The Cryosphere Discuss.,
https://doi.org/10.5194/tc-2020-204-RC2, 2020

Interactive comment on "Glacier Image Velocimetry: an open-source toolbox for easy and rapid calculation of high-resolution glacier-velocity fields" *by* Maximillian Van Wyk de Vries and Andrew D. Wickert

Anonymous Referee #2

Received and published: 1 October 2020

The toolbox presented by the authors is a GUI to ease feature tracking routines within a MATLAB environment. The naming of the toolbox is a reference to Particle Image Velocimetry (PIV), which is a well established technique within fluid mechanics. It is based upon particle seeding within a fluid, illuminated by light behind a narrow slit to get a within-plane velocity profile, thus in a well-controlled environment. When particles are not well stirred, have a lot of out-of-plane displacements, too large time separation, or other effects complicating the tracking: the experiment is done all over. The authors lend the methodologies from this domain, and implement this on natural environment

to extract glacier surface velocity. This transfer is more complicated as it might initially seem to be, as seeding is absent (though people in hydrology are experimenting on this issue [Pizarro et al. 2020]), and more importantly experiments can't be redone.

The toolbox has some application specific adjustments, which is in line with other application domains, where similar toolbox are introduced, such as: Optical tracking velocimetry (OTV) [Tauro et al 2018], Surface Structure Image Velocimetry (SSIV) [Leitaoa et al. 2018], Kanade–Lucas Tomasi Image Velocimetry (KLT-IV) [Perks 2020], Part2Track [Janke et al. 2020], TecPIV [Boutelier 2016], Rectification of Image Velocity Results (RIVeR) [Patalano et al. 2017], Waves Acquisition Stereo System (WASS) [Bergamasco et al. 2017], or more specific to glaciers: Environmental Motion Tracking (EMT) [Schwalbe et al. 2017], PyTrx [How et al. 2020], Image GeoRectification and feature tracking toolbox (IMGRAFT) [Messerli et al. 2015], Pointcatcher [James et al. 2016]. Given the list above, and the journals where these toolboxes are presented, it might be a valid question why the authors have opted for a submission to The Cryosphere, and not a technical EGU journal like Geoscientific Instruments or Geoscientific Model Development?

There are some implementations which one could expect for a toolbox tailored towards glaciology to be presented, but are not included. In addition some features are included, which need further assessment, to highlight their improvement, if they do.

To be more specific the comments are grouped into different paragraphs: Analysis of the 10-bit data of Landsat 8 [Jeong et al. 2015] and Sentinel-2 [Kääb et al. 2016] over dark overshadowed terrain have shown there is a significant benefit over 8bit data. Why do the authors downgrade their imagery to 8-bit RGB? In the same line, the use of non-georeferenced ".jpg" or ".png" data is strange. MATLAB supports mapping tools for such issues, why are these not adopted. While the toolbox needs a specific dimension in order to work, it is very strange this is not present in the toolbox. The choice of using angles (lat,lon) for metric data is confusing.

The authors present a new image transform dubbed "NAOF", but is there a significant improvement, over for example typical orientation filtering approach? At least to me, the 45 degrees filters seem like a redundant feature, as 90deg angled steerable filters incorporate all information [Freeman et al. 1991]. It seems the implementation is a combination of work similar to [Ahn et al. 2011], and orientation filtering [Heid et al. 2012]. But it is questionable, if the additional calculations add towards improvement. The filter banks are correlated (as just mentioned), furthermore, the visible bands are similiarly correlated over glaciers. Hence, the information gain might be very limited.

The paragraph on velocity calculations talks about two methods to extract displacements from an image pair. However, this distinction is more about the option if refinement is applied or not. The authors limit themselves to frequency domain methods, which is the de-facto procedure for PIV. However, spatial domain methods can be more favorable for glacier related applications. This comes down to the point given above, of the transfer from fluid mechanics towards an application domain without the ability to redo an experiment. Hence, the velocity profile can be highly variable (extensive shear, valley walls, diverging flow, fast flow, ...). Spatial domain methods are more resistant to shear flow, the chip is not related to the search space (as with fourier methods). The authors might have considered spatial domain methods, but for clarity it might be fruitfull to mention this, and why. In PIV experiments, one is able to adjust the image sampling rate, to be in accordance with the maximum flow velocity, this is less the case for glaciers.

Technical comments: In general the code is very messy, many duplicate lines exist, commenting is absent ("strcmpi(inputs{51,2}, 'Yes')"). At multiple locations indents are used in-appropriately. Documentation and commenting within the code is limited. Every function has an extensive help section, which is typically an acknowledgment and info about GIV. Input or output variables are not described. For many or all functions, tests are lacking. Factorization is on a low level. Given this situation of the code, how do the authors see adoption by others? The objective of the authors is to be a wrap-

per (based upon ImGraft and matPIV), then one would expect documentation to be extensive.

Sub-pixel: There does not seem to be a sub-pixel estimation present? Why is this?

The domain of PIV is also extensively populated with papers about peak-locking, it would be good if the authors put some effort in this as well. This bias is especially present in sub-pixel estimation directly done on the correlation surface. Other methods, such as TPSS, as implemented in Cosi-Corr are less sensitive to this effect (which to a large extent might explain its popularity in geodetic imaging).

Geo-referencing: The gridspacing as it is implemented now assumes all rectangular and upright pixels, please adjust this.

Matlab: To my knowledge Matlab is not open access, the code might be open, but many of the algorithms are hidden away and licensing is needed. Hence, I am not so sure if the description about FFTW is needed, as one is not able to access this. Secondly, it is a standard routine within MATLAB.

Merging satellite data: The authors describe a time-series construction method, illustrate a toy example. But it seems they apply a sigma-filter, and do simple infilling. There are more advanced methods available, which are more adaptive towards velocity data. Hence, a simple reference to (e.g.: [Mouginot et al. 2017]) might suffice for the implementation here, and this section can thus be reduced considerably.

The temporal sampling is a bit counter intuitive, as this is an opposite direction to the workflow of [Millan et al. 2019]. Where they found the time-span needs to be of sufficient size, in order to be of most use. Why then do the authors prefer the shorter time steps?

Code specific: There is a function about intensity capping, while this might be of interest to PIV, it is questionable if this is of interest to environmental signals. Bright intensity of seed points within dark water are very much different from glacier surfaces. Also how does this merge with the high-pass filtering... ?

Even though your algorithms are optimized, they might not pull out the best of the machinery, yet. For example, the function "neighbourfilter" contains a double loop, to process a kernel. An internal function of Matlab called "nlfilter" might be much faster. A good reference for such implementations can be found in "Accelerating MATLAB Performance" (http://undocumentedmatlab.com/books/matlab-performance). In the same function from line 95 onward, the authors use double indexing. While linear indexing is possible as well, which make vectorization possible (which is MATLABs strong point). This greatly improves the processing time as well.

I am not sure if the variance calculation of the flow direction is calculated correctly. Offsetting the direction and applying a weighting might work, but maybe correct circular statistics might be more appropriate, see [Berens 2009] for a toolbox implementation. It might be more easy to use the mapping coordinates instead...?

It might be good to give an example; here is a sniplet of your code (a straight copy):

```
if smoothsize == 2;
mask = [0 1 0; 1 0 1; 0 1 0 ];
elseif smoothsize == 3;
mask = [0 1 0;1 1 1; 1 0 1; 1 1 1; 0 1 0 ];
elseif smoothsize == 4;
mask = [0 0 1 0 0 ;0 1 1 1 0 ;0 1 1 1 0; 1 1 1 1 1 ; 0 1 1 1 0; 0 1 1 1 0; 0 0 1 0 0 ];
elseif smoothsize == 5;
mask = [0 0 1 0 0 ;0 1 1 1 0 ;0 1 1 1 0;1 1 1 1 1 ;1 1 1 1 1 ; 1 1 1 1 1 ; 0 1 1 1 0; 0 1 1 1 0; 0 0 1 0 0 ];
end
```

why isn’t this in a function? Something like:

```
function [mask] = make_mask(smoothsize)
"
generates a neighborhood kernel in the form of a diamond,
though excludes the central pixel
input:
smoothsize - integer value
output:
mask - logical array
"
if nargin<1, smoothsize = 3; end
mask_radius = floor(smoothsize/2);
mask = strel(’diamond’, mask_radius); % make a diamond shape
mask = mask.Neighborhood;
mask(mask_radius+1, mask_radius+1) = 0; % exclude the central element
end
```

In all I am aware this is, like all research, work in progress, and I think this is a very useful direction. But in its current state and form, sufficient work needs to be done to be of interest, please see [Perks 2020] for a nice example. Here workflows are nicely presented, a dashboard is present, etc. All in all, I think a transfer towards a technical journal of EGU might be more in place.

Textual: p63: "broken down" is misleading, as overlapping chips can also be used p140: superscript the 2, to make it a square

references:

Ahn et al. 2011 10.1109/TGRS.2011.2114891

Berens 2009 10.18637/jss.v031.i10

Bergamasco et al. 2017 10.1016/j.cageo.2017.07.001

Boutelier 2016 10.1016/j.cageo.2016.02.002

Freeman et al. 1991 10.1109/34.93808

James et al. 2016 10.1017/jog.2016.27

Janke et al. 2020 10.1016/j.softx.2020.100413

Jeong et al. 2015 10.1016/j.rse.2015.08.023

Heid et al. 2012 10.1016/j.rse.2011.11.024

How et al. 2020 10.3389/feart.2020.00021

Kääb et al. 2016 10.3390/rs8070598

Leitaoa et al. 2018 10.1016/j.jhydrol.2018.09.001

Messerli et al. 2015 10.5194/gi-4-23-2015

Millan et al. 2019 10.3390/rs11212498

Mouginot et al. 10.3390/rs9040364

Tauro et al. 2018 10.3390/rs10122010

Patalano et al. 2017 10.1016/j.cageo.2017.07.009

Perks 2020 10.5194/gmd-2020-187

Pizarro et al. 2020 10.5194/hess-2020-188

Schwalbe et al. 2017 10.5194/esurf-5-861-2017

Interactive comment on The Cryosphere Discuss., https://doi.org/10.5194/tc-2020-204, 2020.