Meshroom

Release v2021.0.1

Meshroom Contributors

Jul 14, 2021

Revision: v2021.0.1
https://alicevision.org

FIRST STEPS

Meshroom is a free, open-source 3D Reconstruction Software based on the AliceVision[1] framework.

AliceVision is a Photogrammetric Computer Vision Framework which provides 3D Reconstruction and Camera Tracking algorithms. AliceVision comes up with strong software basis and 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.

[1] https://github.com/alicevision/AliceVision

CHAPTER

ONE

INSTALLATION

1.1 Requirements

Warning: Meshroom requires an NVIDIA GPU card with a CUDA compute capability >= 3.0 for the MVS part. You can check your CUDA Properties here[2] or on the NVIDIA dev page[3].

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

Here are the minimum requirements for Meshroom:

Minimum requirements	
Operating systems	Windows x64, Linux, macOS (some work required)
CPU	Recent Intel or AMD cpus
RAM	8 GB
Hard Drive	~400 MB for Meshroom + space for your data

To obtain better performances on a desktop/laptop machine the recommended requirements are:

Recommended requirements	
CPU	Intel Core i7 or AMD Ryzen 7
RAM	32 GB
Hard Drive	20 GB+ HDD or SSD
GPU	Recent NVIDIA GPU

1.2 Getting Meshroom

1.2.1 Pre-built binaries

Meshroom binaries for Windows platform and Linux platform can be downloaded from here[4].

Prebuilt binaries on this page are all-in-one packages including AliceVision and all required resources. The pre-built binaries also contain the `meshroom_compute` and `meshroom_photogrammetry` to run and create pipelines from the command line.

[2] https://github.com/tpruvot/ccminer/wiki/Compatibility
[3] https://developer.nvidia.com/cuda-gpus#compute
[4] https://alicevision.github.io/#meshroom

Note: Check the changelog[5] to see what features are included in the release.

Platform specific instructions:

Platforms

Windows

1. Download Meshroom from Meshroom home page[6]
2. extract ZIP to a folder of your choice
3. If you don't have it installed already, you need to install the Microsoft Visual C++ Redistributable Package 2015, 2017 and 2019 available on Microsoft's Support portal[7].
4. You can start Meshroom by clicking on the executable. No extra installation is required.

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

Linux

1. Extract the .tar.gz file in any folder.

```
tar -xf Meshroom-2020.1.0-linux-cuda10.tar.gz

cd Meshroom-2020.1.0
```

2. From this folder run:

```
./Meshroom
```

to launch the GUI.

From The Arch User Repository

```
yay --needed -S popsift uncertainty-framework cuda

yay -S meshroom

Meshroom
```

See the AUR page[8] for more information.

[5] https://github.com/alicevision/meshroom/blob/develop/CHANGES.md
[6] https://alicevision.org/#meshroom
[7] https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads.
[8] https://aur.archlinux.org/packages/alice-vision/

Set up environment

Meshroom requires a build of AliceVision and need to have AliceVision[9] installation in your `PATH` and `LD_LIBRARY_PATH`.

Your `PATH` should contain the folder where the AliveVision binaries can be found. Suppose ALICEVISION_INSTALL contains the directory where the library is installed, then

```
PATH=$PATH:${ALICEVISION_INSTALL}/bin
```

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:$LD_LIBRARY_PATH
```

You may need to adjust the folder `/usr/lib/nvidia-340` with the correct driver version (e.g. `330`, `350` etc..).

We suggest to create a bash executable `meshroom.sh` in the root of the meshroom folder to ease the task:

```
#!/bin/bash

# this should point to the installation folder of AliceVision, for the pre-built binaries
# it would be the full path to the folder aliceVision
export ALICEVISION_INSTALL=/path/to/aliceVision

# if you are using the plugins, here list all the paths to find them
#f or the pre-built binaries it is the full path to the folder qtPlugins/qml/
export QML2_IMPORT_PATH=/path/to/qmlAlembic/build/install/qml:/path/to/QtAliceVision/
↪build/install/qml:/path/to/QtOIIO/build/install/qml/:$QML2_IMPORT_PATH

# location of the sensor database
export ALICEVISION_SENSOR_DB=${ALICEVISION_INSTALL}/share/aliceVision/cameraSensors.db

# adjust according to your driver and cuda version
export LD_LIBRARY_PATH=${ALICEVISION_INSTALL}/lib:/usr/lib/nvidia-384:/usr/local/cuda-8.
↪0/lib64/:$LD_LIBRARY_PATH

# the meshroom path (the current directory)
export MESHROOMPATH=$PWD

# this line launch whatever script and relevant options that are given as input ($@)
PYTHONPATH=${MESHROOMPATH} PATH=$PATH:${ALICEVISION_INSTALL}/bin python ${MESHROOMPATH}
↪/$@
```

Then you can also create an executable `meshroom_ui.sh` to launch the GUI:

```
#!/bin/bash
./meshroom.sh meshroom/ui $@
```

Don't forget to make the two files executable:

[9] https://github.com/alicevision/AliceVision

```
chmod +x meshroom.sh meshroom_ui.sh
```

Launch the User Interface

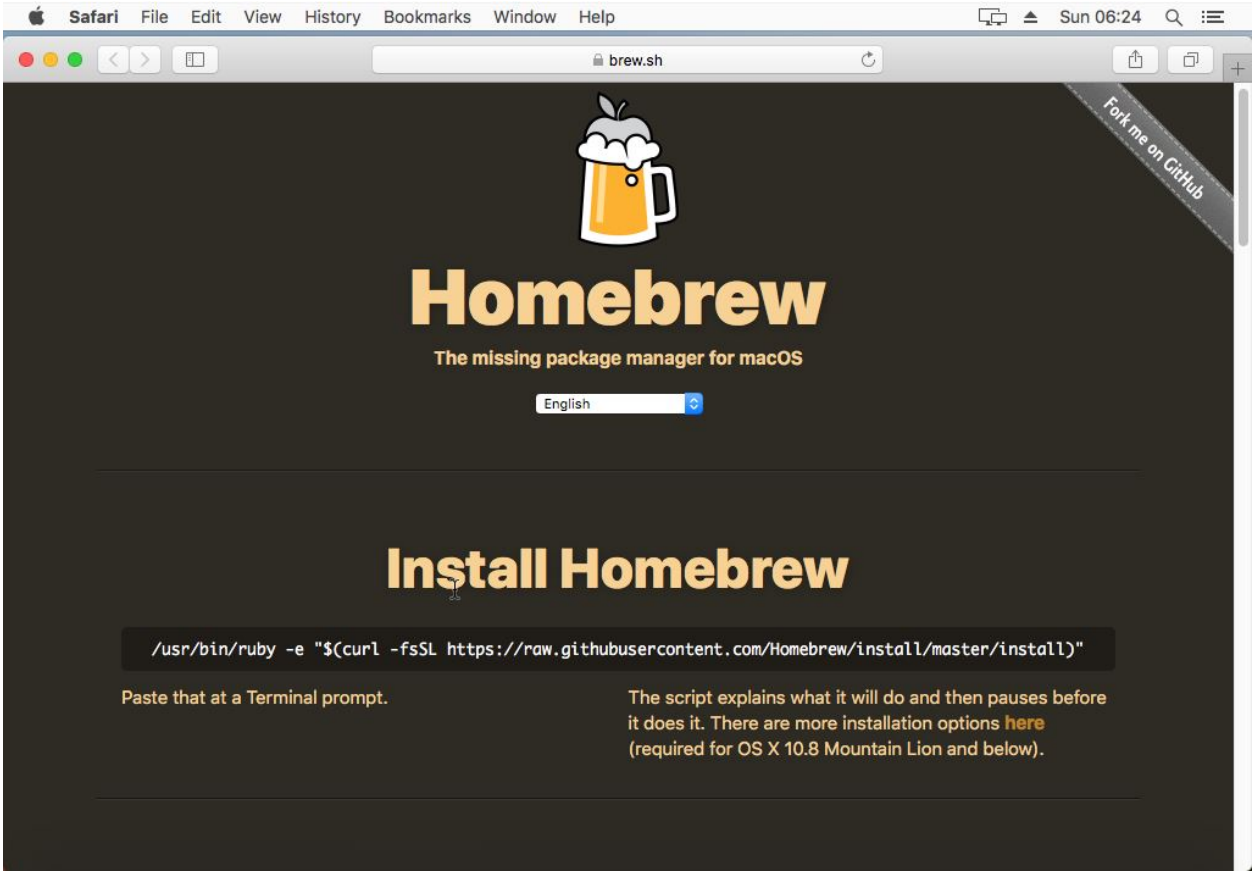
To launch the user interface simply use the previous shell script:

```
# launch the gui
./meshroom_ui

# launch the gui with e.g. a given Project
./meshroom_ui --project myProject.mg

# launch with --help for the list of supported parameters
./meshroom_ui --help
```

MacOS

Most of MacOS workstations do not have any NVIDIA graphic boards, hence they cannot run CUDA, for MVS part. So compiling and using Meshroom is not exactly straightforward[10]. However, Ryan Baumann has compiled his own Homebrew tap[11] which includes the necessary formulae, and you can use this post to get an idea of 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.

[10] https://github.com/alicevision/AliceVision/issues/444
[11] http://github.com/ryanfb/homebrew-alicevision



System Requirements

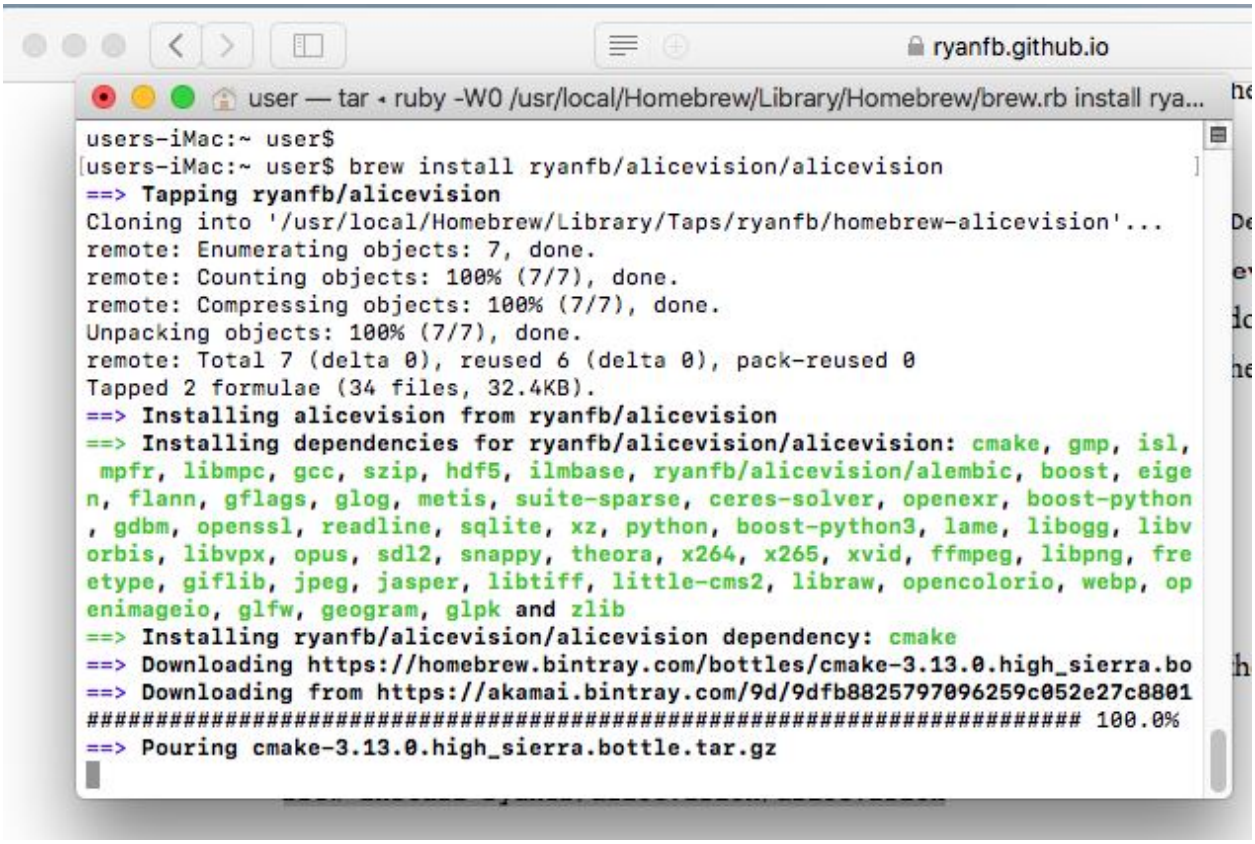
First off, your Mac will currently need some NVIDIA GPU with a CUDA compute capability of 2.0 or greater. This is probably a pretty small portion of all Macs available, but you can check your GPU by looking in "About This Mac" from the Apple icon in the top left corner of the screen, under "Graphics". If you have an NVIDIA GPU listed there, you can check its compute capability on the NVIDIA CUDA GPUs page[12].

Second, you're going to need to install the latest CUDA toolkit[13]. As of this writing, that's CUDA 10.1, 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 (but not Mojave!) if you haven't already. Alongside this it is aloso suggested to instal the latest NVIDIA CUDA GPU webdriver, which as of this writing is 387.10.10.10.40.118[14].

Third, *CUDA 10.1 is only compatible with the version of ``clang`* distributed with Xcode 10.1 <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 10.1. To get back to Xcode 10.1, what you can do is go to Apple's Developer Downloads page[15] (for which you'll need a free Apple developer account), then search for "Xcode 10.1", then install the Command Line Tools for Xcode 10.1 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 10.0.0`.

[12] https://developer.nvidia.com/cuda-gpus
[13] https://developer.nvidia.com/cuda-downloads
[14] https://www.nvidia.com/download/driverResults.aspx/142160/en-us
[15] https://developer.apple.com/download/more/

Once you've done all this, you can verify a working CUDA install by going to `/Developer/NVIDIA/CUDA-10.1/samples/1_Utilities/deviceQuery` and running `sudo make && ./deviceQuery`, which should output your GPU information. If it doesn't build correctly (i.e. you see `nvcc fatal : The version ('??.?') of the host compiler ('Apple clang') is not supported`), 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).



The following instructions also assume a working Homebrew[16] install.

MacOS 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
```

[16] https://brew.sh/

Meshroom Installation & Usage

This tutorial does not provide a Homebrew formulae for the Meshroom package itself[17], 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 `pip3` instead of `pip` and `python3` instead of `python`):

```
wget 'https://github.com/alicevision/meshroom/archive/v2019.1.0.zip'
unzip v2019.1.0.zip
cd meshroom-2019.1.0
pip install -r requirements.txt
```

Note: 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.10.1.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[18]

```
for i in /Developer/NVIDIA/CUDA-10.1/lib/*.a /Developer/NVIDIA/CUDA-10.1/lib/*.dylib; do␣
↪ln -sv "$i" "/usr/local/lib/$(basename "$i")"; done
```

You can undo/uninstall this with:

```
for i in /Developer/NVIDIA/CUDA-10.1/lib/*.a /Developer/NVIDIA/CUDA-10.1/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/2.1.0/share/aliceVision/vlfeat_K80L3.SIFT.
↪tree
```

Then launch with:

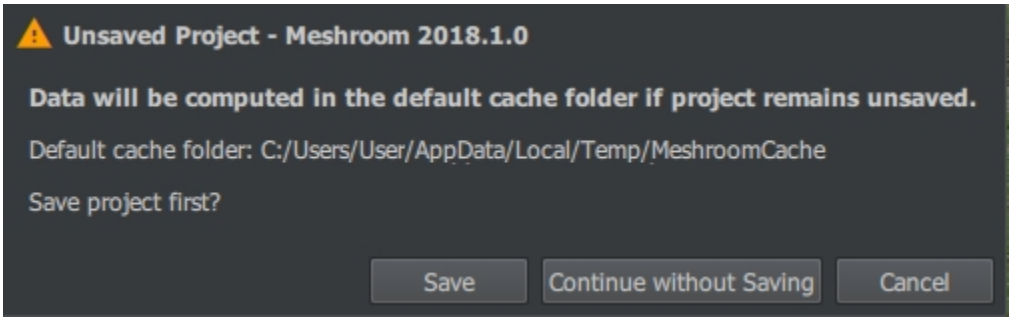
```
ALICEVISION_SENSOR_DB=/usr/local/Cellar/alicevision/2.1.0/share/aliceVision/
↪cameraSensors.db ALICEVISION_VOCTREE=/usr/local/Cellar/alicevision/2.1.0/share/
↪aliceVision/vlfeat_K80L3.SIFT.tree PYTHONPATH=$PWD python meshroom/ui
```

Import some photos, click "Start", wait a while, and hopefully you should end up with a reconstructed and textured mesh (here's an example of my own which I uploaded to SketchFab[19]). By default, the output will be in `MeshroomCache/Texturing/` (relative to where you saved the project file).

By default, the output will be in MeshroomCache/Texturing/ (relative to where you saved the project file).

[17] https://github.com/alicevision/meshroom
[18] https://ryanfb.github.io/etc/2018/08/17/alicevision_and_meshroom_on_mac_os_x.html#fn:dyldpath
[19] https://skfb.ly/6ARpx

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-10.1/lib" ' + cmdPrefix + chunk.
   ↪node.nodeDesc.commandLine.format(**chunk.node._cmdVars) + cmdSuffix
   ```

   ` <https://ryanfb.github.io/etc/2018/08/17/alicevision_and_meshroom_on_mac_os_x.html#fnref:dyldpath>`___

Originally published on 2018-08-17 by Ryan Baumann[20]

This guide was updated on 2019-03-20 to reflect the latest CUDA 10.1 and Xcode 10.1 versions. The Homebrew formula was also updated to AliceVision 2.1.0 to support Meshroom 2019.1.0.

Modified for the Meshroom documentation 2019-07-25

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 (accessed 25 Jul 2019).

Docker

An official docker image of Meshroom can be found on Docker Hub[21]. The relevant Dockerfile can be found in the root directory of the sources[22]

The image is based on the NVIDIA docker which needs to be installed. You can follow the official NVIDIA tutorial here[23].

To execute the docker image:

```
docker pull alicevision:meshroom
docker run -it --runtime=nvidia meshroom
```

[20] https://ryanfb.github.io/
[21] https://hub.docker.com/r/alicevision/meshroom
[22] https://github.com/alicevision/meshroom/blob/master/Dockerfile
[23] https://github.com/nvidia/nvidia-docker/wiki/Installation-(version-2.0)

Google Colaboratory

https://github.com/alicevision/meshroom/wiki/Meshroom-in-Google-Colab-(cloud)

1.2.2 From the sources

Instructions available here[24].

[24] https://github.com/alicevision/meshroom/blob/develop/INSTALL.md

CHAPTER

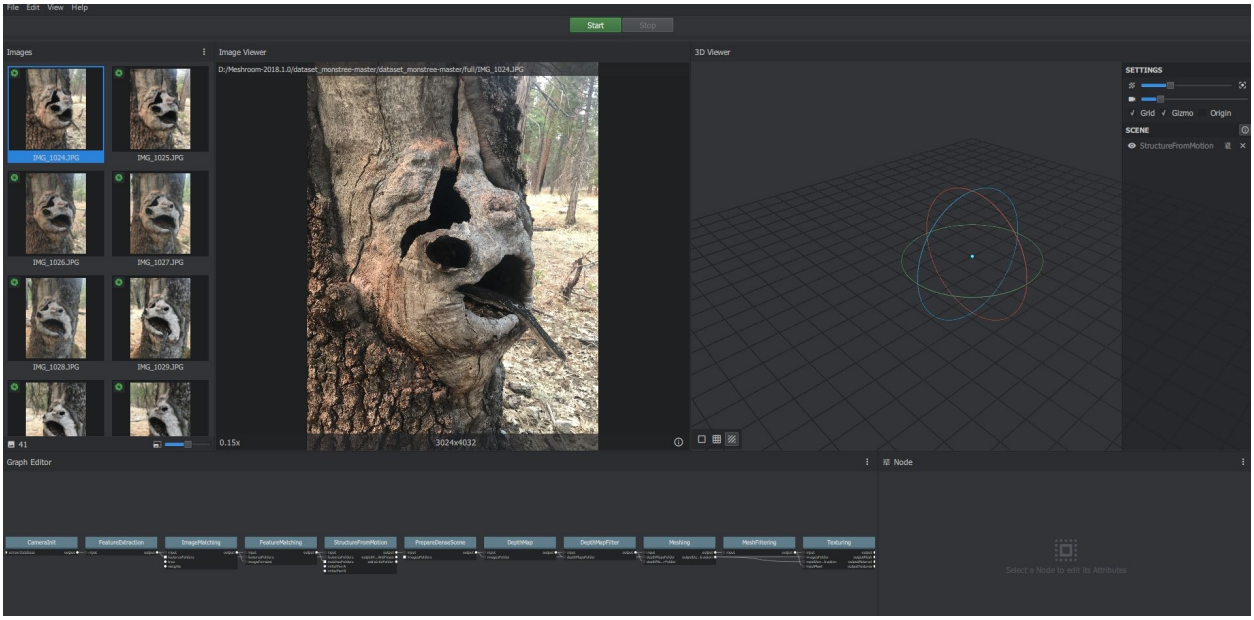
TWO

TEST MESHROOM

For your first reconstruction in Meshroom, download the Monstree Image Dataset[25]. You can preview the Monstree model on Sketchfab[26].

The Monstree dataset is known to work, so there should be no errors during the reconstruction. This might be different when using your own image dataset.

Import the images in Meshroom by dragging and dropping them in the Images pane (left). Alternatively, you can use the file dialog (File -> Import Images). There are different folders in the Monstree dataset: full (all images), mini6 (6 images) and mini3 (3 images) to test out.



Press the 'Start' button (top) to run the reconstruction pipeline. A progress bar will appear under the button. When the progress bar gets to the end, the reconstruction is done. This should take no more than 30 minutes on recent hardware. Double-click the 'Texturing' node to load the final 3D output into the viewer. Congratulations! You have successfully used Meshroom!

[25] https://github.com/alicevision/dataset_monstree
[26] https://sketchfab.com/models/92468cb8a14a42f39c6ab93d24c55926

CHAPTER

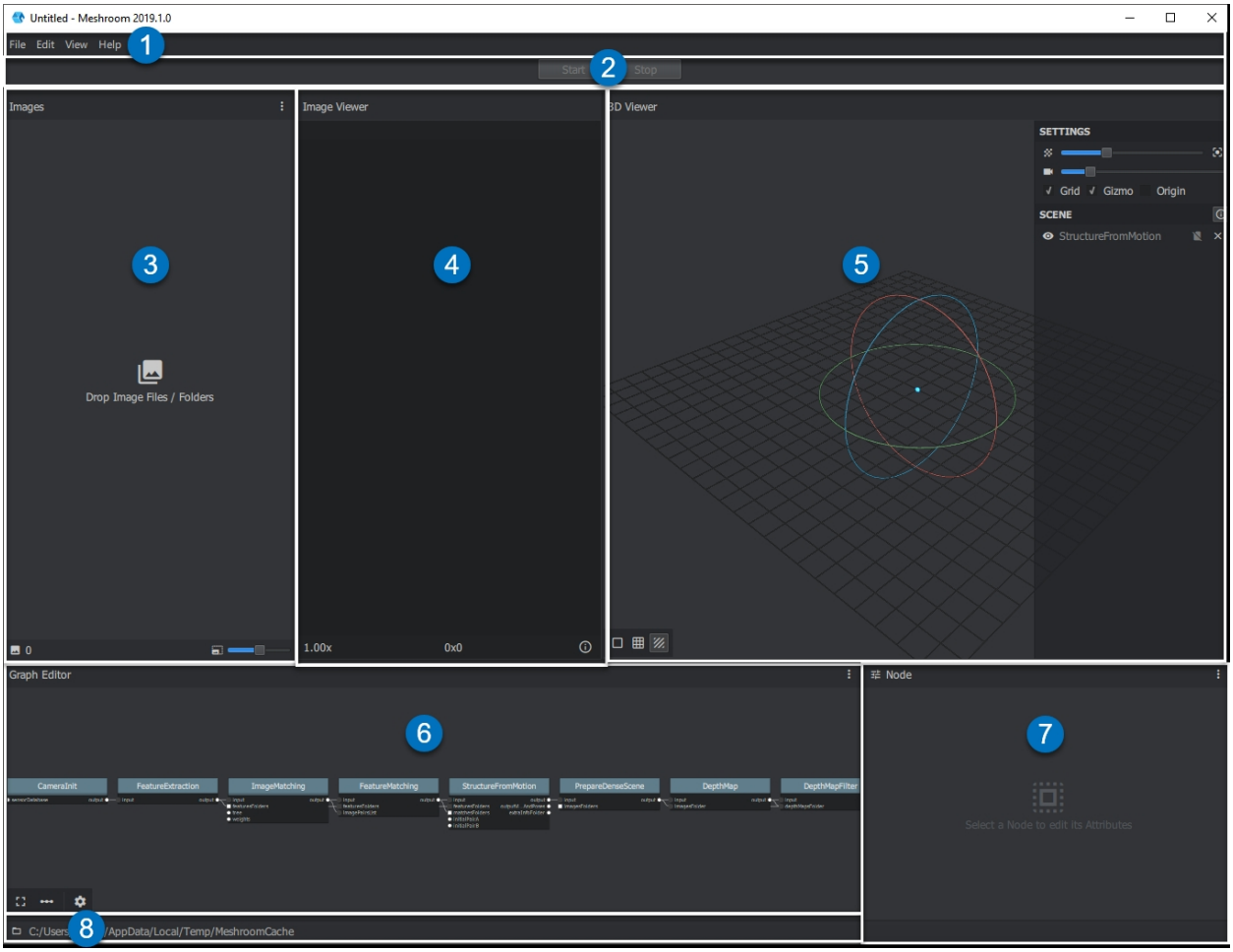
THREE

GRAPHICAL USER INTERFACE FEATURES

When you first start Meshroom, two windows open:

- the Command-line interface window (You can ignore or minimize this window. Do not close this window or Meshroom will terminate).
- the main Graphical User Interface (GUI) with different panes:

1. Menu bar: File / Edit / 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)*

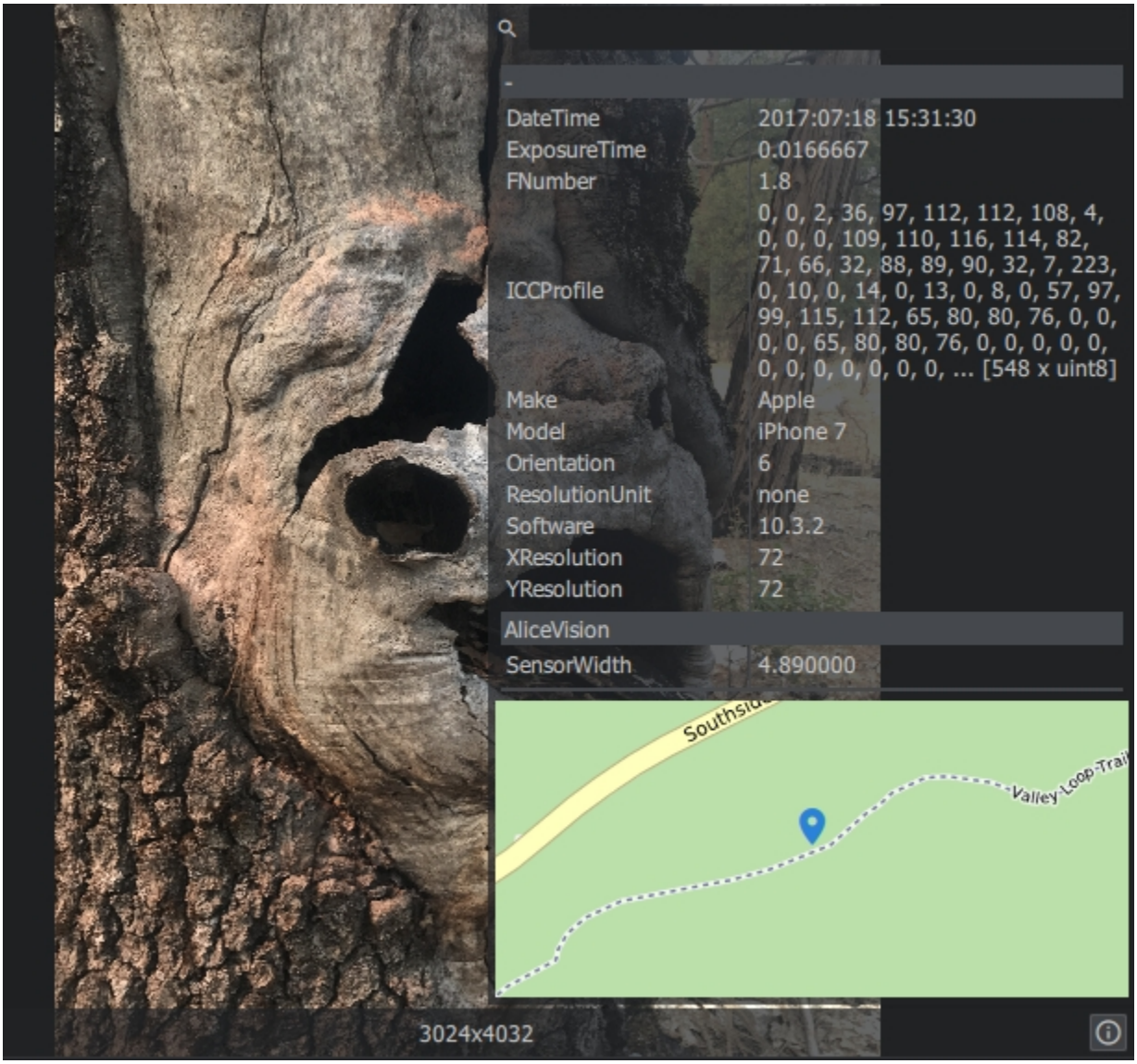


You can grab a Pane border and move it to change the **pane** size.

3.1 Import Images

Drag and 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.



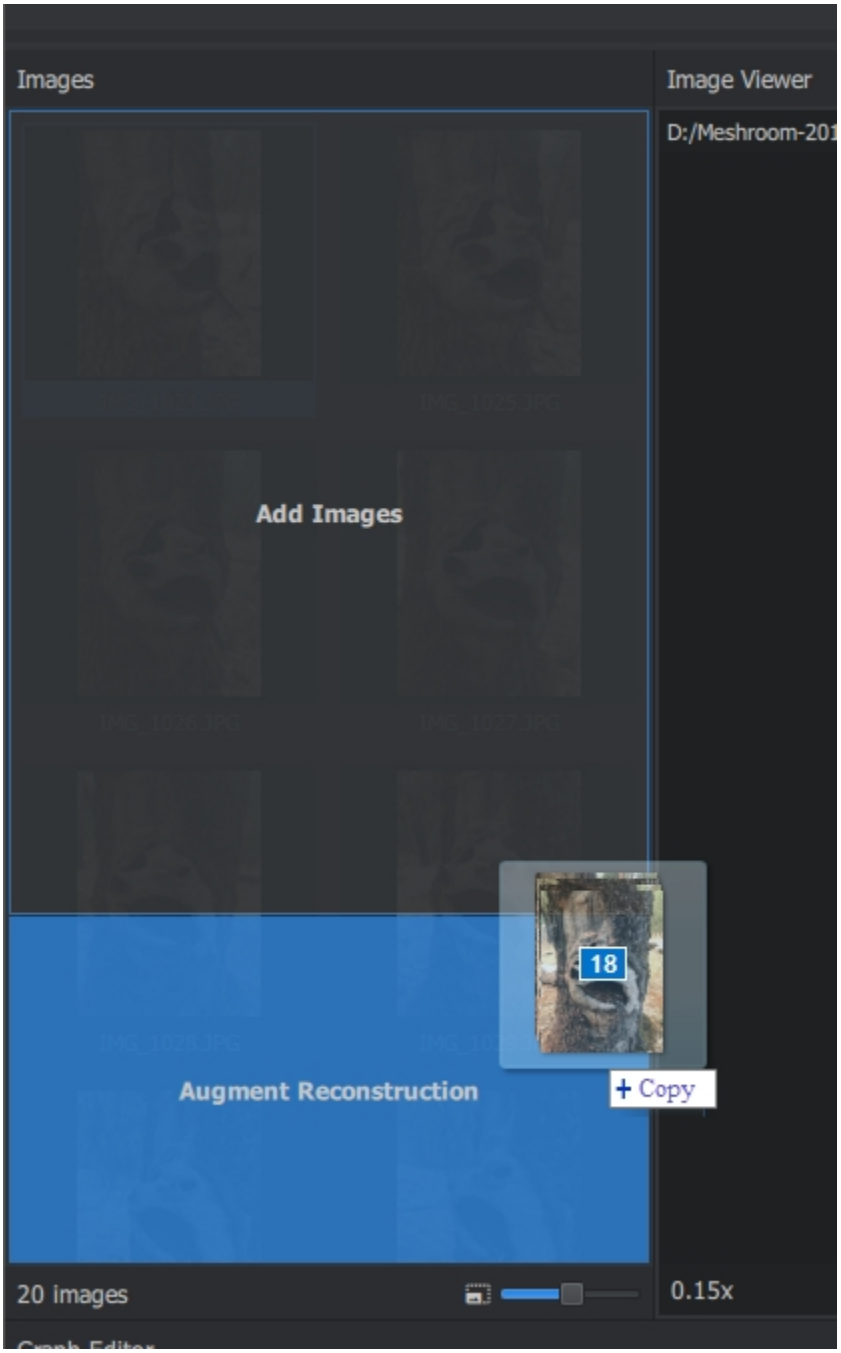
Note: If your images won't appear in the **Images** pane after you imported them, your camera was not recognized correctly. Make sure the EXIF data contains all relevant camera information. If the import still fails, your camera is not in the database or your image files are not valid.

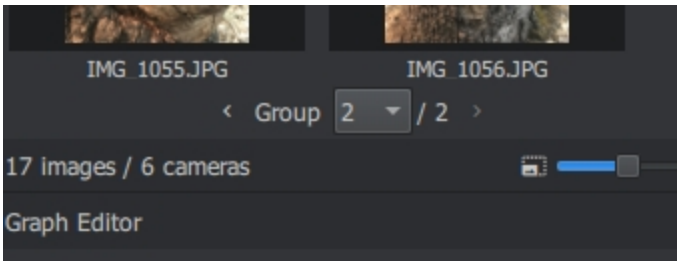
3.1.1 Augment Reconstruction

You can drag and 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: The groups will be merged using the ImageMatchingMultiSfM node. Read the node description for details


Images
Image Viewer
D:/Meshroom-201
Add Images
18
Augment Reconstruction
+ Copy
20 images
0.15x



3.1.2 Live Reconstruction

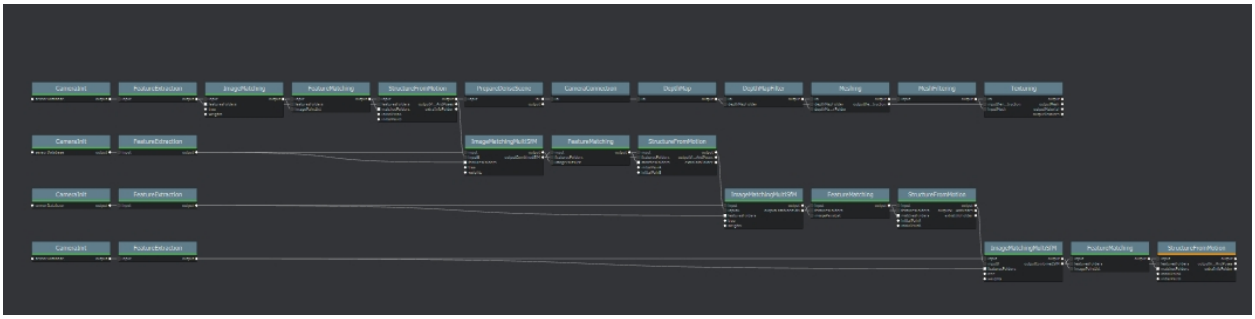
Live reconstruction is meant to be used along with a camera that can transfer images to a computer while shooting (using wifi, a wifi sd-card or Tethering). Meshroom can watch a folder for new images and successively augment previous SfM (point clouds + cameras) after each {Min. Images} per Step. This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

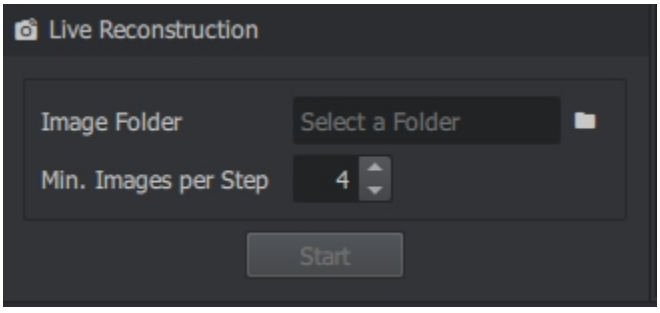
To enable **Live Reconstruction** go to the menu bar **View :math:`Rightarrow` Live Reconstruction** A new **Live Reconstruction** pane will appear under the Images pane.

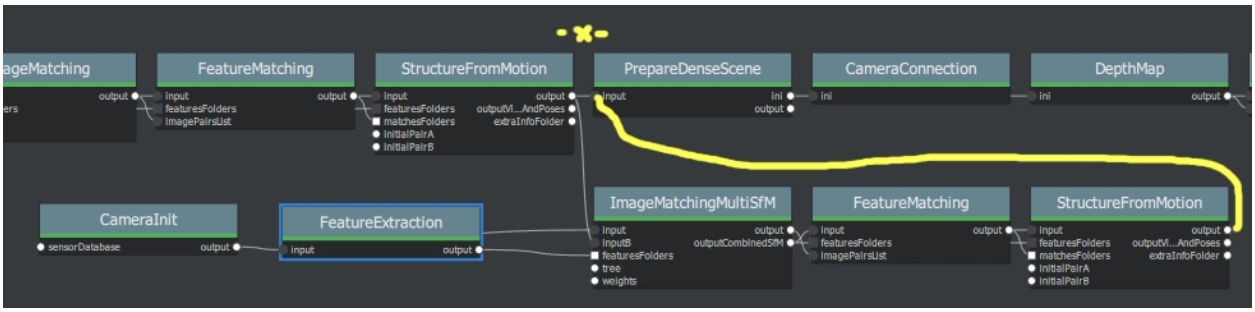
For each new import, a new **Image Group** inside the **Images** pane will be created. Also the **Graph Editor** updates the graph, adding nodes to process the newly added images and add them to the pipeline.

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

3.2 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**.

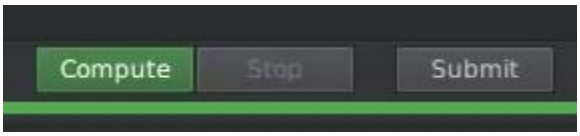
You can open the (Your-Project-Folder) ⇒ **MeshroomCache** to see the output of each node. (Shortcut: Icon and path at the bottom left side of the main window)

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.

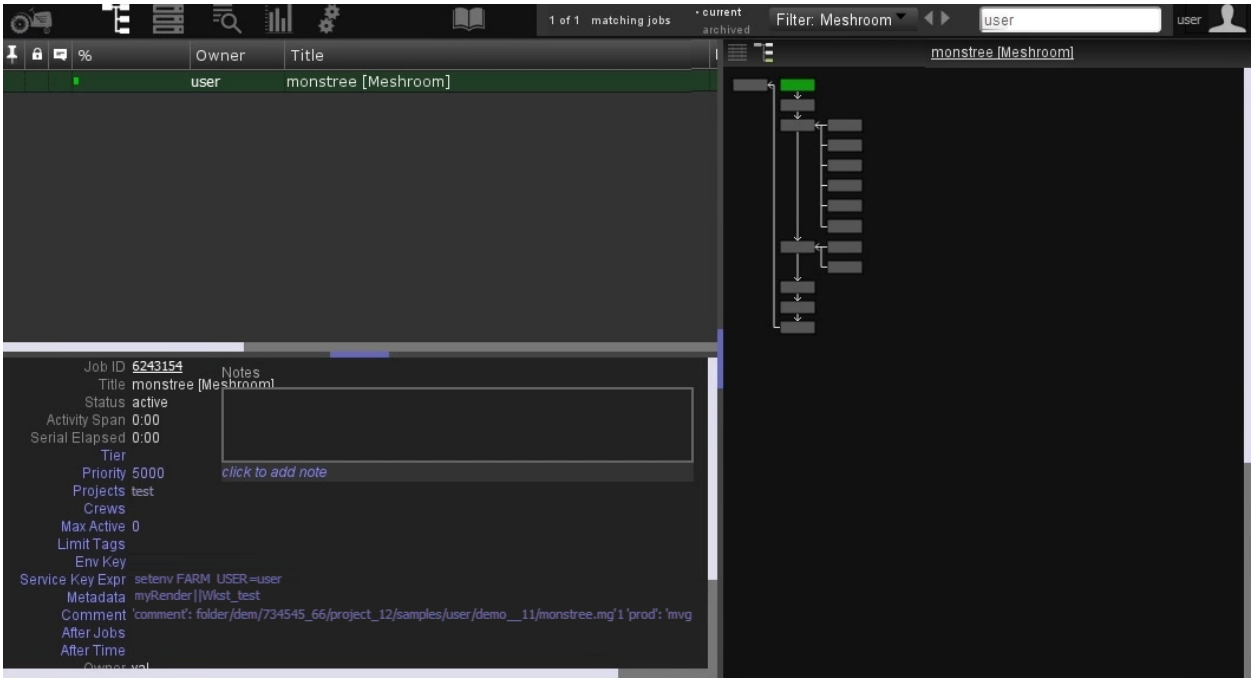
You can change the **name of the output folders** of your nodes by clicking on the node and changing the **Output Folder** name in the **Attributes tab** of the **Graph Editor** Properties pane.

3.2.1 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
- [WIP] Fireworks (https://materialsproject.github.io/fireworks/)
- [WIP] OpenCue (https://www.opencue.io/)

3.3 Graph Editor

3.3.1 Controls

Move:

- left mouse + shift
- middle mouse

Zoom: scroll wheel

Add node menu: right mouse

Fit: key f

Delete selected node: key delete

3.3.2 Nodes

A node can be selected by left-clicking on it. The output of a node can be viewed by double-clicking on it (either in the image viewer or the 3D viewer depending on the node). Right-clicking on a node opens a menu.

For information on how nodes work, see the *core documentation*.

3.3.3 Edges

Edges can be deleted by right-clicking them and selecting 'Remove'. Edges can be created by left-clicking on the vertex and dragging it to another.

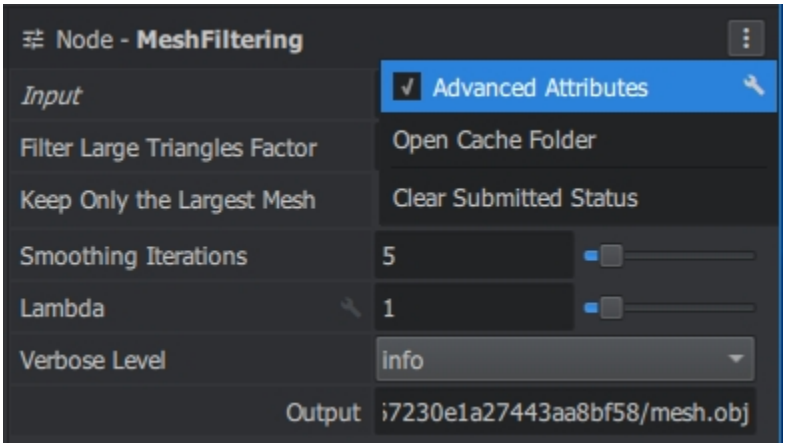
3.3.4 Editor

Attributes

The name of the attribute is bold when it is not set to its default value. Set to the default value by right-clicking on it.

Advanced Node Parameters

The goal is to separate experimental/debug/advanced from end-user attributes. The attribute editor provides an additional option to show/hide those advanced parameters.



Log

Progress can be monitored and any errors will appear here.

Statistics

Statistics about hardware reasources usages will appear here.

Status

Debug status of node.

Documentation

Briefly describes what the node does and how it works.

3.4 Image Viewer

3.4.1 Controls

Drag:

- left mouse + shift
- middle mouse

Zoom: scroll wheel

Context menu: right mouse

3.4.2 Display Features

Display an overlay showing the features used in the reconstruction.

3.4.3 Select Image Type

Input image, depth map and sim map can be viewed. Double click either a `DepthMap` or `DepthMapFilter` node to view the outputs of that node.

3.4.4 View Depth Map In 3D

The depth map will be loaded into the 3D viewer. Be aware that lots of RAM is used.

3.4.5 `StructureFromMotion` Statistics

There are 2 types of statistics that can be shown:

Statistics for that paticular image is the reprojection errors, observations scale, observations lengths.

Global statistics are residuals per view, landmarks per view, observations lengths per view.

3.4.6 Image Metadata

All of the metadata associated with that image such as make and model.

3.5 3D Viewer

3.5.1 Controls

Rotate: left mouse

Move:

- left mouse + shift
- middle mouse
- double click

Zoom: scroll wheel

Context menu: right mouse

Reset camera position: key f

Set render mode: keys 1 - 3

3.5.2 Render Modes

Solid: mesh with no texture

Wireframe: same as solid but edges are marked

Textured: mesh with texture

3.5.3 Camera

Camera perspectives can be selected and synchronized with the selected image in the images pane. Image overlay can be enabled and opacity of the image overlay changed.

3.5.4 Media Library

Multiple 3D files can be viewed at the same time using the media library. Visibility of files can controlled by clicking the button. Statistics relevant to the file can be shown.

CHAPTER

FOUR

CORE FEATURES

4.1 Nodes

Meshroom uses a nodes to control the reconstruction. A node represents a task that can be executed.

4.1.1 Attributes

This is how we control what goes in the node and what comes out. There are many types of attributes that are used.

4.1.2 Edges

File attributes can be connected. This represents a dependency for that node. If the output of node A and node B is connected to the input of node C, C will not be executed until A and B are successful. This makes it easy to set up complicated workflows.

4.1.3 Hashes

Every node has a hash based on its attributes. If an attribute that changes the output of the node is changed, that node will now have a different hash and so any previous computation done by this node will be invalid and so the progress is now gone. Any nodes that depend on this node will also change their hash because their file input attribute changed the directory which is based on the hash of the first node. And then that happens to any nodes that depend on those nodes and so on. Since all of the data is still stored in the cache folder under the previous hash, no time is lost if the attribute is changed back to the first value because the new hash will match the first hash.

4.1.4 Files

Every node has associated log(s), status(es) and statistics files. These allow the progress and performance of the process to be monitored.

4.1.5 API

Naming

Use UpperCamelCase for both the class (like normal) and the file. For example 'CameraInit.py' contains 'class CameraInit'

Node Type

```
from meshroom.core import desc


class MyNode(desc.CommandLineNode):
    command = 'myExecutable {allParams}'

# or

class MyNode(desc.Node):
    def processChunk(self, chunk):
        # code for the node to run goes here

    # optional
    def stopProcess(self, chunk):
        # code here runs when the stop button is clicked
```

For `desc.CommandLineNode`, `{allParams}` is the name of the group. All parameters in this group will be added to the command in the form `--name value`. To add only the value to the command, use `{myParameterValue}` for a parameter called `myParameter`.

Class variables

Node

`cpu`, `gpu`, `ram`:

Used for submitters to be efficient in allocating resources.

level	value	description
`desc.Level.NONE`	0	Does not use this resource.
`desc.Level.NORMAL`	1	Uses this resource but not much.
`desc.Level.INTENSIVE`	2	Uses a lot of this resource.

`size`:

size	description
`desc.DynamicNodeSize`	Expresses a dependency to an input attribute to define the size of a Node in terms of individual tasks for parallelization. If the attribute is a link to another node, Node's size will be the same as this connected node. If the attribute is a ListAttribute, Node's size will be the size of this list.
`desc.MultiDynamicNodeSize`	Expresses dependencies to multiple input attributes to define the size of a node in terms of individual tasks for parallelization. Works as DynamicNodeSize and sum the sizes of each dependency.
`desc.StaticNodeSize`	Expresses a static Node size in terms of individual tasks for parallelization.

`parallelization`: `desc.Parallelization(blockSize)` defines a parallel task with a given block size.

`documentation`: Text is displayed in the 'documentation' tab in the GUI.

CommandLineNode

`commandLine`: the command to execute

`commandLineRange`: the command arguments for a range start and end for a parallelized node using `'{rangeStart}'` and `'{rangeBlockSize}'`

Parameters

2 class variables can be defined, `inputs` and `outputs`, both of which are of type list containing parameter objects.

Table 1: General arguments (applies to all attributes)

argument	type	default	description
`name`	string		The command line option or how the parameter will be accessed by `chunk.node.myParameterName.value`.
`label`	string		What it is called in the GUI.
`description`	string		Description shown in the GUI.
`value`	depends		Default value of an input attribute or value of output attribute.
`uid`	list		Controls if the parameter effects the node hash.
`group`	string	`'allParams'`	To control if it is added to the command line.
`advanced`	boolean	`False`	To make it hidden by default in the GUI.
`enabled`	boolean	`True`	Enabled by default but can be disabled if a criteria is met.

Extra arguments:

Table 2: `desc.ListAttribute`

argument	type	default	description
`elementDesc`	attribute description		The attribute description of elements to store in that list.
`joinChar`	string	`' '`	Character to join the attributes for the command line.

Table 3: `desc.GroupAttribute`

argument	type	default	description
`groupDesc`	list (attribute descriptions)		The description of the attributes composing this group.
`joinChar`	string	`' '`	Character to join the attributes for the command line.

Table 4: `desc.IntParam`, `desc.FloatParam`

argument	type	default	description
`range`	tuple (int/float)		(minimum, maximum, step)

Table 5: `desc.ChoiceParam`

argument	type	default	description
`values`	tuple (string)		Available values to choose from.
`exclusive`	boolean		Can it only be one value at once?
`joinChar`	string	`' '`	Character to join the selected attributes for the command line if not exclusive.

The following parameters have no extra arguments: `desc.File`, `desc.BoolParam`, `desc.StringParam`

Logging

For `desc.CommandLineNode` the standard output will be sent to the log file. For `desc.Node` the logging is handled through `chunk.logManager` and `chunk.logger`.

```
class MyNode(desc.Node):
    def processChunk(self, chunk):
        try:
            chunk.logManager.start('debug')

            chunk.logManager.makeProgressBar(100, 'this is a progress bar')
            chunk.logManager.updateProgressBar(50) # progress bar half way

            chunk.logger.debug('this is a debug log')
            chunk.logger.info('this is an info log')
            chunk.logger.warning('this is a warning log')
            raise RuntimeError('this is an error log')
        except Exception as e:
            chunk.logger.error(e)
            raise RuntimeError()
        finally:
            # required to unlock log file so that it can be deleted if required
            chunk.logManager.end()
```

Examples

https://github.com/alicevision/meshroom/blob/develop/meshroom/nodes/aliceVision/Publish.py https://github.com/alicevision/meshroom/blob/develop/meshroom/nodes/aliceVision/SketchfabUpload.py

4.2 Pipelines

Meshroom comes with the following pipelines:

4.2.1 Photogrammetry Pipeline

This is the default pipeline used to reconstruct a 3D model from 2D images.

Nodes

1. *CameraInit*
2. *FeatureExtraction*
3. *ImageMatching*
4. *FeatureMatching*
5. *StructureFromMotion*
6. *PrepareDenseScene*
7. *DepthMap*
8. *DepthMapFilter*
9. *Meshing*
10. *MeshFiltering*
11. *Texturing*

4.2.2 HDR Panorama Pipeline

- fusion of multi-bracketing LDR images into HDR
- alignment of panorama images
- support for fisheye optics
- automatically estimate fisheye circle or manually edit it
- take advantage of motorized-head file

Nodes

1. *CameraInit*
2. *PanoramaPrepareImages*
3. *LdrToHdrSampling*
4. *LdrToHdrCalibration*
5. *LdrToHdrMerge*
6. *FeatureExtraction*
7. *PanoramaInit*
8. *ImageMatching*
9. *FeatureMatching*
10. *PanoramaEstimation*
11. *SfMTransform*
12. *PanoramaWarping*
13. *PanoramaCompositing*
14. *ImageProcessing*

4.3 Supported Formats

4.3.1 Image File formats

Supported file extensions of Images / Image Viewer:

All image formats supported by the OIIO library[27] 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.

4.3.2 Video File formats

.avi’, ‘.mov’, ‘.qt’, ‘.mkv’, ‘.webm’, ‘.mp4’, ‘.mpg’, ‘.mpeg’, ‘.m2v’, ‘.m4v’, ‘.wmv’, ‘.ogv’, ‘.ogg’, ‘.mxf’

4.3.3 Panoramas

panoramaInfoExtensions: ‘.xml’

[27] https://github.com/OpenImageIO/oiio

4.3.4 3D File formats

Name	Reference	Description
Alembic (.abc)	Alembic[28]	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	PLY[29]	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.

4.3.5 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)

4.4 Submitters

Meshroom supports external graph computation through this API. This allows the process to run on a render farm to reduce computation time.

[28] http://www.alembic.io/
[29] https://people.sc.fsu.edu/~jburkardt/data/ply/ply.html

4.4.1 API

```
from meshroom.core.submitter import BaseSubmitter


class MySubmitter(BaseSubmitter):
    def __init__(self, parent=None):
        super(MySubmitter, self).__init__(name='Submitter', parent=parent)

    def submit(self, nodes, edges, filepath):
        # submit task to render farm
```

Table 6: `submit`

argument	type	description
`nodes`	list (meshroom.core.node.Node)	All of the nodes that need to be submitted
`edges`	set (meshroom.core.node.Node: meshroom.core.node.Node)	{A: B} A depends on B
`filepath`	string	Path to .mg file.

CHAPTER

FIVE

COMMAND LINE FEATURES

5.1 meshroom_compute

Execute a graph of processes.

Table 1: arguments

argument	description
graphFile	Filepath to a graph file.
`--node`	Process the node. It will generate an error if the dependencies are not already computed.
`--toNode`	Process the node with its dependencies.
`--forceStatus`	Force computation if status is RUNNING or SUBMITTED.
`--forceCompute`	Compute in all cases even if already computed.
`--extern`	Use this option when you compute externally after submission to a render farm from meshroom.
`--cache`	Custom cache folder to write computation results. If not set, the default cache folder will be used.
`-i`, `-iteration`	Index of a chunk to compute.

5.2 meshroom_photogrammetry

Launch the full photogrammetry or panorama HDR pipeline.

Table 2: arguments

argument	description
`-i`, `--input`	Input folder containing images or folders of images or file (.sfm or .json) with images paths and optionally predefined camera intrinsics.
`-I`, `--inputRecursive`	Input folders containing all images recursively.
`-p`, `--pipeline`	"photogrammetry" pipeline, "panotamaHdr" pipeline, "panotamaFisheyeHdr" pipeline or a Meshroom file containing a custom pipeline to run on input images. Requirements: the graph must contain one CameraInit node, and one Publish node if –output is set.
`--overrides`	Path to a JSON file containing the graph parameters overrides.
`--paramOverrides`	Override specific parameters directly from the command line (by node type or by node names).
`-o`, `--output`	Output folder where results should be copied to. If not set, results will have to be retrieved directly from the cache folder.
`--cache`	Custom cache folder to write computation results. If not set, the default cache folder will be used.
`--save`	Save the configured Meshroom graph to a project file. It will setup the cache folder accordingly if not explicitly changed by –cache.
`--compute`	You can set it to <no/false/0> to disable the computation.
`--scale`	Downscale factor override for DepthMap estimation. By default (-1): use pipeline default value.
`--toNode`	Process the node(s) with its dependencies.
`--forceStatus`	Force computation if status is RUNNING or SUBMITTED.
`--forceCompute`	Compute in all cases even if already computed.
`--submit`	Submit on renderfarm instead of local computation.
`--submitter`	Execute job with a specific submitter.

5.3 meshroom_submit

not included in binary release

Submit a Graph of processes on renderfarm.

Table 3: arguments

argument	description
`meshroomFile`	Filepath to a graph file.
`--toNode`	Process the node(s) with its dependencies.
`--submitter`	Execute job with a specific submitter.

5.4 meshroom_status

not included in binary release

Query the status of nodes in a Graph of processes.

Table 4: arguments

argument	description
graphFile	Filepath to a graph file.
`--node`	Process the node alone.
`--toNode`	Process the node and all previous nodes needed.
`--verbose`	Print full status information.

5.5 meshroom_statistics

not included in binary release

Query the status of nodes in a Graph of processes.

Table 5: arguments

argument	description
graphFile	Filepath to a graph file.
`--node`	Process the node alone.
`--graph`	Process the node and all previous nodes needed.
`--exportHtml`	Filepath to the output html file.
`--verbose`	Print full status information.

5.6 meshroom_newNodeType

not included in binary release

Create a new Node Type

Table 6: arguments

argument	description
node	New node name.
bin	Output plugin folder.
`--output`	Output plugin folder.
`--parser`	Select the parser adapted for your command line: {boost,cmdLineLib,basic}.
`--force`	Allows to overwrite the output plugin file.

CHAPTER

SIX

NODE REFERENCE

List with all the nodes

Note: Some *experimental/debug/advanced* parameters are only visible with "Advanced Attributes" enabled. To enable "Advanced Attributes", click on the three vertical dots in the upper right corner of the node settings and activate the check box. Some features, settings and nodes are only available in the latest build or developer version.

6.1 CameraCalibration

Description

Note: At the moment this node can not directly be connected to the SfM pipeline in the UI. That would be obviously a nice feature to have. The camera models and parameters can be manually copied to the CameraInit settings. This node just needs a bit more work before using it directly into the Meshroom graph. If someone is interested to contribute to this feature, we would be glad to provide assistance.

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

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 (**7**) Height (**5**) (0-10000)
Square Size	Size of the grid's square cells (0-100mm) (**1**)
Nb Distortion Coef	Number of distortion coefficient (0-5) (**3**)
Max Frames	Maximal number of frames to extract from the video file (0-5) (**0**)
Max Calib Frames	Maximal number of frames to use to calibrate from the selected frames (0-1000)
Calib Grid Size	Define the number of cells per edge (0-50)
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)

Details

Patterns

CHESSBOARD https://github.com/artoolkit/artoolkit5/blob/master/doc/patterns/Calibration%20chessboard%20(A4).pdf

Chessboard calibration video sample https://vimeo.com/141414129

CIRCLES

ASYMMETRIC_CIRCLES https://nerian.com/support/resources/patterns/

ASYMMETRIC_CCTAG https://github.com/alicevision/CCTag

A list with other camera calibration tools and patterns can be found here https://github.com/natowi/CameraCalibTools

6.2 CameraDownscale

Description

Downscale images. Default is 0.5 (half size)

Input	SfM Data File
RescaleFactor	Newsize = rescalefactor * oldsize', (0.0-1.0, **0.5**)
Verbose Level	verbosity level (fatal, error, warning, info, debug, trace)
Output SfMData File	Path to the output sfmdata file (sfmData.abc)

6.3 CameraInit

Description

CameraInit loads image metadata, sensor information and generates viewpoints.sfm and cameraInit.sfm. 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 two 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.

Viewpoints Input	viewpoints (1 Element for each loaded image) - 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: pinhole', '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

Details

The UID is based on the metadata. If there is no metadata it falls back to images file paths.

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

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

6.4 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

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	Describer 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	[voctree] Number of images to retrieve in database (1 - 1000)
Max Results	[voctree] For algorithm AllResults, it stops the image matching when this number of matched images is reached. If 0 it is ignored (1 - 100)
Commonviews	[voctree] Number of minimum images in which a point must be seen to be used in cluster tracking (2 - 50)
Voctree	[voctree] Filename for the vocabulary tree
Voctree Weights	[voctree] Filename for the vocabulary tree weights
Algorithm	[voctree] Algorithm type: (FirstBest, AllResults)
Matching Error	[voctree] 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	[voctree] Number of previous frame of the sequence to use for matching (0 = Disable) (0 - 100)
Robust Matching	[voctree] 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 –refineIntrinsics 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

6.5 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"1 "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 & Barrera, 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://alicevision.github.io/#photogrammetry/localization

References

[KSS11]

[Kneip2013] Using Multi-Camera Systems in Robotics: Efficient Solutions to the NPnP ProblemL. 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

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. –mediapath /path/to/cam1.mov /path/to/cam2.mov)
Camera Intrinsics	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 Describer Types	The describer 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
Matching Estimator	The type of /sac framework to use for matching (acransac, loransac)
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 - 10)
Max Input Frames	Maximum number of frames to read in input. 0 means no limit (0 - 1000)
Voctree	[voctree] Filename for the vocabulary tree
Voctree Weights	[voctree] Filename for the vocabulary tree weights
Algorithm	[voctree] Algorithm type: {FirstBest, AllResults}
Nb Image Match	[voctree] Number of images to retrieve in the database (0 - 50)
Max Results	[voctree] For algorithm AllResults, it stops the image matching when this number of matched images is reached. If 0 it is ignored (0 - 100)
Matching Error	[voctree] 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 - 10)
N Nearest Key Frames	[cctag] Number of images to retrieve in database (0 - 50)
Output File	The name of the file where to store the calibration data (desc.Node.internalFolder + 'cameraRigCalibration.rigCal)

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**.

6.6 CameraRigLocalization

Description

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

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. –mediapath /path/to/cam1.mov /path/to/cam2.mov)
Rig Calibration File	The file containing the calibration data for the rig (subposes)
Camera Intrinsics	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 Describer Types	The describer 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)
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 - 10)
Use Localize Rig Naive	Enable/Disable the naive method for rig localization: naive method tries to localize each camera separately
Angular Threshold	The maximum angular threshold in degrees between feature bearing vector and 3D point direction. Used only with the opengv method (0 - 10)
Voctree	[voctree] Filename for the vocabulary tree
Voctree Weights	[voctree] Filename for the vocabulary tree weights
Algorithm	[voctree] Algorithm type: {FirstBest, AllResults}
Nb Image Match	[voctree] Number of images to retrieve in the database
Max Results	[voctree] For algorithm AllResults, it stops the image matching when this number of matched images is reached. If 0 it is ignored (0 - 100)
Matching Error	[voctree] 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 - 10)
N Nearest Key Frames	[cctag] Number of images to retrieve in database (0 - 50)
Output Alembic	Filename for the SfMData export file (where camera poses will be stored) desc.Node.internalFolder + 'trackedcameras.abc

6.7 ConvertSfMFormat

Description

- creates abc', 'sfm', 'json', 'ply', 'baf SfM File from SfMData file including the selected Describer Types

This node can be used to convert the sparse point cloud sfm.abc from StructureFromMotion node or the dense point cloud densePointCloud.abc from the Meshing node. To convert pointclouds to abc, sfm, json, ply, baf, *disable SIFT* and enable the *unknown* Describer Type.

settings

Name	Description
Input	`SfMData file`
SfM File Format	SfM File Format (output file extension: abc', 'sfm', 'json', 'ply', 'baf)``
Describer Types	Describer types to keep.`'sift', 'sift_float', 'sift_upright', 'akaze', 'akaze_liop', 'akaze_mldb', 'cctag3', 'cctag4', 'sift_ocv', 'akaze_ocv', 'unknown'`
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})
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 - 10)
Use Localize Rig Naive	Enable/Disable the naive method for rig localization: naive method tries to localize each camera separately
Angular Threshold	The maximum angular threshold in degrees between feature bearing vector and 3D point direction. Used only with the opengv method (0 - 10)
Voctree	[voctree] Filename for the vocabulary tree
Voctree Weights	[voctree] Filename for the vocabulary tree weights
Algorithm	[voctree] Algorithm type: {FirstBest, AllResults}``
Nb Image Match	[voctree] Number of images to retrieve in the database
Max Results	[voctree] For algorithm AllResults, it stops the image matching when this number of matched images is reached. If 0 it is ignored (0 - 100)
Matching Error	[voctree] 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 - 10)
N Nearest Key Frames	[cctag] Number of images to retrieve in database (0 - 50)
Output Alembic	Filename for the SfMData export file (where camera poses will be stored) desc.Node.internalFolder + 'trackedcameras.abc

Input nodes: StructureFromMotion:output:math:`Rightarrow` input:ConvertSfMFormat

Can I convert between Openmvg and alicevision SfM formats?

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

6.8 DepthMap

Note: This node requires CUDA

Description

Retrieves the depth value of each pixel for all cameras that have been resolved by SfM.

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)
Max View Angle	Maximum angle between two views. (10.0 - 120.0)
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

Detailed description

For all cameras that have been resolved by SfM, we want to retrieve the depth value of each pixel. Many approaches exist, like Block Matching, Semi-Global Matching (SGM) [Hirschmüller2005], [Hirschmüller2008] or ADCensus [Xing2011]. We will focus on the SGM method implemented in AliceVision.

For each image, we select the N best/closest cameras around. We select fronto-parallel planes based on the intersection of the optical axis with the pixels of the selected neighboring cameras. This creates a volume W, H, Z with many depth candidates per pixel. We estimate the similarity for all of them. The similarity is computed by the Zero Mean Normalized Cross-Correlation (ZNCC) of a small patch in the main image

reprojected into the other camera. This create a volume of similarities. For each neighboring image, we accumulate similarities into this volume. This volume is very noisy. We apply a filtering step along X and Y axes which accumulates local costs which drastically reduce the score of isolated high values. We finally select the local minima and replace the selected plane index with the depth value stored into a depth map. This depth map has banding artifacts as it is based on the original selection of depth values. So a refine step is applied to get depth values with sub-pixel accuracy.

All these depth maps can be computed independently in parallel. Then we apply a filtering step to ensure consistency between multiple cameras. A compromise is chosen based on both similarity value and the number of coherent cameras to keep weakly supported surfaces without adding artefacts.

[Hirschmüller2005]	Accurate and efficient stereo processing by semi-global matching and mutual information, H. Hirschmüller. CVPR 2005
[Hirschmüller2008]	Stereo processing by semiglobal matching and mutual information, H. Hirschmüller, 2008
[Strecha2006]	Combined depth and outlier estimation in multi-view stereo, C. Strecha, R. Fransens, and L. Van Gool, CVPR 2006
[Scharstein2002]	A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, D. Scharstein and R. Szeliski, 2002
[Xing2011]	On building an accurate stereo matching system on graphics hardware. Xing, M., Xun, S., Mingcai Z., Shaohui J., Haitao, W., Xiaopeng Z., 2011

6.9 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.

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, error, warning, info, debug, 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

6.10 ExportAnimatedCamera

Description

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

The Animated Camera export feature is not optimized at the moment and requires a sequence of images with corresponding names (1-n) from the same folder. Unstructured images, naming conventions, folder structures... will not work or result in an error.

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, error, warning, info, debug, trace)
Output filepath	Output filepath for the alembic animated camera
Output Camera Filepath	Output filename for the alembic animated camera internalFolder + 'camera.abc'

SFM to ExportAnimatedCamera Details: https://www.youtube.com/watch?v=1dhdEmGLZhY

6.11 ExportColoredPointCloud

Description

Allows the export of the colored point cloud.

Name	Description
Input SfMData	SfMData file containing a complete SfM.
Verbose Level	Verbosity level (fatal, error, warning, info, debug, trace).
Output Point Cloud Filepath	Output point cloud with visibilities as SfMData file. {cache}/{nodeType}/{uid0}/pointCloud.abc

6.12 ExportMatches

Description

Saves features and descriptors files (.feat, .desc) to folder

settings

Name	Description
Input	SfMData file
Describer Types	Describer types used to describe an image. [**'sift'**, 'sift_float', 'sift_upright', 'akaze', 'akaze_liop', 'akaze_mldb', 'cctag3', 'cctag4', 'sift_ocv', 'akaze_ocv'],
Features Folder	
Features Folders	Folder(s) containing the extracted features and descriptors.
Matches Folder	
Matches Folders	Folder(s) in which computed matches are stored.
Verbose Level	['fatal', 'error', 'warning', 'info', 'debug', 'trace']
Output	Output path for the features and descriptors files (.feat, .desc). (internalFolder)

6.13 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.

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

6.14 FeatureExtraction

Description

This step extracts features from the images, as well as descriptors for those features. It will change the file extension based on what type of feature you are extracting.

Name	Description
Input	SfMData file.
Describer Types	Describer types used to describe an image. **'sift'**, 'sift*float', 'sift*upright', 'akaze', 'akaze*liop', 'akaze*mldb', 'cctag3', 'cctag4', 'sift*ocv', 'akaze*ocv'
Describer Preset	Control the ImageDescriber configuration (low, medium, **normal**, high, ultra). Configuration "ultra" can take long time !
Force CPU Extraction	Use only CPU feature extraction.
Max Nb Threads	Specifies the maximum number of threads to run simultaneously (0 for automatic mode). (0-24) **0**
Verbose Level	verbosity level (fatal, error, warning, info, debug, trace).
Output Folder	Output path for the features and descriptors files (*.feat, *.desc).

Increase number of features

- add more describer types (akaze)
- increase describer preset

Force CPU Extraction

Experimental feature. When disabled, GPU will be used. Speeds up computation. Requires CUDA CC3+.

Detailed description

The objective of this step is to extract distinctive groups of pixels that are, to some extent, invariant to changing camera viewpoints during image acquisition. Hence, a feature in the scene should have similar feature descriptions in all images.

The most well-know feature detection method is the SIFT (Scale-invariant feature transform) algorithm. The initial goal of SIFT is to extract discriminative patches in a first image that can be compared to discriminative patches of a second image irrespective of rotation, translation, and scale. As a relevant detail only exists at a certain scale, the extracted patches are centered at stable points of interest. The key idea is that, to some extent, one can use the SIFT invariance to deal with the image transformations occurring when the viewpoints are changing during image acquisition.

From the representation of one image at different scales, which is technically done by computing a pyramid of downscaled images. SIFT computes scale-space maxima of the Laplacian representation, which is a specific image energy-based representation of the image, using so-called differences of Gaussians. These maxima correspond to points of interest. It then samples for each one of these maxima a square image patch whose origin is the maximum and x-direction is the dominant gradient at the origin. For each keypoint, a description of these patches is associated.

The description, which is typically stored in 128 bits, consists of a statistics of gradients computed in regions around the keypoint. The region size is determined by the keypoint scale and the orientation is determined by the dominant axis.

As the number of extracted features may vary a lot due to the variability of textures complexity (from one image to another or in different parts of the image), a post-filtering step is used to control the number of

extracted features to reasonable limits (for instance between one and ten thousands per image). We use a grid filtering to ensure a good repartition in the image.

[Lowe2004]	Distinctive image features from scale-invariant keypoints, David G. Lowe, 2004[30]
[Otero2014]	Anatomy of the SIFT Method, Ives Rey Otero, Mauricio Delbracio, 2014[31]
[Yu2011]	ASIFT: An Algorithm for Fully Affine Invariant Comparison, Guoshen Yu, Jean-Michel Morel, 2011[32]
[Alcantarilla2013]	AKAZE Fast explicit diffusion for accelerated features in nonlinear scale spaces, P.F. Alcantarilla, J. Nuevo, A. Bartoli, 2013[33]
[Li2015]	A survey of recent advances in visual feature detection, Yali Li, Shengjin Wang, Qi Tian, Xiaoqing Ding, 2015[34]
[VLFEAT2008]	VLFeat: An Open and Portable Library of Computer Vision Algorithms A. Vedaldi and B. Fulkerson, 2008[35] VLFeat SIFT detailed presentation[36]

6.15 FeatureMatching

Description

Finds the correspondence between the images, using feature descriptors.

settings

[30] http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf
[31] http://www.ipol.im/pub/art/2014/82/
[32] http://www.ipol.im/pub/art/2011/my-asift/
[33] http://www.bmva.org/bmvc/2013/Papers/paper0013/paper0013.pdf
[34] https://www.researchgate.net/profile/Yali_Li3/publication/273841042_A_survey_of_recent_advances_in_visual_feature_detection/links/5707d38408ae2eb9421bda3e.pdf
[35] http://www.vlfeat.org/
[36] http://www.vlfeat.org/overview/sift.html

Name	Description
Input	SfMData file
Features Folder	
Features Folders	Folder(s) containing the extracted features and descriptors
Image Pairs List	Path to a file which contains the list of image pairs to match
Describer Types	Describer types used to describe an image **sift**'/ 'sift_float'/ 'sift_upright'/ 'akaze'/ 'akaze_liop'/ 'akaze_mldb'/ 'cctag3'/ 'cctag4'/ 'sift_ocv'/ 'akaze_ocv
Photometric Matching 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'
Geometric Estimator	Geometric estimator: (acransac: A-Contrario Ransac // loransac: LO-Ransac (only available for fundamental_matrix model)
Geometric Filter Type	Geometric validation method to filter features matches: **fundamental_matrix** // essential_matrix // homography_matrix /// homography_growing // no_filtering'
Distance Ratio	Distance ratio to discard non meaningful matches 0.8 (0.0 - 1)
Max Iteration	Maximum number of iterations allowed in ransac step 2048 (1 - 20000)
Max Matches	Maximum number of matches to keep (0 - 10000)
Save Putative Matches	putative matches (True/False)
Guided Matching	the found model to improve the pairwise correspondences (True/False)
Export Debug Files	debug files (svg/ dot) (True/False)
Verbose Level	verbosity level (fatal, error, warning, info, debug, trace)
Output Folder	Path to a folder in which computed matches will be stored

Detailed description

The objective of this step is to match all features between candidate image pairs.

First, we perform photometric matches between the set of descriptors from the 2 input images. For each feature in image A, we obtain a list of candidate features in image B. As the descriptor space is not a linear and well defined space, we cannot rely on absolute distance values to know if the match is valid or not (we can only have an absolute higher bound distance). To remove bad candidates, we assume that there's only one valid match in the other image. So for each feature descriptor on the first image, we look for the 2 closest descriptors and we use a relative threshold between them. This assumption will kill features on repetitive structure but has proved to be a robust criterion [Lowe2004]. This provide a list of feature matching candidates based only on a photometric criterion. Find the 2 closest descriptors in the second image for each feature is computationally intensive with a brute force approach, but many optimized algorithms exists. The most common one is Approximate Nearest Neighbor, but there are alternatives like, Cascading Hashing.

Then, we use the features positions in the images to make a geometric filtering by using epipolar geometry in an outlier detection framework called RANSAC (RANdom SAmple Consensus). We randomly select a small set of feature correspondences and compute the fundamental (or essential) matrix, then we check the number of features that validates this model and iterate through the RANSAC framework.

[Lowe2004]	Distinctive image features from scale-invariant keypoints, David G. Lowe, 2004[37]
[FLANN2009]	Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. Muja, Marius, and David G. Lowe. VISAPP (1). 2009

6.16 GlobalSfM

Description

GlobalSfM

MR version: 2020.x

settings

[37] http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

Name	Description
Input	SfM Data File
Features Folder	
Features Folders	Folder(s) containing the extracted features.
Matches Folder	
Matches Folders	Folder(s) in which computed matches are stored.
Describer Types	Describer types used to describe an image. [**'sift'**, 'sift_float', 'sift_upright', 'akaze', 'akaze_liop', 'akaze_mldb', 'cctag3', 'cctag4', 'sift_ocv', 'akaze_ocv']
Rotation Averaging Method	Method for rotation averaging : • L1 minimization • **L2 minimization**
Translation Averaging Method	Method for translation averaging : • L1 minimizationn" • L2 minimization of sum of squared Chordal distancesn" • **L1 soft minimization**
Force Lock of All Intrinsic Camera Parameters.	Force to keep constant all the intrinsics parameters of the cameras (focal length, principal point, distortion if any) during the reconstruction. This may be helpful if the input cameras are already fully calibrated.
Verbose Level	verbosity level (critical, error, warning, info, debug).
Output Folder	internalFolder
Output SfMData File	Path to the output sfmdata file (internalFolder + 'SfmData.abc')

[Moulon2013] Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion. Pierre Moulon, Pascal Monasse and Renaud Marlet. ICCV 2013

GlobalSFM vs SequentialSFM https://github.com/openMVG/openMVG/issues/1037

6.17 HDRIstitching

Description

hdri panorama stitching

MR version: 2020.x

settings

Name	Description
Input Files	Input File/Folder
Input Folder	List of fisheye images or folder containing them.
Blur Width	Blur width of alpha channel for all fisheye (between 0 and 1). "Determine the transitions sharpness. (0-1, **0.2**)
Image X Rotation	Image X Rotation (-20-20, **0**)
X Rotations	Rotations in degree on axis X (horizontal axis) for each image.
Image Y Rotation	Image Y Rotation (-30-30, **0**)
Y Rotations	Rotations in degree on axis Y (vertical axis) for each image.
Image Z Rotation	Image Z Rotation (-10-10, **0**)
Z Rotations	Rotations in degree on axis Z (depth axis) for each image.
Verbose Level	verbosity level (critical, error, warning, info, debug).
Output Panorama	Output folder for panorama (internalFolder)

6.18 ImageMatching

Description

This is a preprocessing step which figures out which images make sense to match to each other.

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 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, warning, info, debug, trace)
Output List File	Filepath to the output file with the list of selected image pairs

Detailed descriptioin

The objective of this part is to find images that are looking to the same areas of the scene. For that, we use the image retrieval techniques to find images that share some content without the cost of resolving all feature matches in details. The ambition is to simplify the image in a compact image descriptor which allows to compute the distance between all images descriptors efficiently.

One of the most common method to generate this image descriptor is the vocabulary tree approach. By passing all extracted features descriptors into it, it makes a classification by comparing their descriptors to the ones on each node of this tree. Each feature descriptor ends up in one leaf, which can be stored by a simple index: the index of this leaf in the tree. The image descriptor is then represented by this collection of used leaves indices.

It is now possible to see if different images share the same content by comparing these image descriptors.

[Nister2006] Scalable Recognition with a Vocabulary Tree, David Nister and Henrik Stewenius, CVPR 2006

6.19 ImageMatchingMultiSfM

Description

This node can combine image matching between two input SfMData.

Used for live-reconstruction and augment-reconstruction.

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

6.20 ImageProcessing

Description

Basic node for image processing. It replaces the cameraDownscale node.

- Convert image files into another file format
- Downscale images
- Apply exposure compensation, constrast, median filter, sharpen

Input	SfM Data File
File Extension	[‘’, ‘exr’, ‘jpg’, ‘tiff’, ‘png’]
Only Reconstructed Views	Process Only Reconstructed Views
Exposure Compensation	True/**False**
Downscale	Downscale (0.0 - **1.0**)
Contrast	(0.0 - 100.0) **1.0**
Median Filter	(**0** - 10)
Sharpen Width	(**1** - 9)
Sharpen Contrast	(0.0 - 100.0) **1.0**
Sharpen Threshold	(**0.0** - 1.0)
Verbose Level	verbosity level (fatal, error, warning, info, debug, trace)
Output SfMData File	Path to the output sfmdata file (sfmData.abc)

6.21 KeyframeSelection

Description

This is a node for keyframe selection from video, which removes too similar or too blurry images.

Note: At the moment, KeyframeSelection can not be used as input for CameraInit. We cannot know in advance how many images will be extracted, but the total number of images is required for render farm submisson. So when adding a video file to Meshroom, the following Message will appear: “You need to manually compute the KeyframeSelection node and then reimport the created images into Meshroom for the reconstruction.”

To automatically add extracted frames to your project you can copy the output folder path of KeyframeExtraction and set it as the Live Reconstruction Image Folder Path. Then start watching the folder and execute the graph. `https://github.com/alicevision/meshroom/issues/232`

Supported file types:

.avi, .mov, .qt, .mkv, .webm, .mp4, .mpg, .mpeg, .m2v, .m4v, .wmv, .ogv, .ogg, .mxf

FFmpeg can be used for video file conversion.

MR version 2020.x

settings

Name	Description
Media Path	Media Path
Media Paths	Input video files or image sequence directories.
Brand	Camera brand.
Brands	Camera brands.
Model	Camera model.
Models	Camera models.
mmFocal	Focal in mm (will be use if not 0). (**0.0**-500)
mmFocals	Focals in mm (will be use if not 0).
pxFocal	Focal in px (will be use and convert in mm if not 0). (**0.0**-500)
pxFocals	Focals in px (will be use and convert in mm if not 0).
Frame Offset	Frame Offset **0**-100
Frame Offsets	Frame Offsets
Sensor Db Path	Camera sensor width database path. (ALICEVISION_SENSOR_DB)
Voctree Path	Vocabulary tree path. (ALICEVISION_VOCTREE)
Use Sparse Distance Selection	Use sparseDistance selection in order to avoid similar keyframes. (**True**)
Use Sharpness Selection	Use frame sharpness score for keyframe selection. (**True**)
Sparse Distance Max Score	Maximum number of strong common points between two keyframes. (1-200, **100**)
Sharpness Preset	Preset for sharpnessSelection : {ultra, high, **normal**, low, very_low, none}
Sharp Subset	sharp part of the image (1 = all, 2 = size/2, ...) (1-100, **4**)
Min Frame Step	minimum number of frames between two keyframes (1-100, **12**)
Max Frame Step	maximum number of frames after which a keyframe can be taken (2-1000, **36**)
Max Nb Out Frame	maximum number of output frames (0 = no limit) (**0**-10000)
Verbose Level	['fatal', 'error', 'warning', 'info', 'debug', 'trace']
Output Folder	Output keyframes folder for extracted frames. (internalFolder)

6.22 LDRToHDR

Description

Name	Description
Input	List of LDR images or a folder containing them
Calibration Method	Method used for camera calibration. - **linear** - robertson - debevec - beta: grossberg
Input Response	external camera response file path to fuse all LDR images together.
Target Exposure Image	LDR image at the target exposure for the output HDR image to be centered.
Calibration Weight	Weight function type (default, gaussian, triangle, plateau). ['default', 'gaussian', 'triangle', 'plateau']
Fusion Weight	Weight function used to fuse all LDR images together (**gaussian**, triangle, plateau).
Oversaturated Correction	Oversaturated correction for pixels oversaturated in all images: - use 0 for no correction - use 0.5 for interior lighting - use 1 for outdoor lighting (0-1)
Recover Path	Path to write recovered LDR image at the target exposure by applying inverse response on HDR image.
Verbose Level	Verbosity level (fatal, error, warning, **info**, debug, trace).
Output	Output HDR image path. desc.Node.internalFolder + 'hdr.exr'
Output Response	Output response function path. desc.Node.internalFolder + 'response.ods'

6.23 LdrToHdrCalibration

Description

Calibrate LDR to HDR response curve from samples

settings

Name	Description
Input	SfM Data File
Samples folder	Samples folder
Bypass	Bypass HDR creation and use the medium bracket as the source for the next steps
Calibration Method	Method used for camera calibration
Calibration Weight	Weight function used to calibrate camera response
Number of Brackets	Number of exposure brackets per HDR image (0 for automatic detection).
Automatic Nb Brackets	Number of exposure brackets used per HDR image. It is detected automatically from input Viewpoints metadata if "userNbBrackets" is 0, else it is equal to "userNbBrackets".
Channel Quantization Power	Quantization level like 8 bits or 10 bits.
Max Number of Points	Max number of points used from the sampling. This ensures that the number of pixels values extracted by the sampling can be managed by the calibration step (in term of computation time and memory usage).
Verbose Level	['fatal', 'error', 'warning', 'info', 'debug', 'trace']
Output response File	Path to the output response file

6.24 LdrToHdrMerge

Description

Calibrate LDR to HDR response curve from samples

settings

Name	Description
Input	SfM Data File
Response file	Response file
Number of Brackets	Number of exposure brackets per HDR image (0 for automatic detection).
Automatic Nb Brackets	Number of exposure brackets used per HDR image. It is detected automatically from input Viewpoints metadata if "userNbBrackets" is 0, else it is equal to "userNbBrackets".
Offset Ref Bracket Index	Zero to use the center bracket. +N to use a more exposed bracket or -N to use a less exposed backet.
Bypass	Bypass HDR creation and use the medium bracket as the source for the next steps
Fusion Weight	Weight function used to fuse all LDR images together
Channel Quantization Power	Quantization level like 8 bits or 10 bits.
Highlights Correction	Pixels saturated in all input images have a partial information about their real luminance. We only know that the value should be >= to the standard hdr fusion. This parameter allows to perform a post-processing step to put saturated pixels to a constant value defined by the *highlightsMaxLuminance* parameter. This parameter is float to enable to weight this correction.
Highlight Target Luminance (Lux)	This is an arbitrary target value (in Lux) used to replace the unknown luminance value of the saturated pixels.
Storage Data Type	Storage image data type
Verbose Level	['fatal', 'error', 'warning', 'info', 'debug', 'trace']
Output SfMData File	Path to the output sfmdata file

6.25 LdrToHdrSampling

Description

Sample pixels from Low range images for HDR creation

settings

Name	Description
Input	SfM Data File
Number of Brackets	Number of exposure brackets per HDR image (0 for automatic detection).
Automatic Nb Brackets	Number of exposure brackets used per HDR image. It is detected automatically from input Viewpoints metadata if "userNbBrackets" is 0, else it is equal to "userNbBrackets".
Bypass	Bypass HDR creation and use the medium bracket as the source for the next steps
Channel Quantization Power	Quantization level like 8 bits or 10 bits.
Block Size	Size of the image tile to extract a sample.
Patch Radius	Radius of the patch used to analyze the sample statistics.
Max Number of Samples	Max number of samples per image group.
Export Debug Files	Export debug files to analyze the sampling strategy.
Verbose Level	['fatal', 'error', 'warning', 'info', 'debug', 'trace']
Output	Output path for the samples.

6.26 MeshDecimate

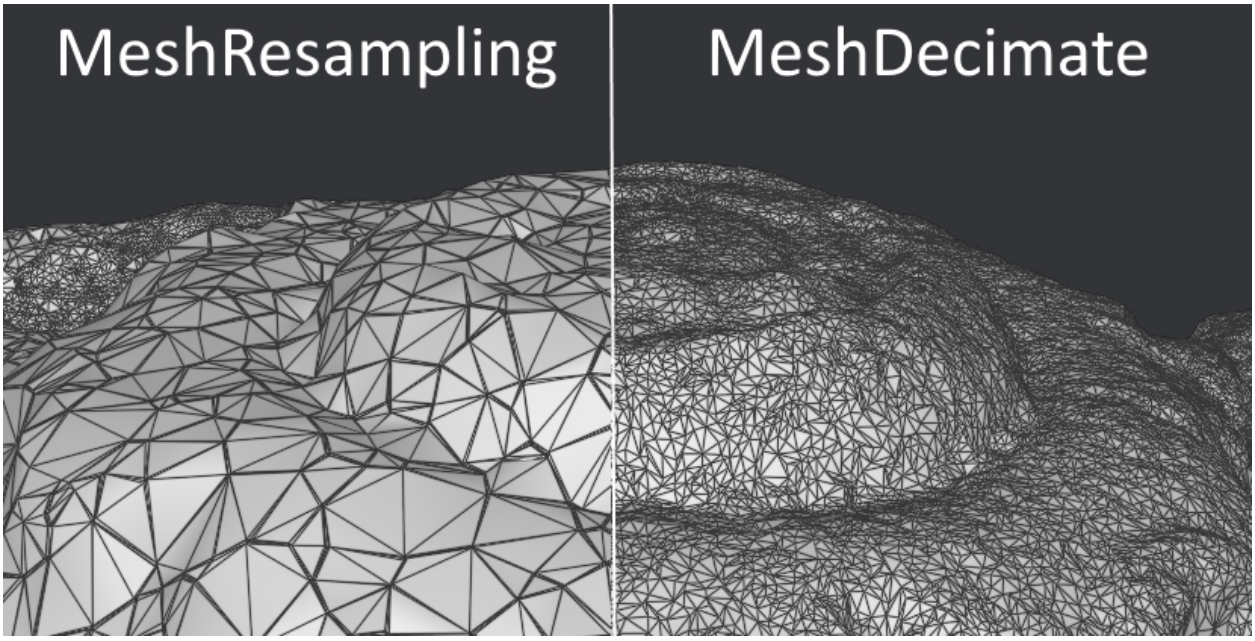
Description

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

settings

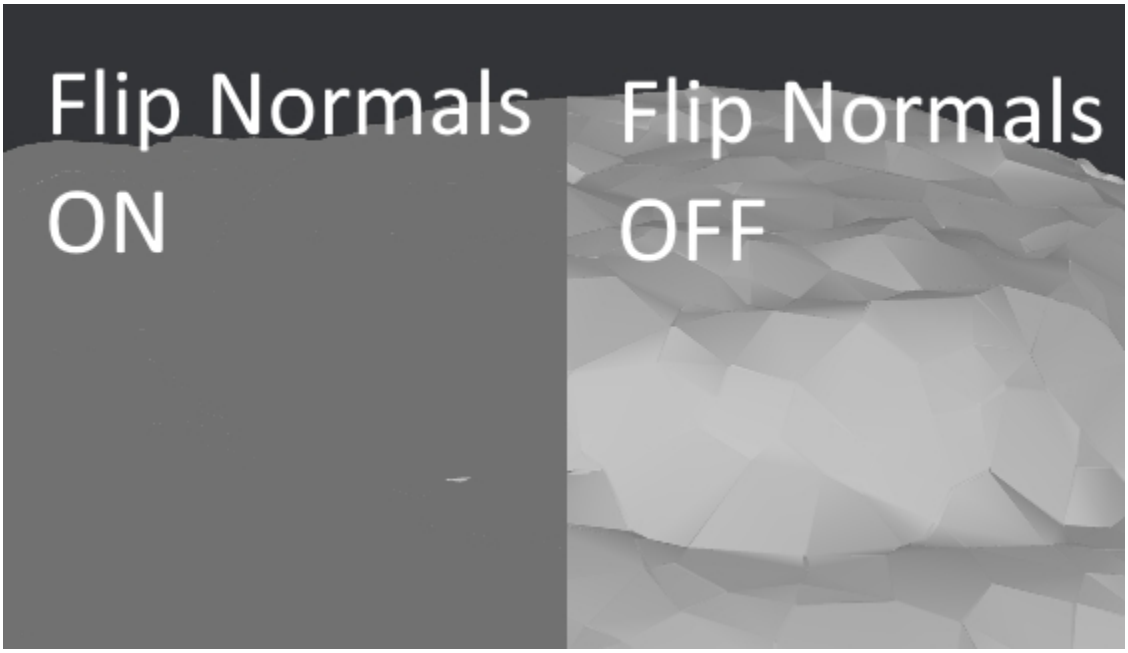
Name	Description
Input Mesh (OBJ file format)	
Simplification factor	Simplification factor 0.5 (0 - 1)
Fixed Number of Vertice	Fixed number of output vertices 0 (0 - 1 000 000)
Min Vertices	Min number of output vertices 0 (0 - 1 000 000)
Max Vertices	Max number of output vertices 0 (0 - 1 000 000)
Flip 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

Comparison MeshDecimate and MeshResampling



MeshDecimate kills vertices to reduce the density, so the vertices at the end already exist in the original mesh. MeshResampling will recreate vertices on the surface with a uniform density, so there is no common vertice with the original mesh.

Flip Normals



6.27 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`

settings

Name	Description
input	Input Mesh (OBJ file format)
Denoising Iterations	Number of denoising iterations (0 - 30)
Mesh Update Closeness Weight	Closeness weight for mesh update, must be positive(0 - 0.1) (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) (1.5)
	Gaussian standard deviation for guidance weight (0.0-10) (1.5)
	Gaussian standard deviation for signal weight. (0.0-5) (0.3)
Mesh Update Mesh	Mesh Update Method * ITERATIVE*UPDATE (default): ShapeUp styled iterative solver * POISSON*UPDATE: Poisson-based update from [Wang 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`

Wang et al. `https://dl.acm.org/citation.cfm?id=2818068`

Detailed Description

A larger value of Lambda or Eta leads to a smoother filtering result.

From: “Static/Dynamic Filtering for Mesh Geometry” by Zhang Et al. https://arxiv.org/pdf/1712.03574.pdf

6.28 MeshFiltering

Description

Filter out unwanted elements of your mesh

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).

6.29 Meshing

Description

Generate Mesh from SfM point cloud or DepthMap

Name	Description
Input	SfMData file.
Depth Maps Folder	Input depth maps folder
Filtered Depth Maps Folder	Input filtered depth maps folder
Estimate Space From SfM	Estimate the 3d space from the SfM
Min Observations For SfM Space Estimation	Minimum number of observations for SfM space estimation. (0-100) **3**
Min Observations Angle For SfM Space Estimation	Minimum angle between two observations for SfM space estimation. (0-120) **10**
Max Input Points	Max input points loaded from depth map images (500**000** - 500**000**000)
Max Points	Max points at the end of the depth maps fusion (100**000** - 10**000**000)
Max Points Per Voxel	(500**000** – 30**000**000)
Min Step	The step used to load depth values from depth maps is computed from maxInputPts. Here we define the minimal value for this step, so on small datasets we will not spend too much time at the beginning loading all depth values (1- 20) **2**
Partitioning	(**singleBlock**, auto)
Repartition	(**multiResolution**, regularGrid)
angleFactor	(0.0-200.0) **15.0**
simFactor	(0.0-200.0) **1.0**
pixSizeMarginInitCoef	(0.0-10.0) **2.0**
pixSizeMarginFinalCoef	(0.0-10.0) **4.0**
voteMarginFactor	(0.1-10.0) **4.0**
contributeMarginFactor	(0.0-10.0) **2.0**
simGaussianSizeInit	(0.0-50) **10.0**
simGaussianSize	(0.0-50) **0.1**
minAngleThreshold	(0.0-10.0) **0.01**
Refine Fuse	Refine depth map fusion with the new pixels size defined by angle and similarity scores.
Add Landmarks To The Dense Point Cloud	Add SfM Landmarks to the dense point cloud.
Colorize Output	Whether to colorize output dense point cloud and mesh.
Save Raw Dense Point Cloud	Save dense point cloud before cut and filtering.
Verbose Level	verbosity level (fatal, error, warning, **info**, debug, trace).
Output Mesh	Output mesh (OBJ file format). mesh.obj
Output Dense Point Cloud	Output dense point cloud with visibilities (SfMData file format). densePointCloud.abc

Draft meshing

A meshing node without a connection to the depth maps folder attribute will create a mesh based on the *structure from motion* point cloud. This is much faster than using depth maps but the result is low quality.

Detailed description

The objective of this step is to create a dense geometric surface representation of the scene.

First, we fuse all the depth maps into a global octree where compatible depth values are merged into the octree cells.

We then perform a 3D Delaunay tetrahedralization. Then a complex voting procedure is done to compute weights on cells and weights on facets connecting the cells as explained in [Jancosek2011] and [Jancosek2014].

A Graph Cut Max-Flow [Boykov2004] is applied to optimally cut the volume. This cut represents the extracted mesh surface. We filter bad cells on the surface. We finally apply a Laplacian filtering on the mesh to remove local artefacts.

At this point, the mesh can also be simplified to reduce unnecessary vertices.

[Jancosek2014]	Exploiting Visibility Information in Surface Reconstruction to Preserve Weakly Supported Surfaces, Michal Jancosek, Tomas Pajdla
[Jancosek2011]	Multi-view reconstruction preserving weakly-supported surfaces, Michal Jancosek, Tomas Pajdla, CVPR 2011
[Jancosek2010]	Hallucination-free multi-view stereo, M. Jancosek and T. Pajdla, ECCV 2010
[Labatut2009]	Robust and efficient surface reconstruction from range data, P. Labatut, J.-P. Pons, and R. Keriven, 2009
[Boykov2004]	An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Computer Vision, Yuri Boykov and Vladimir Kolmogorov. 2004

6.30 MeshResampling

Description

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

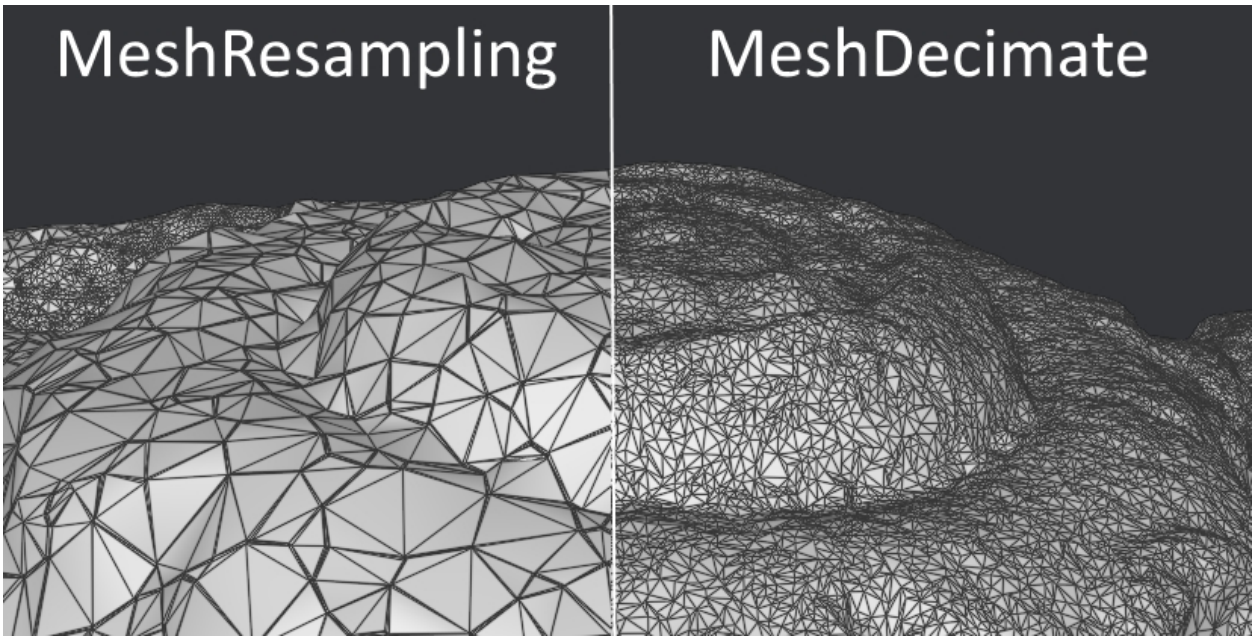
This is different from MeshDecimate!

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

settings

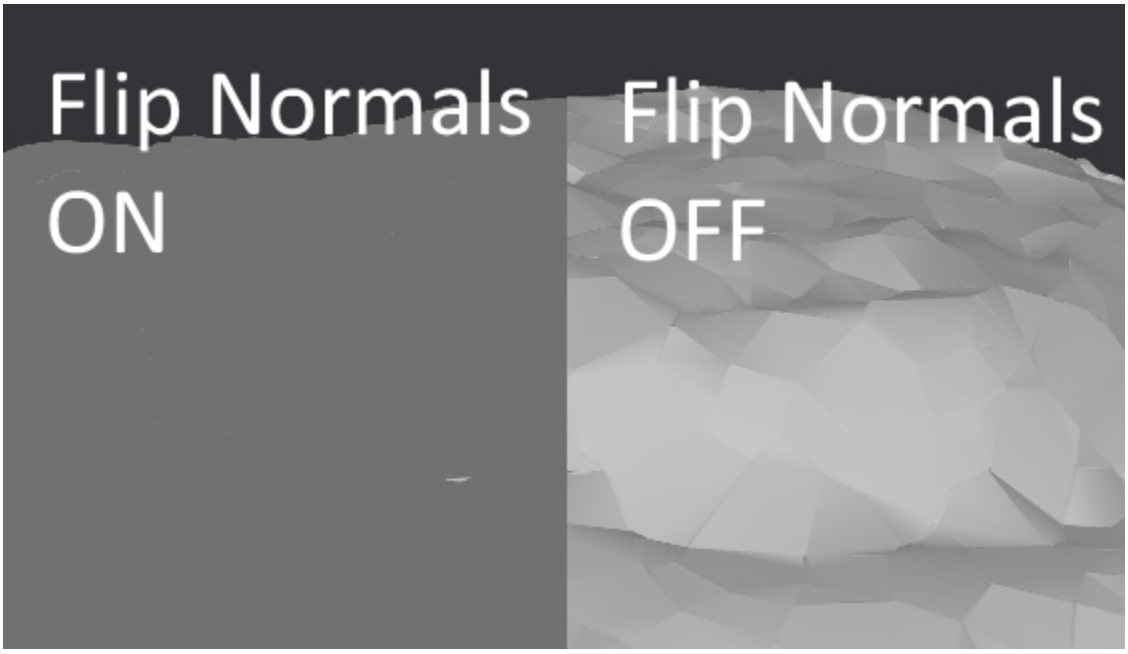
Name	Description
Input	Input Mesh (OBJ file format)
Simplification factor	Simplification factor 0.5 `(0 - 1)`
Fixed Number of Vertice	Fixed number of output vertices 0 `(0 - 1 000 000)`
Min Vertices	Min number of output vertices 0 `(0 - 1 000 000)`
Max Vertices	Max number of output vertices 0 (0 - 1 000 000)
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', 'trace']
Output mesh	Output mesh (OBJ file format) `internalFolder + mesh.obj`

Comparison MeshDecimate and MeshResampling



MeshDecimate kills vertices to reduce the density, so the vertices at the end already exist in the original mesh. MeshResampling will recreate vertices on the surface with a uniform density, so there is no common vertice with the original mesh.

Flip Normals



6.31 PanoramaCompositing

Description

settings

Name	Description
Input	Panorama Warping result
Output File Type	Output file type for the undistorted images. [‘jpg’, ‘png’, ‘tif’, **‘exr’**]
Compositer Type	Which compositer should be used to blend images [‘replace’, ‘alpha’, **‘multiband’**]
Verbose Level	[‘fatal’, ‘error’, ‘warning’, ‘info’, ‘debug’, ‘trace’]
Output	Output Panorama (internalFolder + ‘panorama.FileType)

6.32 PanoramaEstimation

Description

settings

Name	Description
Input	SfM Data File
Features Folder	
Features Folders	Folder(s) containing the extracted features.
Matches Folder	
Matches Folders	Folder(s) in which computed matches are stored.
Describer Types	Describer types used to describe an image. [**'sift'**, 'sift_float', 'sift_upright', 'akaze', 'akaze_liop', 'akaze_mldb', 'cctag3', 'cctag4', 'sift_ocv', 'akaze_ocv']
Orientation	Orientation (**0**-6)
Longitude offset (deg.)	Offset to the panorama longitude (-180.0-180.0, **0**)
Latitude offset (deg.)	Offset to the panorama latitude (-90.0-90.0, **0**)
Rotation Averaging Method	Method for rotation averaging : • L1 minimization • **L2 minimization**
Relative Rotation Method	Method for relative rotation : • from essential matrix • from **homography matrix**
Refine	Refine camera relative poses, points and optionally internal camera parameter
Force Lock of All Intrinsic Camera Parameters.	Force to keep constant all the intrinsics parameters of the cameras (focal length, principal point, distortion if any) during the reconstruction. This may be helpful if the input cameras are already fully calibrated.
Verbose Level	['fatal', 'error', 'warning', 'info', 'debug', 'trace']
Output Folder	internalFolder
Output SfMData File	Path to the output sfmdata file (internalFolder + 'sfmData.abc)

6.33 PanoramaExternalInfo

Description

WORKAROUND for valid Tractor graph submission

settings

Name	Description
Input	SfMData file
Xml Config	XML Data File
Matches Folder	
Matches Folders	Folder(s) in which computed matches are stored. (WORKAROUND for valid Tractor graph submission)
Verbose Level	['fatal', 'error', 'warning', 'info', 'debug', 'trace']
Output	Path to the output sfmdata file (internalFolder + 'sfmData.abc')

6.34 PanoramaInit

Description

This node allows to setup the Panorama: 1/ Enables the initialization the cameras from known position in an XML file (provided by a motorized panorama head). 2/ Enables to setup Full Fisheye Optics (to use an Equirectangular camera model). 3/ To automatically detects the Fisheye Circle (radius + center) in input images or manually adjust it.

settings

Name	Description
Input	SfM Data File
Xml Config	XML Data File (Papywizard xml file format)
Dependency	Folder(s) in which computed features are stored. (WORKAROUND for valid Tractor graph submission)
Full Fisheye	To declare a full fisheye panorama setup
Estimate Fisheye Circle	Automatically estimate the Fisheye Circle center and radius instead of using user values.
Fisheye Center	Center of the Fisheye circle (XY offset to the center in pixels).
Radius	Fisheye visibillity circle radius (% of image shortest side).
input Angle offset	Add a rotation to the input XML given poses (CCW).
Debug Fisheye Circle Detection	Debug fisheye circle detection.
Verbose Level	[‘fatal’, ‘error’, ‘warning’, ‘info’, ‘debug’, ‘trace’]
Output	Output sfmData.

6.35 PanoramaPrepareImages

Description

Prepare images for Panorama pipeline: ensures that images orientations are coherent.

settings

Name	Description
Input	SfM Data File
Verbose Level	[‘fatal’, ‘error’, ‘warning’, ‘info’, ‘debug’, ‘trace’]
Output	Output sfmData.

6.36 PanoramaWarping

Description

settings

Name	Description
Input	SfMData file
Panorama Width	Panorama width (pixels). 0 For automatic size (0-50000, **10000**)
Verbose Level	[‘fatal’, ‘error’, ‘warning’, ‘info’, ‘debug’, ‘trace’]

6.37 PrepareDenseScene

Description

- This node undistorts the images and generates EXR images

  settings

Name	Description
Input	SfMData file
ImagesFolders	Use images from specific folder(s). Filename should be the same or the image uid.
Output File Type	Output file type for the undistorted images. (jpg, png, tif, **exr**)
Save Metadata	Save projections and intrinsics information in images metadata (only for .exr images).
Save Matrices Text Files	Save projections and intrinsics information in text files.
Correct images exposure	Apply a correction on images Exposure Value
Verbose Level	[‘fatal’, ‘error’, ‘warning’, ‘info’, ‘debug’, ‘trace’]
Output	MVS Configuration file (desc.Node.internalFolder + ‘mvs.ini)
Undistorted images	List of undistorted images.

ImagesFolders

ImagesFolders option allows to override input images. This enables to use images with light patterns projected for SfM and MVS parts and do the Texturing with another set of images.

6.38 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

settings

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

6.39 SfMAlignment

Description

Align SfM file to a scene

settings

Name	Description
Input	SfMData file
Reference	Path to the scene used as the reference coordinate system
Alignment Method	Alignment Method: • **from_cameras_viewid**: Align cameras with same view Id • from_cameras_poseid: Align cameras with same pose Id • from_cameras_filepath: Align cameras with a filepath matching, using 'fileMatchingPattern' • from_cameras_metadata: Align cameras with matching metadata, using 'metadataMatchingList' • from_markers: Align from markers with the same Id
File Matching Pattern	Matching regular expression for the "from_cameras_filepath" method. You should capture specific parts of the filepath with parenthesis to define matching elements. Some examples of patterns: • Match the filename without extension (default value): **".*/(.*?).w{3}"** • Match the filename suffix after "_": ".*/.*(_.*?.w{3})" • Match the filename prefix before "_": ".*/(.*?)_.*.w{3}"
Metadata	
Metadata Matching List	List of metadata that should match to create the correspondences. If the list is empty, the default value will be used: ["Make", "Model", "Exif:BodySerialNumber", "Exif:LensSerialNumber"].
Scale	Apply scale transformation. (**True**)
Rotation	Apply rotation transformation. (**True**)
Translation	Apply translation transformation. (**True**)
Verbose Level	['fatal', 'error', 'warning', 'info', 'debug', 'trace']
Output	Aligned SfMData file internalFolder + 'alignedSfM.abc

6.40 SfMTransfer

Summary

Retrieve poses and intrinsics from another reconstruction with matching views

settings

Name	Description
Input	SfMData file
Reference	Path to the scene used as the reference to retrieve resolved poses and intrinsics.
Matching Method	Matching Method: • from_viewid: Align cameras with same view Id • from_filepath: Align cameras with a filepath matching, using 'fileMatchingPattern' • from_metadata: Align cameras with matching metadata, using 'metadataMatchingList' (**from_viewid**, from_filepath, from_metadata)
File Matching Pattern	Matching regular expression for the "from_cameras_filepath" method. You should capture specific parts of the filepath with parenthesis to define matching elements. Some examples of patterns: • Match the filename without extension (default value): **".*/(.*?).w{3}"** • Match the filename suffix after "_": ".*/.*(_.*?.w{3})" • Match the filename prefix before "_": ".*/(.*?)_.*.w{3}"
Metadata	
Metadata Matching List	List of metadata that should match to create the correspondences. If the list is empty, the default value will be used: ["Make", "Model", "Exif:BodySerialNumber", "Exif:LensSerialNumber"].
Poses	Transfer poses. (**True**)
Intrinsics	Transfer cameras intrinsics. (**True**)
Verbose Level	verbosity level (fatal, error, warning, **info**, debug, trace).
Output	SfMData file.

6.41 SfMTransform

Description

Transform/Scale SfM using given transformation, cameras, landmarks, markers. Can be used to scale SfM to real-world size.

settings

Name	Description
Input	SfMData file
Transformation Method	Transformation method: • transformation: Apply a given transformation • auto_from_cameras: Use cameras • **auto_from_landmarks**: Use landmarks • from_single_camera: Use a specific camera as the origin of the coordinate system • from_markers: Align specific markers to custom coordinates
Transformation	Required only for 'transformation' and 'from_single_camera' methods: • transformation: Align [X,Y,Z] to +Y-axis, rotate around Y by R deg, scale by S; syntax: X,Y,Z;R;S • from_single_camera: Camera UID or image filename
Landmarks Describer Types	Image describer types used to compute the mean of the point cloud. (only for "landmarks" method). (**'sift'**, 'sift_float', 'sift_upright', **'akaze'**, 'akaze_liop', 'akaze_mldb', 'cctag3', 'cctag4', 'sift_ocv', 'akaze_ocv')
Additional Scale	Additional scale to apply. (0.0-100.0, default **1.0**)
Markers	Markers alignment points
Scale	Apply scale transformation.
Rotation	Apply rotation transformation.
Translation	Apply translation transformation.
Verbose Level	verbosity level (fatal, error, warning, **info**, debug, trace).

usage:

Details:

Transformation Method: transformation

Align [X,Y,Z] to +Y-axis, rotate around Y by R deg, scale by S; syntax: X,Y,Z;R;S (all five parameters are required) This allows the user to align and scale the point cloud by explicitly specifying the scale and "up" vector [X,Y,Z] in the point cloud's reference system. The rotation is such that the specified [X,Y,Z] vector is aligned with [0,1,0] after the transformation.

The use-case to allow the user to derive the desired rotation by interactive manipulation of the point cloud in a 3D program (Meshlab), read off the transformation parameters and transform the point cloud. https://github.com/alicevision/AliceVision/pull/206

Transformation Method: from single camera

Sets a specific camera as origin and applies correct orientation if possible Provide Camera **UID** or **image filename**

Transformation Method: autofromlandmarks

Select Landmarks Describer Type CCTAG to apply a scale

6.42 SketchfabUpload

Description

Sketchfab is a popular website to share and view 3D files, this provides a node to allow direct upload to Sketchfab from Meshroom. The API key is provided by the user in the node settings.

MR version: 2020.x

settings

Name	Description
Input Files	Input Files to export
API Token	Get your token from https://sketchfab.com/settings/password
Title	Title cannot be longer than 48 characters.
Description	Description cannot be longer than 1024 characters.
License	'CC Attribution', 'CC Attribution-ShareAlike', 'CC Attribution-NoDerivs', 'CC Attribution-NonCommercial', 'CC Attribution-NonCommercial-ShareAlike', 'CC Attribution-NonCommercial-NoDerivs'
Tag	Tag cannot be longer than 48 characters.
Tags	Maximum of 42 separate tags.
Category	Adding categories helps improve the discoverability of your model. ('none', 'animals-pets', 'architecture', 'art-abstract', 'cars-vehicles', 'characters-creatures', 'cultural-heritage-history', 'electronics-gadgets', 'fashion-style', 'food-drink', 'furniture-home', 'music', 'nature-plants', 'news-politics', 'people', 'places-travel', 'science-technology', 'sports-fitness', 'weapons-military')
Publish	If the model is not published it will be saved as a draft. (False)
Inspectable	Allow 2D view in model inspector. (True)
Private	Requires a pro account. (False)
Password	Requires a pro account.
Verbose Level	verbosity level (critical, error, warning, info, debug).

6.43 StructureFromMotion

Description

The StructureFromMotion (Incremental SfM) will reconstruct 3D points from the input images. For Global SfM use the GlobalSfM node.

Input	SfMData file
Features Folder	Folder(s) containing the extracted features and descriptors.
Matches Folders	Folder(s) in which computed matches are stored.
Describer Types	Describer types used to describe an image. '**sift**', 'sift*float', 'sift*uprigh
Localizer Estimator	Estimator type used to localize cameras (**acransac**, ransac, lsmeds, lora
Observation Constraint	Observation contraint mode used in the optimization: **Basic**: Use stand
Localizer Max Ransac Iterations	Maximum number of iterations allowed in ransac step. (1-20000) **4096**
Localizer Max Ransac Error	Maximum error (in pixels) allowed for camera localization (resectioning)
Lock Scene Previously Reconstructed	This option is useful for SfM augmentation. Lock previously reconstruct
Local Bundle Adjustment	It reduces the reconstruction time, especially for large datasets (500+ in
LocalBA Graph Distance	Graph-distance limit to define the Active region in the Local Bundle Ad
Maximum Number of Matches	Maximum number of matches per image pair (and per feature type). Th
Minimum Number of Matches	Minimum number of matches per image pair (and per feature type). Th
Min Input Track Length	Minimum track length in input of SfM (**2**-10)
Min Observation For Triangulation	Minimum number of observations to triangulate a point. Set it to 3 (or
Min Angle For Triangulation	Minimum angle for triangulation. (0.1-10) **3.0**
Min Angle For Landmark	Minimum angle for landmark. (0.1-10) **2.0**
Max Reprojection Error	Maximum reprojection error. (0.1-10) **4.0**
Min Angle Initial Pair	Minimum angle for the initial pair. (0.1-10) **5.0**
Max Angle Initial Pair	Maximum angle for the initial pair. (0.1-60) **40.0**
Use Only Matches From Input Folder	Use only matches from the input matchesFolder parameter. Matches fol
Use Rig Constraint	Enable/Disable rig constraint.
Force Lock of All Intrinsic Camera Parameters.	Force to keep constant all the intrinsics parameters of the cameras (foca
Filter Track Forks	Enable/Disable the track forks removal. A track contains a fork when in
Initial Pair A	Filename of the first image (without path).
Initial Pair B	Filename of the second image (without path).
Inter File Extension	Extension of the intermediate file export. ('.abc', '.ply')
Verbose Level	Verbosity level (fatal, error, warning, info, debug, trace).
Output SfMData File	Path to the output sfmdata file (sfm.abc)
Output SfMData File	Path to the output sfmdata file with cameras (views and poses). (camer
Output Folder	Folder for intermediate reconstruction files and additional reconstructio

Point cloud density

Based on the number of features extracted from *feature extraction* and then matched in *feature matching*.

Use Rig Constraint Add support for rig of cameras. This information is used as a new constraint in the SfM. This option can now be combined with localBA. You need to use a specific folder hierarchy in the input images files (for instance: "/my/dataset/rig/0/DSLR_0001.JPG", "/my/dataset/rig/1/DSLR_0001.JPG") to provide this information.

Detailed description

The objective of this step is to understand the geometric relationship behind all the observations provided by the input images, and infer the rigid scene structure (3D points) with the pose (position and orientation) and internal calibration of all cameras. The Incremental pipeline is a growing reconstruction process. It first computes an initial two-view reconstruction that is iteratively extended by adding new views.

First, it fuses all feature matches between image pairs into tracks. Each track is supposed to represent a point in space, visible from multiple cameras. However, at this step of the pipeline, it still contains many outliers. During this fusion of matches, we remove incoherent tracks.

Then, the incremental algorithm has to choose the best initial image pair. This choice is critical for the quality of the final reconstruction. It should indeed provide robust matches and contain reliable geometric information. So, this image pair should maximize the number of matches and the repartition of the corresponding features in each image. But at the same time, the angle between the cameras should also be large enough to provide reliable geometric information.

Then we compute the fundamental matrix between these 2 images and consider that the first one is the origin of the coordinate system. Now that we know the pose of the 2 first cameras, we can triangulate the corresponding 2D features into 3D points.

After that, we select all the images that have enough associations with the features that are already reconstructed in 3D. This algorithm is called next best views selection. Based on these 2D-3D associations it performs the resectioning of each of these new cameras. The resectioning is a Perspective-n-Point algorithm (PnP) in a RANSAC framework to find the pose of the camera that validates most of the features associations. On each camera, a non-linear minimization is performed to refine the pose.

From these new cameras poses, some tracks become visible by 2 or more resected cameras and it triangulates them. Then, we launch a Bundle Adjustment to refine everything: extrinsics and intrinsics parameters of all cameras as well as the position of all 3D points. We filter the results of the Bundle Adjustment by removing all observations that have high reprojection error or insufficient angles between observations.

As we have triangulated new points, we get more image candidates for next best views selection. We iterate like that, adding cameras and triangulating new 2D features into 3D points and removing 3D points that became invalidated, until we can't localize new views.

Many other approaches exists like Global [Moulon2013], Hierarchical [Havlena2010], [Toldo2015] or multi-stage [Shah2014] approaches.

References

[Cheng2014]	Fast and Accurate Image Matching with Cascade Hashing for 3D Reconstruction Jian Cheng, Cong Leng, Jiaxiang Wu, Hainan Cui, Hanqing Lu. CVPR 2014
[Fischler1981]	Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Fischler, Martin A., and Robert C. Bolles. 1981
[Moulon2013]	Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion. Pierre Moulon, Pascal Monasse and Renaud Marlet. ICCV 2013
[Moulon2012a]	Adaptive structure from motion with a contrario model estimation. Pierre Moulon, Pascal Monasse, and Renaud Marlet. ACCV 2012
[Moulon2012b]	Automatic homographic registration of a pair of images, with a contrario elimination of outliers. Moisan, Lionel, Pierre Moulon, and Pascal Monasse. IPOL 2012
[Moulon2012c]	Unordered feature tracking made fast and easy, Pierre Moulon and Pascal Monasse, CVMP 2012
[Kneip2011]	A Novel Parametrization of the P3P-Problem for a Direct Computation of Absolute Camera Position and Orientation. Kneip, L.; Scaramuzza, D. ; Siegwart, R. CVPR 2011
[Lepetit2009]	EPnP: An Accurate O(n) Solution to the PnP Problem. V. Lepetit and F. Moreno-Noguer and P. Fua, IJCV 2009
[Nister2004]	An Efficient Solution to the Five-Point Relative Pose. D. Nister PAMI 2004
[Havlena2010]	Efficient Structure from Motion by Graph Optimization. M. Havlena, A. Torii, and T. Pajdla. ECCV 2010
[Toldo2015]	Hierarchical structure-and-motion recovery from uncalibrated images. R. Toldo, R. Gherardi, M. Farenzena and A. Fusiello. CVIU 2015
[Shah2014]	Multistage SFM: Revisiting Incremental Structure from Motion, Rajvi Shah, Aditya Deshpande, P J Narayanan, 2014
[Moulon2015]	Robust and precise positioning of image networks, Pierre Moulon 2015 (in French)[38]
[Martinec2008]	Robust Multiview Reconstruction. Daniel Martinec, 2008
[Hartley2000]	Multiple view geometry in computer vision. Richard Hartley and Andrew Zisserman. Cambridge, 2000
[Ceres]	Ceres Solver, Sameer Agarwal and Keir Mierle and Others[39]
[OpenGV]	The OpenGV library[40]

6.44 Texturing

Description

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

[38] https://hal.archives-ouvertes.fr/file/index/docid/996935/filename/These_MOULON.pdf
[39] http://ceres-solver.org/
[40] https://github.com/laurentkneip/opengv

MVS Configuration file	.../mvs.ini
Input Dense Reconstruction	Path to the dense reconstruction result (mesh with per vertex visibility)
Other Input Mesh	Optional input mesh to texture. By default, it will texture the result of the reconstruction.
Texture Side	Output texture size 1024, 2048, 4096, **8192**, 16384
Texture Downscale	Texture downscale factor1, **2**, 4, 8
Texture File Type	Texture File Type 'jpg', **'png'**, 'tiff', 'exr'
Unwrap Method	Method to unwrap input mesh if it does not have UV coordinates **Basic** (> 600k faces) fast and simple. Can generate multiple atlases LSCM (<= 600k faces): optimize space. Generates one atlas ABF (<= 300k faces): optimize space and stretch. Generates one atlas
Fill Holes	Fill Texture holes with plausible values True/**False**
Padding	Texture edge padding size in pixel (0-100)
Max Nb of Images For Fusion	Max number of images to combine to create the final texture (0-10)
Best Score Threshold	0.0 to disable filtering based on threshold to relative best score (0.0-1.0)
Angle Hard Threshold	0.0 to disable angle hard threshold filtering (0.0, 180.0)
Force Visible By All Vertices	Triangle visibility is based on the union of vertices visiblity.True/**False**
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.
Visibility Remapping Method	Method to remap visibilities from the reconstruction to the input mesh (Pull, Push, **Pull-Push**).
Verbose Level	verbosity level (fatal, error, warning, **info**, debug, trace).
Output Folder	Folder for output mesh: OBJ, material and texture files.
Output Mesh	Folder for output mesh: OBJ, material and texture files. internalFolder + 'texturedMesh.obj
Output Material	Folder for output mesh: OBJ, material and texture files. internalFolder + 'texturedMesh.mtl
Output Textures	Folder for output mesh: OBJ, material and texture files. internalFolder + 'texture_*.png

About:

Texture Downscale

Downscaling to 4 or 8 will reduce the texture quality but speed up the computation time.

Set Texture Downscale to 1 instead of 2 to get the maximum possible resolution with the resolution of your images.

Best Score Threshold

This parameter is a contraint to limit the number of source images we use in the color fusion. It is not related to the number of output texture files. There is no such parameter, the only thing you can do is to increase the image resolution.

Unwrap Method

If you decimate your mesh to a reasonable size, you can also change the unwrapMethod to LSCM or ABF which will generate only one texture file. But it will not work if you mesh is too heavy, check the tooltip:

Method to unwrap input mesh if it does not have UV coordinates.

- Basic (> 600k faces) fast and simple. Can generate multiple atlases.
- LSCM (<= 600k faces): optimize space. Generates one atlas.
- ABF (<= 300k faces): optimize space and stretch. Generates one atlas.

  https://github.com/alicevision/meshroom/issues/211#issuecomment-416184229

The approach is based on a generalization of the multi-band blending in [Burt1983] applied to 3D texturing with weighting strategies based on visibility and varying resolution. It is in the same spirit than [Baumberg2002] and [Allene2008].

References

[Burt1983]	A Multiresolution Spline with Application to Image Mosaics. P. J. Burt and E. H. Adelson. ACM Trans. Graph. 1983[41]
[Baumberg2002]	Blending images for texturing 3D models. A. Baumberg. BMVC 2002[42]
[Allene2008]	Seamless image-based texture atlases using multi-band blending. C. Allene and J. Pons and R. Keriven. ICPR 2008[43]

[41] https://doi.org/10.1145/245.247
[42] http://www.bmva.org/bmvc/2002/papers/49/full_49.pdf
[43] https://doi.org/10.1109/ICPR.2008.4761913

CHAPTER

SEVEN

TUTORIALS

7.1 Turntable

It is possible to use a turntable. To improve the results it might be useful to mask the images.

Currently, Meshroom does not support masking but you can see #188[44] 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/

https://conservancy.umn.edu/handle/11299/172480?show=full

or create your own.

7.2 Tutorial: Meshroom for Beginners

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

[44] https://github.com/alicevision/meshroom/issues/188

7.2.1 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.

7.2.2 Step 0: System requirements and installation

Meshroom software releases are self-contained portable packages. They are uploaded on the project's GitHub page[45]. 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[46] 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.

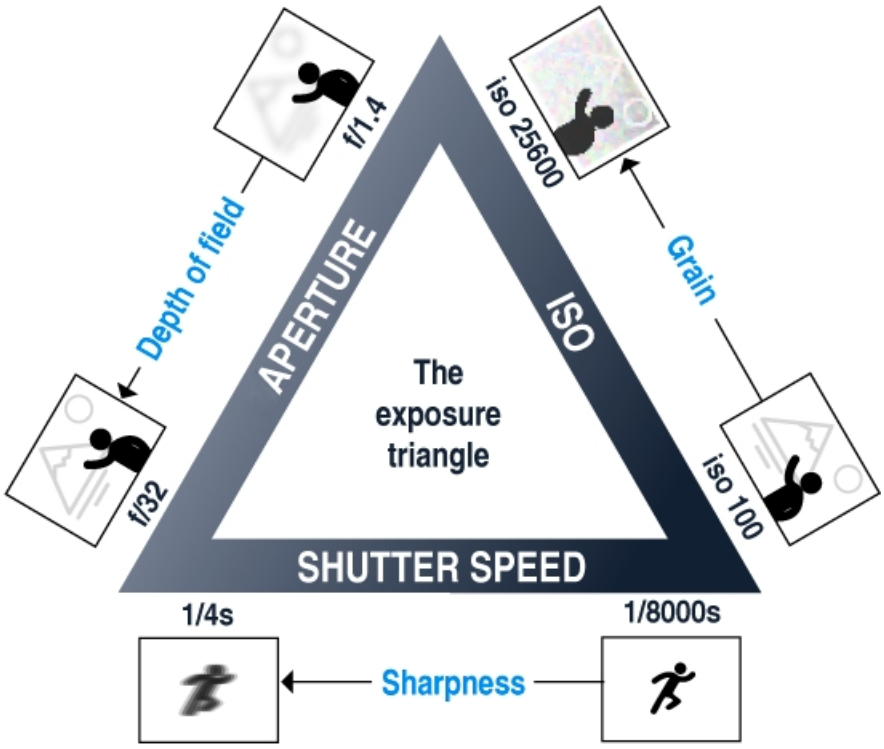
Meshroom is released in open source under the permissive MPLv2 license[47], see Meshroom COPYING[48] for more information.

7.2.3 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 accomodate 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.

[45] https://github.com/alicevision/meshroom/releases
[46] https://developer.nvidia.com/cuda-gpus
[47] https://www.mozilla.org/en-US/MPL/2.0
[48] https://github.com/alicevision/meshroom/blob/develop/COPYING.md



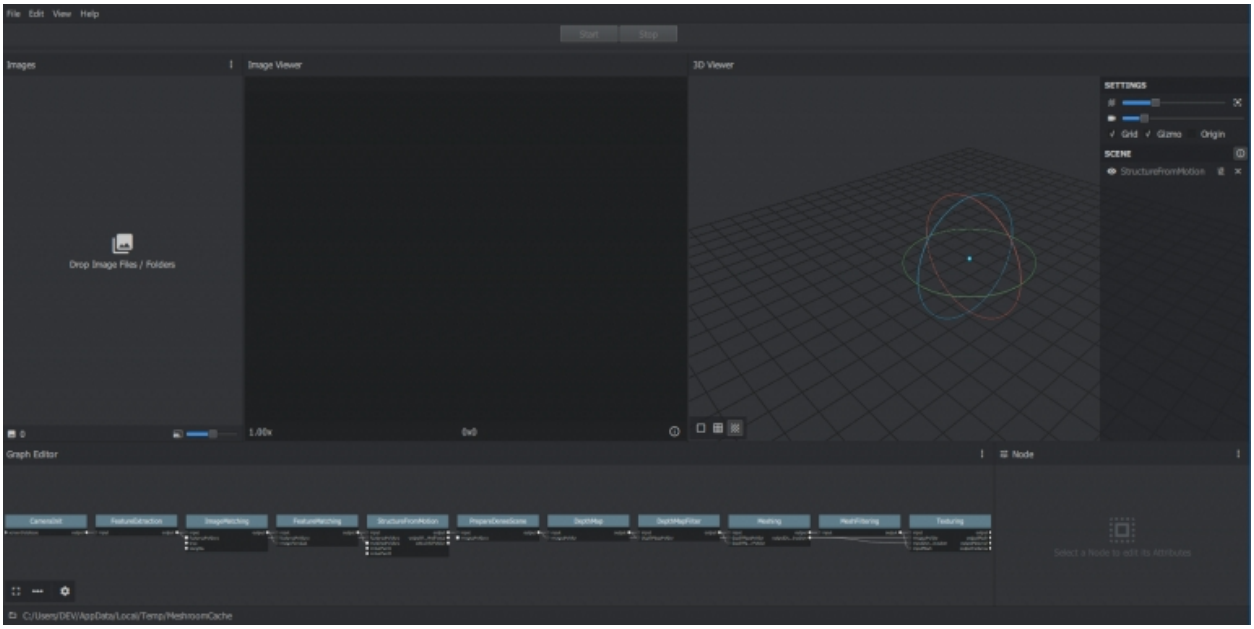
The main goal is to have sharp images without motion blur and without depth blur. So you should use tripods or fast shutter speed to avoid motion blur, reduce the aperture (high f-number) to have a large depth of field, and reduce the ISO to minimize the noise.

7.2.4 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.



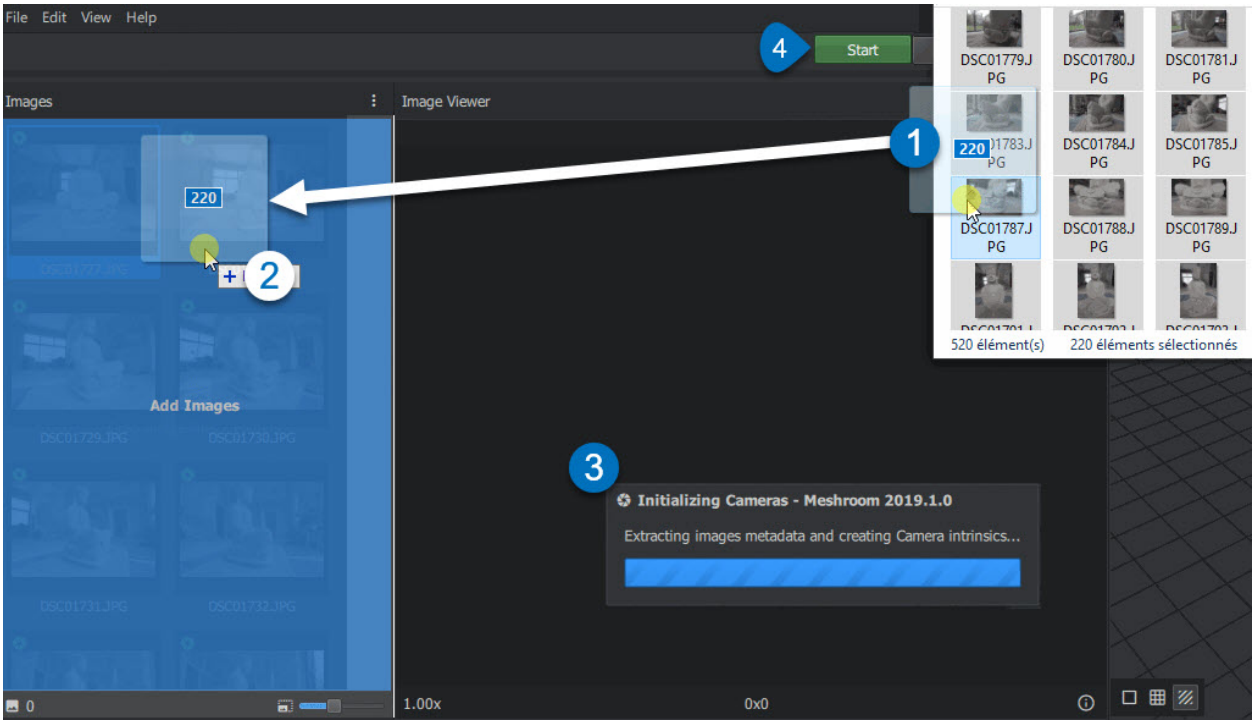
Meshroom User Interface

7.2.5 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

7.2.6 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.

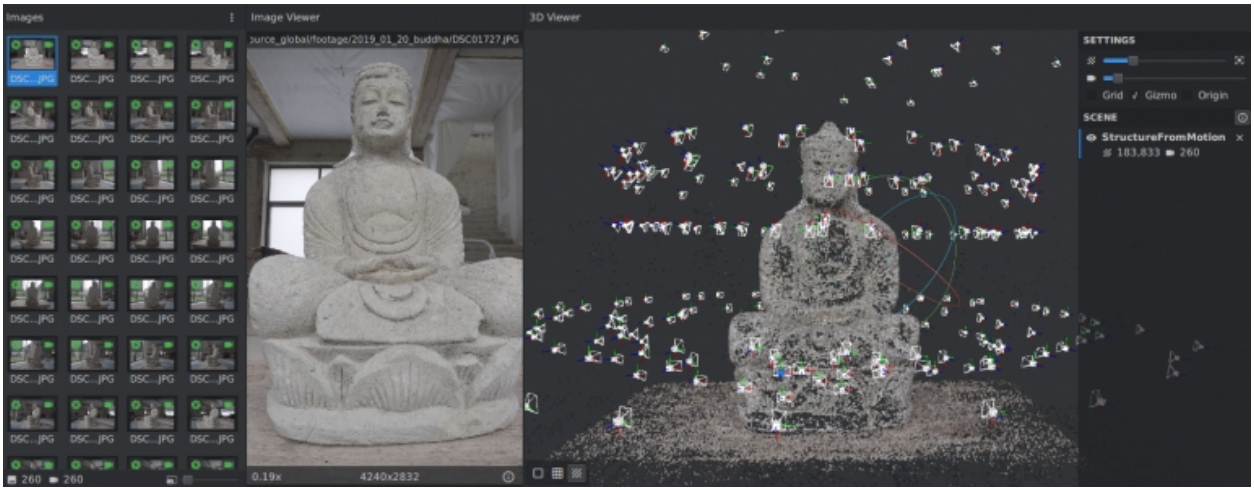


Image selection is synchronized between "Images" and "3D Viewer" panels.

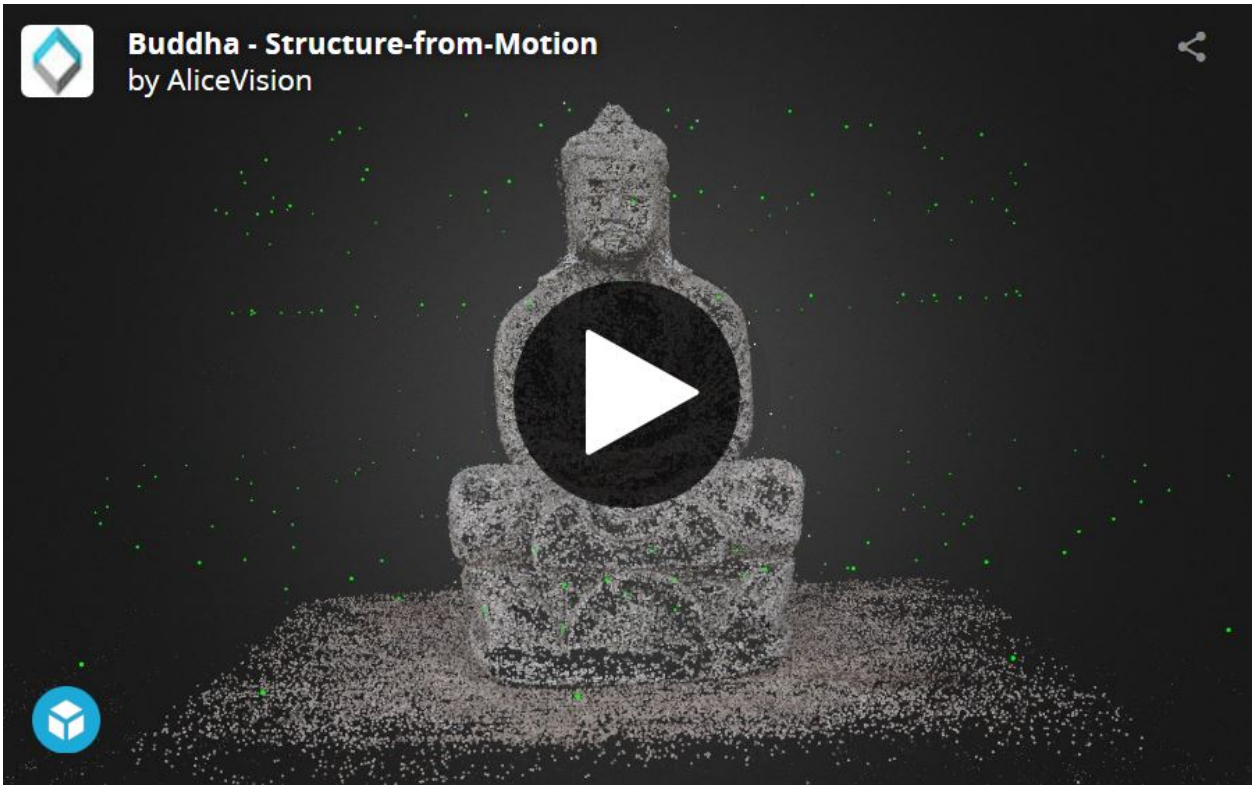
3D Viewer interactions are mostly similar to Sketchfab's:

- **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[49] by AliceVision[50] on Sketchfab[51]

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.

[49] https://sketchfab.com/3d-models/buddha-structure-from-motion-0983e6ab444f47789ca3ce2a5fcdf2b9?utm_campaign=0983e6ab444f47789ca3ce2a5fcdf2b9&utm_medium=embed&utm_source=oembed

[50] https://sketchfab.com/AliceVision?utm_campaign=0983e6ab444f47789ca3ce2a5fcdf2b9&utm_medium=embed&utm_source=oembed

[51] https://sketchfab.com?utm_campaign=0983e6ab444f47789ca3ce2a5fcdf2b9&utm_medium=embed&utm_source=oembed

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

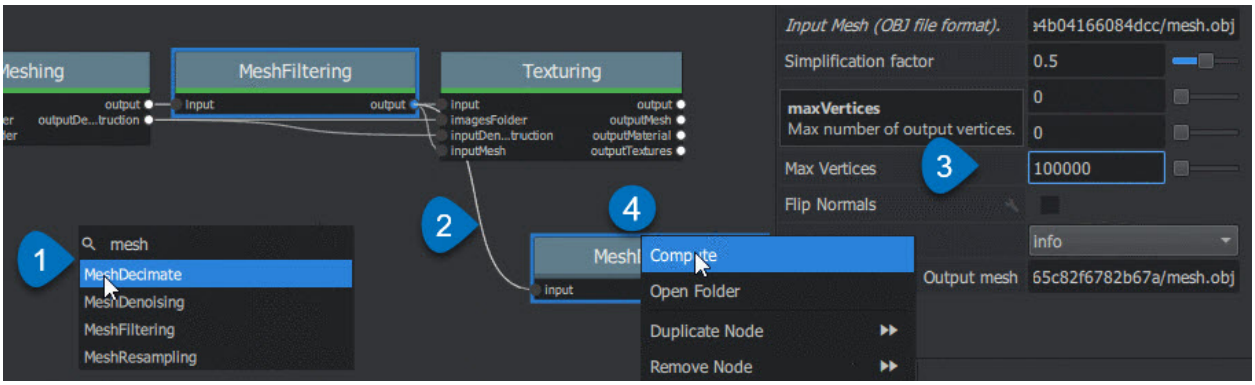
There is no export step at the end of the process: the resulting files are already available on disk. You can right-click on a media and select "Open Containing Folder" to retrieve them. By doing so on "Texturing", we get access to the folder containing the OBJ and texture files.

Buddha – Default Pipeline[52] by AliceVision[53] on Sketchfab[54]

7.2.7 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
-

[52] https://sketchfab.com/3d-models/buddha-default-pipeline-65ed60e8d72645ce83017d848611be32?utm_campaign=65ed60e8d72645ce83017d848611be32&utm_medium=embed&utm_source=oembed

[53] https://sketchfab.com/AliceVision?utm_campaign=65ed60e8d72645ce83017d848611be32&utm_medium=embed&utm_source=oembed

[54] https://sketchfab.com?utm_campaign=65ed60e8d72645ce83017d848611be32&utm_medium=embed&utm_source=oembed

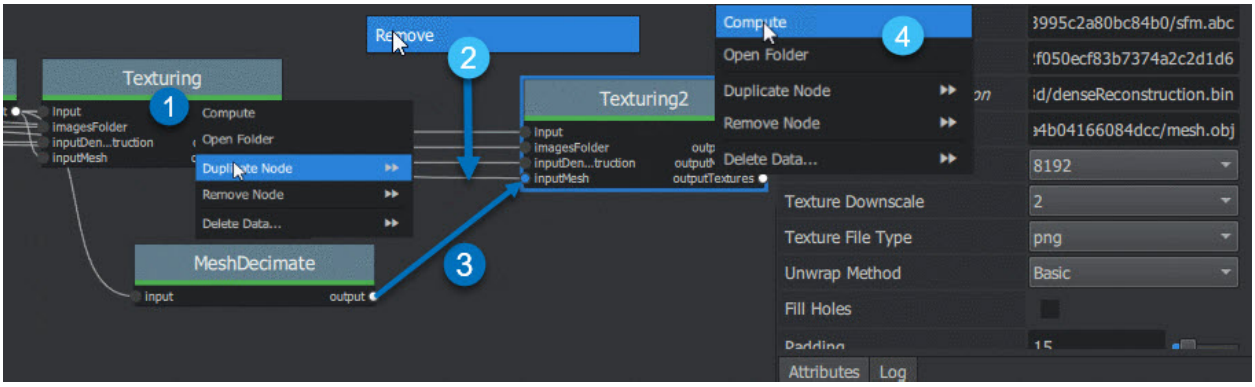
7.2.8 Step 6: Retexturing after Retopology

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.

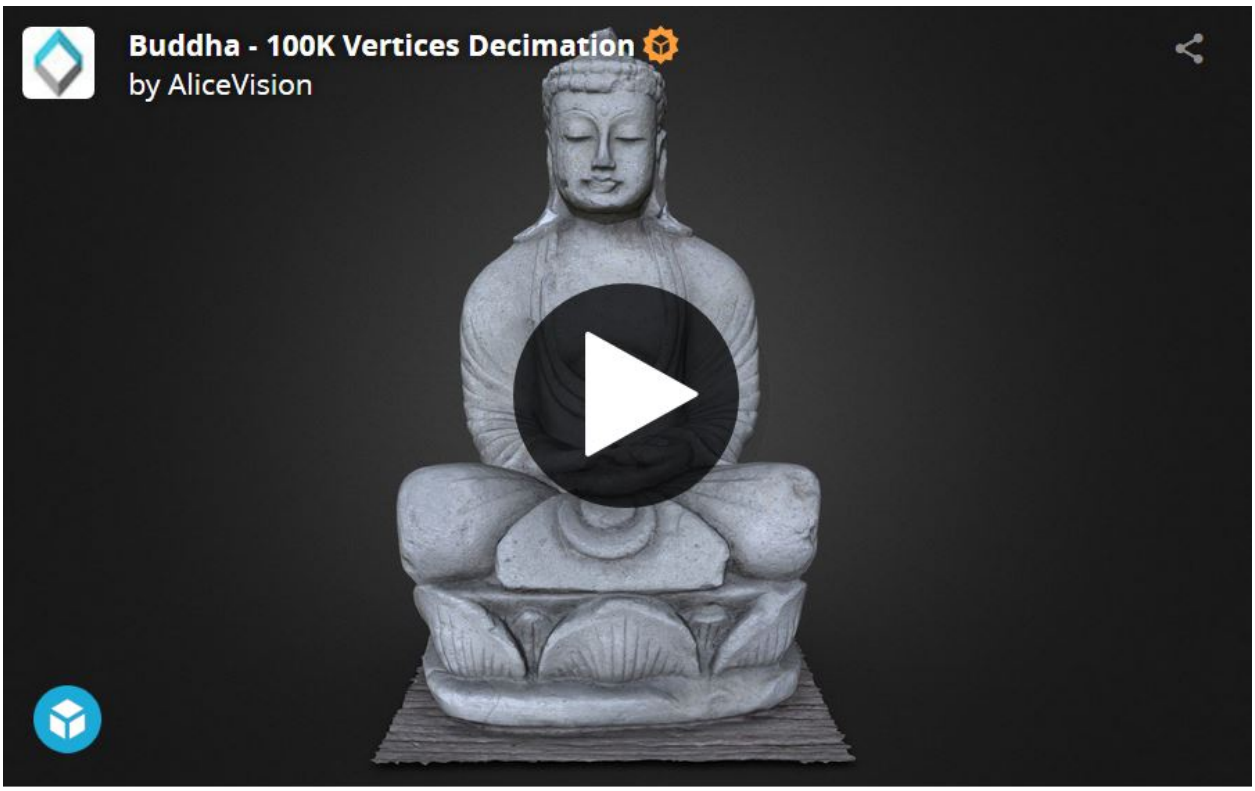
Let's head back to the Graph Editor and do the following operations:

- **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.



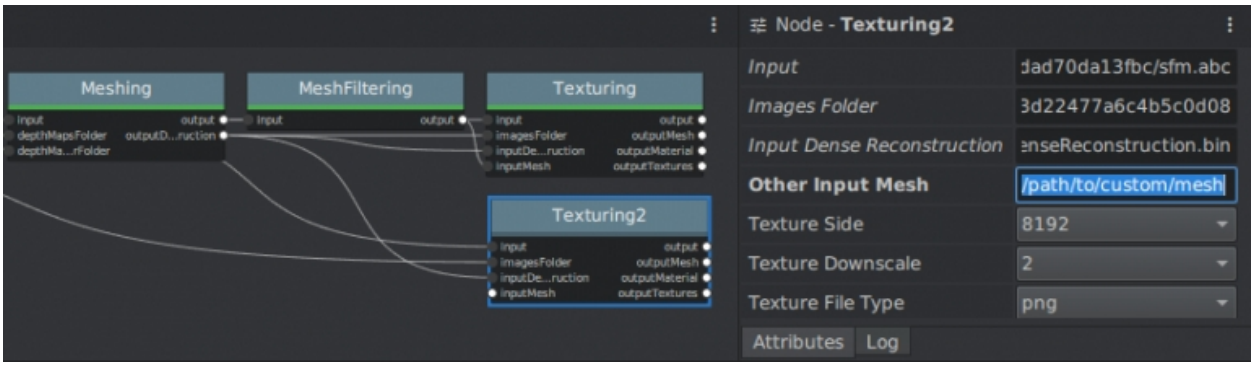
Retexture the decimated mesh using a second Texturing node



Buddha – 100K Vertices Decimation[55] by AliceVision[56] on Sketchfab[57]

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

7.2.9 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

  ("

  ⇒

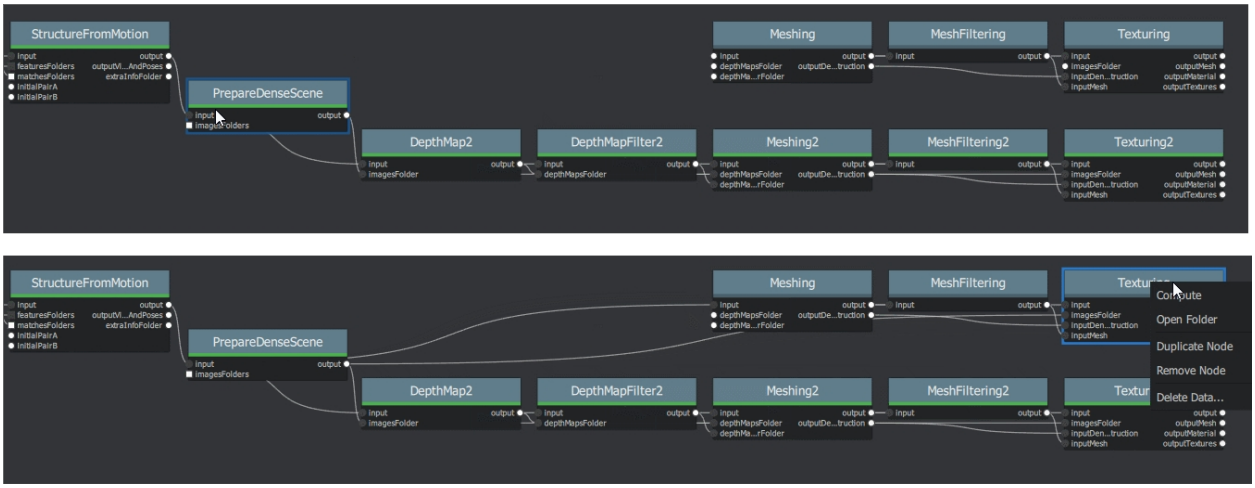
  " icon) to create a branch in the graph and keep the previous result available.

  - 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

[55] https://sketchfab.com/3d-models/buddha-100k-vertices-decimation-7648dd79fc294bba85f1bd4ff629c1d1?utm_campaign=7648dd79fc294bba85f1bd4ff629c1d1&utm_medium=embed&utm_source=oembed

[56] https://sketchfab.com/AliceVision?utm_campaign=7648dd79fc294bba85f1bd4ff629c1d1&utm_medium=embed&utm_source=oembed

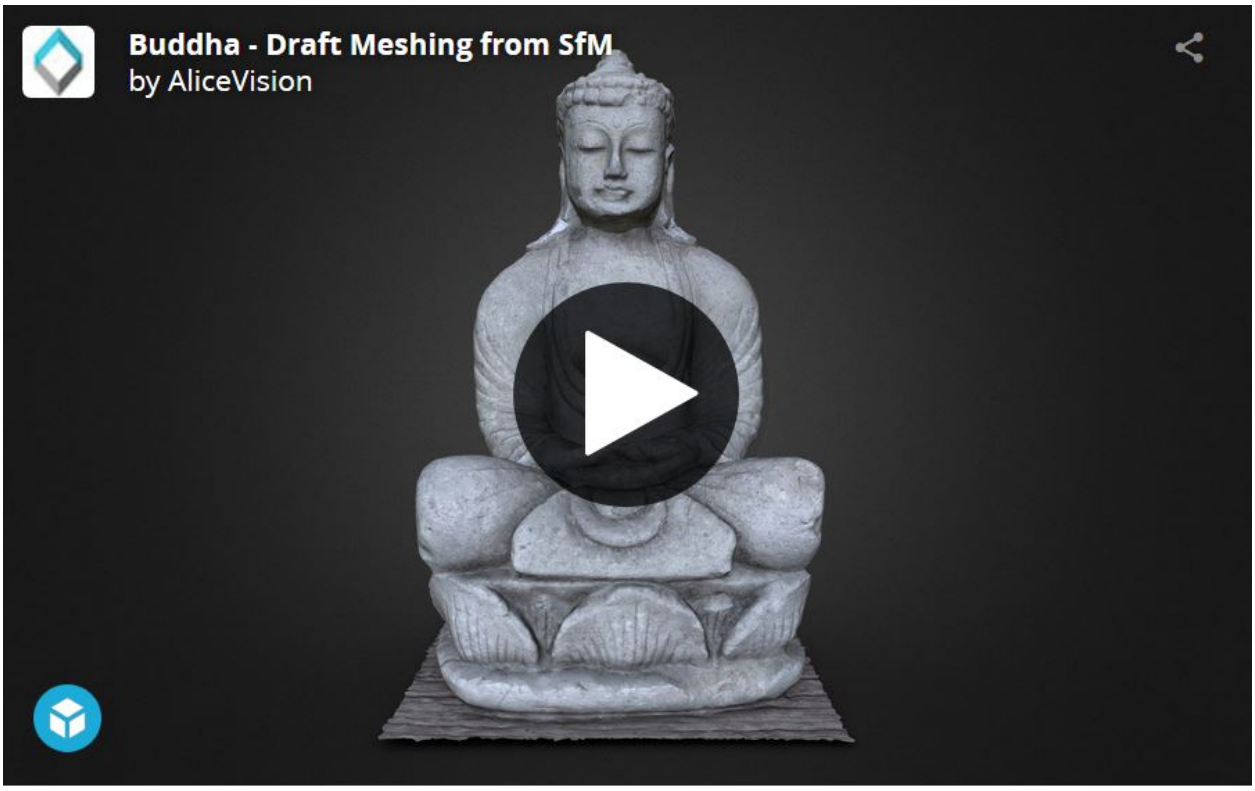
[57] https://sketchfab.com?utm_campaign=7648dd79fc294bba85f1bd4ff629c1d1&utm_medium=embed&utm_source=oembed



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[58] by AliceVision[59] on Sketchfab[60]

7.2.10 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, FeatureMatching 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**).

\3. 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.

[58] https://sketchfab.com/3d-models/buddha-draft-meshing-from-sfm-4c4219b78c804deb95f7ef3b456c721c?utm_campaign=4c4219b78c804deb95f7ef3b456c721c&utm_medium=embed&utm_source=oembed

[59] https://sketchfab.com/AliceVision?utm_campaign=4c4219b78c804deb95f7ef3b456c721c&utm_medium=embed&utm_source=oembed

[60] https://sketchfab.com?utm_campaign=4c4219b78c804deb95f7ef3b456c721c&utm_medium=embed&utm_source=oembed

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[61].

7.2.11 Step 9: Upload results on Sketchfab

Results can be uploaded using the Sketchfab web interface, but Meshroom also provides an export tool to Sketchfab.

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
- To directly publish your model from Meshroom, create a new SketchfabUpload node and connect it to the Texturing node.

You can see some 3D scans from the community here[62] and on our **Sketchfab page**.

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

[61] https://github.com/alicevision/meshroom/wiki
[62] https://sketchfab.com/AliceVision/likes

CHAPTER

EIGHT

CAPTURING

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

8.1 Basics

- Your scene/object should be well lit.
- Avoid shadows, reflections, and transparent objects.
- Take the photos in diffuse or indirect lighting, such as on an overcast day (outdoor) or using multiple light sources (indoor).
- Don´t use the flash setting on the camera.
- Do not change the focal length/zoom while shooting. Use a fixed focal length lens if possible.
- Try to take pictures from all angles.
- Avoid moving objects in the scene or background.
- If taking pictures using a rotating rig, make sure to use a plain color background with no distiguishable features.
- The object of interest should always fill most of the image.
- Take images with a side overlap of 60% minimum and frontal overlap of 80% minimum.
- For each shot, move to a new position (or rotate the object).
- Do not take multiple images from the same spot.
- For better coverage, you can photograph an area multiple times in different acquisition patterns.
- Avoid shaky, blurry, or warped images.
- The more images you have, the better. You can always filter out repetitive or poor quality images to reduce processing time.

Details

Tutorials

CHAPTER

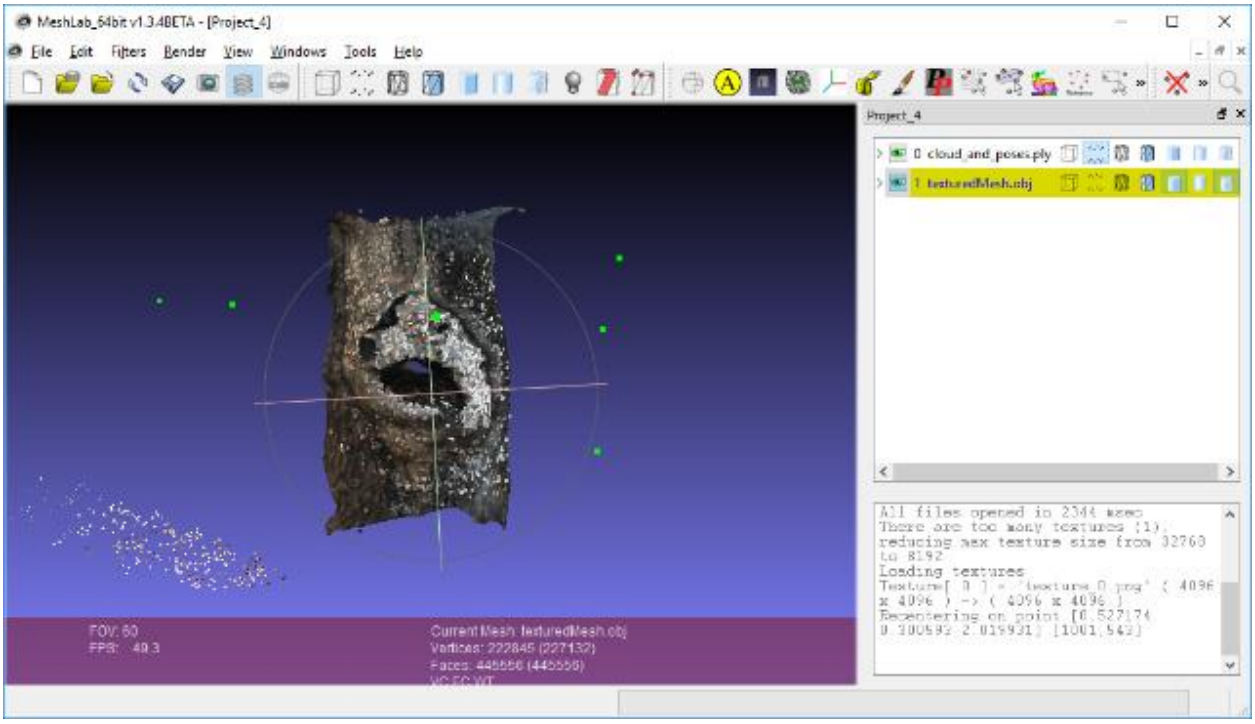
NINE

MORE

9.1 View and Edit Models

9.1.1 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[63] then go to **Filters :math:`Rightarrow` Normals, Curvatures ** and **Orientation :math:`Rightarrow` Transform: Rotate** ** ** and align it yourself from there.

[63] http://www.meshlab.net/#download

** **

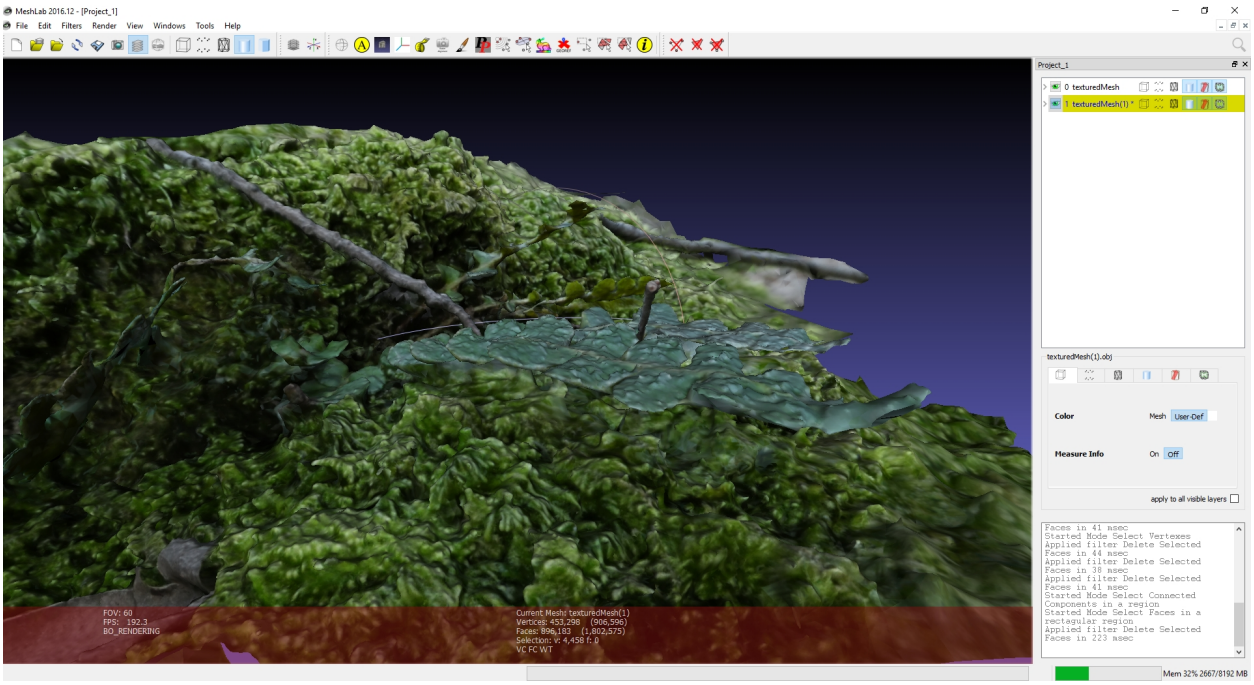
There might be some parts of the model or the scene you want to remove.

You can select ….. then remove…

http://www.banterle.com/francesco/courses/2017/be_3drec/slides/Meshlab.pdf

http://[64]

www.scanner.imagefact.de/tut/meshlabTut.pdf[65]



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[66] MeshLab Basics: Navigation[67]

MeshLab Basics: Selection, part one[68]

MeshLab Basics: Selection, part two[69]

Cleaning: Triangles and Vertices Removal[70]

Cleaning: Basic filters[71]

Mesh Processing: Decimation[72] Meshlab Processing: Smoothing[73]

[64] http://www.scanner.imagefact.de/tut/meshlabTut.pdf
[65] http://www.scanner.imagefact.de/tut/meshlabTut.pdf
[66] https://www.youtube.com/channel/UC70CKZQPj_ZAJ0Osrm6TyTg
[67] https://www.youtube.com/watch?v=Sl0vJfmj5LQ
[68] https://www.youtube.com/watch?v=xj3MN7K6kpA
[69] https://www.youtube.com/watch?v=Bc3GdJ6Ddsc
[70] https://www.youtube.com/watch?v=m2nmeJj5Ij4
[71] https://www.youtube.com/watch?v=aoDLrXp1sfY
[72] https://www.youtube.com/watch?v=PWM6EGVVNQU
[73] https://www.youtube.com/watch?v=4mwm9eMJaXY

MeshLab Basics: Scale to real measures[74]

9.1.2 Blender

For detailed instructions visit the blender homepage[75] or the blender[76] youtube channel[77] .

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

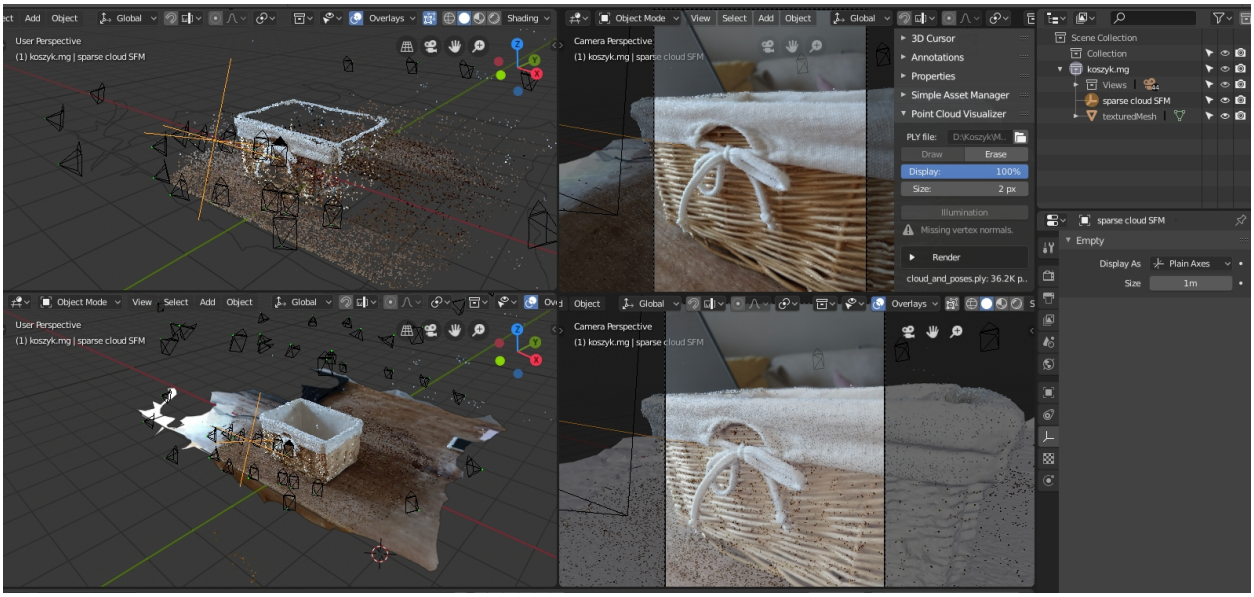
https://www.youtube.com/watch?v=RmMDFydHeso

9.1.3 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[78] for details.



[74] https://www.youtube.com/watch?v=6psAppbOOXM
[75] https://www.blender.org/
[76] https://www.youtube.com/user/BlenderFoundation
[77] https://www.youtube.com/user/BlenderFoundation
[78] https://github.com/tibicen/meshroom2blender

9.1.4 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

9.1.5 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

9.1.6 CloudCompare

3D point cloud and mesh processing software

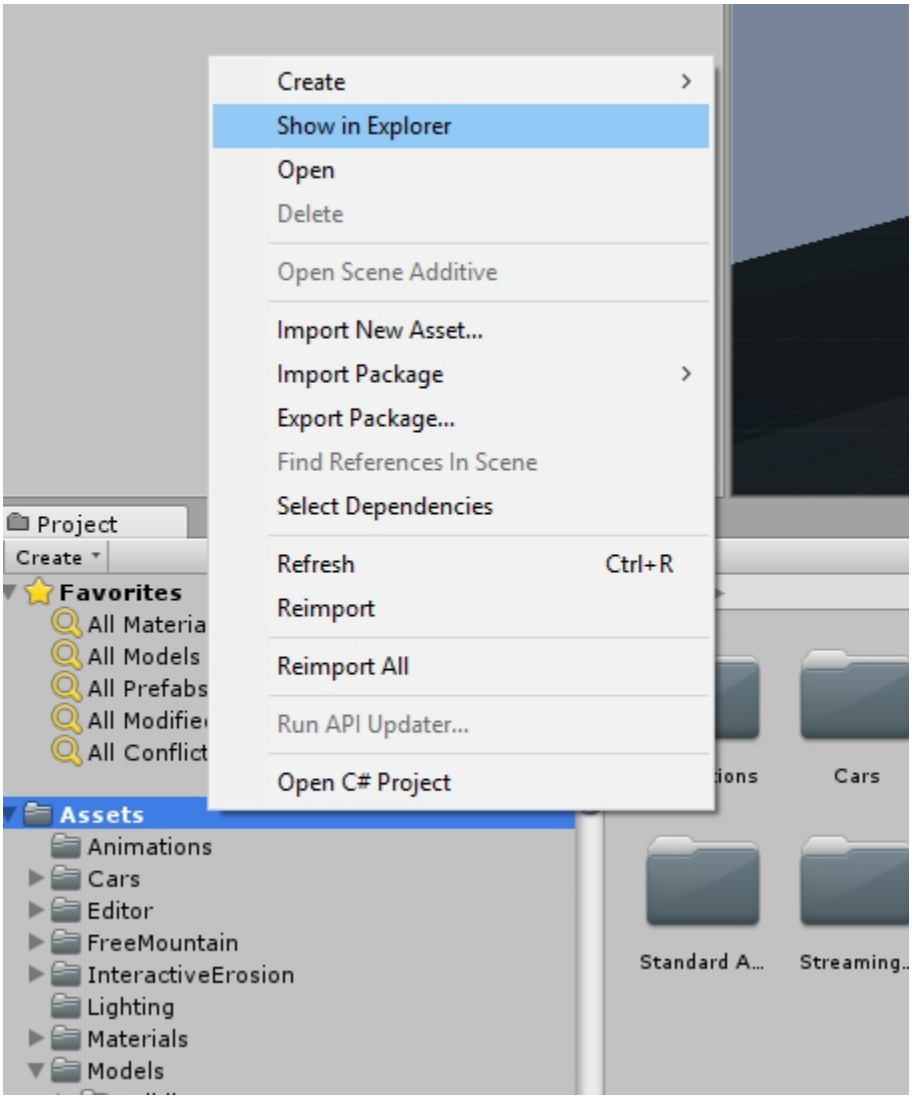
Open Source Project

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

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

tutorial

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



9.1.7 Export model to Unity

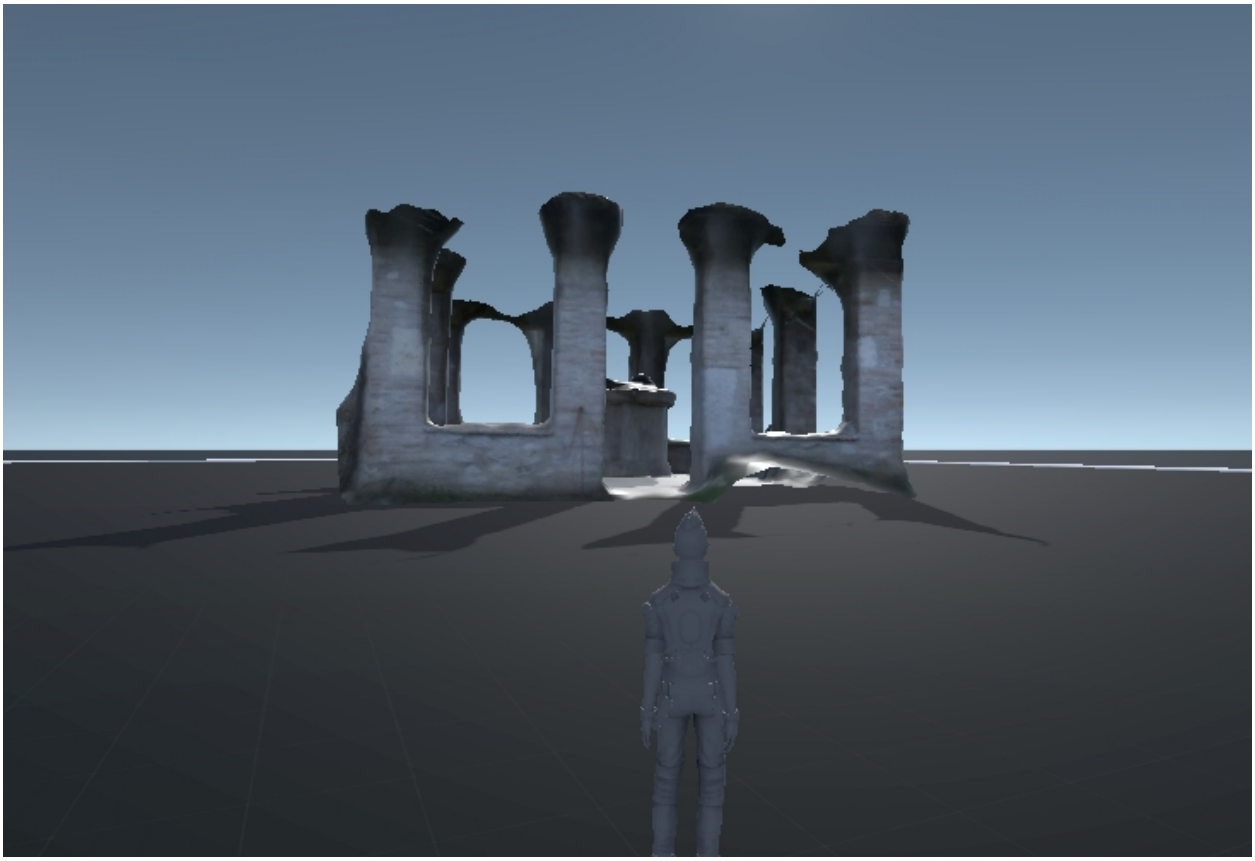
Start Unity, open your project and your asset folder.

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[79] .

Here is a manual on how to optimize photogrammetry objects inside Unity: Unity Photogrammetry Workflow[80] .. image:: 100000000000076E00000401AC14E84A53702851.jpg

9.1.8 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.

[79] https://unity3d.com
[80] https://unity3d.com/files/solutions/photogrammetry/Unity-Photogrammetry-Workflow_2017-07_v2.pdf

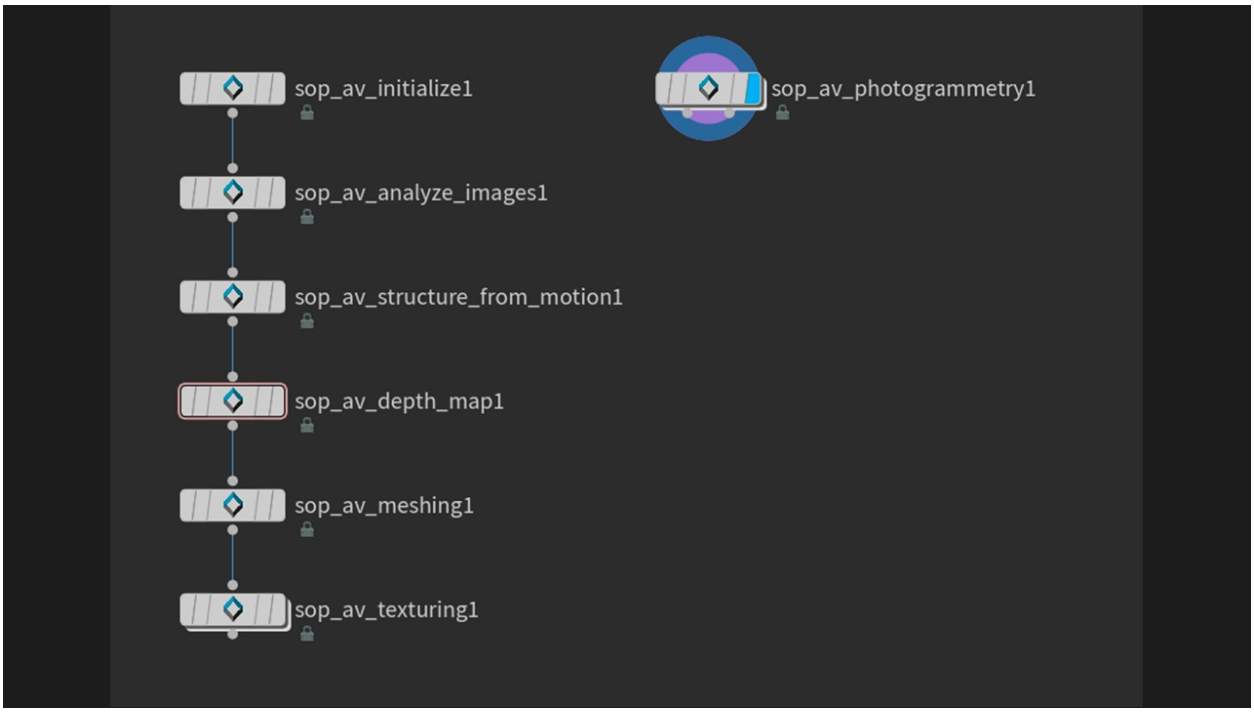
Alembic bridge

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.



9.1.9 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[81] . 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.

[81] https://www.sidefx.com/products/compare/

9.2 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...

9.3 Print your model

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

9.4 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[82] will work with Entangle.

3 GPhoto (Linux)

http://www.gphoto.org/

4 Sofortbildapp (OSX)

http://www.sofortbildapp.com/

[82] http://www.gphoto.org/doc/remote/

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

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...

9.5 Related Projects

..image:: ofxMVG.jpg

9.5.1 ofxMVG

Camera Localization OpenFX Plugin for Nuke

https://github.com/alicevision/ofxMVG

Not available at the moment.

..image:: marker2.jpg

9.5.2 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

Marker library

Markers to print are located here[83] .

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).

[83] https://github.com/alicevision/CCTag/blob/develop/markersToPrint

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[84] .

..image:: marker3.jpg

9.5.3 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

[84] https://github.com/tpruvot/ccminer/wiki/Compatibility

CHAPTER

TEN

FAQ FROM GH-WIKI

10.1 Crashed at Meshing

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

#243[85] #303[86]

10.2 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[87])

10.3 Error: Graph is being computed externally

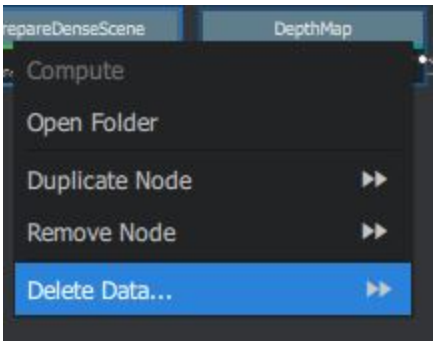
Unexpected exit of Meshroom while processing can cause the “Graph is being computed externally” problem.`#249`_

The Start and Stop buttons are greyed out.

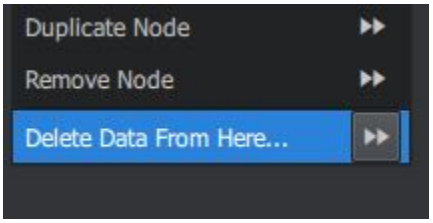
Background: When Meshroom is terminated unexpectedly, files are left in the cache folders. When you open such a project, Meshroom will think, based on the residual files, that parts of the pipeline are computed externally. (This feature ([Renderfarm](https://github.com/alicevision/meshroom/wiki/Large-scale-dataset)) 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).

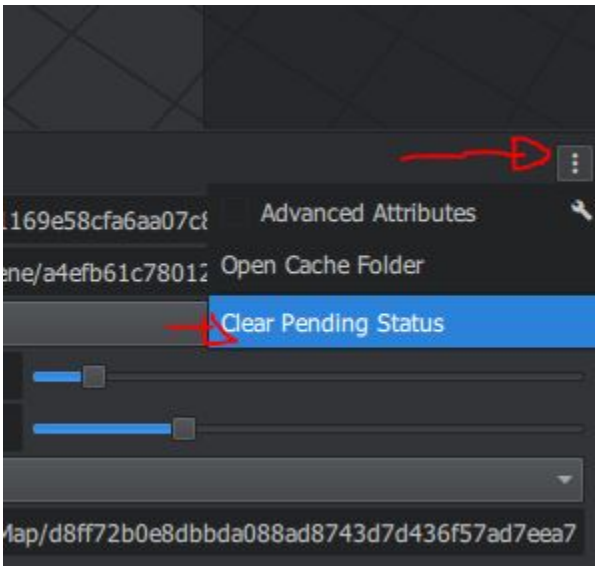
[85] https://github.com/alicevision/meshroom/issues/243
[86] https://github.com/alicevision/meshroom/issues/303
[87] https://github.com/alicevision/meshroom/issues/228#issuecomment-418329506



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.

10.4 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[88] to check for errors[89] 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 IrfranView `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[90]) 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[91])

10.5 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 thresholds 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[92] as examples. Contact us if you need more information to use it with a new renderfarm system.

10.6 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:

[88] https://www.coderslagoon.com/
[89] http://openpreservation.org/blog/2016/11/29/jpegvalidation/
[90] https://github.com/alicevision/meshroom/issues/149
[91] https://github.com/alicevision/meshroom/issues/209
[92] https://github.com/alicevision/meshroom/tree/develop/meshroom/submitters

```
+ 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).

10.7 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[93] #197[94] #203[95]

10.7.1 This Error message on a computer *without* NVIDIA GPU

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

#218[96] #260[97]

[Request] Remove CUDA dependency alicevision/#439[98]

> 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.*[99]

10.7.2 Can I use Meshroom without an NVIDIA GPU?

Yes, but you must use Draft Meshing[100] to complete the reconstruction.

[93] https://github.com/alicevision/meshroom/issues/182
[94] https://github.com/alicevision/meshroom/issues/197
[95] https://github.com/alicevision/meshroom/issues/203
[96] https://github.com/alicevision/meshroom/issues/218
[97] https://github.com/alicevision/meshroom/issues/260
[98] https://github.com/alicevision/AliceVision/issues/439
[99] https://github.com/alicevision/AliceVision/issues/439#issuecomment-403820801
[100] https://github.com/alicevision/meshroom/wiki/Draft-Meshing

10.7.3 Does my GPU support CUDA?

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

10.8 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.

10.8.1 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 `StuctureFromMotion` 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 `DescriberTypes` 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.

10.8.2 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.

10.8.3 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.

10.9 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 `DescriberType` 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).

10.10 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[101], so all formats supported by OpenImageIO can be imported to Meshroom. 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[102]) Convert your RAW image to '.jpg', '.jpeg', '.tif', '.tiff' or '.png' to resolve this problem.

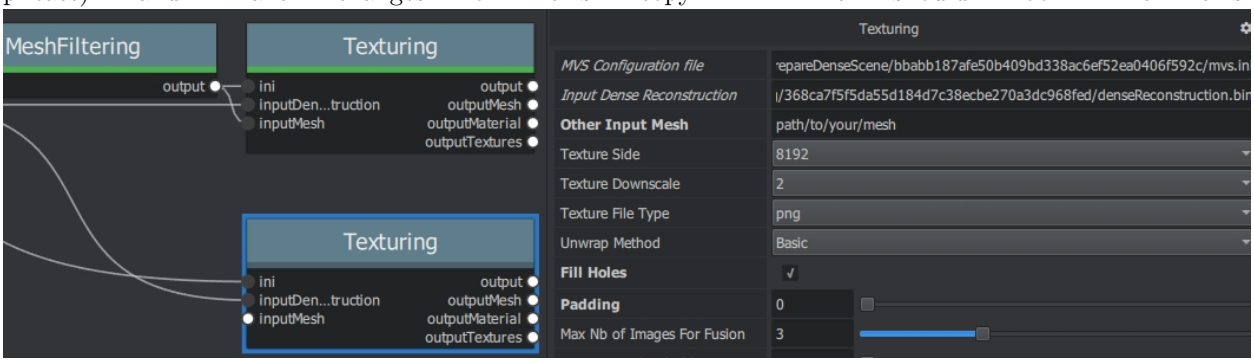
[101] https://sites.google.com/site/openimageio/home
[102] https://groups.google.com/forum/#!searchin/alicevision/raw|sort:date/alicevision/TzOcYo7tI9c/ihW70a9mCAAJ

10.11 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, your 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.

10.12 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[103])
- make sure your antivirus program does not interfere with Meshroom ((#178[104])/(#342[105]))
- 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[106], DepthMap and some photo datasets which worked in 2018.1 #409[107]. These problems will be addressed in the next release)
- Test Meshroom with the Monstree[108] dataset

[103] https://github.com/alicevision/meshroom/issues/209
[104] https://github.com/alicevision/meshroom/issues/178
[105] https://github.com/alicevision/meshroom/issues/342
[106] https://github.com/alicevision/meshroom/issues/449
[107] https://github.com/alicevision/meshroom/issues/409
[108] https://github.com/alicevision/dataset_monstree

- 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

CHAPTER

ELEVEN

REFERENCES

Bibliography

Text publications

. . .

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

https://medium.com/realities-io/getting-started-with-photogrammetry-d0a6ee40cb72

http://benvancitters.com/tag/photogrammetry/

Videos

Meshroom live reconstruction (LADIO project)

https://www.youtube.com/watch?v=DazLfZXU_Sk

Meshroom: Open Source 3D Reconstruction Software

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

Meshroom Introductory Project Tutorial

https://www.youtube.com/watch?v=bYzi5xYlYPU

Meshroom: Camera Sensor DB Error

https://www.youtube.com/watch?v=EOc4Utksk2U

How to 3D Photoscan your Face for Free!

https://www.youtube.com/watch?v=9Ul9aYhm7O4

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

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

CHAPTER

TWELVE

GLOSSARY

Alicevision AKAZE

CCTAG

SIFT

CHAPTER

THIRTEEN

ABOUT

13.1 About Meshroom

Meshroom is a free, open-source 3D Reconstruction Software based on the AliceVision[110] 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.

13.1.1 Project history

In 2010, the IMAGINE[111] research team (a joint research group between Ecole des Ponts ParisTech[112] and Centre Scientifique et Technique du Batiment[113]) and Mikros Image started a partnership around Pierre Moulon's thesis[114], supervised by Renaud Marlet and Pascal Monasse on the academic side and Benoit Maujean 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[115].

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[116] to create a Previz system. In 2017, CTU joined the team in the EU project LADIO[117] to create a central hub with structured access to all data generated on set.

[110] https://github.com/alicevision/AliceVision
[111] http://imagine.enpc.fr/
[112] http://www.enpc.fr/en
[113] http://international.cstb.fr/
[114] http://www.theses.fr/2014PEST10337
[115] https://en.wikipedia.org/wiki/Matte_painting
[116] http://www.alicevision.org/popart
[117] http://www.alicevision.org/ladio

13.1.2 Partners

Czech Technical University (CTU)[118] in Prague, Czech Republic

IMAGINE[119] from the Universite Paris Est, LIGM Gaspard-Monge, France

Institut National Polytechnique de Toulouse (INPT)[120], France

Mikros Image[121] , Post-Production Company in Paris, France

Simula Research Laboratory AS[122] in Oslo, Norway

Quine[123] in Oslo, Norway

See AliceVision Contributors[124] for the full list of contributors.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, see POPART[125], Project ID: 644874 and LADIO[126], project ID: 731970.

13.1.3 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.

[118] http://people.ciirc.cvut.cz/~pajdla
[119] http://imagine.enpc.fr/
[120] http://www.inp-toulouse.fr/
[121] http://www.mikrosimage.com/
[122] https://www.simula.no/
[123] http://www.quine.no/
[124] https://github.com/alicevision/AliceVision/blob/develop/CONTRIBUTORS.md
[125] http://www.alicevision.org/popart
[126] http://www.alicevision.org/ladio

13.2 About the manual

This manual is a compilation of the resources found on alicevision.github.io, information collected from github issues, other web resources and new content, created for this manual. This manual is work in progress.

You are welcome to comment and contribute. Check out https://github.com/alicevision/meshroom-manual/ for details.

13.3 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!

13.4 Contact us

You can contact us on the public mailing list at alicevision@googlegroups.com

You can also contact us privately at alicevision-team@googlegroups.com

13.5 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[127] . 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/

[127] https://help.github.com/articles/creating-a-pull-request

13.6 List of contributors

13.6.1 Meshroom manual

Github names, listed in alphabetical order:

- bmaujean
- bormm
- CaliLuke
- fabiencastan
- ChemicalXandco
- julianrendell
- natowi
- SBCV
- simogasp

13.7 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**[128] **.** This is a Meshroom community project.

Meshroom is released under MPLv2[129]

13.7.1 Third parties licenses

- **Python** https://www.python.org[130] Copyright (c) 2001-2018 Python Software Foundation Distributed under the PSFL V2[131] .
- **Qt/PySide2** https://www.qt.io[132] Copyright (C) 2018 The Qt Company Ltd and other contributors. Distributed under the LGPL V3[133] .
- **qmlAlembic** https://github.com/alicevision/qmlAlembic Copyright (c) 2018 AliceVision contributors. Distributed under the MPL2 license[134] .
- **QtOIIO** https://github.com/alicevision/QtOIIO Copyright (c) 2018 AliceVision contributors. Distributed under the MPL2 license[135] .

[128] http://creativecommons.org/licenses/by-sa/4.0/
[129] https://github.com/alicevision/meshroom/blob/develop/LICENSE-MPL2.md
[130] https://www.python.org/
[131] https://www.python.org/download/releases/2.7/license/
[132] https://www.qt.io/
[133] https://opensource.org/licenses/LGPL-3.0
[134] https://opensource.org/licenses/MPL-2.0
[135] https://opensource.org/licenses/MPL-2.0

CHAPTER

FOURTEEN

COPYRIGHT

[136] https://creativecommons.org/licenses/by-sa/4.0/

BIBLIOGRAPHY

[KSS11] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2969–2976. 06 2011. doi:10.1109/CVPR.2011.5995464[109].

[109] https://doi.org/10.1109/CVPR.2011.5995464

INDEX