AN INTRODUCTION TO 3D RECONSTRUCTION
using VisualSFM and PMVS2/CMVS

16 February 2013

Kumpee Teeravech, PhD Student

Remote Sensing and Geographic Information Systems
School of Engineering and Technology. Asian Institute of Technology.

1. Introduction

1.1. About VisualSFM

VisualSFM is a GUI application for 3D reconstruction using structure from motion (SFM). The reconstruction system integrates and improves upon several of my previous projects: A course project to implement a PhotoTourism-like incremental SFM system, SIFT on GPU(SiftGPU), and the recent Multicore Bundle Adjustment. VisualSFM is able to run very fast by exploiting multicore parallelism in feature detection, feature matching, and bundle adjustment. In addition, VisualSFM provides the interfaces to run Yasutaka Furukawa's PMVS/CMVS tool and to prepare data for Michal Jancosek's CMP-MVS. This software also comes with two research projects: Schematic Surface Reconstruction and Single View Repetition Analysis. (source: http://homes.cs.washington.edu/~ccwu/vsfm/).

1.2. About PMVS2/CMVS

PMVS is a multi-view stereo software that takes a set of images and camera parameters, then reconstructs 3D structure of an object or a scene visible in the images. Only rigid structure is reconstructed, in other words, the software automatically ignores non-rigid objects such as pedestrians in front of a building. The software outputs a set of oriented points instead of a polygonal (or a mesh) model, where both the 3D coordinate and the surface normal are estimated at each oriented point.

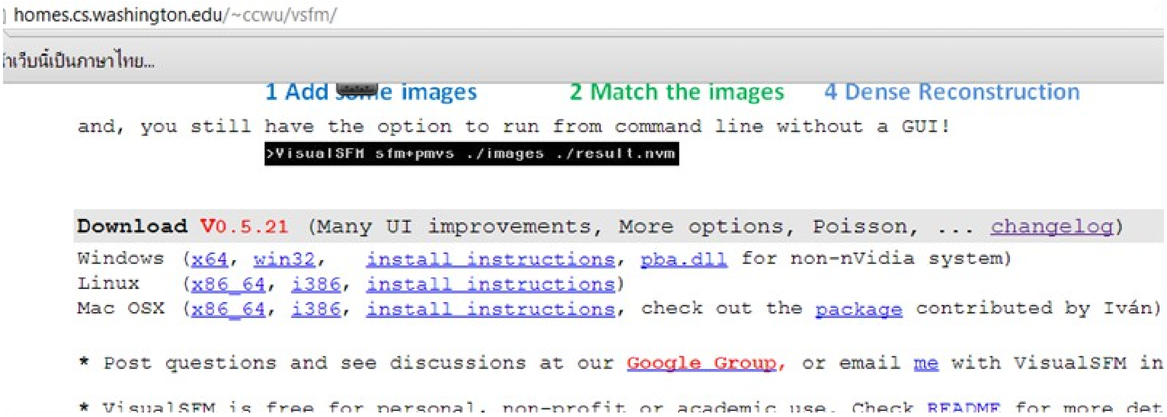
Many multi-view stereo (MVS) algorithms do not scale well to a large number of input images (lack of computational and memory resources). This software (CMVS) takes the output of a structure-from-motion (SfM) software as input, then decomposes the input images into a set of image clusters of managable size. An MVS software can be used to process each cluster independently and in parallel, where the union of reconstructions from all the clusters should not miss any details that can be otherwise obtained from the whole image set. CMVS should be used in conjunction with an SfM software Bundler and an MVS software PMVS2 (PMVS version 2).

Multi-view stereo software PMVS2, which is included in the CMVS package, was developped when Yasutaka Furukawa was a graduate student at University of Illinois at Urbana-Champaign under the supervision of Prof. Jean Ponce, who was affiliated with University of Illinois at Urbana-Champaign and Ecole Normale Supérieure. Further modifications and enhancements were added when Yasutaka Furukawa was a postdoc at University of Washington. (source: http://www.di.ens.fr/pmvs/).

2. Installations

2.1. VisualSFM

- Goto http://homes.cs.washington.edu/~ccwu/vsfm/
- Current version is 0.5.21
- Choose your operating system



homes.cs.washington.edu/~ccwu/vsfm/

ำเว็บนี้เป็นภาษาไทย...

and, you still have the option to run from command line without a GUI!

```
>VisualSFM sfm+pmvs ./images ./result.nvm
```

Download V0.5.21 (Many UI improvements, More options, Poisson, ... changelog)
Windows (x64, win32, install instructions, pba.dll for non-nVidia system)
Linux (x86_64, i386, install instructions)
Mac OSX (x86_64, i386, install instructions, check out the package contributed by Iván)

* Post questions and see discussions at our Google Group, or email me with VisualSFM in

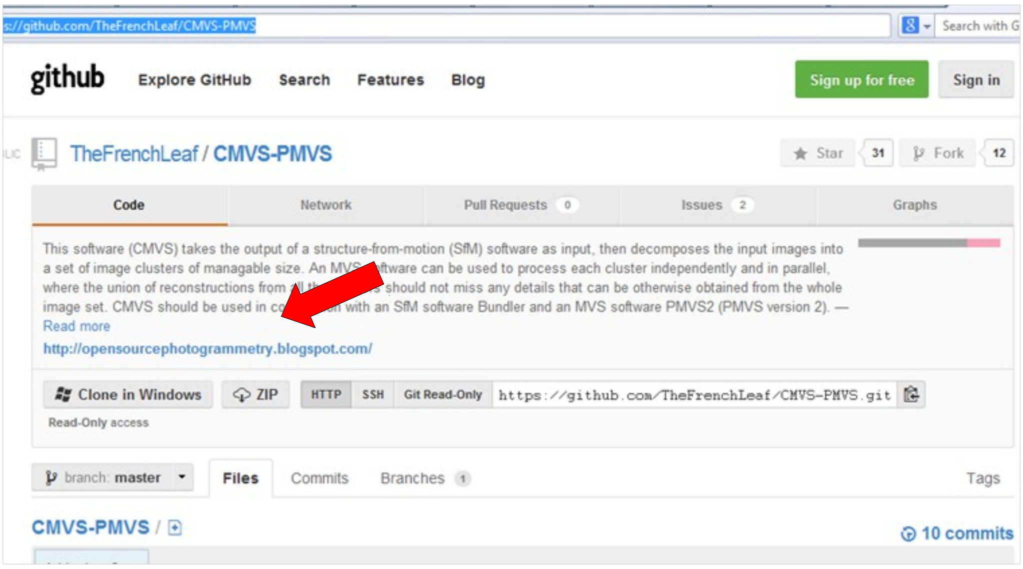
* VisualSFM is free for personal, non-profit or academic use. Check README for more det

Figure 1 Downloaing VisualSFM.

- Unzip to your desired directory

2.2. PMVS2/CMVS

- Goto https://github.com/TheFrenchLeaf/CMVS-PMVS/



s://github.com/TheFrenchLeaf/CMVS-PMVS

TheFrenchLeaf / CMVS-PMVS

This software (CMVS) takes the output of a structure-from-motion (SfM) software as input, then decomposes the input images into a set of image clusters of managable size. An MVS software can be used to process each cluster independently and in parallel, where the union of reconstructions from all the clusters should not miss any details that can be otherwise obtained from the whole image set. CMVS should be used in conjunction with an SfM software Bundler and an MVS software PMVS2 (PMVS version 2). — Read more

http://opensourcephotogrammetry.blogspot.com/

https://github.com/TheFrenchLeaf/CMVS-PMVS.git

Read-Only access

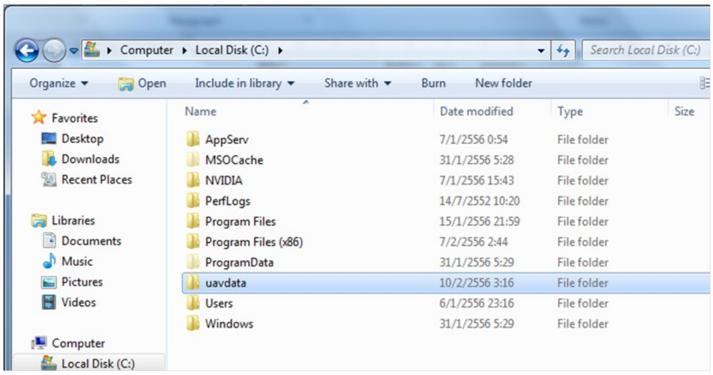
Figure 2 Downloaing PMVS2/CMVS.

- Press 'ZIP' button to compress and download the software.

- Open the compressed file, copy the following files to your VisualSFM's directory.
    - cmvs.exe
    - genOption.exes
    - pmvs2.exe
    - pthread.dll

3. Data Preparations

3.1. Create a working directory, i.e. C:\uavdata, we will call this directory as ROOT.



Name	Date modified	Type	Size
AppServ	7/1/2556 0:54	File folder	
MSOCache	31/1/2556 5:28	File folder	
NVIDIA	7/1/2556 15:43	File folder	
PerfLogs	14/7/2552 10:20	File folder	
Program Files	15/1/2556 21:59	File folder	
Program Files (x86)	7/2/2556 2:44	File folder	
ProgramData	31/1/2556 5:29	File folder	
uavdata	10/2/2556 3:16	File folder	
Users	6/1/2556 23:16	File folder	
Windows	31/1/2556 5:29	File folder	

Figure 3 Creating working directory.

- Put all of your images inside the ROOT directory.
- All images should be saved in JPEG format.
- For testing, you may resize the images to smaller dimensions, i.e. 800x600. This may decrease the number of correspondences; but it will run much more faster than using the full resolutions.

4. Sparse Reconstruction

- Start the application software by double clicking at VisualSFM_win32.exe icon.

- IMPORTANT: By default, the software will run its feature detection and matching algorithms in parallale using the multi-threading technique. However, this may crashs in some machine. Therefore, you may need to disable this option by going to `SfM->Pairwise Matching` and uncheck the `Multi-threaded Match` option.

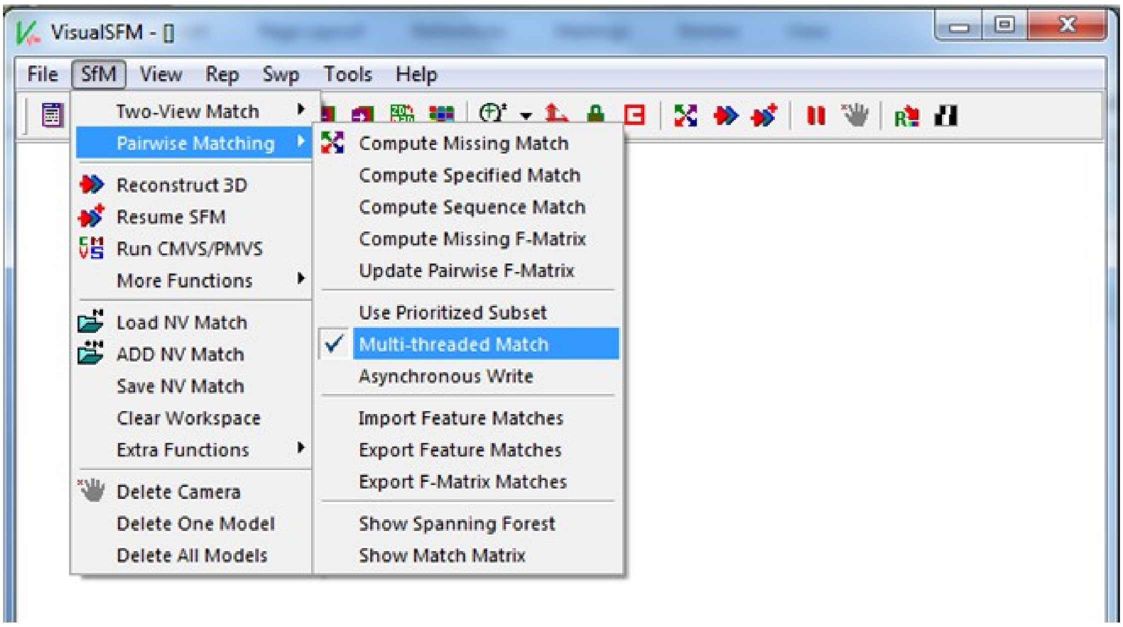


Figure 4 VisualSFM's mult-theading option.

- Import your images into VisualSFM by selecting `File -> Open + Multi Images`, or clicking on at the toolbar.

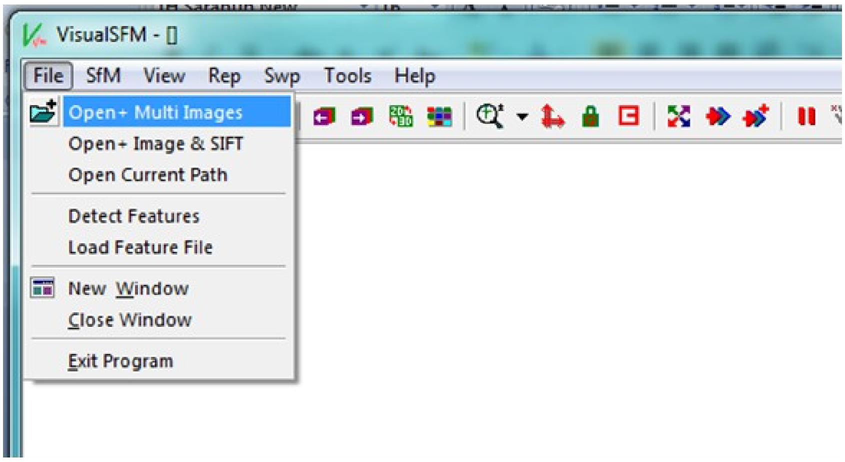
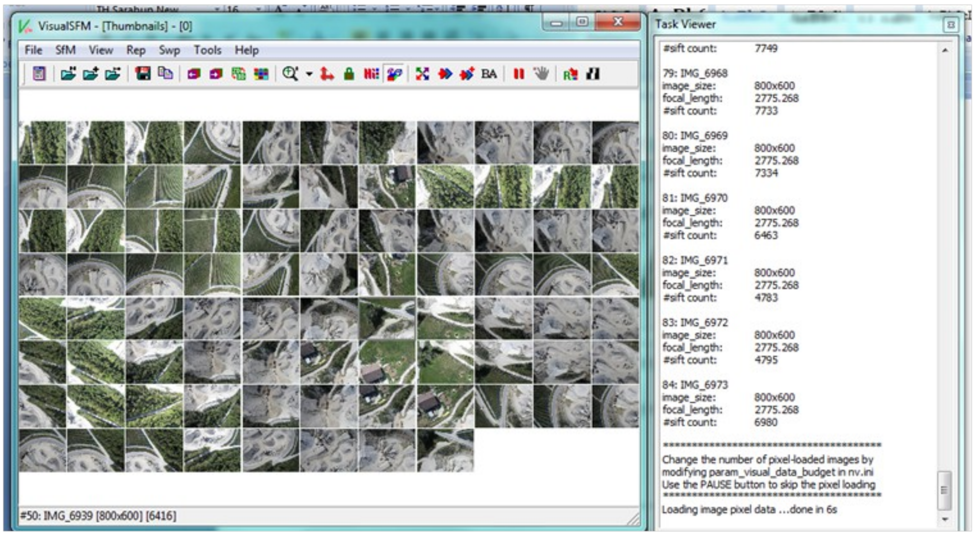


Figure 5 Loading images.

- You can select multiple images by holding `SHIFT` or `CTRL` key while clicking on the images.

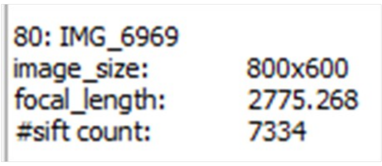
- The program will then loads and shows some image information in the Task Viewer window.


VisualSFM - [Thumbnails] - [0]
File SfM View Rep Swp Tools Help
BA
#50: IMG_6939 [800x600] [6416]
Task Viewer

```
#sift count:        7749

79: IMG_6968
image_size:         800x600
focal_length:       2775.268
#sift count:        7733

80: IMG_6969
image_size:         800x600
focal_length:       2775.268
#sift count:        7334

81: IMG_6970
image_size:         800x600
focal_length:       2775.268
#sift count:        6463

82: IMG_6971
image_size:         800x600
focal_length:       2775.268
#sift count:        4783

83: IMG_6972
image_size:         800x600
focal_length:       2775.268
#sift count:        4795

84: IMG_6973
image_size:         800x600
focal_length:       2775.268
#sift count:        6980

**************************************
Change the number of pixel-loaded images by
modifying param_visual_data_budget in nv.ini
Use the PAUSE button to skip the pixel loading
**************************************
Loading image pixel data ...done in 6s
```

Figure 6 Viewing image thumbnails.

- In the Task Viewer, you should see the following information:



```
80: IMG_6969
image_size:         800x600
focal_length:       2775.268
#sift count:        7334
```

- Run SIFT feature detector and matcher. Select `SFM->Pairwise Matching->Compute Missing Match` to run the detector and matcher.

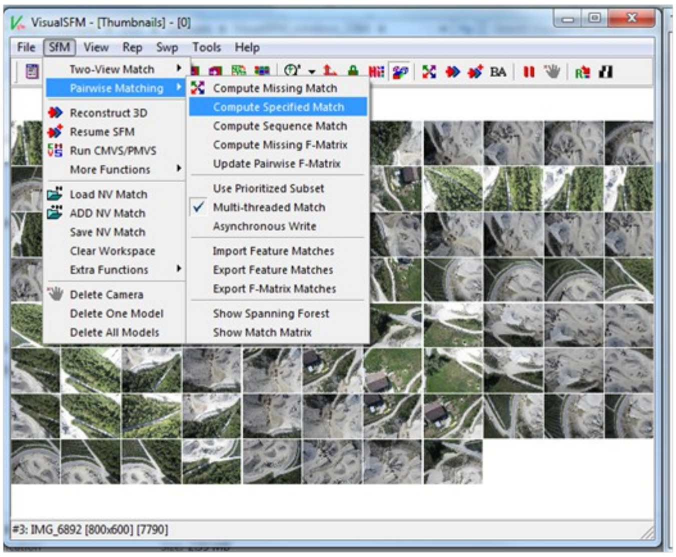
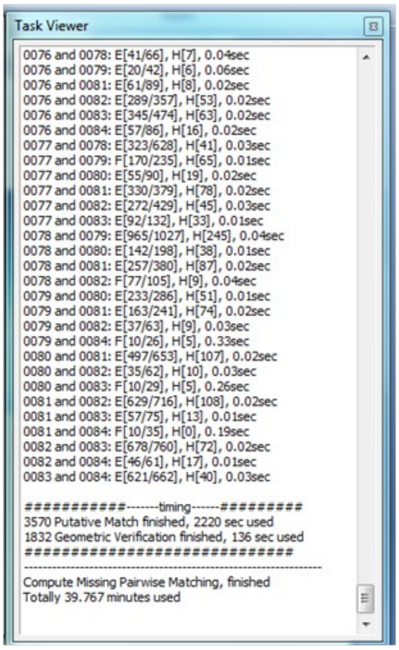


Figure 7 Compute missing matches.

- Wait until the algorithm finishs its task. The Task Viewer should look like this.



```
0076 and 0078: E[41/66], H[7], 0.04sec
0076 and 0079: E[20/42], H[6], 0.06sec
0076 and 0081: E[61/89], H[8], 0.02sec
0076 and 0082: E[289/357], H[53], 0.02sec
0076 and 0083: E[345/474], H[63], 0.02sec
0076 and 0084: E[57/86], H[16], 0.02sec
0077 and 0078: E[323/628], H[41], 0.03sec
0077 and 0079: F[170/235], H[65], 0.01sec
0077 and 0080: E[55/90], H[19], 0.02sec
0077 and 0081: E[330/379], H[78], 0.02sec
0077 and 0082: E[272/429], H[45], 0.03sec
0077 and 0083: E[92/132], H[33], 0.01sec
0078 and 0079: E[965/1027], H[245], 0.04sec
0078 and 0080: E[142/198], H[38], 0.01sec
0078 and 0081: E[257/380], H[87], 0.02sec
0078 and 0082: F[77/105], H[9], 0.04sec
0079 and 0080: E[233/286], H[51], 0.01sec
0079 and 0081: E[163/241], H[74], 0.02sec
0079 and 0082: E[37/63], H[9], 0.03sec
0079 and 0084: F[10/26], H[5], 0.33sec
0080 and 0081: E[497/653], H[107], 0.02sec
0080 and 0082: E[35/62], H[10], 0.03sec
0080 and 0083: F[10/29], H[5], 0.26sec
0081 and 0082: E[629/716], H[108], 0.02sec
0081 and 0083: E[57/75], H[13], 0.01sec
0081 and 0084: F[10/35], H[0], 0.19sec
0082 and 0083: E[678/760], H[72], 0.02sec
0082 and 0084: E[46/61], H[17], 0.01sec
0083 and 0084: E[621/662], H[40], 0.03sec

###########-------timing------#########
3570 Putative Match finished, 2220 sec used
1832 Geometric Verification finished, 136 sec used
#############################
------------------------------------------------------------
Compute Missing Pairwise Matching, finished
Totally 39.767 minutes used
```

Figure 8 Results of feature matching.

- Next, run Sparse Reconstruction. Select `SFM->Reconstruct 3D` or clicking on at the application toolbar.

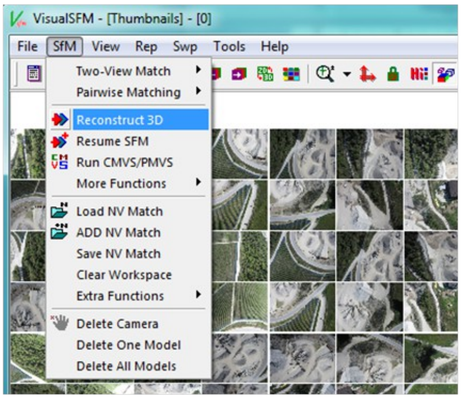
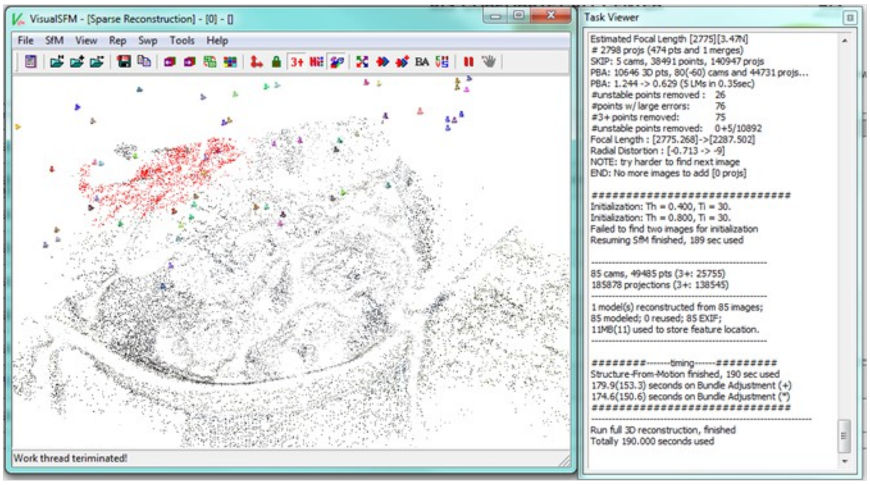


Figure 9 Reconstruct sparse 3D point.

- Wait until the reconstruction is completed.

- You should see something like this.



```
Work thread teriminated!
```

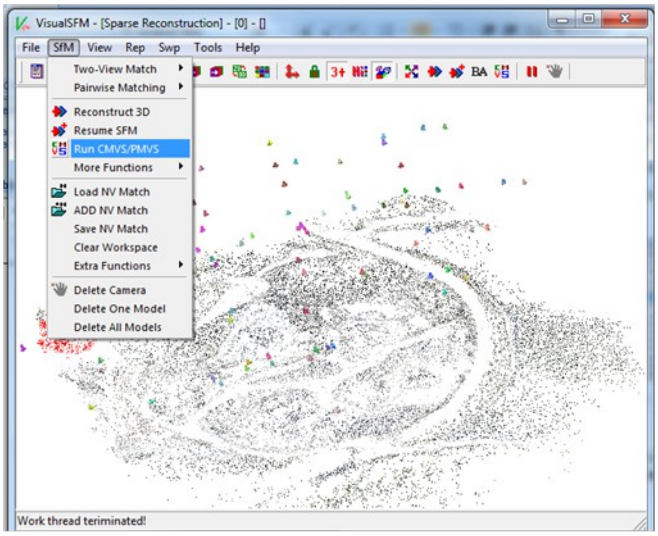
```
Estimated Focal Length [2775][3.47N]
# 2798 projs (474 pts and 1 merges)
SKIP: 5 cams, 38491 points, 140947 projs
PBA: 10646 3D pts, 80(-60) cams and 44731 projs...
PBA: 1.244 -> 0.629 (5 LMs in 0.35sec)
#unstable points removed :    26
#points w/ large errors:      76
#3+ points removed:           75
#unstable points removed:     0+5/10892
Focal Length : [2775.268]->[2287.502]
Radial Distortion : [-0.713 -> -9]
NOTE: try harder to find next image
END: No more images to add [0 projs]

##################################
Initialization: Th = 0.400, Ti = 30.
Initialization: Th = 0.800, Ti = 30.
Failed to find two images for initialization
Resuming SfM finished, 189 sec used

--------------------------------------------------
85 cams, 49485 pts (3+: 25755)
185878 projections (3+: 138545)
--------------------------------------------------
1 model(s) reconstructed from 85 images;
85 modeled; 0 reused; 85 EXIF;
11MB(11) used to store feature location.
--------------------------------------------------

#########-------timing------##########
Structure-From-Motion finished, 190 sec used
179.9(153.3) seconds on Bundle Adjustment (+)
174.6(150.6) seconds on Bundle Adjustment (*)
##################################
------------------------------------------------------------------
Run full 3D reconstruction, finished
Totally 190.000 seconds used
```

Figure 10 Sparse pointcloud.

- **Mouse Controls and Navigation:**
  - Right double click on a camera in 3D point mode to view the selected image
  - Left double click on an image in thumbnail mode to view the selected image
  - Right double click in the single image mode to return to previous view mode.
  - Pan/rotate by Left/Right click + hold + move.
  - Zooming by Shift + Mouse, or mouse wheel
  - change point size by using Alt + Mousewheel
  - Change the relative camera size by Ctrl + mouse wheel.
  - Just try left/right double/single clicking (on empty space or on camera, on image)
- You may now save the reconstructed data. Select `SFM -> Save NV Match` and give your desired NVM filename. Please note that the NVM must be saved in the same directory as the image files.
- You can add more images, compute the missing matches and run the sprase reconstruction again.

5. Dense reconstruction using PMVS2/CMVS

- Run dense reconstruction by selecting the menu: `SFM -> Run CMVS/PMVS.` You will be ask to save the NVM file. Please note that the NVM file must be saved inside the same directory as the image files.



Work thread teriminated!

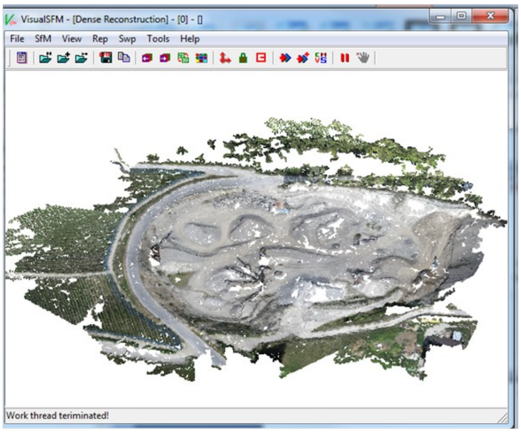
Figure 11 Runing Dense reconstruction.

- The CMVS will analyze your sparse results and create multiple clusters and input files for feeding into PMVS2. After that, PMVS2 will run dense reconstruction cluster-by-cluster.

- IMPORTANT: the dense recosntruction processes are both resource and time consuming and they may take all of you machine resources, i.e. 100% of both RAM and CPU cores. Therefore, please be patient!.

- For each cluster, the number of reconstructed 3D points will be shown in the Task Viewer. If the number of vertices is reported as 0, then the dense reconstruction is failed. When PMVS2 finishes its jobs, the dense pointcloud will be saved in the PLY format. It will have the same filename as you have entered, but the file extension will be xxxx.0.ply.

- View the dense pointcloud by selecting the menu: `View->Dense 3D Points`.

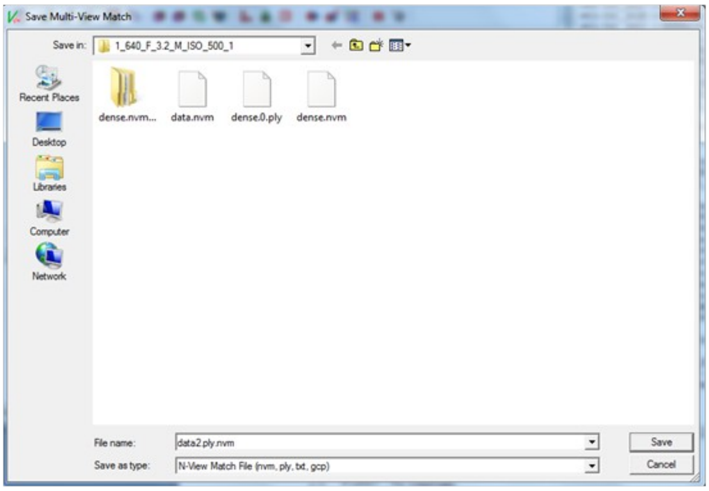


Work thread teriminated!

Figure 12 Dense pointcloud.

6. Save/Export

- Select the menu: `SFM -> Save NV Match`
- Select your target file type in the Save in type dropdown menu.



Save Multi-View Match

Save in: 1_640_F_3.2_M_ISO_500_1

File name: data2.ply.nvm

Save as type: N-View Match File (nvm, ply, txt, gcp)

Figure 13 Exporting data.