal is to have sharp images without motion blur and without depth blur. So you should use tripods or fast shutter speed to avoid motion blur, reduce the aperture (high f-number) to have a large depth of field, and reduce the ISO to minimize the noise.

## 17.6 Step 2 – Meshroom concept and UI overview

Meshroom has been conceived to address two main use-cases:

- Easily obtain a 3D model from multiple images with minimal user action.
- Provide advanced users (eg: expert graphic artists, researchers) with a solution that can be modified to suit their creative and/or technical needs.

For this reason, Meshroom relies on a nodal system which exposes all the photogrammetry pipeline steps as nodes with parameters. The high-level interface above this allows anyone to use Meshroom without the need to modify anything.



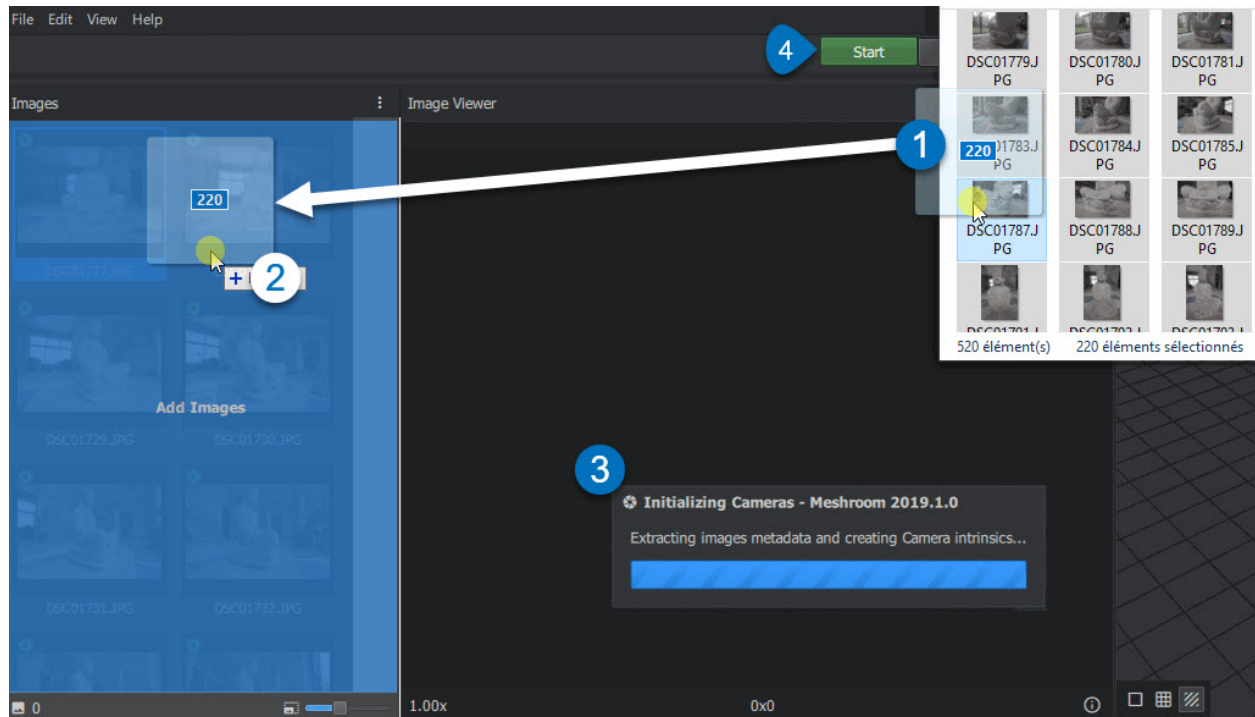
Meshroom User Interface

## 17.7 Step 3 – Basic Workflow

For this first step, we will only use the high-level UI. Let's save this new project on our disk using "File > Save As...".

All data computed by Meshroom will end up in a "MeshroomCache" folder next to this project file. Note that projects are portable: you can move the ".mg" file and its "MeshroomCache" folder afterwards. The cache location is indicated in the status bar, at the bottom of the window.

Next, we import images into this project by simply dropping them in the "Images" area – on the left-hand side. Meshroom analyzes their metadata and sets up the scene.



Meshroom relies on a Camera Sensors Database to determine camera internal parameters and group them together. If your images are missing metadata and/or were taken with a device unknown to Meshroom, an explicit warning will be displayed explaining the issue. In all cases, the process will go on but results might be degraded.

Once this is done, we can press the “Start” button and wait for the computation to finish. The colored progress bar helps follow the progress of each step in the process:

- green: has been computed
- orange: is being computed
- blue: is submitted for computation
- red: is in error

## 17.8 Step 4 – Visualize and Export the results

The generic photogrammetry pipeline can be seen as having two main steps:

- SfM: Structure-from-Motion (sparse reconstruction)
  - Infers the rigid scene structure (3D points) with the pose (position and orientation) and internal calibration of all cameras.
  - The result is a set of calibrated cameras with a sparse point cloud (in Alembic file format).
- MVS: MultiView-Stereo (dense reconstruction)
  - Uses the calibrated cameras from the Structure-from-Motion to generate a dense geometric surface.
  - The final result is a textured mesh (in OBJ file format with the corresponding MTL and texture files).

As soon as the result of the “Structure-from-Motion” is available, it is automatically loaded by Meshroom. At this point, we can see which cameras have been successfully reconstructed in the “Images” panel (with a green camera icon)

and visualize the 3D structure of the scene. We can also pick an image in the “Images” panel to see the corresponding camera in the 3D Viewer and vice-versa.

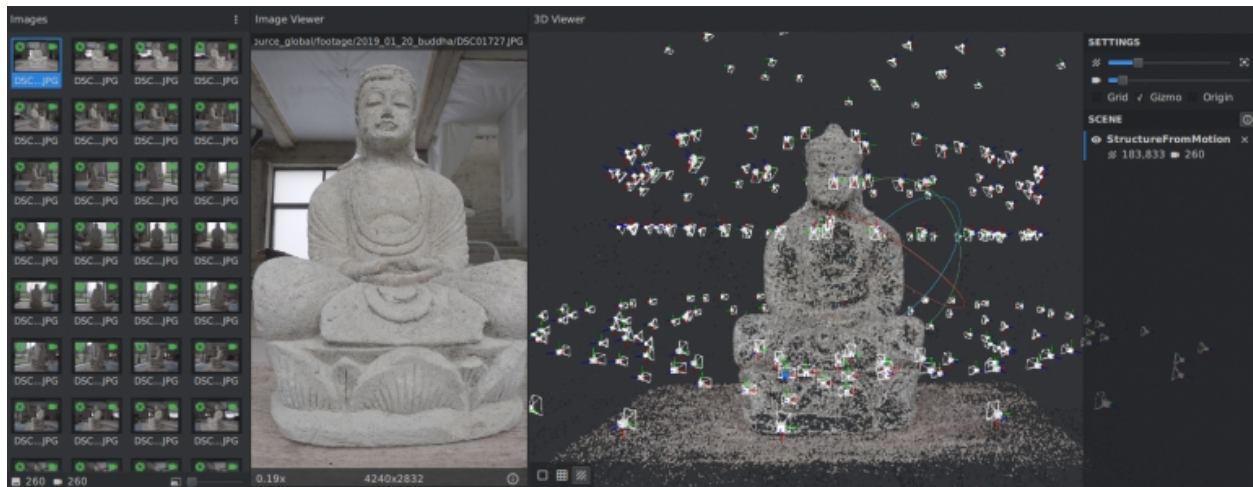
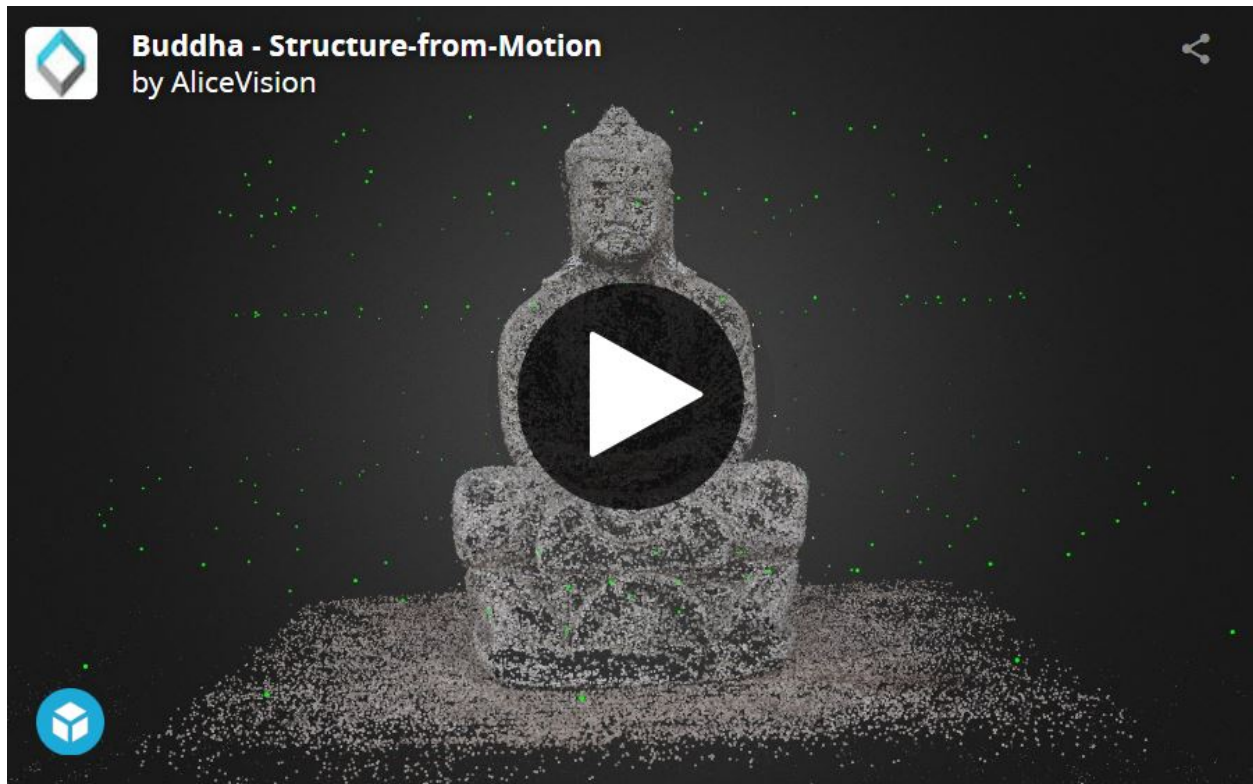


Image selection is synchronized between “Images” and “3D Viewer” panels.

3D Viewer interactions are mostly similar to Sketchfab’s:

- **Click and Move** to rotate around view center
- Double Click
  - on geometry (point cloud or mesh) to define view center
    - alternative: **Ctrl+Click**
- Middle-Mouse Click
  - to pan
    - alternative: **Shift+Click**
- Wheel Up/Down
  - to Zoom in/out
    - alternative: **Alt+Right-Click and Move Left/Right**





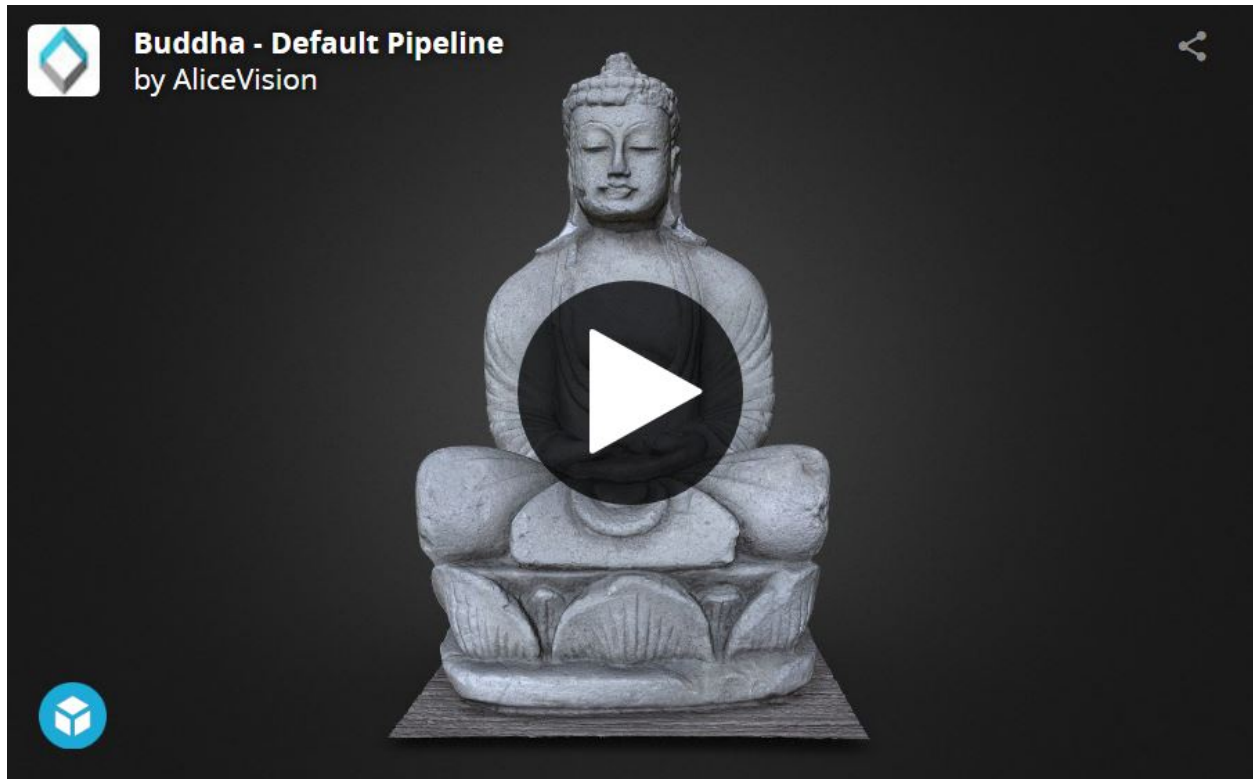
Buddha – Structure-from-Motion by AliceVision on Sketchfab

Once the whole pipeline has been computed, a “Load Model” button at the bottom of the 3D Viewer enables you to load and visualize the textured 3D mesh.



Visualize and access media files on disk from the 3D Viewer

There is no export step at the end of the process: the resulting files are already available on disk. You can right-click on a media and select “Open Containing Folder” to retrieve them. By doing so on “Texturing”, we get access to the folder containing the OBJ and texture files.



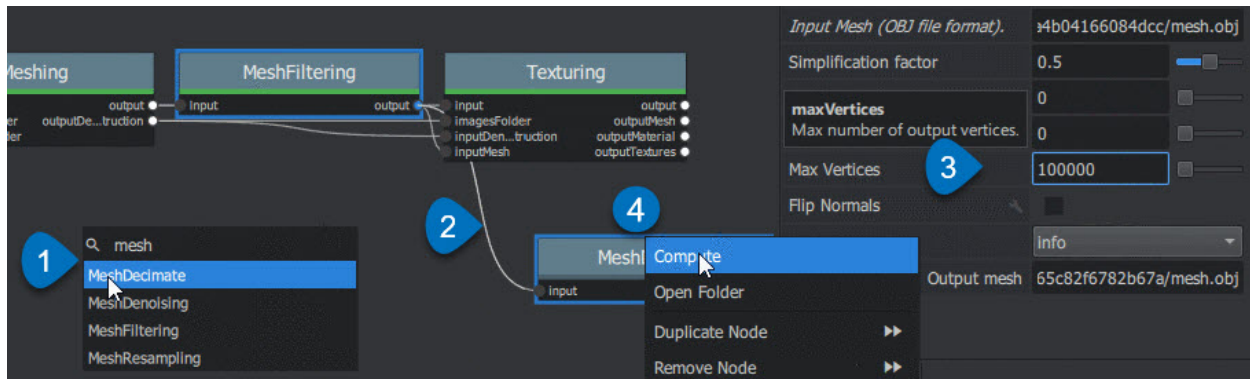
Buddha – Default Pipeline by AliceVision on Sketchfab

## 17.9 Step 5 – Post-processing: Mesh Simplification

Let’s now see how the nodal system can be used to add a new process to this default pipeline. The goal of this step will be to create a low-poly version of our model using automatic mesh decimation.

Let’s move to the “Graph Editor” and right click in the empty space to open the node creation menu. From there, we select “MeshDecimate”: this creates a new node in the graph. Now, we need to give it the high-poly mesh as input. Let’s create a connection by clicking and dragging from MeshFiltering.output to MeshDecimate.input. We can now select the MeshDecimate node and adjust parameters to fit our needs, for example, by setting a maximum vertex count to 100,000. To start the computation, either press the main “Start” button, or right-click on a specific node and select “Compute”.





Create a MeshDecimate node, connect it, adjust parameters and start computation

By default, the graph will become read-only as soon as a computation is started in order to avoid any modification that would compromise the planned processes.

Each node that produces 3D media (point cloud or mesh) can be visualized in the 3D viewer by simply double-clicking on it. Let's do that once the MeshDecimate node has been computed.

- **Double-Click** on a node to visualize it in the 3D viewer. If the result is not yet computed, it will automatically be loaded once it's available.
- **Ctrl+Click** the visibility toggle of a media to display only this media alternative from Graph Editor: **Ctrl+DoubleClick** on a node
- 

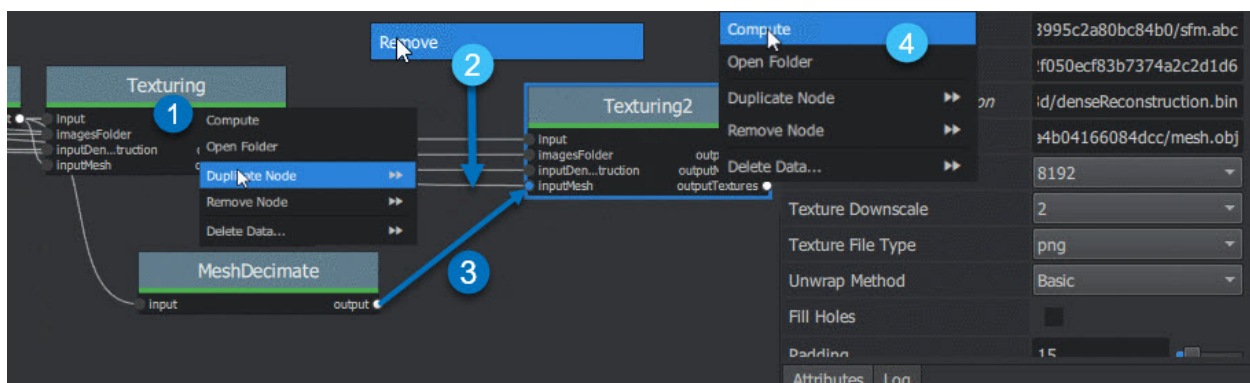
## 17.10 Step 6 – Retexturing after Retopology

Making a variation of the original, high-poly mesh is only the first step to creating a tailored 3D model. Now, let's see how we can re-texture this geometry.

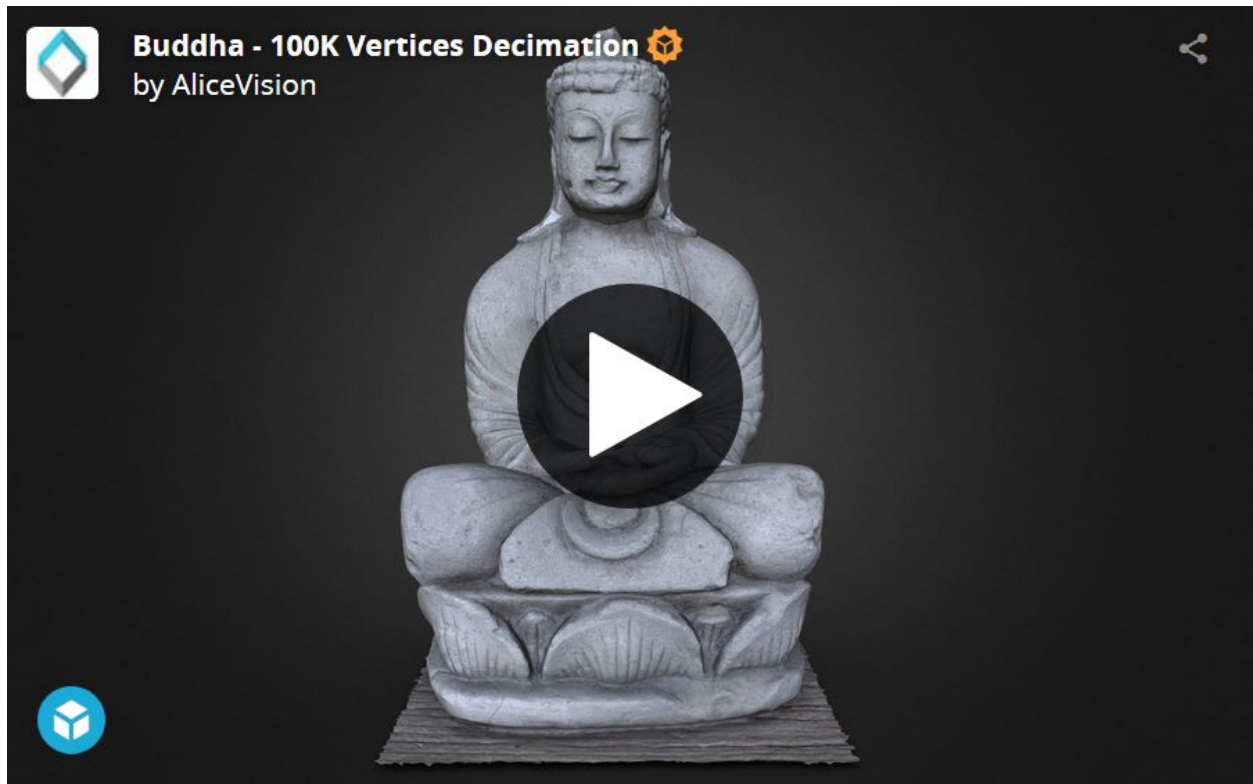
Let's head back to the Graph Editor and do the following operations:

- **Right Click** on the Texturing node > **Duplicate**
- **Right Click** on the connection MeshFiltering.output → Texturing2.inputMesh > **Remove**
- Create a connection from MeshDecimate.output to Texturing2.inputMesh

By doing so, we set up a texturing process that will use the result of the decimation as input geometry. We can now adjust the Texturing parameters if needed, and start the computation.



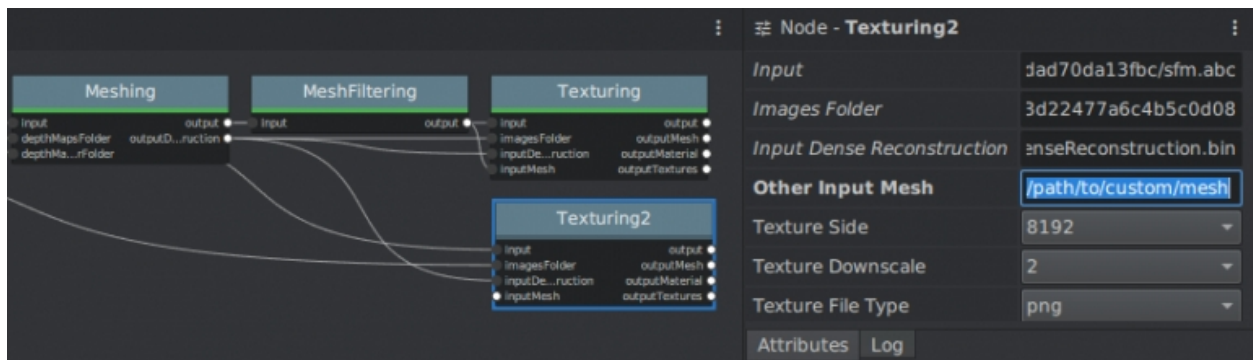
Retexture the decimated mesh using a second Texturing node



Buddha – 100K Vertices Decimation by AliceVision on Sketchfab

**External retopology and custom UVs** This setup can also be used to reproject textures on a mesh that has been modified outside Meshroom (e.g: retopology / unwrap). The only constraint is to stay in the same 3D space as the original reconstruction and therefore **not** change the scale or orientation.

Then, instead of connecting it to MeshDecimate.output, we would directly write the filepath of our mesh in Texturing2.inputMesh parameter from the node Attribute Editor. If this mesh already has UV coordinates, they will be used. Otherwise it will generate new UVs based on the chosen “Unwrap Method”.



Texturing also accepts path to external meshes

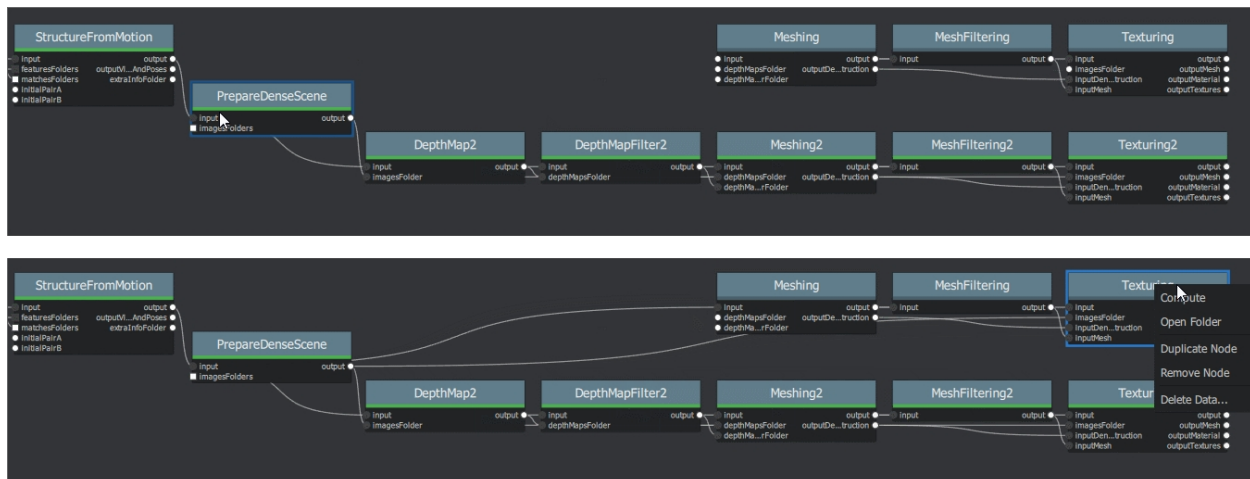
## 17.11 Step 7 – Draft Meshing from SfM

The MVS consists of creating depth maps for each camera, merging them together and using this huge amount of information to create a surface. The generation of those depth maps is, at the moment, the most computation intensive part of the pipeline and requires a CUDA enabled GPU. We will now explain how to generate a quick and rough mesh

directly from the SfM output, in order to get a fast preview of the 3D model. To do that we will use the nodal system once again.

Let's go back to the default pipeline and do the following operations:

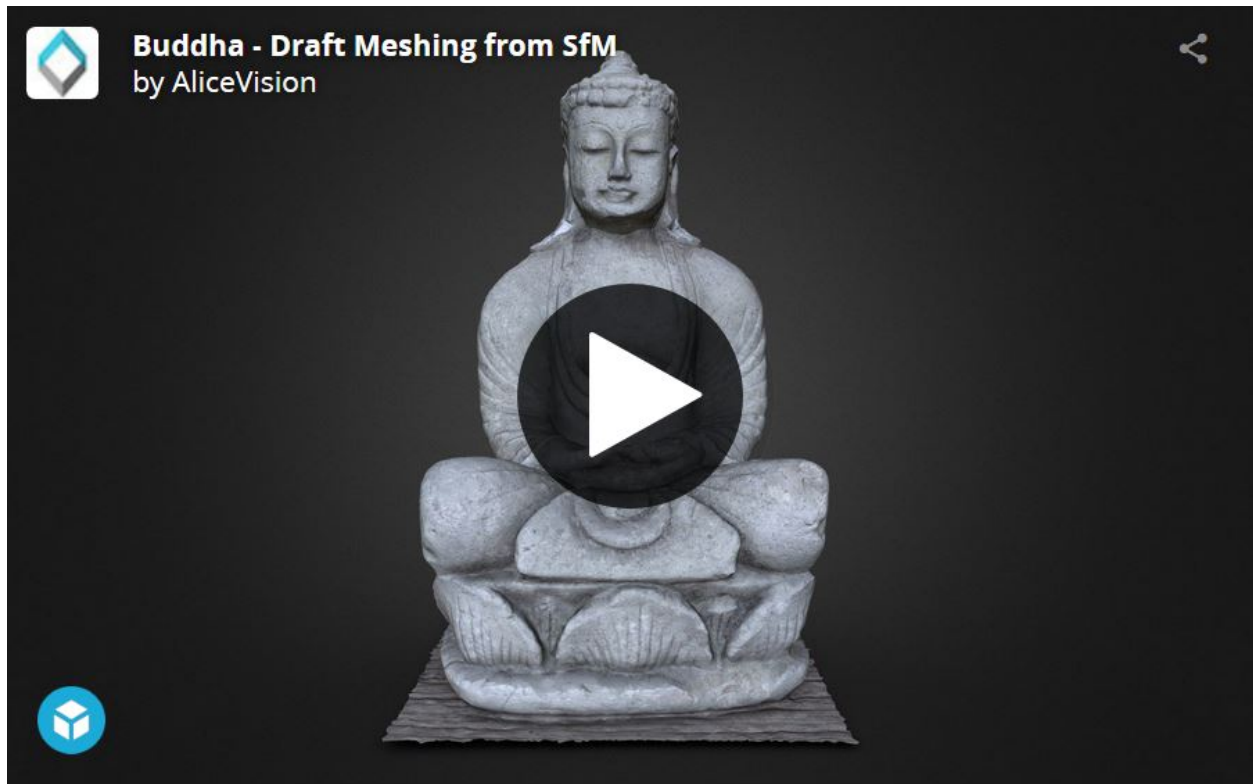
- Right Click  
on DepthMap >  
Duplicate Nodes from Here  
( “  
>>  
” icon) to create a branch in the graph and keep the previous result available.
- alternative: **Alt + Click** on the node
- Select and remove (**Right Click > Remove Node** or **Del**) DepthMap and DepthMapFilter
- Connect PrepareDenseScene.input Meshing.input
- Connect PrepareDenseScene.output Texturing.inputImages



### Draft Meshing from StructureFromMotion setup

With this shortcut, the Meshing directly uses the 3D points from the SfM, which bypass the computationally intensive steps and dramatically speed up the computation of the end of the pipeline. This also provides a solution to get a draft mesh without an Nvidia GPU.

The downside is that this technique will only work on highly textured datasets that can produce enough points in the sparse point cloud. In all cases, it won't reach the level of quality and precision of the default pipeline, but it can be very useful to produce a preview during the acquisition or to get the 3D measurements before photo-modeling.



Buddha – Draft Meshing from SfM by AliceVision on Sketchfab

## 17.12 Step 8 – Working Iteratively

We will now sum up by explaining how what we have learnt so far can be used to work iteratively and get the best results out of your datasets.

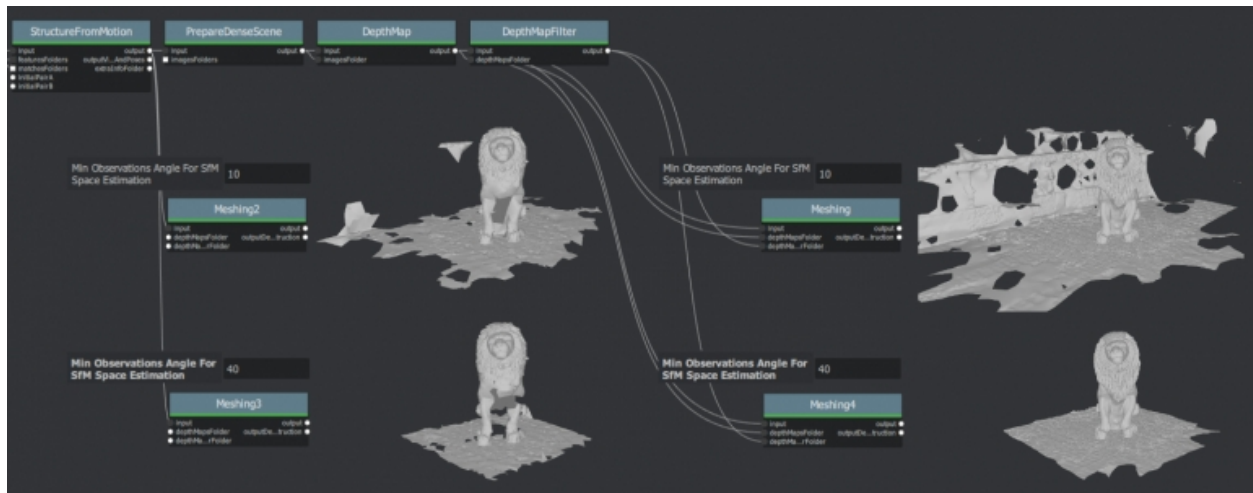
### 1. Computing and analyzing Structure-from-Motion first

This is the best way to check if the reconstruction is likely to be successful before starting the rest of the process (**Right click > Compute** on the StructureFromMotion node). The number of reconstructed cameras and the aspect/density of the sparse point cloud are good indicators for that. Several strategies can help improve results at this early stage of the pipeline:

- Extract more key points from input images by setting “Describer Preset” to “high” on FeatureExtraction node (or even “ultra” for small datasets).
- Extract multiple types of key points by checking “akaze” in “Describer Type” on FeatureExtraction, Feature-Matching and StructureFromMotion nodes.

### 2. Using draft meshing from SfM to adjust parameters

Meshing the SfM output can also help to configure the parameters of the standard meshing process, by providing a fast preview of the dense reconstruction. Let’s look at this example:



With the default parameters, we can preview from **Meshing2** that the reconstructed area includes some parts of the environment that we don't really want. By increasing the “Min Observations Angle For SfM Space Estimation” parameter, we are excluding points that are not supported by a strong angle constraint (**Meshing3**). This results in a narrower area without background elements at the end of the process (**Meshing4** vs default **Meshing**).

### 13. Experiment with parameters, create variants and compare results

One of the main advantages of the nodal system is the ability to create variations in the pipeline and compare them. Instead of changing a parameter on a node that has already been computed and invalidate it, we can duplicate it (or the whole branch), work on this copy and compare the variations to keep the best version.

In addition to what we have already covered in this tutorial, the most useful parameters to drive precision and performance for each step are detailed on the [Meshroom Wiki](#).

## 17.13 Step 9 – Upload results on Sketchfab

Meshroom does not yet provide an export tool to Sketchfab, but results are all in standard file formats and can easily be uploaded using the Sketchfab web interface. Our workflow mainly consists of these steps:

- Decimate the mesh within Meshroom to reduce the number of polygons
- Clean up this mesh in an external software, if required (to remove background elements for example)
- Retexture the cleaned up mesh
- Upload model and textures to Sketchfab

You can see some 3D scans from the community [here](#) and on our **Sketchfab page**.

Don't forget to tag your models with “alicevision” and “meshroom” if you want us to see your work!



If this is the first time you are using photogrammetry software, read the following chapter on how to take good photos for your project.

### 18.1 Basics

- Your Scene/Object should be well lit.
- Avoid shadows and reflections and transparent objects.
- Best shoot in indirect light such as in the daylight shadow of a building avoid plain and one-colored surfaces
- Don't use a flash.
- Do not change the focal length while shooting or use a fixed lens.
- Make sure you can take pictures from all angles.
- Avoid moving objects in the scene or background!
- Rotate only objects with a plain background
- The Object of interest should always fill most of the image
- Take images with a side overlap of min. 60% and frontal overlap of 80%.
- For each shot, move to a new position (or rotate the object)
- Do not take multiple images from the same spot.
- You can photograph multiple times in different patterns to leave no blind spots
- avoid shaking
- The more images you have, the better. You can always decide not to use them. . .

## 18.2 Details

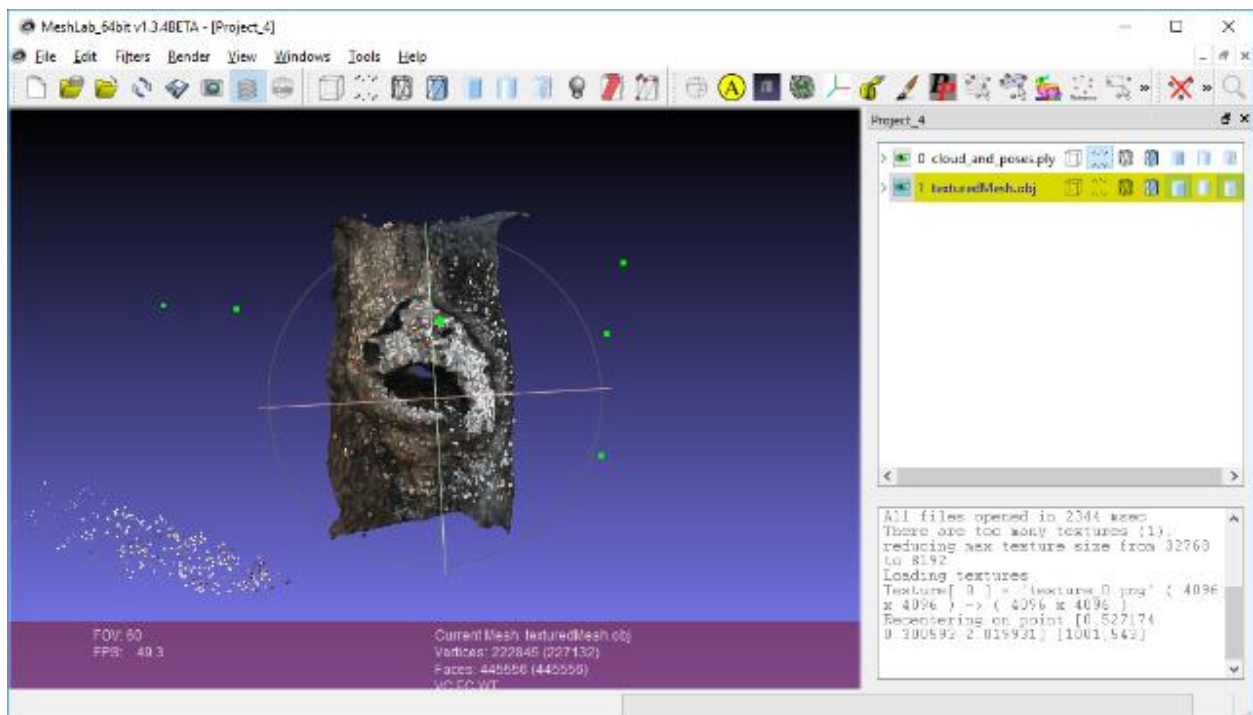
## 18.3 Tutorials



## 19.1 View and Edit Models

### 19.1.1 Meshlab

*You can drag and drop different OBJ and PLY files as layers.*



*So in this case I have a layer for both the final mesh and the SFM points/cameras. Sometimes the mesh smoothing step can be a little too aggressive so I find it useful to compare between the original mesh and the smooth mesh. If the*

*mesh looks broken, the PLY sfm data and the OBJ meshes are great for tracing through the pipeline.*

clean up / delete / smooth

The first thing you want to do is to rotate your model and align it with the coordinate system.

You can import the obj into [Meshlab](#) then go to **Filters > Normals, Curvatures \*\* and \*\*Orientation > Transform: Rotate \*\* \*\*** and align it yourself from there. \*\* \*\*

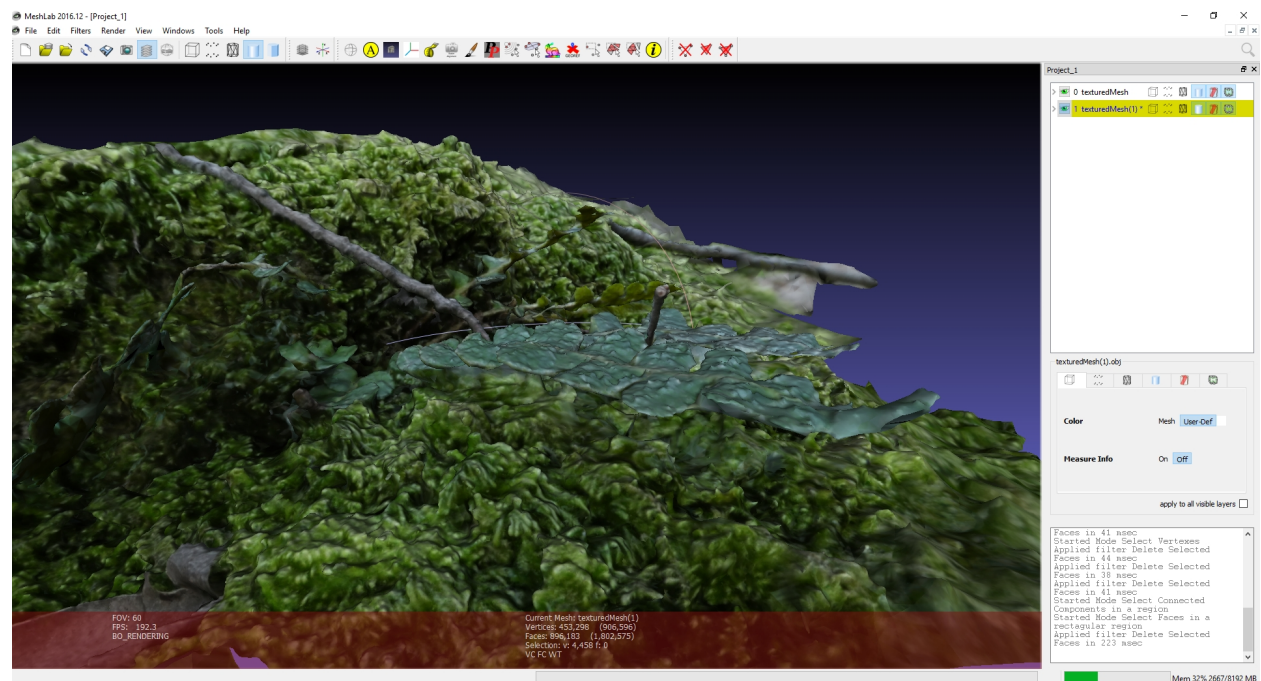
There might be some parts of the model or the scene you want to remove.

You can select ..... then remove...

[http://www.banterle.com/francesco/courses/2017/be\\_3drec/slides/Meshlab.pdf](http://www.banterle.com/francesco/courses/2017/be_3drec/slides/Meshlab.pdf)

<http://>

[www.scanner.imagefact.de/tut/meshlabTut.pdf](http://www.scanner.imagefact.de/tut/meshlabTut.pdf)



Smooth mesh

If you don't like the smoothing results from Meshroom, you can smooth the mesh yourself.

<http://www.cs.cmu.edu/~reconstruction/advanced.html#meshlab>

Tutorials by Mister P. [MeshLab Tutorials](#) [MeshLab Basics: Navigation](#)

[MeshLab Basics: Selection, part one](#)

[MeshLab Basics: Selection, part two](#)

[Cleaning: Triangles and Vertices Removal](#)

[Cleaning: Basic filters](#)

[Mesh Processing: Decimation](#) [Meshlab Processing: Smoothing](#)

[MeshLab Basics: Scale to real measures](#)

### 19.1.2 Blender

For detailed instructions visit the [blender homepage](#) or the [blender youtube channel](#).

Here is a quick tutorial on how to optimize photogrammetry objects inside Blender: How to 3D Photoscan Easy and Free!

<https://www.youtube.com/watch?v=k4NTf0hMjtY>

meshing filtering 10:18 / 13:17 blender import

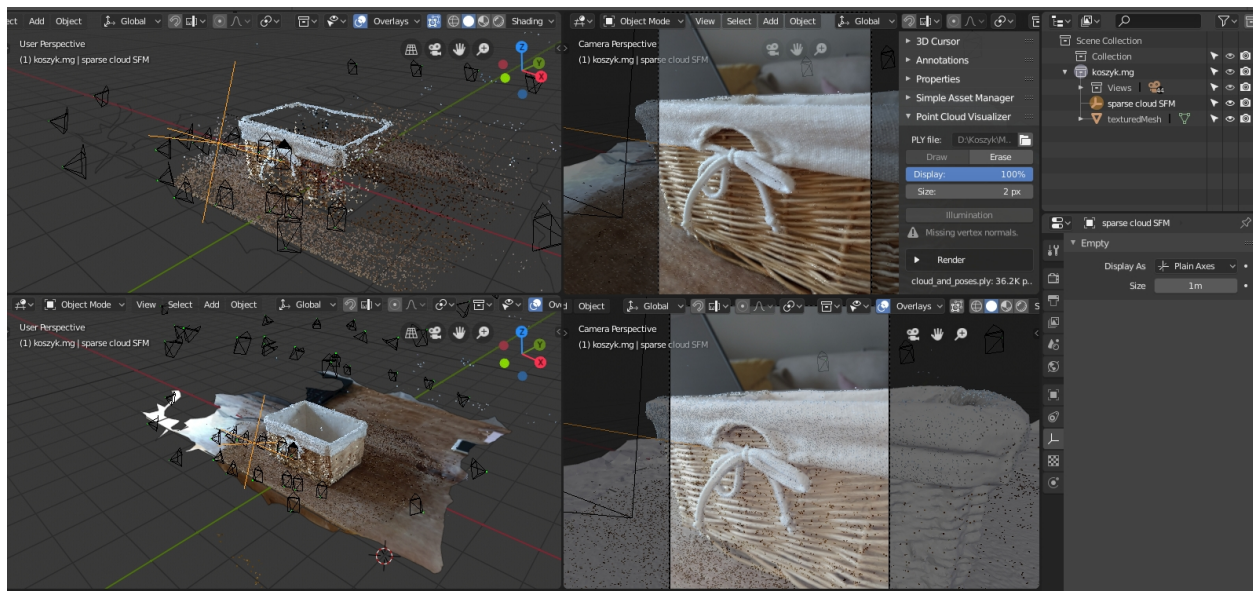
<https://www.youtube.com/watch?v=RmMDFydHeso>

### 19.1.3 Meshroom2Blender Blender Plugin

Blender importer for AliceVision Meshroom

datafiles: cameras, images, sparse pointcloud and obj's.

Basic implementation of Meshroom importer. If you have sophisticated node tree it will use only the first nodes from the file. Addon assumes you did compute each stages/nodes, and the output is same. Visit the [Github project site](#) for details.



### 19.1.4 BlenderLandscape

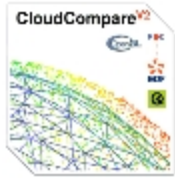
Addon for Blender 2.79b. 3DSurvey Collection of tools to improve the work-flow of a 3D survey (terrestrial or UAV photogrammetry). Import multiple objs at once (with correct orientation), for instance a bunch of models made in Meshroom. <https://github.com/zalmoxes-laran/BlenderLandscape>

### 19.1.5 Instant Meshes

<https://github.com/wjakob/instant-meshes>

includes quick intro

why do we want to use it? It is a really fast auto-retopology solution and helps you create more accurate meshes



### 19.1.6 CloudCompare

3D point cloud and mesh processing software

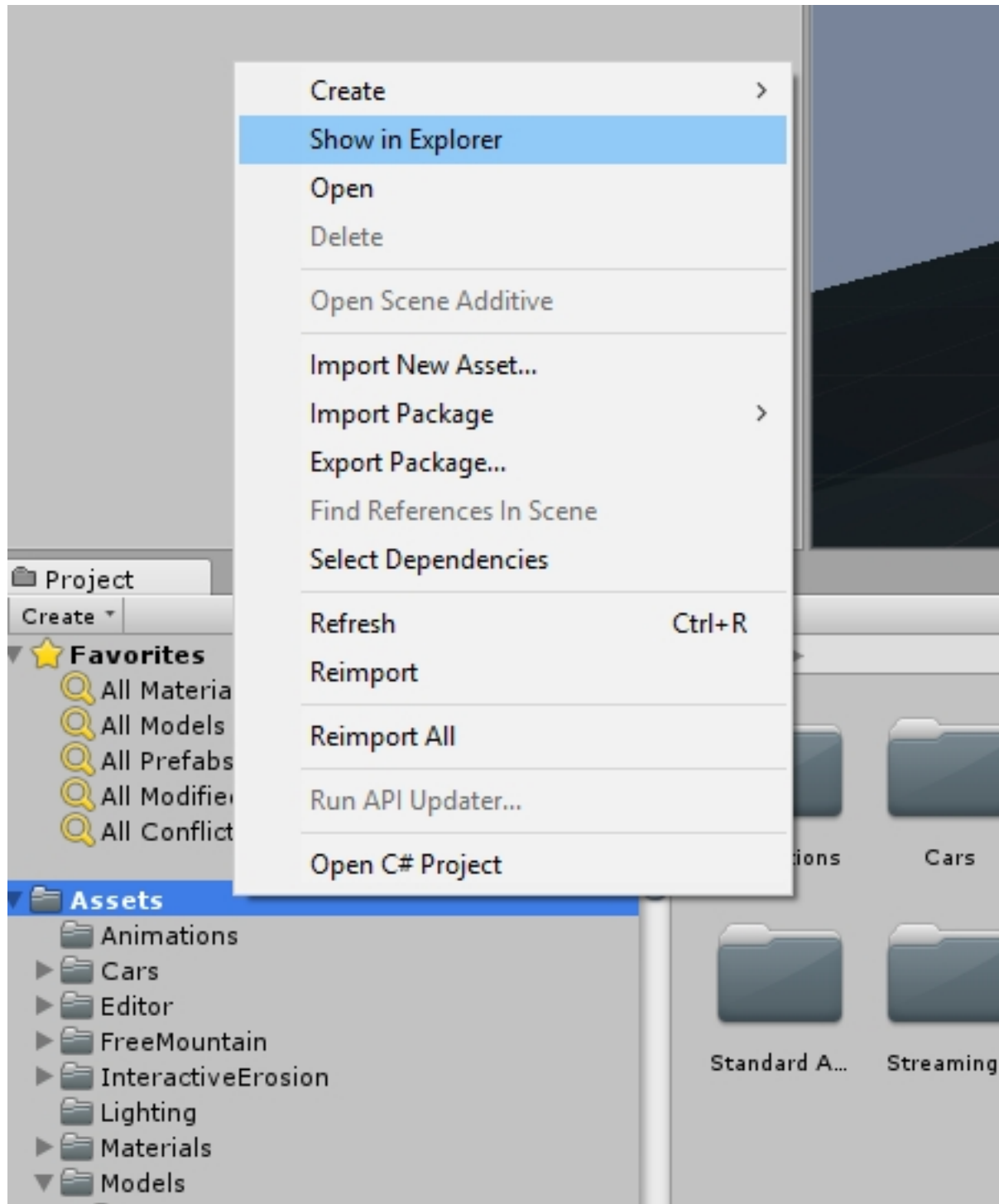
Open Source Project

<https://www.danielgm.net/cc/>

<http://www.danielgm.net/cc/release/>

tutorial

<http://www.danielgm.net/cc/tutorials.html>



### 19.1.7 Export model to Unity

Start Unity, open your project and your asset folder.

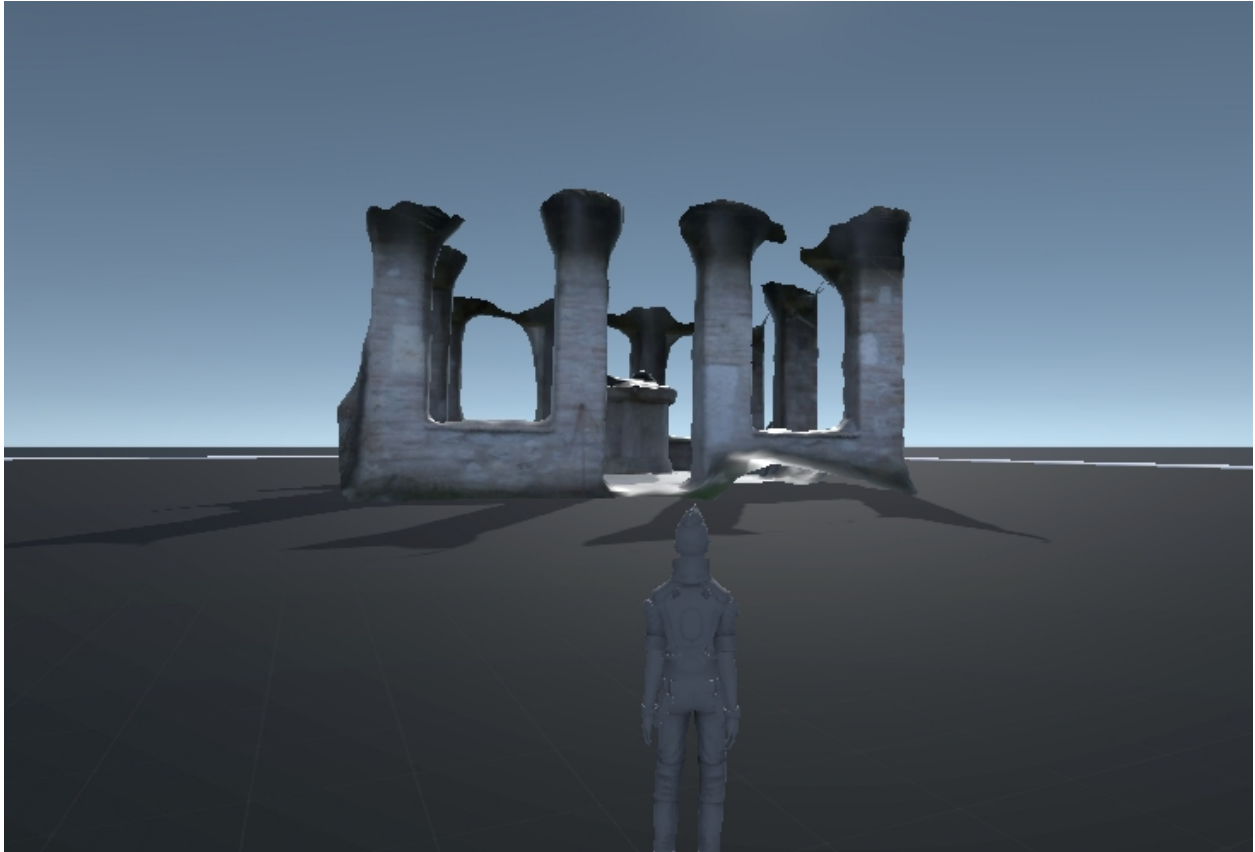
Navigate in the file Explorer of your OS to the assets subfolder where you want to store your Photogrammetry object.

Copy the model.obj and texture.jpg (or other supported file types) from the Meshroom Export folder to the Unity

assets subfolder.

Open Unity and wait for the auto-import to complete.

You might want to optimize your mesh and texture for ingame use.



Now you can add your model to the scene.

There is a little more to do to create a simple demo game, like adding a Mesh collider, optimize the texture,...

For detailed instructions visit the [Unity homepage](#) .

Here is a manual on how to optimize photogrammetry objects inside Unity: [Unity Photogrammetry Workflow ..](#)  
image:: 100000000000076E00000401AC14E84A53702851.jpg

### 19.1.8 Export to Maya (Plugin)

**MeshroomMaya** (v0.4.2) is a Maya plugin that enables to model 3D objects from images.

<https://github.com/alicevision/MeshroomMaya>

This plugin is not available at the moment.

Use the Export to Maya node instead.

#### Alembic bridge

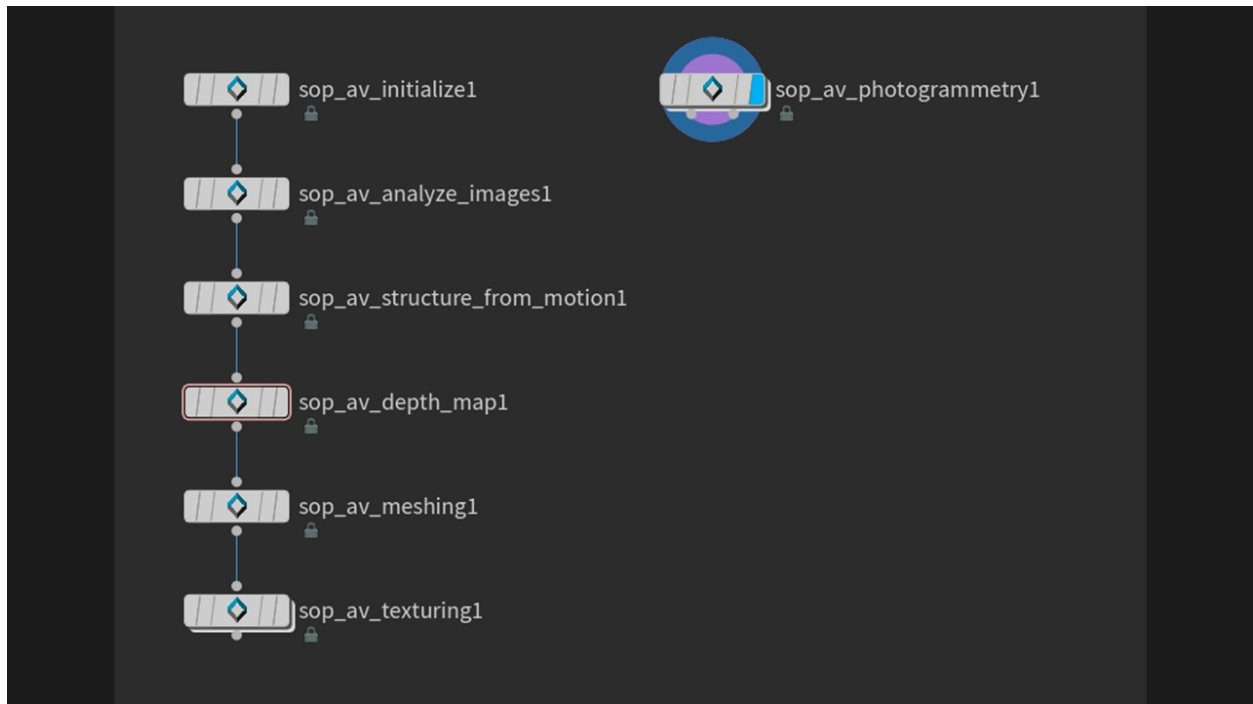
*Export from Meshroom for Maya*

Use the Export to Maya node to export the Alembic ABC file



*Import in Nuke/Mari*

In menu “NukeMVG > Import Alembic” , .abc file can be loaded. The tool create the graph of camera projection. Result can be export to Mari via Nuke <-> Mari bridge.



### 19.1.9 SideFX Houdini Plugin

An implementation of *Alicevision* is available in Houdini as part of the (free) GameDevelopmentToolset.

You can find Installation Instructions on the following page: <https://www.sidefx.com/tutorials/alicevision-plugin/>

Review (german):

<https://www.digitalproduction.com/2019/02/26/alicevision-photogrammetrie-in-houdini/>

Students can download the free learning edition called ‘<<https://www.sidefx.com/products/compare/>>’\_ Houdini Apprentice . This is a node-locked license that has all the features of Houdini FX with some restrictions such as a limited render size and a watermark on final renderings.

## 19.2 Share your model

(A build in upload module is on the wishlist. Read github)

clip area

reduce polycount

reduce resolution

<https://sketchfab.com/>

Short description

<https://www.thingiverse.com/>

<https://pointscene.com/>  
<https://www.pointbox.xyz/>  
and more...

## 19.3 Print your model

<https://groups.google.com/forum/#!topic/alicevision/RCWKoevn0yo>

## 19.4 Tethering software

Remote control your camera via USB cable. For use with a **turntable and/or Live Reconstruction**.

Some manufacturers (Sony, Panasonic, FUJIFILM, Hasselblad, Canon EOS..) provide a free tool for your software others sell them (Nikon, Canon). Some commercial third party solutions are out there, too.

This list only contains free open-source projects.

### 1 DigiCamControl (Windows)

- Multiple camera support

<http://digicamcontrol.com/download>

Supports many Nikon, Canon, Sony SLR models and a few other cameras.

Full list here: <http://digicamcontrol.com/cameras>

### 2 Entangle Photo (Linux)

<https://entangle-photo.org/>

Nikon or Canon DSLRs camera supporting ‘<<http://www.gphoto.org/doc/remote/>>’\_ remote capture in libgphoto2 will work with Entangle.

### 3 GPhoto (Linux)

<http://www.gphoto.org/>

### 4 Sofortbildapp (OSX)

<http://www.sofortbildapp.com/>

### 5 PkTriggerCord (Windows, Linux, Android)

for Pentax cameras

<http://pktriggercord.melda.info/>

<https://github.com/asalamon74/pktriggercord/>

### 4 Darktable (Windows, Linux, OSX)

<http://www.darktable.org/>

[https://www.darktable.org/usermanual/en/tethering\\_chapter.html](https://www.darktable.org/usermanual/en/tethering_chapter.html)

### WifiRemoteControl

For some cameras wifi control can be used.

LMaster <https://github.com/Rambalac/GMaster> for some Lumix cameras for example.



There are even tools for PC to connect to ActionCams using Wifi...

## 19.5 Related Projects

..image:: ofxMVG.jpg

### 19.5.1 ofxMVG

Camera Localization OpenFX Plugin for Nuke

<https://github.com/alicevision/ofxMVG>

Not available at the moment.

..image:: marker2.jpg

### 19.5.2 CCTag

Concentric Circles Tag

This library allows you to detect and identify CCTag markers. Such marker system can deliver sub-pixel precision while being largely robust to challenging shooting conditions. <https://github.com/alicevision/CCTag>

#### CCTag library

Detection of CCTag markers made up of concentric circles. Implementations in both CPU and GPU.

**See paper** : “Detection and Accurate Localization of Circular Fiducials under Highly Challenging Conditions.” Lilian Calvet, Pierre Gurdjos, Carsten Griwodz and Simone Gasparini. CVPR 2016.

[https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Calvet\\_Detection\\_and\\_Accurate\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Calvet_Detection_and_Accurate_CVPR_2016_paper.pdf)

#### Marker library

Markers to print are located [here](#) .

**WARNING** Please respect the provided margins. The reported detection rate and localization accuracy are valid with completely planar support: be careful not to use bent support (e.g. corrugated sheet of paper).

The four rings CCTags will be available soon.

CCTags requires either CUDA 8.0 and newer or CUDA 7.0 (CUDA 7.5 builds are known to have runtime errors on some devices including the GTX980Ti). The device must have at least compute capability 3.5.

Check your graphic card CUDA [compatibility here](#) .

..image:: marker3.jpg

### 19.5.3 PopSIFT

Scale-Invariant Feature Transform (SIFT)

This library provides a GPU implementation of SIFT. 25 fps on HD images on recent graphic cards. <https://github.com/alicevision/popsift>



## 20.1 Crashed at Meshing

Solution: try to reduce the value of maxPoints on the Meshing node to avoid using too much RAM & SWAP

#243 #303

## 20.2 DepthMap node too slow

You can speed up the Depth Map process. Here is what you need to do:

Augment the downscale factor to directly reduce the precision.

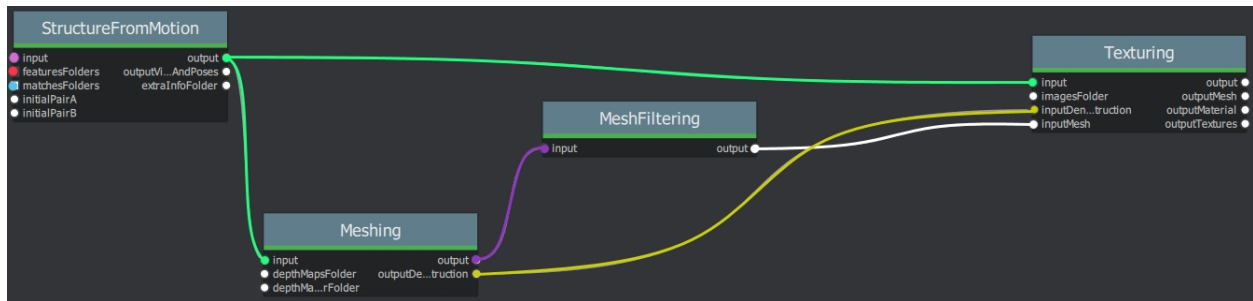
Reduce the number of T cameras (sgmMaxTCams, refineMaxTCams) will directly reduce the computation time linearly, so if you change from 10 to 5 you will get a 2x speedup.

A minimum value of 3 is necessary, 4 already gives decent results in many cases if the density of your acquisition process regular enough.

The default value is necessary in large scale environment where it is difficult to have 4 images that cover the same area.(#228)

## 20.3 Draft Meshing

As of version 2019.1.0 of meshroom, it is possible to do a reconstruction without using the Depthmap node (depthmap requires CUDA). It is much faster than depth map but the resulting mesh is low quality, so it is still recommended that the depthmap is used to generate the mesh if possible. This can be done using the following node configuration:



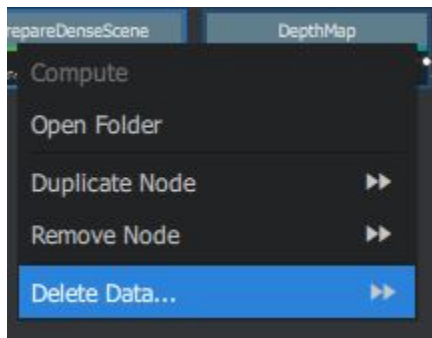
You should use the HIGH preset on the FeatureExtraction node to get enough density for the Meshing. [Reconstruction-parameters](#)

## 20.4 Error: Graph is being computed externally

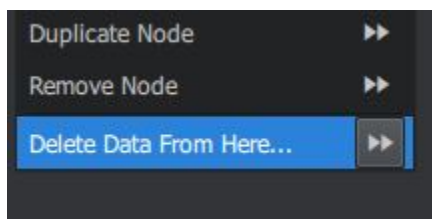
Unexpected exit of Meshroom while processing can cause the “Graph is being computed externally” problem. ‘#249’\_ The Start and Stop buttons are greyed out.

Background: When Meshroom is terminated unexpectedly, files are left in the cache folders. When you open such a project, Meshroom will think, based on the residual files, that parts of the pipeline are computed externally. (This feature ([Renderfarm](<https://github.com/alicevision/meshroom/wiki/Large-scale-dataset>)) is not included in the binary Release 2019.1.0) So the buttons are greyed out because Meshroom is waiting for an external source to compute the graph. Obviously, this won’t go anywhere. This behaviour can also occur, when you modify nodes in the advanced mode while the graph is being computed.

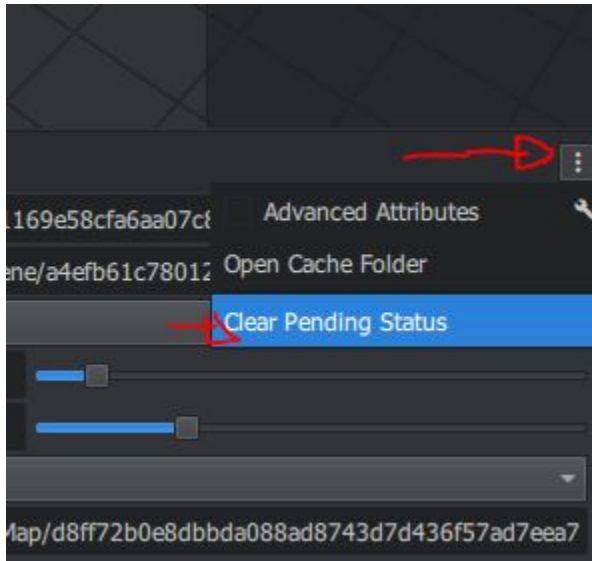
To fix this problem, first try to ‘Clear Submitted Status’ by clicking on the bad node (right click->delete data).



If this does not work, also clear the submitted statuses of the following nodes (right click->delete data >>>)



You have a menu on the top-right of the graph widget with “Clear Pending Status” to do it **on all nodes** at once.



Alternatively, go to the cache folder of your project and delete the contents of the node folders starting with the node where Meshroom stopped working (marked in dark green). You can keep successful computed results (light green). Now you can continue computing the graph on your computer.

## 20.5 Images cannot be imported

The import module from AliceVision has problems parsing corrupted image files. Some mobile phone cameras and action cams/small cameras like the CGO3+ from Yuneec produce images which are not valid. Most image viewers and editing software can handle minor inconsistencies.

Use tools like [Bad Peggy](#) to [check for errors](#) in your image files.

e.g. "...extraneous bytes before marker 0xdb".

or "Truncated File - Missing EOI marker" on a raspberry camera

To fix this problem, you need to bulk convert your dataset (this is why downscaling worked too). You can use `Irfran-View File->Batch Conversion` or `Imagemagick`. Make sure you set the quality to 100%. Now you can add the images to Meshroom (assuming the camera is in the sensor db).

—

**drag and drop of images does not work (#149)** mouse over the with any photos the cursor is disabled and dropping photos into the viewport has no effect. Do you run Meshroom as admin? If yes, that's the cause. Windows disables drag and drop on applications being run as admin.

—

Note: avoid special characters/non-ASCII characters in Meshroom and images file paths (#209)

## 20.6 Large scale dataset

Can I use Meshroom on large datasets with more than 1000 images?

Yes, the pipeline performance scales almost linearly. We recommend adjusting the SfM parameters to be a bit more strict, as you know that you have a good density / good connections between images. There are 2 global thresh-

olds on the Meshing node (`maxInputPoints` and `maxPoints`) that may need to be adjusted depending on the density/quality you need and the amount of RAM available on the computer you use.

Can I use Meshroom on renderfarm?

Meshroom has been designed to be used on renderfarm. It should be quite straightforward to create a new submitter, see the [available submitters](#) as examples. Contact us if you need more information to use it with a new renderfarm system.

## 20.7 Multi Camera Rig

If you shoot a static dataset with a moving rig of cameras (cameras rigidly fixed together with shutter synchronization), you can declare this constraint to the reconstruction algorithm.

Currently, there is no solution to declare this constraint directly within the Meshroom UI, but you can use the following file naming convention:

```
+ rig/ # "rig" folder
|-+ 0/ # sub-folder with the index of the camera (starting at 0)
|---- DSC_0001.JPG # Your camera filename (the is no constraint on the filename,
↪here "DSC_" prefix is just an example)
|---- DSC_0002.JPG
|-+ 1/ # sub-folder with the index of the camera
|---- DSC_0001.JPG
|---- DSC_0002.JPG
```

All images with the same name in different “rig/cameraIndex” folder will be declared linked together by the same transformation. So in this example, the relative pose between the 2 “DSC\_0001.JPG” images from the camera 0 and camera 1 will be the same than between the 2 “DSC\_0002.JPG” images.

When you drop your images into Meshroom, this constraint will be recognized and you will be able to see it in the CameraInit node (see Rig and Rig Sub-Pose of the Viewpoints parameter).

## 20.8 Error: This program needs a CUDA Enabled GPU

[error] This program needs a CUDA-Enabled GPU (with at least compute capability 2.0), but Meshroom *is* running on a computer with an NVIDIA GPU.

Solution: update/reinstall your drivers Details: [#182](#) [#197](#) [#203](#)

### 20.8.1 This Error message on a computer *without* NVIDIA GPU

The depth map computation is implemented with CUDA and requires an NVIDIA GPU.

[#218](#) [#260](#)

[Request] Remove CUDA dependency [alicevision/#439](#)

Currently, we have neither the interest nor the resources to do another implementation of the CUDA code to another GPU framework. If someone is willing to make this contribution, we will support and help for integration.\*

## 20.8.2 Can I use Meshroom without an NVIDIA GPU?

Yes, but you must use [Draft Meshing](#) to complete the reconstruction.

## 20.8.3 Does my GPU support CUDA?

Check <https://developer.nvidia.com/cuda-gpus>

# 20.9 Reconstruction parameters

The default parameters are optimal for most datasets. Also, many parameters are exposed for research & development purposes and are not useful for users. A subset of them can be useful for advanced users to improve the quality on specific datasets.

The first thing is to verify the number of reconstructed cameras from your input images. If a significant number are not reconstructed, you should focus on the options of the sparse reconstruction.

## 20.9.1 Sparse reconstruction

1. **FeatureExtraction:** Change `DescriberPreset` from `Normal` to `High` If your dataset is not big (<300 images), you can use `High` preset. It will take more time for the `StructureFromMotion` node but it may help to recover more cameras. If you have really few images (like <50 images), you can also try `Ultra` which may improve or decrease the quality depending on the image content.
2. **FeatureMatching:** Enable `Guided Matching` This option enables a second stage in the matching procedure. After matching descriptor (with a global distance ratio test) and first geometric filtering, we retrieve a geometric transformation. The guided-matching use this geometric information to perform the descriptors matching a second time but with a new constraint to limit the search. This geometry-aware approach prevents early rejection and improves the number of matches in particular with repetitive structures. If you really struggle to find matches it could be beneficial to use `BRUTE_FORE_L2` matching, but this is not good in most cases as it is very inefficient.
3. **Enable AKAZE as DescriptorTypes** on `FeatureExtraction`, `FeatureMatching` and `StructureFromMotion` nodes It may improve especially on some surfaces (like skin for instance). It is also more affine invariant than SIFT and can help to recover connections when you have not enough images in the input.
4. To improve the robustness of the initial image pair selection/initial reconstruction, you can use a SfM with `minInputTrackLength` set to 3 or 4 to keep only the most robust matches (and improve the ratio inliers/outliers). Then, you can chain another SfM with the standard parameters, so the second one will try again to localize the cameras not found by the first one but with different parameters. This is useful if you have only a few cameras reconstructed within a large dataset.

## 20.9.2 Dense reconstruction

1. **DepthMap**

You can adjust the `Downscale` parameter to drive precision/computation time. If the resolution of your images is not too high, you can set it to 1 to increase precision, but be careful, the calculation will be ~4x longer. On the contrary, setting it to a higher value will decrease precision but boost computation.

Reduce the number of neighbour cameras (SGM: Nb Neighbour Cameras, Refine: Nb Neighbour Cameras) will directly reduce the computation time linearly, so if you change from 10 to 5 you will get a 2x speedup. A minimum value of 3 is necessary, 4 already gives decent results in many cases if the density of your acquisition process regular enough. The default value is necessary in a large scale environment where it is difficult to have 4 images that cover the same area.

#### 2. DepthMapFilter

If you input images are not dense enough or too blurry and you have too many holes in your output. It may be useful to relax the Min Consistent Cameras and Min Consistent Cameras Bad Similarity to 2 and 3 respectively.

#### 3. Meshing

If you have less than 16G of RAM, you will need to reduce the Max Points to fit your RAM limits. You may also augment it, to recover a more dense/precise mesh.

#### 4. MeshFiltering

Filter Large Triangles Factor can be adjusted to avoid holes or on the other side to limit the number of large triangles. Keep Only The Largest Mesh: Disable this option if you want to retrieve unconnected fragments that may be useful.

#### 5. Texturing

You can change the Texture Downscale to 1 to improve the texture resolution.

### 20.9.3 Describer Types

You can choose to use one or multiple describer types. If you use multiple types, they will be combined together to help get results in challenging conditions. *The values should always be the same between FeatureExtraction, FeatureMatching and StructureFromMotion.* The only case, you will end up with different values is for testing and comparing results: in that case you will enable all options you want to test on the FeatureExtraction and then use a subset of them in Matching and SfM.

## 20.10 StructureFromMotion fails

StructureFromMotion may fail when there is not enough features extracted from the image dataset (weakly textured dataset like indoor environment). In this case, you can try to augment the amount of features:

- DescriberPreset to High or Ultra in FeatureExtraction
- Add AKAZE as DescriptorType on FeatureExtraction, FeatureMatching and StructureFromMotion nodes

Using more features will reduce performances on large datasets. Another problem is that adding too much features (less reliable) may also reduce the amount of matches by creating more ambiguities and conflicts during features matching.

- Guided Matching parameter on FeatureMatching is useful to reduce conflicts during feature matching but is costly in performance. So it is very useful when you have few images (like a cameras rig from a scan studio).

## 20.11 Supported image formats

Meshroom supports most image formats, including many RAW formats such as '.exr', '.rw2', '.cr2', '.nef', '.arw', ... The image importer is based on [OpenImageIO](#), so all formats supported by OpenImageIO can be imported to Mesh-



room. However it is recommended to use '.jpg', '.jpeg', '.tif', '.tiff' or '.png' at the moment.

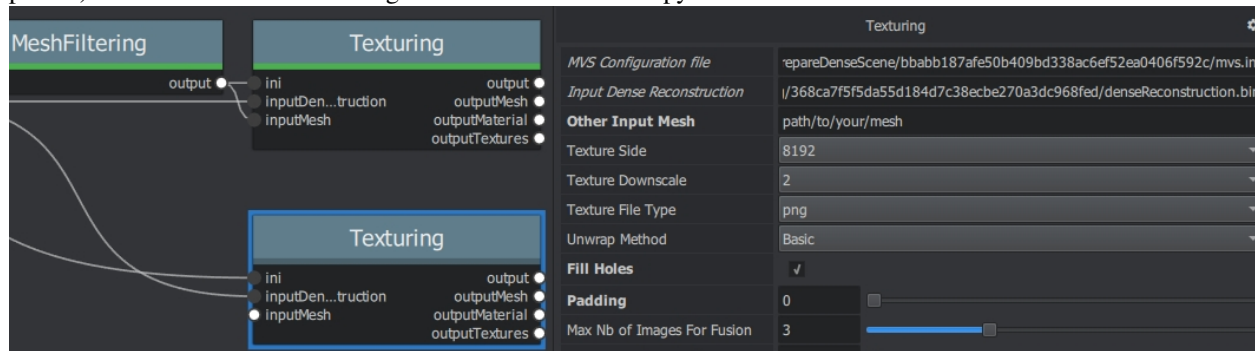
Note: On some datasets the reconstruction quality could be reduced or cause unexpected interruption of the pipeline. (#G) Convert your RAW image to '.jpg', '.jpeg', '.tif', '.tiff' or '.png' to resolve this problem.

## 20.12 Texturing after external re topology

It is possible to reproject textures after re-topology and custom unwrap. The only constraint is to **NOT** modify scale/orientation of the model, in order to stay in the same 3D space as the original reconstruction.

To retexture a user mesh, you need to remove the input connection on Texturing node's `inputMesh` (right click connection > Remove) and write the path to your mesh in the attribute editor. If you have custom UVs, they will be taken into account.

You can also duplicate the original Texturing node (right click > Duplicate) and make changes on this copy. It should look like this:



(optional) You can also set "Padding" to 0 and check "Fill Holes" instead if you want to completely fill texture's blank space with plausible values.

## 20.13 Troubleshooting

Things you can check/try:

- make sure the downloaded Meshroom files are not corrupted (incomplete/interrupted download)
- avoid special characters/non-ASCII characters in Meshroom and images file paths (#209)
- make sure your antivirus program does not interfere with Meshroom ((#178))/(#342))
- are you running Meshroom as Admin? (This will disable drag-and-drop on windows)
- Check your Python installation /reinstall as admin and check the PATH if there are any conflicts
- update/install latest NVIDIA drivers
- set your NVIDIA GPU as primary GPU for Meshroom. (NVIDIA Control Panel->Manage 3D Settings)
- Try the Meshroom 2018.1 release; when using windows 7 try the corresponding release (Meshroom 2019.1 has some problems with Texturing #449, DepthMap and some photo datasets which worked in 2018.1 #409. These problems will be addressed in the next release)
- Test Meshroom with the [Monstree](#) dataset
- Sometimes the pipeline is corrupted. Clear the cache for the node (and following nodes) with the error. Sometimes restarting the application / the computer might help. #201

- [check your images](#) for problems

### Text publications

...

### Videos

Meshroom live reconstruction (LADIO project)

[https://www.youtube.com/watch?v=DazLfZXU\\_Sk](https://www.youtube.com/watch?v=DazLfZXU_Sk)

Meshroom: Open Source 3D Reconstruction Software

[https://www.youtube.com/watch?v=v\\_O6tYKQEBA](https://www.youtube.com/watch?v=v_O6tYKQEBA)

How to 3D Photoscan Easy and Free!

mesh filtering 10:18 / 13:17 blender import

<https://www.youtube.com/watch?v=k4NTf0hMjtY>

Meshroom: 3D Models from Photos using this Free Open Source Photogrammetry Software

<https://www.youtube.com/watch?v=R0PDCp0QF1o>

Free Photogrammetry: Meshroom

<https://www.youtube.com/watch?v=NdpR6k-6SHs>

MeshRoom Vs Reality Capture with blender

<https://www.youtube.com/watch?v=voNKSkuP-RY>

MeshRoom and Blender walkthrough

<https://www.youtube.com/watch?v=VjBMfVC5DSA>

Meshroom and Blender photoscanning tutorial (+ falling leaf animation)

[https://www.youtube.com/watch?v=3L\\_9mf2s2lw](https://www.youtube.com/watch?v=3L_9mf2s2lw)

Meshroom Introductory Project Tutorial

<https://www.youtube.com/watch?v=bYzi5xYIYPU>

Meshroom: Camera Sensor DB Error

<https://www.youtube.com/watch?v=EOc4Utsk2U>

How to 3D Photoscan your Face for Free!

<https://www.youtube.com/watch?v=9UI9aYhm7O4>

Meshroom: créez des objets 3D à partir de photos, grâce à une solution libre... — François Grassard

<https://www.youtube.com/watch?v=CxKzHJEff4w>

Meshroom vs 3DZephyr vs Dronemapper Part 1

<https://www.youtube.com/watch?v=zfj9u84bQUs>

Meshroom vs 3DZephyr vs Dronemapper Part 2

<https://www.youtube.com/watch?v=qyIW3cvtbiU>

Character Photogrammetry for Games - Part 1 - Meshroom

[https://www.youtube.com/watch?v=GzDE\\_K\\_x9eQ](https://www.youtube.com/watch?v=GzDE_K_x9eQ)

Meshroom | Photoscan to Camera Track (Matchmove)

<https://www.youtube.com/watch?v=1dhdEmGLZhY>

Photogrammetry 2 – 3D scanning simpler, better than ever!

<https://www.youtube.com/watch?v=1D0EhSi-vvc>

**Alicevision** Short description

**CCTAG** The CCTAG marker system can deliver sub-pixel precision while being largely robust to challenging shooting conditions. <https://github.com/alicevision/CCTag>

**SIFT**

**Meshroom uses PopSIFT, Scale-Invariant Feature Transform (SIFT).** This library provides a GPU implementation of SIFT. <https://github.com/alicevision/popsift>



## 23.1 About Meshroom

Meshroom is a free, open-source 3D Reconstruction Software based on the [AliceVision](#) framework. AliceVision is a Photogrammetric Computer Vision Framework which provides 3D Reconstruction and Camera Tracking algorithms. AliceVision aims to provide strong software basis with state-of-the-art computer vision algorithms that can be tested, analyzed and reused. The project is a result of collaboration between academia and industry to provide cutting-edge algorithms with the robustness and the quality required for production usage.

### 23.1.1 Project history

In 2010, the [IMAGINE](#) research team (a joint research group between [Ecole des Ponts ParisTech](#) and [Centre Scientifique et Technique du Batiment](#)) and Mikros Image started a partnership around [Pierre Moulon's thesis](#), supervised by Renaud Marlet and Pascal Monasse on the academic side and Benoit Maugean on the industrial side. In 2013, they released an open source SfM pipeline, called openMVG (“Multiple View Geometry”), to provide the basis of a better solution for the creation of visual effects [matte-paintings](#).

In 2009, the CMP research team from CTU started Michal Jancosek's PhD thesis supervised by Tomas Pajdla. They released Windows binaries of their MVS pipeline, called CMPMVS, in 2012.

In 2009, INPT, INRIA and Duran Duboi started a French ANR project to create a model based Camera Tracking solution based on natural features and a new marker design called CCTag.

In 2015, Simula, INPT and Mikros Image joined their efforts in the EU project [POPART](#) to create a Previz system. In 2017, CTU joined the team in the EU project [LADIO](#) to create a central hub with structured access to all data generated on set.

### 23.1.2 Partners

[Czech Technical University \(CTU\)](#) in Prague, Czech Republic

[IMAGINE](#) from the [Universite Paris Est](#), [LIGM Gaspard-Monge](#), France

Institut National Polytechnique de Toulouse (INPT), France

Mikros Image , Post-Production Company in Paris, France

Simula Research Laboratory AS in Oslo, Norway

Quine in Oslo, Norway

See Github authors for the full list of contributors.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, see [POPART](#), Project ID: 644874 and [LADIO](#), project ID: 731970.



### 23.1.3 Open Source

We build a fully integrated software for 3D reconstruction, photo modelling and camera tracking. We aim to provide a strong software basis with state-of-the-art computer vision algorithms that can be tested, analyzed and reused. Links between academia and industry is a requirement to provide cutting-edge algorithms with the robustness and the quality required all along the visual effects and shooting process. This open approach enables both us and other users to achieve a high degree of integration and easy customization for any studio pipeline.

Beyond our project objectives, open source is a way of life. We love to exchange ideas, improve ourselves while making improvements for other people and discover new collaboration opportunities to expand everybody’s horizon.

## 23.2 About the manual

This manual is a compilation of the resources found on [alicevision.github.io](https://alicevision.github.io), breadcrumbs of information collected from github issues, other web resources and new content, created for this manual.

WORK IN PROGRESS! (last update 03.06.19)

You want to help? Missing something?

You are welcome to comment and contribute. This document is in “Suggest edits” mode.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. This is a Meshroom community project.





All product names, logos, and brands are property of their respective owners. All company, product and service names used in this document are for identification purposes only. Use of these names, logos, and brands does not imply endorsement.

## 23.3 Acknowledgements

A big thanks to the many researchers, who made their work available online so we can provide free, additional background information with this guide through references.

And finally thank **you** for using Meshroom, testing, reporting issues and sharing your knowledge.

To all Meshroom contributors: keep up the good work!

## 23.4 Contact us

You can contact us on the public mailing list at [alicevision@googlegroups.com](mailto:alicevision@googlegroups.com)

You can also contact us privately at [alicevision-team@googlegroups.com](mailto:alicevision-team@googlegroups.com)

## 23.5 Contributing

Alice Vision relies on a friendly and community-driven effort to create an open source photogrammetry solution.

The project strives to provide a **pleasant** environment for everybody and tries to be as **non-hierarchical** as possible. Every **contributor** is considered as a member of the team, regardless if they are a newcomer or a long time member. Nobody has **special** rights or prerogatives. The contribution workflow relies on [Github Pull Request](#) . We recommend to discuss new features before starting the development, to ensure that development is efficient for everybody and minimize the review burden.

In order to foster a friendly and cooperative atmosphere where technical collaboration can flourish, we expect all members of the community to be courteous, polite and respectful in their treatment of others helpful and constructive in suggestions and criticism stay on topic for the communication medium that is being used be tolerant of differences in opinion and mistakes that inevitably get made by everyone.

### Join us on Github

<https://github.com/alicevision/>

## 23.6 List of contributors

## 23.7 Licenses

This manual **This manual is licensed under a** ‘<<http://creativecommons.org/licenses/by-sa/4.0/>>’\_ Creative Commons Attribution-ShareAlike 4.0 International License . This is a Meshroom community project.

Meshroom is released under [MPLv2](#)

### 23.7.1 Third parties licenses

- **Python** <https://www.python.org> Copyright (c) 2001-2018 Python Software Foundation Distributed under the [PSFL V2](#) .
- **Qt/PySide2** <https://www.qt.io> Copyright (C) 2018 The Qt Company Ltd and other contributors. Distributed under the [LGPL V3](#) .
- **qmlAlembic** <https://github.com/alicevision/qmlAlembic> Copyright (c) 2018 AliceVision contributors. Distributed under the [MPL2](#) license .
- **QtOIO** <https://github.com/alicevision/QtOIO> Copyright (c) 2018 AliceVision contributors. Distributed under the [MPL2](#) license .



#### Documentation

Meshroom is a free, open-source 3D Reconstruction Software based on the [AliceVision](#) framework.

AliceVision is a Photogrammetric Computer Vision Framework which provides 3D Reconstruction and Camera Tracking algorithms. AliceVision aims to provide strong software basis with state-of-the-art computer vision algorithms that can be tested, analyzed and reused. The project is a result of collaboration between academia and industry to provide cutting-edge algorithms with the robustness and the quality required for production usage.

## A

Alicevision, [97](#)

## S

SIFT, [97](#)