Supplement of Nat. Hazards Earth Syst. Sci., 22, 3247–3270, 2022
https://doi.org/10.5194/nhess-22-3247-2022-supplement

Supplement of

Automated snow avalanche release area delineation in data-sparse, remote, and forested regions

John Sykes et al.

Correspondence to: John Sykes (john_sykes@sfu.edu)

DSM production in mountainous, forested terrain using SPOT 6 tri-stereo imagery with Ames Stereo Pipeline

1 Overview

This document summarizes our workflow to create a 5m DSM for high resolution potential avalanche release area modelling using SPOT6 tri-stereo imagery and open source software tools. The primary tools used are QGIS, GDAL, and Ames Stereo Pipelines (ASP). ASP requires a Linux or Mac operating system, detailed instructions on downloading and using the software are available from the developers (Beyer et al., 2019). Due to the computer processing resources necessary for creating high resolution DSM using ASP we utilized the Compute Canada research computing system (https://www.computecanada.ca/). The processing settings presented in this document are oriented towards that computer system.

Our DSM workflow is broken into three steps: 1. Pre-processing the imagery 2. Producing stereo DSM point clouds 3. Post-processing DSM output into a final product. The pre-processing covers taking the imagery from the raw format we received from the imagery providers and converts it into a stereo ready product. The DSM production briefly describes how we customized the ASP settings to produce an accurate DSM in forested and mountainous terrain. The post-processing details how we combined the output DSMs into a single mosaic and filled holes using existing DEM data. The data and processing settings from our research are available in our Open Science Framework (OSF) directory (https://osf.io/yq5s3/)

1.1 Imagery pre-processing

1.1.1 Image mosaic

Each of the images for our study area were delivered in two separate tiles, with the majority of the study area captured in one tile. In order to georeference and orthorectify the imagery with ASP we merged the tiles using the GDAL build virtual raster tool. The raw imagery files we purchased had no spatial reference system which makes it trickier to use standard GIS tools. The spatial reference data is contained in separate rational polynomial coefficient (RPC) files which were used as input to subsequent ASP processing tools to georeference the imagery.

1.1.2 Bundle adjustment

The ASP bundle adjustment tool allows users to input ground control points (GCP) to aid in accurately georeferencing image files. We collected a large dataset of GCP using a Trimble handheld DGNSS unit and post processed the points with the Trimble pathfinder software. The final GCP data set was composed of a point shapefile with attributes for the coordinates, elevation, and error estimates. To apply the GCP to our imagery required manually identifying their locations in each of the SPOT6 tri-stereo images. Our process of carrying this out was time consuming and required manually recording the pixel coordinates from the SPOT6 images for each GCP. We used Google Earth imagery and photographs taken during the GCP data collection to help locate precise GCP locations in the SPOT6 images. Including GCP in ASP required a text file with the following structure:

```
Point #, lat, long, elev, sd lat, sd long, sd elev, Image file, column, row, sd column, sd row
```

You can include multiple image files for each GCP with distinct column and row numbers for each image, in our case we had three images. To estimate the standard deviation for the GCP we used the error estimates from the GPS post processing software. We used the QGIS line measurement tool to estimate the standard deviations of the column and row numbers based on how accurately we could identify the location of each GCP. Here is an example line from our .gcp file, see 'SPOT6_GCP.gcp' in the OSF directory for complete copy of GCP file:

```
50, 50.877556716, -117.412691840, 2223.398, 0.2, 0.2, 0.2, $Image_1, 10062.57, 2545.97, 0.8, 0.8, $Image_2, 10951.96, 2425.2, 0.8, 0.8, $Image_3, 10943.38, 2595.82, 0.8, 0.8
```

We selected 27 out of 66 GCP for the bundle adjust tool based on the accuracy of the points and the distribution across the area and elevation of the study area. We evaluated the accuracy of the bundle adjust tool using the residual errors found in the '-final_residuals_no_loss_function_averages.txt' file. The mean residual error for our GCP bundle adjustment was 2.43 m. We detected a notable improvement in the visual alignment of the images compared to the GCP points. One challenge we experienced was that some of our GCP were located near steep terrain which led to higher residual errors for these points. However, we elected to keep these points in the bundle adjustment GCP data set because the visual alignment of the imagery was better overall compared to tests with smaller sets of GCP. Here is a sample of our use of the bundle adjustment for our images:

```
Bundle_adjust $Image_1 $Image_2 $Image_3 $RPC_1 $RPC_2 $RPC_3 --robust-threshold 10 -o $OUTPATH/CMHGLfinal_27GCPrt10 -t rpc
```

1.1.3 Point cloud alignment

Based on the recommendation of the ASP developers we processed our imagery through ASP multiple times to improve the alignment with our GCP data in steep terrain. In the first iteration we orthorectified our bundle adjusted imagery onto the ~18m Canadian national DEM (CDEM) prior to stereo processing. The output of this process, SPOT6_DEMv1, was used as the orthorectification layer in our second iteration. To improve the alignment between SPOT6_DEMv1 and our GCP we used the point cloud alignment (pc align) tool. This process involves converting the .gcp file into a point cloud by extracting the latitude, longitude, and height coordinates from each point. The SPOT6_DEMv1 layer and GCP point cloud are then aligned using the pc align tool and the inverse transform of the alignment is saved in order to apply the alignment to the reference layer, which in our case was the SPOT6_DEMv1. See Shean et al., 2016 and the ASP book for a detailed description of this process. See 'GCP_pc.csv' in the OSF directory for a copy of our GCP point cloud file. Here is a sample code for applying the pc align tool:

```
pc_align SPOT6_DEMv1.tif GCP_pc.csv --max-displacement 100 --save-inv-transformed-reference-points -o GCP27_align
```

The output of pc align is a point cloud which needs to be converted back to a DEM for input to the orthorectification tool:

```
point2dem GCP27_align-trans_reference.tif --dem-spacing 20 -o SPOT6_DEMv1.1_20m_align --threads 8
```

1.1.4 Map projection

The final step in pre-processing the imagery for the stereo pipeline is to apply the map project tool, which orthorectifies the images onto a local DEM. This process corrects the geometry of the raw images and improves pixel matches in steep terrain. As mentioned previously, we carried out this step twice. First using the CDEM as input and second using the aligned SPOT6_DEMv1 as input. This was suggested by ASP developers to improve performance in steep terrain. The map project tool uses the output of the bundle adjust tool as well as the RPC data from the raw imagery and a local DEM to correct the geometry of the images. Here is a code sample for one of the images from our study area:

```
mapproject -t rpc --mpp 1.5 --ot UInt16 --bundle-adjust-prefix $GCP_BA_path $SPOT6_DEMv1.1_20m_align.tif $Image_1
$RPC_1 $Output_Image
```

In summary our pre-processing workflow creates a single mosaic from the raw images, uses GCP to accurately georeference the images (bundle adjust), aligns our orthorectification DEM to the GCP point cloud (pc align), and orthorectifies the imagery using the aligned DEM (map project). We developed this rather complex workflow to help improve the performance of ASP in forested and steep terrain, which makes up the majority of our imagery. Our impression is that overall we saw large improvements in these areas thanks to the careful pre-processing.

1.2 Stereo

We tested a variety of stereo input parameters to optimize performance in our study area. The best performance was achieved with the smooth semi global matching (SGM) stereo correlation algorithm using the following input parameters: stereo algorithm (2 - smooth SGM), cost mode (3 - census transform), kernel size (3, 3), median filter size (5), texture smooth size (25), and texture smooth scale (0.15). We chose to use parameters that created a smoother DSM with less artifacts and overall surface roughness because our application is modelling of avalanche release areas, where the snowpack naturally creates a smoothed ground surface. For applications where detecting small scale bare ground surface roughness is desirable we recommend decreasing the median filter size, texture smooth size, and texture smooth scale parameters. A copy of our stereo default file is available in the OSF directory to reproduce the stereo settings that we used.

We processed each pairwise combination of the SPOT6 tri-stereo imagery, which resulted in six output DSMs. This approach takes advantage of the three viewing angles of the tri-stereo imagery and aims to minimize the number of DEM holes due to shading and cloud cover. Using the Compute Canada cedar research computer cluster we allocated 48 CPUs working on 12 parallel processing tasks and roughly 190 gb of RAM to generate the DSMs. For more details on parallel processing efficiency for high resolution stereo imagery see the ASP book or Shean et al., 2016.

1.3 DSM post-processing

1.3.1 Point cloud to DSM

To process the six output point clouds from the stereo tool into DEM layers and remove pixels with triangulation errors greater than the original image pixel size (1.5 m) we used the point to DEM tool (point2DEM). Our output DEM had a resolution of 5m based on the requirements of our PRA modelling and the original imagery resolution. We used the hole filling function of point to DEM to interpolate across small holes in the DSM layers up to 10 pixels. To quantify the errors of our stereo workflow we also produced an error image using point to DEM which produces an additional image whose values represent the triangulation ray intersection error in meters. Here is a sample of our code:

```
point2dem $PC_1 --dem-spacing 5 -o $OUTPATH/DEM1 --errorimage --threads 8 --max-valid-triangulation-error 1.5 --dem-hole-fill-len 10
```

1.3.2 DSM mosaic

To merge our 6 output DSMs we used the DEM mosaic tools with the default blending type. We tested several different approaches for blending and hole filling using this tool and found that simply applying the default method to the entire list of 6 DSM created the most consistent and artifact free output. We also generated normalized median absolute deviation (NMAD) and standard deviation (SD) layers between the 6 input DSMs using this tool.

```
dem_mosaic --threads 8 $DEM_1 $DEM_2 $DEM_3 $DEM_4 $DEM_5 $DEM_6 -o $OUTDIR/DSM_mosaic.tif
```

After generating the DSM mosaic we manually identified artifacts and errors in the final output by creating a polygon shapefile. We used the shapefile to remove portions of the DSM mosaic with significant errors by clipping the DSM mosaic raster with the polygon dataset. Most of the areas with errors and artifacts either had cloud cover in multiple images or were shaded and very steep north facing terrain.

1.3.3 Point cloud align

With our DSM mosaic complete and any artifacts removed we then performed a final iteration of pc align using the DSM mosaic as the reference layer and the GCP point cloud as the source layer. Using the same process described for creating the orthorectification DEM, we saved in inverse transform and applied it to the DSM mosaic. This further improves the alignment of the DSM mosaic to the GCP. We also performed pc align with the Canadian national DEM to improve the alignment to our DSM, using the aligned DSM mosaic as the reference layer and the Canadian national DEM and the source layer.

1.3.4 DEM mosaic

The final step in our DSM pipeline is to fill in holes with the best available DEM. In our case that is the Canadian national DEM (CDEM) which has a native resolution of ~18m. Prior to aligning the CDEM with the DSM mosaic using pc align, we down sampled the CDEM to 5m to match our DSM resolution and pixel spacing. We then used DEM mosaic to blend the final DSM and CDEM together within a 60 m buffer of any remaining holes. We tested a wide range of blending lengths to determine the optimal balance of preserving the accuracy of the DSM while avoiding artifacts around the filled holes. This process inevitably results in smoothing of the DSM mosaic in the areas adjacent to holes, but it was necessary to avoid abrupt transitions between the boundaries of the DSM layer and the CDEM. For our modelling purposes abrupt transitions would result in unrealistic output from dynamic avalanche simulation software, which is more of a concern than having smoothed values around DEM holes.

```
dem_mosaic $DSM_mosaic $CDEM_5m --priority-blending-length 60 --weights-blur-sigma 5 --weights-exponent 2 -o
$OUTPATH/SPOT6_DEMv2.tif
```

References

Beyer, R., Alexandrov, O., and McMichael, S.: NeoGeographyToolkit/StereoPipeline: Ames Stereo Pipeline version 2.6.2, https://doi.org/10.5281/zenodo.3247734, 2019.

GDAL Contributors: GDAL/OGR Geospatial Data Abstraction software Library., 2021.

QGIS Development Team: QGIS Geographic Information System, [online] Available from: http://qgis.org, 2021.

Shean, D. E., Alexandrov, O., Moratto, Z. M., Smith, B. E., Joughin, I. R., Porter, C., and Morin, P.: An automated, open-source pipeline for mass production of digital elevation models (DEMs) from very-high-resolution commercial stereo satellite imagery, 116, 101–117, https://doi.org/10.1016/j.isprsjprs.2016.03.012, 2016.

Sykes, J., Haegeli, P., and Bühler, Y.: Automated snow avalanche release areas delineation in data sparse, remote, and forested regions -- Code and Data, https://doi.org/10.17605/OSF.IO/YQ5S3, 2021.