How to VisualSFM

Jacob A. Morgan & Daniel J. Brogan

Department of Civil & Environmental Engineering
Colorado State University
Fort Collins, Colorado

January 2016

Contents

1 Introduction

This document was prepared by Jacob Morgan and Dan Brogan as a simple tutorial on how to download and use VisualSFM to create 3-dimensional (3D), scaled, and georeferenced point clouds. This tutorial includes a set of photographs as an example for the workflow.

Structure from motion (SfM) is a technique that can be used for obtaining topographic data from digital imagery that is becoming rapidly popular in the geosciences. A number of software options are available for processing photographs to obtain 3D point clouds with XYZ coordinates, as well as RGB color values.

Freely available software for SfM processing includes Bundler Photogrammetry Package (Snavely et al. 2006), SFMToolkit, PhotoSynth Toolkit, Python Photogrammetry Toolbox, VisualSFM (Wu 2013), and 3DF Samantha. Proprietary software includes Agisoft PhotoScan, Acute3D, PhotoModeler, and 3DF Zephyr. Web-based services include Photosynth, Arc3D (Tingdahl and Van Gool 2011), CMP SfM Web Service, and Autodesk 123D Catch.

One of the more popular free options is VisualSFM, which was created by Changchang Wu by combining a few of his previous projects. More information about VisualSFM can be found on Dr. Wu's website (`http://ccwu.me/vsfm`).

We are by no means experts in this technology. We suggest using the plethora of online resources to increase your own knowledge of these topics and methods. A good place to find general information about SfM methods for geomorphology is James Dietrich's Advanced Geographic Research blog (`http://adv-geo-research.blogspot.com/`).

For general SfM references consider:

- Furukawa, Y., and J. Ponce, 2010, Accurate, dense, and robust multiview stereopsis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(8): 1362–1376, doi:10.1109/TPAMI.2009.161.
- Lowe, D.G., 2004, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, **60**(2): 91–110, doi:10.1023/B:VISI.0000029664.99615.94.
- Snavely, N., S.M. Steitz, and R. Szeliski, 2006, Photo tourism: Exploring photo collections in 3D, in Proceedings of ACM SIGGRAPH 2006, *ACM Transactions on Graphics*, **25**(3): 835–846, doi:10.1145/1141911.1141964.
- Snavely, N., S.M. Steitz, and R. Szeliski, 2008, Modeling the world from internet photo collections, *International Journal of Computer Vision*, **80**(2): 189–210, doi:10.1007/s11263-007-0107-3.
- Tingdahl, D., and L. Van Gool, 2011, A public system for image based 3D model generation, in *Computer Vision/Computer Graphics Collaboration Techniques*, A. Gagalowicz and W. Philips (eds.), Springer-Verlag: Berlin, 262–273, doi:10.1007/978-3-642-24136-9_23.
- Ullman, S., 1979, The interpretation of structure from motion, *Proceedings of the Royal Society of London, Biological Sciences*, **203**(1153), 405–426, doi:10.1098/rspb.1979.0006
- Wu, C., 2013, Towards linear-time incremental structure from motion, in *3DV 2013, 2013 International Conference on 3D Vision*, Seattle, Wash., 127–134, doi:10.1109/3DV.2013.25.
- Wu, C., S. Agarwal, B. Curless, and S.M. Seitz, 2011, Multicore bundle adjustment, *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3057–3064, doi:10.1109/CVPR.2011.5995552.

For SfM references related to geoscience applications consider:

- Clapuyt, F., V. Vanacker, and K. Van Oost, 2015, Reproducibility of UAV-based earth topography reconstructions based on Structure-from-Motion algorithms, *Geomorphology*, doi:10.1016/j.geomorph.2015.05.011.
- Deitrich, J.T., 2014, Applications of Structure-from-Motion photogrammetry to fluvial geomorphology, PhD Dissertation, Department of Geography, University of Oregon, 109 p., available online.
- Dietrich, J.T., 2016, Riverscape mapping with helicopter-based Structure-from-Motion photogrammetry, *Geomorphology*, **252**: 144–157, doi:10.1016/j.geomorph.2015.05.008.
- Favalli, M., A. Fornaciai, I. Isola, S. Tarquini, and L. Nannipieri, 2012, Multiview 3D reconstruction in geosciences, *Computers & Geosciences*, **44**: 168–176, doi:10.1016/j.cageo.2011.09.012.
- Fonstad, M.A., J.T. Dietrich, B.C. Courville, and P.E. Carbonneau, 2013, Topographic structure from motion: a new development in photogrammetric measurements, *Earth Surface Processes and Landforms*, **38**(4): 421–430, doi:10.1002/esp.3366.
- Gomez-Gutierrez, A., S. Schnabel, F. Berenguer-Sempere, F. Lavado-Contador, and J. Rubio-Delgado, 2014, Using 3D photo-reconstruction methods to estimate gully headcut erosion, *Catena*, **120**: 90–101, doi:10.1016/j.catena.2014.04.004.
- James, M.R., and S. Robson, 2012, Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application, *Journal of Geophysical Research: Earth Surface*, **117**(F3), doi:10.1029/2011JF002289.
- Javernick, L., J. Brasington, and B. Caruso, 2014, Modeling the topography of shallow braided rivers using Structure-from-Motion photogrammetry, *Geomorphology*, **213**: 166–182, doi:10.1016/j.geomorph.2014.01.006.
- Micheletti, N., J.H. Chandler, and S.N. Lane, 2015, Investigating the geomorphological potential of freely available and accessible structure-from-motion photogrammetry using a smartphone, *Earth Surface Processes and Landforms*, **40**(4): 473–486, doi:10.1002/esp.3648.
- Westoby, M., J. Brasington, N.F. Glasser, M.J. Hambrey, and M.J. Reynolds, 2012, 'Structure-from-Motion' photogrammetry: a low-cost, effective tool for geoscience applications, *Geomorphology*, **179**: 300–314, doi:10.1016/j.geomorph.2012.08.021.
- Woodget, A.S., P.E. Carbonneau, F. Visser, and I.P. Maddock, 2015, Quantifying submerged fluvial topography using hyperspatial resolution UAS imagery and structure from motion photogrammetry, *Earth Surface Processes and Landforms*, **40**(1): 47–64, doi:10.1002/esp.3613.

1.1 Example Photographs

The example dataset is comprised of photographs taken in Skin Gulch, Larimer County, Colorado on 6 October 2015. The photographs were collected using a Canon Rebel T3i with a 24 mm prime lens mounted on a 12 ft painter's pole. A remote with a cable extension was used to initiate the camera shutter. 8 in × 8 in aluminum composite targets were pre-made using target designs from Agisoft PhotoScan and evenly placed on the ground throughout the region of interest. RTK-GPS surveying was used to determine the absolute position of these targets that can then be used to orient the 3D point cloud in real space.

1.2 Tips for Acquiring Photographs

There are no hard and fast rules for taking photographs for SfM applications. There are, however, some tips that can increase the likelihood of successfully creating a point cloud from digital imagery. In general, convergent camera views produce better results than either divergent or parallel views (Fig. 1). In fact, parallel views have been known to result in significant distortion in the point cloud.

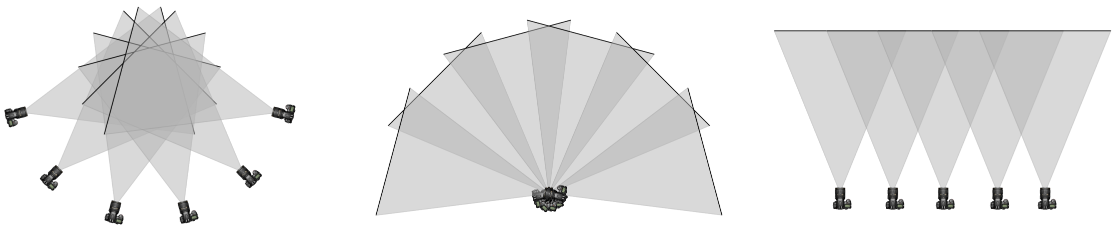

Figure 1: Convergent (left), divergent (center), and parallel (right) camera views. Graphics from J. Dietrich's Advanced Geographic Research blog (`http://adv-geo-research.blogspot.com/2014/02/camera-geometries-for-structure-from.html`).

Inevitably you will likely have some combination of these camera view types; that is okay. Once you have acquired your photographs it is best to go through them individually to remove any blurred images.

NOTE! This tutorial has not been endorsed or approved by Changchang Wu or any other researcher associated with VisualSFM or any technologies on which it depends.

2 Downloading VisualSFM

2.1 VisualSFM

VisualSFM software can be downloaded from the VisualSFM website (`http://ccwu.me/vsfm`). The homepage of the website has a section titled "Download," where the most recent version is available to download for Windows, Linux, and Mac (Fig. 2). We use 64-bit Windows and so that is what this tutorial is set up to show; however, we believe the same general steps can be applied to other operating systems to achieve the same results.

Figure 2: The VisualSFM website, showing both the Download section and Documentation section.

Choose the appropriate operating system and version. The corresponding hyperlink will connect to a .zip folder containing all of the necessary files. Simply download and unzip this file to a convenient location on your computer. Free unzipping software is available via 7-Zip (`http://www.7-zip.org`). After unzipping the file you should have a folder whose name corresponds to the operating system and version you chose (e.g. `VisualSFM_windows_64bit`). This folder contains all the necessary files to simply run VisualSFM (Fig. 3). The executable file `VisualSFM.exe` can be launched immediately to open the graphical user interface (GUI). See the file `README.txt` for information regarding the proper citations to use when publishing results from VisualSFM.

Installation help is also provided on the VisualSFM website at `http://ccwu.me/vsfm/install.html`. The site additionally contains a documentation page that is often useful (`http://ccwu.me/vsfm/doc.html`). A good deal of the work flow information contained in this tutorial is coincident with and derived from information there.

2.2 PMVS/CMVS

Dense reconstruction in VisualSFM is achieved with an integration of Patch-based Multi-view Stero (PMVS) software developed by Yasutaka Furuwaka and Jean Ponce, as well as Clustering Views for Multi-view Stereo (CMVS) software developed by Yasutaka Furuwaka.

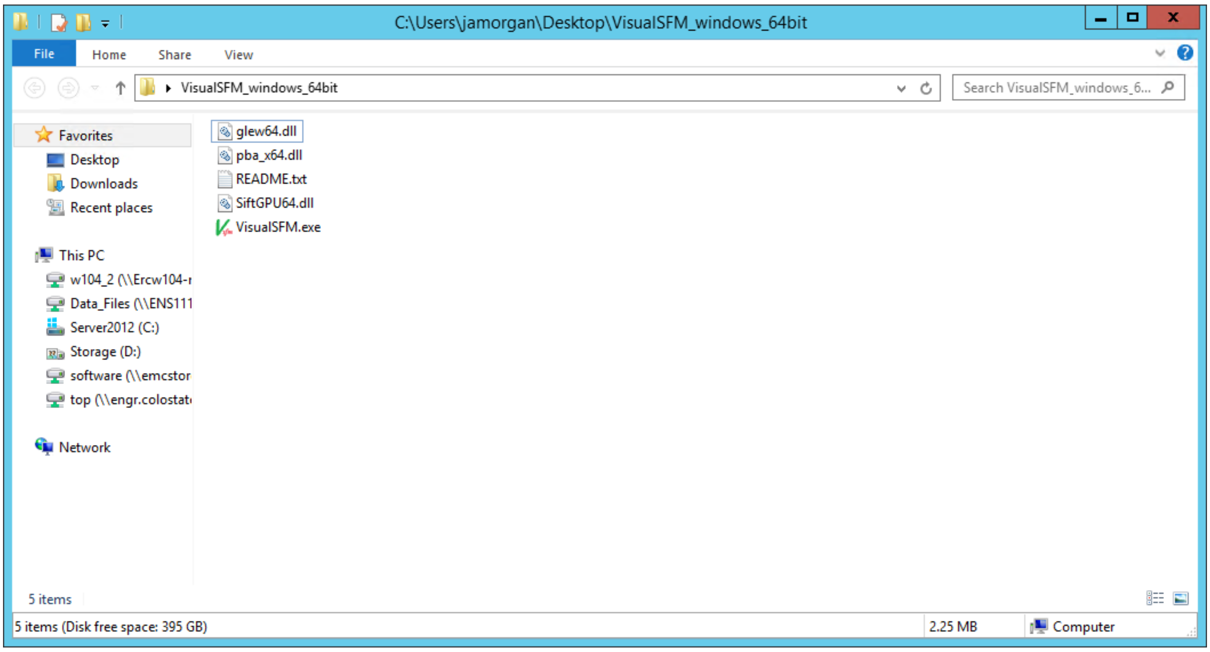

Figure 3: Contents of the VisualSFM folder after unzipping.

Background information and downloadable files for these methods and files can be obtained from at least a few places. The websites for both PMVS (`http://www.di.ens.fr/pmvs`) and CMVS (`http://www.di.ens.fr/cmvs`) are useful, but Pierre Moulon's GitHub (`https://github.com/pmoulon/CMVS-PMVS`) is recommended for downloading as both required files can be pulled from this single location. All that is required for VisualSFM to use PMVS/CMVS is three executable files: `pmvs2.exe`, `cmvs.exe`, and `genOption.exe`. The files can be downloaded by selecting the folder `binariesWin-Linux` in GitHub, selecting the folder corresponding to your operating system and version (e.g. `Win64-VS2010`), selecting `pmvs2.exe`, and finally selecting the link that says "view the full file." Your browser may either automatically download the file or prompt you to navigate to the location where you want the file saved. Save the file in the VisualSFM folder created previously (e.g. `VisualSFM_windows_64bit`, Fig. 3). Repeat the same process for the `cmvs.exe` and `genOption.exe` files. Once these files are in your VisualSFM folder you are ready to use the GUI to create a point cloud from photographs. When you open the GUI it will look something like Figure 4.

2.3 Customization

Once you have all the necessary files in the VisualSFM folder and have opened the GUI for the first time a configuration file is created called `nv.ini`. This file is where parameters related to both VisualSFM and PMVS/CMVS are stored. This file provides a breadth of user control for optimizing the process. We **highly** recommend changing the variable `param_less_visualization_data` from zero to a value of one (Fig. 5), especially if you have a large number of images or high quality images. When the value is zero VisualSFM will load pixel data from all of the images for thumbnail visualization. Setting the value to one will suppress this feature, saving both memory and time! This tutorial will assume that such an edit has been made. If VisualSFM is open while `nv.ini` is being edited it may be necessary to close and reopen before the changes take effect. There are additional parameters in `nv.ini` that can be altered to change how the software processes the images.

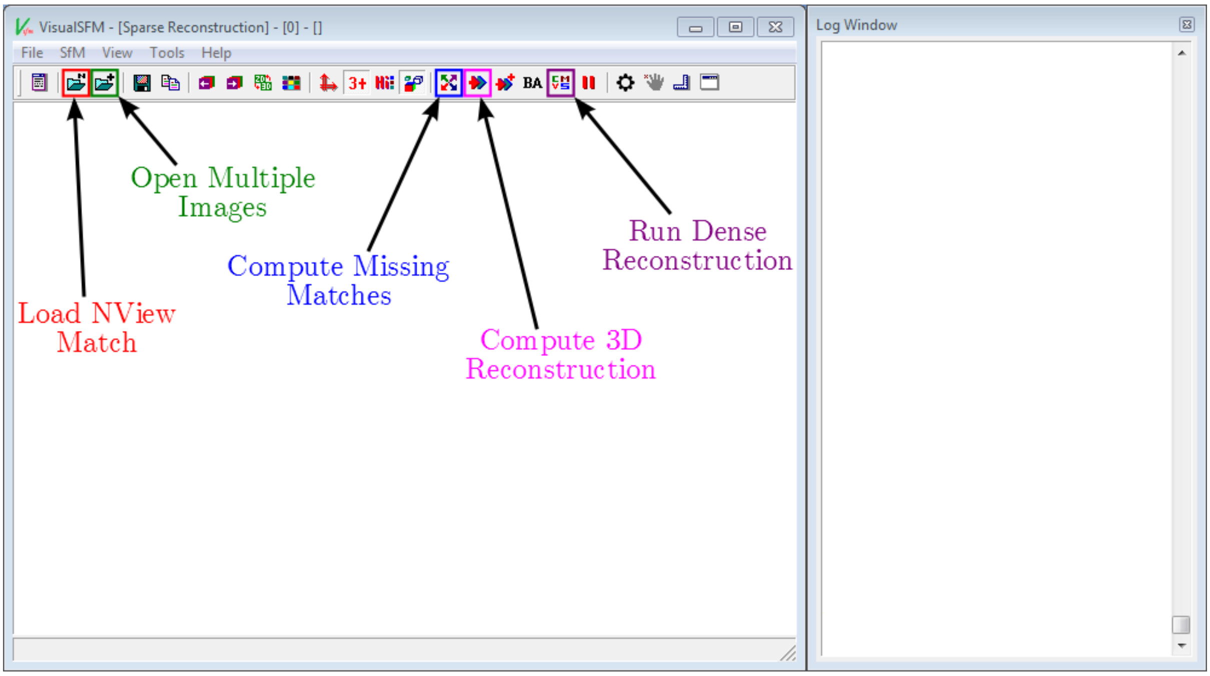

Figure 4: VisualSFM GUI with the most commonly used shortcuts highlighted. Note: All of the shortcuts may not be visible when you first open the GUI.

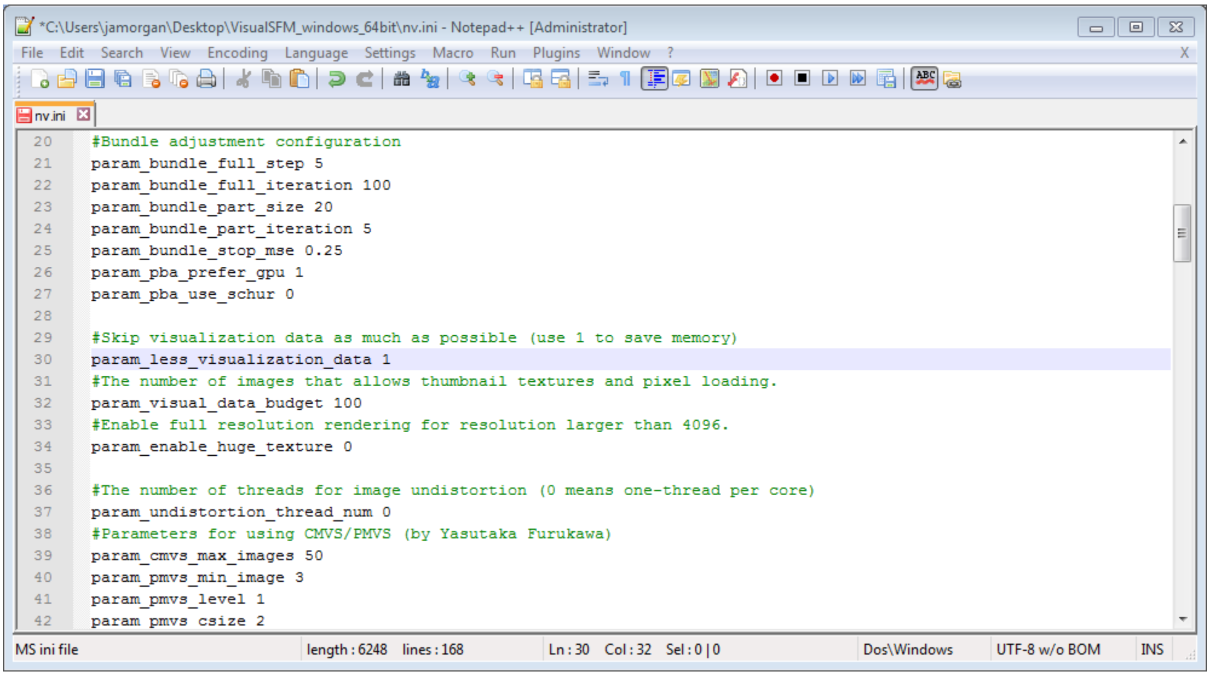

Figure 5: Skip loading of image pixels into VisualSFM.

WARNING! Some necessary components of VisualSFM will not run over Remote Desktop!

3 VisualSFM Workflow

Open VisualSFM by double-clicking `VisualSFM.exe`. Most of the tools for this tutorial will be accessible through the shortcut menu shown in Figure 4.

3.1 Importing Images

Images can be imported in VisualSFM by various methods, available both through the menu bar and as shortcuts (Fig. 4).

1. Load NView Match (shortcut or through the SfM menu tab)
   - Choose a .txt file containing relative image paths for all relevant images.
   - Choose "All JPEGs in Folder" from the format menu and select any file in the folder containing all relevant images.
2. Open Multiple Images (shortcut or through the File menu tab)
   - You can do this multiple times to import all images.

It does not matter which method you use as long as all of your relevant images are uploaded. We generally keep all photos for a project in the same folder, so using the "All JPEGs in Folder" option in Load NView Match works quite well. After you have instructed VisualSFM to import your images the log window should load each photo and display pixel information as well as other data as it imports them. Once this step is complete the Log Window should look something like Figure 6.

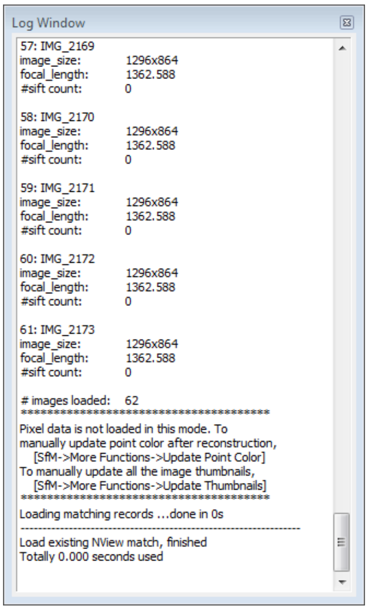

Figure 6: Log Window after importing images.

Throughout the workflow the Log Window will display relevant information related to each process and also inform you of the progress during that process. If, for some reason, your Log Window is not showing up you can go to the Tools menu at the top and select Show Log Window. For each session of VisualSFM the Log Window contents are automatically saved to a .log text file in a folder called `log` that is automatically created in your VisualSFM folder.

3.2 Matching Images

The next step is to have VisualSFM detect features in each image, and find matches. The tool for this step is called Compute Missing Matches and is accessible through the the SfM menu under Pairwise Matching or as a shortcut (Fig. 4).

VisualSFM completes two things here:

- Scale Invariant Feature Transform (SIFT)
- Full pairwise image matching

The SIFT process is generally pretty quick depending on your images sizes and the total number of images. The second step of finding matches between images generally takes longer. If you have large images and lots of them it could take hours or longer! You can go back and import additional images later and compute missing matches again if you need to. After running this process the Log Window should look something like Figure 7.

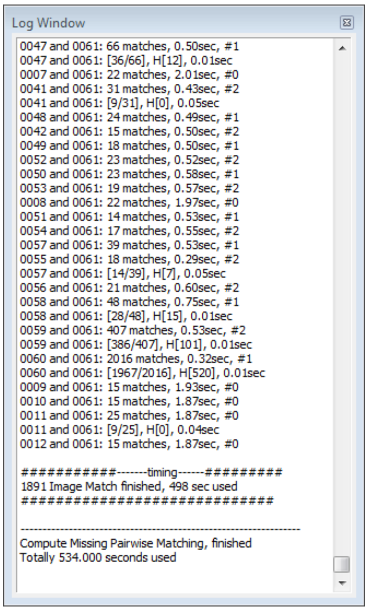

Figure 7: Log Window after computing matches.

This step also creates new files in the folder where your images are located. Each image file should now have a corresponding .sift and .mat file. If for any reason you wish to start completely over, these .sift and .mat files will need to be deleted. Otherwise, if VisualSFM sees that these files already exist, it will use them rather than creating new ones. In that case the "#sift count" parameter shown in the Log Window previously (Fig. 6) would have nonzero values.

If you have images with known pairing or want to force image pairing you can create a .txt file comprised of pairs of image paths. Go to the SfM menu and under Pairwise Matching select Compute Specified Match. If you select Compute Missing Matches full pairwise matching is performed for every image rather than the pairs you specify in your .txt file. If you have sequence images (e.g. video footage) you can go to the SfM menu and under Pairwise Matching select Compute Sequence Match.

3.3 Creating the Sparse Reconstruction

Now we want VisualSFM to turn those matches it found in the previous step into points in 3D space. This can be achieved by going to the SfM menu and selecting Reconstruct Sparse or by using the Compute 3D Reconstruction shortcut (Fig. 4).

Once VisualSFM starts creating the sparse point cloud the viewer window begins to be populated by those points as well as the camera locations. You can now begin navigating around the 3D space using your mouse and keyboard (Fig. 8):

- left-click: pan
- right-click: rotate
- wheel: zoom
- ctrl + wheel: adjust camera size
- alt + wheel: adjust point size

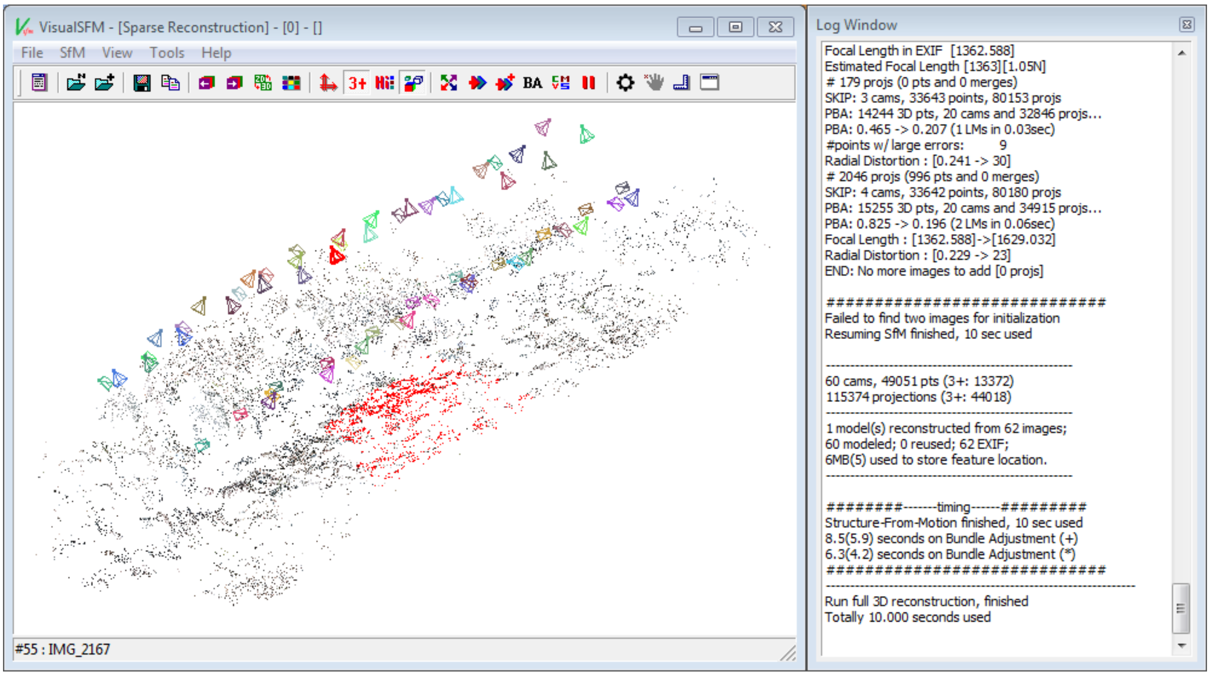

Figure 8: Sparse point cloud from Skin Gulch images.

This step generally does not take as long as finding matches. If for some reason VisualSFM begins a new point cloud partway through the computation process, try repeating the Compute 3D Reconstruction command. Again if it does not put the clouds together it may be necessary to remove those images that VisualSFM cannot match. This can be done by right-clicking the image(s) and selecting the "Remove Selected Camera/Points" shortcut (also located in the SfM menu tab).

When you right-click on a camera in the scene both the camera and the points that are associated with that image will be highlighted red. Otherwise point colors correspond to the color of the matched pixels from the images. You can double right-click on a camera to view its image in the window. You can get back to the 3D viewer with the Switch Between 2D and 3D shortcut or by going to View and selecting N-View 3D Points. There are additional keyboard shortcuts and navigation tools that can be found at `http://ccwu.me/vsfm/doc.html#gui`.

You can explore the 3D space by rotating, panning, and zooming to see the topography of the area. By default only points that are matched between three or more cameras are shown. You can toggle back and forth between showing only these points or all computed matching points by clicking on the shortcut Show Points Seen by 3+ Cameras that looks like “3+”.

3.4 Specifying Ground Control Points

At this point, after creating a sparse 3D model, we can enter ground control point (GCP) data from RTK-GPS surveying in order to scale and orient the point cloud. For the Skin Gulch example the visible targets in the included images are shown in Figure 9, and the RTK-GPS and adjusted XYZ coordinates are shown in Table 1.

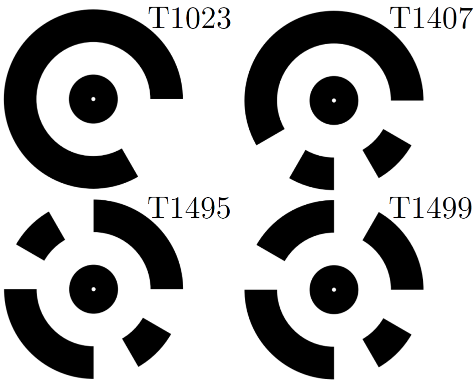

Figure 9: Targets used in Skin Gulch example images.

Table 1: Target RTK-GPS and adjusted coordinates. All values are in meters.

ID	Easting	Northing	Elevation	X_{adj}	Y_{adj}	Z_{adj}
T1023	466,927.862	4,502,539.238	1,920.079	8.862	15.238	3.079
T1407	466,926.289	4,502,533.445	1,920.289	7.289	9.445	3.289
T1495	466,931.668	4,502,539.190	1,919.379	12.668	15.190	2.379
T1499	466,928.614	4,502,532.280	1,919.901	9.614	8.280	2.901

The GCP process in VisualSFM works best if the input coordinates do not contain too many significant figures. Therefore, the adjusted XYZ values shown in Table 1 correspond to easting, northing, and elevation values in meters in UTM Zone 13.

$$X_{adj} = \text{Easting} - 466,919 \tag{1}$$

$$Y_{adj} = \text{Northing} - 4,502,524 \tag{2}$$

$$Z_{adj} = \text{Elevation} - 1,917 \tag{3}$$

You can begin specifying GCPs in VisualSFM by either clicking on the shortcut on the right that looks like a carpenter’s square or by going to the SfM menu and in More Functions selecting GCP-based Transform. From here we need to go through the indivudal images and select the GCP targets and input their coordinates. Change the view by going to the View menu and selecting Single Image.

You can use the left and right arrow keys to cycle through the images, the mouse wheel to zoom in and out, the left- or right-click to pan, and ctrl+left-click to add or modify a GCP.

Now we can choose our GCPs. Check each image to see if a target was captured in the frame. If there is a target in the frame zoom in and ctrl+left-click on the center of the target. Input the target ID and XYZ coordinates, separated by spaces, into the dialogue box that appears. The target ID can be alphanumeric but it must begin with a letter. For example, in IMG_2095 there is one target, T1495 (Fig. 10). If we zoom in on the target and select the center as the GCP we can enter "T1495 12.668 15.190 2.379" in the dialogue box (Fig. 11). Then click "OK" to set the GCP. By pressing the left or right arrow we can move to another image and repeat the process. Once you have entered the information for one target its ID and XYZ values are stored so that you can select it from the drop-down menu in the dialogue box.

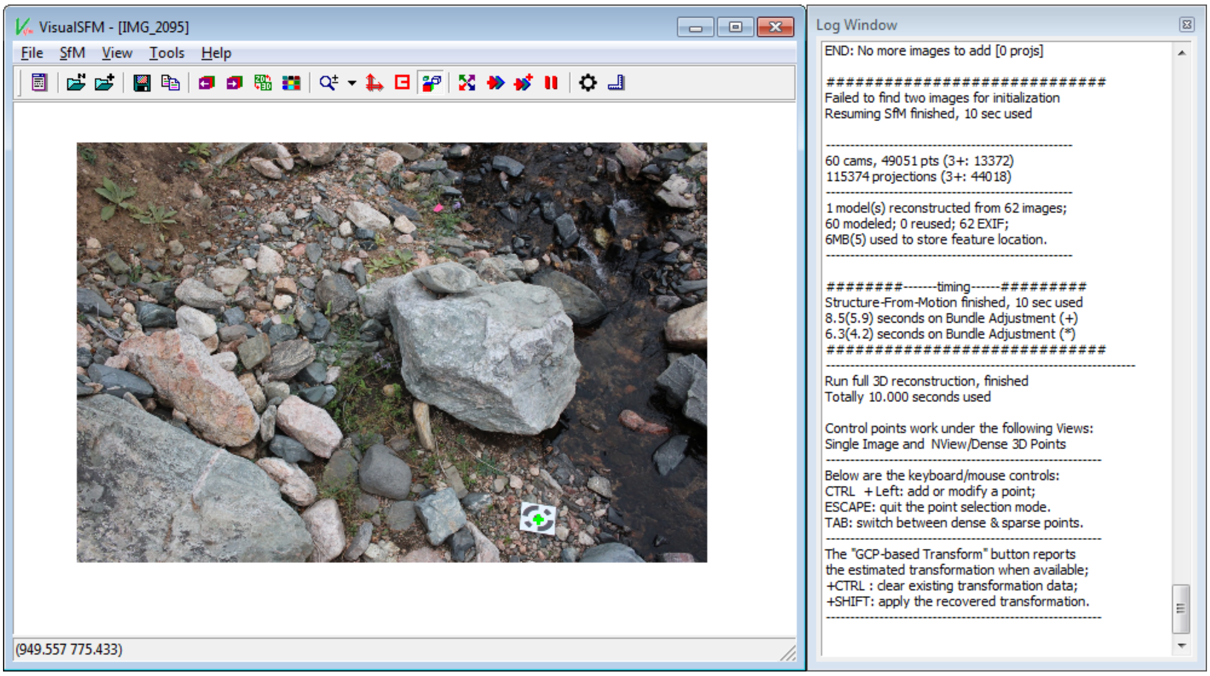

Figure 10: Image with a GCP target.

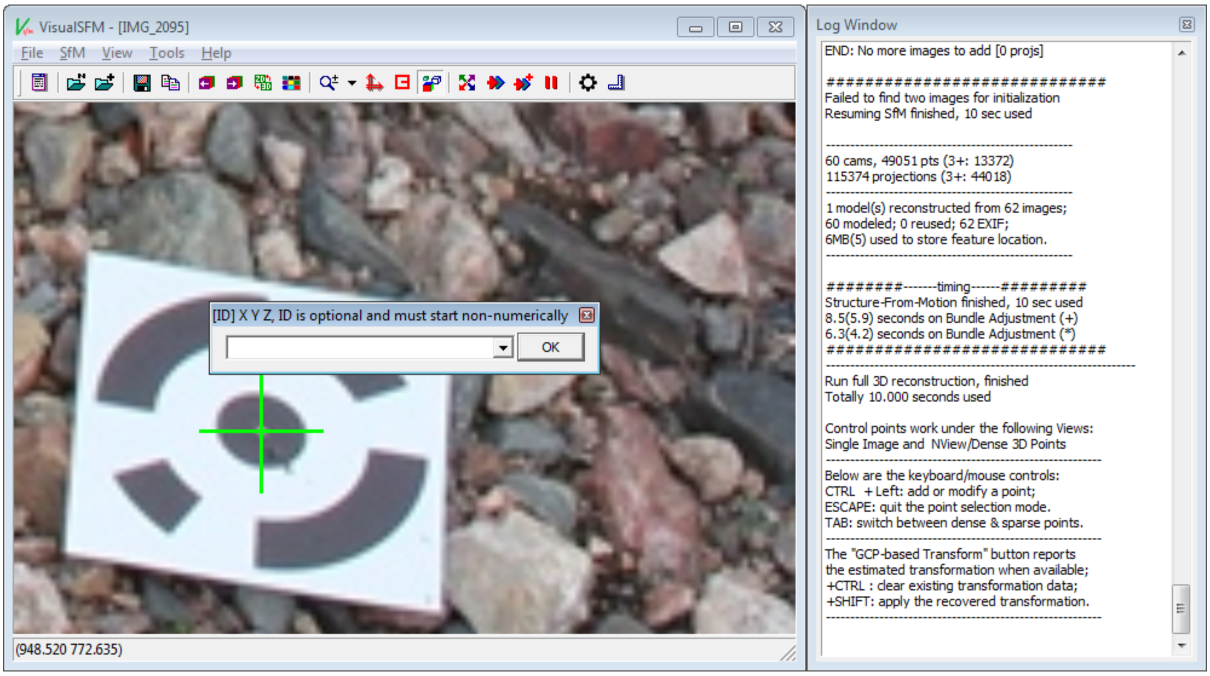

Figure 11: Specifying GCP ID and coordinates.

If you hover over a GCP that has already been identified a box shows up around it and you can ctrl+left-click to edit that GCP. After you've specified enough GCPs in multiple images the Log Window should begin saying "Transformation estimated successfully!" When you are choosing GCPs from the images aim to click with the cross-hairs over the center white point of the target. If the target is only partially visibile and you cannot see the center, either because it is out of the frame or behind an obstruction (e.g. vegetation, boulder, etc.), it is best to not select that GCP at all.

After you have specified all the GCPs in the images shift+left-click on either the GCP-based Transform shortcut or shift+left-click on GCP-based Transform in the SfM menu under More Functions. This applies the estimated transformation to the sparse point cloud. Switch back to the point cloud viewer by either selecting the Switch Between 2D and 3D shortcut or by selecting N-View 3D Points under the View menu. Your point cloud should now be georeferenced and the GCPs highlighted (Fig. 12). The Log Window displays the transformation parameters and the errors associated with the best fit of the transformed GCPs.

You can press esc to quit the point selection mode.

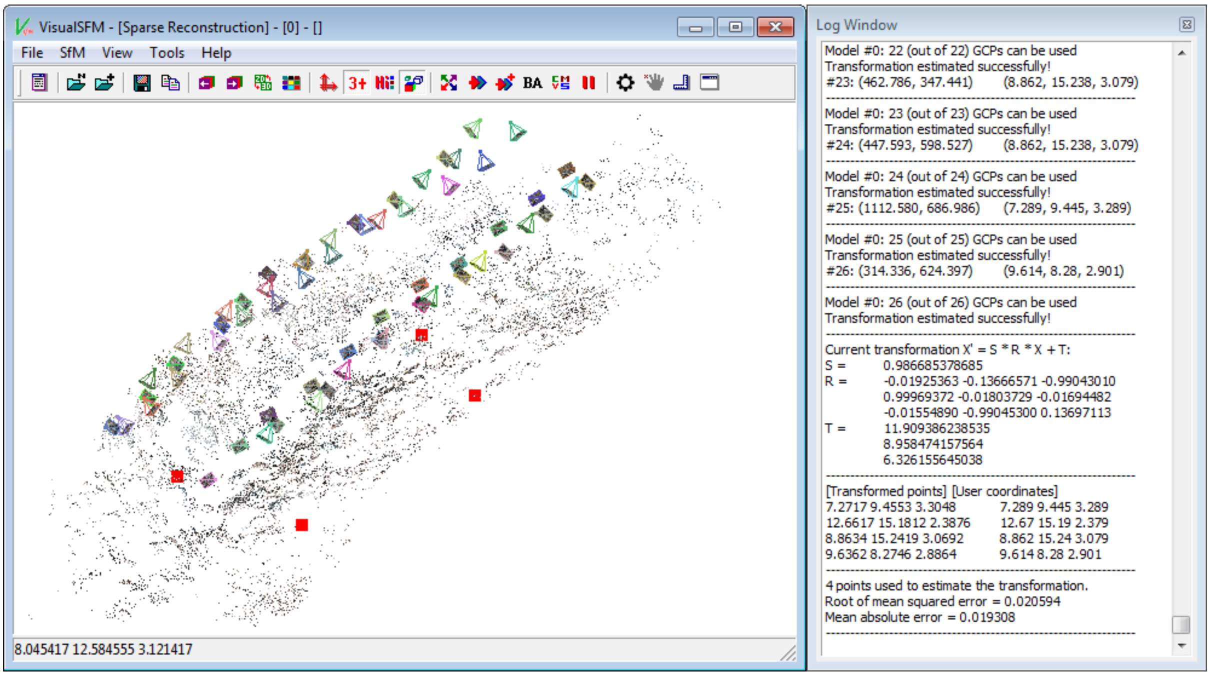

Figure 12: Georeferenced sparse point cloud showing GCPs.

3.5 Creating the Dense Reconstruction

The sparse point cloud is cool and all, but we can get a denser cloud by using the PMVS/CMVS tool. You can do this by either going to the SfM menu and selecting Reconstruct Dense or by clicking on the Run Dense Reconstruction shortcut (Fig. 4). This prompts you to save a .nvm file. When you save `[name].nvm` VisualSFM creates a folder in that location called `[name].nvm.cmvs` where the CMVS is run and the files are saved. The .ply files of your 3D model will be automatically saved here.

With more and larger images this step can also take a long time. You can keep track of the progress by watching the Log Window. Paths for output files are also shown in the Log Window.

Once it is complete you can view the dense cloud by going to the View menu and selecting Dense 3D Points. You can also toggle between the sparse and dense clouds by pressing the tab key. You can navigate around using the same methods as with the sparse reconstruction. The dense point cloud has considerably more points (Fig. 13).

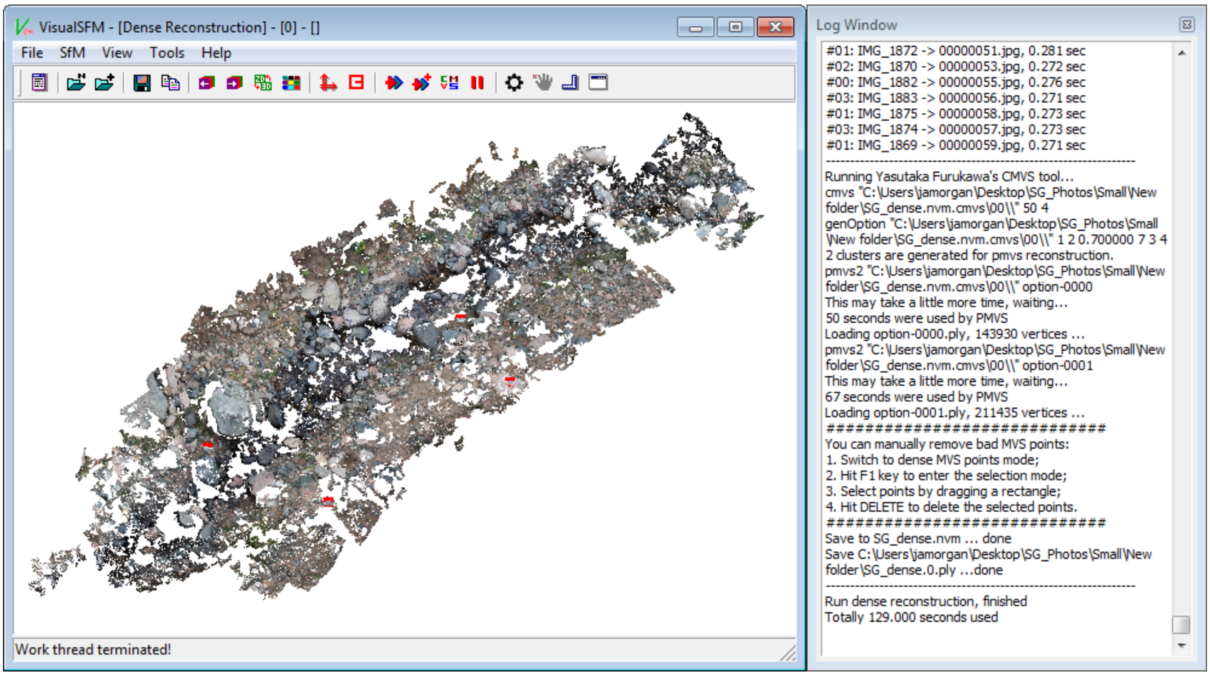

Figure 13: Dense point cloud.

There are still some large gaps in the dense point cloud; this is likely due to a number of factors. First, sub-aqueous areas are difficult for the software to render because they are both dark and somewhat featureless in the images, and because of light refraction through the water surface. Second, the resolution of the images were greatly reduced for this example so some of the detail was smoothed as the relative pixel size increased for each image. Finally, more images could have been taken and/or included to fill in some of the areas that are simply not captured in any or enough images. All of these variations should be taken into consideration depending on the objectives of your study.

3.6 Exporting the 3D Model

When you create the dense point cloud VisualSFM automatically saves the point cloud for you. CMVS is setup to use a limited number of images at one time for the dense cloud reconstruction. As a result there are often multiple .ply output files in the `[name].nvm.cmvs\00\models` folder. However, these multiple clouds are merged into a single .ply file in the folder containing your `[name].nvm.cmvs` folder. This file is called `[name].i.ply`, where i is the ith model. The .ply file contains both XYZ location values as well as RGB color values for each point in the dense point cloud. You can open this file directly into your favorite point cloud editing software.

A couple free point cloud editing software options we suggest are CloudCompare (Fig. 14) and MeshLab (Fig. 15). With these, and other point cloud editors, you can create 3D colored meshes from the point cloud, import XYZ files of surveyed points to compare against your VisualSFM point cloud, and if repeat data is available then differences of the point clouds can be computed. CloudCompare is an especially useful tool for this as it includes algorithms to compare cloud to cloud (C2C), cloud to mesh (C2M) datasets, and Multiscale Model to Model

Cloud Comparison (M3C2).

REMEMBER! Your dense point cloud that was generated with VisualSFM and PMVS/CMVS is in arbitrary coordinates and not aligned in real-space. Because we uniformly adjusted our GCP coordinates the relative locations of the points in our dense cloud are correct. Therefore, we can translate the dense cloud into real-space using the same equations we used to obtain the adjustd XYZ coordinate values.

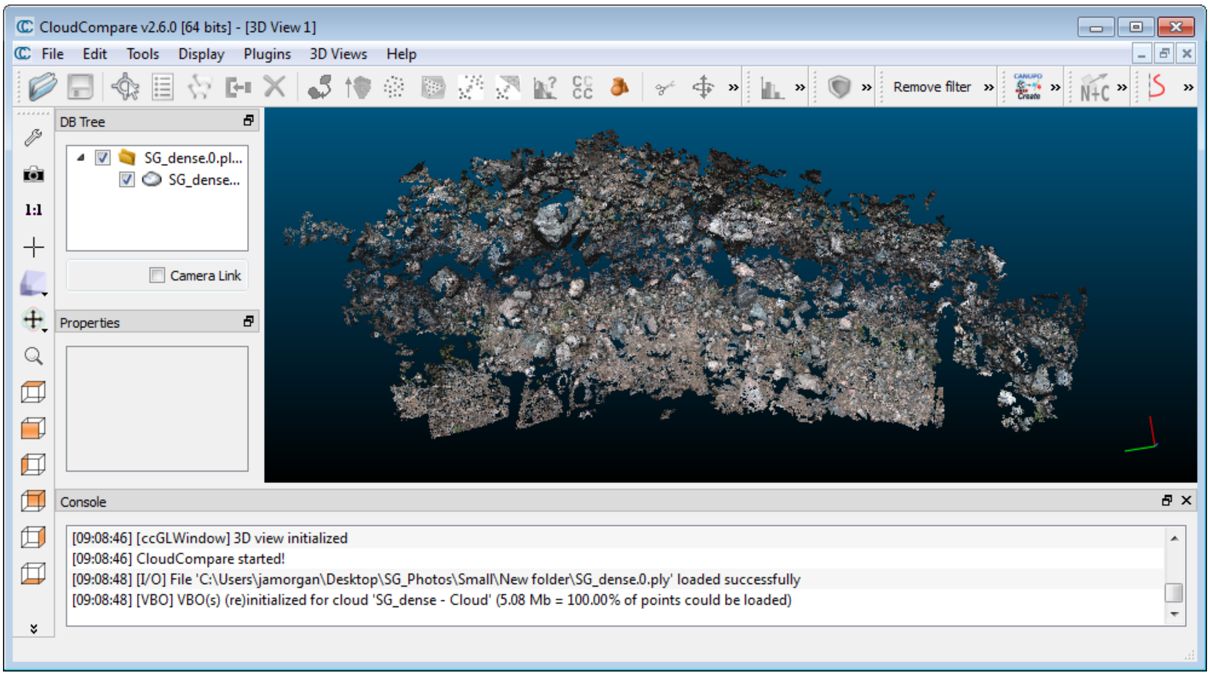

Figure 14: CloudCompare point cloud editor.

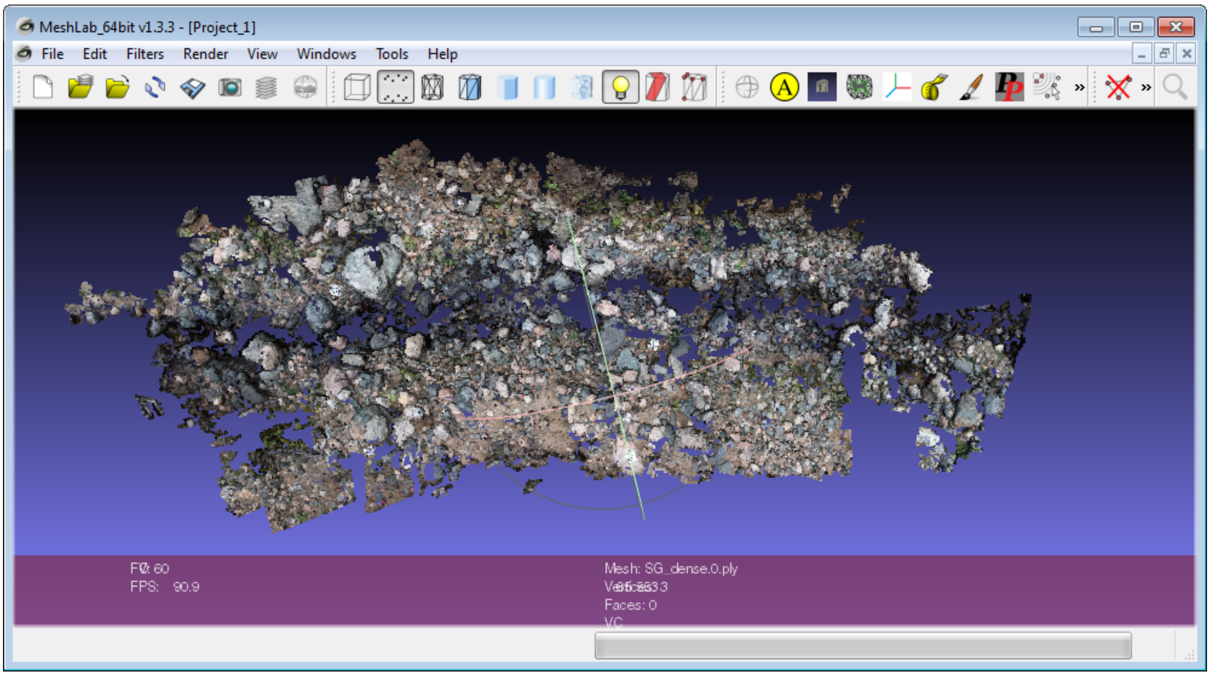

Figure 15: MeshLab point cloud editor.

4 Validation using CloudCompare

CloudCompare, developed by Daniel Girardeau-Montaut, is software used to view and process 3D point cloud and mesh data. It supports a wide variety of file types for both input and output. In this part of the tutorial we will use CloudCompare to measure the error of our dense point cloud from VisualSFM using RTK-GPS ground shots.

The CloudCompare wiki (`http://www.cloudcompare.org/doc/wiki`) is the best place for more information about CloudCompare. The user manual is also helpful; a PDF of the most current CloudCompare version can be found at `http://www.cloudcompare.org/documentation.html`. There is also a set of tutorial videos using example data freely available from OpenTopography.

One of the advantages of CloudCompare against other 3D data processing options is the fact that there is a plugin available to impliment the M3C2 method. The Multiscale Model to Model Cloud Comparison (M3C2) is a method of measuring changes between two point clouds developed by D. Lague et al. (2013, *ISPRS Journal of Photogrammetry and Remote Sensing*) that does not require meshing or gridding. It uses normal surface directions to compute local distances, and provides estimates of confidence intervals for each measurement.

4.1 Downloading and Installing CloudCompare

CloudCompare is available at no cost from the software website (`http://www.danielgm.net/cc`). The available download options are accessed through the "Download" link on the website homepage. If you use Windows the easiest way to download and install CloudCompare is to choose the installer version that corresponds to your Windows version (e.g. 32 or 64 bit). Double-click the .exe file to launch the installer tool. Step through the tool to select the location where you want CloudCompare installed and any options you would like included.

4.2 Importing Point Clouds

There are a few options for bringing point clouds into CloudCompare. The software has an open tool that is available through the shortcut at the top left (which looks like a folder), the file menu, or by pressing ctrl+O. From the dropdown menu at the bottom right choose the filetype of your point cloud. In our case the file type is .ply. Navigate to the location of your .ply file and press the "Open" button. However, the easiest way to import a point cloud is to drag and drop the file itself from its location into an open instance of CloudCompare.

When you bring a file into CloudCompare, depending on the file type, you may need to specify the format. For example, when you bring your .ply file into CloudCompare a "Ply File Open" window appears (Fig. 16). We will see and use a similar feature more when we bring in the RTK-GPS points, but for now we will accept the format CloudCompare has determined by pressing the "Apply All" button.

Once the point cloud loads into CloudCompare you can navigate around the scene using your mouse.

- left-click: rotate
- right-click: pan
- wheel: zoom

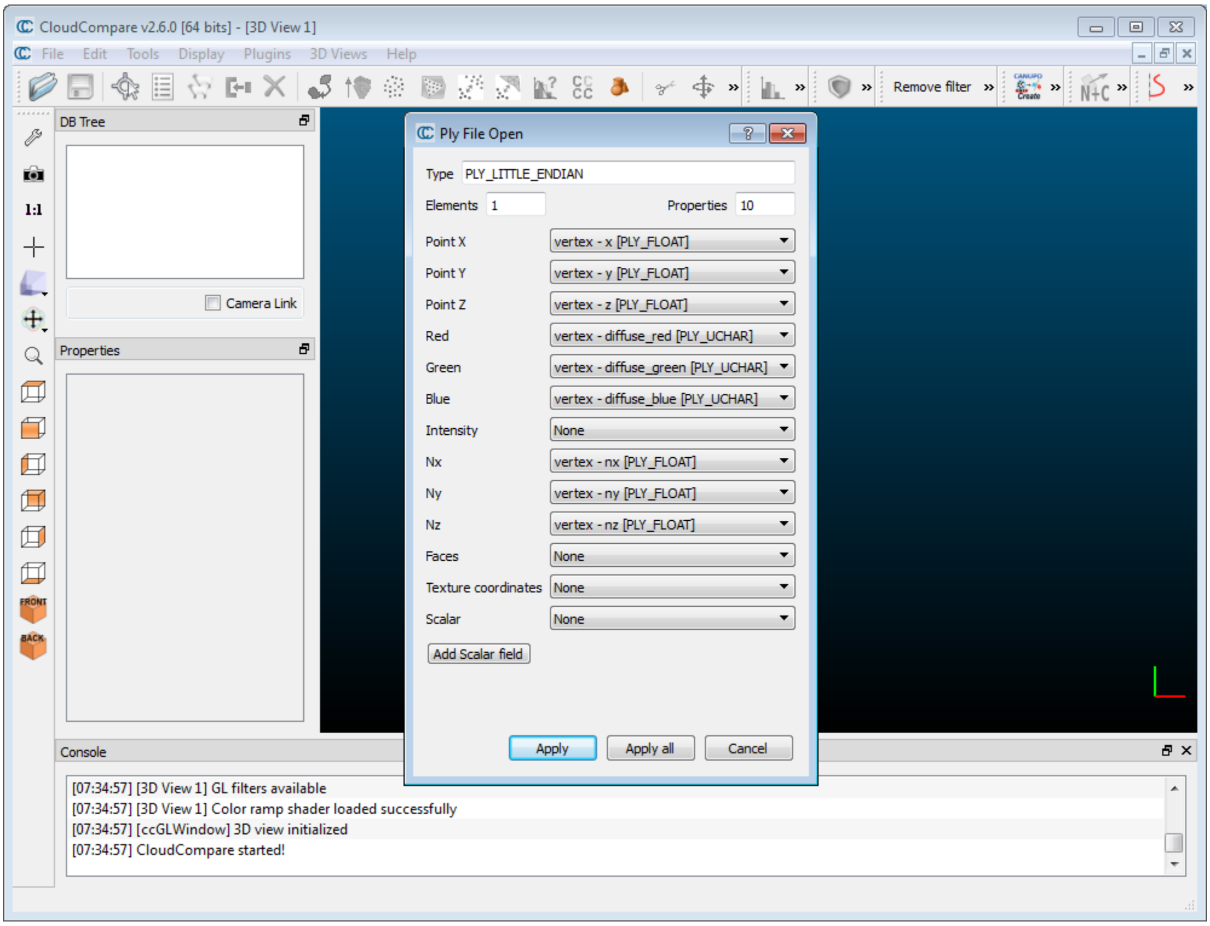

Figure 16: Ply File Open window.

After you feel comfortable moving about the scene you can import the file named `SG_RTKGPS.csv` using one of the aforementioned methods. This file contains the surveyed RTK-GPS points from Skin Gulch. Now an "Open Ascii File" window pops up (Fig. 17). Because our file type is a .csv (comma-separated values) we need to specify that our text file data are separated by commas. At the bottom of the "Open Ascii File" window there is a box to input the separator. You can either type "," into this box or press the button to the right with "," in it. Now we need to tell CloudCompare which columns correspond to the relevant parameters. Under each column number there is a drop-down menu where you can select a parameter. The default selection is to ignore all columns. For now the first row in each column has the column heading from the .csv file. We can use these headings to determine which column corresponds to which parameter. The first three columns are not data that are necessary for our purposes. In column 4 the heading is "Northing" so we need to select "coord. Y" from the drop-down menu. The heading for column 5 is "Easting" so we can select "coord. X" as that column's parameter. Finally, column 6 "Ht(G)" corresponds to elevation so select "coord. Z" from the drop-down menu. The "Open Ascii File" window should now look like Figure 18. At this point we need to tell CloudCompare to skip the first line of the file since it contains the headers. At the bottom left where it says "Skip lines" either enter the number one or press the up arrow once. Sometimes if you skip lines after specifying the column attributions the order changes. Make sure columns 4, 5, and 6 correspond to Y, X, and Z, respectively. Press the "Apply all" button.

Now a "Global shift/scale" window appears. If you'll remember, the target coordinates we used to orient and scale our point cloud in VisualSFM were translated from the original northing, easting, and elevation values. However the .csv file we just imported remains oriented with the

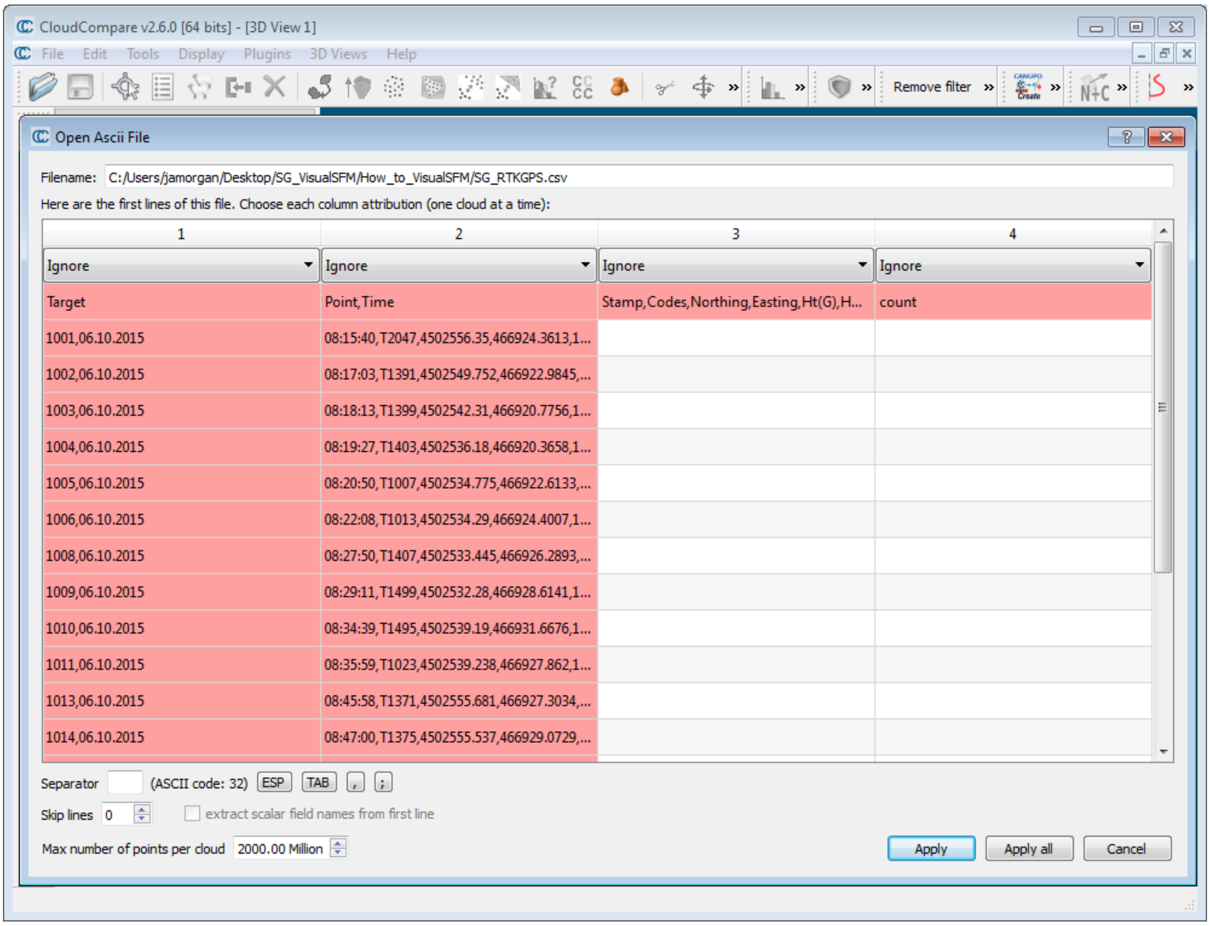

Figure 17: Open Ascii File window.

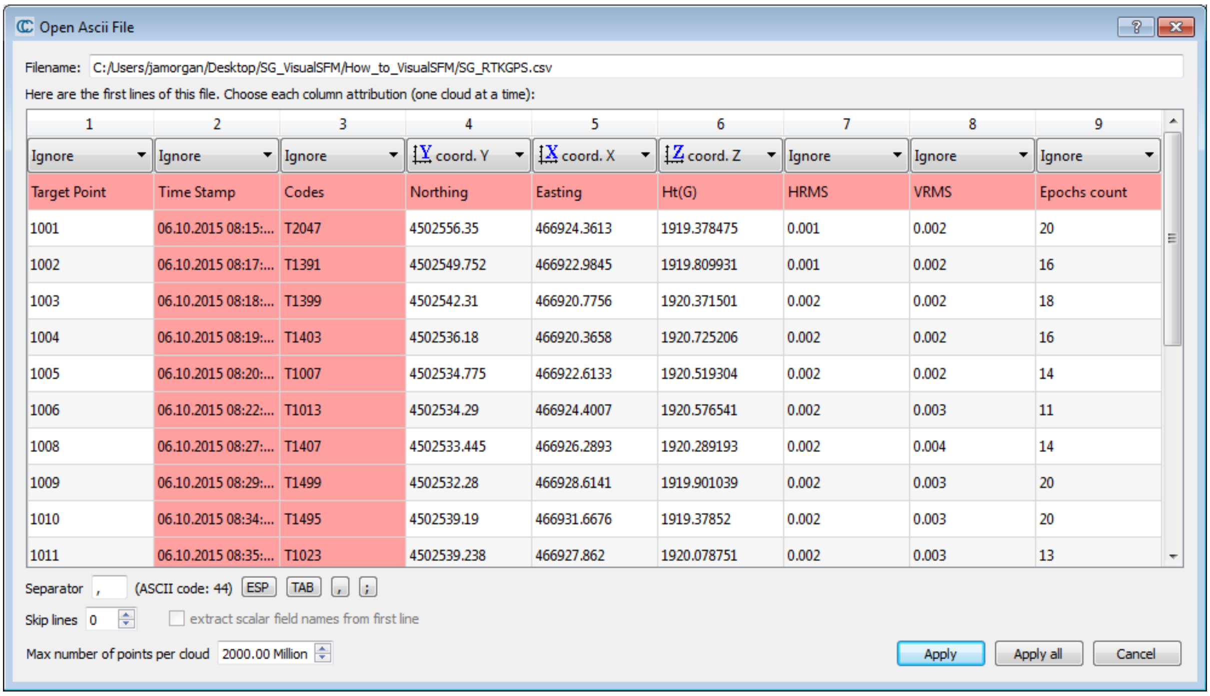

Figure 18: Importing the .csv file.

original northing, easting, and elevation values. Remember, we changed the XYZ coordinate values because VisualSFM does not really like a large number of significant figures for the coordinates. Because of a similar situation in CloudCompare we will re-orient the RTK-GPS points by translating them using the same numbers we used for the GCPs.

In the “Global shift/scale” window we need to apply Equations 1–3 to the RTK-GPS point cloud. In the top three input boxes type in the appropriate numbers needed to shift the point cloud. Your “Global shift/scale” window should look like Figure 19. Press the “Yes to All” button.

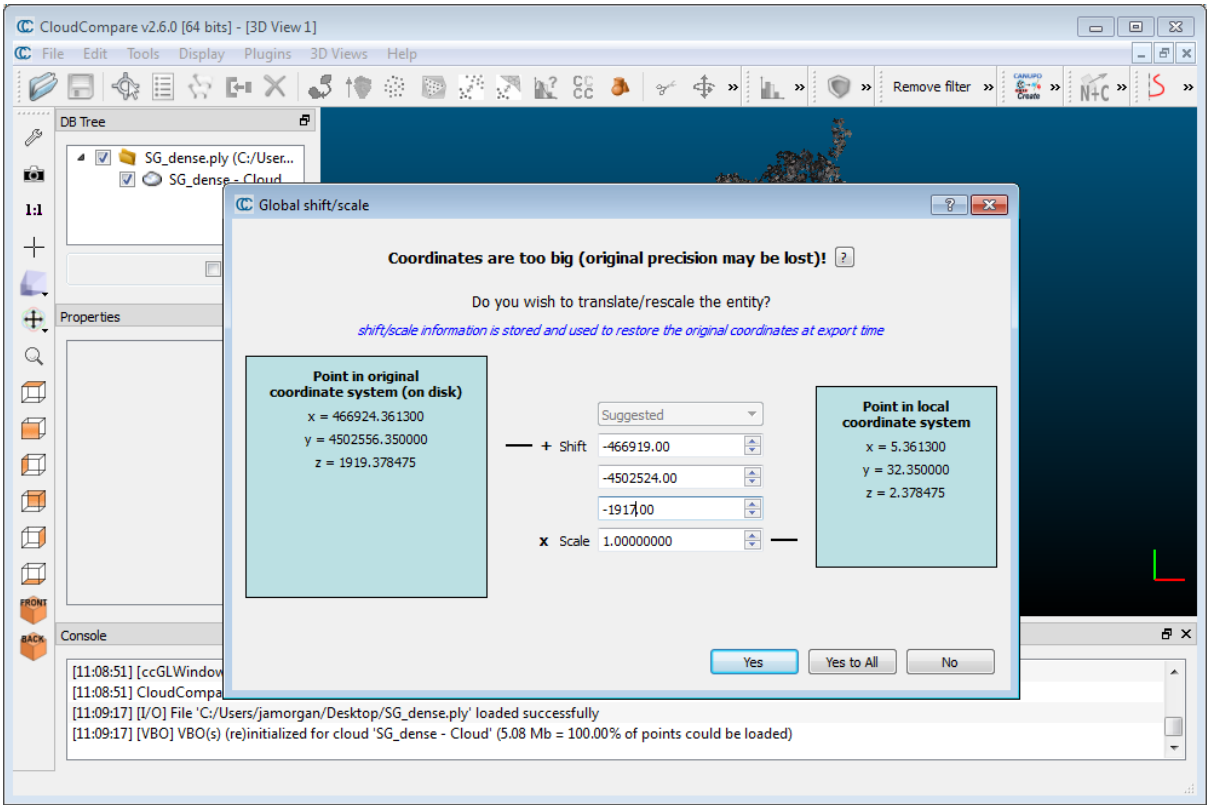

Figure 19: Global shift/scale window.

The RTK-GPS points should now show up in the viewer. Because the RTK-GPS cloud is sparse in comparison to the dense cloud from VisualSFM it may help in visualization to increase the point size. To do this, first select the RTK-GPS point cloud in the “DB Tree” window. Make sure you select the one with the cloud icon rather than the folder icon. In the “Properties” window scroll down to the “Point size” parameter and increase the value.

4.3 Comparing Point Clouds

We can now compare the two point clouds in order to get an accuracy estimate for our dense point cloud from VisualSFM. Start by selecting both point clouds in the DB Tree by clicking on one and then holding the ctrl key while you click the other. You can now reach the cloud-to-cloud (C2C) comparison tool by the shortcut on the menu, which looks like two point clouds with a red arrow between them, or you may also open the Tools menu from the menu bar at the top and in the Distances menu you can select “Cloud/Cloud Dist.”.

At this point a window appears prompting you to choose the role of each point cloud in the comparison. You must choose one to be the reference and the other as the compared cloud.

Generally you would choose your surveyed cloud to be the reference, however because our surveyed cloud has considerably less dense coverage and we are more concerned with the magnitude, rather than direction of the differences we will set “SG_RTKGPS - Cloud” to have the compared role and the VisualSFM point cloud as the reference. Press the “OK” button to run the comparison. A “Distance computation” window appears. For our purposes simply click the red “Compute” button and then the “OK” button to let CloudCompare decide the best parameters.

The colors of the RTK-GPS points now correspond to the C2C distance. It may be easier to visualize the results if the VisualSFM point cloud is not visible. You can achieve this by unchecking the box next to the cloud in the DB Tree menu. Because many of the RTK-GPS points are outside the area covered by our SfM point cloud we will apply a filter to our RTK-GPS cloud. There is a shortcut option that looks like a colorbar with the words “min” and “max” above and below it that corresponds to the “Filter points by value” tool. It can also be reached in the Edit menu under Scalar fields. Now input a minimum and maximum window to filter the point cloud. For our model all of the relevant points fell under a maximum value of 0.2. Therefore we kept the minimum at the default and set the maximum to 0.2. A new point cloud appears in the DB Tree window that is composed of the filtered point cloud.

To help visualize the differences you can select the filtered point cloud in the DB Tree window and in the Properties window scroll down to the “Visible” option under the “Color Scale” heading and check the box on. A colorbar should appear at the right. You can also view a histogram of the difference values with the “Show histogram” tool located on the shortcut menu as well as in the Edit menu under Scalar fields. In the histogram window you can use the mouse wheel to change the number of classes. For our model we achieved results indicating nearly all of our RTK-GPS points have distances <0.110 m, and most of them <0.041 m, from the SfM point cloud (Fig. 20).

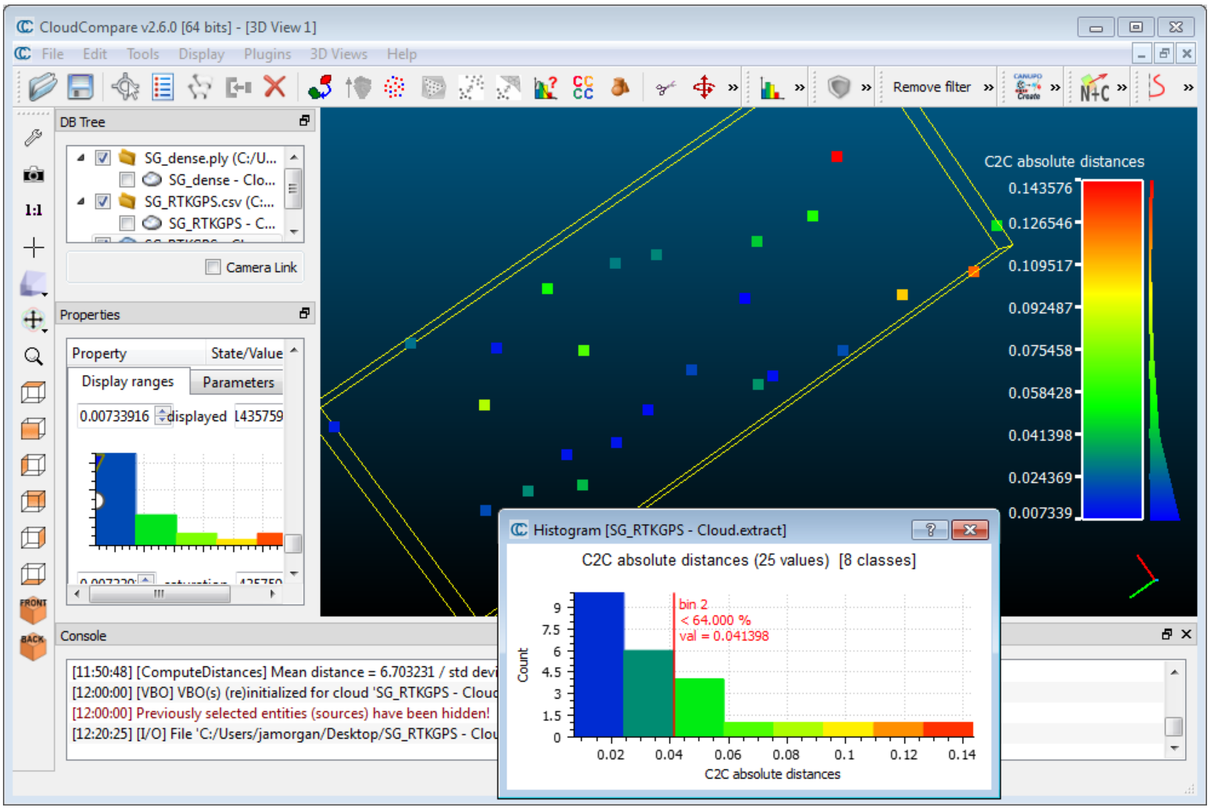

Figure 20: Cloud to cloud absolute differences.