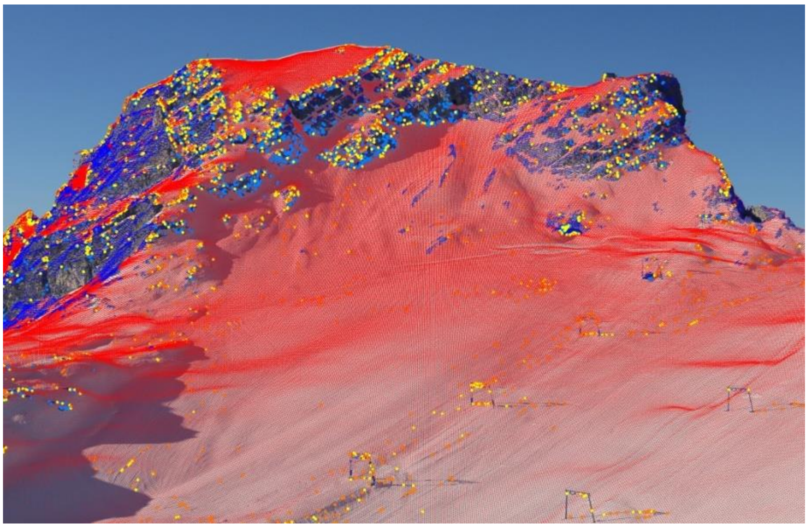
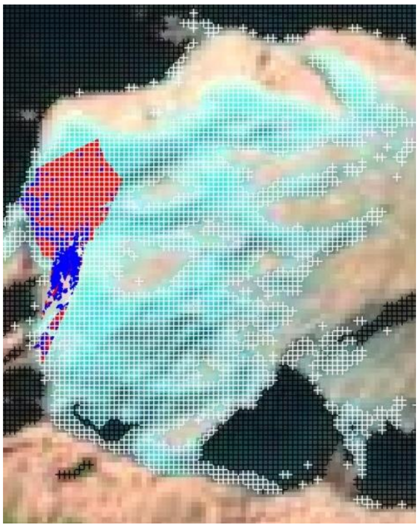
PRACTISE
–
Photo Rectification And ClassificaTIon SoftwarE





– PRACTISE V.2.0 User's Manual –

Stefan Härer,
Matthias Bernhardt,
&
Karsten Schulz

Stefan Härer
Institute of Water Management, Hydrology and Hydraulic Engineering (IWHW)
University of Natural Resources and Life Sciences (BOKU), Vienna
Muthgasse 18
1190 Vienna
Austria
Email: stefan.haerer@boku.ac.at

Contents

1. Introduction

Digital cameras project the real world (3D) into an image plane (2D) storing the spectral information of the landscape pixelwise in three visible bands: red (R), green (G) and blue (B). The high quality of commercial camera systems today further allows re-projecting the 2D photograph to the 3D landscape again. Camera images thus have become a valuable source of information in the quantitative description of spatially distributed land surface parameters. This development is also enhanced as the camera systems are easy-to-use and cost-effective, and as the cameras can deliver data from areas where in-situ measurements are often prevented for reasons of remoteness and safety, e.g. mountainous terrain.

The Matlab routine PRACTISE (Photo Rectification And ClassificaTIon SoftwarE V.1.0; Härer et al., 2013) was developed to map alpine snow cover extents from oblique terrestrial RGB photography. The new version of PRACTISE (V.2.0) now additionally presents a new method to monitor alpine snow cover patterns in satellite data. Within one program evaluation, a photograph snow cover map is derived and then directly utilised as in-situ information to calibrate the snow threshold in the Normalised-Difference Snow Index (NDSI; Dozier, 1989) map of a simultaneously captured satellite image. The automatic calibration of the NDSI threshold in the overlap area of photograph and satellite image allows an objective, reproducible and user-independent generation of an optimal satellite snow cover map. Additional new features of PRACTISE V.2.0 are a new principal component analysis (PCA) based snow classification routine introduced to improve the snow classification accuracy in shadow-affected photographs, and enhanced user friendliness by allowing direct user interaction within single modules during runtime. Moreover, the software offers all modules of version 1.0 with a revised and more performant code and is still freely available.

PRACTISE V.2.0 includes the following modules:

- (Global optimisation of the camera location and orientation)
- (Accuracy assessment of the camera location and orientation)*
- (Viewshed generation)
- Photograph georectification
- Manual, automatic blue band or automatic PCA-based snow cover detection*
- (Automatic derivation of an optimal NDSI-based satellite snow cover map)

A bulk mode is also available facilitating the analysis of a complete time series of photographs and it is a user's choice to (de)activitate optional features given in brackets. Direct user interaction within the program evaluation can be activated in modules marked with *.

We herein focus on the application of PRACTISE to derive photography- and satellite-based snow cover maps while we refer the reader to the publication "Härer, S., Bernhardt, M. and Schulz, K. (2015): PRACTISE – Photo Rectification And ClassificaTIon SoftwarE (V.2.0), GMD" and "Härer, S., Bernhardt, M., Corripio, J. G. and Schulz, K. (2013): PRACTISE – Photo Rectification And ClassificaTIon SoftwarE (V.1.0), GMD" if a detailed description of the applied algorithms is needed.

2. PRACTISE examples and input data

The functionality of PRACTISE V.2.0 will be presented analysing the snow cover extent in the Zugspitze massif, Germany, on November 17, 2011. Inputs are a DEM (Fig. 1a), a single lens reflex (SLR) camera photograph (Fig. 1b), a Landsat 7 Enhanced Thematic Mapper (ETM+) scene (Fig. 1c), and a satellite image cloud mask (Fig. 1d). DEM and photograph are supplied in the distribution of PRACTISE for demonstration. The Landsat data can be downloaded in the USGS archives and the cloud mask can be generated using the satellite image and the freely available Fmask software (Zhu et al., 2015; more information in Appendix 1).

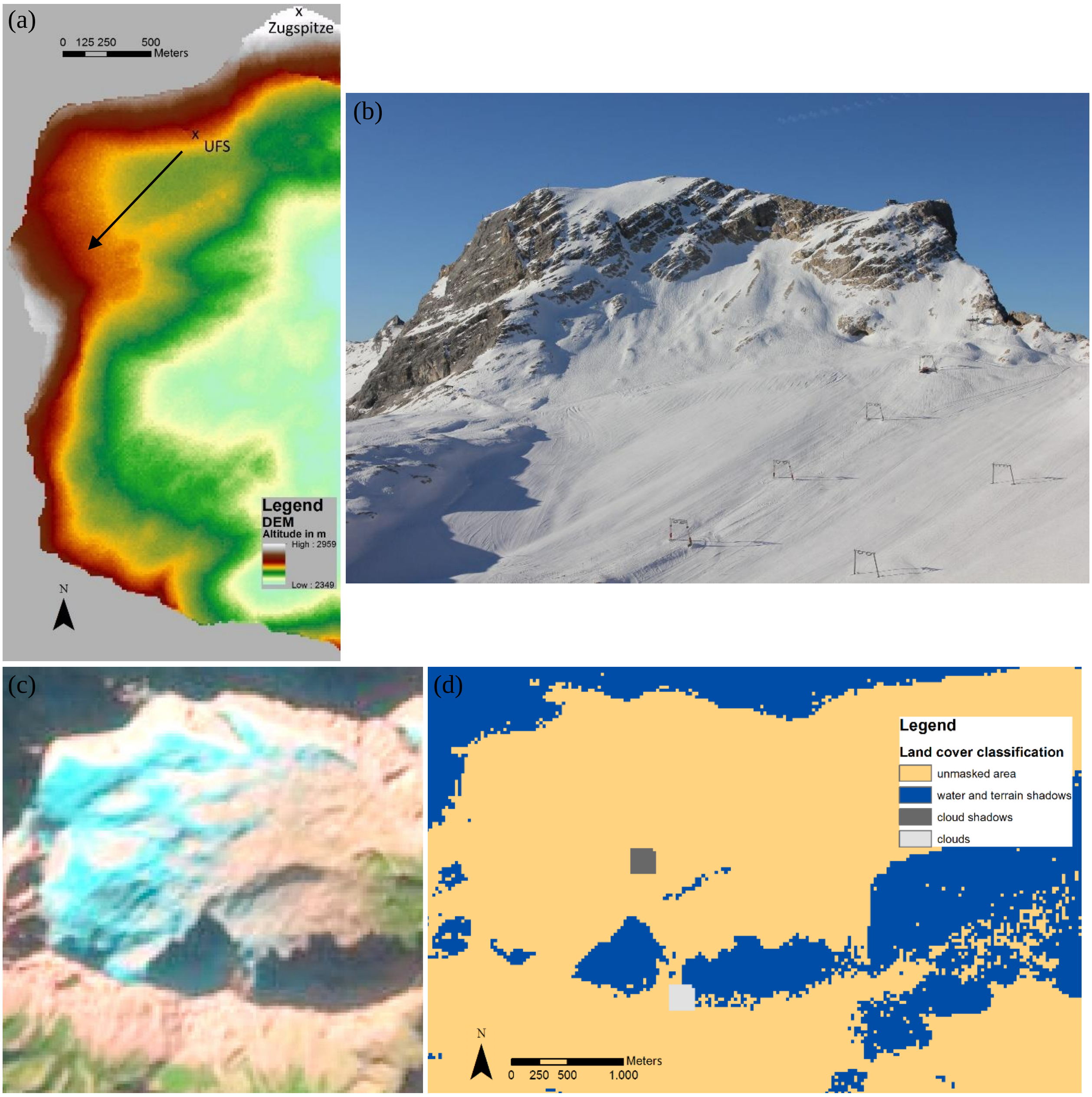


Figure 1: (a) The DEM of the area in the southwest of Zugspitze mountain in the Bavarian Alps (Germany) is distributed together with the manual in a spatial resolution of 10 m. In addition, some random noise is superimposed on the elevation data due to licensing reasons. (b) The SLR photograph is also part of the distribution (4.5 Mpx) and shows the northeast facing slope of Schneefernerkopf recorded from the Environmental Research Station Schneefernerhaus (UFS) on November 17, 2011. The Landsat Look image (c) and the Fmask cloud mask (d) illustrate the simultaneously acquired satellite data of the Zugspitze massif. Note: (c) and (d) have the same scale and the same spatial extent in the North, South and West.

Two input configurations (A and B) and hence two PRACTISE examples will be presented in the manual. Section 2.1 presents mandatory and optional input data of PRACTISE V.2.0 while the Sections 2.2 and 3.1 describe the input file, processing steps and results of configuration A. The input file and results of configuration B are shown in Sections 2.3 and 3.2.

Beforehand, we want to emphasize that from a user's point of view, files with the word *"Input"* in the file name are the only files where changes should be specified by a user. *"Input_PRACTISE.m"* is thereby the organisation file setting the path and name of the input file used, in the examples of *"Input_17112011_A.m"* for configuration A and of *"Input_17112011_B.m"* for configuration B. This setup allows flexibility in the storage paths of input files for different configurations and facilitates switching fast between different program runs without the need for user changes in *"PRACTISE.m"*, the main file of PRACTISE. Please also note that lines in the input files denoted at the end with "`%KEEP`" should not be changed at all, and lines denoted with "`%keep`" only from sophisticated users.

2.1 Mandatory and optional input data

This section gives an overview on mandatory and optional input data in PRACTISE V.2.0 whereas almost all input data options are presented by the two described PRACTISE examples.

Input data that has to be provided by the user for the photograph georectification is:

- obviously a photograph,
- a digital elevation model (DEM),
- the camera location $\boldsymbol{C_o}$,
- the camera target point $\boldsymbol{T_t}$ (centre of the photograph),
- and the inner camera orientation (focal length f, camera sensor width w and height h).

Additional input data needed for the georectification is derived by the program itself, for example the vertical and horizontal pixel dimensions of the photograph (N_v and N_h).

The parameters in the snow classification module also have to be defined by the user. However, parameter changes are normally not necessary except for the manual classification method.

If at least one of the optional modules for camera location and orientation is activated ground control points (GCPs) are needed as an input. The GCPs are necessary for assessing the accuracy of the georectification and for optimising the interior and exterior camera orientation if the camera parameters are not exactly known.

Input required if the optional viewshed generation is deactivated is a viewshed, for example generated in a geoinformation system or a previous PRACTISE run.

At last, the new optional module for calibrating the NDSI threshold needs satellite data as a key input. This can either be unzipped, radiometrically and geometrically corrected (L1G) data of the Landsat satellites 5, 7 or 8 or a NDSI map calculated before from Landsat or other satellite data (e.g. MODIS or SPOT). Optional in this module is the definition of a specific region of interest in the satellite image, and the use of a water, cloud and shadow mask using the Fmask program (Zhu et al., 2015) or similar software.

DEM, viewshed, NDSI map, and also the satellite image cloud mask if not derived with Fmask have to be given in ARC/INFO ASCII GRID format. For information on the GCPs file format, please have a look at the provided example in the PRACTISE distribution.

2.2 Input (configuration A)

This section describes stepwise the input file of configuration A *"Input_17112011_A.m"* in the data folder *"input_17_11_2011_configA"*.

I. The optional modules, optimisation of the camera location and orientation (`os>1`), accuracy assessment of the camera parameters (`os>0`), and generation of a viewshed (`vs=1`) are activated in configuration A. The optimisation of the camera parameters (`os=3`) as well as the snow classification (`cs=3`) are used in interactive mode and the file names of photograph and GCPs file are automatically searched for in the data folder of configuration A *"input_17_11_2011_configA"* (`is=1`). The output folder is also automatically generated in this data folder (`is=1`). The remote sensing modules are deactivated in configuration A (`rs=0`; `rsps=0`; `rsms=0`). (lines 14 to 49 in *"Input_17112011_A.m"*)

II. The input data except the DEM has been physically copied in the photograph input folder *"slr_ufs"* inside the configuration A-data folder and is automatically found by the file extensions *".tif"* and *".gcp.txt"*. Please note that the automatic file name search can only be successful if not more than one viewshed and not more than one GCPs file is stored in the photograph input folder. Lines with `XYZ` in the folder or file name specifications in *"Input_17112011_A.m"* are not utilised in this configuration. (lines 51 to 116 in *"Input_17112011_A.m"*)

III. After the declaration of the input data, the camera location and its field of view are specified (lines 118 to 142 in *"Input_17112011_A.m"*):

a) `cam(:,1)=[6.4929997e+05; 5.25335826e+06]` (longitude and latitude of $\boldsymbol{C_o}$ in metres, the altitude will be calculated using the DEM and adding the first value of the vector `cam_off`)

b) `buffer_radius=250` (radius of a transparent radial zone around the camera location in metres applied in the viewshed generation, more information below)

c) `cam(:,2)=[6.4874085e+05, 5.25277133e+06]` (longitude and latitude of $\boldsymbol{T_t}$ in metres, the altitude will be calculated using the DEM and adding the second value of the vector `cam_off`)

d) `cam_off=[1.5, 0]` (offset height of camera position and camera target position in metres, more information below)

e) `cam_rol=0.0` (roll angle of the camera in degree, where horizontal is equal to 0 and clockwise spans from 0 to +90, while anti-clockwise ranges from -90 to 0)

f) `cam_foc=0.031` (focal length of the camera lens in metres)

g) `cam_hei=0.0149` and `cam_wid=0.0223` (height and width of the camera sensor in metres)

to III. b): `buffer_radius` is only active if a viewshed is generated in the program evaluation. The radius defined here creates a radial buffer zone around the camera location and allows the processing of camera locations sheltered by for example a roof. Here, the camera location would be assumed below ground in a DEM. The pixels in the radial buffer zone are however defined as transparent and hence do not obstruct the view.

to III. d): The values defined by `cam_off` are added to the altitude of the camera location and the camera target position derived from the DEM. The first value of the vector `cam_off` usually represents the installation height of the camera above the DEM surface but can also become negative if the camera is below a roof. The second value of `cam_off` is an option to create an artificial camera target position. This is necessary if the centre of the camera is directed to an area outside of the DEM, for example if the camera points towards the sky.

IV. The next step is the definition of the parameters for the snow classification of the photograph. The different snow classification routines and adapted parameter values can be tested by the user as the interactive mode is activated (`cs=3`). The decision on classification routine and parameters is thus made within the program evaluation. (lines 143 to 157 in *"Input_17112011_A.m"*)

V. The parameters for the satellite image data are not utilised in configuration A and hence, are not described here. (lines 158 to 196 in *"Input_17112011_A.m"*)

VI. Then, the DDS parameters for the optimisation of the camera parameters are defined (lines 199 to 229 in *"Input_17112011_A.m"*):

a) `UBD=[50, 50, 50, 3, 100, 100, 0, 0.0025, 0, 0]` and `LBD=[-50, -50, -50, -3, -100, -100, 0, -0.0025, 0, 0]` (upper and lower boundary deviations from the seed parameters of the optimisation: `cam(:,1)`, `cam_off(1)`, `cam_rol`, `cam(:,2)`, `cam_off(2)`, `cam_foc`, `cam_hei`, `cam_wid`)

b) `DDS_R=0.2` (neighbourhood perturbation size, i.e. the maximum change of a parameter in one iteration step in relation to the corresponding parameter range)

c) `DDS_MaxEval=3000` (maximum number of optimisation iterations)

d) `gcpRMSE_optthres=1` (target value for the root mean square error (RMSE) between calculated and real GCPs in px, more information below)

e) `gcpRMSE_countthres=10` (Maximum number of DDS optimisation restarts to find a RMSE between calculated and real GCPs smaller than `gcpRMSE_optthres`, more information below)

to VI. d) and VI. e): If the interactive mode in the optimisation of the camera parameters is active (`os=3`), the DDS optimisation is restarted when the optimised RMSE between calculated and real GCPs is larger or equal to the value of `gcpRMSE_optthres`. To omit an endless loop of DDS optimisation runs, the DDS optimisation restarts are fixed to a maximum number (`gcpRMSE_countthres`). If the maximum number of runs is reached, the program will propose to utilise the camera parameters of the DDS optimisation run with the lowest RMSE value. The user can then interactively decide to completely restart the optimisation, to optimise the best camera parameters further or to directly apply the new camera parameters in the next module. Please note that it is recommended to use a very small value for `gcpRMSE_optthres` (like 1) and a huge value for `gcpRMSE_countthres` (up to 1000) to ensure optimal georectification results. This is more effective than increasing `DDS_MaxEval`.

VII. The DDS parameters for the optimisation of the NDSI threshold in the satellite image are not used here. (lines 230 to 240 in *"Input_17112011_A.m"*)

VIII. At last, the marker size in the snow classification figure is defined. This can also be adapted later in the figure and is again not necessary in configuration A for the satellite data. (lines 242 to 248 in "Input_17112011_A.m")

2.3 Input (configuration B)

This section describes the input file of configuration B *"Input_17112011_B.m"* in the data folder *"input_17_11_2011_configB"*.

I. Configuration B is using the optimised camera orientation and the generated viewshed (`vs=0`) from a configuration A-run. The optional module accuracy assessment of the camera parameters (`os=1`) is activated and the PCA-based snow classification (`cs=2`) is applied. In addition, the remote sensing modules are switched on using L1G Landsat data (`rs=1`) and a satellite image cloud mask externally generated with the Fmask software (`rsms=2`). Further, all PCA-based snow classification results are included in the NDSI threshold optimisation of the Landsat NDSI data (`rsps=1`). This means that unsure snow classification results in the photograph are used but do have less weight in the objective function which tests the agreement in the overlap area of photograph and satellite image. At last, the automatic file name search of photograph, viewshed and GCPs file in the data folder of configuration

B *“input_17_11_2011_configB”* as well as the automatic output folder generation are active (`is=1`). (lines 14 to 49 in *”Input_17112011_B.m”*)

II. By contrast to configuration A, the input data in configuration B is not physically copied inside the data folder *“input_17_11_2011_configB”* but linking files are used to establish the connection to the input data. Linking files readable by PRACTISE need to have the following file extensions: *“.dem.LINK”* (DEM file), *“.img.LINK”* (photograph file), *“.view.LINK”* (viewshed file), *“.gcp.LINK”* (GCPs file), and *“.satfolder.LINK”* (satellite data folder). The linking files to the input data are text files consisting of a first line with the folder address and a second line for the file name. The linking text file to the satellite folder has only the address to the folder where the mandatory folders for this configuration *“data”* and *“fmask_3_3_0_95”*, and an optional folder *“look”* are inside. Please note that linking files are only used by the program if the automatic file search in the photograph input folder *“slr_ufs”* is not successful for the respective file extensions *“.tif”, ”.view.asc”* and *“.gcp.txt”* and if in the satellite input folder *“le7”* no data folder *“data”* exists. Lines with `XYZ` in the folder or file name specifications in *“Input_17112011_B.m”* are not active in this configuration. (lines 51 to 116 in *”Input_17112011_B.m”*)

III. The parameters for camera location and the field of view used here have been optimised by running PRACTISE in configuration A except that the value of `gcpRMSE_countthres` was increased to 1000 in the interactive optimisation. For a detailed description of camera parameters, we refer to Section 2.2. (lines 118 to 142 in *”Input_17112011_B.m”*)

IV. The decision on the snow classification routine and parameters used for the photograph classification was also interactively made running PRACTISE in configuration A. The PCA-based snow classification routine is applied and the moving average window size smoothing the blue band histogram was increased from the standard value of 5 to 7. (lines 143 to 157 in *”Input_17112011_B.m”*)

V. Then, the parameters for the satellite image data are defined (lines 158 to 196 in *”Input_17112011_B.m”*):

a) First, a region of interest is defined in this case using coordinates for the maximum processing extent in the North, East, South and West:

```
satBB=[5255000.5, 654000.5, 5250199.5, 648099.5]
```

The boundaries can also be set by uncommenting `satBB='full'` for processing the complete satellite image or by uncommenting `satBB=6000` for processing a rectangle buffer area of 6 km in each of the four compass direction around the maximum extent of the photograph snow cover map. The box size can obviously be adjusted. The last uncommented definition of `satBB` in the input file is applied in the program evaluation.

b) Setting date and time of photograph and satellite data acquisition manually is not needed in this configuration (`is=1`; `rs=1`). The date and time of the photograph are automatically derived from the file name of the photograph or with a lower priority from the file name of the photograph linking file (more information below). Date and time of the satellite image are automatically found in the metadata-file (*“.MTL”*) in the satellite data folder *“data”*. Acquisition date and time are necessary for ensuring the comparison of simultaneously captured photography- and satellite-based snow cover maps.

c) The third remote sensing parameter is a code that decides which areas in the satellite image are masked if a satellite image mask externally generated with the Fmask software is applied. Excluded pixels in the satellite image using `satmask_code=[1,2,4]` are water, cloud shadows, and clouds whereas water also masks strongly shaded areas.

To V. b) The file name of the photograph or photograph linking file needs to be written in a specific format so that PRACTISE can automatically derive date and time. It consists of any text at the beginning, then the specific date and time code and at the end the file extension. For the photograph linking file of configuration B, this is *"slr_ufs_"*, then *"20111117_1007utc"* and at the end *".img.LINK"*. The date and time code always has to be given in Coordinated Universal Time (UTC) using *"utc"* at the end.

VI. The DDS parameters for the optimisation of the georectification using GCPs are not used and hence, not described here. (lines 199 to 229 in *"Input_17112011_B.m"*)

VII. The DDS parameters for the optimisation of the NDSI threshold in the satellite image are however utilised. The seed for the NDSI threshold (`NDSIthres0`) is set to the standard literature value of 0.4. The parameter range for the NDSI threshold is theoretically -1 to 1 but in PRACTISE automatically defined by the maximum and minimum NDSI value in the overlap region of photograph snow cover map and satellite NDSI map. The neighbourhood perturbation size is assigned the standard literature value of 0.2 and the maximum number of iterations is set to 150. This is quite conservative for a single value-optimisation and ensures highly reliable optimisation results for the NDSI threshold. (lines 230 to 240 in *"Input_17112011_B.m"*)

VIII. At last, the marker size in the snow classification figure and the satellite snow cover map figure are defined. (lines 242 to 248 in *"Input_17112011_B.m"*)

For configuration B, the organisation file *"Input_PRACTISE.m"* in the folder *"PRACTISE_V_2_0"* has to be adapted. This is described in Appendix 1.

3. Processed PRACTISE examples

The results of PRACTISE V.2.0 in configuration A and B are generated by executing the main file *"PRACTISE.m"* and are saved to output folders with time stamp inside the data folders *"input_17_11_2011_configA"* and *"input_17_11_2011_configB"*. If *"PRACTISE.m"* is executed without any changes to the distribution of PRACTISE V.2.0, configuration A is used. To run configuration B, some data preparation is necessary and the file *"Input_PRACTISE.m"* has to be adapted (see Appendix 1).

The results of the PRACTISE runs are shown for configuration A in Section 3.1 and for configuration B in Section 3.2. In addition, important processing steps of the interactive PRACTISE run in configuration A are illustrated using commented screenshots in Section 3.1.

3.1 Processing steps and results (configuration A)

Interactive processing steps:

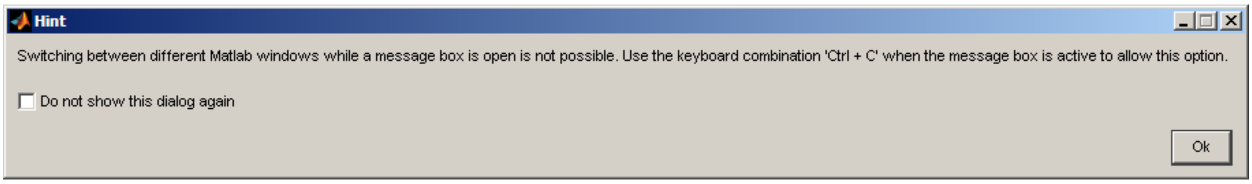


Figure 2: When running for the first time modules in interactive mode, a message pops up that explains how switching between different Matlab windows can be activated on a case by case basis as this is not possible if a message box is open in the foreground. The pop up will occur in every program evaluation with interactive modules until the check mark "Do not show this dialog again" is set.

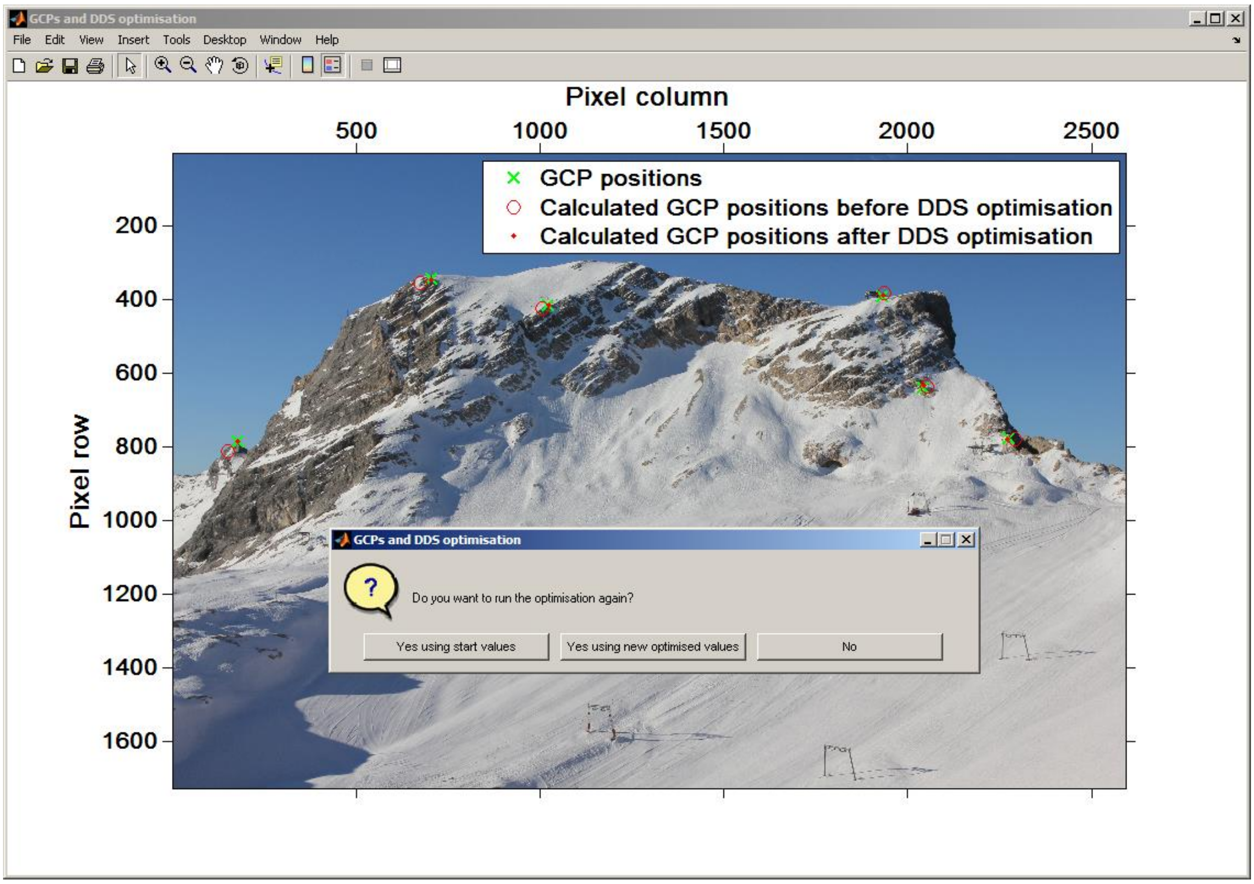


Figure 3: This question dialog opens in the interactive optimisation of the camera parameters when the RMSE between calculated and real GCPs is smaller than gcpRMSE_optthres or when the number of DDS optimisation restarts is equal to gcpRMSE_countthres. Now, the user can either completely restart the optimisation, optimise the run with the lowest RMSE or directly apply the optimised camera parameters in the next module. The GCP optimisation figure in the background illustrates the optimisation result by displaying the convergence of the calculated GCPs, before (red circles) and after (red dots) the optimisation, to the real GCPs (green crosses).

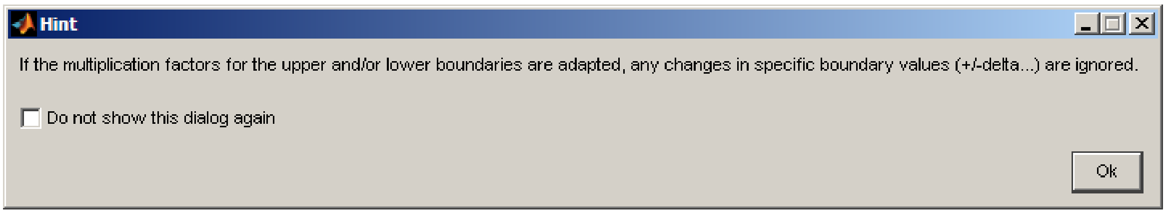


Figure 4: If the user selects to further optimise the new camera parameters, this message box shows up, explaining that if the first two lines of the upcoming dialog box are changed, any changes to the range of a specific camera parameter are ignored. The pop up will occur in every program evaluation with interactive modules until the check mark "Do not show this dialog again" is set.

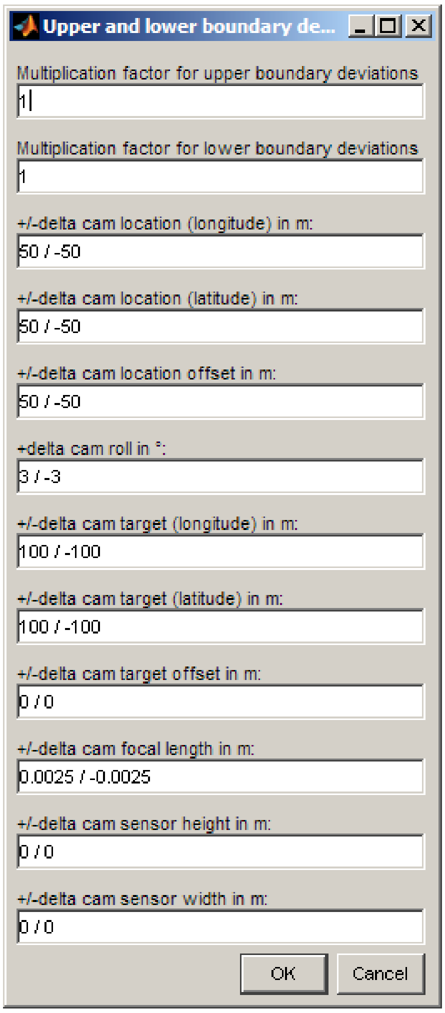


Figure 5: This dialog box can adapt the upper and lower boundaries of the camera parameters when the new camera parameters should be further optimised. This is done either by multiplying the positive (and negative) values of each camera parameter with the numbers in the first (and second) line of the dialog box or by making direct changes to the boundary deviations of specific camera parameters. If nothing is changed in the question dialog, the multiplication factors of 1 are applied and hence, the parameter range does not change.

... now the DDS optimisation run starts and the GCPs and DDS optimisation figure and the question dialog pop up again ... and so on until no further optimisation of camera parameters is selected.

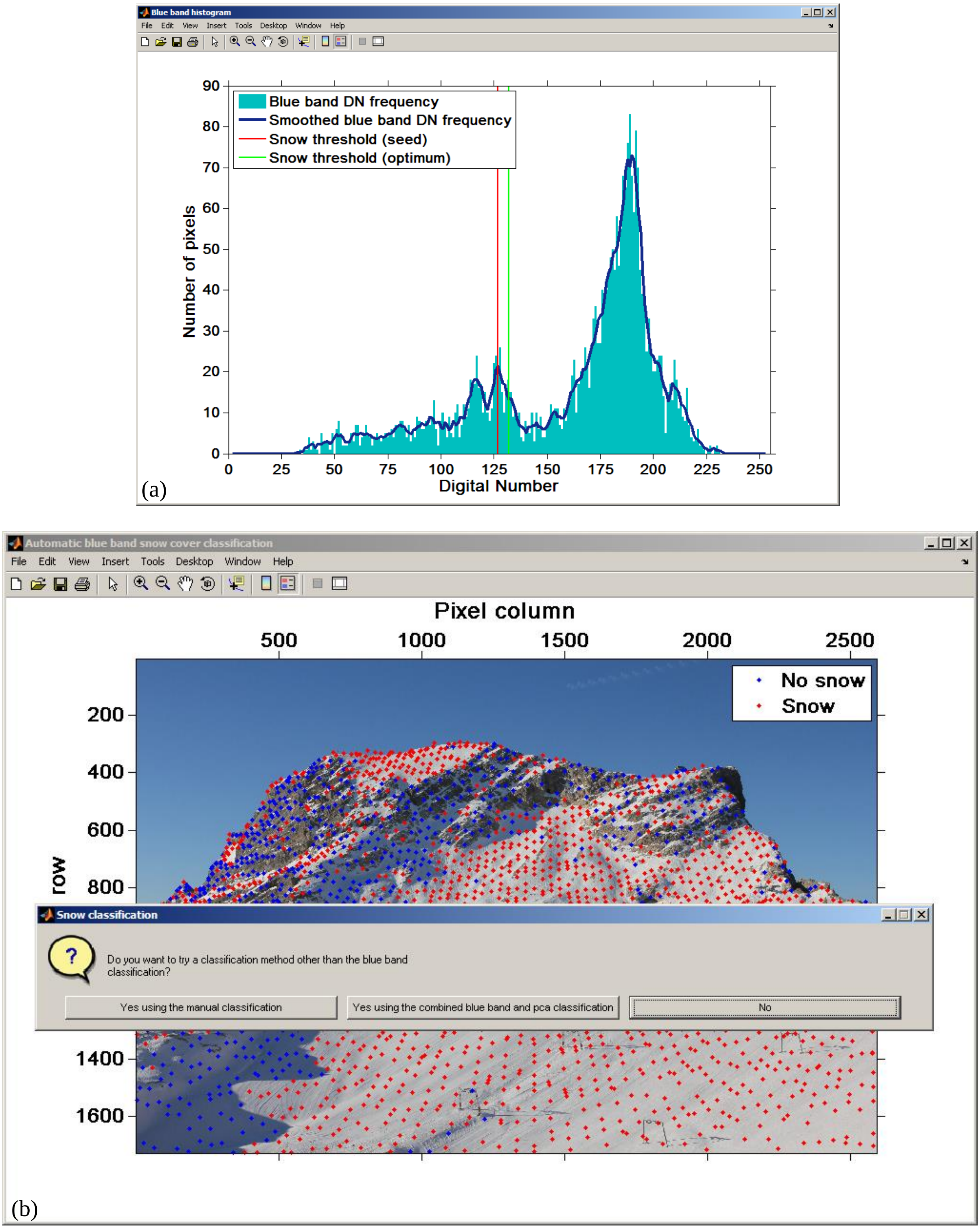


Figure 6: When using the interactive classification mode, the automatic blue band classification is applied as a first guess. (a) A blue band DN histogram is generated and the histogram is smoothed using a moving average with window size 5. The first local minimum larger than or equal to the initial blue band threshold of 127 is defined as new blue band threshold. Blue band digital numbers above or equal to this threshold are classified as snow. (b) The derived snow classification is superimposed on the photograph in another figure and a question dialog pops up. The user can then decide on the quality of the snow classification and either start the next module or test another snow classification routine. Please remember using "Ctrl + C" to allow switching between and zooming in the Matlab figures. The classification result presented has a low quality as shaded snow cover is misclassified as no snow, and as the blue band threshold is not set optimal due to the relatively small number of pixels in the histogram. Please remember, the horizontal spatial resolution of the example DEM is 10m.

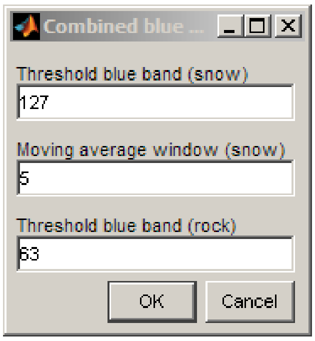


Figure 7: When the combined blue band and PCA snow classification routine is selected, a new dialog box pops up where the blue band threshold for snow (lower limit), the moving average window size and the blue band threshold for rock (upper limit) can be adjusted. It might be necessary here to increase the moving average window size of the blue band DN histogram for stronger smoothing (in this case from 5 to 7) to adapt for the small number of DEM pixels. Note: Always use odd numbers for the window size.

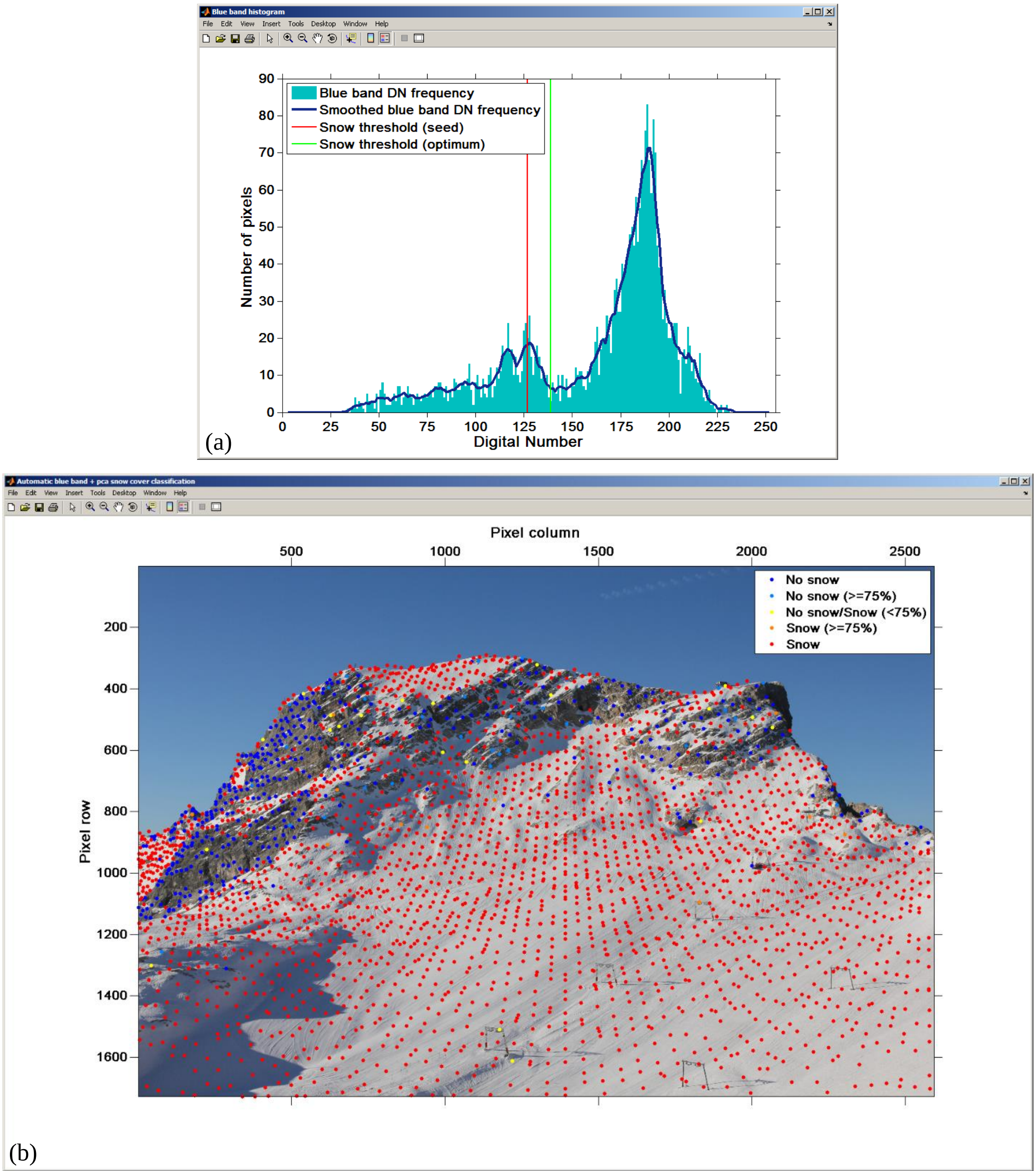


Figure 8: (a) Using the larger window size and hence stronger smoothing, a more pronounced local minimum is detected in the blue band DN histogram. (b) In addition, the PCA-based snow classification routine correctly classifies the shaded snow cover and depicts unsure classification results in the photograph.

Results:

The generated viewshed (Fig. 9 a) and the photograph snow cover map (Fig. 9 b) are stored in ARC/INFO ASCII GRID format in the output folder. Both files can be directly loaded in most geoinformation systems.

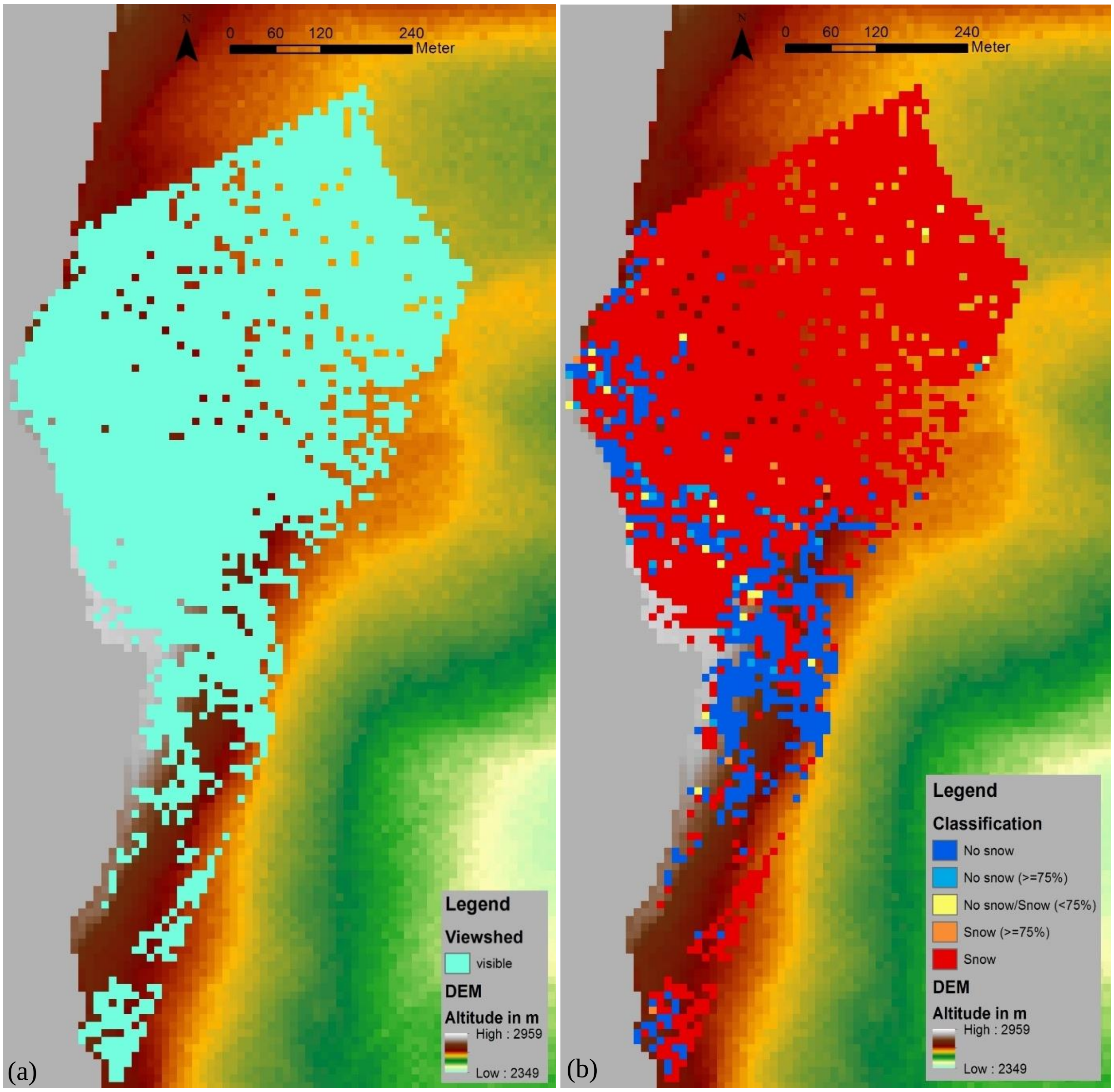


Figure 9: (a) The optimised viewshed and (b) the snow cover map using the PCA-based classification are illustrated for the example run in configuration A for the photograph on November 17, 2011. The large number of single pixel gaps in the viewshed and hence also in the snow cover map is due to the random noise in the DEM data.

Additionally, three Matlab figures (*".fig"*) are stored by default in the output folder:

- Georectification accuracy between real and calculated GCPs (Fig. 3)
- Blue band digital number (DN) histogram of the photograph with snow threshold (Fig. 8 a)
- Superimposed snow cover classification in the photograph (Fig. 8 b)

Other saved files are

- a logfile (*".txt"*) of all screen outputs during the program evaluation,
- a copy of the used input file *"Input_17112011_A.m"*,
- and the Matlab workspace (*".mat"*) including all important input and output variables.

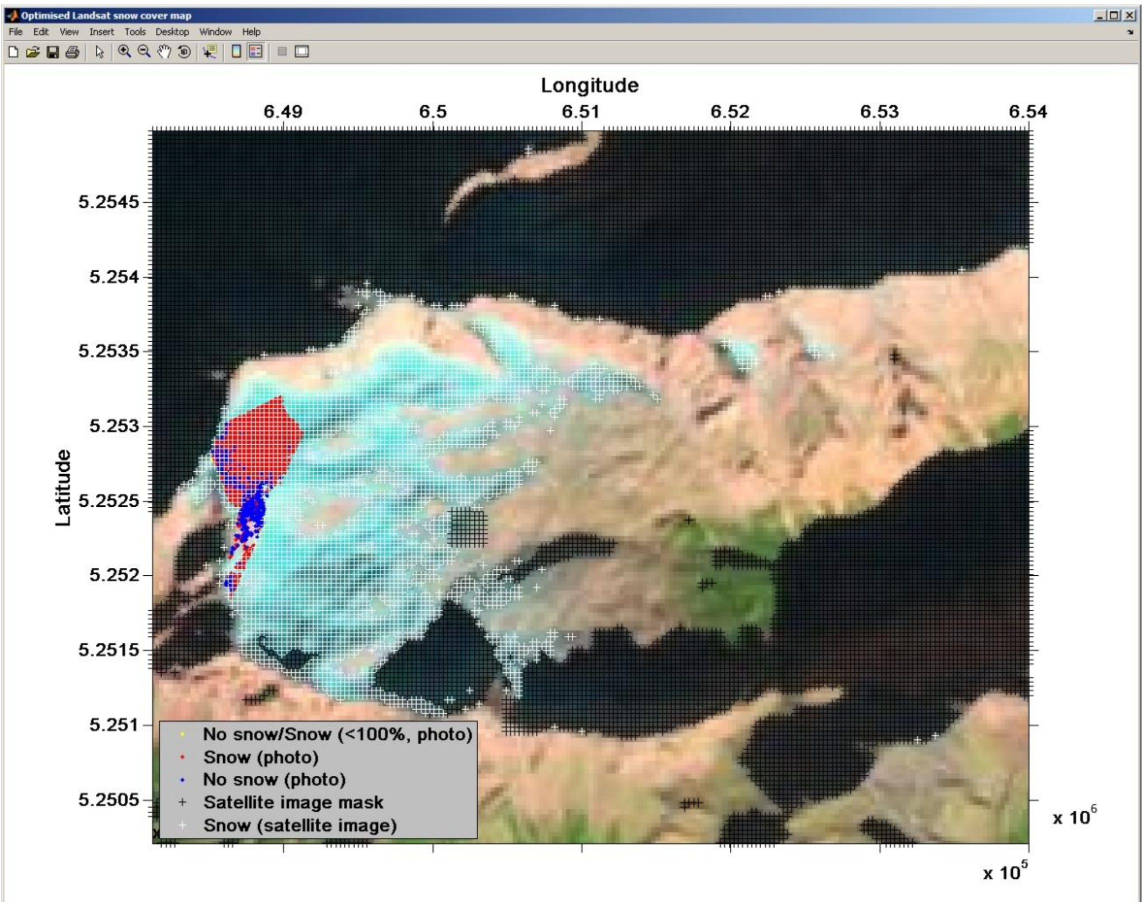
The variable descriptions of the saved Matlab workspace can be found in Appendix 2.

3.2 Results (configuration B)

The resulting files from the PRACTISE run in configuration B are almost identical to the results of configuration A (Sect. 3.1). The differences are that no viewshed is generated in configuration B and that additional remote sensing output data is stored:

- Matlab figure (*".fig"*) superimposing photograph snow cover map, optimised Landsat snow cover map, and satellite image mask on the Landsat Look image (Fig. 10 a and b)
- Landsat NDSI map (ARC/INFO ASCII GRID, Fig. 11 a)
- Optimised Landsat snow cover map (ARC/INFO ASCII GRID, Fig. 11 b)

Both maps can be directly loaded in most geoinformation system. The variable descriptions of the saved Matlab workspace can be found in Appendix 2.



(a)

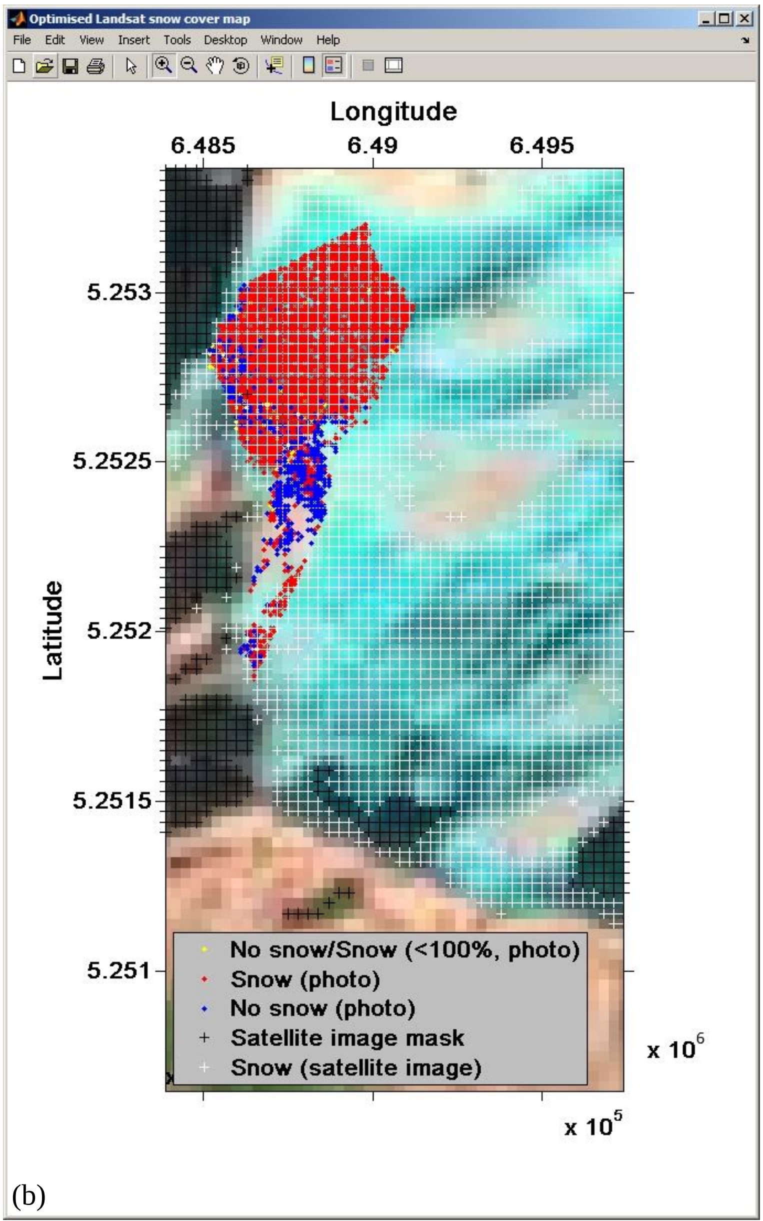


Figure 10: Detected snow cover using Landsat data and the optimised NDSI threshold of 0.17 are superimposed with white crosses on (a) the Landsat Look image of Zugspitze massif and (b) an enlarged view of the area photographed. Pixels masked in the satellite image using the Fmask satellite cloud mask are depicted with black crosses and all remaining areas are classified as free of snow. The photograph snow cover map is also superimposed to allow a comparison between snow cover detected in the Landsat image and the photograph.

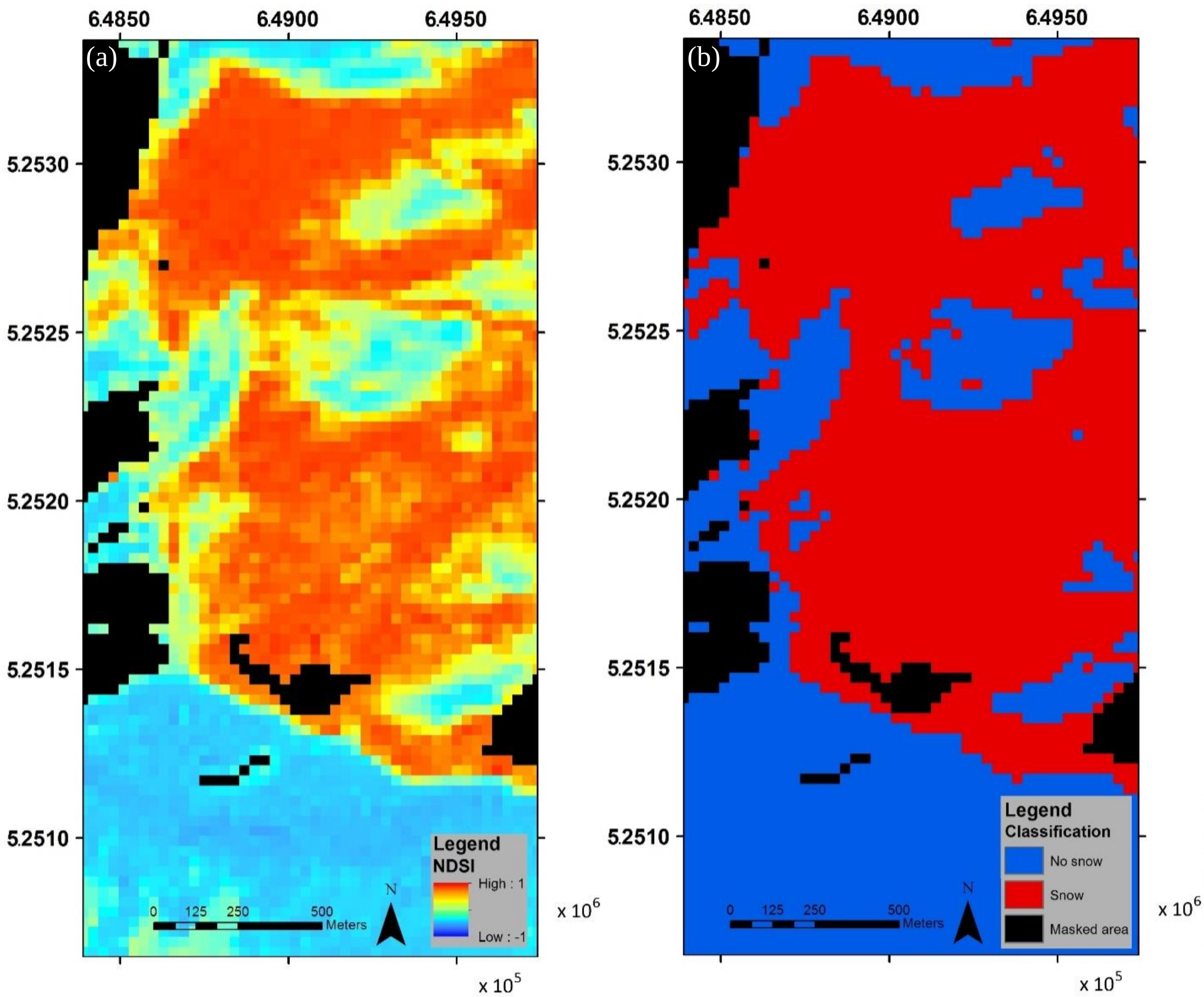


Figure 11: (a) The derived NDSI map and (b) the optimal Landsat snow cover map are shown here for the same extent as the enlarged view of the area photographed (Fig. 10 b). Masked areas using the Fmask satellite cloud mask are depicted in black.

References

Dozier, J.: Spectral signature of alpine snow cover from the Landsat Thematic Mapper, Remote Sens. Environ., 28, 9–22, doi:10.1016/0034-4257(89)90101-6, 1989.

Härer, S., Bernhardt, M., Corripio, J. G., and Schulz, K.: PRACTISE – Photo Rectification And ClassificaTIon SoftwarE (V.1.0), Geosci. Model Dev., 6, 837-848, doi:10.5194/gmd-6-837–2013, 2013.

Zhu, Z., Wang, S., and Woodcock, C. E.: Improvement and expansion of the Fmask algorithm: cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images, Remote Sens. Environ., 159, 269–277, doi:10.1016/j.rse.2014.12.014, 2015.

Appendix 1

Here, some additional information to the input data for running PRACTISE in configuration B is given.

Configuration A can be directly executed after downloading and unzipping the PRACTISE distribution as an example DEM and photograph come along with the distribution. For running PRACTISE in configuration B however some input data preparation by the user is needed.

At first, the L1G Landsat data of Zugspitze area on November 17, 2011 (*“LE71930272011321ASN00.tar.gz”*) cannot be provided in the distribution due to licensing reasons. The data can though be downloaded free of charge from the USGS archives. The satellite data has to be unzipped in the folder *“landsat\le7_17112011\”* and the extracted folder *“LE71930272011321ASN00”* renamed to “data”.

We further recommend downloading the Landsat Look image for this date from the USGS archives (*“LE71930272011321ASN00.zip”*) for visualisation reasons. The data also has to be unzipped in the folder *“landsat\le7_17112011\”* and the extracted folder *“LE71930272011321ASN00”* renamed to “look”.

A last preparation step is the generation of a satellite cloud mask with the Fmask software (Zhu et al., 2015). The software can be downloaded from *https://code.google.com/p/fmask/*. Installation and running instructions are also given there. The only difference to the standard procedure is that we mask areas with a cloud probability of 95%. The Fmask run will generate two files with the word *“Fmask”* in the file name inside the satellite image folder *“data”*. Copy both files to the already existing folder *“fmask_3_3_0_95”*.

PRACTISE can then be executed in configuration B if the organisation file *“Input_PRACTISE.m”* in the folder *“PRACTISE_V_2_0”* is also adapted according to this screenshot:

Input_PRACTISE.m

```
% Haerer, Bernhardt and Schulz (2015)
% "PRACTISE - Photo Rectification And ClassificaTIon SoftwarE (V.2.0)"
%
%   written by
%   Stefan Härer (LMU Munich)
%   08/2012
%   contact: stefan.haerer@boku.ac.at
%       updated by Stefan Härer (BOKU Vienna, 06/07/2015)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Name:       Input_PRACTISE
%   Purpose:    Organisation file for the input of PRACTISE
%   Comment:    This file defines and loads the input m-file. Edit the
%               variable with the path and file name of the input m-file as
%               needed, but leave the run-expression untouched. Comment and
%               uncommenting line 20 and 22 here facilitates switching fast
%               between different input m-files.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define path and file name of the input m-file
%     for configuration A
% Input_PRACTISE_m='..\17112011_configA\Input_17112011_A'; % uncomment/comment in case
%     for configuration B
Input_PRACTISE_m='..\17112011_configB\Input_17112011_B'; % uncomment/comment in case

% Load the input m-file
run(Input_PRACTISE_m)                                                      %KEEP
```

Appendix 2

A complete list and description of variables saved in the Matlab workspaces in configuration A and B is presented in the following table. Variables that might occur with a different choice of switches are declared in grey.

Variable name	**Description**
DDS_MaxEval_os	Maximum number of optimisation iterations (camera parameters); os>1
DDS_MaxEval_rs	Maximum number of optimisation iterations (NDSI threshold); rs>0
DDS_R_os	Neighbourhood perturbation size, i.e. the maximum change of a parameter in one iteration step of the optimisation (in relation to the camera parameter range); os>1
DDS_R_rs	Neighbourhood perturbation size, i.e. the maximum change of a parameter in one iteration step of the optimisation (in relation to the NDSI threshold range); rs>0
LB_NDSI	Vector(1,10): Lower boundary of NDSI threshold; rs>0
LB_gcp	Vector(1,10): Lower boundary of camera parameters before last successful optimisation; os>1
LB_gcp_orig	Vector(1,10): Lower boundary of camera parameters before optimisation, does not exist if identical to LB_gcp; os>1
NDSIOpt	Disagreement between overlapping photograph and satellite pixel classifications for optimised NDSI threshold in percent; rs>0
NDSIbct10	Disagreement between overlapping photograph and satellite pixel classifications for initial NDSI threshold in percent; rs>0
NDSIbct1Opt	Disagreement between overlapping photograph and satellite pixel classifications for optimised NDSI threshold in percent, equal to NDSIOpt; rs>0
NDSIthres0	Initial NDSI threshold; rs>0
NDSIthresOpt	Optimised NDSI threshold, rounded to two decimal places; rs>0
NDSIthresOpt_detail	Optimised NDSI threshold; rs>0
N_gcpW	Number of GCPs; os>0
PhRs	Array(2, Number of projected DEM pixels): Photograph snow classification and satellite NDSI value of the DEM pixel (1st column), same for the next DEM pixel (2nd column), …; rs>0
UB_NDSI	Upper boundary of NDSI threshold set by highest NDSI value in the overlap region of photograph and satellite image; rs>0
UB_gcp	Vector(1,10): Upper boundary of camera parameters before last successful camera parameter optimisation; os>1
UB_gcp_orig	Vector(1,10): Upper boundary of camera parameters before optimisation, does not exist if identical to LB_gcp; os>1
Vert	Vector(2,1): Boundaries of the vertical viewing angle of $\boldsymbol{C_o}$; from maximum to minimum; horizontal is 0, upwards is positive, downwards is negative in radiant; vs=1
X0	Vector(1,10): Initial camera parameters consisting of longitude and latitude of $\boldsymbol{C_o}$, cam_off(1), cam_rol, longitude and latitude of $\boldsymbol{T_t}$, cam_off(2), cam_foc, cam_hei and cam_wid, all parameters in metres except cam_rol in degree
Xopt	Vector(1,10): Camera parameters after the last successful optimisation consisting of longitude and latitude of $\boldsymbol{C_o}$, cam_off(1), cam_rol, longitude and latitude of $\boldsymbol{T_t}$, cam_off(2), cam_foc, cam_hei and cam_wid, all parameters in metres except cam_rol; os>1

azi_sec_name	Name of the compass direction(s) used in the viewshed generation; vs=1
buffer_radius	Radius around camera location assumed transparent in viewshed generation in metres; vs=1
cam	Array(3,2): Longitude, latitude and altitude of $\boldsymbol{C_o}$ in metres (1st column), same for $\boldsymbol{T_t}$ (2nd column); if os>1, then optimised values, else original values
cam_0	Array(3,2): Initial longitude, latitude and altitude of $\boldsymbol{C_o}$ in metres (1st column), same for $\boldsymbol{T_t}$ (2nd column); os=3, does not exist if identical to cam_orig
cam_foc	Focal length of the camera lens in metres
cam_hei	Height of the camera sensor in metres
cam_off	Vector(1,2): Offset of the camera location and camera target point above or below the surface in metres
cam_orig	Array(3,2): Longitude, latitude and altitude of $\boldsymbol{C_o}$ in metres (1st column), same for $\boldsymbol{T_t}$ (2nd column); os>1, if cam_0 exists, optimised values before last successful DDS optimisation, else original values
cam_rc	Array(3,2): column position, row position of $\boldsymbol{C_o}$ in the DEM grid and altitude of $\boldsymbol{C_o}$ in metres divided by DEM pixel length (1st column), same for $\boldsymbol{T_t}$ (2nd column); if os>1, then optimised values, else original values
cam_rc_0	Array(3,2): Initial column position, row position of $\boldsymbol{C_o}$ in the DEM grid and altitude of $\boldsymbol{C_o}$ in metres divided by DEM pixel length (1st column), same for $\boldsymbol{T_t}$ (2nd column); if os=3 and cam_0 exists
cam_rc_orig	Array(3,2): column position, row position of $\boldsymbol{C_o}$ in the DEM grid and altitude of $\boldsymbol{C_o}$ in metres divided by DEM pixel length (1st column), same for $\boldsymbol{T_t}$ (2nd column); os>1, if cam_0 exists, optimised values before last successful DDS optimisation, else original values
cam_rol	Roll angle of the camera in degree; horizontal is 0, clockwise spans from 0 to +90, anti-clockwise from -90 to 0
cam_wid	Height of the camera sensor in metres
cs	Switch for the used classification method
cs3	Switch for the classification method; cs=3
delta_rgb	Maximum of the RGB difference (delta of max(RGB) to min(RGB)) in a “snow” pixel; cs=0
demW	Array(3,Number of projected DEM pixels): Longitude, latitude and altitude of a DEM pixel in metres (1st column), same for the next DEM pixel (2nd column), …
fin_demW	Location and name of DEM file
fin_folder	Location of input folders (for photograph with in case GCPs file and/or viewshed and for satellite data)
fin_gcpW	File name of GCPs file; os>0
fin_gcpW_path	Location of GCPs file; os>0
fin_image	File name of photograph
fin_imagepath	Location of photograph
fin_images	Vector(Number of photographs,1): File names of photographs; if is=1 and number of photographs>1
fin_satfolder_image	Location of satellite image in fin_satpath; rs>0
fin_satfolder_look	Location of Landsat Look image in fin_satpath; rs=1
fin_satfolder_mask	Location of satellite image mask in fin_satpath; rsms>0
fin_satname_DistEtoS	File name of earth-sun distance file according to the day of the year; if rs=1 and Landsat 5 or 7

fin_satname_RadSol	File name of solar spectral irradiances file for the Landsat data; if rs=1 and Landsat 5 or 7
fin_satname_look	File name of Landsat Look image; rs=1
fin_satname_mask	File name of satellite image mask; rsms>0
fin_satname_meta	File name of radiometrically and geometrically Landsat metadata file; rs=1
fin_satname_ndsi	File name of satellite NDSI map; rs=2
fin_satpath	Location of satellite input folders (satellite image and in case satellite image mask and/or Landsat Look image); rs>0
fin_viewW	File name of the input viewshed; vs=0
fin_viewW_path	Location of input viewshed; vs=0
fout_classW_asc	File name of derived photograph snow cover map
fout_folder	Location of output files
fout_mat	File name of saved Matlab workspace
fout_ndsi_asc	File name of calculated NDSI map; rs=1
fout_satmap_asc	File name of derived satellite image snow cover map; rs>0
fout_viewW_asc	File name of calculated viewshed; vs=1
gcpP	Array(2,N_gcpW): column and row position of a GCP in the image plane (1st column), same for the next GCP (2nd column), …; os>0, if os>1, then optimised values, else original values
gcpP_0	Array(2,N_gcpW): Initial column and row position of a GCP in the image plane (1st column), same for the next GCP (2nd column), …; if os=3 and cam_0 exists
gcpP_orig	Array(2,N_gcpW): column and row position of a GCP in the image plane (1st column), same for the next GCP (2nd column), …; os>1, if cam_0 exists, optimised values before last successful DDS optimisation, else original values
gcpRMSE_0	Initial root mean square error (RMSE) between calculated and real GCP positions; if os=3 and cam_0 exists
gcpRMSE_countthres	Maximum number of camera parameter DDS optimisation restarts; os=3
gcpRMSE_opt	Minimised root mean square error (RMSE) between calculated and real GCP positions; os>1
gcpRMSE_optthres	Stop criterion (maximum root mean square error (RMSE) between calculated and real GCP positions) for DDS optimisation restarts of camera parameters; os=3
gcpRMSE_orig	Root mean square error (RMSE) between calculated and real GCP positions; os>0, if os>1 and cam_0 exists, then optimised values before last successful DDS optimisation, else original values
gcpW	Array(5,N_gcpW): longitude, latitude, altitude, column and row position of a GCP (1st column), same for the next GCP (2nd column), …; os>0
gcpW_name	Vector(1, N_gcpW): Names of the GCP positions; os>0
header_W	Vector(6,1): Complete headerlines (names and values) of the DEM file in ARC/INFO ASCII GRID format
header_satmask	If rsms=1 (ARC/INFO ASCII GRID format), then vector(6,1), elseif rsms=2 (ENVI GRID format), then vector(12,1): Complete input headerlines (names and values) of the satellite image mask file
headerv_W	Vector(6,1): Values of header_W
headerv_satmask	If rsms=1, then vector(6,1), elseif rsms=2, then vector(12,1): Values of header_satmask
i5050	Vector(Number of pixels classified as unsure, 1): column indices of pixels classified as unsure in the combined blue band and PCA classification; cs=2

irock	Vector(Number of pixels classified as free of snow, 1): column indices of pixels classified as free of snow in the combined blue band and PCA classification; cs=2
is	Switch for the classification of single or multiple photographs with automatic file name search in case
isnow1	Vector(Number of pixels classified as snow using blue band digital number (DN) histogram, 1): column indices of pixels classified as snow in the automatic blue band classification; if cs=1 or cs=2
isnow2	Vector(Number of pixels classified as snow using PCA method, 1): column indices of pixels classified as snow in the PCA classification; cs=2
movavgwindow	Size of the window to calculate a moving average of rgbhist; if cs=1 or cs=2
os	Switch for accuracy assessment and optimisation of C_o and the field of view using GCPs
pca	Array(Number of projected DEM pixels,3): Projection of the red, green and blue (RGB) digital numbers (DN) to the principal component space, scaled between 0 and 1; cs=2
pix_c	Number of pixel columns in the photograph
pix_r	Number of pixel rows in the photograph
rgb_pc	Array(3,3): Principal component (PC) eigenvectors with decreasing eigenvalues (explained variance) from column 1 to 3; cs=2
rgbhist1	Histogram of extracted DEM pixels in the blue band; if cs=1 or cs=2
rgbhistmean1	Calculated moving average of rgbhist; if cs=1 or cs=2
rgbimage	If rs=0, then array(8,Number of projected DEM pixels): 1st and 2nd row, row and column position in the DEM grid; 3rd and 4th row, column and row position in the image plane; 5th to 7th row, red, green and blue band digital number (DN) in the corresponding photograph pixel; and 8th row, the snow classification of the DEM pixel (1st column), same for the next DEM pixel (2nd column), … If rs>0, then array(11,Number of projected DEM pixels): 1st to 8th row, identical to rgbimage when rs=0; 9th row, not a number; 10th to 11th row, longitude and latitude of the DEM pixel (1st column), same for the next DEM pixel (2nd column), …
rs	Switch for the remote sensing module
rsms	Switch to use a satellite image mask in the remote sensing module
rsps	Switch to use or to omit utilising unsure photograph snow classifications in the remote sensing module
satBB	Vector(1,4): Initial region of interest in satellite image clockwise defined by coordinates for North, East, South and West; rs>0
satBBext	Vector(1,4): Processed region of interest in satellite image including all satellite pixels with pixel centre in satBB, clockwise defined by coordinates for North, East, South and West; rs>0
satBBg_dn	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Digital numbers (DN) of the green band (~0.55 µm) Landsat data in satBBext; rs=1
satBBg_rad	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Spectral radiance at the sensor's aperture in W/(m² sr µm) of the green band (~0.55 µm) Landsat data in satBBext; if rs=1 and Landsat 5 or 7

satBBg_refl	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Planetary reflectance of the green band (~0.55 µm) Landsat data in satBBext using a normalization for solar irradiance; rs=1
satBBin_ext	User-defined region of interest in satellite image either as rectancle buffer zone of a specified distance in m around the maximum extent of the photograph snow cover map or as full satelite image extent ('full'); rs>0, does not exist if the four coordinates of satBB are given as input
satBBlat	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Latitude values of the satellite image pixels in satBBext; rs>0
satBBlong	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Longitude values of the satellite image pixels in satBBext; rs>0
satBBmir_dn	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Digital numbers (DN) of the mid infrared band (~1.6 µm) Landsat data in satBBext; rs=1
satBBmir_rad	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Spectral radiance at the sensor's aperture in W/(m^2 sr µm) of the mid infrared band (~1.6 µm) Landsat data in satBBext; if rs=1 and Landsat 5 or 7
satBBmir_refl	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Planetary reflectance of the mid infrared band (~1.6 µm) Landsat data in satBBext using a normalization for solar irradiance; rs=1
satBBndsi	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): NDSI values [-1, 1] of the satellite image pixels in satBBext, including areas masked [2]; rs>0
satBBnir_dn	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Digital numbers (DN) of the near infrared band (~0.85 µm) Landsat data in satBBext; rs=1
satBBnir_rad	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Spectral radiance at the sensor's aperture in W/(m^2 sr µm) of the near infrared band (~0.85 µm) Landsat data in satBBext; if rs=1 and Landsat 5 or 7
satBBnir_refl	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Planetary reflectance of the near infrared band (~0.85 µm) Landsat data in satBBext using a normalisation for solar irradiance; rs=1
satDOY	Day of the year of the Landsat data; if rs=1 and Landsat 5 or 7
satDistEtoS	Daily earth-sun distances in Astronomical Units; if rs=1 and Landsat 5 or 7
satHeader	Complete output headerlines (names and values) of satellite image snow cover map and in case the Landsat NDSI map in ARC/INFO ASCII GRID format; rs>0
satJDAY	Serial date number with the whole and fractional number of days counted from the reference point January 1, 0000; rs>0
satMETA	Structure: Metadata information of the satellite image including date and time, sun elevation, upper left corner longitude and latitude coordinates, number of pixel rows and columns, cell size, no data value; rs>0, if rs=1, then additionally information on green, mid infrared and near infrared band file names and parameters to calculate planetary reflectance from digital numbers (DN) according to the Landsat user handbook
satMTL	Structure: Complete metadata information of Landsat metadata file

satRadSol	Band-specific solar spectral irradiances in W/(m² sr µm) using the combined Chance-Kurucz Solar Spectrum within the software MODTRAN 5 according to the Landsat user handbook; if rs=1 and Landsat 5 or 7
satmaskBB	Array(Number of processed satellite pixel rows, Number of processed satellite pixel columns): Logical values of the satellite image cloud mask in satBBext with masked areas being one and areas not masked being zero or no data; rsms>0
satmaskEXT	Vector(1,4): Initial extent of the satellite image cloud mask clockwise defined by coordinates for North, East, South and West; rsms>0
satmask_code	Vector(1,Number of values used to mask areas in satellite image): Values used to mask areas in satellite image; rsms>0, if rsms=1, then all values not zero and not a no data value in satellite image cloud mask
thres_b_image1	Calculated minimum snow threshold in the blue band searching the first local minimum in rgbhistmean greater than or equal to thres_b_orig; if cs=1 or cs=2
thres_b_low	First guess of the maximum rock threshold in the blue band; cs=2
thres_b_orig	First guess of the minimum snow threshold in the blue band; if cs=1 or cs=2
time	Structure: Recording date and time of the photograph; rs>0
thres_rgb	Vector(3,1): Minimum snow threshold in the red, green and blue band; cs=0
viewAzi	Array(2,2): Boundaries of the azimuth viewing angle at $\boldsymbol{C_o}$, clockwise and with north being 0 in radiant (1st column), sector numbers for the compass direction(s) of the boundaries, clockwise and with the NE sector being 1 (2nd column); vs=1
viewAziSec	Sector numbers for the compass direction(s) used in the viewshed generation, clockwise and with the NE sector being 1; vs=1
vs	Switch for the viewshed generation