UAV OS

VSfM,Meshlab, CMPMVS for 3D Scene Reconstruction

UAV OS Handout #01

This handout describes how VisualSfM can be used for 3D scene reconstruction from 2D drone Imagery. The output then can be fed to Meshlab for meshing and texturing which can result into full 3D textured model which can be scaled for measurement analysis or exported for further analysis. The output of VSfM can also be fed to CMPMVS, clustered based patched based multi view stereo, for generating orthomosaic, DEMs and video output of the data products.

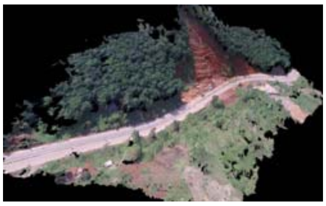
PS Singh

UAV OS 2019

UAV OS

HandOut #01 VisualSfM, CMPMVS and Meshlab for 3D Model Reconstructions

2D UAV Images



3D Dense Point Cloud

Full 3D Textured Model

Goals :

- Generate 3D Texture Model for Interactive Visualisation
- Meshing : 3D Point -> 3D Polygonal Meshes
- Cleaning/Deleting of 3D Point
- Rescale Model for Real World Measurement
- (optional) Use CMPMVS : DEM, Orthophoto

Dataset : 70-90 Images of Landslide Scar

PS Singh, NESAC

Background

Visual SfM (VSfM) is a GUI application for 3D reconstruction using structure from motion pipeline. It can be used for any 3D reconstruction application producing high accuracy results in short time. The simple GUI enables the user to control the stages of the reconstruction without using code. When needed, the reconstruction, matches, and key-points can be exported into a file and post-process with any programming language desired. VSfM runs very fast by exploiting multicore parallelism for feature detection, feature matching, and bundle adjustment.

VSfM is an easy and intuitive software for generating 3-D reconstruction. It executes state of the art SfM algorithm and produces high-quality reconstructions. It does not require programming knowledge, though it enables the user to export the model for further research, using any desired programming language.

One of the greatest advantages of VSfM is that we are able to export the reconstruction into a text file and continue to work on it. Using this data, we can deploy new models, examine the reconstruction and develop it. To do that we just press SfM->Extra Function->Save Current Model. This file contains first the location and names of all cameras, then 3D points and finally it contains the 2D correspondences

Meshlab is great for generating and managing mesh models. Further, it has tons of tools and algorithms for generating 3D texture model including other analysis such as measurements etc

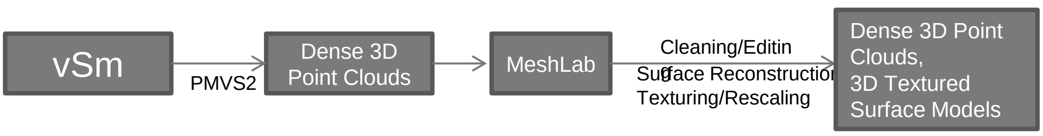
CMPMVS uses the output of VisualSfM to create other textured model. Also, it has powerful config files which can be authored for cerating useful products such as DEM and Orthophotos. Additionally, the CMPMVS can be used to generate video outputs of the products which is handy in checking the products region-wise. Note : The High-End Graphics enabled system with adequate RAMs is needed for running CMPMVS

Some Open Source Tools for UAV Data Processing

Tools	Process Outputs	System Requirements
VisualSfM+Meshlab	Dense 3D Point Clouds,3D Textured Surface Models	Windows/Linux (64 bit)
Python osm-bundler	Dense 3D Point Clouds	Windows/Linux (32/64 bit)
VisualSfM+CMPMVS	OrthoMosaic, DEM, 3D Textured Surface Models	Windows/Linux (64 bit)
OpenDroneMap	OrthoMosaic, Georeferenced Point Clouds, 3D Textured Surface Models	Linux (Ubuntu LTS 14 N above)
SfMToolkit3	OrthoMosaic, Dense 3D Point Clouds, DSM	Windows(32/64 bit)

1 VisualSfM, Meshlab

Dense 3D Point Clouds, 3D Textured Surface Models



VisualSFM

VisualSFM finds unique features in each image, matches the features and constructs sparse 3D point cloud, The model is then refined into a dense point cloud.

VisualSFM gives two main outputs -
.out file (bundler format) which stores the calculated (solved) cameras' positions and a sparse point cloud representing the points in the scene that the software used to determine the camera positions.

.ply file which stores a denser cloud of points, each with position, color and normal (a vector perpendicular to the surface the point is on) data.

Meshlab

Meshlab works on 3D mesh or a point cloud input files

Meshlab produces textured, clean, high-res mesh.
Meshlab also automatically calculates UV maps (the basis for 3D texturing) and builds a texture for us by estimating which projector is most accurate on each spot of the model. The outputs of this step are:

.obj file of a mesh (with UVs) that can be easily interchanged with various 3D softwares
.png file of varying size representing the texture of the mesh

VisualFSM

Example Scene :

Landslide Scar, Umling, Ribhoi

90 Images

Few Points to get the best possible reconstruction

1. **Parallax**: MOVE AROUND!. Move in a 360 degree circle around an object, taking snaps every 10-20 degrees, as well as at several heights
2. **Overlap**: minimum 75% overlap, 40% sidelap
3. **Photograph all angles :** to get everything
4. **Scale**: reconstructing entire environments is totally possible, but be prepared to take a lot more photographs. Smaller objects require fewer images and the process will take less time
5. **Minimize subject motion**: The software expects that everything in the scene is stationary when it analyzes the scene. if objects move around, the assumption that features are in the same space between photographs will be false, and VisualSFM will build a distorted model

Step 1: Import the images

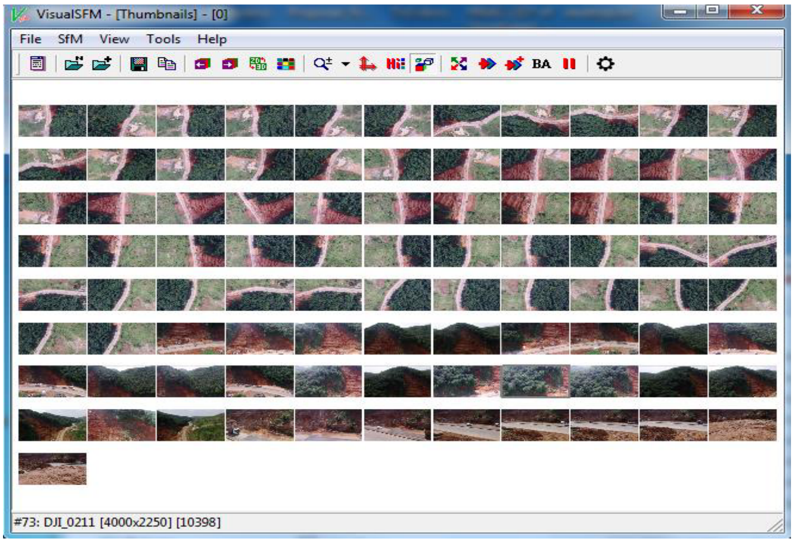
Note : **VisualSFM** will produce generally better results the higher-resolution your images are, **BUT**, after 3200px, the results get worse !

Before you import the images, make sure your images are no larger than 3200px tall or wide

Note : *Some experiments have been performed, that smaller images (<2k) actually improve feature detection and matching success. The process of matching features between images is done on the GPU. If you have high-end GPUs such as Tegra,...then you may try with better res imagery.*

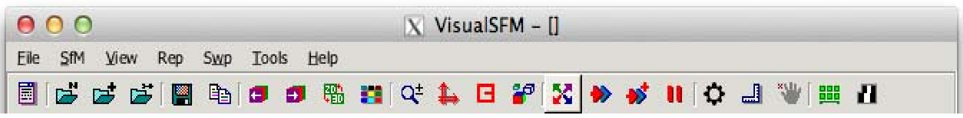
If you do have to resize your images, make sure to retain any EXIF metadata logged by the camera as VisualSFM can use it in order to pick more accurate focal length and sensor size settings. VisualSFM installs a FOSS program called jhead for reading EXIF data from the command line

Step 2: click the second folder icon to import your image sequence



Note : Enable the task, it's in Tools –> Show TaskViewer and gives logs of what VisualSFM is doing

Step 2: SIFT and match : Click on icon that looks like 4 arrows in an X shape on the toolbar.



Detection :"SIFTing" is where unique features of each image are detected and logged

(This process can accelerate on machines with better GPU)

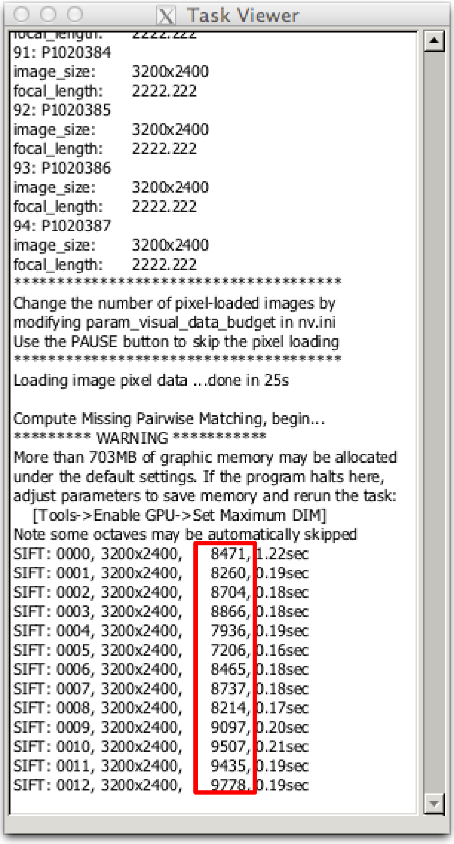
Matching : Features are matched between images

(amount of comparisons that have to be made is exponentially related to the number of images)

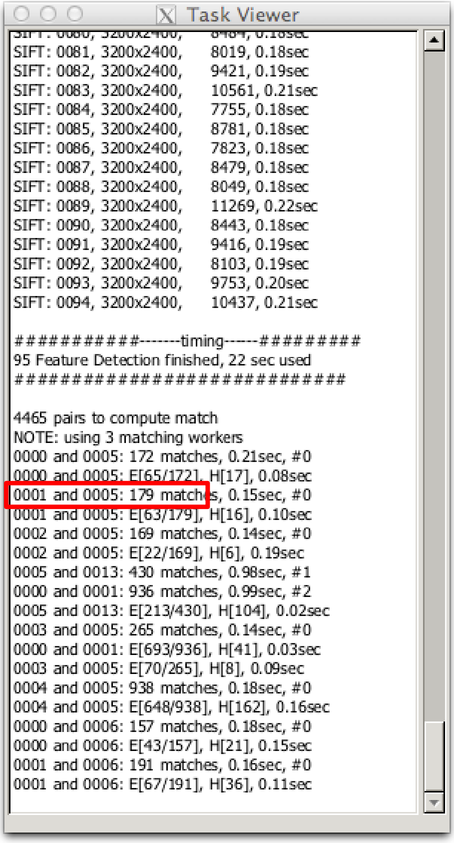
Neighbour comparison : comparison to only the n images on either side of each image taken with for a linear camera move.

Note : In case the process is interupted or crashing out/hanged, process resumes where it left off the next time you try to SIFT and match those images

Taskviewer Logs



Feature Detection



Feature Matching

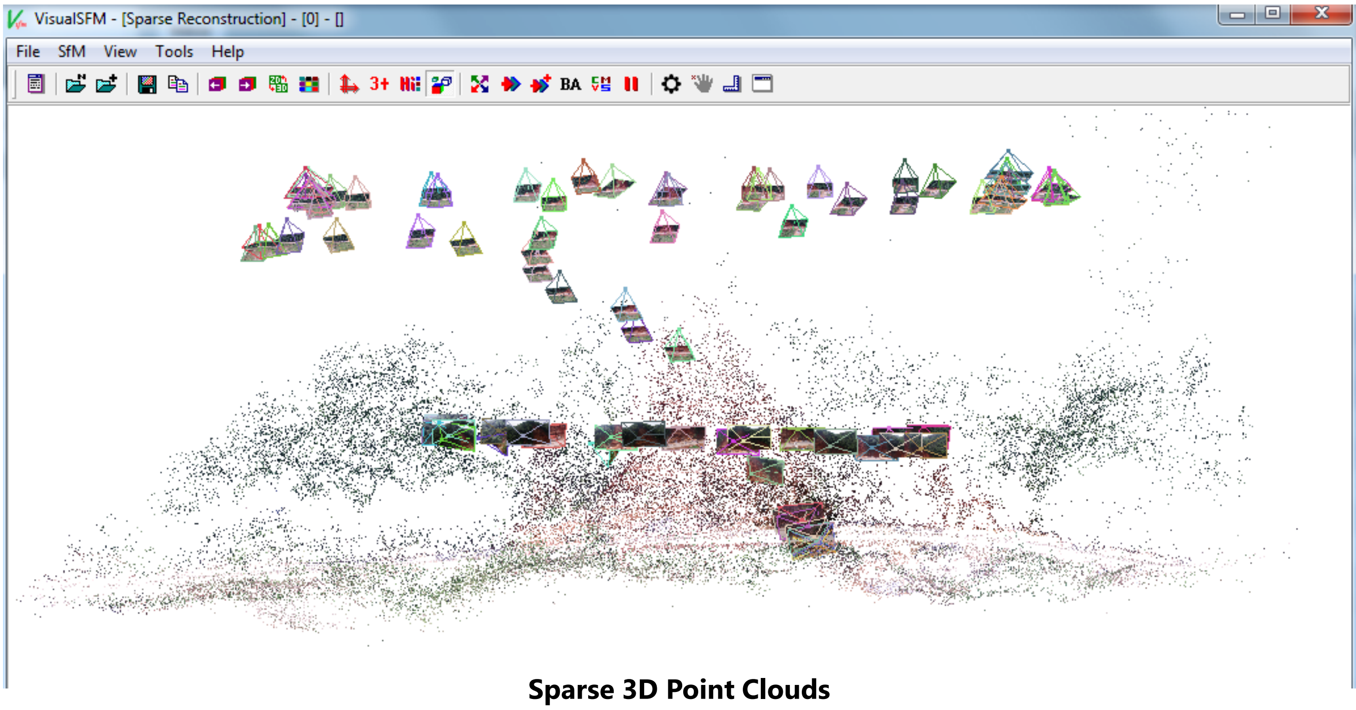
Step 3: Sparse 3D reconstruction : Click on button with the two arrows on it (>>) to start this process.

Once the overlapping features have been detected, you can begin a sparse 3D reconstruction

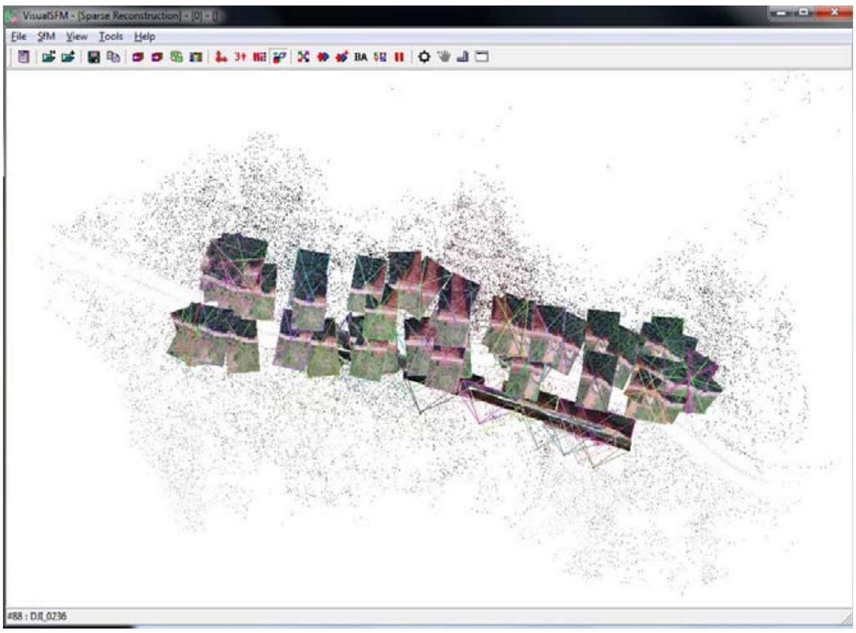
This process is interactive! : Pan, rotate and zoom through the 3D space as it tries to solve the feature overlaps into a 3D form!

See the position and rotation of the cameras that recorded the original images

ctrl+mousewheel up or down in order to scale the photos that are reconstructing the scene to get an idea of which photos are being places where



Sparse 3D Point Clouds



Adjusted the size of the projectors up a bit so you can how it's fitting each camera / image into the scene

This process is interactive in that you can influence the outcome of the solve by removing bad cameras

Removing bad cameras (Optional)

Method 1.

1.Turn off 3+
2.Click F2, select the bad 3D points
3.Click the 'hand' icon twice, it will delete the camera that sees the largest number of selected points
4.Go to step 1 if there are still bad 3D points
5.Click >>+ to grow the modified 3D model

Method 2.

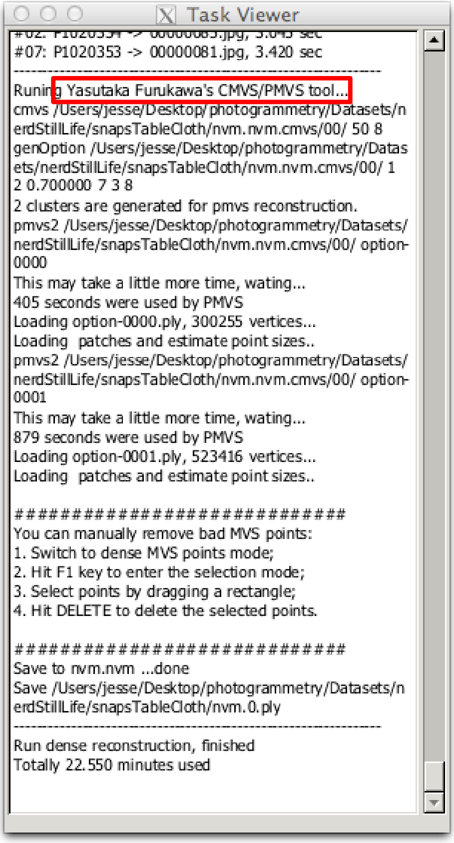
1. Click F1, draw a rectangle to select all the bad cameras (at most 250 cameras at a time)
2. Click the "hand" icon to remove all the bad cameras
3. Click >>+ to grow the modified 3D model.

Step 4: Dense reconstruction : Click on the "cmvs" button, and save the cmvs folder structure

Takes Sparse Point cloud and computes a much denser version of the point cloud

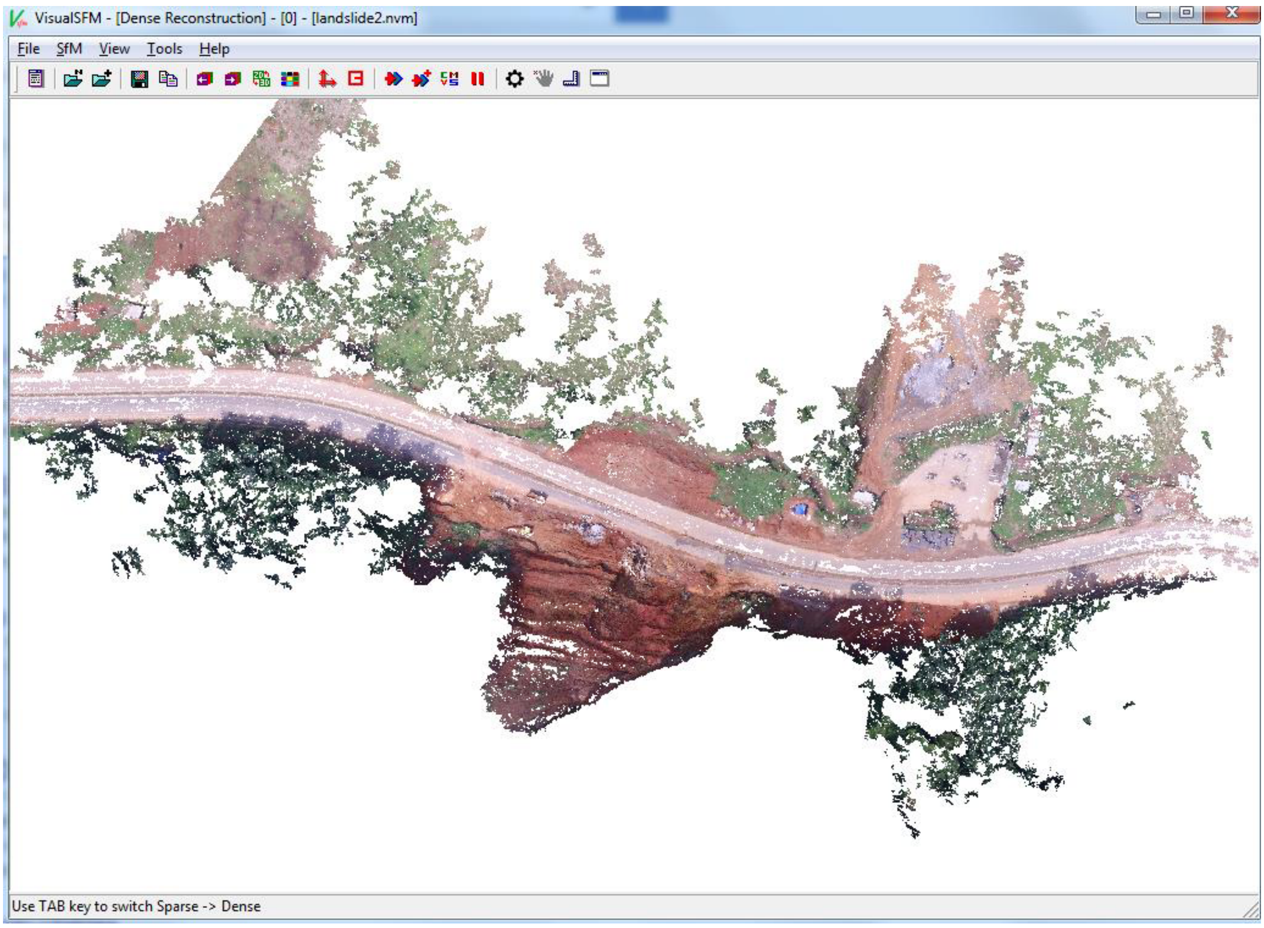
VisualSFM will export its internal sparse reconstruction to that folder in the format needed by cmvs / pmvs2 and that software is then run in order to populate that folder structure with the dense point cloud

Taskviewer Logs



Once the dense reconstruction is complete, Click tab to see a visualization within VisualSFM of 3D Dense Point Cloud

14



Click on 'F' to hide/show camera position

Note : you can also perform a dense reconstruction with cmp-mvs

15

VisualSFM Process Conclusion

1. Debugging – Remove bad points / cameras that have been added in the wrong position / orientation in order to improve
2. The SIFT and matching process is stored on disk

   .sift and .mat file for each image

Meshlab

"**MeshLab** is an advanced 3D mesh processing software system that is oriented to the management and processing of unstructured large meshes and provides a set of tools for editing, cleaning, healing, inspecting, rendering, and converting these kinds of meshes. MeshLab is free and open-source software, subject to the requirements of the GNU General Public License (GPL), version 2 or later, and is used as both a complete package and a library powering other software. It is well known in the more technical fields of 3D development and data handling "

Input : Meshlab operates on mesh and point-cloud data

Caution :

1. it doesn't have undo button/action!

Changes are only made to a mesh if you "export" it.

If you want to go back, you can "reload" the current mesh you're editing

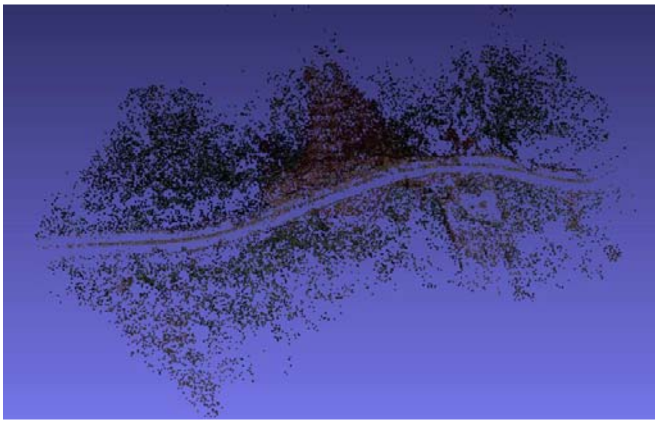
Note : Be careful and save your meshlab project AND export your mesh ("export as…" in order to save a new file) frequently

Step 1. Open the bundle.rd.out file
(to import the bundle.rd.out file to get the camera positions and rasters)

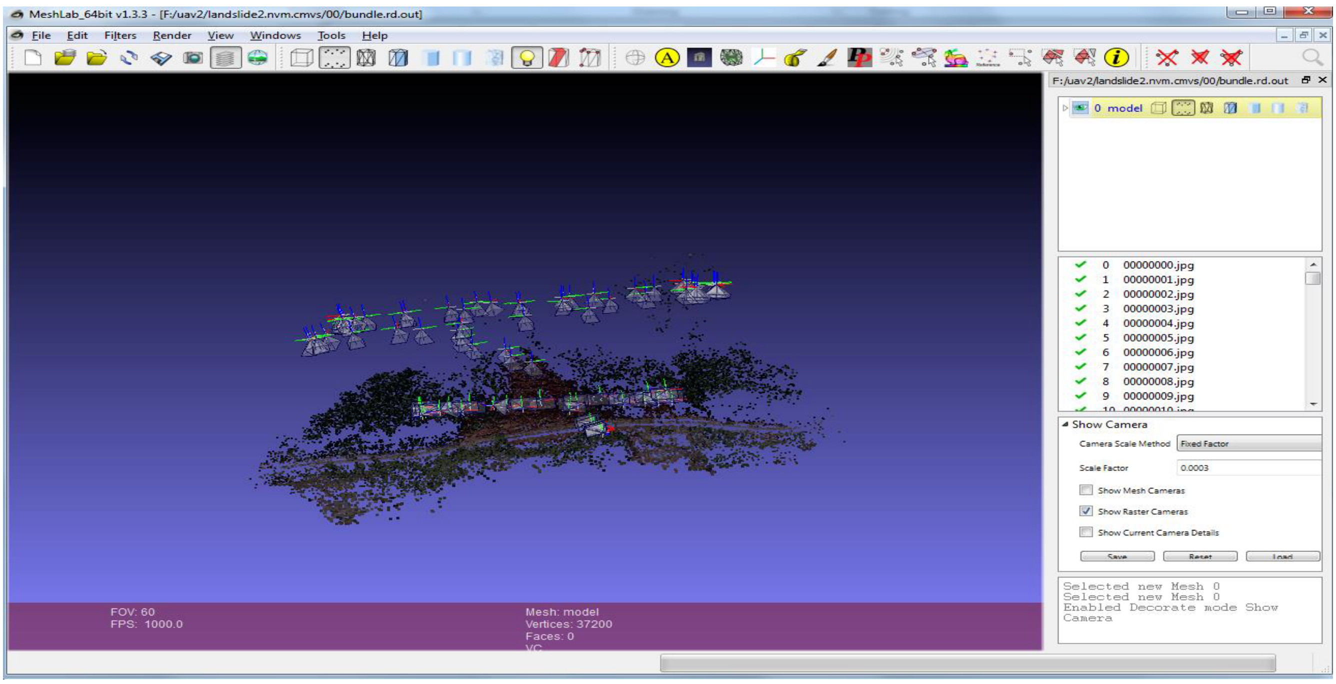
bundle.rd.out stores internal representation of the sparse reconstruction

Go to File –> Open Project, and navigate to your bundle.rd.out file within the nvm directory

Meshlab will ask you as soon as you select the bundle.rd.out file to select the camera file. That's in a file in the same directory called, "list.txt", so select that next



The above steps will import the cameras from your solve into the scene as well as the photos they're associated with (they're called raster layers in Meshlab). This is important for the texturing process later on



click on the layers icon on the toolbar to open up the layers palette. You should see your geometry layers on the top right, and the raster layers on the bottom right of the screen.

Go to Render –> Show Camera. This will display the camera frustums from all of your projectors in the viewer. The default scale of the camera visualization is huge compared to the size of our mesh, so click on the disclosure triangle next to "Show Camera" that has appeared on our layers sidebar:
change the scale factor to something like 0.001. Keep decreasing it until you can see both your mesh and the camera positions around your mesh.

Disable the camera visualization by unchecking Render –> Show Camera.

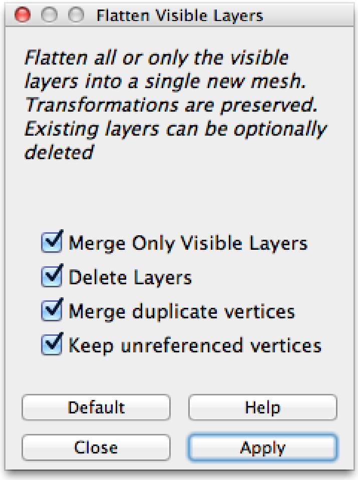
Step 2. Replace sparse cloud with the dense one

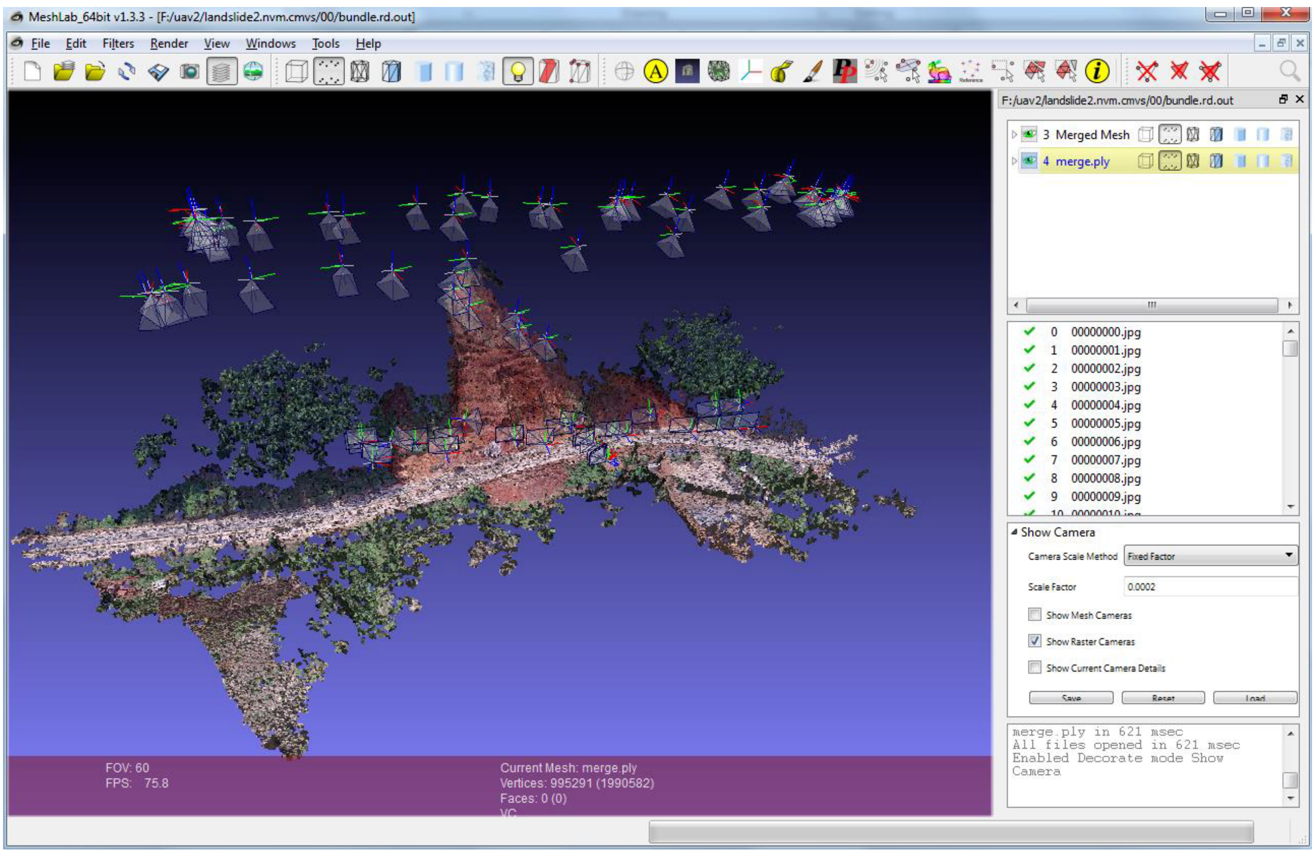
1. In the layers palate, right click on the mesh layer, and select “delete current mesh” in order to remove it entirely.
2. select File –> Import Mesh and load the dense mesh. It should be located in your NVM_FOLDER/00/models/option-0000.ply

Note : It’s possible that VisualSFM (really pmvs2) generated multiple .ply files (called option-0000.ply, option-0001.ply etc). Because the dense mesher only works with 50 cameras at a time.
 If you have multiple .ply files, you’ll need to flatten them into one mesh before you can proceed.

Simply right click on any of the mesh layers, and select “Flatter Visible Layers”. Make sure to check “Keep unreferenced vertices”:

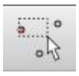
Save as the merge ply file and import it afresh





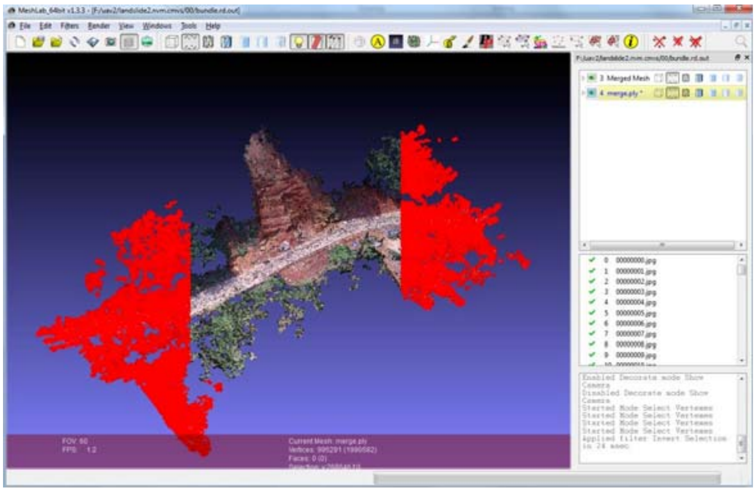
Dense point cloud AND the camera positions / raster layers!

Step 3. Clean up dense 3d points

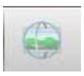


box selection tool : select regions of points that are bad

delete points button



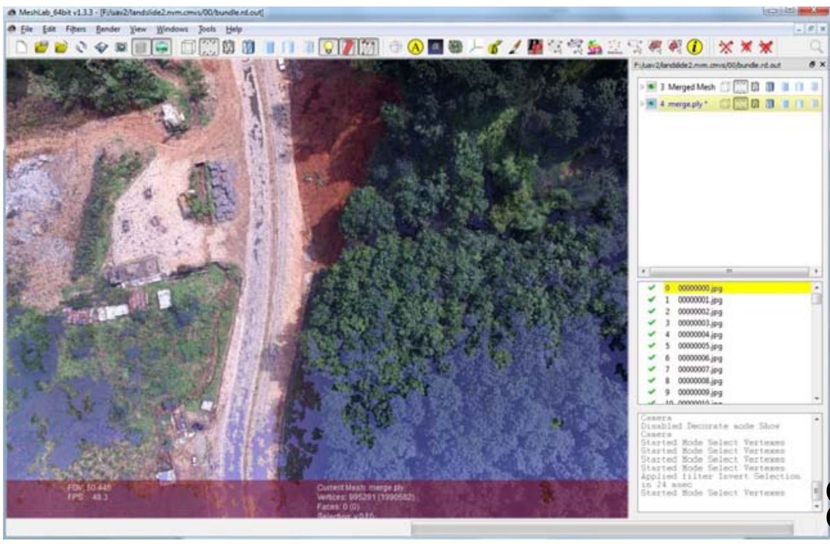
Step 4. Check the Cameras



Click on "Show current raster mode" button

The 3D camera will snap to the position where VisualSFM thought that image was taken from in 3d space! You can scroll your mousewheel to alter the transparency of your image and double check the alignment!

After, you've checked your cameras, disable the "Show current raster mode" by hitting the icon again.



Step 5. Meshing

Convert point cloud into an actual polygonal mesh

Several algorithms for doing this meshing step, but the best seems to the Poisson Surface Reconstruction algorithm by *Kazhdan, Bolitho, and Hoppe*

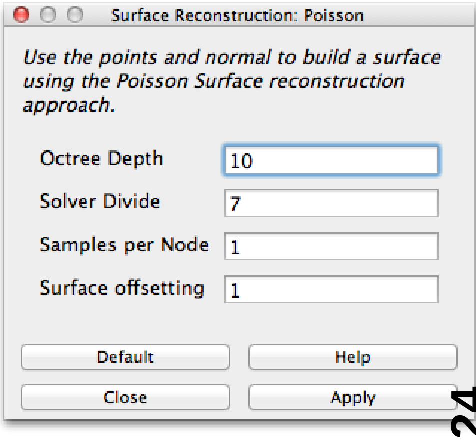
The algorithm is Open source : See the source code at
http://www.cs.jhu.edu/~misha/Code/PoissonRecon/Version6/

Select Filter –> Points –> Surface Reconstruction: Poisson

Spend some time and experiment here,

Keep increasing the Octree Depth till you're getting meshes that have more or less the same level of detail.

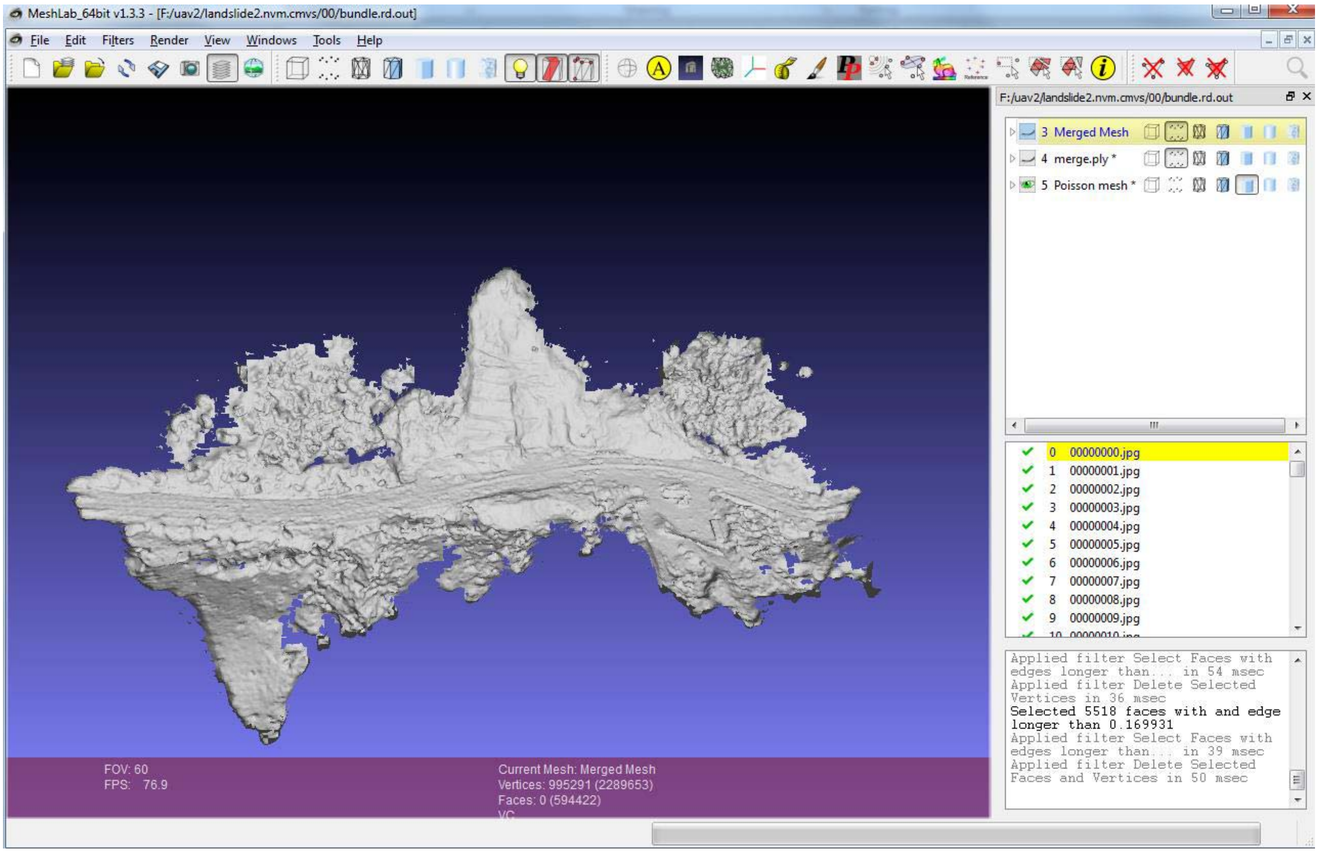
Note : Each time you click on "apply" in the Poisson Surface Reconstruction filter, meshlab will create a new layer with your mesh in it. Instead of deleting the ones that are bad, keep them around and maybe even rename them with the settings you chose



Caution : Meshing

- Octree Depth seems to have the most control over the detail in the mesh. The larger that number gets, the slower things seem to go / more memory you need, so be careful
- The Solver Divide seems to help use less memory when the Octree depth increases
- "Samples per Node" setting seems to smooth the mesh out if it has a lot of errors in it
  (Leave it at 1-5 to start with and increase up to 20 to try to smooth your mesh out a bit if there are problems)

The Poisson Mesher always tries to return "watertight" meshes which means that often it will return a mesh-ball with your object /environment inside. In order to excavate your mesh, you'll need to spend some time deleting faces. You can try "Filters –> Selection –> Select faces with edges longer than"



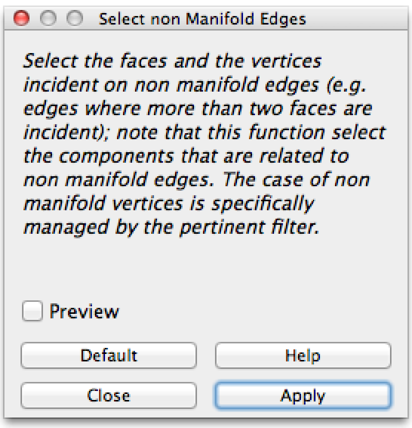
Save both mesh and project

Step 6. Fix manifold edges

After meshing, there are non-manifold edges which need to be removed before we can proceed.

Technically, non-manifold geometry is "*any edge shared by more than two faces.*" Non-technically, non-manifold geometry is bad geo that that certain algorithms can't deal with

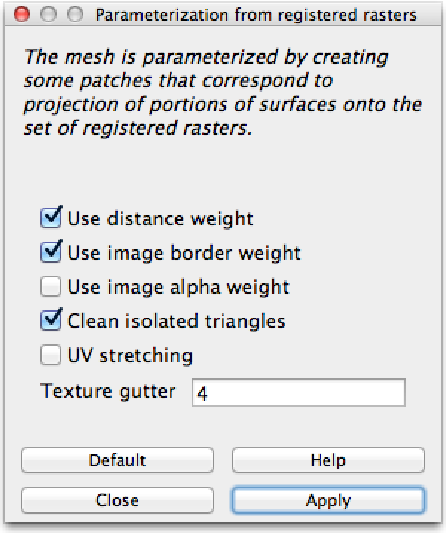
Filters –> Selection –> Select Non-Manifold edges, hit Apply, then delete them with the delete points button.



Step 7. Parameterization

Automatically builds a UV map by figuring out which projector cameras have the best view of each face of the model

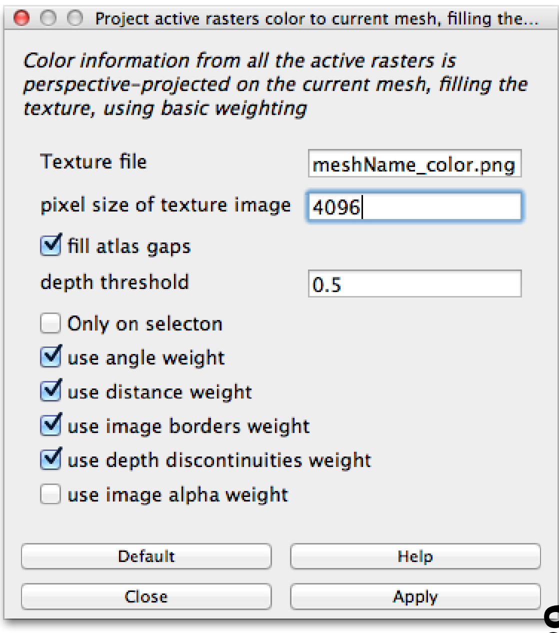
Filter –> Texture –> Parameterization from registered rasters, and match the following settings



Save (both the project, and the mesh).

Step 8. Project textures

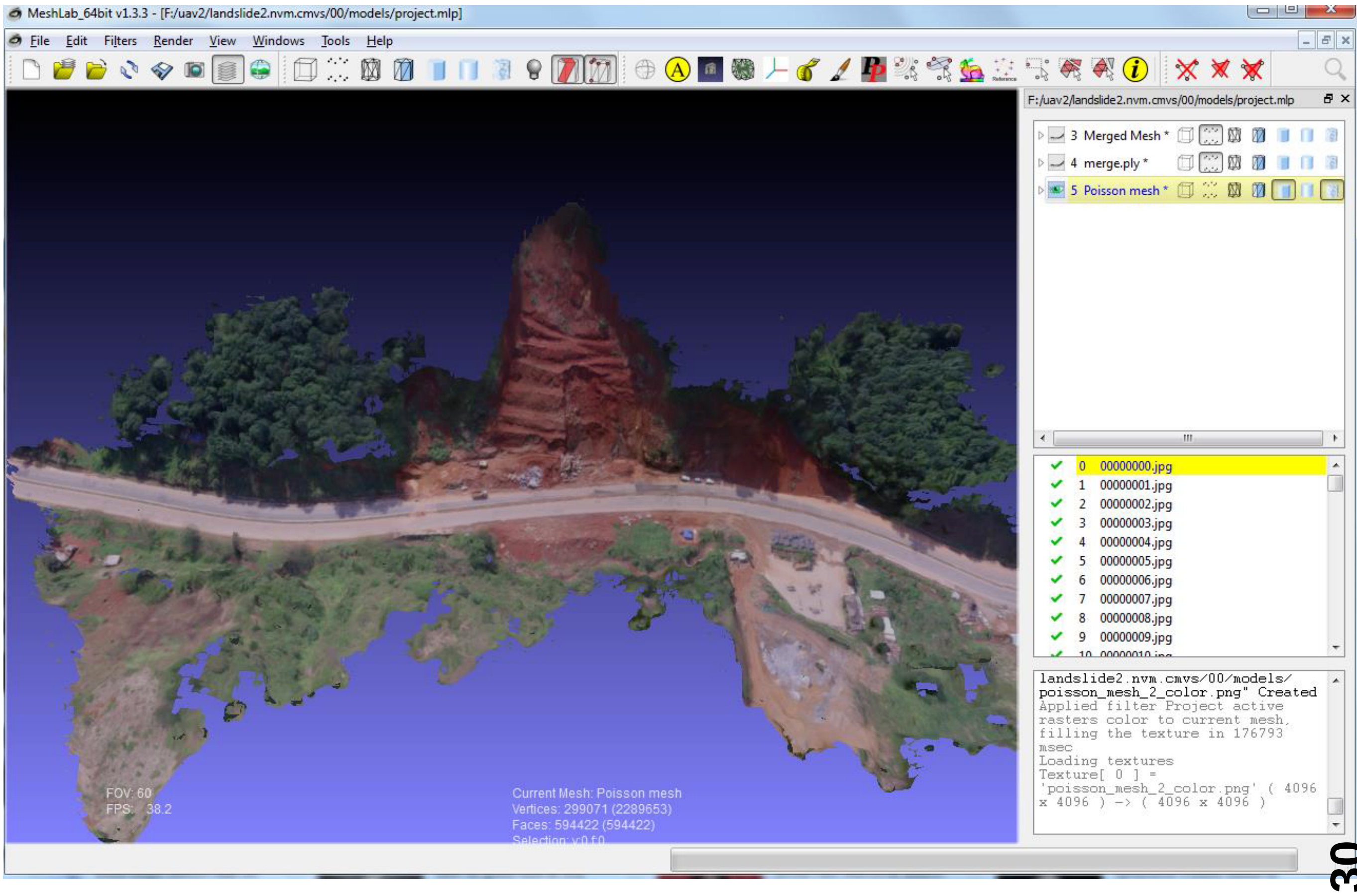
Project the texture from the projectors and make a texture map. Filter –> Texture –> Project active rasters color to current mesh, filling the texture, and match the following settings.



The texture filename you choose will be the name of the texture image that gets written out. The pixel size can be any power of 2 greater than, I'd say, 512. 512 / 1024 / 2048 / 4096 / 8192...the sky (really your computer) is the limit.

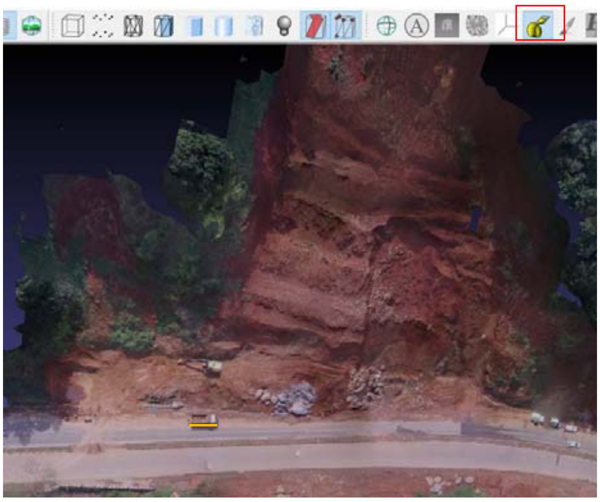
Shortcut: Parameterization + texturing from registered rasters

The Textured 3D Model!



Meshlab : Rescale the entire 3D model for Measurement

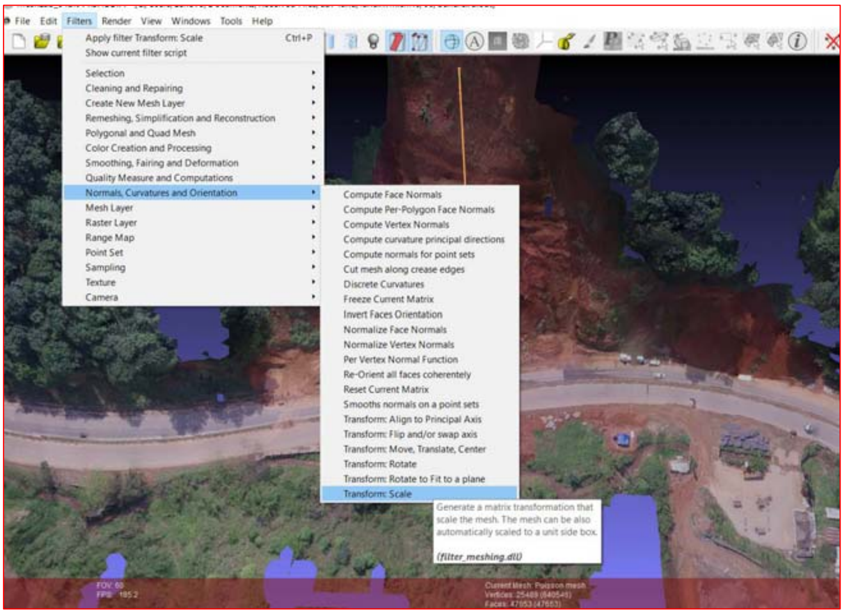
Measurement Tool

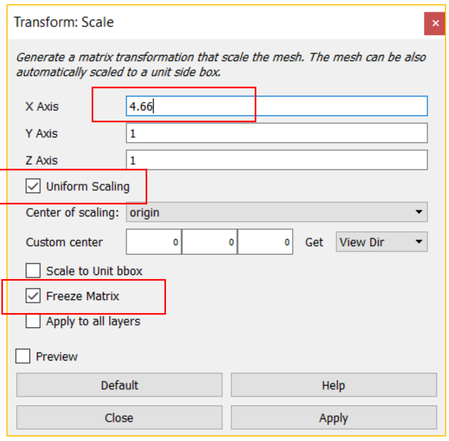


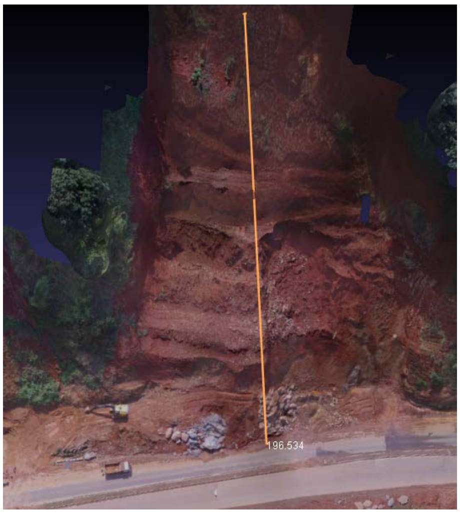
Assume that, you know actual physical length of an object in the scene, say the length of the truck (as shown on the scene) say 13m

Step1 : Use Measuring Tool and measure the length of the truck on the scene. Suppose the length is 0.0013 (as per current model)

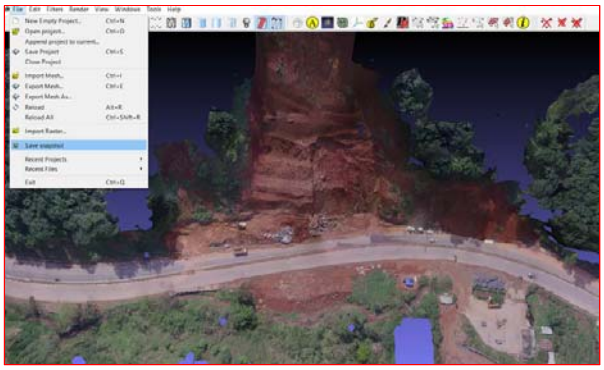
Step2 : Calculate the transformation ratio = Actual length of the truck / length of the truck as per the model (0.0013)







Take Snapshots of your 3D Model!

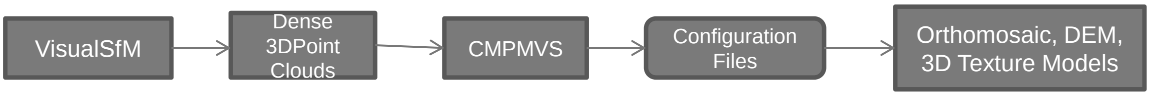


Start measurement of other objects on the scene!

VisualSfM+CMPMVS

OrthoMosaic, DEM, 3D Textured Surface Models

Process Flow



Issues : It gives DEM data as **DEM.png** which 16-bit data and needs to be converted to 32 bit (float) .tiff where each pixels will represent the pixels value as altitude in meters

CMPMVS is contains computer vision algorithms and run using libraries from OpenCV

DEM

Orthophoto

VisualSfM+CMPMVS

OrthoMosaic, DEM, 3D Textured Surface Models

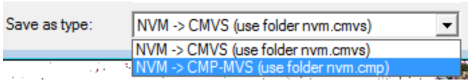
- Install and Setup visualSfM as usual
- Run VisualSfm till 3D sparse reconstruction
- In VisualSfM, save the dense 3D reconstruction output to folder of type
  - NVM.CMP
- Install CMPMVS version 0.6
  - Configure mvs.ini (output from VisualSfM) according to type of products that needs to be generated
  - Eg. For DEM, set `doComputeDEM=TRUE` ; For, Orthomosaic, set `doComputeOrthomosaic=TRUE`
  - Once .ini file has been configured
    - Open command prompt, Go to CMPMVS installation path
    - Execute >>cmpmvs "path to .ini"

VSfM + CMPMVS

The CMPMVS will takes the input dense 3D model generated by VSfM, and do mesh reconstruction and also perform other useful products such as DEM and Orthophotos.

1. Repeat to the first three steps in VSfM

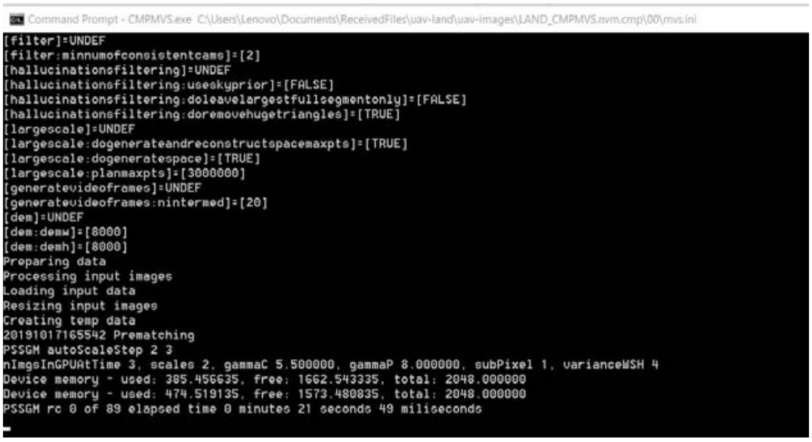
2. Click on the  button. In the popped up window, look for ‘Save as type:’, then select the second option, i.e. ‘NVM -> CMP-MVS’. Name your file and click ‘Save’. This process may take a few seconds.



3. Browse to the newly exported .nvm.cmp folder and open the mvs.ini file with a text editor (like notepad). In the [global] section, ensure that the directory in the line dirName="<path_to_your_folder>.nvm.cmp\00\data\" is correct for the machine you will be running CMPMVS on, including the trailing slash ("\"). **If you change machines, you will need to change this to the absolute path of the 00\data\ folder.**
4. Run CMPMVS from the command line as shown below.
   >cmpmvs. exe <path to mvs.ini>

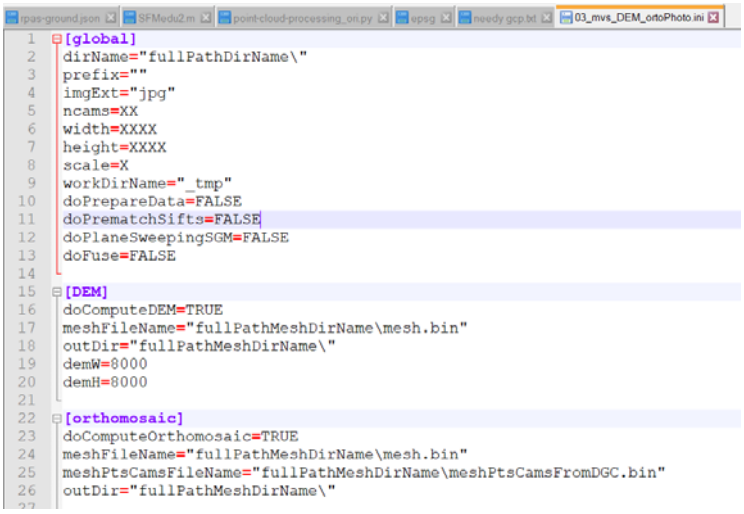
   In command line, change directory (type ‘cd’ followed by a space) to the folder where CMPMVS executable is installed/located. To run the application, type CMPMVS.exe, followed by a space, followed by the directory of the mvs.ini file. To find the directory of the mvs.ini file, browse to the .nvm.cmp folder (which is newly exported from VisualSFM steps) > 00, copy the route and paste into the command line, and type the name of the mvs.ini file. Hit Enter to run with default settings.

The command line example:
C:\user\...\cmpmvs\CMPMVS_6>CMPMVS.exe E:\...\00\mvs.ini
Note that there should be no space in the path to mvs.ini file. Otherwise error occurs. The mvs.ini file must be modified as per our requirement (the kinds of products we wish to generate). The output of the successful command must show something like this :



```
Command Prompt - CMPMVS.exe  C:\Users\Lenovo\Documents\ReceivedFiles\uav-land\uav-images\LAND_CMPMVS.nvm.cmp\00\mvs.ini
[filter]=UNDEF
[filter:minnumofconsistentcams]=[2]
[hallucinationsfiltering]=UNDEF
[hallucinationsfiltering:useskyprior]=[FALSE]
[hallucinationsfiltering:doleavelargestfullsegmentonly]=[FALSE]
[hallucinationsfiltering:doremovehugetriangles]=[TRUE]
[largescale]=UNDEF
[largescale:dogenerateandreconstructspacemaxpts]=[TRUE]
[largescale:dogeneratespace]=[TRUE]
[largescale:planmaxpts]=[3000000]
[generatevideoframes]=UNDEF
[generatevideoframes:nintermed]=[20]
[dem]=UNDEF
[dem:demw]=[8000]
[dem:demh]=[8000]
Preparing data
Processing input images
Loading input data
Resizing input images
Creating temp data
20191017165542 Prematching
PSSGM autoScaleStep 2 3
nImgsInGPUAtTime 3, scales 2, gammaC 5.500000, gammaP 8.000000, subPixel 1, varianceWSH 4
Device memory - used: 385.456635, free: 1662.543335, total: 2048.000000
Device memory - used: 474.519135, free: 1573.480835, total: 2048.000000
PSSGM rc 0 of 89 elapsed time 0 minutes 21 seconds 49 miliseconds
```

You can also add more options to change parameters. Based on our requirement, we must set out .ini file. Pl see the sample .ini files located within ini folders. For DEM, set doComputeDEM=TRUE ; For, Orthomosaic, set doComputeOrthomosaic=TRUE



```
[global]
dirName="fullPathDirName\"
prefix=""
imgExt="jpg"
ncams=XX
width=XXXX
height=XXXX
scale=X
workDirName="_tmp"
doPrepareData=FALSE
doPrematchSifts=FALSE
doPlaneSweepingSGM=FALSE
doFuse=FALSE

[DEM]
doComputeDEM=TRUE
meshFileName="fullPathMeshDirName\mesh.bin"
outDir="fullPathMeshDirName\"
demW=8000
demH=8000

[orthomosaic]
doComputeOrthomosaic=TRUE
meshFileName="fullPathMeshDirName\mesh.bin"
meshPtsCamsFileName="fullPathMeshDirName\meshPtsCamsFromDGC.bin"
outDir="fullPathMeshDirName\"
```

The command line below will create the output product along with the video also:
C:\user\...\cmpmvs\CMPMVS_6>CMPMVS.exe E:\...\00\mvs.ini DoGenerateVideo=True.

CMPMVS process may run for several hours. When it's finished, the command line status should be like:
C:\user\...\cmpmvs\CMPMVS_6>_
The reconstruction results are saved in the ~\00\data_OUT\simplified10 folder. The colored mesh is named 'meshAvImgCol.ply'. The Textured mesh is named 'meshAvImgTex.wrl'. In case you want a denser reconstruction, go to the \00\data_OUT folder to find the original files with same names. Now, You can use Meshlab or CloudCompare for further analysis.

Errors Scenarios and possible for running CMPMVS:

1. Missing DirectX dll files :
    a. Update DirectX for your OS
2. CMPMVS crash Error r128 -- Absolute path issue
    a. n the 00 folder, open the mvs.ini file with wordpad or any text editor. Note the dirName. and check whether it is the correct directory you typed (copied) in to the cmd. Correct it to be the same as it's current directory. Save it.
3. CMPMVS Errors occur during solving maxflow
    a. This may because you have too many images, or not enough GPU capacity. Our biggest dataset so far is a set of 630 images, size 1280x960, with modified settings. And the images have not too many feature points.
    fix it by
    1. Set:
    doPrepareData=FALSE
    doPrematchSifts=FALSE
    doPlaneSweepingSGM=FALSE
    doFuse=FALSE
    in your ini file in order to skip some previous computation. Don't delete the _tmp directory.
    2. Close all other applications and run it again. Don't open any other application during the computation.
    3. Change planMaxPts from 3000000 to 2000000 and run it again. Don't open any other application during the computation.

    If crash with modified settings, change:
    doPrepareData=TRUE
    doPrematchSifts=TRUE