Glacier Image Velocimetry: an open-source toolbox for easy and rapid calculation of high-resolution glacier-velocity fields

Maximillian Van Wyk de Vries[1,2] and Andrew D. Wickert[1,2]

[1]Department of Earth & Environmental Sciences, University of Minnesota, Minneapolis, MN
[2]Saint Anthony Falls Laboratory, University of Minnesota, Minneapolis, MN

Correspondence: Maximillian Van Wyk de Vries (vanwy048@umn.edu)

Author Response

Contents of this pdf:

- Response to editor
- Response to reviewer 3

Response to editor:

Comments to the Author:

Dear authors,

Your manuscript was now reviewed by a new/external reviewer. This reviewer has several (some rather technical) comments, which I now invite you to address in detail (point-by-point rebuttal + manuscript update in accordance). This includes a comment related to the suitability of TC as a journal, which I also ask you to answer for the record. As previously indicated, as an editor, I understand a part of the critique by the reviewers, but I nevertheless also think that TC is a suitable option and therefore support your choice.

Many thanks for your efforts,

Kind regards,

Harry

Dear Dr. Harry Zekollari,

We thank you for the comments above and have responded to the reviewer's comments in detail below. In particular:

- We have responded to the reviewer's comments about the choice of TC as a journal, laying out in more detail our objectives with this manuscript and why we think they are of interest to TC's readership.
- We have run a new test on a synthetic image to highlight some of the characteristics of the new image pre-filter (NAOF) and explained our reasoning while creating this filter.
- We have repeated the experiments in the supplementary materials with a different method (orientation correlation) in response to a query about this.

The manuscript has also been updated accordingly, as shown in the tracked changes version. We also update the acknowledgements to include the third reviewer – comments by yourself and all three reviewers have helped improve the manuscript and code, and refine the objectives of this paper.

M. Van Wyk de Vries and Dr. A. D. Wickert

Response to reviewer (3):

We thank the reviewer for the detailed comments and suggestions, which we have answered between the lines below.

Overall, I think the this is a useful tool (i have not actually tested it). However, there is not much which is new beyond the tool itself, and it seems like it is misplaced in TC. I feel like this is an editorial decision though. Atleast it targets the correct audience.

In essence, we have three primary objectives with this paper:

1. Provide background on past advances and the current state of feature tracking in glaciology, and why these methods are valuable.

2. Present a new, open-source feature tracking toolbox, GIV, and explain in accessible terms the advances of this toolbox (handling large timeseries, new image pre-processing and velocity map post-processing, etc.), and its objectives (speed, flexibility, and ease of use on laptops and personal computers).

3. Examine the outputs of this toolbox through a variety of glacier types, both to confirm its pertinence in a range of scenarios and evaluate the type of glaciological problems which it may help solve.

Based on these objectives, we believe that this paper will both be relevant to the scope of The Cryosphere, and of interest to its readership. We agree that some of the computational methods described are not new- feature tracking has been in use in the earth sciences for several decades.

However, GIV does combine new methods of pre and post processing imagery, as well as convenient methods for handling large timeseries – with no coding required - in glaciology.

To provide one example where we think GIV can be particularly useful, it can be used to rapidly process a long timeseries of velocities following a localized natural disaster (e.g. landslide, glacier detachment/surge, etc.). Users with no prior experience with feature tracking may obtain a monthly velocity timeseries (using several hundred image pairs) in only a few hours of work, for a rapid evaluation of the hazard and prior conditions.

In equation 1 you write a sum over α. Can you please write it as a sum over i=1 to 4 and then write α_i inside the sum.

We have updated the equation and associated code to instead read:

$$I_f = \sum_{i=1}^{4} \cos\left[\arctan 2(I_o * \alpha_i, I_o * R[\alpha_i])\right]$$

I have some issues with the NAOF filter in equation 1. I appreciate the idea of extracting angles using multiple filtering kernels in order to get a more robust estimate of the local gradient. However, I miss an explanation or theoretical justification of why you believe this filter is good. Why is it constructed the way it is? I need some motivation for it.

The name of the NAOF suggests that this is similar to the filter used in orientation correlation (OC). But in the OC filter the output is a complex number which preserved the local gradient direction. Here you take the real part and any information about the gradient orientation is lost.

In equation 1 you convolute with four different α's. Each rotated by 45 degrees. But since each arctan2 operation uses both α, and R[α], 2 out of those 4 convolutions seem to me to be giving no additional information. Here's how I imagine the orientations of the different filtering kernels to be (based on your description). In order to help visualize it i define a matrix: a·α+b·R[α] - That gives me these four matrices.

matrix1 = [+a 0 -b;0 0 0;+b 0 -a]

matrix2 = [0 -b 0;+a 0 -a;0 +b 0]

matrix3 = [-b 0 -a;0 0 0;+a 0 +b]

matrix4 = [0 -a 0;-b 0 +b;0 +a 0]

So, you calculate the two gradients in the a and b-directions. Then you plug that into exp(i*arctan2(..)). That gives you a complex unit vector in that direction. Then you throw away the imaginary part and project it down on the x-axis. -But consider how the imaginary part of the filtering associated with matrix1 is almost identical to the real part of the matrix3 situation. This wastes multiple convolutions and slows the prefiltering down. Further, I do not understand the motivation for taking summing the x-component of 4 different unit vectors representing the intensity gradients in different coordinate systems. This seems very arbitrary. There is no explanation for why the NAOF prefilter is constructed in this way. This suggests to me that there is no theoretical justification or motivation, and that this new NAOF prefilter is an ad-hoc construction.

Note also: that real(exp(i*atan2(6,4))) is the same as cos(atan2(6,4))

Without a clear and convincing motivation for the form of the equation I find it difficult to understand why it is named "near anisotropic orientation filter". The filter is demontrated to be pretty good in practice in the selected examples. That is nice. -But is it an advancement over existing methods (CLAHE+H, or Orientation Correlation prefiltering)? I am not convinced that it is better in general.

Many thanks for these comments. We have made a number of changes to the text, and moved some details about the filter to the supplementary materials to not emphasize it more than required. We did not mean to present the NAOF as the 'perfect' pre-filter for use in feature tracking. We simply are presenting it as a viable alternative, which in some cases produces less noisy velocity maps than the other filters included in GIV. Its effectiveness has mostly been tested empirically. We also include other common filters such as CLAHE, highpass, Laplacian and Gaussian, such that users may test image filter combinations depending on the local glacier characteristics.

The objective for the NAOF was to:

1) Be insensitive to feature orientation.
2) Retain feature uniqueness.
3) Not shift the location of features within the image(s).
4) Remove contrasts between areas of differing pixel intensity (cloud cover, shadows, etc).
5) Retain the image as a real number matrix with only one band (two dimensional).

We hope that the improvements to the wording of this paragraph clarify our design and use of this filter- which was calibrated based on empirical tests. As the reviewer notes, the NAOF is based on the filter used in orientation correlation (OC). Rather than retaining the real and imaginary components, we retain only the real component and sum across four different filter angles.

The reasoning for editing the orientation filter used in OC is that despite producing good results, it was sensitive to feature orientation for many filter kernel choices. For a filter α = [1 0 -1] and the formula for calculating an orientation image given below:

$$I_{OC} = \exp\left(i \times \arctan 2(I_o * \alpha, I_o * R[\alpha]\right)$$

both the real and imaginary portions of the resulting image are more sensitive to certain feature orientations than others. On glaciers the features tend to be highly directional (crevasse fields), and this can result in a degradation of feature tracking results for portions of the glacier with surface features oriented in-line with the filter. In addition, certain undesirable features (e.g. medial moraine) can be emphasized over local crevasse fields, reducing the number of correct matches. Therefore, we decide to combine the four orientations, rather than using a single oriented filter. Certain filter kernels (e.g. [-1 1]) are not centered on the feature, and so do not preserve feature location.

Running some tests on example images showed that combining the four directions into one single filter (e.g. filter kernel [-1 -1 -1; -1 8 -1; -1 -1 -1]) provided adequate edge/feature enhancement, however resulted in a loss of feature uniqueness and a much greater number of false matches. Calculating the four components separately however allows each to be weighted differently according to the original pixel values, and better preserve feature uniqueness. Figure RR1 below shows the filter results for a synthetic image example, and figure RR2 shows the resulting filtered values plotted on a histogram. NAOF both detects all line orientations, and provides a broader distribution of resulting pixel values than the other single kernels.

Based on the comments here, we have rewritten NAOF to be more computationally efficient. We now pre-calculate each filtered image and 2-argument arctangent. Commuted 2-argument arctangent values are calculated as arctan2(b,a) = pi/4 - arctan2(a,b), reducing the number of operations. None of the convolutions are redundant, as each is sensitive to a different feature orientation. See figure RR3 in which a portion of the 2020/01/01 Sentinel-2 image of Glaciar Perito Moreno is shown under the different filter constituents and full NAOF.

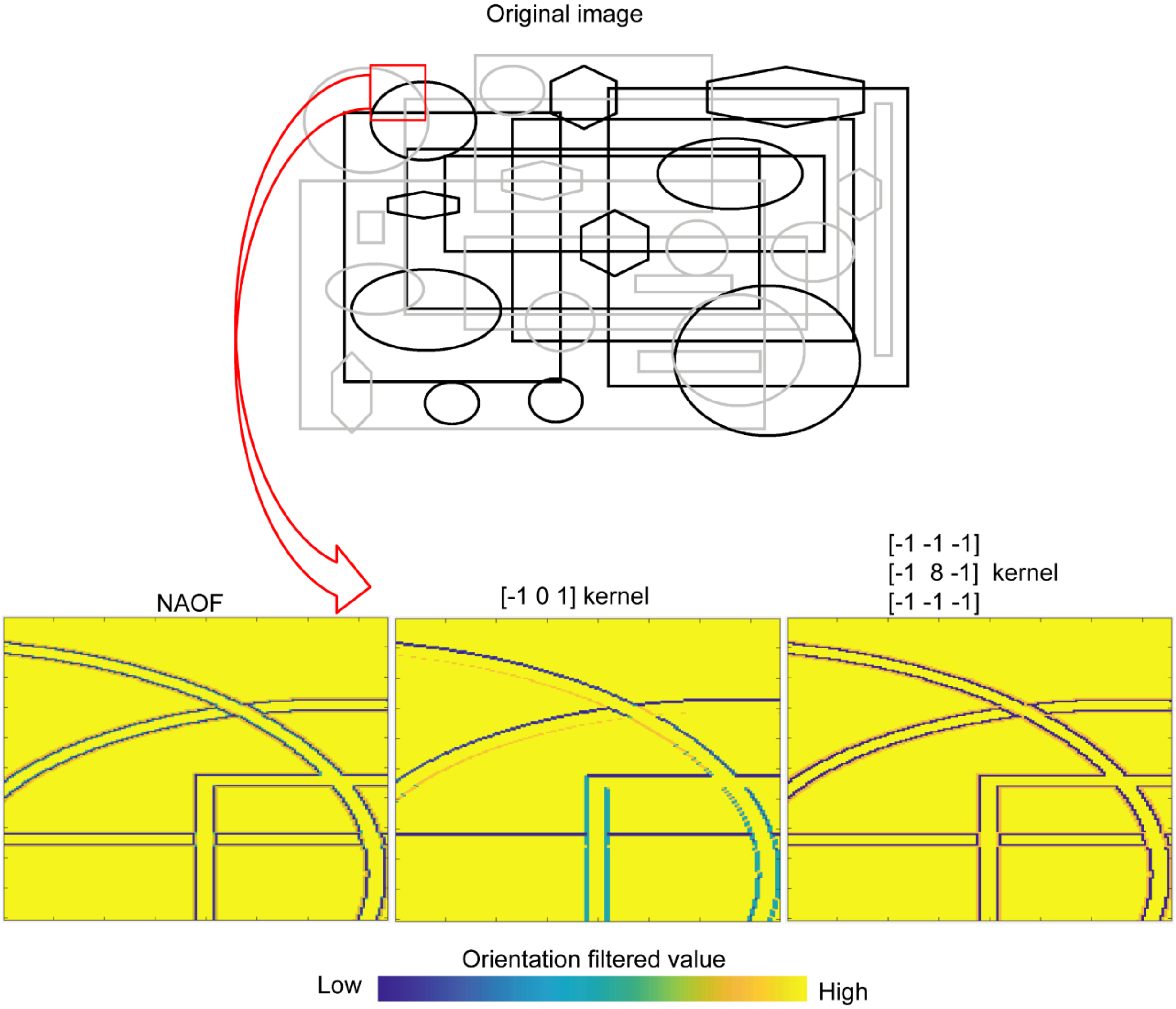

Figure RR1: Resulting filtered images for a small portion of the synthetic image above. Note how the [-1 0 1] kernel loses information on some horizontal features.

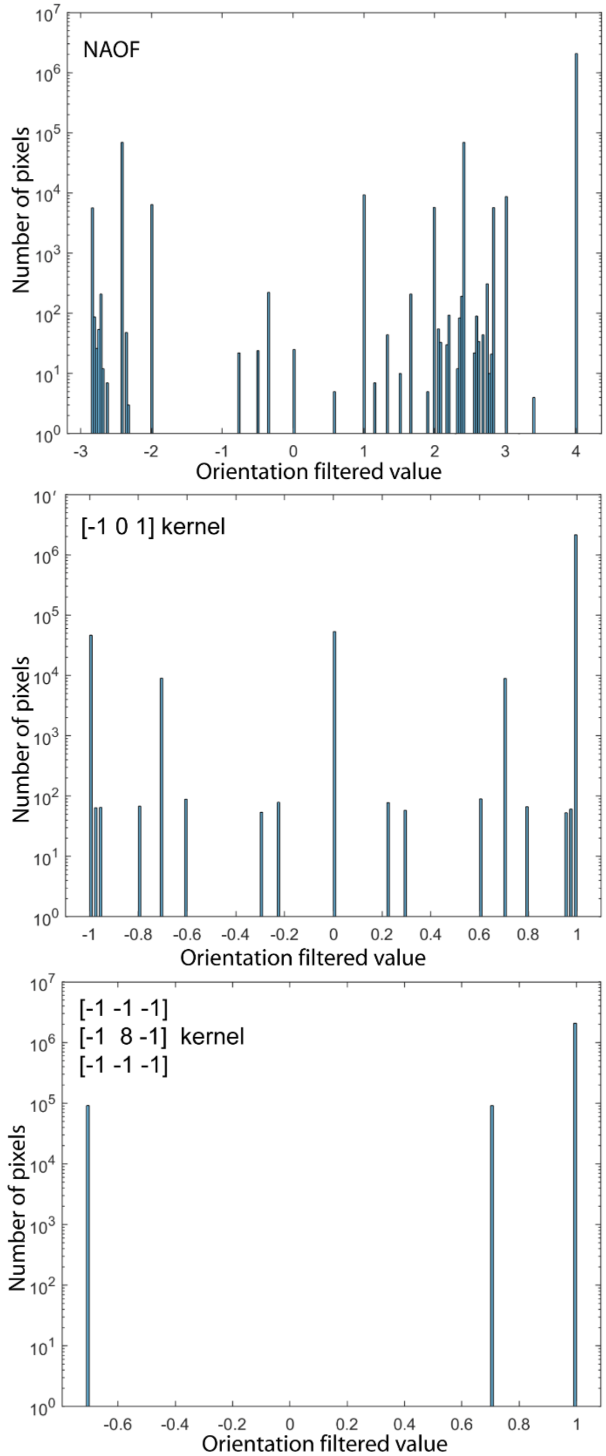

Figure RR2: Histograms of filtered pixel values for the three filtering options and original image shown in figure RR1. Note how NAOF produces the largest spread of unique values, whereas the [-1 -1 -1; -1 8 -1; -1 -1 -1] produces only a very limited range of values.

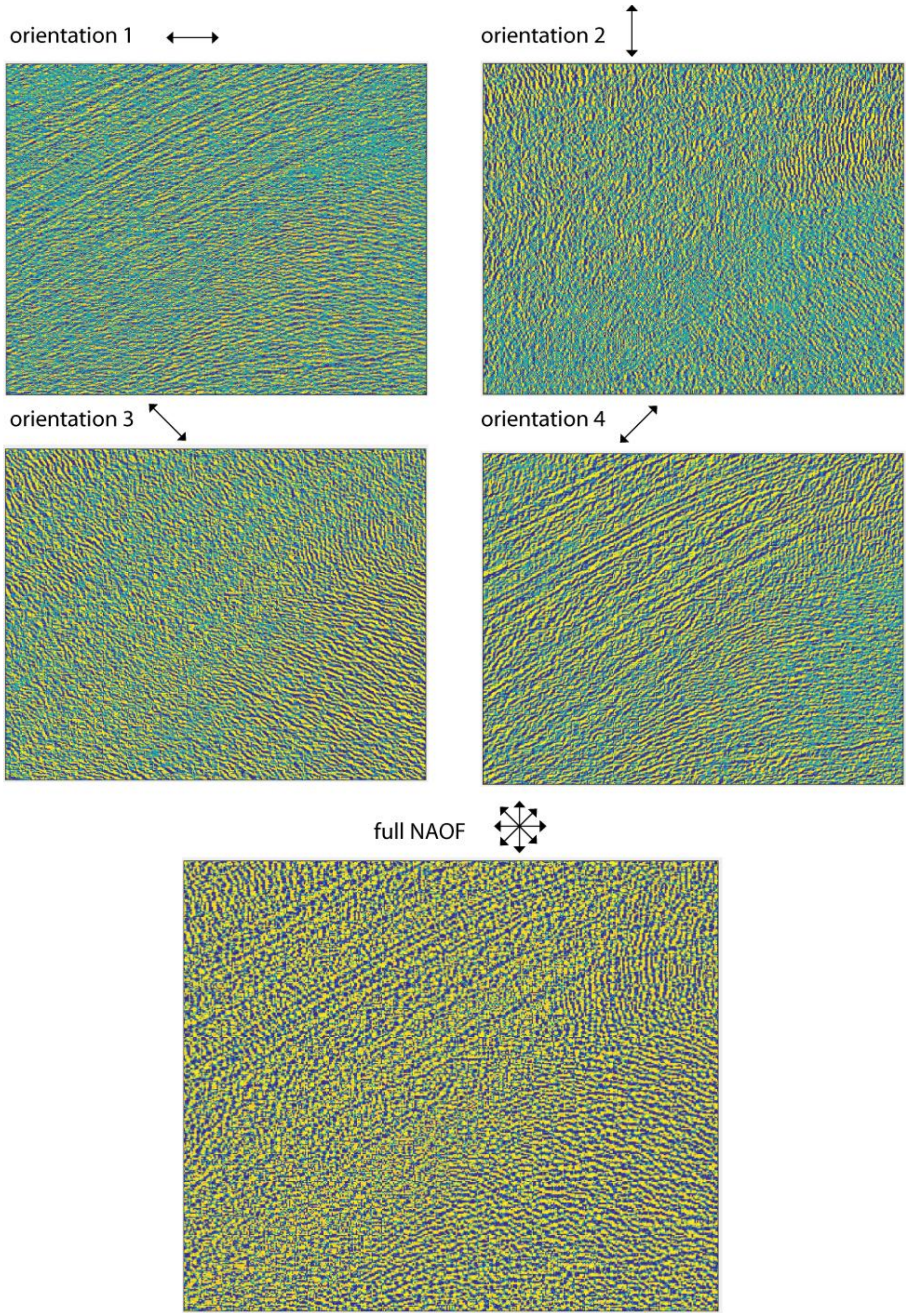

Figure RR3: 2020/01/01 Sentinel-2 image of Glaciar Perito Moreno shown under the different filter constituents and full NAOF, and full NAOF. Note how orientations 1 and 4 are particularly sensitive to the longitudinal features (medial moraine + flow bands), while orientation 3 clearly delineates the local crevasse field. Note also how crevasse uniqueness is preserved in the final summed filter.

Line116: It is fine to include a reasonably good new prefilter approach even if it is not theoretically justified. But I dont think it is a major step forward without more theory behind it, and/or a much larger test dataset. So, I recommed that you are conservative in your claims as to how good this new prefilter is. E.g. "NOAF shows comparable performance to CLAHE+H (see supplement)".

We have adjusted the description of the image filters in the text, and do not present NAOF as superior to other filters in all circumstances. Our evidence for the value of NAOF is mostly empirical. We do nevertheless note that it has produced the best velocity maps over a range of glaciers.

From the supplement the new NAOF performs similar to CLAHE, except CLAHE seems to almost fail when bitdepth=12. That to me seems weird. (Is it because CLAHE is extremely sensitive to noise in the least significant bits). Please explain why CLAHE+H is almost broken when bitdepth=12 if you can. If this is a real problem with CLAHE, then I think you should highlight it.

We have repeated the experiment as it appears that CLAHE was not correctly applied to the 12-bit image pairs. The supplementary materials have been updated to reflect this (12-bit and 8-bit CLAHE produce very similar results).

In the supplement you show some examples of NAOF, CLAHE, Raw, and combined with FCC and NCC and different bit depths. The issues with NCC+Raw are well known. ImGRAFT has Orientation Correlation as the default option, which presumably would work much better. This makes me ask why have you deliberately chosen to compare to NCC instead of Orientation Correlation.

We used NCC in response to some particular questions by the previous reviewer (anonymous reviewer 2). For completeness, we have also repeated the same experiments with CCF-O (Orientation correlation) option of IMGRAFT and the same parameter options as used for NCC. This produces improved results for the raw imagery (as expected based on prior findings), and similar results to NCC for CLAHE+HiPass and NAOF filtered results. Running NAOF filtered images with orientation correlation may ‘over-filter’ the image, as it effectively applies a double orientation filter. See figure RR4 below.

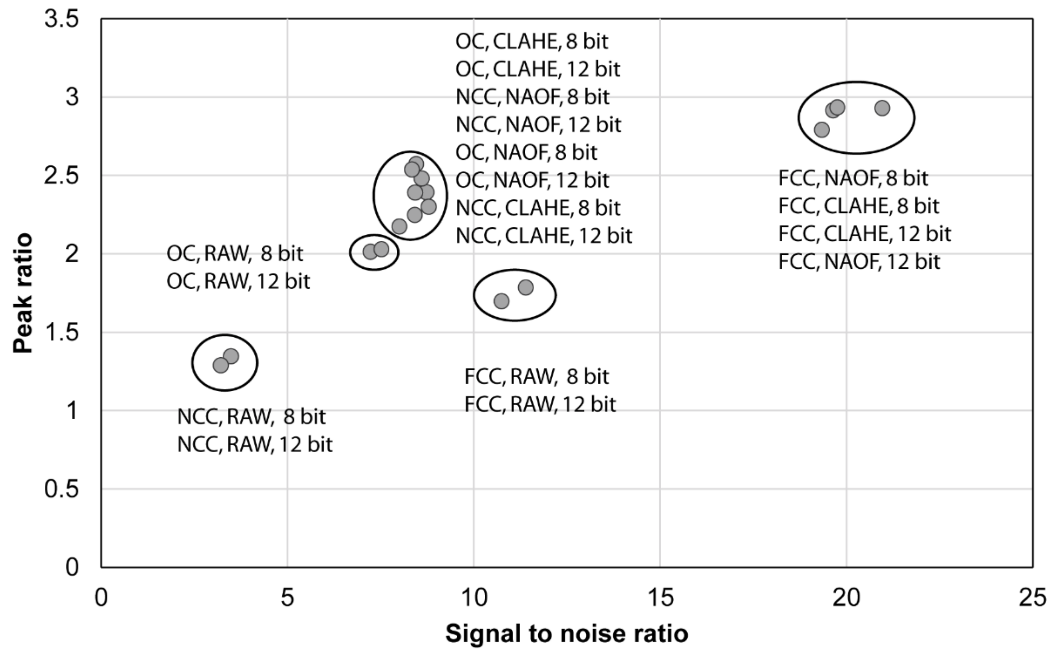

Figure RR4: Results of experiment 1 (see supplementary materials) updated to include orientation correlation results.

Line 108-110: You say this is better than a applying a single filtering kernel. This statement should be weakened if you do not provide data to demonstrates this. (That would be difficult because there are so many ways you could design the filtering kernel.) So you probably have to content with saying it is better than some specified filtering kernel.

We have adjusted the wording of this sentence to highlight which single filtering kernel we are referring to. Our objective was to state that the multi-orientation NAOF approach produces better results than the single filter summing the multiple oriented filters together (e.g. see figure RR1 and RR2).

Section 2.2: This section distinguishes FFT vs spatial domain methods. But in my view the FFT methods are really just a faster implentation of the convolutions.

We have made some adjustments to this section to clarify the key points. Spatial domain convolution and Fast Fourier Transform convolution can be equivalent (as per the convolution theorem), however they are used in different ways. Spatial convolution is used for matching one small region to surrounding small regions using a sliding-window approach, while frequency correlation is used to compute the match between two regions in one single calculation. As such, they do ultimately result in differing correlation surfaces- see Figure 4.9 from Altena (2018) for one example (NCC vs FCC correlation surface). Thielicke and Stamhuis (2015) also have a description of this in page 3 of their paper. We hope that there is less confusion in this section after the adjustments.

Line 347: "incorporates" is not a good word here, maybe "is able to exploit" would be better. I dont think you can claim credit for the new data availability.

We agree and have adjusted the wording to 'builds upon'. We do not claim credit for the availability of satellite imagery.

Line 162-174: It is much safer to do statistics and outlier rejection and interpolation steps based on the x and y component velocities rather than on |V| and flow direction. -line 172 makes me uncomfortable. Please clarify. Here's a little more detail of what i mean: "mean(sqrt(vx^2+vy^2))" will tend to be greater than "sqrt(mean(vx)^2+mean(vy)^2)" if there is any noise in vx and vy. The latter is more correct.

This is a very good point, and is key to obtaining good velocity maps in some cases (particularly with extremely slow moving glaciers). We do outlier rejection based on both x-y component data and speed/flow direction.

For instance:

-The stable ground correction is applied in the x and y components

-Average velocity maps are calculated based on averaging of x-y components "sqrt(mean(vx)^2+mean(vy)^2)" & "sqrt(median(vx)^2+median(vy)^2)" (some small adjustments made to the code to ensure that monthly values, etc are also calculated this way)

-A maximum speed filter is applied (from "sqrt(vx^2+vy^2)").

-Flow direction filters may also be applied.

Supplement

The SNR definition as written is dangerous to apply in practice. E.g. C_NCC can be both positive and negative values. therefore in some rare cases mean(C) can end up being negative. mean(C) also includes the peak which will bias the Sis not NR ratio as the "noise" now contains some signal.

Indeed, we had missed out the absolute value symbol in the equation. SNR is calculated as [peak/abs(mean(C))]. Note that the 3x3 pixel area surrounding the peak (used in subpixel peak finding) is cropped out of the correlation matrix prior to calculating the mean.

For that reason i think it is better to estimate the noise level using something like median(abs(C)). abs to deal with negative. Median is more robust to outliers (=peaks).

As mentioned above, the primary peak is cropped out of the correlation surface. Both mean(abs(C)) and median(abs(C)) have their advantages, as sensitivity to outliers can be desirable. Due to the variety of formulations, care must be taken when comparing SNR scores, particularly between different codes.

Figure S5: For the different methods (FCC/NCC) the C entering the SNR calculation has a different definition. Therefore you cannot make a straight comparison between SNR_NCC and SNR_FCC. E.g. Imagine i now define a new matching metric I called KFC. In my KFC method I simply calculate C_KFC = exp(C_NCC). It will have exactly the same peaks just mapped through a nonlinear transformation. However, the SNR will be completely different.

This is true; however KFC is not really a different matching metric and this example presents a worst case scenario. The signal to noise ration and peak ratio in FCC and NCC do result from different calculations, but these are not nonlinear transformations. In each case the signal to noise ration and peak ratio represent the same property ('how confident are we that this peak is a real match, rather than an artifact'). Using ratios (as opposed to raw correlation scores) and both the peak and signal to noise ratio reduces the dependence on the matching method.

Other comparison metrics exist, such as the 'percentage of correct matches' used by Heid and Kaab (2012). This is very sensitive to the methods used to evaluate correct matches, particularly where no external velocity results are available. Calculation of velocities over stable ground can also be used, but is potentially sensitive to the matching method and does not always represent the quality of matches over the glacier.

In an ideal situation, we would evaluate the 'correctness' of each matching technique with reference to external, ground-based glacier velocity measurements. This comparison should ideally include other feature-tracking codes (e.g. AutoRIFT, CARST, IMGRAFT, etc.), and is beyond the scope of this paper.

In summary, SNR and PKR are great metrics for comparing different options (e.g. image filters) across one feature tracking method. SNR and PKR are not perfect for comparing between feature tracking methods, but (particularly in combination with each other) do provide useful information about the relative quality of matches. We apply them with caution here, and have added a brief note to the limitations paragraph.

Figures: The number of figure are a bit excessive to me. E.g. fig 10 looks cool, but what does it bring to the manuscript. Fig11 seems unnecessary too.

Figure 10 shows the velocity results over a small, slow moving glacier system (which is typically challenging to resolve via feature tracking). Figure 11 shows GIV's graphical user interface, through which all calculations can be made. We do think these figures are useful, complement the manuscript text, and will be particularly relevant to The Cryosphere's glaciological audience.

References:

Heid & Kaab, 2012. 10.1016/j.rse.2011.11.024

Thielicke & Stamhuis 2014. 10.5334/jors.bl

Altena 2018. https://www.duo.uio.no/handle/10852/61747