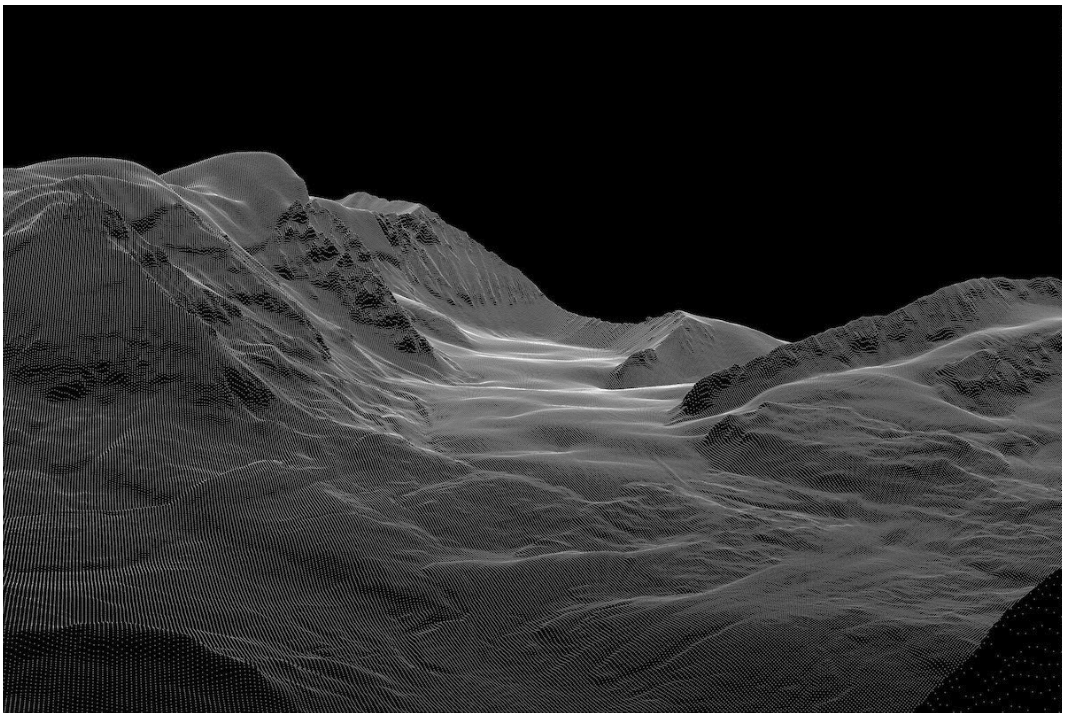
Georeferencing oblique terrestrial photography

photogeoref V. 1.0 User's Manual



photogeoref home page: https://www.meteoexploration.com/python/photogeoref/index.html

1 Introduction

Conventional photography is a powerful media for collecting and storing information. If this information can be geo-located precisely, then photography becomes a powerful tool for quantitative analysis. Such a use is well developed and well documented in photogrammetry, though not so much in conventional photography. Here we present a python tool for georeferencing oblique photography using a single image and a digital elevation model (DEM). The accuracy of the technique will depend on the accuracy of the DEM and on the quality of the photographic image, especially the degree of distortion and aberration produced by the lens. This technique does not produce elevation data. It actually requires an existing DEM. What the tool does is locating the geographical position of every pixel in a photographic image. It is therefore useful to map land cover and to assess surface cover change. It has applications in snow hydrology, glaciology, forestry and other disciplines.

The results are better for a near perpendicular view. A slant view could distort and project objects further away of their actual position. It is a suitable tool for mountain terrain or small parcels from an elevated viewpoint. Georeferencing oblique photography can be an efficient complement to remote sensing, or a good alternative when high resolution satellite images are too expensive. This technique works on cloudy and overcast days and can be applied to multispectral cameras with some modifications. Some examples are provided in the project web page.

2 Overview

Runing the script will display a Graphical User Interface (GUI) where settings can be adjusted, the results are visualized on the display window. The GUI reads the dimensions of the screen and fit the display accordingly.

To run the script, type in a terminal:

```
python3 photogeoref.py -h
```

Shows the command syntax and available options.

Available options are:

```
options:
  -h, --help            show this help message and exit
  -s SETTINGS, --settings SETTINGS
                        Full path to yaml settings file
```

Calling the script without arguments runs it with the default settings, which are stored in a yaml file. This file needs to be edited before running any examples. Simply change 'full_path_to' to the actual path where your files are stored in your local computer. The default yaml file is 'georefsettings.yml' and has the following content:

```
demfname: '/full_path_to/dem.tif'
visfname: '/full_path_to/demviewshed.tif'
imgfname: '/full_path_to/RGB_image.tif'
GCPfname: '/full_path_to/x_y_z_description.csv'
obscoords:
- 728360.0
- 4762763.0
- 1465.0
tgtcoords:
- 728268.0
- 4763407.0
- 1497.0
fwidth: 0.0359
fheight: 0.024
focallength: 0.028
roll: 2.5
```

Path to filenames can only be edited in the yaml file, while the other field can be edited interactively.

```
demfname
```

Full path to a GeoTIFF with elevation data (DEM). ΔX, ΔY and Z should be in metres, DEM should be a UTM projection with squared pixels.

```
visfname
```

Full path to a GeoTIFF with the calculated DEM viewshed from the observer position. Visible values should be 1 and non-visible values zero. The script maps only visible cells in front of the camera and ignores grid cells that are behind the observer. A simple way to calculate viewshed is with gdal:

```
gdal_viewshed -md 30000 -ox observerX -oy ObserverY
              -oz heightoverdem -vv 1 inputdem.tif outputviewshed.tif
```

```
imgfname
```

Full path to a RGB tiff with the photograph to be georeferenced. It should be the full image, not cropped. Images can be processed, contain annotations and modifications but dimensions should be preserved

```
GCPfname
```

Full path to a a comma-separated-values file (csv) with Ground Control Points. Its is advisable to include the observer and the target position.

```
tgtcoords
```

Array with the observer X, Y, Z coordinates (camera position). Observer coordinates should be inside the DEM domain and in the same reference system. Coordinates should be easting and northing in metres, not degrees of latitude or longitude.

```
obscoords
```

Array with the target X, Y, Z coordinates. The target is the real world position of the mid point of the photograph. Target coordinates should be inside the DEM domain and in the same reference system. Coordinates should be easting and northing in metres, not degrees of latitude or longitude.

```
fwidth
```

Camera sensor witdh in metres. This example is for a Nikon D800 with full width sensor at ≈36 x 24 mm, exactly 0.0359 m. A good source of camera specifications is https://www.dpreview.com/

```
fheight
```

Camera sensor height in metres. This example is for a Nikon D800 with full width sensor at 36 x 24 mm, exactly 0.024 m.

```
focallength
```

Camera lens focal lentgh in metres. Compact cameras and zoon lenses can have a very different nominal and actual value, they can differ up to a 10%.

```
roll
```

Camera roll in degrees. If the camera is hand held, it is very difficult to keep it perfectly horizontal. The roll helps correcting lateral inclination of the camera. A negative roll will rotate the image to the right.

3 Usage

Simply call the script from a terminal with all the required python modules installed:

```
python 3x
argparse
copy
cv2
json
matplotlib
numpy
os
osgeo
sys
time
tkinter
urllib.request
yaml
```

The option `–s` or `––settings` will read a specified settings file, if this option is not provided, it will read the default file 'georefsettings.yml'

Calling the script will open a GUI with the default or prescribed settings file at the top entry field. Check that the file is correct and exists and click the button "Load settings". All fields should be populated and the photography displayed on the GUI canvas (Figure 1).

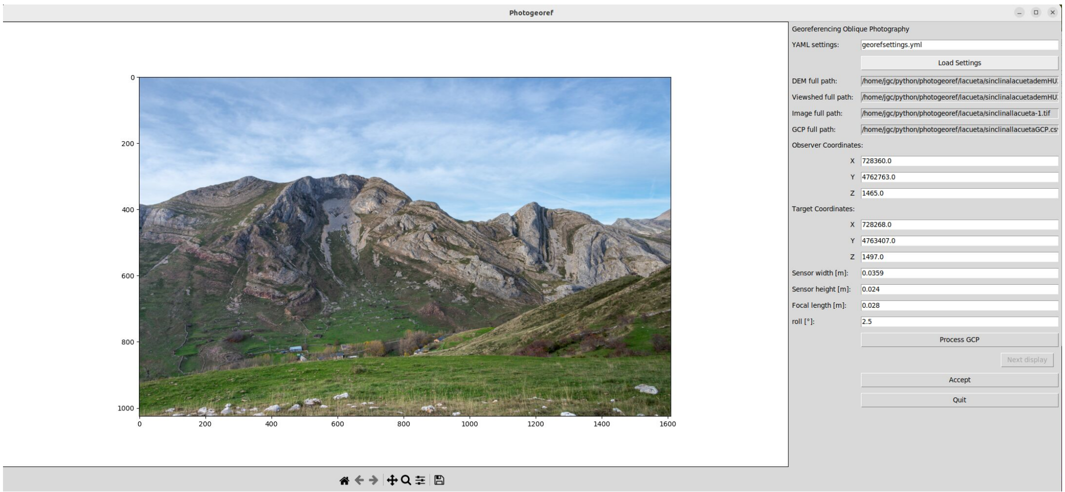


Figure 1: Initial GUI display after loading settings.

Click the button "Process GCP", a grey scale DEM should be displayed, with the camera position in blue, the target position in green and the GCP as black dots (Figure 2).

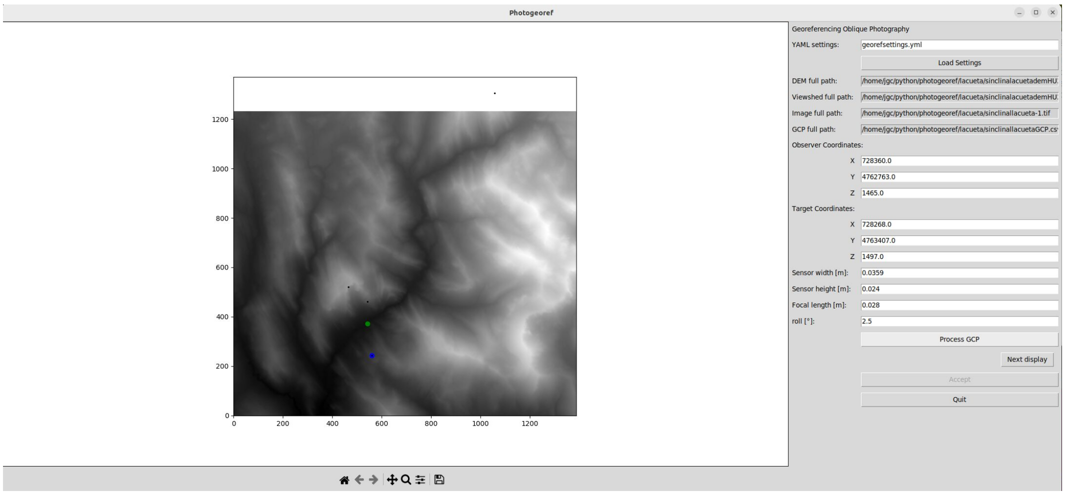


Figure 2: GUI display after processing GCP.

Click the button "Next display" to show the DEM as a perspective projection from the observer point of view. Click again to show the DEM grid points superimposed on the image. Fields with a white background can be edited to improve the match between the projected DEM and the photograph (Figure 3). Click the 'Process GCP' button again to apply the changes. After the second time 'Process GCP' is pressed, only the DEM grid points superimposed on the image will be shown. Corrections can be applied as many times as necessary. If the match between DEM and image is correct click on the 'Accept' button, a georeferenced image and contour lines will be displayed (Figure 4).

When clicking 'Accept' a few additional files will be created:

- '[imagename]DEMGCP.jpg' DEM plot with target, observer and GCP
- 'demperspective.png' a perspective projection of the DEM
- '[imagename]coimg.jpg' perspective projection of DEM superimposed on photograph
- '[imagename]refplot.png' georeferenced image with contour lines plot
- '[imagename]ref.tif' RGB GeoTIFF of georeferenced photograph with the same projection and extent as the original DEM

'[imagename]' is the photograph filename without path or suffix.

The script creates an additional log file 'georefGCP.log' in the same folder as the photograph. The log registers different changes and the last settings used. Logs are appended, it is up to the user deleting the log file when it is too large or no longer needed.

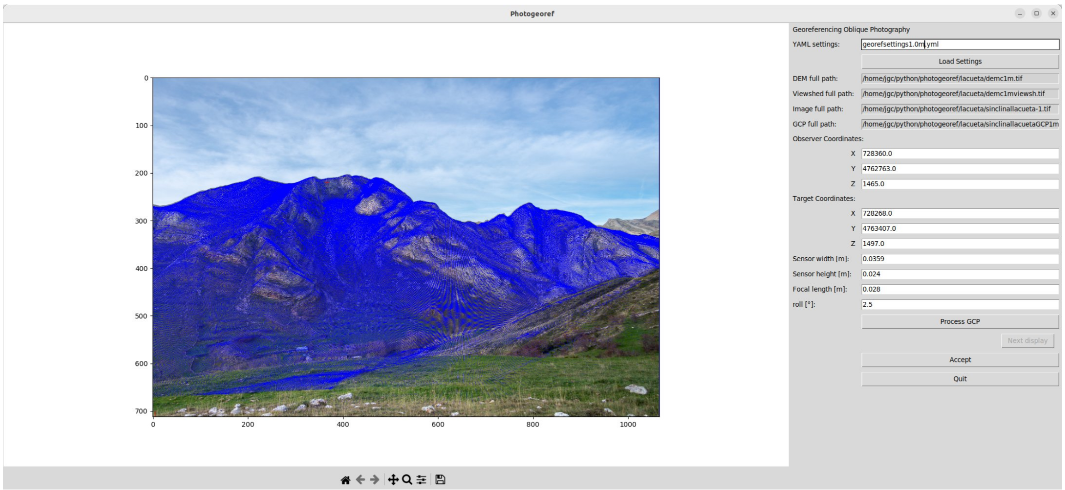


Figure 3: Photograp with perspective projection of DEM superimposed as blue dots

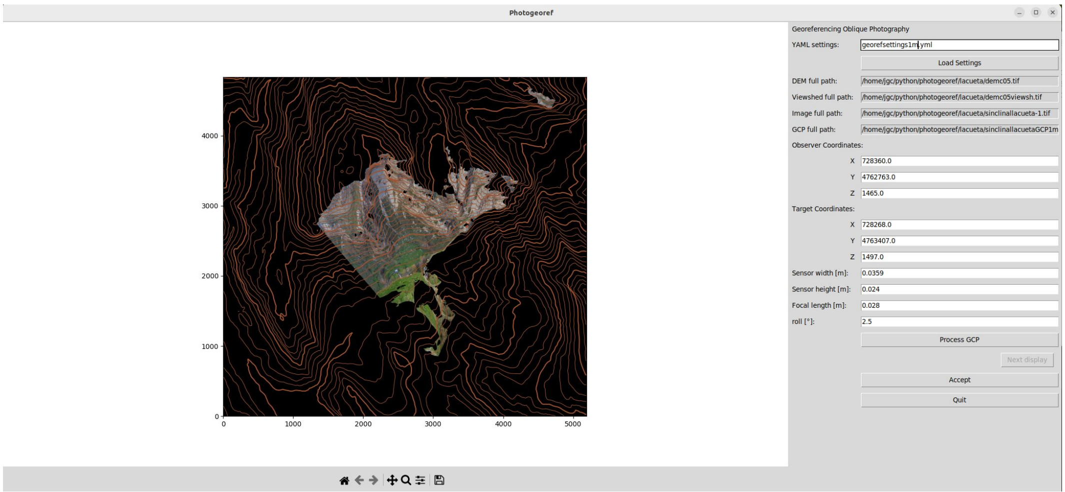


Figure 4: Final georeferenced image with DEM contours superimposed.

At exit the script creates a new settings file with the last values used, the file name will be the last settings file provided plus a timestamp.

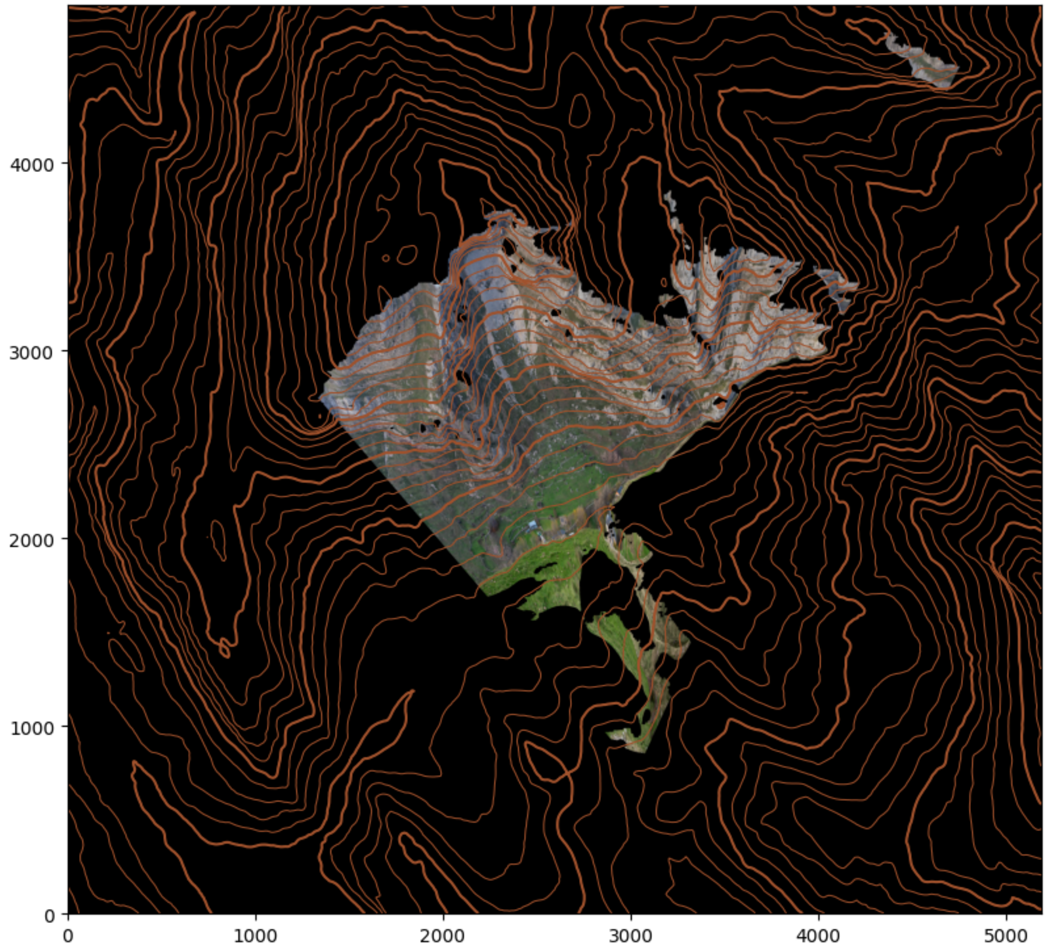


Figure 5: The final product: a georeferenced map of reflectance values.

4 Geometrical transformations

In order to georeference terrestrial photographs we need to find a function relating two-dimensional pixels in the photograph to three-dimensional points in the digital elevation model. A DEM is a discrete mathematical representation of the terrain. We can, *a priori*, replicate the geometry of the image formation inside the camera on the DEM by applying the necessary viewing transformations and perspective projections. In this way we produce a "virtual" photograph of the digital elevation model, that is, a two-dimensional representation of the relief information contained in the DEM, as seen from the point of view of the camera. By scaling this representation according to the resolution of the photograph, we can establish the necessary correspondence between pixels in the image, screen coordinates of the perspective projection of the DEM and their geographic location.

The first viewing transformation redefines the original coordinates of the DEM, taking as origin of coordinates the position of the camera. This is equivalent to a simple translation. The second viewing transformation rotates the original orthogonal Axis of the DEM, aligned Eastwards, Northwards, and Upwards to make them coincide with the orientation of the camera as shown in Figure 6.

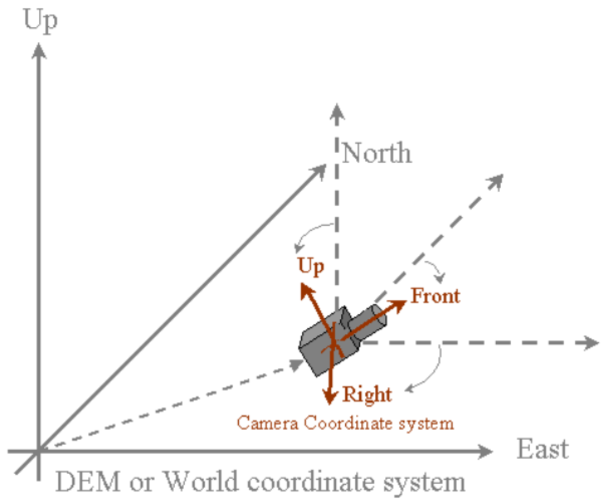


Figure 6: Schematic representation of the world (or DEM) and the camera reference systems.

A detailed description of the geometric transformation is given by Corripio (2004) and a summary of the equations are below. Additional useful information on rotational matrices and geometrical transformations can be found in Fiume (1989), Foley, van Dam, Feiner and Hughes (1990), Watt and Watt (1992) or in Goldstein (1980).

The georeferencing process consists of a viewing transformation applied to the DEM in which the coordinates of every grid cell are firstly translated to refer them to a coordinate system with origin at the camera position. Then a transformation is applied according to the viewing direction and focal length of the camera (Equation 1). This results in a three dimensional set of points corresponding to the cells in the DEM as seen from the point of view of the camera (Figure 8). Finally, the resulting viewing transformation is projected into a two dimensional viewing "window", corresponding to the area of the film and scaled proportionally as indicated in Figure 7 and Equation 6. The process allows a direct comparison of the image and the perspective projection of the DEM. If the result is satisfactory, the pixels in the image are allocated the x,y,z values of the corresponding grid cells in the DEM. The result is a georeferenced map of reflectance values shown in Figure 5.

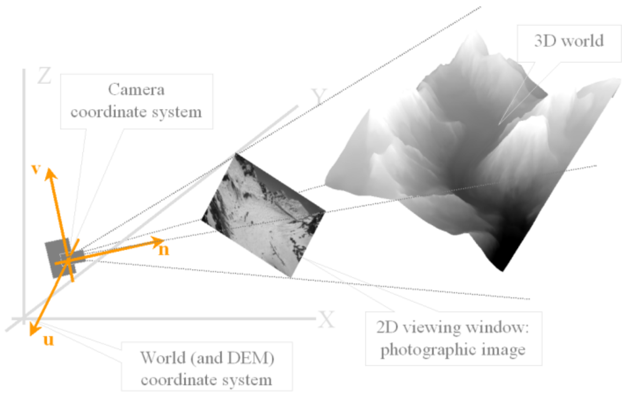


Figure 7: Georeferencing process. The coordinates of the original 3D landscape are transformed to the camera referenced system and then projected onto a viewing window corresponding to the flat 2D camera sensor.

The viewing transformation, which rotates the translated coordinates according to the viewing reference system is:

$$\begin{pmatrix} P_{cx} \\ P_{cy} \\ P_{cz} \\ w \end{pmatrix} = \begin{pmatrix} U_x & U_y & U_z & 0 \\ V_x & V_y & V_z & 0 \\ N_x & N_y & N_z & 0 \\ 0 & 0 & 1/f & 1 \end{pmatrix} \begin{pmatrix} P_{tx} \\ P_{ty} \\ P_{tz} \\ 1 \end{pmatrix}, \quad (1)$$

where f is the focal length of the lens, P_t is the translated point to a reference system with origin at C, the camera position (simply by adding the inverse of the camera coordinates

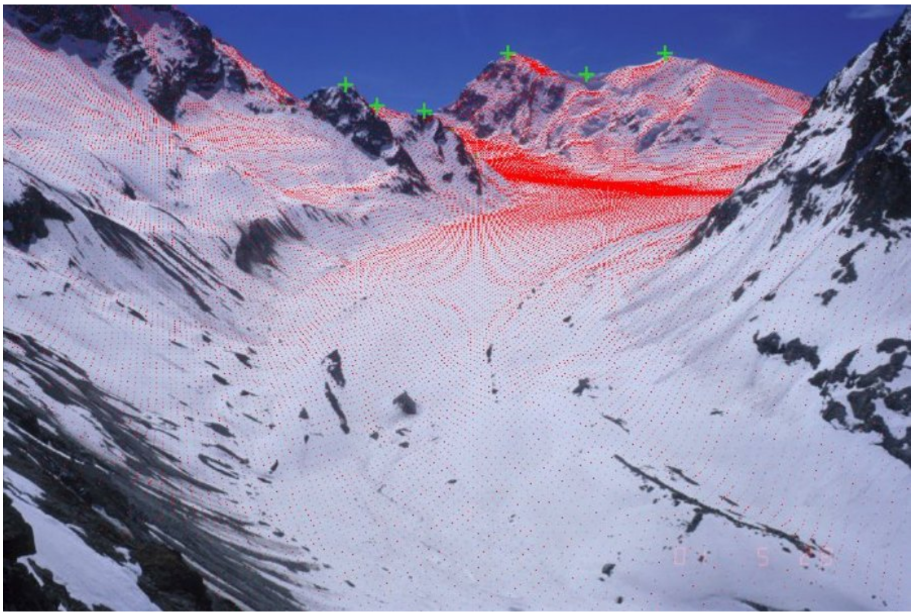


Figure 8: Grid cells from a DEM represented as red dots are transformed into a perspective projection and superimposed on the photograph of the glaciers. Green crosses are ground control points taken with a differential GPS and used for fast photo orientation

to the original world coordinates), N, U, V are vectors defining the viewing geometry and P_c is the resulting coordinates of a point in camera coordinate system.

The calculation of the unit vectors defining the viewing geometry is as follows. The viewing direction or vector $\vec{N}$ is :

$$\vec{N}_0 = T - C, \tag{2}$$

$$\vec{N} = \frac{\vec{N}_0}{|\vec{N}_0|}, \tag{3}$$

where T is the coordinates of the the target (aim of the camera) and C the coordinates of the camera with respect to the DEM origin of coordinates.

From this vector, and applying a procedure slightly modified from that used by Fiume (1989) or Watt and Policarpo (1998), $\vec{U}$ and $\vec{V}$ are calculated using simple vector calculus, by finding the cross products:

$$\vec{U} = \begin{cases} \vec{N} \times \frac{\vec{N}_{xy}}{|\vec{N}_{xy}|} & \text{if} N_z > 0 \\ \frac{\vec{N}_{xy}}{|\vec{N}_{xy}|} \times \vec{N} & \text{if} N_z < 0 \end{cases} \tag{4}$$

$$\vec{V} = \vec{N} \times \vec{U}, \tag{5}$$

where $\vec{N}_{xy}$ is the projection of $\vec{N}$ onto the horizontal plane, or $(N_x, N_y, 0)$, and N_z the z coordinate of vector $\vec{N}$.

The resulting x, y coordinates of the final perspective projection, following Watt and Watt (1992), are calculated as:

$$P_{px} = \frac{fP_{cx}}{\frac{1}{2}wP_{cz}} \quad \text{and} \quad P_{py} = \frac{fP_{cy}}{\frac{1}{2}wP_{cz}}, \tag{6}$$

where $P_{p(x,y)}$ are the new x, y coordinates of the perspective projection of the point $P_{c(x,y,z)}$ onto the projection plane, which in this case is the film. The factor $1/2$ is introduced in the denominator to set the origin of coordinates in the projection plane to the centre of the film.

An additional viewshed calculation is applied to eliminate all points in the DEM behind the observer. This will limit the maximum field of view to $180°$. Once the viewing vector is calculated, we can divide the DEM into two hemispheres, one in the direction of view and another in the opposite direction. We will use only that one in the viewing direction, to avoid mapping the pixels in the photograph to DEM grid cells that are behind the observer. The dot product of two vectors is the projection of the first onto the second, if this value is negative it means that the two vectors form an obtuse angle. If we take the dot product between all vectors from the observer coordinates to every grid cell of the DEM and the viewing vector, all negative values are behind the cameara position in the direction of view, and therefore can be mapped as non visible.

Finally the drop due to the curvature of the earth (Δh) is applied to all cells in the DEM according to the distance (d) to the observer as $\Delta h = d^2/2R$, where R is the radius of the Earth (6371 km).

References

Corripio, J. G.: 2004, Snow surface albedo estimation using terrestrial photography, *International Journal of Remote Sensing* **25**(24), 5705–5729.

Fiume, E. L.: 1989, *The Mathematical Structure of Raster Graphics*, Boston Academic Press.

Foley, J. D., van Dam, A., Feiner, S. K. and Hughes, J. F.: 1990, *Computer Graphics, Principles and Practice*, Adison-Wesley.

Goldstein, H.: 1980, *Classical Mechanics*, Addison-Wesley.

Watt, A. H. and Policarpo, F.: 1998, *The Computer Image*, Addison-Wesley.

Watt, A. H. and Watt, M.: 1992, *Advanced Animation and Rendering Techniques: Theory and Practice*, ACM Press: Addison-Wesley.