
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 manageable 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 developed 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

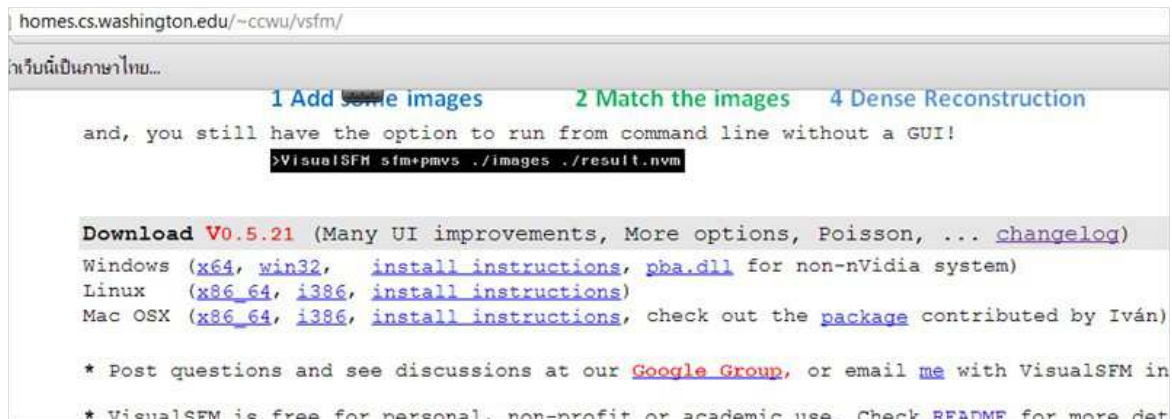


Figure 1 Downloading VisualSfM.

- Unzip to your desired directory

2.2. PMVS2/CMVS

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

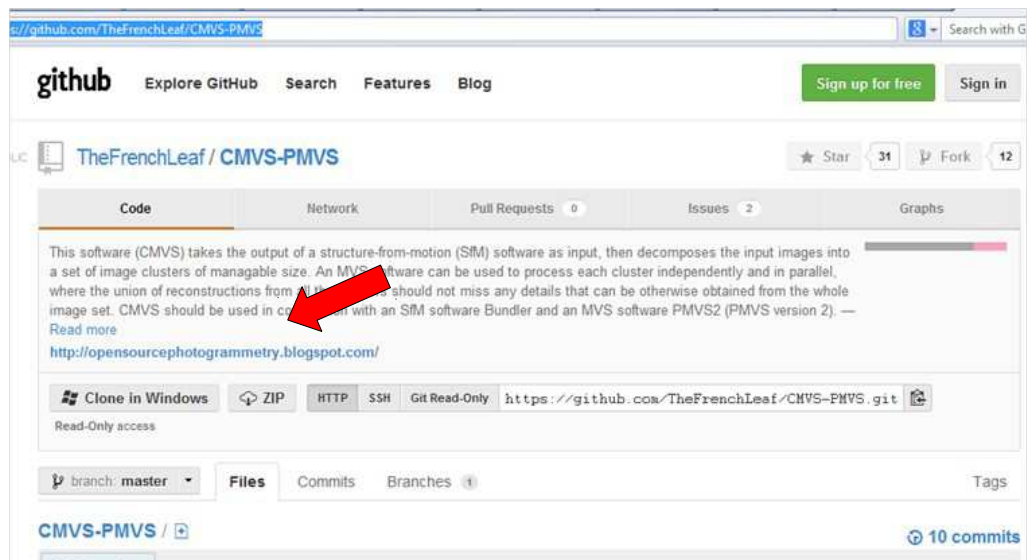


Figure 2 Downloading PMVS2/CMVS.

- 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**.

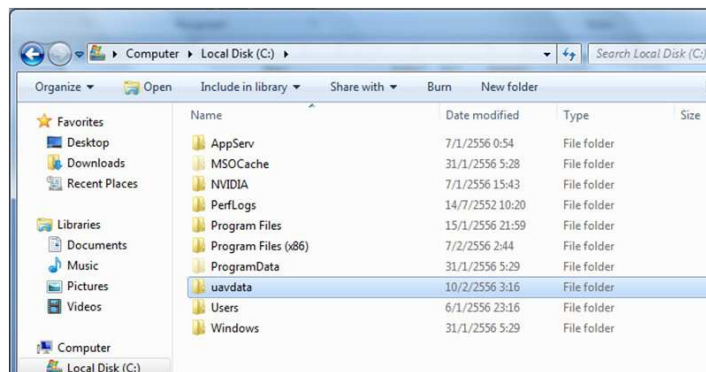


Figure 3 Creating working directory.

- 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 parallel using the multi-threading technique. However, this may crash in some machine. Therefore, you may need to disable this option by going to **SfM->Pairwise Matching** and uncheck the **Multi-threaded Match** option.

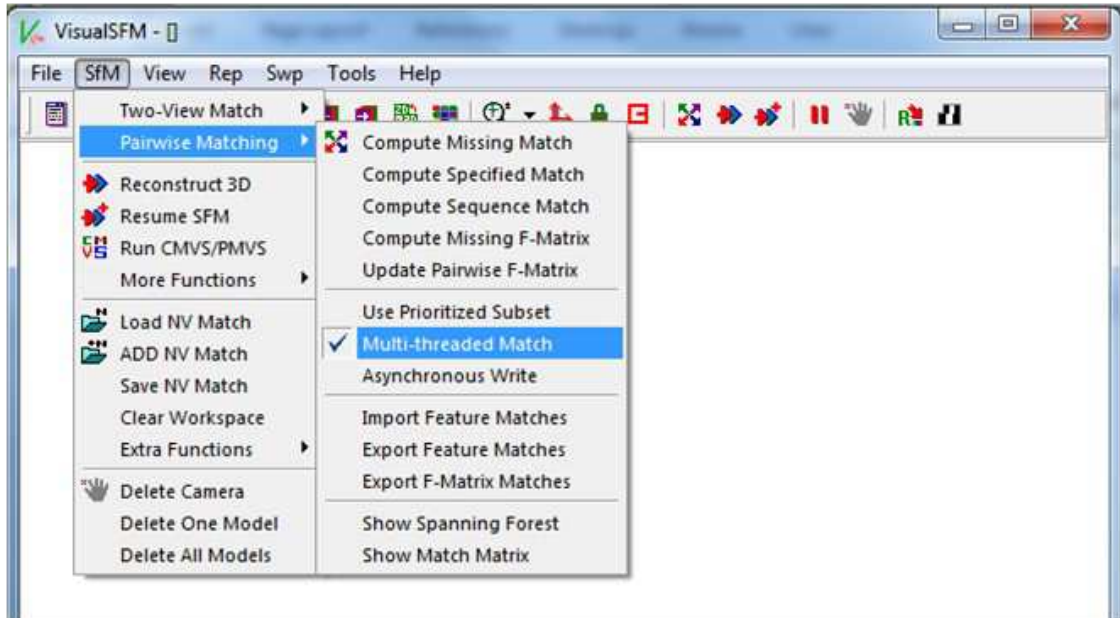



Figure 4 VisualSFM's multi-threading option.

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

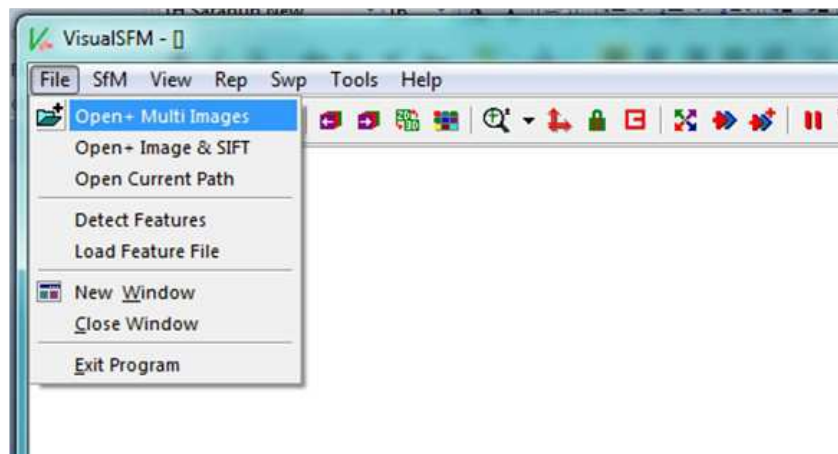


Figure 5 Loading images.

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

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

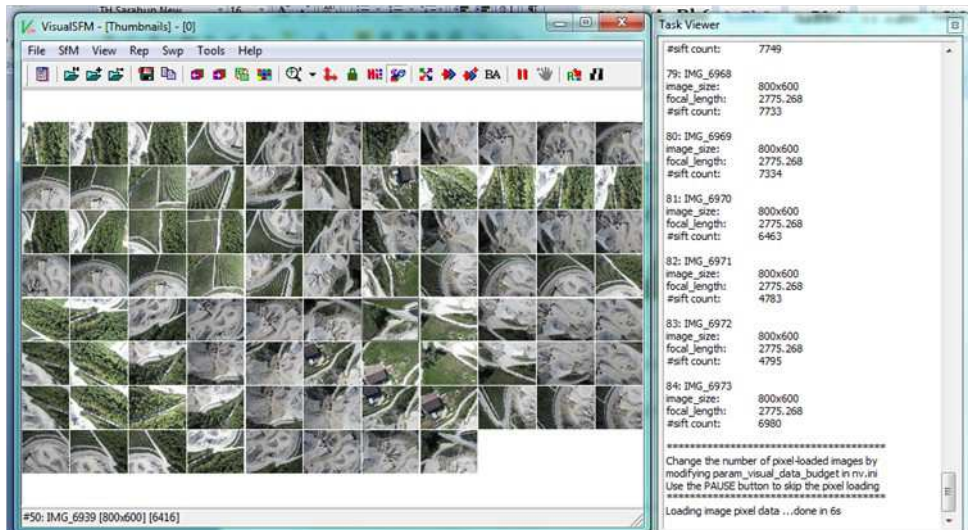


Figure 6 Viewing image thumbnails.

- 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.

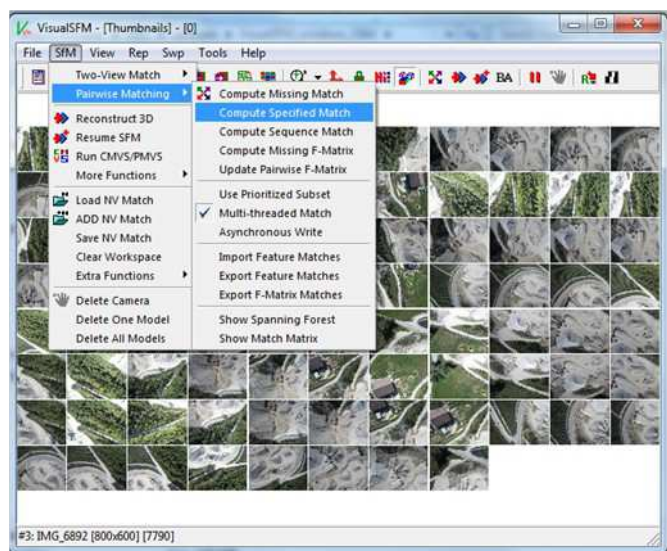


Figure 7 Compute missing matches.

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

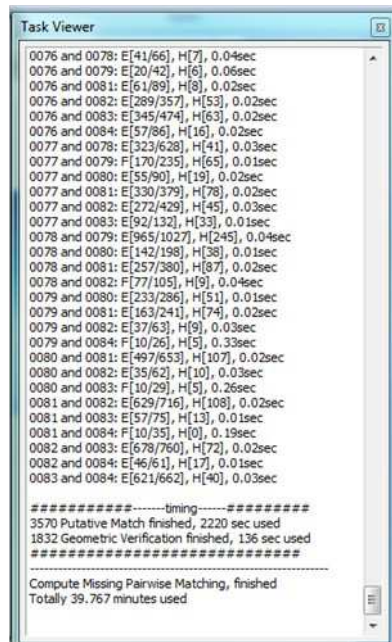



Figure 8 Results of feature matching.

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

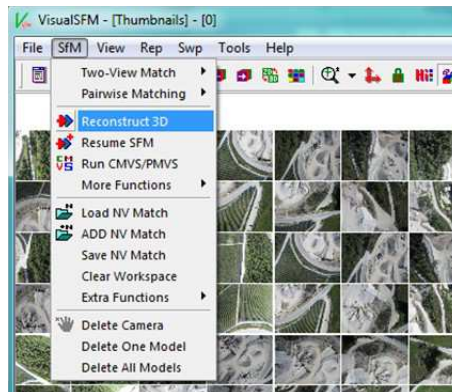


Figure 9 Reconstruct sparse 3D point.

- **Wait** until the reconstruction is completed.

- You should see something like this.

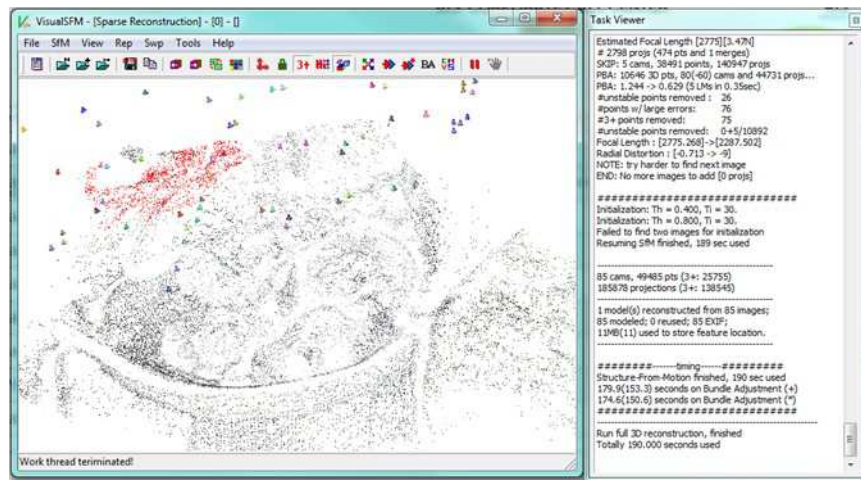


Figure 10 Sparse pointcloud.

- 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 asked to save the NVM file. Please note that the NVM file must be saved inside the same directory as the image files.

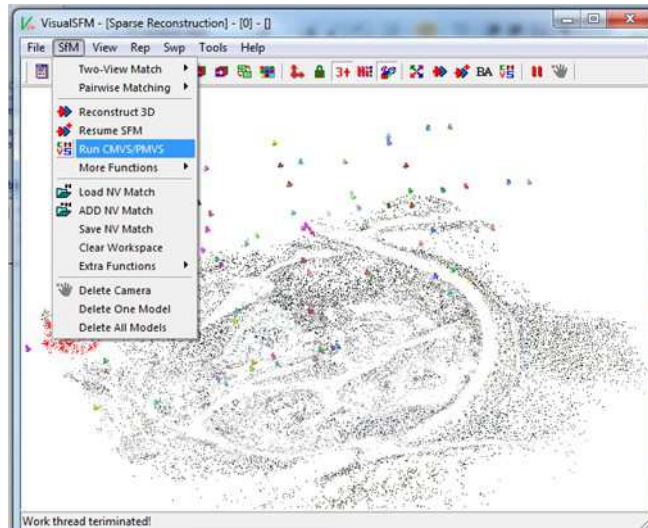


Figure 11 Running Dense reconstruction.

- 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 reconstruction processes are both **resource and time consuming** and they may take all of your 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**.

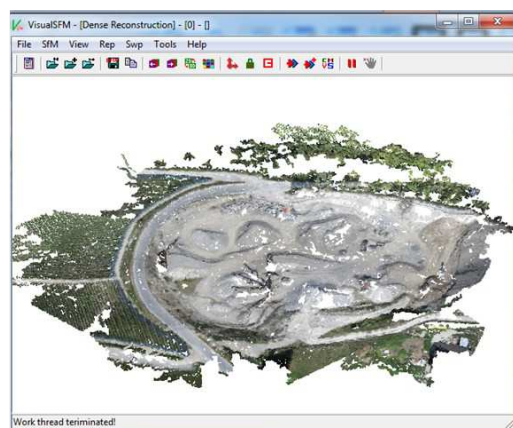


Figure 12 Dense pointcloud.

6. Save/Export

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

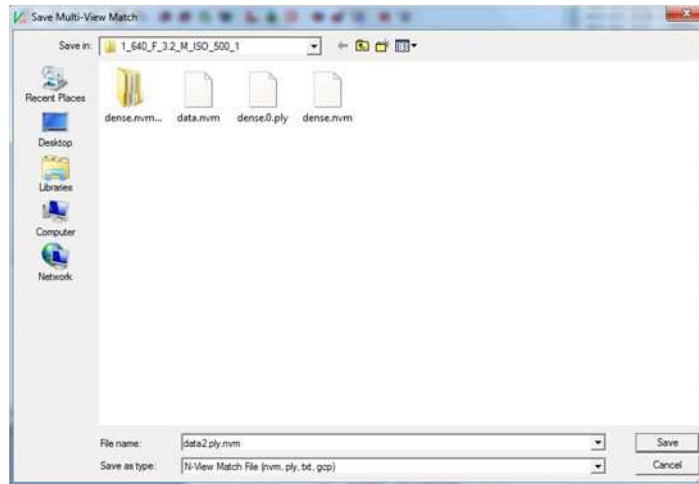


Figure 13 Exporting data.