

**Supplement to:**

**Novel techniques for void filling in glacier elevation change data sets**

Thorsten Seehaus(1), Veniamin I. Morgenshtern(2), Fabian Hübner(2), Eberhard Bänsch(3), Matthias H. Braun(1)

(1) Institute of Geography, Friedrich-Alexander University Erlangen-Nürnberg, Germany

(2) Chair of Multimedia Communications and Signal Processing, Friedrich-Alexander University Erlangen-Nürnberg, Germany

(3) Department of Mathematics, Friedrich-Alexander University Erlangen-Nürnberg, Germany

## Implementation of the novel inpainting approaches

Both, the Telea and the Navier-Stokes inpainting approach, are implemented in the Open Source Computer Vision (OpenCV) library (<https://github.com/opencv/opencv>). It is a collection of programming functions for digital image processing, written in C++, with existing bindings and wrappers for other programming languages.

The processing of the DEMs and elevation change maps, like coregistration, artificial void generation, as well as the analysis of the inpainting results is implemented in R (<https://www.r-project.org/>) using related libraries. Thus, we included the Navier-Stokes and Telea inpainting in the R processing pipeline, allowing a semi-automatic multiparameter analysis.

For the compiling and installation of OpenCV in R, the utility package ROpenCVLite (<https://github.com/swarm-lab/ROpenCVLite>) was used. The Rvision library (<https://github.com/swarm-lab/Rvision>) is needed to use the OpenCV library in R. It provides functions to import and edit images using the power of the OpenCV library.

A voided elevation change data set and a mask, indicating the areas that need to be inpainted, have to be provided to carry out the inpainting using OpenCV. Further, it is necessary to specify the radius of the neighborhood surrounding the voids that is used to supply the initial information to the inpainting algorithms. We applied search radii of 2, 5, 8, 10, 15 and 20 pixels to evaluate the dependency of the inpainting results on the considered area around the voids.

Since the void filling is only carried out on the glaciated areas, voids touching the glacier margins need to be handled with care. When converting the on-glacier elevation change maps (GeoTIFF format) into the OpenCV image format, “no-data” pixels outside the glaciers are treated as zero values, since the OpenCV image format does not support “no-data” or “NaN” values. Pixel information surrounding the data voids are used to constrain the void filling. Glaciers can show considerable surface elevation changes next to their margins. Thus a boundary condition of zero elevation changes at the glacier margins can bias the filling of data voids at the glacier margins towards zero, which is not desired. In a similar way, the consideration of measured elevation changes in the ice-free areas, which should be theoretically close to zero, would cause a bias towards zero. Therefore, we added a 1 km wide buffer zone around the glacier outlines, which was inpainted as well. By doing this, the influence of the zero values in the ice free areas on data voids situated at the glacier margins is sufficiently minimized and the void filling is dominated by on-ice pixel information.

The whole study area extends over 6033x9883 pixels and the void pixels on glacier areas amounts to 1,287,899 pixels for the Correlation setup. Additionally 6,287,250 pixels needed to be inpainted by adding the 1km buffer zone around the glacier areas. The Navier-Stokes and Telea inpainting was run on Intel(R) Xeon(R) E5-2650 v4 CPU and lasted between 70 and 140 min, depending on the algorithm (in general longer for Telea) and search radius.

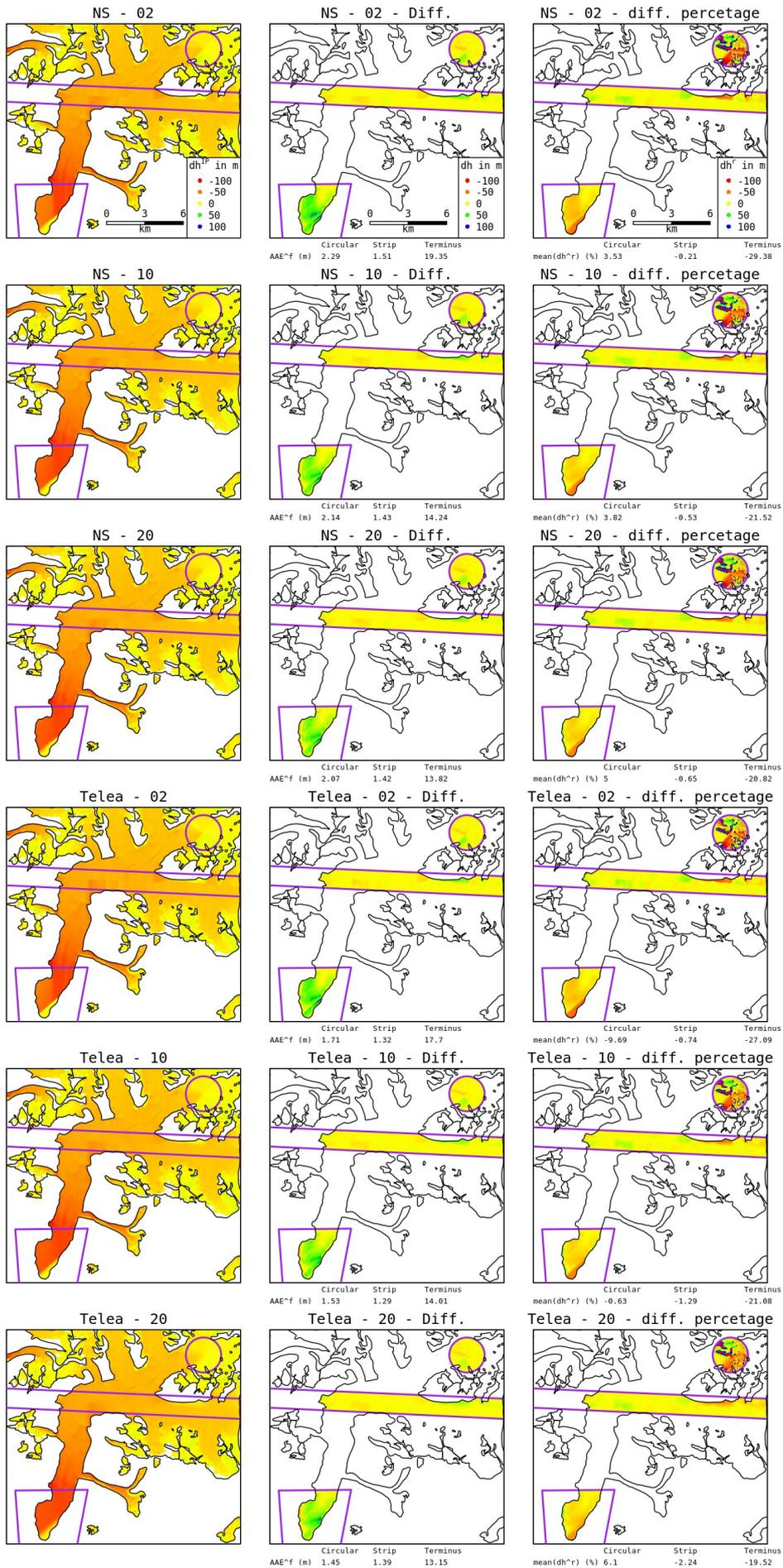
The Shearlet inpainting was implemented in Python. For the Shearlet transform the pyshearlab library (<https://github.com/stefanloock/pyshearlab>) was used and the iterative thresholding was translated from the Matlab code from [www.shearlab.org](http://www.shearlab.org) [1] to Python.

The Shearlet transform is parameterized and allows to adjust the number and the shape of the basis functions. To inpaint the large void areas of the data set, a system with large basis functions is required. For all experiments, we set the nscales parameter, which specifies the number of scale levels and therefore the size of the largest basis functions to the maximum value that the software framework allows. For the remaining parameters, the default values were used.

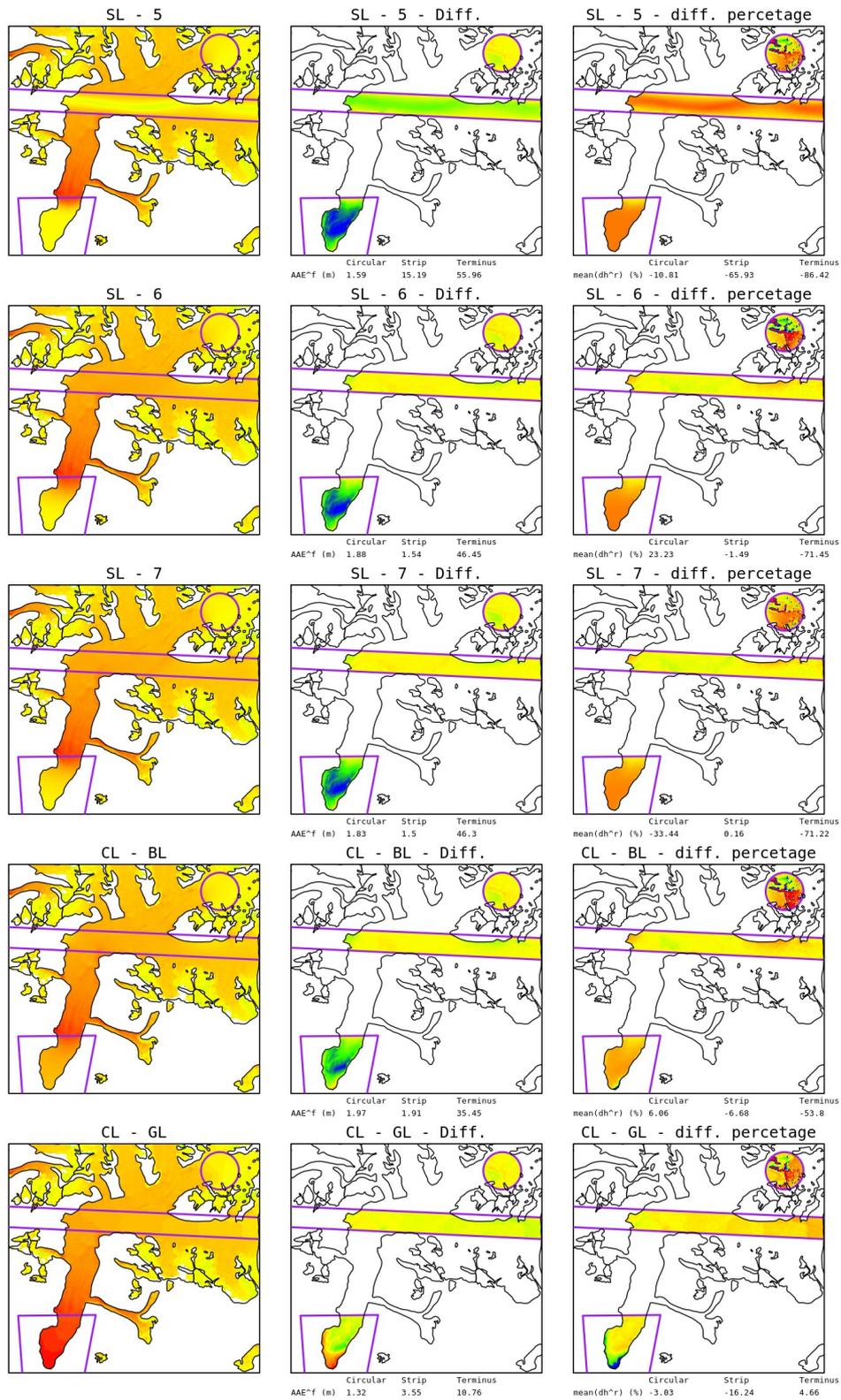
Due to the large image size for the Correlation and SAR setups, a block wise processing scheme with overlapping patches was applied. The overlapping regions have been smoothly joined using trigonometric window functions to reduce border artifacts.

In total 3 experiments with different parameter settings were performed. For the first 2 experiments, the patch size, the stride (i.e., the amount the window is moved to extract a patch) and the nscales parameter were chosen as follows: (i): patch size: 512, stride: 384, nscales: 5; (ii): patch size: 1024, stride: 768, nscales: 6. In the third experiment (iii) we used a patch size of 2048 and a stride of 1536. This time the patches were first subsampled to a size of 1024, then inpainted with a nscales parameter of 6 and finally upsampled to a size of 2048 and merged. The third approach has a similar effect as a direct inpainting with a patch size of 2048 and a nscales parameter of 7 (thus, called nscales 7 in the following) and was performed since the direct processing with these parameters was not possible due to memory limitations. All experiments were executed with 400 iterations and an  $\alpha=0.001$  .

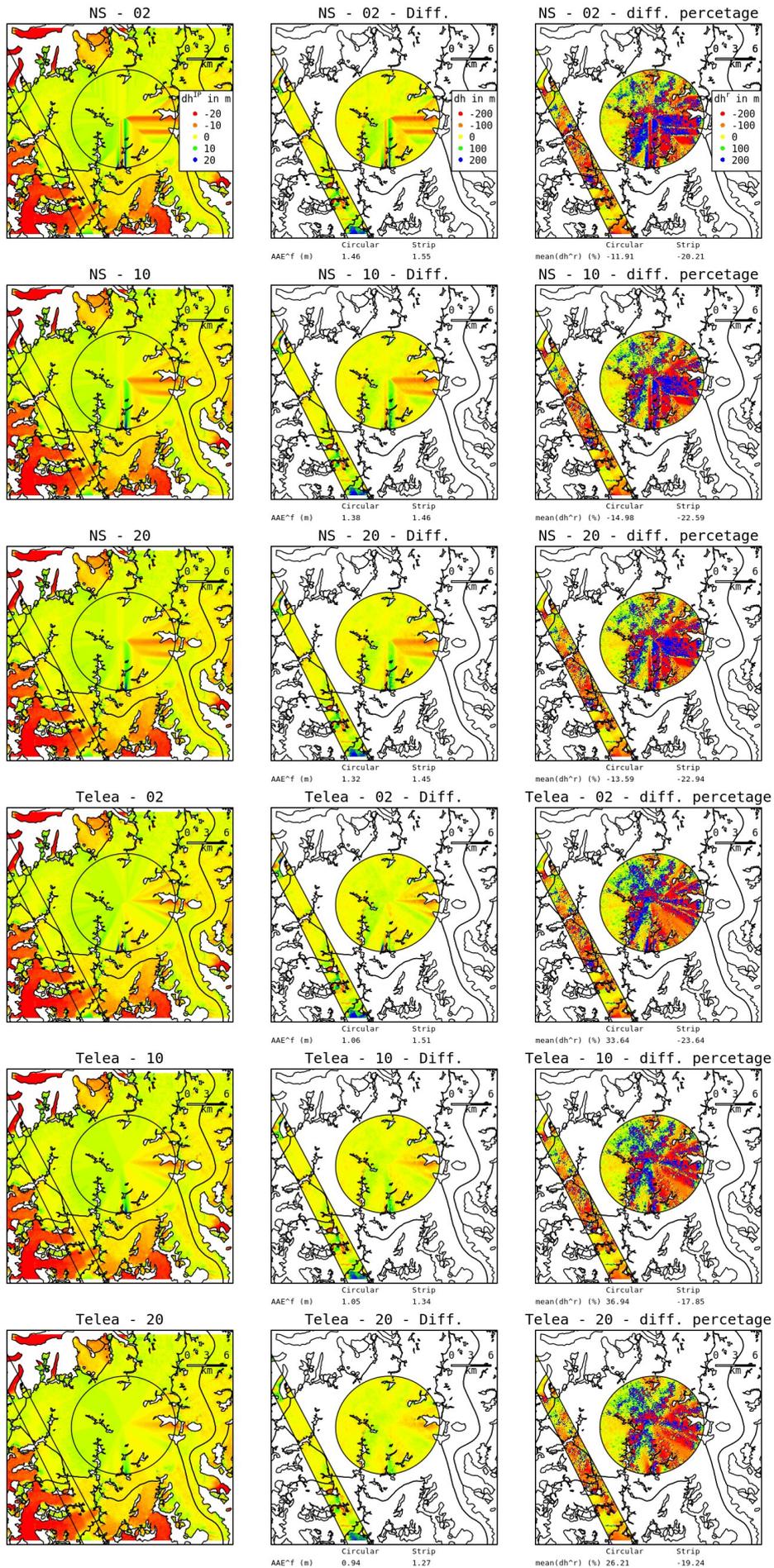
All experiments were performed on an Intel(R) Xeon(R) Silver 4114 CPU. The patches were processed in parallel by 8 processes and execution took 3 to 5 days, depending on the parameter setting. We would like to stress that the optimization of the computing time of the Shearlet approach was beyond the scope of the project; it is likely that using precompiled code and/or GPUs it is possible to speedup the Shearlet-based inpainting significantly. Due to the high computational expenses Shearlet inpainting was not directly included in the R processing pipeline.



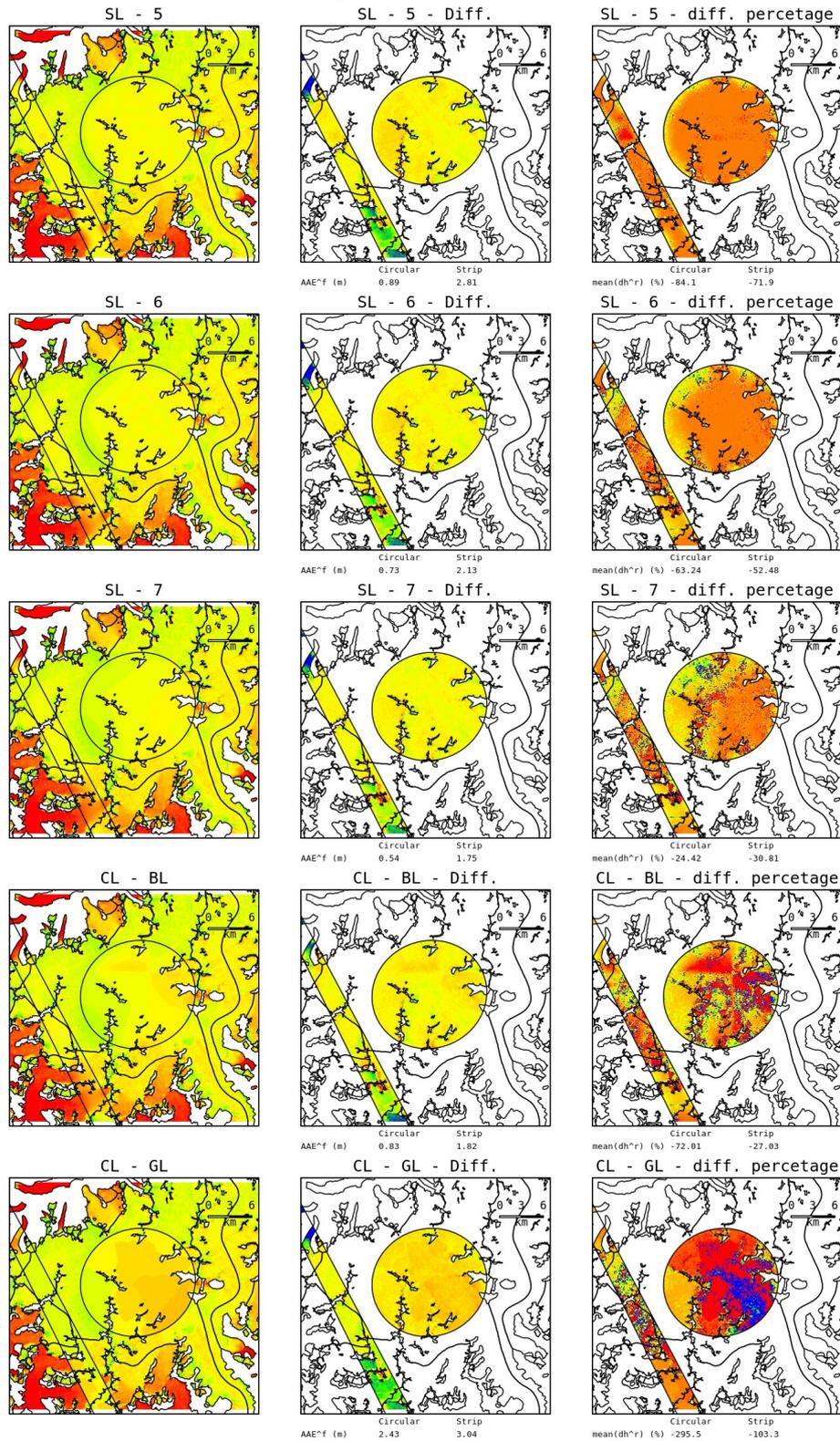
Continues on next page



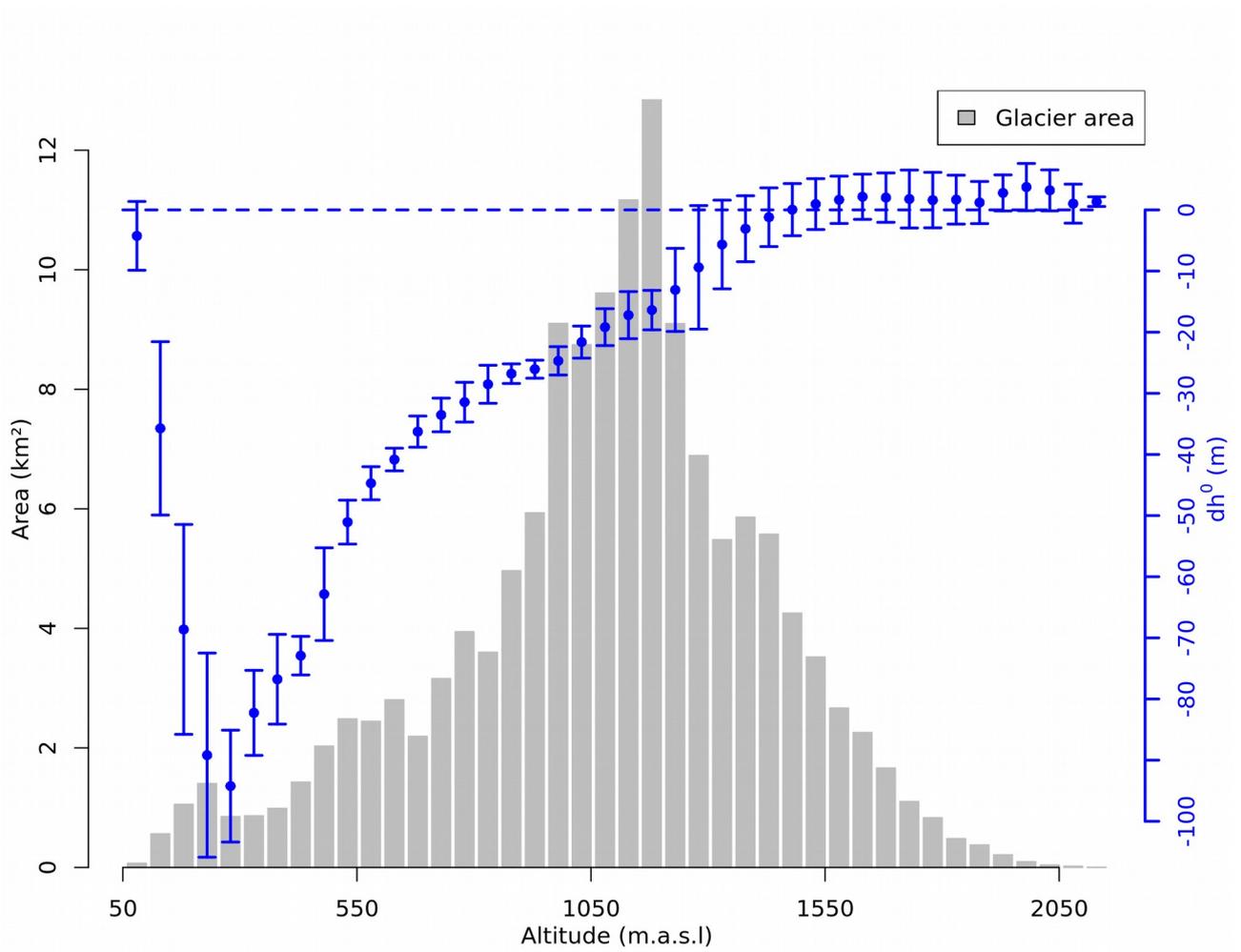
**Figure S1:** Void filling results  $dh^{IP}$  (left column), difference  $dh$  between void filling output and original data (center column) and relative offset  $dh^r$  for different void filling approaches at Center setup. Black polygons: glacier outlines; purple polygons: artificial voids.



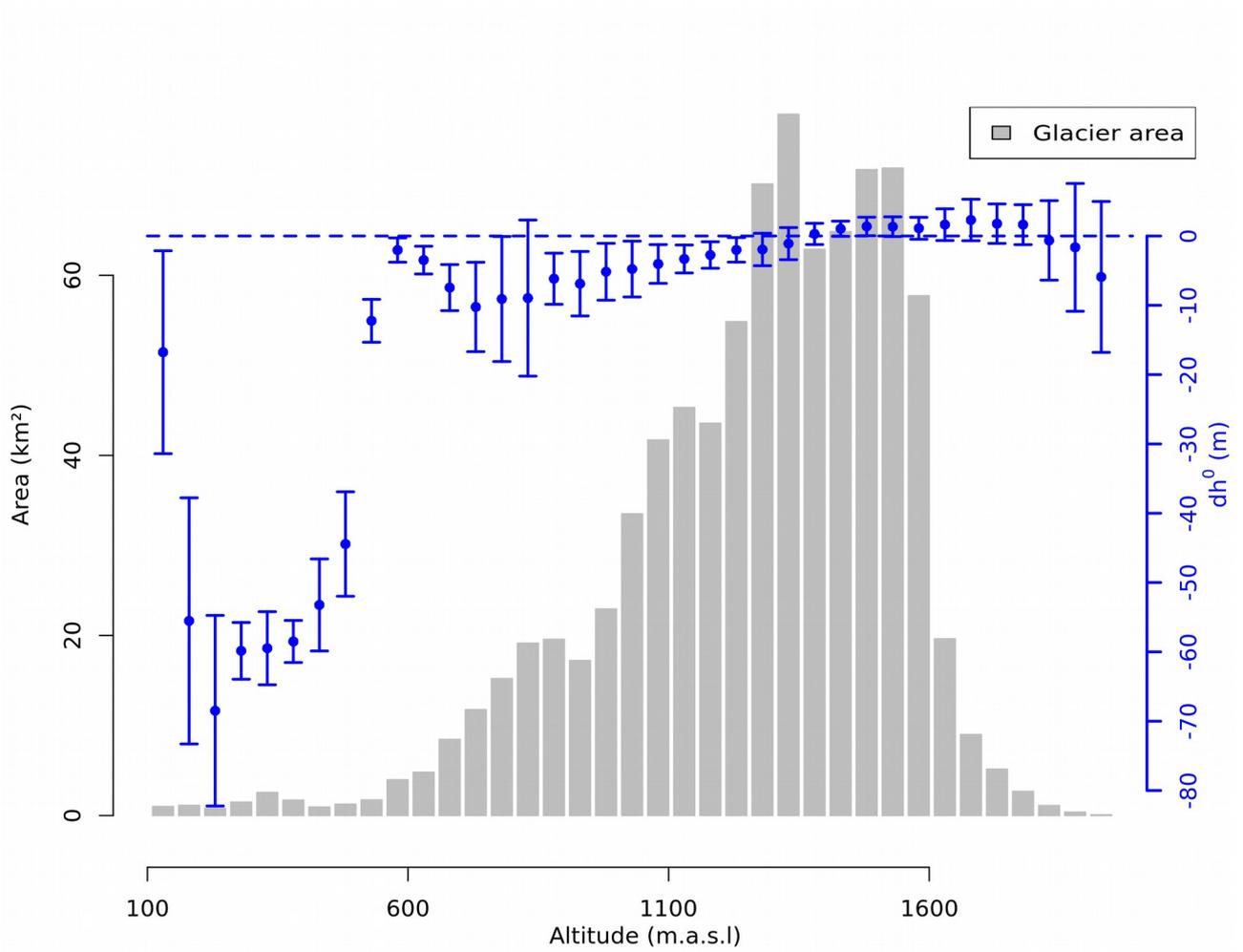
Continues on next page



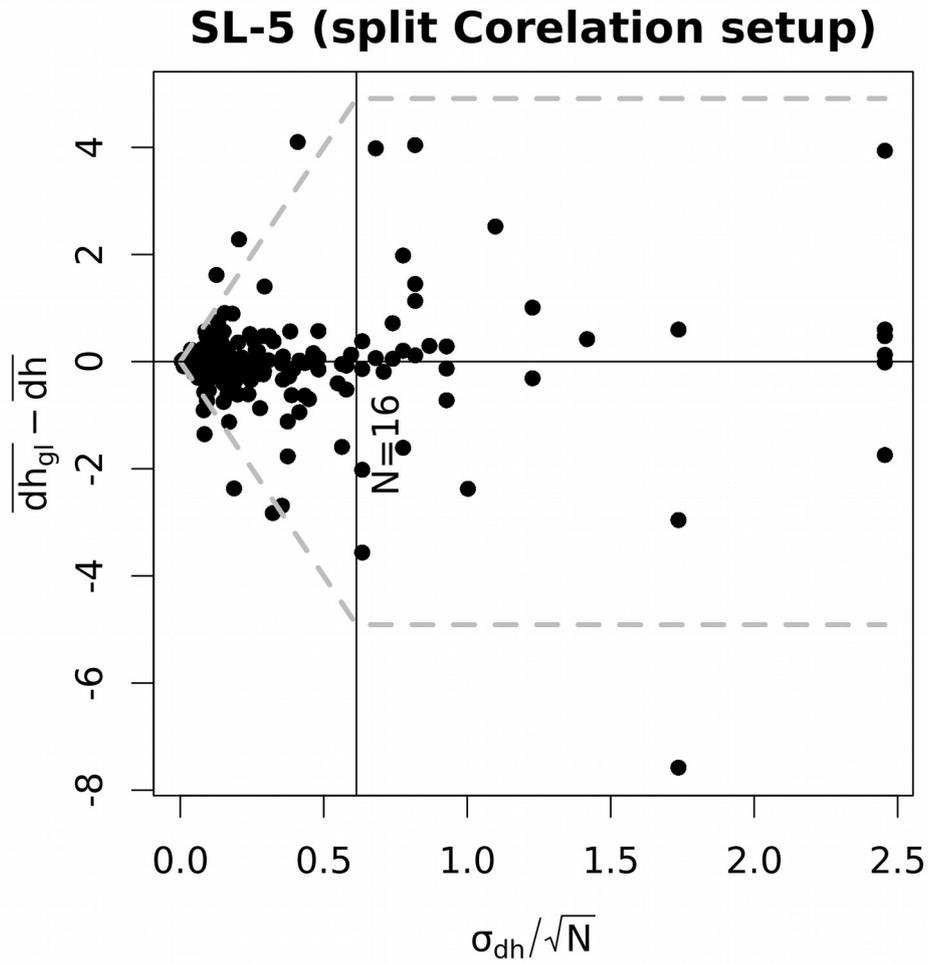
**Figure S2:** Void filling results  $dh^{IP}$  (left column), difference  $dh$  between void filling output and original data (center column) and relative offset  $dh^r$  for different void filling approaches at Juneau setup. Black polygons: glacier outlines; purple polygons: artificial voids.



**Figure S3:** Hypsometric distribution of glacier area and surface elevation changes of Center setup, grouped in 50 m altitude bins.



**Figure S4:** Hypsometric distribution of glacier area and surface elevation changes of Juneau setup, grouped in 50 m altitude bins.



**Figure S5.** Scatter plot of  $(\overline{dh_{gl}} - \overline{dh})$  versus  $\sigma_{dh}/\sqrt{N}$  for the Shearlet ( $nscales=5$ ) approach. Table 2. Gray dashed lines indicate the bounds from Equation (16) and (18) approximated by  $2d_{cor}\sigma_{dh}/\sqrt{N}$  for  $N > d_{cor}^2$  and by  $2\sigma_{dh}$  for  $N < d_{cor}^2$ .  $f^{int}$ : fraction of points within bounds.

## References

1. Kutyniok, G.; Lim, W.-Q.; Reisenhofer, R. ShearLab 3D: Faithful Digital Shearlet Transforms Based on Compactly Supported Shearlets. *ACM Trans. Math. Softw.* 2016, 42, doi:10.1145/2740960.