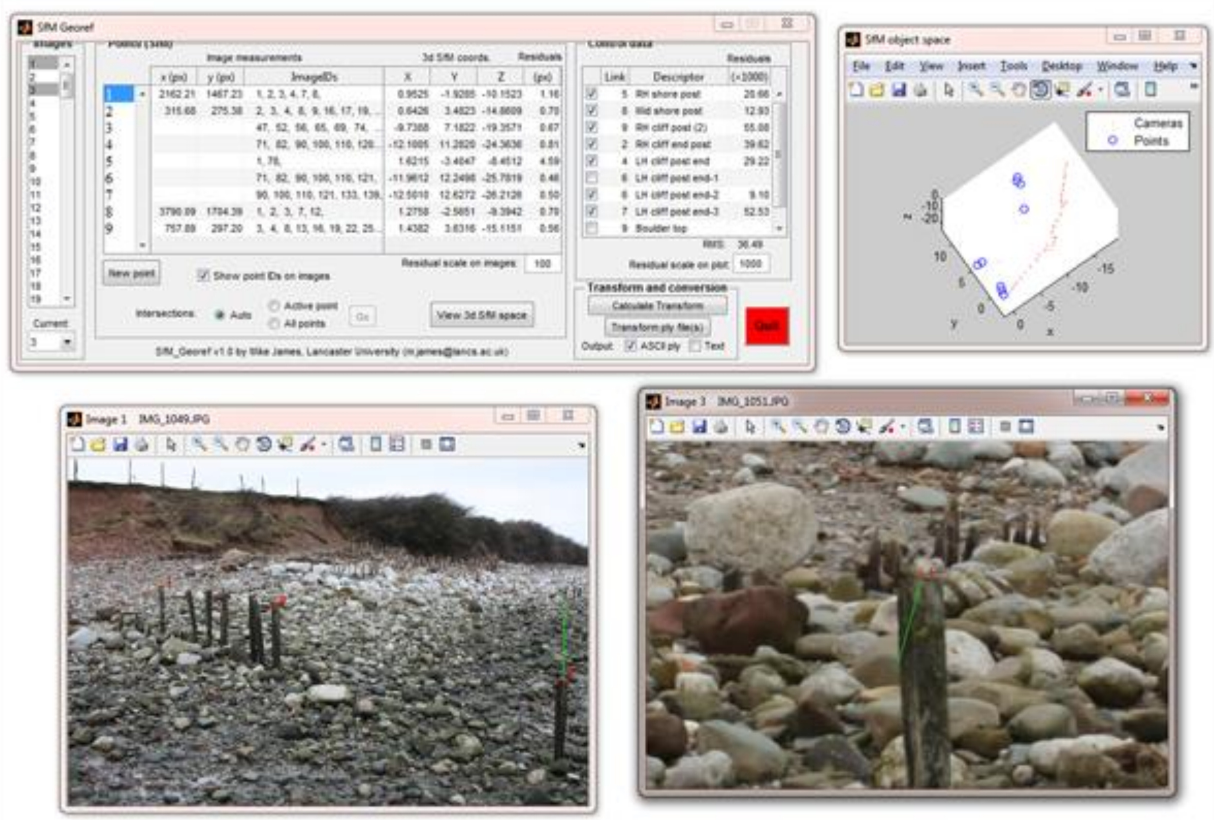
SfM_Georef v.3.1

http://www.lancs.ac.uk/staff/jamesm/software/sfm_georef.htm

Mike James, Lancaster University
m.james@lancaster.ac.uk



Index

1. Introduction

Sfm_georef is a gui-based tool for scaling and orienting SfM point clouds to real-world coordinates, using observations made directly in the SfM image set. It also provides a number of extra functions for analysis of Monte Carlo results (which do not require loading of an SfM project directly). Sfm_georef does not carry out any SfM reconstruction, but uses the camera data output by other software to enable point clouds (as .ply files) to be scaled or geo-referenced. Output from the following SfM software or reconstruction pipelines are supported:

- Photoscan (http://www.agisoft.com/, v.1.3, with some limitations)
- Bundler Photogrammetry Package (http://tacticalspace.org/archives/bundler-photogrammetry-package/)
- SFMToolbox (http://www.visual-experiments.com/demos/sfmtoolkit/)
- VisualSFM (http://ccwu.me/vsfm/index.html)
- OSM-Bundler or Python Photogrammetry Toolbox (http://code.google.com/p/osm-bundler/)

For scaling a project, at least one distance (a control length) between positions that can be directly observed in two or more images is needed. For scaling and orienting a project (full geo-referencing), the real 3-D coordinates of three or more points (control points) that can be observed in two or more images, is necessary, or coordinates for the camera positions. Once the transform is determined, this can be applied to dense point cloud (.ply) files generated by PMVS2, with provision given for merging multiple files (as produced by CMVS) into one.

SfM_Georef v3.1 is written in Matlab (release 2015a, Win.7 64-bit) and has been compiled as a stand-alone executable. To run the executable, you will need to download and install Matlab runtime libraries (see instructions below). NOTE: This is a newer compiler version than used for versions of sfm_georef prior to v.3.0, so if you used an earlier version, you may need to get the newer runtime libraries.

Use of sfm_georef for georeferencing a project can be divided into three stages:

1) Reading the SfM project and importing control measurements
2) Identifying the control points in the image set, calculating the equivalent 3-D positions in the SfM coordinate system and calculating the transform that relates the SfM and the control coordinate systems
3) Transforming and merging dense point clouds associated with the SfM project.

If you only need the Monte Carlo processing routines, install then jump to Sections 9 & 10 for:

1) Assessing the influence of GCP quality and quantity on your SfM project (James *et al*. 2017a).
2) Determining point coordinate precision estimates across an SfM survey, and using them for confidence-bounded 3-D surface change detection (James *et al*. 2017b).

2. Installation (stand-alone executable)

1) Download and then uncompress the file sfm_georef_3.1.zip to a folder. The executable itself (sfm_georef.exe) needs no further installation.
2) Download and install the Matlab Compiler Runtime. For sfm_georef v3.1 the version required is R2015a (8.5) , 64-bit and can be downloaded from:

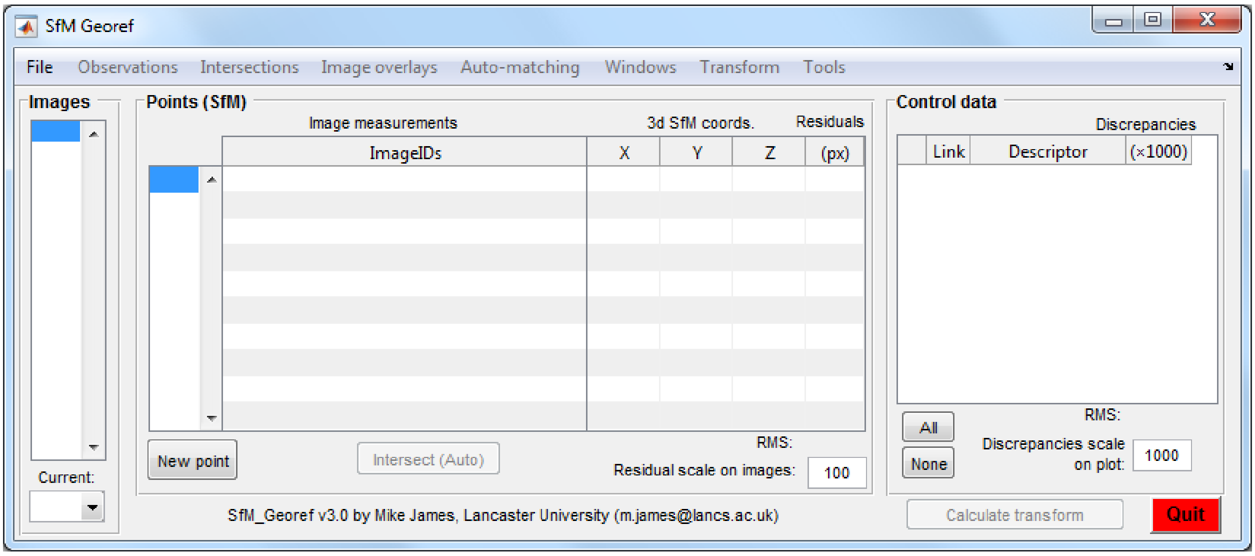
   http://www.mathworks.co.uk/products/compiler/mcr/

   Sfm_georef will not run unless the correct runtime version is available. If you already have other versions of Matlab Compiler Runtime installed, you will still need to download this one, and multiple versions can co-exist without difficulty.

3. Getting started

Start sfm_georef by double clicking on the .exe file (Windows); the main control window will appear as well as a console window (black). The console window will display any error or other information, as you use sfm_georef.

The main SfM_Georef control window is divided into a menu bar and three main panels:



The main panels:

Images – controls which images are visible and which image is ready to make measurements in

Points (SfM) – records image observations, controls intersection strategy and gives 3-D point coordinates in SfM coordinate system

Control Data – Lists the control data imported, allows point selection for transform and shows the resulting residuals

Additionally, there are buttons for:

Calculate transform – Determines the transform between SfM and control coordinate systems.

Quit – Use this to exit sfm_georef. Warning – this does not yet prompt you to save any data… unsaved data will be lost

To geo-reference a new SfM reconstruction, the appropriate files need to be loaded using the File menu command. For:

SFMToolkit or Bundler Photogrammetry Package projects:

File → Import project folder; select the folder that contains your images. For Bundler Photogrammetry Package projects this will also have a subfolder 'bundler' with the file 'bundle.out'. For SFMToolkit projects, bundle.out will be in a 'bundler' subfolder. The import also reads the file list.txt in the image directory. If any these files are not present, the import will fail. [Note – I have not used this for a long time, and bugs may have developed]

VisualSFM or Photoscan:

File → Import project file; select the .nvm or .psz file from your project. If you loaded a Photoscan project that contains markers and control points, these will also be imported. However, the 3-D points associated with the image observations will not be used, so all observations will have to be intersected (check the Intersections → All points menu, then click the Intersect (All) button in the Points (SfM) panel).

If all the appropriate data are found, then you will see no error messages, the main sfm_georef window will become populated with the image numbers found in the project, the first image will be displayed as well as a 3-D plot of the relative camera positions in the project.

1) Image window, showing image 1

   Image windows are where observations (mouse clicks) in the images can be made. Zooming in/out in the image can be carried out by scrolling the mouse-wheel, and panning by right-click-and-drag.

   Alternatively, use the standard Matlab buttons  to navigate around and zoom in to the image – click the button once to activate a navigation tool, click again to deactivate. [Note: Use of the other buttons and menus is not directly supported and may produce undesired behaviour]. To define an observation of a point, locate the point in the image, *ensure that none of the Matlab image navigation tools are active and that you have a cross-hairs cursor*, and click. A red cross will mark the location. Note – if the SfM project did not contain any orientation information for the image, observations are not allowed.

2) SfM object space 3-D plot

   This window can be brought up at any time from the main menu (Windows → 3-D SfM space). The plot gives the 3-D positions of cameras in the SfM project coordinate system. Camera positions are marked as red dots and any intersected point locations are given as blue circles. Use the standard Matlab buttons to navigate around the plot.

Saving your sfm_georef project:

All measurements, transformations, control and the image set can be saved as a sfm_georef project file via the File →Save SfM_Georef project as menu item. The file produced is a binary Matlab data file, that can be read by Matlab or by sfm_georef.
To keep file sizes small the images themselves are not stored, but their locations are recorded as full paths. If images or SfM folders are moved, the images will have to be relinked. Furthermore, if sparse points have been read, the default is not to save them. Enable sparse point saving by checking the File → Import/save options → Save sparse feature data menu item. Once the current project is saved, File →Save SfM_Georef project can be used to update the project file.

Loading a pre-existing sfm_georef project:

A previous project can be loaded via the File →Import Control menu item.

4. Importing control data

Control data can be imported via the File →Import Control menu item. Control data can be supplied to sfm_georef in different formats, but you can start with a text file containing the data in three tab or space separated columns, described by a header line. For example, for 3 control points with (optional) descriptors:

```
x        y     z     descriptor
1.122    2.335 0.363 target_1
0.204    34.24 8.929 target_2
9.324    5.252 3.443 target_4
```

Note: Descriptors cannot contain characters that could be determined as delimiters (e.g. spaces or tabs – 'target_1' is OK, 'target 1' would produce an error).

If control distances have been measured rather than point coordinates, these are provided with the header 'distances'

```
distances      descriptor
6.993    tar_1_to_tar_2
3.442    tar_1_to_tar_3
9.322    tar_2_to_tar_3
```

If camera positions are being supplied, then you need the same number of lines as there are cameras (i.e. a position for each camera). The order of the lines must match the order of the cameras read in sfm_georef. The syntax is:

```
cam_x    cam_y cam_z descriptor
1.122    2.335 0.363 photo_1
0.204    34.24 8.929 photo_2
9.324    5.252 3.443 photo_3…
```

Control data can also be directly imported from previously saved sfm_geof projects by importing the project file via the File →Import Control menu item. In this case, only the control information is imported from the read project, all other data are neglected.

5. Identifying points in the SfM image set and making point observations

To scale or geo-reference the SfM project, the control points need to be identified in the images so their coordinates in the SfM coordinate system can be determined. (Note this is not required if you provide camera positions as control data).

1) Identify a control point that will be visible in multiple images.
2) Show the appropriate images by selecting their image numbers in the Images list box, using control-click or shift-click as appropriate for multiple selection.
3) Bring the first image to use to the front by either clicking on the window frame or by changing the 'Current' drop down box to the required image number.
4) Locate the feature in the image and click. The image coordinates will be stored as observations of Point 1, and listed in the 'Image measurements' table.
5) Locate the feature in another image and mouse-click to make the observation. The second observation will be added to the table. With two observations, an intersection will be calculated; if successful, the 3-D coordinates will be provided in the '3-D SfM coords' column.

Note:

Image residuals are calculated by projecting the 3-D position of the determined point onto the image plane. Image residuals are shown by green lines on the images, and their scale can be changed by changing the number in the 'Residual scale on images:' box. The default of 100 draws residual vectors that are 100x as long as the real image residuals. Usually, residual values are around 1 pixel or smaller. Significantly larger residuals indicate that either the image has not been incorporated well in the SfM project (i.e. the imported camera model or the camera orientations are inaccurate) or – more usually – that one of the image measurements is actually of the wrong feature.

The residuals column in the table gives the RMS residual value for all points.

You can show the point IDs on the image windows from the main menu (Image overlays → Point IDs – useful if you have many points.

To add more points:

To add another point, click the 'New point' button. Another point will be added to the point list box, and will be highlighted. Any observations in image windows will now be attributed to this point. To make further observations of a different point, **make sure that the point number is selected in the point list box**.

To delete an observation:

Make the image in which you want to delete the observation the current image, select the point number in the point number list box, and press the 'delete' key. Note – the point number list box must be the active control (i.e. the last control clicked on) for the delete to be successful.

Intersections:

By default, intersection calculations are automatically updated for each change in the image observations. If intersections want to be carried out manually, select the appropriate strategy from the Intersections menu. For either Intersections → Active point or Intersections → All points, intersections are only carried out when the Intersect button below the Image measurements table is clicked.

Current image:

Use this list box to change the image number that is currently active (i.e. the image in which measurements are expected). Useful when you have many windows open and the one you want is buried.

Organising control data:

Control data are listed in the order supplied in the input. Descriptors (very helpful) can be added or edited by typing in the Descriptor column.

The first column in the table is the 'active' column. If ticked, the control point is active and, if it has a valid link to a point in the SfM project, will be used in the calculation of the transform.

Unselect any control points that you do not want to be included in the calculation.

The Link column indicates which point in the SfM project (i.e. in the Points SfM panel) represents the same feature as the control point defined by that row. The transform will be calculated to minimise the square of the distance (i.e. least squares) between the transformed SfM points and their linked, active control points.

Residuals are given when a transform has been calculated, and represent the 'mismatch' distances between the control points and the transformed SfM points, in the control coordinate system. In the table, the residual units are given as a thousandth of the units used for the control coordinates. For example, if control data are provided in metres, residuals are quoted in millimetres.

6. Transform and conversion

The Control data table comprises four columns:

Tick box – indicates whether or not this control measurement is to be used in calculations of scale or geo-reference transform

Link – gives the ID number of the SfM point that represents this control point. If using control distances, each distance measurement is represented by two lines in the Control data table, giving the IDs of the two points between which the distance measurement has been made

Descriptor – a text description of the control point.

Residuals – gives the mismatch between the control data and the transformed SfM points, once a transform has been determined.

If sufficient SfM points and active control exist for a scale or 3-D transform to be calculated the 'Calculate scale/transform' button will become active. Scale can be determined from one or more distance measurement or two 3-D points, full geo-referencing requires three or more 3-D points.

If the transform is successful, residuals will be calculated and a 3-D plot given, showing the control points and the residuals. The File → Export menu item will become active allowing conversion of:

SfM points – The transformed coordinates of the points identified and intersected in the SfM project are exported to a text file

SfM project – The transform can be applied to a bundler file, with the output written as a new Bundler-format (.out) file. Select the Bundler SfM file to read, then a new filename for the ouput. NOTE: The read Bundler file should be the same one used for the sfm_georef project – only the sparse points are extracted and are then combined with the camera data in the current sfm_georef project. The SfM project transform is not implemented for VisualSFM projects.

Ply files – Points in .ply files can be imported, transformed and then written back to a .ply file. If multiple .ply files are selected then they are merged and the output written as one file. This feature enables the dense point clouds output by CMVS/PMVS to be transformed to the geo-referenced coordinate system. Note only the ASCII ply files written by Bundler or PMVS are explicitly supported.

7. Auto-matching

From v.3.0, sfm_georef offers an auto-matching facility to aid collection of control (or other) point observations. Auto-matching searches other images for a point identified in one image, using a normalised cross-correlation method. When control points are natural features or asymmetric targets, auto-matching can be used on a point-by-point basis to accelerate and improve the precision of point identification in images.

Auto-matching is enabled by selecting one of the following options in the Auto-matching menu:

Open images only – only other currently open images are searched

Images with existing observations – any image with an existing observation of the active point are searched, and their observations refined.

All potential images – SfM_georef calculates which other images could view the point and searches them. Note, this may involve loading many images (slow), and occlusions are not accounted for.

The auto-matching process uses a square region of pixels located around a point observation in a base image (the base image is the one clicked in). These pixels are reprojected onto an oriented 3-D patch to form the texture to match in other images. The patch orientation is described by a normal vector, determined from a local best-fit plane to nearby sparse feature points if they are available or, if not, by the viewing direction to the base image. The size of the image patch used in the base image can be set by Auto-matching → Set match patch size, which should be set to a suitable half-width value (i.e. a patch radius) that captures the texture required. In the Image IDs table, base image IDs are distinguished from other images by being listed first, followed by a colon.

7.1. Fully automatic target searching

If symmetric and identical control targets have been deployed, auto-matching can be used for a fully automated target search. This is carried out by identifying a control target in one image as a generic template, which is then used to search for all other targets. So that entire images do not have to be searched, this approach also requires the 3D coordinates of all control points and the transform that links the SfM and Control coordinate systems to be defined. The full workflow is described below.

When target observations have been identified through use of a generic template, base image IDs are not automatically assigned. However, this can be done subsequently by Auto matching → Define 3D patches. This selects the image in which the patch is seen at highest resolution as the base image, then carries out an auto-matching again (i.e. without using the generic template). See the workflow below.

7.2. Example workflow for auto-matching and target searching

The following steps detail how to use template-based auto-matching in a project, for example, for determining image observations of ground control points (GCP targets) in a UAV survey. To run through the example, you need an SfM project and a control point data file.

1. Start sfm_georef.
2. Ensure that File → Import/save options → Read sparse feature data and Save sparse feature data are both checked (this is the default). Import your SfM project via File → Import SfM project folder or Import SfM project file for VisualSfM and PhotoScan projects.
3. Import your control data, File → Import control data.
4. Save the sfm_georef project, File → Save SfM_Georef project as…'
5. If your computer has sufficient available memory, then you can load all your image files at this point, to make things quicker later on. To do this, File → Load all image files.

With all the project information loaded, the next step is to make observations of three control points, so that an initial transform can be calculated

6. Open an image where you can see a control point, and zoom in so that you can see its size (in pixels).
7. Now set the image patch size that will be used in the matching, via Auto-matching → Set matching parameters, and set match template size. The size value used should be given in pixels and represent the required patch radius.
8. Switch on the auto-matching by checking Auto-matching → All potential images

9. Return to the open image and click on the centre of the control point. In the Messages window, you should see a number of other observations being made in other images (assuming it is visible in other images), and then listed in the Image Measurements table.
10. In the sfm_georef main window, add a new SfM point, locate another ground control point in an image, zoom in and click on its centre.
11. Repeat the previous step for a third point. You should now have three points identified, hopefully with sub-pixel image residuals. If residuals are large, go to Step 16 below, where approaches are given for identifying and removing bad observations are described.
12. Now link these three SfM points to the appropriate control point, by changing the links in the Control data panel. With the appropriate links made, calculate the transform, and check that the discrepancies on the control points are acceptably small. If they are not, double check you have correctly linked the image observations to the appropriate control points.

With an initial transform to describe how the SfM coordinate system relates to the control coordinate system, the locations of other control points can be used to guide automated searches for them in the images.

13. Before starting full-auto matching, a template is required that determines what a control point looks like. A template is determined by providing an example observation (i.e. an SfM point ID and an image number) of a good image of a ground control point. Identify one from the points you have used so far, then enter the numbers into the dialog box given by: Auto matching → Target-based matching → Set target template
14. Ensure that 'Auto-matching → All potential images' is checked, then start the matching process by selecting Auto matching → Target-based matching → Auto target search. Observations will be incrementally collected for each ground control point. Note that if your control points are numbered, the observations will be correctly linked to them, but will not necessarily be numbered sequentially. To organise the new point observations so that they are sequential with the control points, use Observations → Re-order to simplify control links.
15. In most projects, there will be some gross errors in the automatically generated observations, and usually these can be filtered when an additional matching step is run to define a 3D patch for each SfM point. To do this, Auto matching → Define 3D patches. On completion, the image IDs listed in the SfM points panel will start with one image ID followed by a colon. This is the base image used for that patch.

There may well still be some errors in the collected observations, which can be shown as either a lack of observations or some observations with very high residuals.

If an SfM point has no observations, the full automatic search has effectively failed for this point. However, observations may be recovered by locating the target in one image:

16. Determine which ground control point is missing observations from the Control data panel (it will not be associated with a transform discrepancy value, and will be linked to the SfM point with no observations). Then open an image window with an image that should cover this point.
17. You can help estimate its location by overlaying projected positions and control descriptors: Image overlays → Projected control points and Projected control point IDs. Zoom in to the appropriate region and, ensuring that the correct SfM point is active, click on the centre of the target. All other observations should be automatically made.

If an SfM point has large image residuals:

18. Right-click on the SfM point ID number to bring up a context menu, and select Show summary of all observations.
19. In the summary window that appears, control-click in any image that shows a bad observation – this will delete the observation. The RMS residual value will be recalculated and updated in the table.
20. Identify outliers, run summary.
21. Open main image, show projected point positions and IDs. Verify where the point should be.
22. Delete all observations,
23. Select one with elevated residuals

8. Doming analysis (systematic error)

From v.3.0, sfm_georef provides a tool to assess systematic error in projects where the reconstructed surface is dominantly horizontal (e.g. surface topography reconstructed from UAV imagery). Error in the description of radial lens distortion in such projects can result in systematic Z-error that shows as warped or domed surfaces. If a project shows such a deformation, and has sufficient control points to characterise the error, the approach of James & Robson (2014) can be used to improve the radial distortion estimate. See the paper for further details:

James, M. R. & Robson, S. (2014) Mitigating systematic error in topographic models derived from UAV and ground-based image networks, *Earth Surf. Proc. Landforms* (accepted, May. 2014), doi: 10.1002/esp.3609

To get a metric for doming magnitude, ensure that your project is loaded and geo-referenced with control points (calculate transform).

1) Visualise the distribution of vertical error by Windows → 2-D control space (x-y). Vertical error is indicated by the colours, which should give some sort of ring (target-like) representation.
2) From the main menu again, Windows →Doming analysis brings up the analysis window. The plot shows the vertical error at control points against the distance of points from a nominal centre of deformation. The data are fitted by two models, a polynomial to best fit the data, and a straight line, the gradient of which can be used to give a metric of the deformation magnitude.
3) The initial centre of deformation is estimated purely from the x-y extent of the control points, and can be manually changed in the coordinate edit boxes. Alternatively, the Optim. button will search for a centre that optimises the fit of a polynomial model. Clicking the Optim. button will adjust the centre coordinates to maximise the R^2 value for the polynomial fit. The gradient of the linear fit can then be used as a metric for the magnitude of systematic deformation (see James & Robson, 2014).

9. Monte Carlo analysis of ground control

These processing functions do not require an SfM project to be loaded into SfM_georef, and are purely for post-processing of the results of Monte Carlo GCP analysis carried out in PhotoScan as described in James*, et al.* (2017a). The processing provides insight into GCP and survey performance under two different scenarios:

1) with variation of the prescribed ‘accuracy’ of the GSP ground survey measurements (which gives insight into how well the photogrammetry aligns with the control measurements) and,
2) with variation of the number of GCPs used as control or check points.

Requirements and compatibility

The analyses in James*, et al.* (2017a) were carried out under Windows 7, using a laptop and a PC with minimum RAM of 8 Gb. Software requirements in addition to SfM_georef v.3.1 are:

PhotoScan Pro – [http://www.agisoft.com/] Initial work was carried out using v.1.1.6; this version of SfM_georef is compatible with output from v.1.3.

Monte_Carlo_BA.py – Python code to run the Monte Carlo analysis in PhotoScan. The script (available as supplemental information for James*, et al.* (2017a)) was written for PhotoScan v.1.1.6, but updates for v.1.2.6 and v.1.3 are available from http://tinyurl.com/sfmgeoref.

Compatibility: *Note that the Python script is written for straightforward single-chunk PhotoScan .psz projects which use control measurements made in a local or projected coordinate system only.*

9.1. Generating Monte Carlo results for GCP analysis

This first stage, to generate the PhotoScan output required for analysis, is carried out using the Python script – you will need your PhotoScan project (.psz file), and a copy of the Monte Carlo python script, Monte_Carlo_BA.py.

1) Make a working copy of your PhotoScan project (so that the original one is retained safely). Open the working copy in PhotoScan and:
    - Check that in the Reference Settings, the **correct Coordinate System is set and there are appropriate values for 'Image coordinates accuracy'** (these values should be the same as the RMS image error (in pixels) on the tie points and markers – see James*, et al.* (2017a) for a detailed explanation).
    - Check that **control points (markers) or camera positions also have appropriate 'accuracy' values** (this can either be set globally in the Reference Settings dialog box, or individually for each point/photo under the Reference tab.
    - **Make sure that the bounding box fully encloses all the sparse points** – with room to spare – so that all points are always exported on each Monte Carlo iteration.
    - Open the Console panel so that you can see progress (or errors) when running the script.
    - Remove any dense cloud, mesh, DEM etc. – only the initial sparse cloud of tie points is required.
    - Save the project as a .psz archive.

    If you make any adjustments, save the working copy, to enable you to return to it later if required.

2) Next, decide whether you are going to vary number of GCPs or GCP 'accuracy', open the Monte_Carlo_BA.py Python script in a text editor and go through the Setup section making any edits required (all information is provided within the script). Save the updated copy in a convenient location near your PhotoScan project.

3) Run the Python script in PhotoScan via the menu commands:

    Tools → Run Script; navigate to the script file then click OK.

Correct execution of the script will result in repeated bundle adjustments being carried out, which you can see in the console window. Each Monte Carlo iteration will result in 5 files being written to a the directory constructed on the output path given in the Python script.

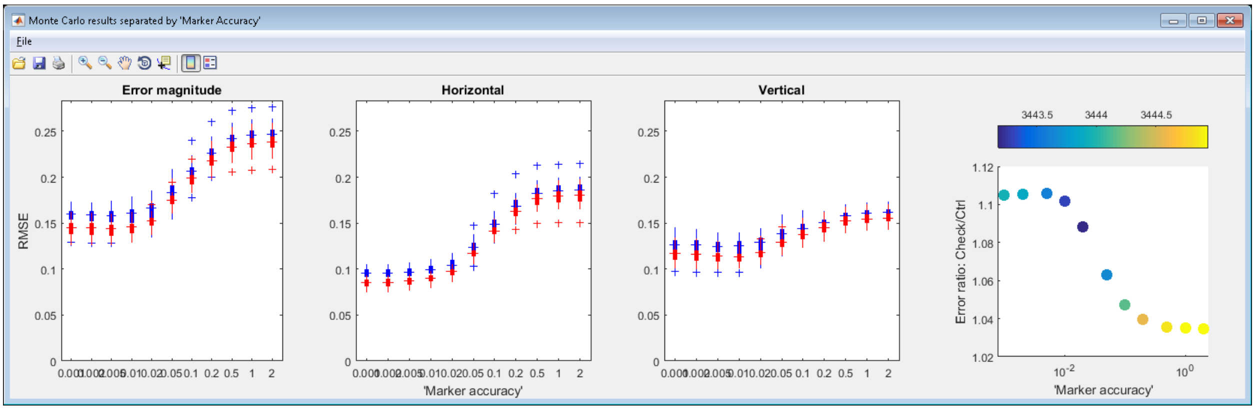
To help speed things up, minimise the PhotoScan window. When the required number of iterations has been completed, PhotoScan can be closed (without saving the project).

9.2. Analysing Monte Carlo results with varying GCP 'accuracy'

If the analyses were carried out with varying the GCP 'accuracy', the results can be analysed by:

1) Loading the output files into SfM_georef using the file menu:
   File → Miscellaneous extras → Monte Carlo GCP analysis → Read results files
2) Process the files:
   File → Miscellaneous extras → Monte Carlo GCP analysis → Plot results against Marker Accuracy

You should see something similar to the figure below – a detailed explanation is given in James*, et al.* (2017a). Error units are metres, with control point data in red, and check point data in blue. The colour scale on the right-most plot gives the mean principal distance determined for each GCP 'accuracy' value (assuming that principal distance was optimised during the bundle adjustments).

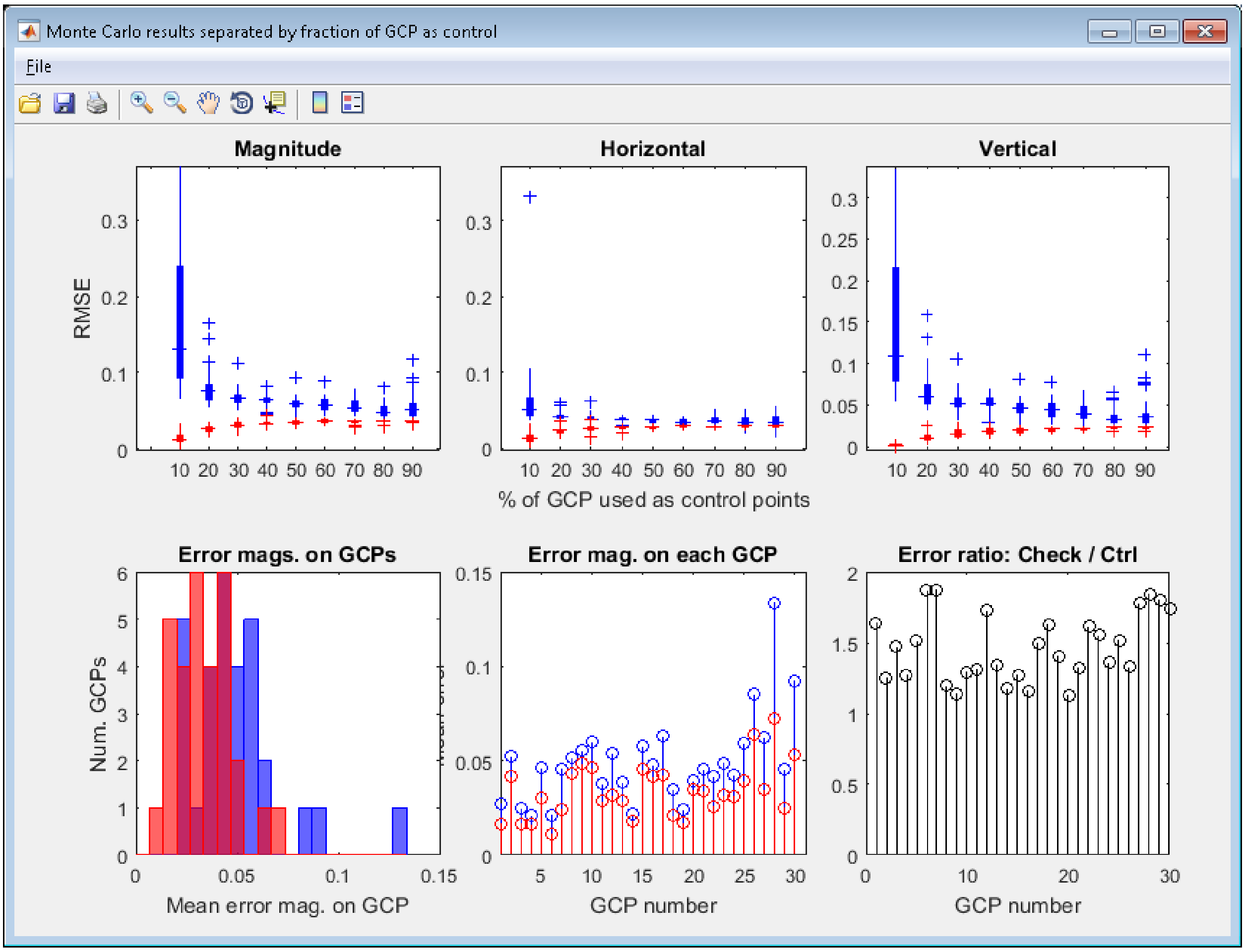


9.3. Analysing Monte Carlo results with varying the nuber of GCP used as control

If the analyses were carried out with varying the number of GCP used as control, the results can be analysed by:

3) Loading the output files into SfM_georef using the file menu:
   File → Miscellaneous extras → Monte Carlo GCP analysis → Read results files
4) Process the files:
   File → Miscellaneous extras → Monte Carlo GCP analysis → Plot results against number of GCP as control

You should see something similar to the figure below – a detailed explanation is given in James, *et al.* (2017a). Error units are metres, with control point data in red, and check point data in blue.



10. Monte Carlo precision analysis and 3-D uncertainty-based topographic change detection

These processing functions do not require an SfM project to be loaded into SfM_georef, and are purely for post-processing of the results of Monte Carlo precision analysis carried out in PhotoScan as described in James*, et al.* (2017b). The processing generates 3-D precision estimates of SfM-based topographic surveys, encompassing georeferencing and photogrammetric components, which can then be used to provide spatially varying confidence bounds for surface change detection. The instructions below are largely reproduction of the instructions provided as supplemental material for the paper, updated for version changes where appropriate. There are three main stages to the analysis:

3) Using a Monte Carlo approach in PhotoScan Pro to provide output from which precision can be estimated.
4) Deriving point precision estimates and constructing precision maps from this output (optionally, precision and correlation information on the cameras can also be derived).
5) Using the precision maps to determine 3-D change between surfaces, with confidence bounds.

Requirements and compatibility

The analyses in James*, et al.* (2017b) were carried out under Windows 7, using a laptop and a PC with minimum RAM of 8 Gb. Deriving the precision estimates is memory intensive, and larger projects could require more RAM. Software requirements in addition to SfM_georef v.3.1 are:

PhotoScan Pro – [http://www.agisoft.com/] Initial work was carried out using v.1.2.3; this version of SfM_georef is compatible with output from v.1.3.

CloudCompare – [http://www.danielgm.net/cc/] tested with version 2.6.0 [Windows 64 bits]

precision_estimates.py – Python code to run the Monte Carlo analysis in PhotoScan. The script (available as supplemental information for James*, et al.* (2017b)) was written for PhotoScan v.1.2.3, but should also run on v.1.2.6; if you are using a more recent PhotoScan version, check http://tinyurl.com/sfmgeoref for any updates.

Compatibility: *Note that the Python script is written for straightforward single-chunk PhotoScan .psz projects which use control measurements made in a local or projected coordinate system only. For precision analysis (i.e. prior to dense matching), projects must be saved in .psz format.*

10.1. Generating Monte Carlo results from which precision can be estimated

This first stage, to generate PhotoScan output from which precision can be estimated, is carried out using the Python script – you will need your PhotoScan project (.psz file), and a copy of the Monte Carlo python script, precision_estimates.py.

4) Make a working copy of your PhotoScan project (so that the original one is retained safely). Open the working copy in PhotoScan and:
    - Check that in the Reference Settings, the **correct Coordinate System is set and there are appropriate values for 'Image coordinates accuracy'** (these values should be the same as the RMS image error (in pixels) on the tie points and markers – see James*, et al.* (2017a) for a detailed explanation).
    - Check that **control points (markers) or camera positions also have appropriate 'accuracy' values** (this can either be set globally in the Reference Settings dialog box, or individually for each point/photo under the Reference tab.
    - **Ensure that appropriate control data are selected as active** (e.g. markers or cameras are checked).
    - **Make sure that the bounding box fully encloses all the sparse points** – with room to spare – so that all points are always exported on each Monte Carlo iteration.
    - Open the Console panel so that you can see progress (or errors) when running the script.
    - Remove any dense cloud, mesh, DEM etc. – only the initial sparse cloud of tie points is required.
    - Save the project as a .psz archive.

    If you make any adjustments, save the working copy, to enable you to return to it later if required.

5) Next, open the precision_estimates.py Python script in a text editor and go through the Setup section making any edits required (all information is provided within the script). Save the updated copy in a convenient location near your PhotoScan project.

6) Run the Python script in PhotoScan via the menu commands:

Tools → Run Script; navigate to the script file then click OK.

Correct execution of the script will result in a copy of the chunk being made, and then repeated bundle adjustments being carried out, which you can see in the console window. Each Monte Carlo iteration will result in 5 files being written to a new 'Monte_Carlo_output' directory constructed on the output path given in the Python script.

To help speed things up, minimise the PhotoScan window. For processing the survey presented in James*, et al.* (2017b)*, 4000 Monte Carlo iterations took a few hours on a modern desktop PC, but this will vary from survey to survey.*

When the required number of iterations has been completed, PhotoScan can be closed (without saving the project).

10.2. Deriving precision estimates and constructing precision maps

The second stage compiles the all the files output by the PhotoScan Monte Carlo iterations into point precision estimates, which can then be interpolated into maps for further use. The code to carry out this processing is integrated into sfm_georef v.3.0. Because the Monte Carlo files are imported simultaneously (computer-memory intensive), a good machine may be needed for large projects. Nevertheless, James*, et al.* (2017b) derived precision estimates from 4,000 iterations of ~40,000 tie points within a few minutes on an Windows 8 laptop with 8 Gb RAM.

5) Start sfm_georef; to facilitate monitoring progress, move and resize the Messages window to full screen height.
6) To derive point precision estimates, use the file menu:

   File → Miscellaneous extras → Monte Carlo precision analysis → Determine point precisions

   Navigate to, and then select, the folder containing the output from the PhotoScan Monte Carlo analysis (it should contain a sub-folder 'Monte_Carlo_output').

7) In the options dialog box that appears, the offsets used as a local origin in the PhotoScan output files should have the correct values (read from the file _local_coordinate_origin.txt), but adjust if necessary. For all other options, the default values are recommended. If you wish to see how estimates of point coordinate variance decrease with the number of Monte Carlo iterations (i.e. Figure 8 of James*, et al.*, 2017b) then check the 'Detail changes with number of iterations (slow)' checkbox.

   The files will be processed to give text and graphical output:

```
4000 .ply files found. First file has 41989 points.
Reading point cloud (.ply) files... Done.
Calculating mean point coordinates... Done.
Calculating point precisions and covariances... Done.
Writing text file output: _point_precision_and_covars.txt... Done.
Calculating overall georeferencing precision... Done.

Differences between Monte Carlo means and initial error-free values:
(Differences should be small to indicate that sufficient Monte Carlo iterations have been completed).
Rotation (as Euler angles; o, p, k; mdeg):     0.02026 0.01306 0.00100
Translation (in X, Y, Z; mm):                  -0.03560-0.003740.04436
```

```
Scale (%):          0.0003

Overall survey georeferencing precision:
Rotation (as Euler angles; o, p, k; mdeg):      18.419   9.815    4.017
Rotation (around fixed X, Y, Z axes; mdeg):  7.471    17.389   0.313
Translation (in X, Y, Z; mm):                   2.56     2.40     5.57
Scale (%):          0.0072

RMS residual discrepancies on point coordinates after overall georeferencing:
Mean for all points (in X, Y, Z; mm):                    56.50    34.46    38.50
Std. dev. across all points (in X, Y, Z; mm):  0.73     0.43     0.34

Calculating shape point precisions and covariances (i.e. excluding overall georeferencing errors)... Done.
Writing text file output: _point_precision_and_covars_shape_only.txt... Done.

Mean point precisions (mm):
Full (X, Y, Z, [Mean], Hz, full magn.:          37.06    24.49    40.85    [34.13]   46.23    63.82
Shape only (X, Y, Z, [Mean], Hz, full magn.:  36.19    23.17    33.19    [30.85]   44.88    58.14

Median point precisions (mm):
Full (X, Y, Z, [Median], Hz, full magn.:        20.69    15.59    35.10    [24.22]   28.31    47.12
Shape only (X, Y, Z, [Median], Hz, full magn.:           19.75    13.90    26.50    [20.78]   26.76    40.46

Generalised survey imaging characteristics:
Max. survey dimension:             1011.71 m
Mean observation distance:         140.00 m
Mean ground pixel dimension:       37.0 mm

Dimensionless relative precision ratios (i.e. mean point precision with respect to):
                            Full             Excluding overall georeferencing
Survey extent:              1 : 29,600       1 : 32,800
Observation distance:       1 : 4,100        1 : 4,540
Pixel size (XY precision):  1.3              1.2   (pixels)
Pixel size (Z precision):   1.1              0.9   (pixels)
Point precision analysis complete.
```

As well as the summary information above, the main precision and variance-covariance outputs of the processing are given in the text files:

_point_precision_and_covars.txt
_point_precision_and_covars_shape_only.txt

which are tab-delimited text files with 13 columns, and one row for each point (plus a header line). For each row, the columns are:

Column header	Data
X(m), Y(m) , Z(m)	mean coordinates (in m) for that point from all the Monte Carlo iterations
sX(mm) , sY(mm) , sZ(mm)	precision estimates (in mm) for the point coordinates
covXX(m2) , covXY(m2) , covXZ(m2) , covYY(m2) , covYZ(m2) , covZZ(m2)	variance-covariance estimates (in m^2) for the point coordinates

8) To construct the precision maps, use the file menu:

File → Miscellaneous extras → Monte Carlo precision analysis → Interpolate precision maps

Navigate to, and then select, the output file from the previous stage (e.g. _point_precision_and_covars.txt) containing the point precision estimates. A summary of the point data will be displayed in the Messages window. When asked for a filename for the interpolated grid, enter a new one or use the default. Formats are:

.tif – output will be as one GeoTiff image, with precision in X Y and Z in bands 1, 2 and 3 of the image file

.asc – output will be as three ArcASCII files, one for each X, Y, and Z component of precision. Filenames will be distinguished by having '_x' etc. appended.

9) The settings required for the interpolation must now be entered into the Point precision interpolation dialog box. The Grid bounds will determine the extent of the grid (note that these are cell edges, not centres), and grid interval controls the resolution (equal in *X* and *Y*). The maximum bounds will be adjusted automatically during processing if they are not integer multiples of the grid interval. The interpolation is carried out by determining the median precision for all values within a defined search radius of each grid cell centre.

```
Reading point precision information from: _point_precision_and_covars.txt... Done.
Read data for 41,989 tie points, with coordinate bounds:
X:      265663.30       266674.10       Range = 1010.80
Y:      4701519.81      4701954.36      Range = 434.54
Z:      570.08          658.96          Range = 88.89

Coordinate system: ED50 / UTM zone 31N (EPSG::23031)

Interpolating precision values over 110 x 70 grid... Done.

Writing map file:  _precision_map_x.asc...
Writing map file:  _precision_map_y.asc...
Writing map file:  _precision_map_z.asc...
Construction of point precision maps complete.
```

10.3. Camera orientation precision, parameter precisions and correlations

Although not required as part of the point precision workflow, the Monte Carlo output also enables SfM_georef to estimate camera parameter precision and correlations, and camera orientation precision. These can give useful insight into self-calibrating network performance, and can be derived through the following menu command:

File → Miscellaneous extras → Monte Carlo precision analysis → Determine camera precisions and correlations

Example summary output is:

```
4000 camera files found, each with 104 photos, 1 calibration(s)
Reading camera calibration files... Done.

Camera calibration 1
P.D.:   3786.423 +/- 0.158
CCx:    2295.452 +/- 0.075
CCy:    1570.164 +/- 0.129
```

```
K1:     -9.2484e-02 +/- 8.5199e-05
K2:     3.5033e-02 +/- 3.5716e-04
K3:     3.1925e-03 +/- 4.5061e-04
P1:     0.0000e+00 +/- 0.0000e+00
P2:     0.0000e+00 +/- 0.0000e+00

Camera 1 parameter correlation coefficients:
        CCx     CCy     P.D.    K1      K2      K3      P1      P2
CCx     1.00
CCy     -0.05   1.00
P.D.    -0.09   -0.62   1.00
K1      -0.03   -0.09   -0.03   1.00
K2      0.03    0.08    0.10    -0.96   1.00
K3      -0.03   -0.09   -0.07   0.91    -0.98   1.00
P1      ---     ---     ---     ---     ---     ---     ---
P2      ---     ---     ---     ---     ---     ---     ---     ---

Reading control files... Done.

Mean precision of camera orientations:
Position (X, Y, Z; mm):                 16.39   26.24   30.48
Pointing (Yaw, Pitch, Roll; mdeg):      11.67   17.74   16.23
Writing mean camera orientations and orientation precisions to file (_camera_orientations_and_precisions.txt)...
Done.

Calculating mean precision of camera orientations in OPK:
Pointing (Omega, Phi, Kappa; mdeg):             21.07   9.03    9.09
Camera precision analysis complete.
```

Further details are given in the text file: _camera_orientations_and_precisions.txt

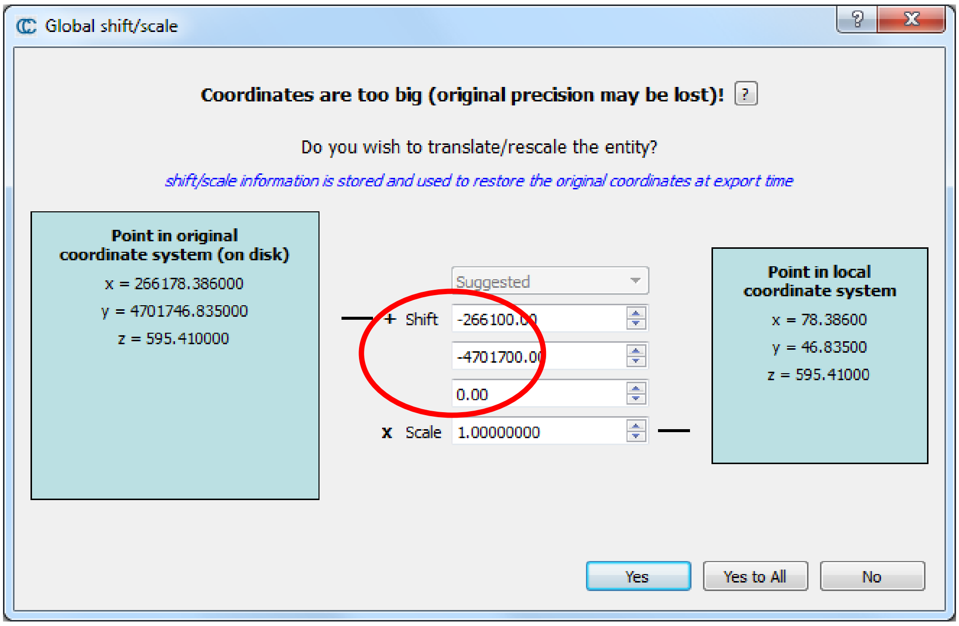
which is tab-delimited, with 13 columns and one row for each photograph (plus a header line). For each row, the columns are:

Column header	Data
Image_name	image file name
X(m), Y(m) , Z(m)	mean coordinates for that camera from all the Monte Carlo iterations
sX(mm) , sY(mm) , sZ(mm)	precision estimates (in mm) for the camera position
Yaw(deg), Pitch(deg), Roll(deg)	mean rotation angles (in degrees) for that camera from all the Monte Carlo iterations
sYaw(mdeg), sPitch(mdeg), sRoll(mdeg)	precision estimates (in milli-degrees) for the camera rotation angles

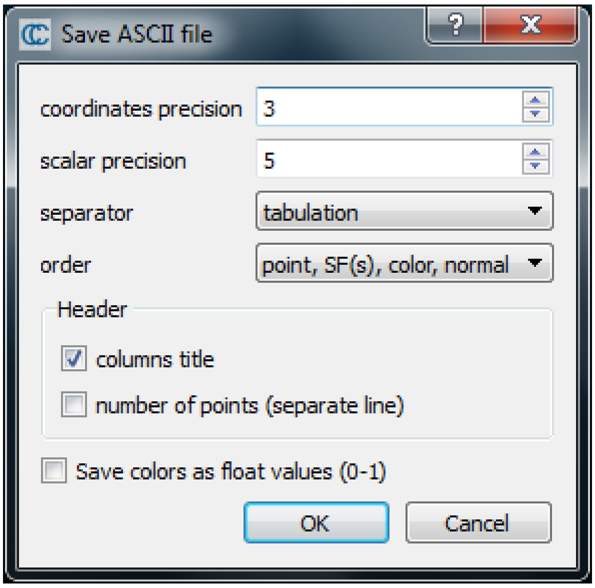
10.4. Using precision maps to derive 3-D change between surfaces, with confidence bounds

Precision maps can be combined to determine confidence bounds when estimating change between two SfM surveys, through using the an adaptation of the M3C2 algorithm (Lague*, et al.*, 2013), i.e. M3C2-PM (James*, et al.*, 2017b). The processing is based on the dense point clouds (not DEMs) that can be exported from PhotoScan.

1) Derive precision maps for two SfM surveys, as described above, and export dense point clouds from both projects. Note that if you are looking to compare SfM data with another survey without precision maps, then the analysis can be carried out using fixed precision values where required.
2) Load both point clouds into **CloudCompare**: make a record of the values of any offset applied by CloudCompare during the import, and ensure that equal offsets are applied to both point clouds.



3) Select the two clouds in CloudCompare's DB Tree window and run the M3C2 plugin from the File menu: Plugins → M3C2.
4) For details on M3C2, see Lague, *et al.* (2013), but the use here is just to derive normal directions and associated distances. Set the Main parameters and click OK.
5) The M3C2 plugin will have output a new pointcloud "M3C2 output scale =xxxx" into CloudCompare's DB Tree. Select this entry in the DB tree window, then export the data to text file in the 'ASCII cloud' format (via the menu File → Save). Typical export parameter are :



6) Returning to **SfM_georef**, the M3C2 output can then be recalculated using the precision maps (M3C2-PM) through the file menu:

File → Miscellaneous extras → Monte Carlo precision analysis → M3C2-PM change detection

7) Through dialog boxes, select the M3C2 output file and the precision maps for both surveys. Currently, the M3C2 plugin in CloudCompare provides only local (shifted) coordinates, so to include the offset used in Step 2 and obtain M3C2-PM results in the original coordinate system, enter the offset values when asked (**note** – negative values in the CloudCompare offset should be added as a positive offset in M3C2-PM). If precision maps are not available for one survey (e.g. a TLS survey), then cancel the dialog box and enter values manually (in mm) to be ascribed to each point. M3C2-PM will process the data and output a text results file (similar to M3C2 output), which contains the confidence bounds and significant changes calculated using the precision maps.
8) These results can then be analysed in a variety of software, including CloudCompare.

Citations, feedback and bug reports

Feedback and bug reports are welcome; please send to: m.james@lancaster.ac.uk
I will endeavour to do my best, but I cannot guarantee answering all emails. Apologies in advance.

If you use sfm_georef, please cite the following work:

James, M. R. and Robson, S. (2012) Automated image-based 3-D surface reconstruction for the geosciences: Accuracy and application, *J. Geophys. Res.*, 117, F03017, doi:10.1029/2011JF002289.

If you use the Monte Carlo analysis of ground control, please also cite:

James, M. R., Robson, S., d'Oleire-Oltmanns, S. and Niethammer, U. (2017a) Optimising UAV topographic surveys processed with structure-from-motion photogrammetry: Ground control quality, quantity and bundle adjustment, *Geomorphology*, 280, 51–66, doi: 10.1016/j.geomorph.2016.11.021

If you use the Monte Carlo precision analysis, please also cite:

James, M. R., Robson, S., Smith, M. W. (2017b) 3-D uncertainty-based topographic change detection with structure-from-motion: precision maps for ground control and directly georeferenced surveys, *Earth Surf. Proc. Landforms*, accepted, February, doi: 10.1002/esp.4125

Mike James, Lancaster University, 19 March 2017