GLACIER IMAGE VELOCIMETRY

(GIV)

USER MANUAL

WHO IS THIS DOCUMENT FOR?

Do you want to calculate the flow speed of a glacier from satellite imagery? If yes, this document is for you.

GIV is intended to be easy to use, fully open source and free of any licensing burdens. It can be run through MATLAB, but can also be run without a MATLAB license as a standalone app. Running GIV requires no coding, and this document provides a step by step guide for getting started - and how to do more complex work.

See the associated Van Wyk de Vries and Wickert 2020 paper in The Cryosphere for more examples, details and discussions.

CONTENTS

1 WHAT IS GIV? **4**

2 BACKGROUND AND THEORY **6**

3 GETTING STARTED: STEP BY STEP GUIDE (WITH IMAGES) **14**

3.1 OBTAINING SATELLITE IMAGE 14

3.2 RUNNING GIV 15

3.3 GIV OUTPUTS 28

4 INPUT PARAMETERS **32**

5 ADVANCED: MATLAB FUNCTIONS AND CODE **50**

6 EXAMPLE: GLACIAR PERITO MORENO **55**

7 FURTHER REMARKS AND CREDITS **58**

1 What is GIV?

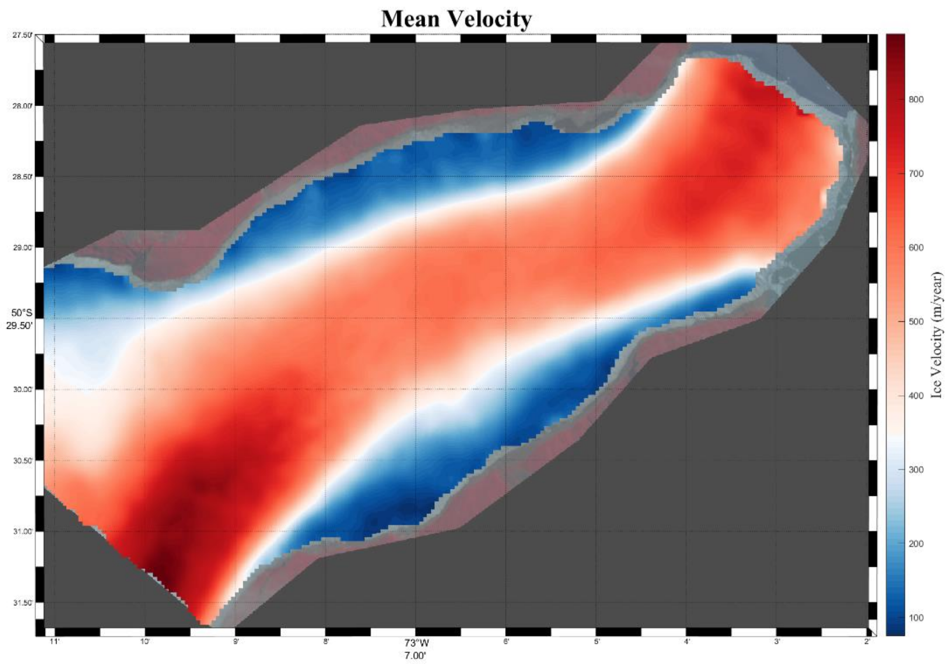
GIV is a toolbox that enables the surface velocities of glaciers to be calculated from satellite images. Glacier surface velocities may be desirable for a number of different purposes, including remote sensing analyses, glacier mass balance studies, assistance for field work, hazard assesments and many more. Measuring glacier velocities in the field is tedious and, in many cases, logistically impossible, thus calculating the velocity of glaciers from afar is useful.

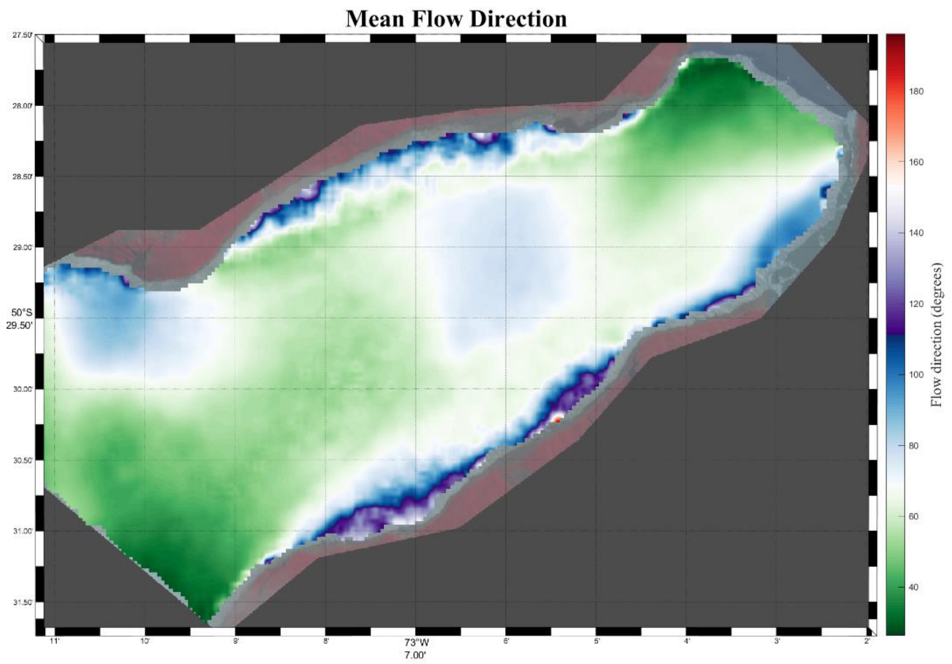
GIV enables rapid calculation of velocity fields for any glacier on Earth: within a few hours you can have usable results to guide field decisions or future research ideas. GIV also enables easy handling of large datasets, and can conveniently load, filter and process datasets containing thousands of image pairs. Finally, GIV saves the results in a number of convenient formats, including georeferenced velocity maps and publication ready figures.

GIV also includes a number of functions that help speed up analyses, including:

- Various types of automatic image filtering (e.g. edge enhancing, contrast reducing, orientations filtering, etc.)
- Automatic filtering of velocity maps to detect and remove outliers
- Automatic generation of monthly velocity timerseries and other timeseries plots.

The two images on the following page represent a mean velocity and mean flow direction map for Glaciar Perito Moreno in Argentina for the first 3 months of 2020. Section 6 of this user guide will run this example, and the following sections describe the basic theory of image tracking and the contents of the GIV toolbox.

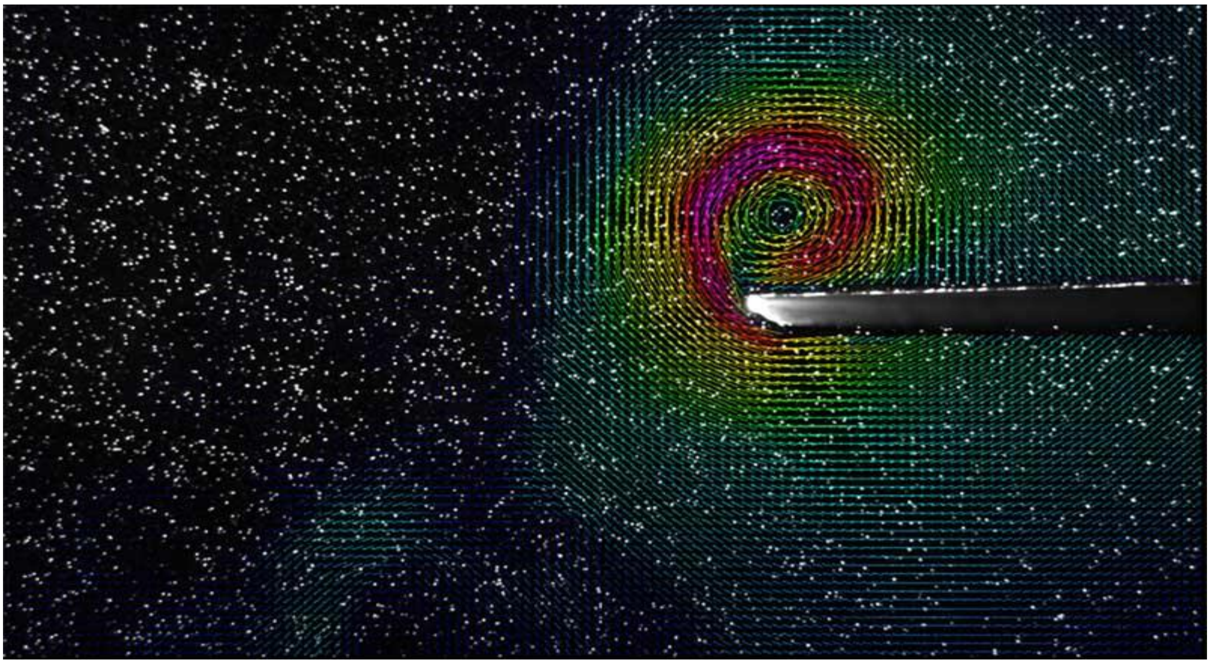




2 BACKGROUND AND THEORY

If you are just reading this user manual to generate some quick velocity maps, feel free to skip this section and move on to the step by step guide in section 3.

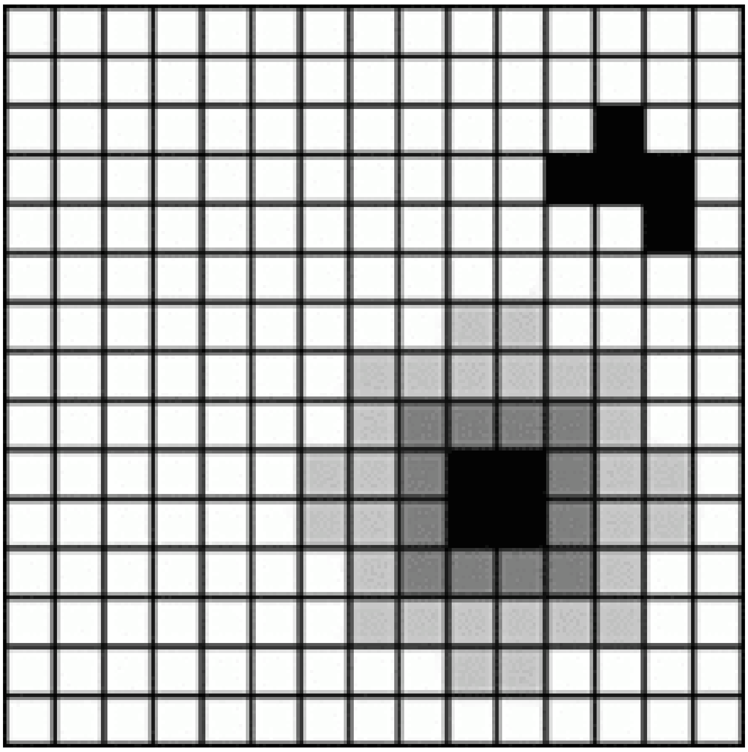
Particle tracking is not a new technique, and has been used for several decades in fluid dynamics experiments to track small tracers introduced into fluids. It has its roots in the search for a flexible tool that enables calculation of 2D, or even 3D velocity fields within fluids. Most early uses involved the introduction of carefully chosen tracers into a fluid under controlled lighting and imaging with a high speed camera. This showed a marked improvement on prior techniques such as Laser Doppler Velocimetry and anemometers which only measure velocity at a single point, however was computationally expensive (for the 1980s). Rapid improvements to computing power and the flexibility of the technique have



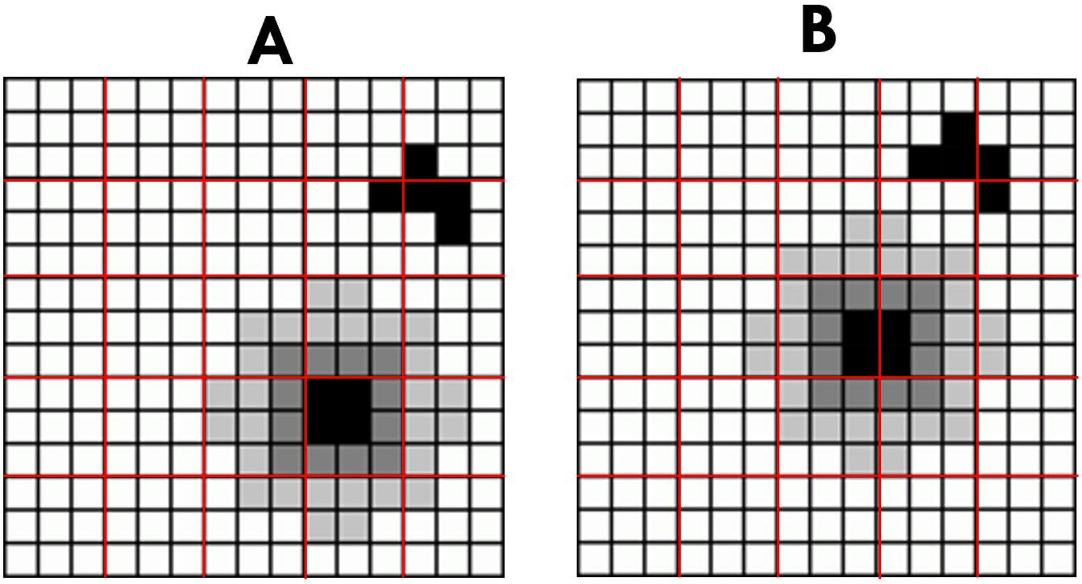
made it the particle velocity analysis technique of choice in a number of fields. The book '*Particle Image Velocimetry'* by Adrian, Adrian and Westerweel (2011) provides a good background to this technique, although is more aimed at engineers than earth scientists.

Traditional Particle Image Velocimetry has some uses in the earth sciences, including tracking volcanic bombs during eruptions (Gaudin et al., 2014, 2016) and tracking river flow (e.g. Mutse et al., 2008). Scambos et al (1992) were the first to adapt these techniques for use on glaciers, investigating the flow of an ice stream in Antarctica using successive Landsat datasets. Since this time the number of satellite imagery sources has greatly increased, and a number of new tools have been released. GIV is the most recent of these, and aims to be rapid, flexible and easy to use. See our Van Wyk de Vries and Wickert (2020) The Cryosphere paper for more details on this and other existing toolboxes,

The basic idea of particle image velocimetry, or feature tracking as it is often called in glaciology, is to trace the movement of features from one image to the next. If the images are accurately georeferenced, their resolution is known and the timing between images is known, the movement of this feature may be converted into a velocity. Consider the following image:

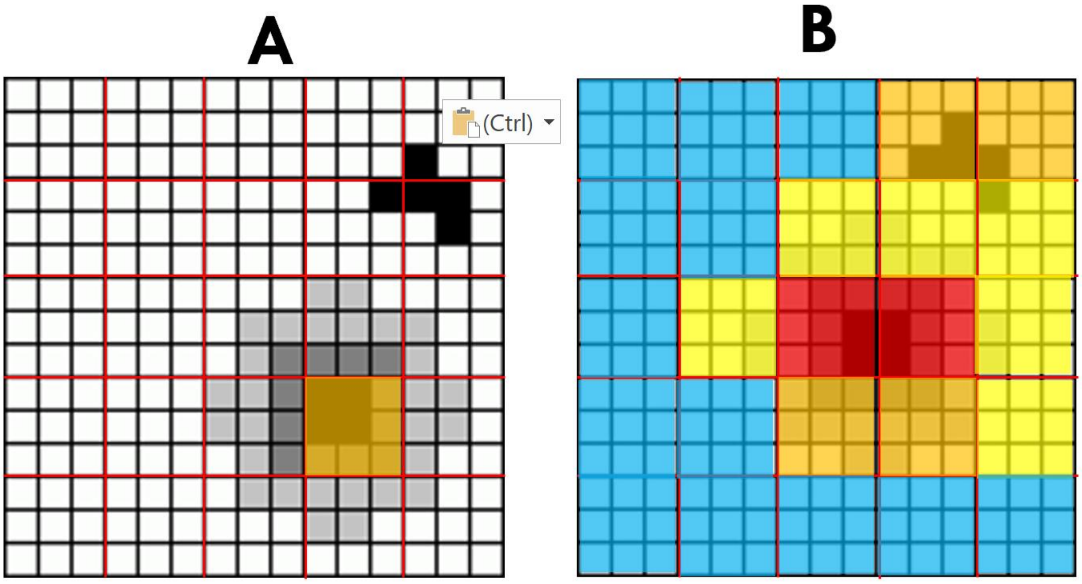


Two ‘features’ are visible in this image, a shape in the top right and a concentric feature in the bottom right. The movement of these features through the grid can be tracked. Consider now the two following images, one (Image A) at time T1 and the second (Image B) at time T2:



A grid has been overlain over the image for better visualisation, see how the features have moved from image A to image B. A good preliminary test of whether a series of images is suitable for feature tracking is whether the movement of features can be seen with the naked eye. This doesn’t always apply, but typically the best results will come from images with clear features, and our eyes are surprisingly good at picking these out.

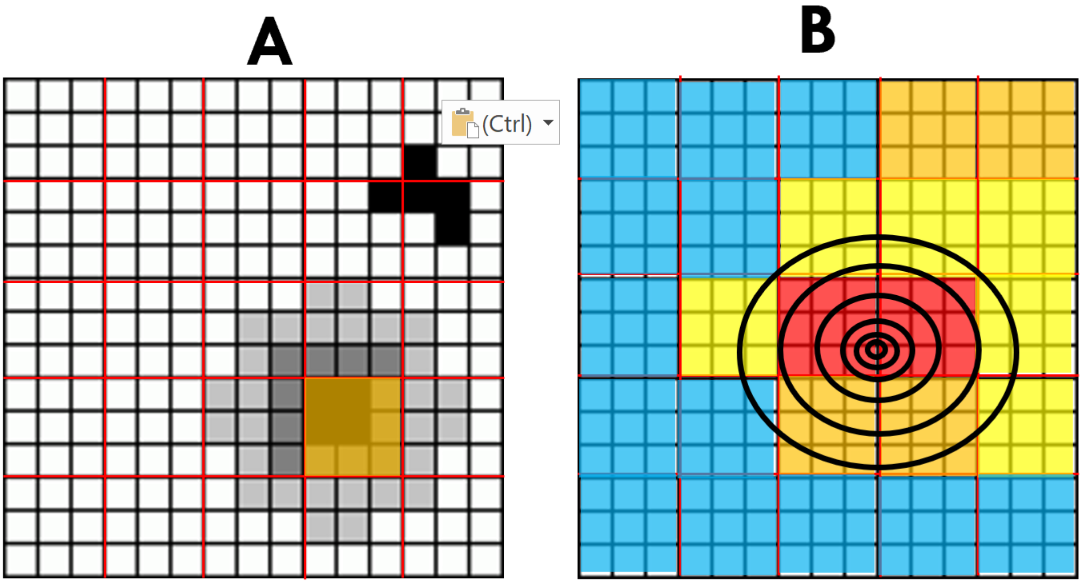
As a next step, each ‘chip’ in image A, defined by the red grid shown above, is compared to the surrounding grids in the associated image B:



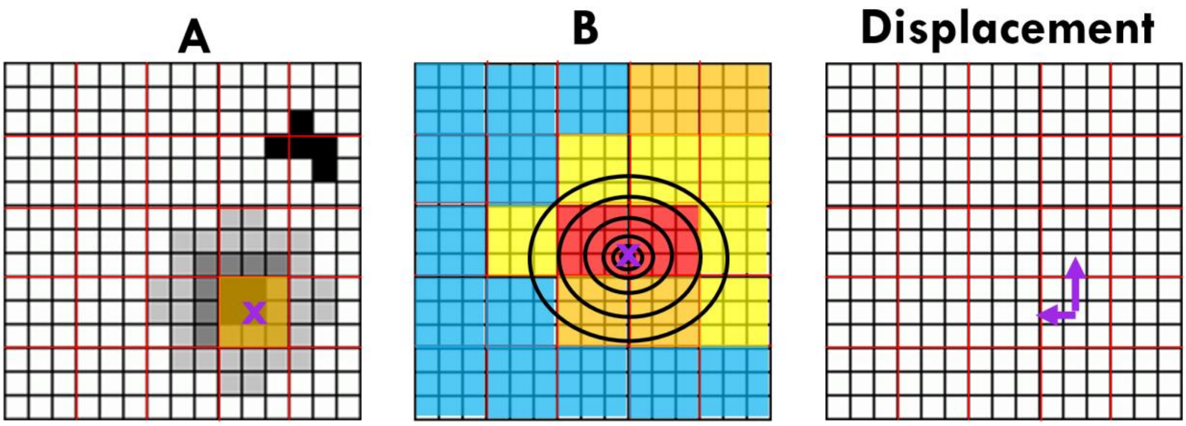
In the example above, the one orange chip from A is compared to all chips in image B. For each chip in the second image (B), a correlation coefficient is calculated with the original chip from A describing how similar the two chips are. A good match will generate a high correlation coefficient, whereas a low match will generate a low correlation coefficient. In the image above high correlation is shown in red colors, while low correlation is shown in blue. Note how correlation is highest in the area the original feature has moved to.

In GIV this correlation is performed in the frequency domain. Both the chip from the original image A and the search region in image B are Fourier transformed (using a Fast Fourier transform algorithm), compared, and then converted back. This additional step improves the speed of the matching algorithm.

A final step of the matching involves fitting a peak, typically a Gaussian curve, to the correlation values surrounding the peak:

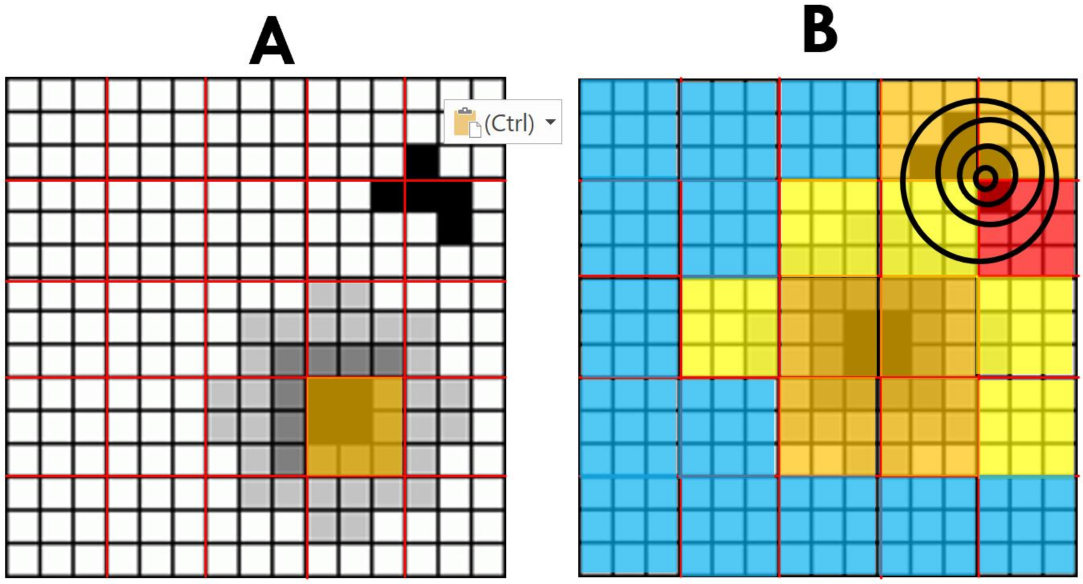


This final step allows the location of the peak to be evaluated to sub-pixel accuracy (i.e. we are not limited by the size of the grid we chose). Putting this all together, we can compute a displacement, typically broken down into a x and y component:

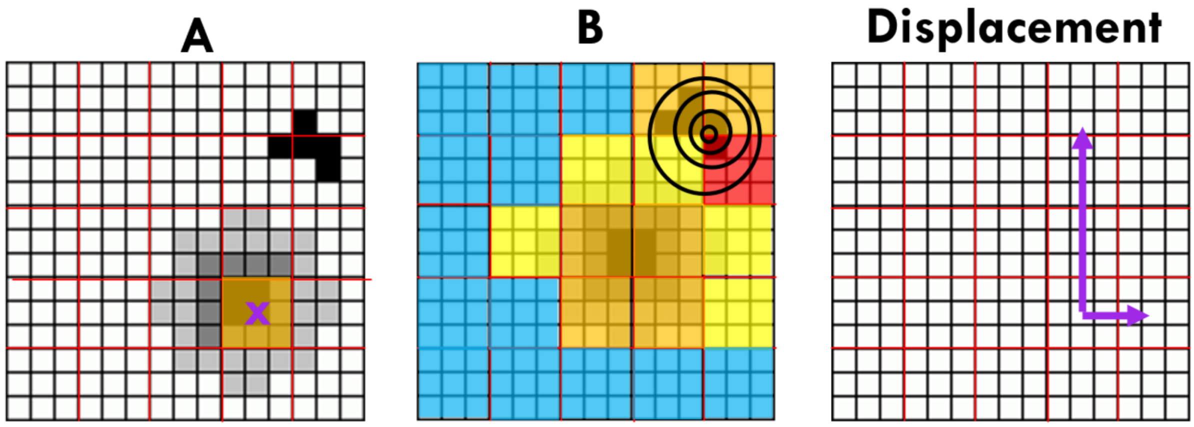


Here the purple x in image A shows the original location of the grid cell, the purple x in image B shows the estimated location of this grid cell following displacement. The purple arrows in the third image show the estimated x and y components of this displacement. Once corrected for the spatial resolution of the grid and time between images A and B, this will provide the velocity of the given feature.

Note that there are times when the correlation may not work, for instance changes in lighting may make a different portion of the image appear more similar. Take this example with the same chip of image A:



In this case the chip is mismatched with a portion of the other feature, and the displacement will be incorrect:



Note how both the magnitude and direction of the displacement are very different from the correct match.

Satellite images of glaciers are complex and changes in the ice surface, changes in lighting and changes in satellite angle can cause a number of incorrect matches. These can however be mitigated by appropriate image filtering and velocity map post-processing. GIV aims to automate both of these processes to output fully filtered velocity maps.

It is not in the scope of this user guide to enter into a discussion of the mathematical procedures involved in feature tracking, however these are in themselves interesting. The GIV matlab code is commented in such a way that these equations can be understood by non-specialists, although the length of the code makes such a task time consuming. A recent review by Heid and Kaab (2012) and references therein provide a good starting point for anyone interested in learning further.

References:

Adrian, L., Adrian, R. J., & Westerweel, J. (2011). Particle image velocimetry (No. 30). Cambridge university press.

Gaudin, D., Taddeucci, J., Scarlato, P., Moroni, M., Freda, C., Gaeta, M., & Palladino, D. M. (2014). Pyroclast Tracking Velocimetry illuminates bomb ejection and explosion dynamics at Stromboli (Italy) and Yasur (Vanuatu) volcanoes. Journal of Geophysical Research: Solid Earth, 119(7), 5384-5397.

Gaudin, D., Taddeucci, J., Houghton, B. F., Orr, T. R., Andronico, D., Del Bello, E., ... & Scarlato, P. (2016). 3-D high-speed imaging of volcanic bomb trajectory in basaltic explosive eruptions. Geochemistry, Geophysics, Geosystems, 17(10), 4268-4275.

Muste, M., Fujita, I., & Hauet, A. (2008). Large-scale particle image velocimetry for measurements in riverine environments. Water resources research, 44(4).

Scambos, T. A., Dutkiewicz, M. J., Wilson, J. C., & Bindschadler, R. A. (1992). Application of image cross-correlation to the measurement of glacier velocity using satellite image data. Remote sensing of environment, 42(3), 177-186.

Heid, T., & Kääb, A. (2012). Evaluation of existing image matching methods for deriving glacier surface displacements globally from optical satellite imagery. Remote Sensing of Environment, 118, 339-355.

3 Getting Started: Step by Step guide (with images)

This portion of the guide aims to guide a new user through the process of running their first glacier velocity calculations and assumes no prior computing knowledge. We will first run through the process of obtaining satellite imagery, then running GIV and finally the main outputs.

3.1 Obtaining satellite images

There are many different methods for obtaining a satellite imagery dataset, and this section is simply intended as a guide for those unsure of where to obtain new satellite imagery for a glacier. If you already have a folder containing a timeseries of satellite images, you may skip to section 3.2.

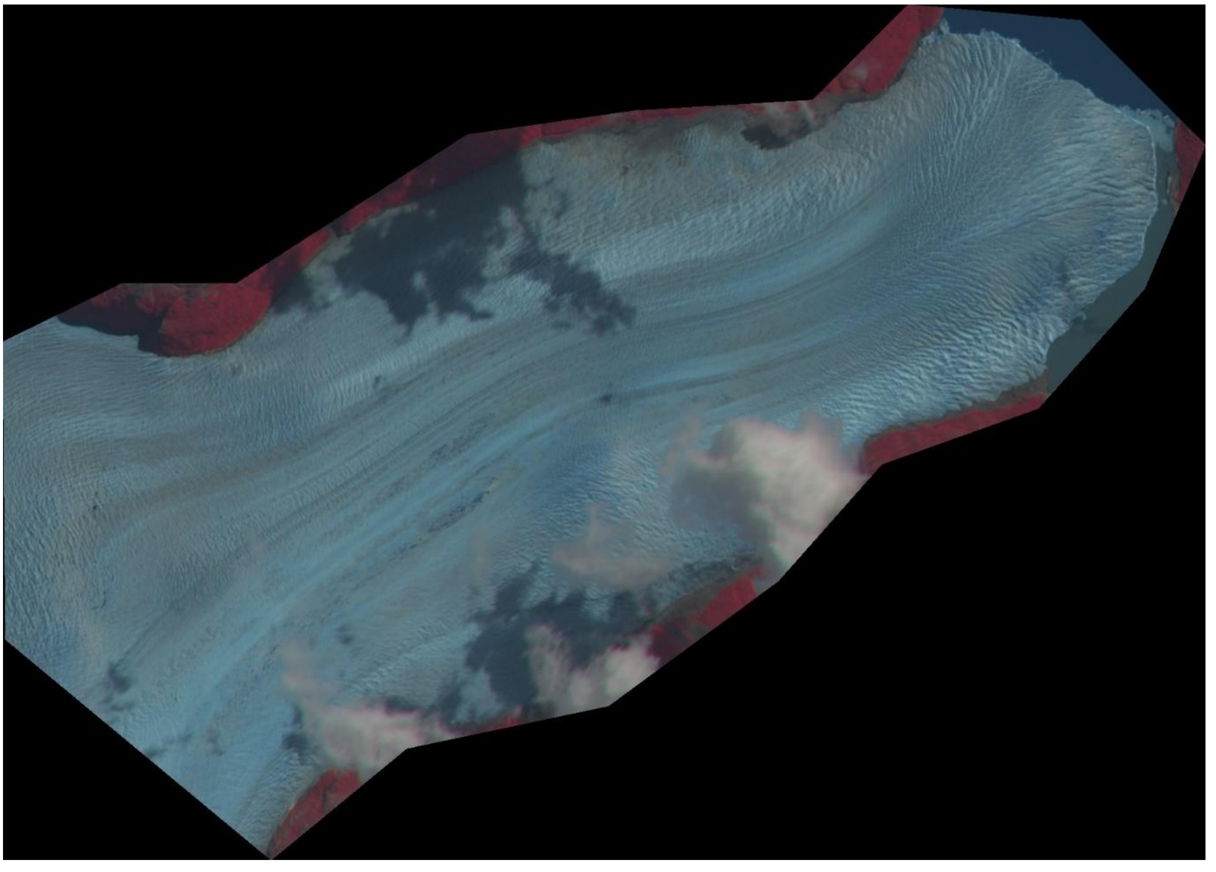
The two main satellite datasets that you may be using (at a first stage at least) will be Landsat (NASA/USGS, running since the 1970s, up to ~15m resolution panchromatic band) and Sentinel 2 (ESA, running since 2015, 10m resolution). For recent images, Sentinel 2 has a better temporal and spatial resolution.

To download Sentinel 2 imagery, use the Sentinel EO-Browser: https://www.sentinel-hub.com/explore/eobrowser/

To download Landsat imagery, use USGS's earthexplorer tool:
https://earthexplorer.usgs.gov/

Both websites will require you to create a free account in order to download imagery, and this will always remain free.

For simplicity, GIV is set up to run on png. or .jpg images. Please download imagery in this format rather than georeferenced formats (e.g. .tif or arc ascii). An example image of Perito-Moreno glacier downloaded from Sentinel EO-Browser is shown below.



3.2 Running GIV

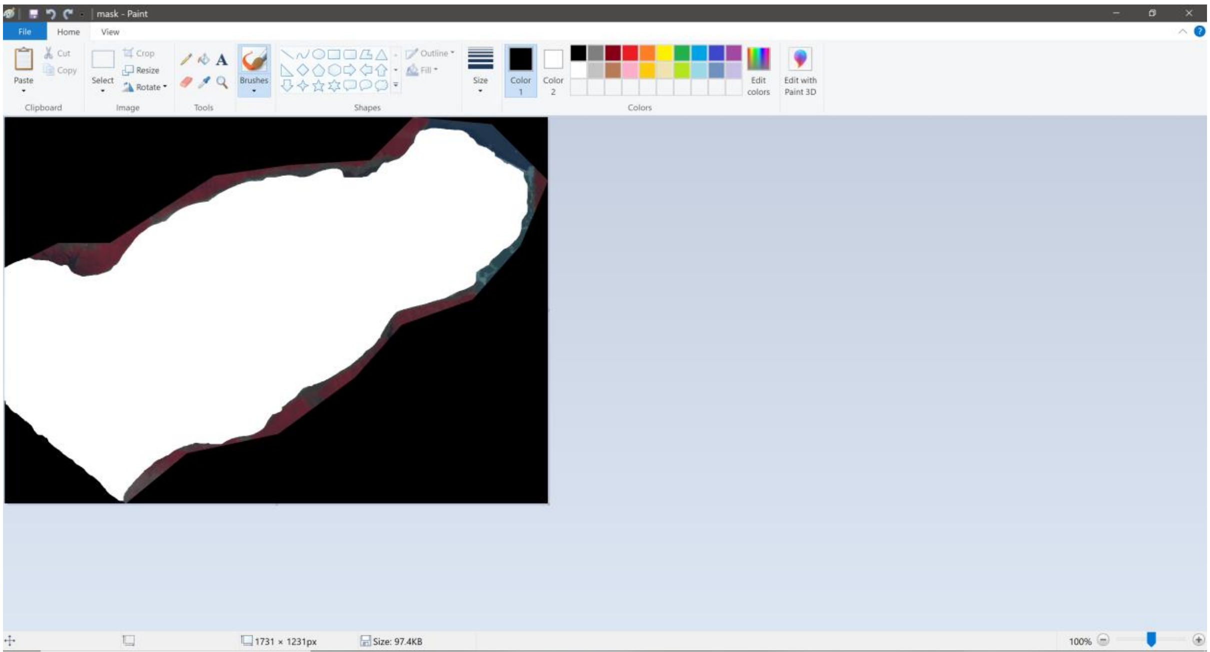
The first step is to ensure that imagery is stored in a folder and named appropriately. Images must be named according to the convention YYYYMMDD (e.g. 20160125 for the 25 th of January 2016). All jpg or png images must be in this format:

20200101	3/30/2020 9:26 AM	JPG File	198 KB
20200106	3/30/2020 9:25 AM	JPG File	71 KB
20200108	3/30/2020 9:25 AM	JPG File	74 KB
20200116	3/30/2020 9:24 AM	JPG File	109 KB
20200118	3/30/2020 9:24 AM	JPG File	163 KB
20200121	3/30/2020 9:24 AM	JPG File	190 KB
20200123	3/30/2020 9:23 AM	JPG File	143 KB
20200126	3/30/2020 9:23 AM	JPG File	141 KB
20200131	3/30/2020 9:22 AM	JPG File	164 KB
20200205	3/30/2020 9:22 AM	JPG File	173 KB
20200207	3/30/2020 9:19 AM	JPG File	208 KB
20200210	3/30/2020 9:19 AM	JPG File	202 KB
20200212	3/30/2020 9:18 AM	JPG File	128 KB
20200215	3/30/2020 9:18 AM	JPG File	97 KB
20200220	3/30/2020 9:18 AM	JPG File	212 KB
20200222	3/30/2020 9:18 AM	JPG File	189 KB
20200227	3/30/2020 9:12 AM	JPG File	211 KB
20200301	3/30/2020 9:12 AM	JPG File	143 KB
20200311	3/30/2020 9:11 AM	JPG File	113 KB
20200313	3/30/2020 9:11 AM	JPG File	224 KB
20200316	3/30/2020 9:10 AM	JPG File	174 KB
20200318	3/30/2020 9:10 AM	JPG File	152 KB
20200328	3/30/2020 9:09 AM	JPG File	189 KB
mask	3/30/2020 9:41 AM	JPG File	98 KB
save_image	3/30/2020 9:26 AM	JPG File	198 KB

Note that two additional files are stored here: mask and save_image. These must also be defined prior to running GIV.

-save_image is simply a good image used in the plotting stage. Simply duplciate the best satellite image in your timeseries and rename this as ‘save_image’ (case sensitive!)

-mask is a mask of the areas GIV will calculate velocities. Relevant areas should be drawn in pure white (RGB 255,255,255). This can for instance be done using MS Paint on Windows by drawing directly onto one of the images:

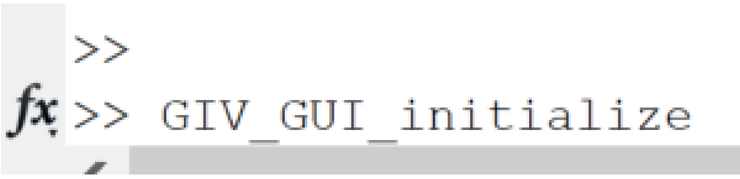


An additional file may be required in some cases, named ‘stable’. This is used to define regions of stable ground for applying a correction. This must be drawn in the same way as the ‘save_image’, with stable ground in pure white.

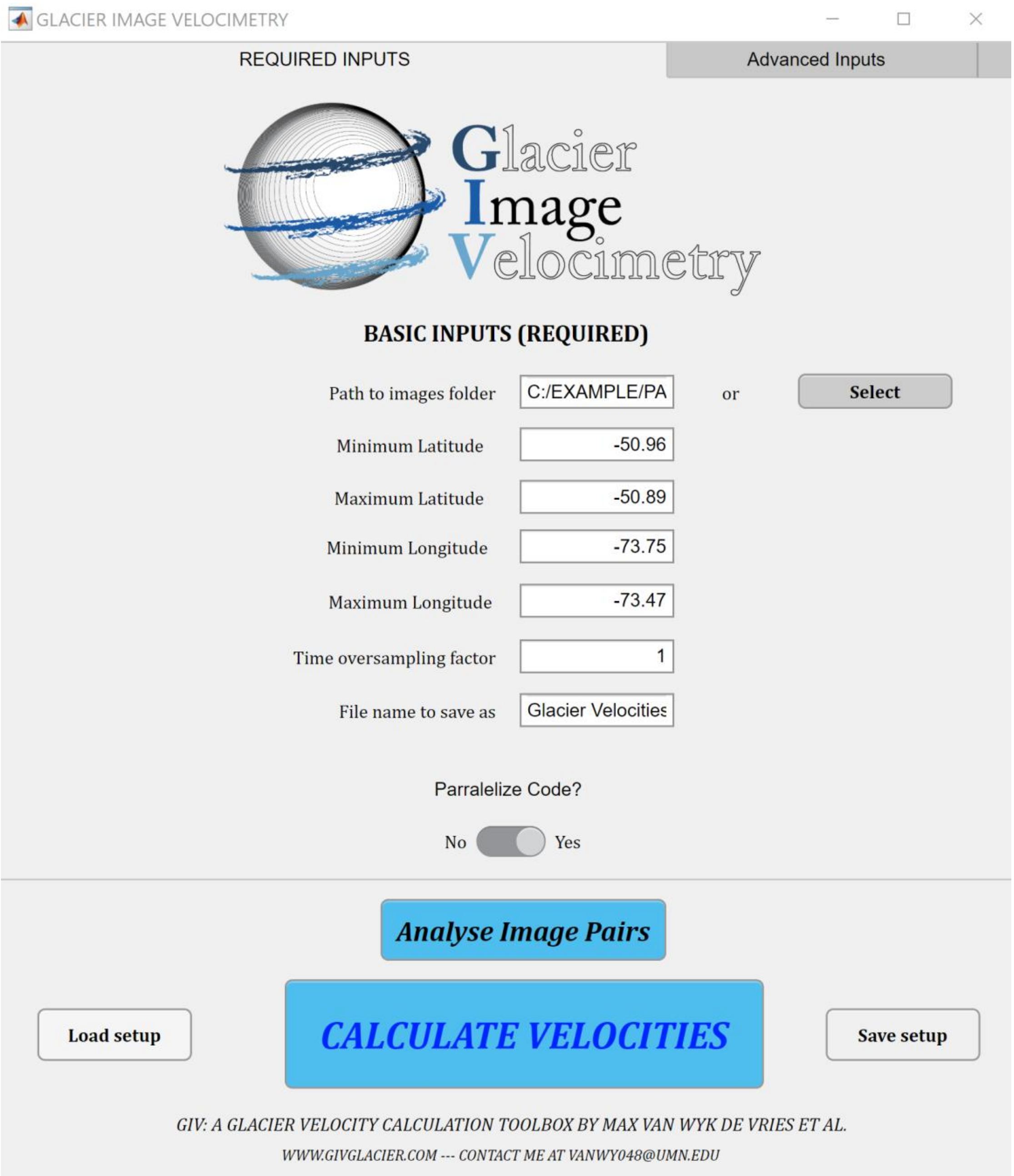
We are now ready to run GIV!

This may be done in two different ways, either by opening up the GUI through MATLAB or by opening the standalone app (requires matlab runtime to be installed to run).

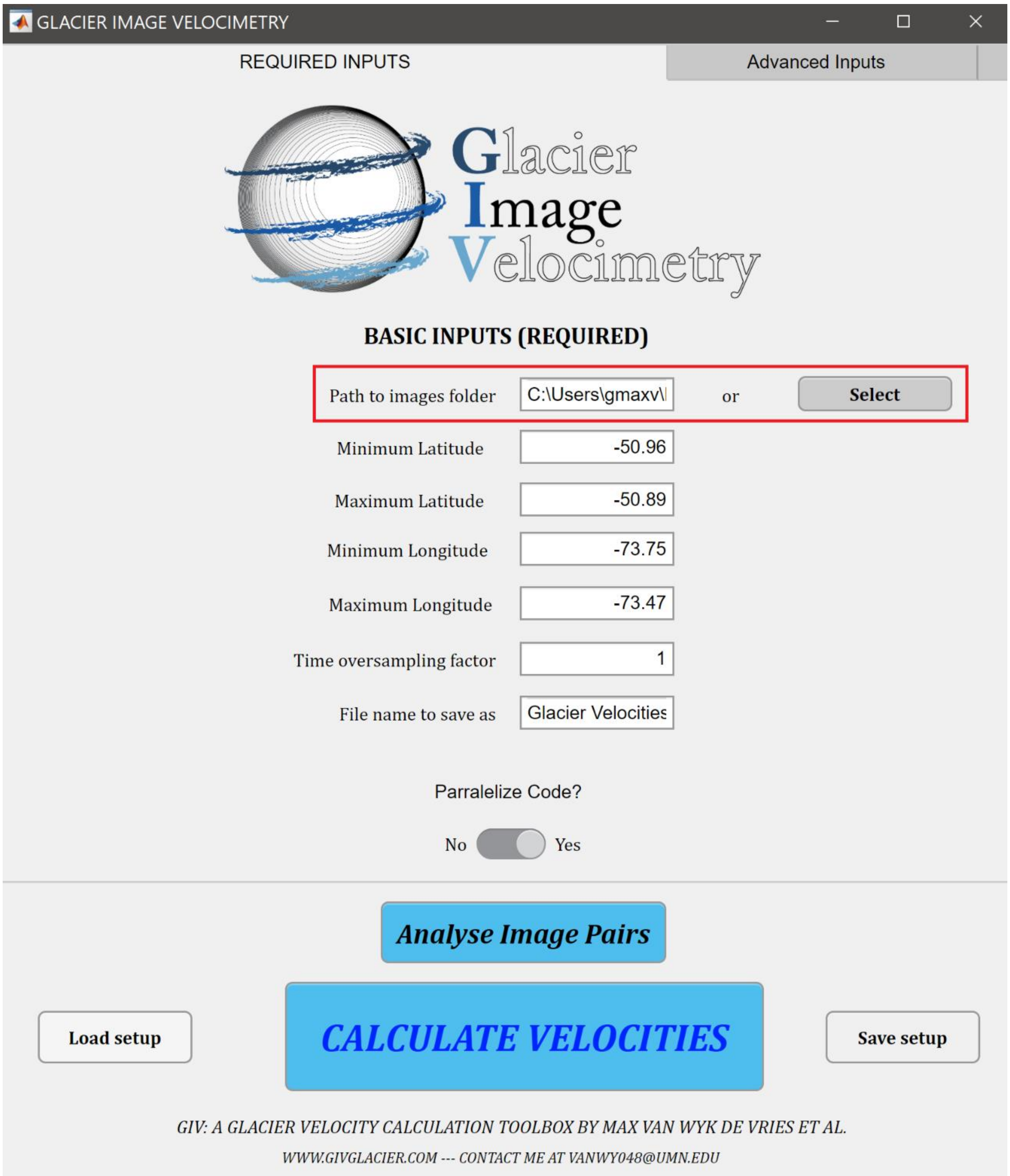
To open the GUI up from MATLAB simply type ‘GIV_GUI_initialize’ into the command line.



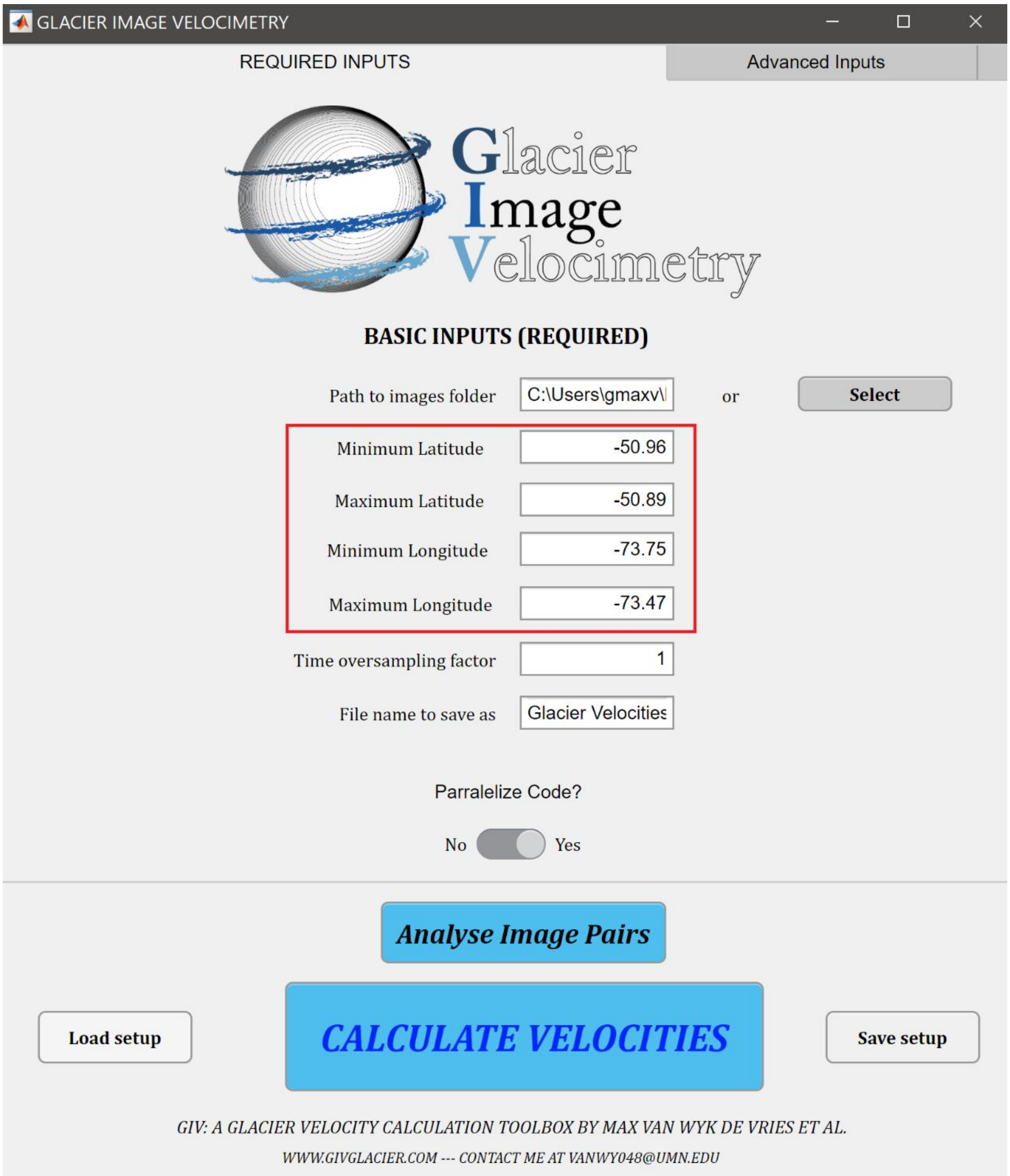
Or open the app. This will open up the GIV GUI:



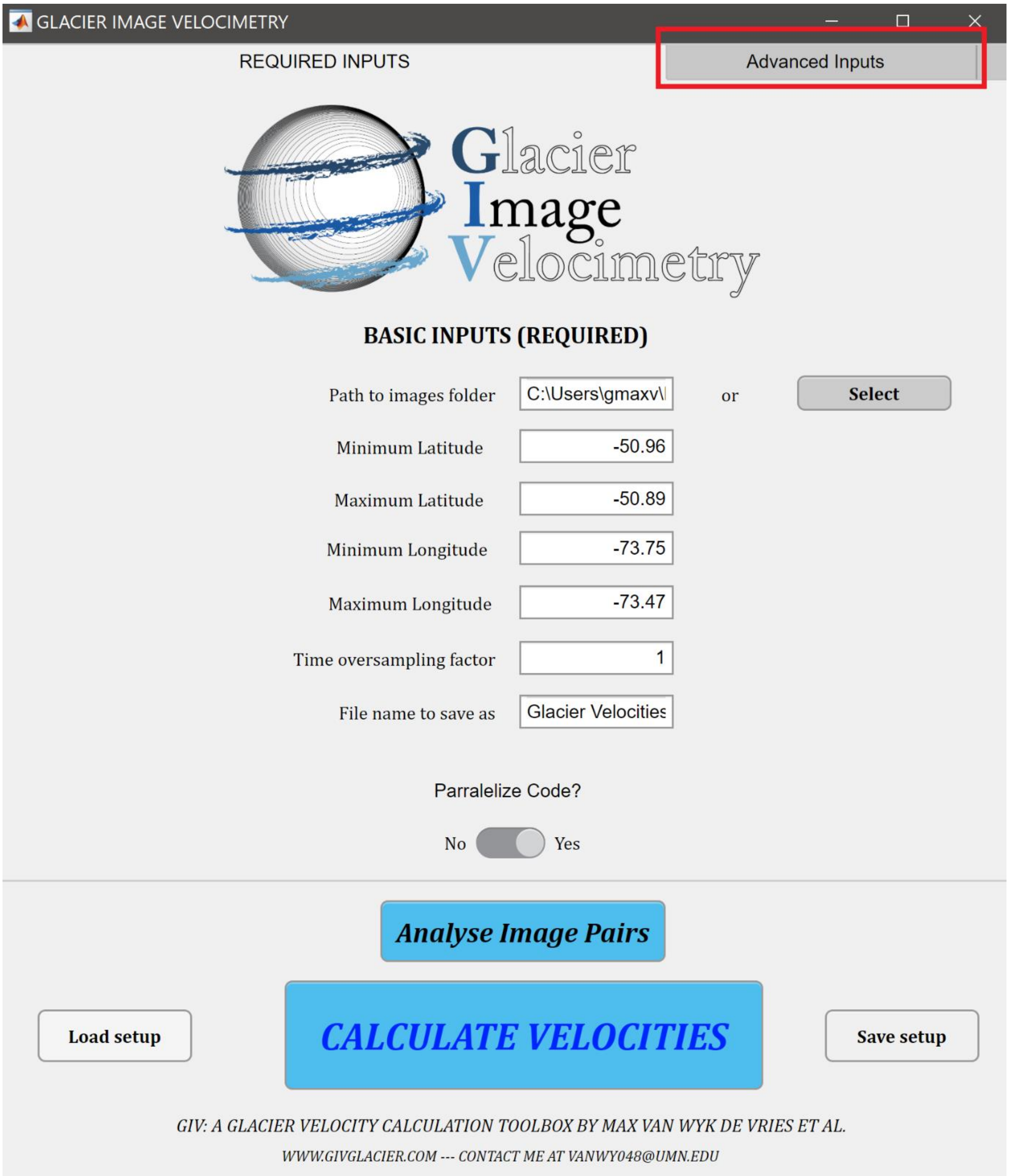
First, click on the Path to images folder SELECT button, this will open up a search window to locate the folder containing your satellite images. Select this folder, and the path will be automatically loaded into the box.



The next step is to enter to Minimum and maximum longitudes and lattitudes for your image.

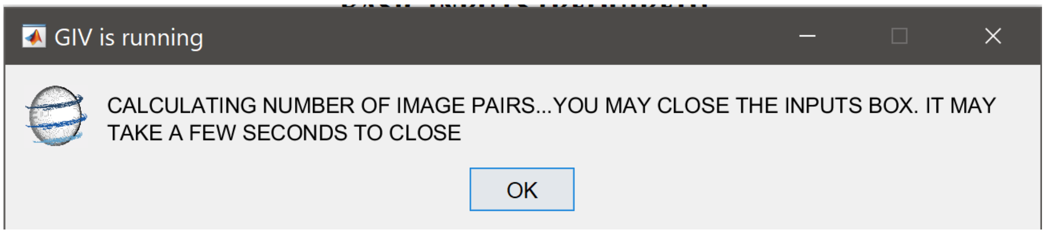


You may now want to enter a name for this model run (e.g. Perito_Moreno_Velocity_test). A number of advanced parameters may also be edited, these are described in the following section. In this case we will show how to run GIV with the default parameters (these should be suitable for a majority of glaciers). If desirable, Advanced parameters may be change from the tab at the top of the GUI:

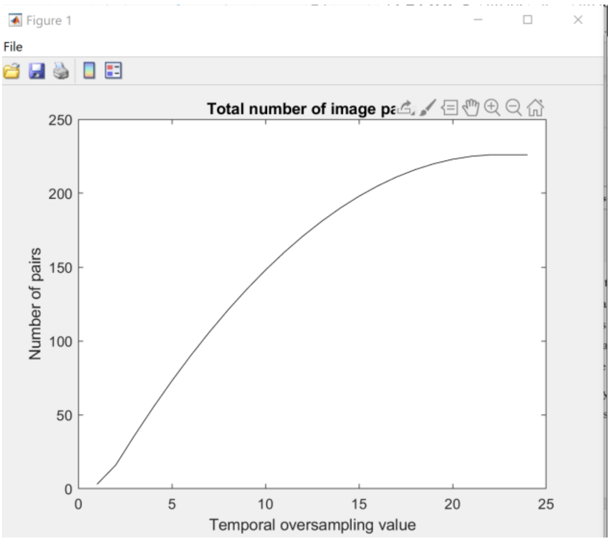


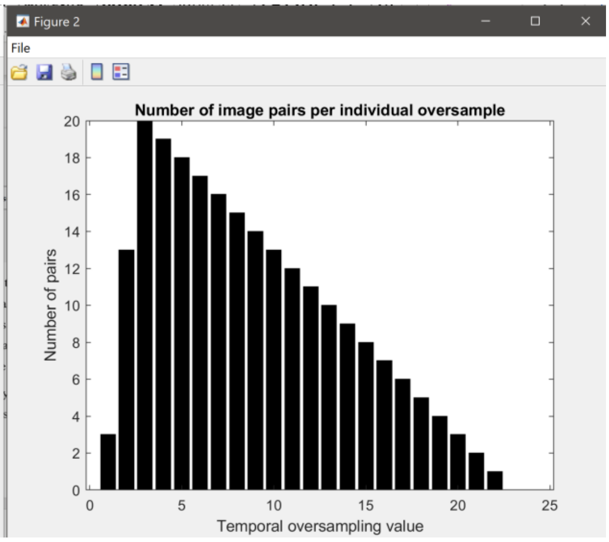
Following this, I recommend running the initial “Analyse Image Pairs” function in order to decide what Time oversampling factor to set. A time oversampling factor of 1 only runs images in series (e.g. image 1 matched with image 2, image 2 matched with image 3, etc), a time oversampling factor of two runs images in series and in series + 1 (e.g. image 1 matched with image 3, image 2 matched with image 4, etc).

The “Analyse image pairs” script will generate two plots that will help understand how many image pairs this results in and what value is suitable. GIV will display the following message box when this begins:



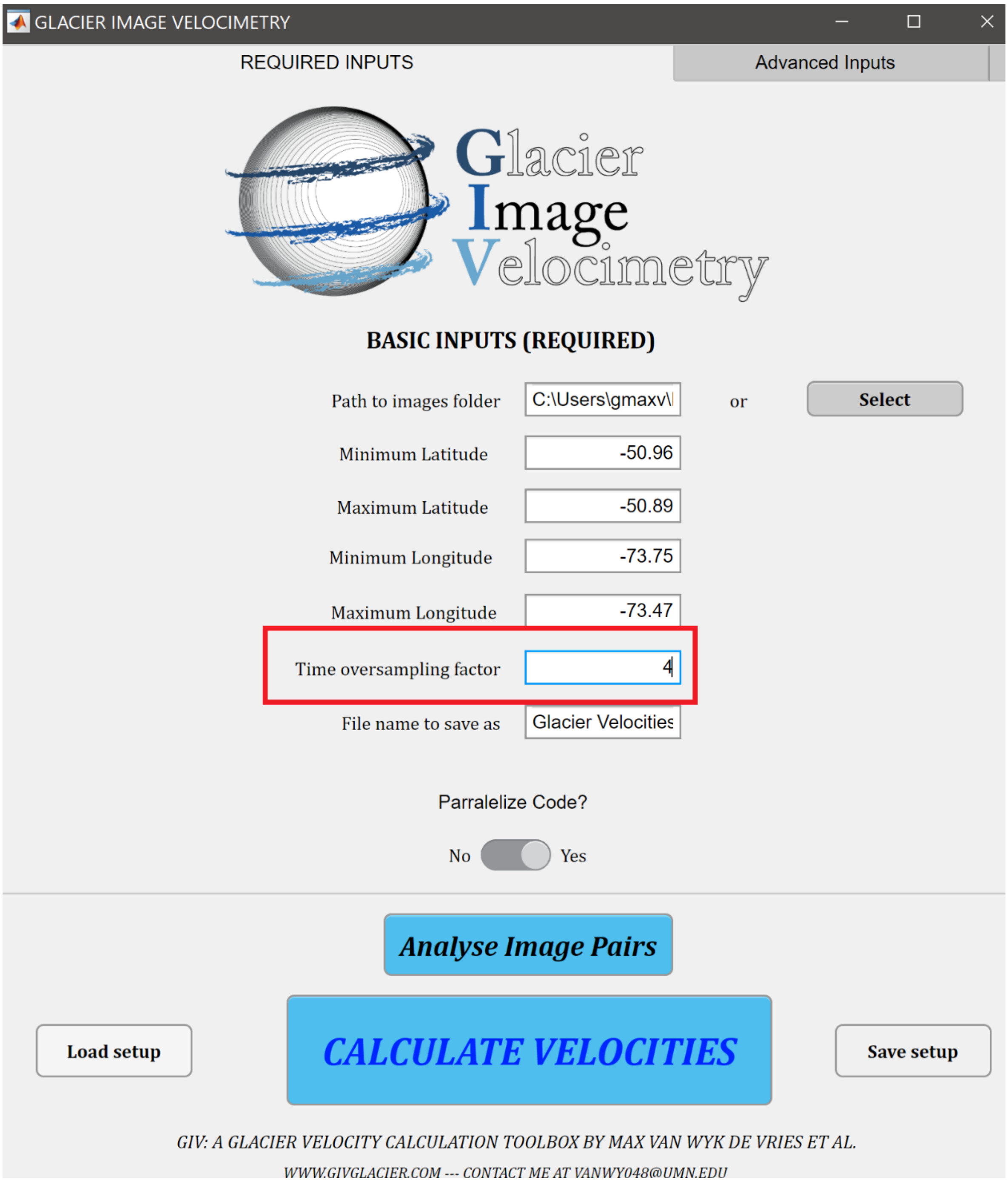
This step should be relatively rapid (a few seconds to a minute). The following two plots will display, one showing the cumulative number of image pairs for a given oversampling value and a second giving the individual increase in number of image pairs for a given oversample increase:



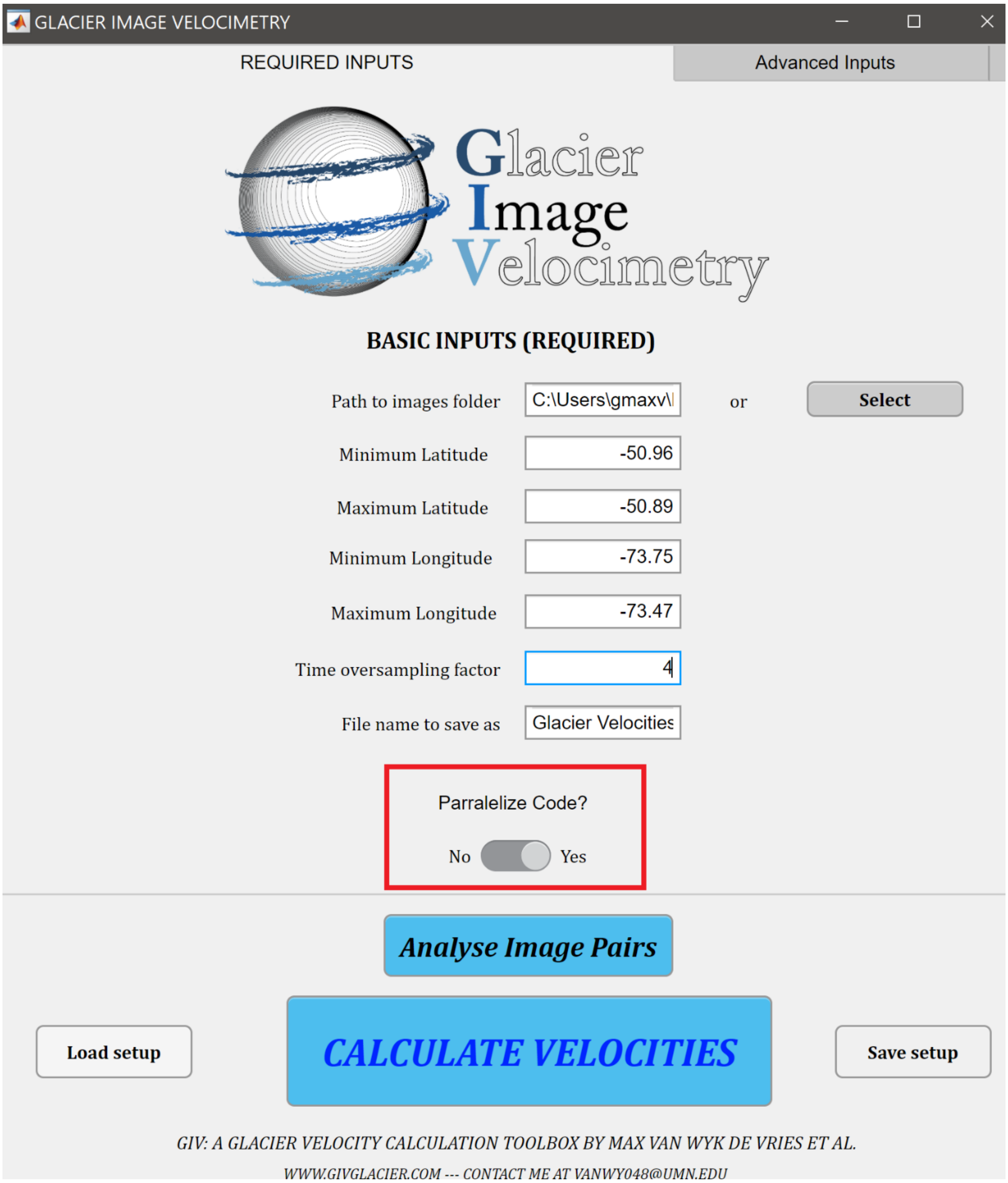


These plots should help guide what temporal oversampling value is ideal. Runtime will be close to proportional to the number of image pairs used, but large numbers of image pairs will generally lead to more accurate velocity maps. For initial tests choosing a relatively small sample (e.g. 50 image pairs) may work best, while for full runs selecting the maximum number of image pairs is generally best.

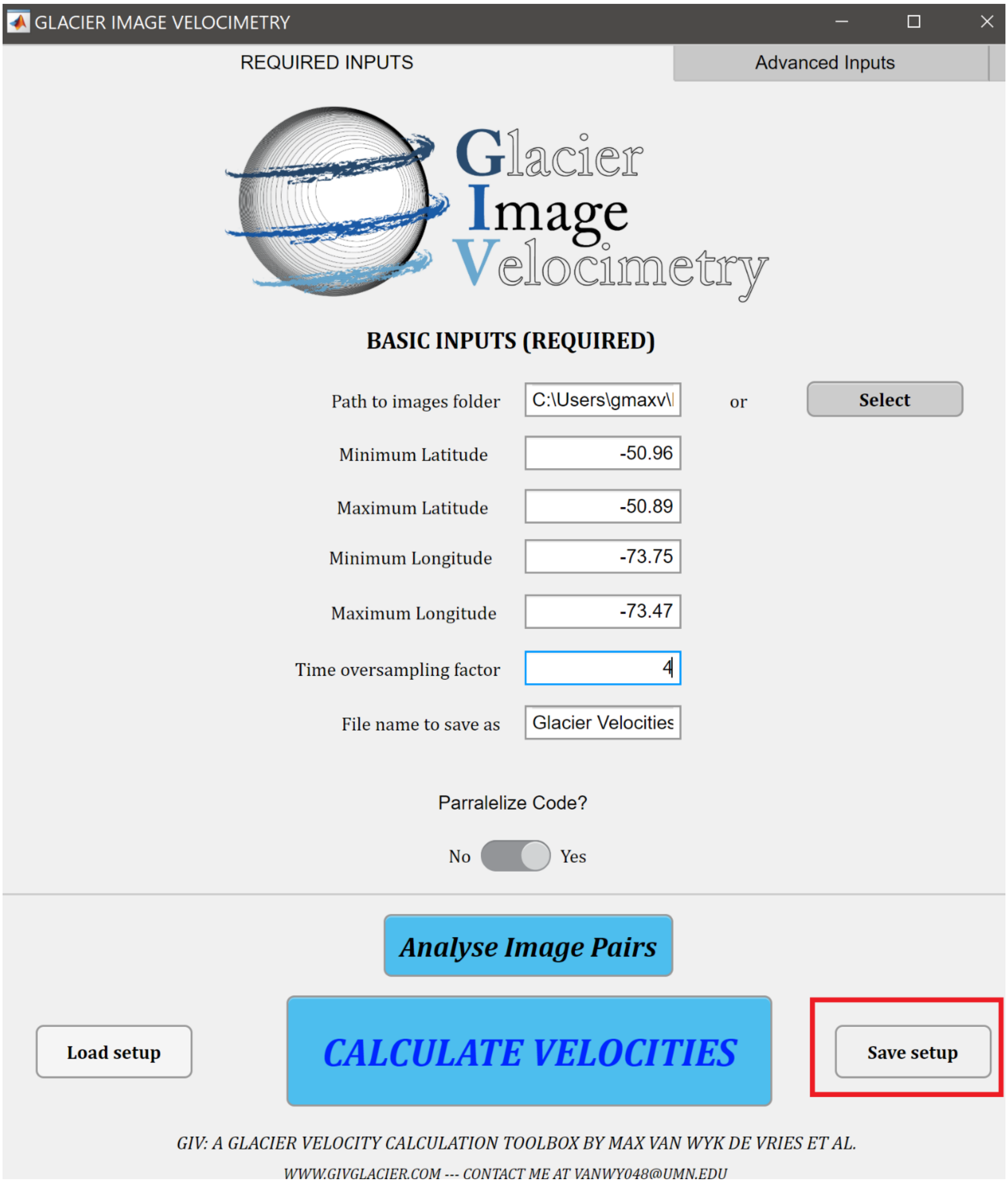
We may then adjust this temporal oversampling parameters. Here we choose a value of 4, resulting in around 50 image pairs:



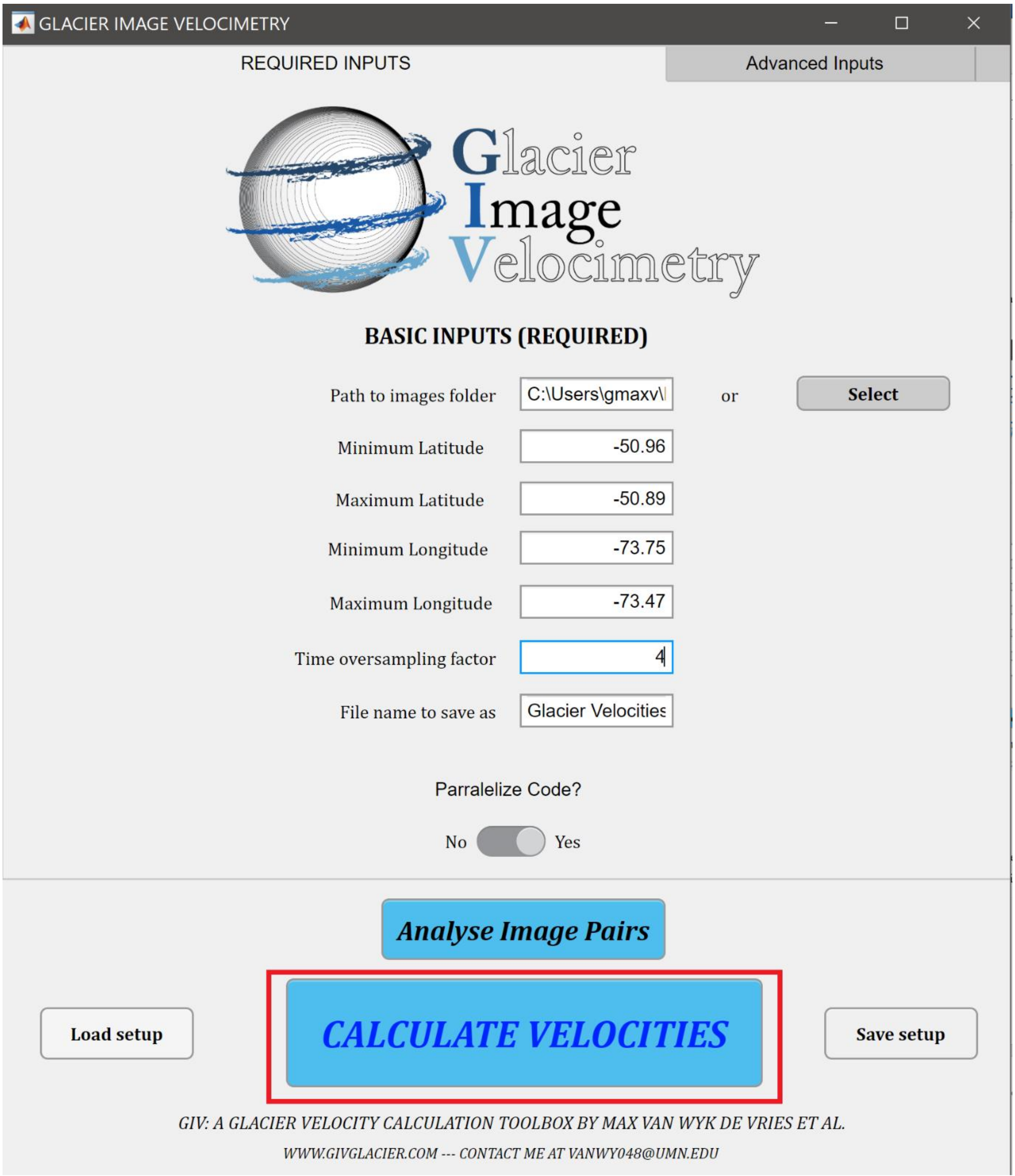
Finally, you will want to decide whether you want to calculate the different image pairs in parallel or separately. The potential for parallelisation will depend on the number of cores in the computer you are running GIV from. If you opened GIV from the command line interface, this option will require MATLAB's parallel computing toolbox.



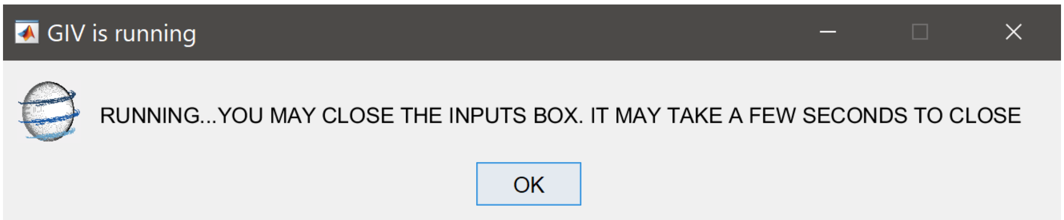
We may now choose to save this setup for future reference. Simply press the 'Save setup' button and it will open a dialogue box prompting you to choose a save location:



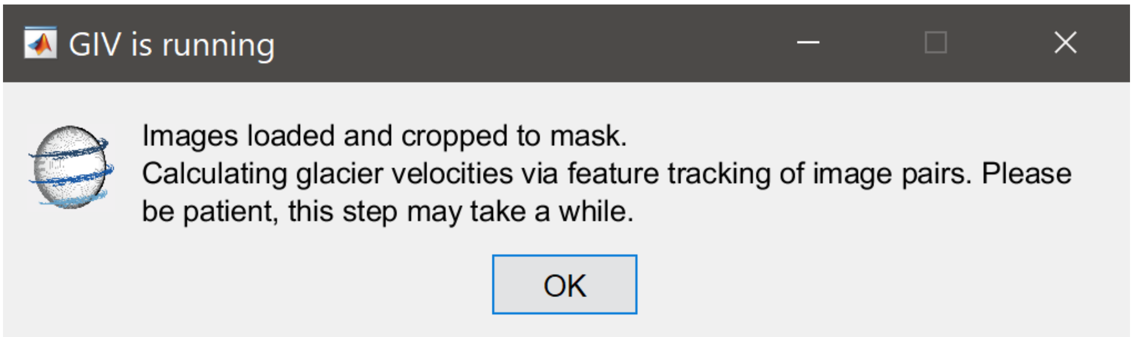
This set of input parameters may be automatically run in the future simply by selecting the 'Load setup' button to the left of the menu. We are now ready to start the run.

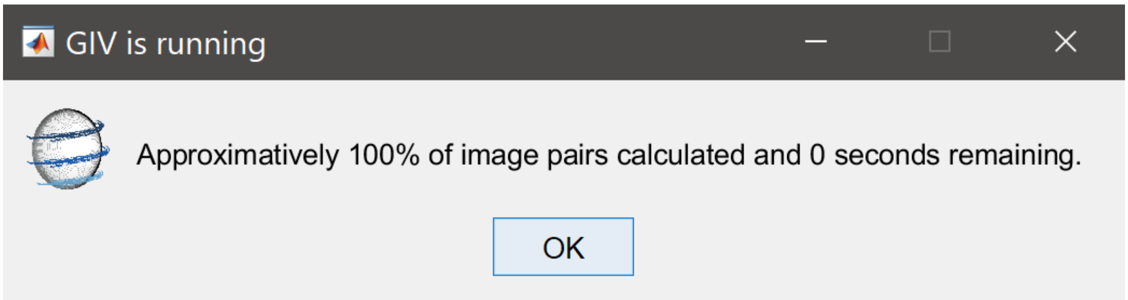


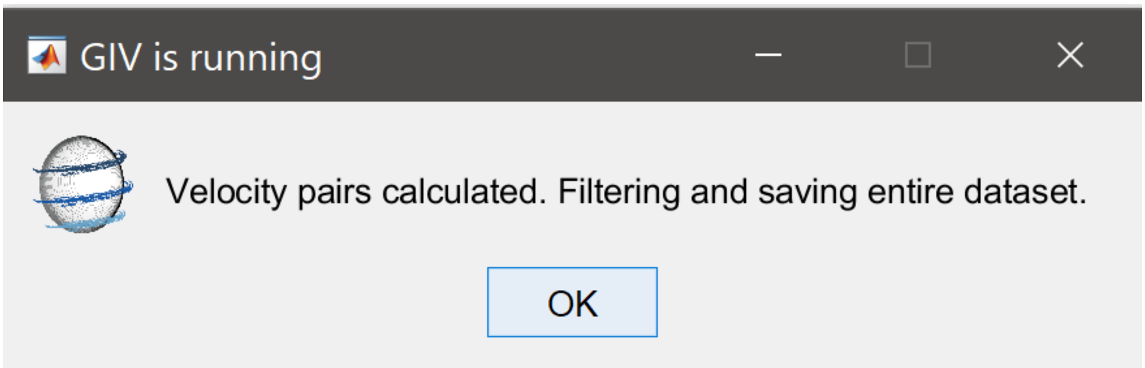
Press the large ‘Calculate Velocities’ button and the following message will pop up:



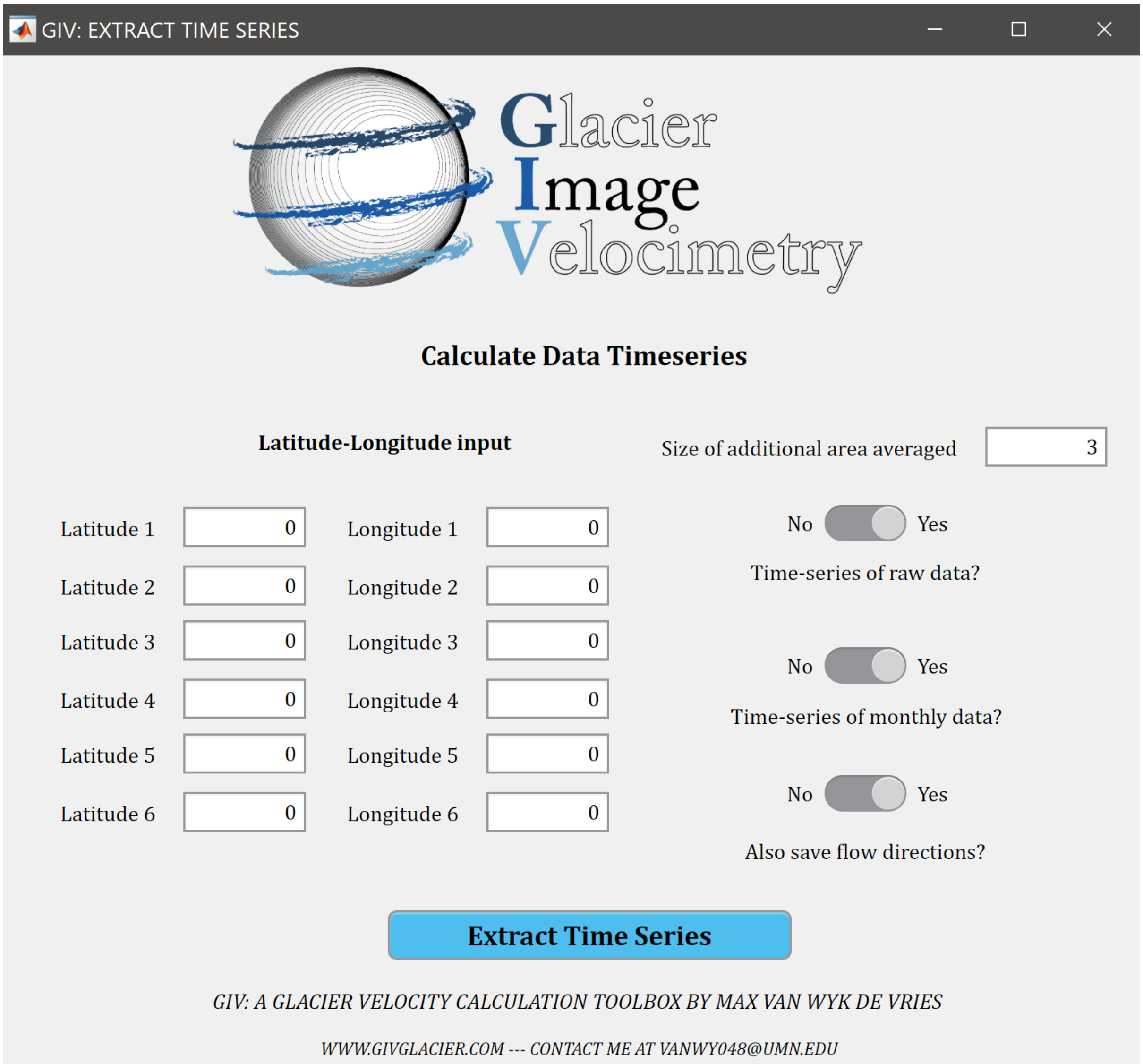
At this stage you may (but are not required to) close the inputs menu, GIV will run regardless. The following messages will appear during the run, with one message providing an estimate of the time remaining prior to run completion. The time is calibrated to your local machine, and will increase in accuracy as the run progresses.







One the run finishes, a separate menu will open up and prompt you to enter additional lattitude and longitude values. If you wish to generate a timeseries plot you may run through with this step immediately, however you may also come back to it at any point in the future as well.



3.3 GIV outputs

GIV will automatically save a number of files to your computer once the run is finished. In particular it will automatically save the run input values and output arrays as .mat files that may be opened in MATLAB at a later date. This also allows any of the filtering steps to be repeated with different parameters if desired, without the need to re-run the entire image dataset.

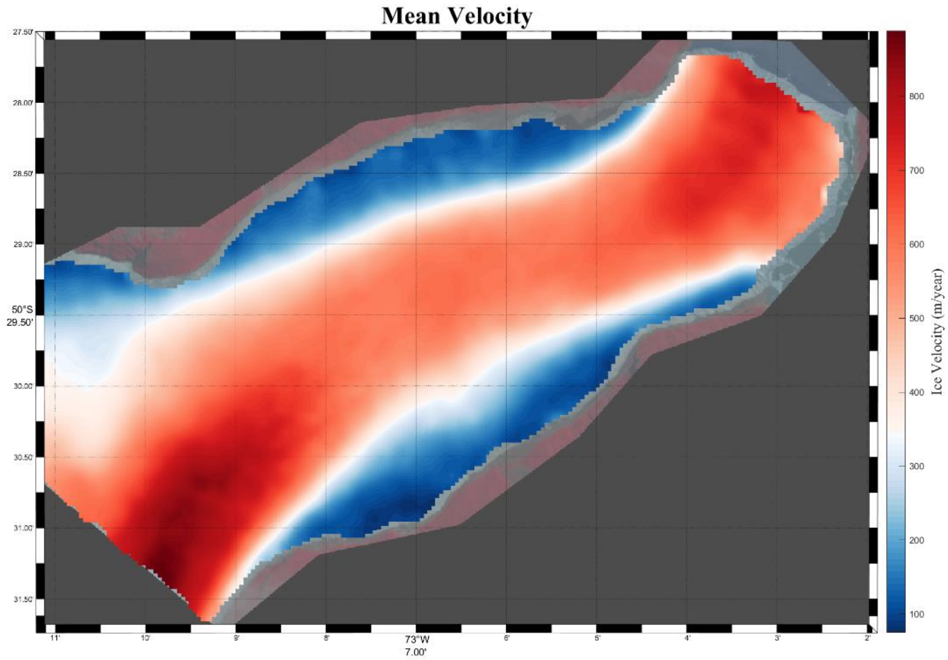
GIV will create a new folder, named 'RESULTS' in your image folder. All runs using this dataset of images will be stored here (although successive runs with the same name will overwrite each other).

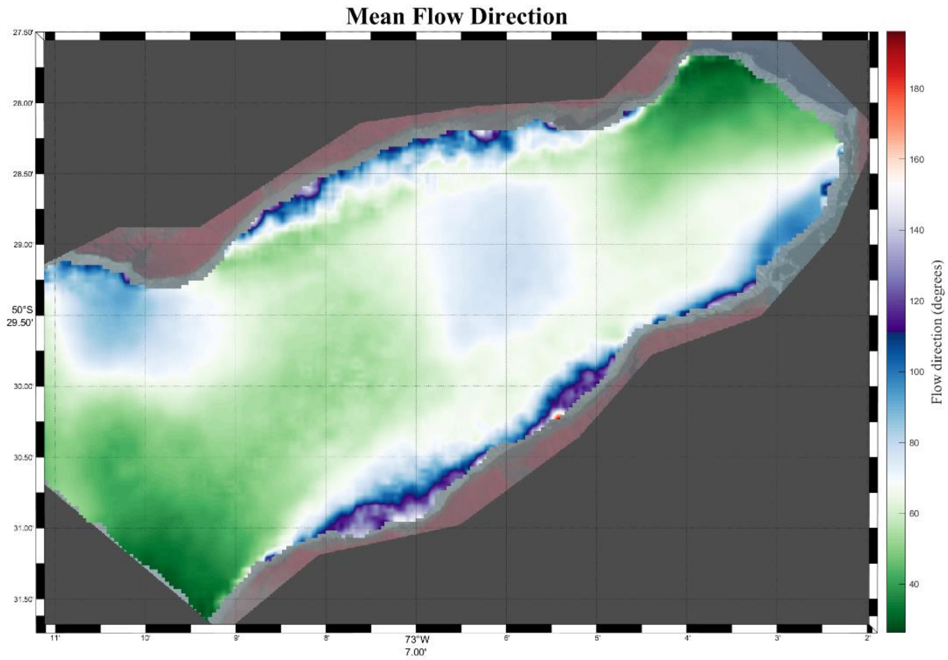
Name	Date	Type	Size	Tags
Results	6/26/2020 12:28 AM	File folder		
20200101	3/30/2020 9:26 AM	JPG File	198 KB	
20200106	3/30/2020 9:25 AM	JPG File	71 KB	
20200108	3/30/2020 9:25 AM	JPG File	74 KB	
20200116	3/30/2020 9:24 AM	JPG File	109 KB	
20200118	3/30/2020 9:24 AM	JPG File	163 KB	
20200121	3/30/2020 9:24 AM	JPG File	190 KB	
20200123	3/30/2020 9:23 AM	JPG File	143 KB	
20200126	3/30/2020 9:23 AM	JPG File	141 KB	
20200131	3/30/2020 9:22 AM	JPG File	164 KB	
20200205	3/30/2020 9:22 AM	JPG File	173 KB	
20200207	3/30/2020 9:19 AM	JPG File	208 KB	
20200210	3/30/2020 9:19 AM	JPG File	202 KB	
20200212	3/30/2020 9:18 AM	JPG File	128 KB	
20200215	3/30/2020 9:18 AM	JPG File	97 KB	
20200220	3/30/2020 9:18 AM	JPG File	212 KB	
20200222	3/30/2020 9:18 AM	JPG File	189 KB	
20200227	3/30/2020 9:12 AM	JPG File	211 KB	
20200301	3/30/2020 9:12 AM	JPG File	143 KB	
20200311	3/30/2020 9:11 AM	JPG File	113 KB	
20200313	3/30/2020 9:11 AM	JPG File	224 KB	
20200316	3/30/2020 9:10 AM	JPG File	174 KB	
20200318	3/30/2020 9:10 AM	JPG File	152 KB	
20200328	3/30/2020 9:09 AM	JPG File	189 KB	
mask	3/30/2020 9:41 AM	JPG File	98 KB	
save_image	3/30/2020 9:26 AM	JPG File	198 KB	

In this 'Results' folder a separate folder will be created for each run name conducted on this dataset. Each of these folders will in turn have four main sub-folders:

Data Figures (Images)	6/26/2020 12:30 AM	File folder
Georeferenced Velocity Data	6/26/2020 12:30 AM	File folder
Initial data backup	6/26/2020 12:28 AM	File folder
Matlab data files	6/26/2020 12:29 AM	File folder

Data figures (Images) will contain .png images of the mean velocity and flow direction for this glacier, as well as a folder containing a monthly breakdown:





Georeferenced velocity data folder will contain georegerenced .GeoTiff images of these same velocity maps that may be loaded into GIS programs, MATLAB/Python, Google Earth or other programs for further analyses or plotting options.

Initial data backup will contain the initial, unfiltered velocity maps for each image pair. This may be used to repeat filtering with different parameters (requires MATLAB).

Finally the MATLAB data files contains saved arrays for different steps of the process that may be loaded into MATLAB for further analyses (e,g, inverting for ice thickness).

Excellent, you have now successfully conducted your first glacier velocity calculation using GIV! Read the next two sections for additional information about advanced input parameters and the raw MATLAB functions.

4 INPUT PARAMETERS

This section will run through the details of each input parameter in the GUI:

1 Path to image folder

This is the path to the folder in which you have saved your images (in yyyymmdd.png or yyyymmdd.jpg format). Either enter the path directly or use the ‘select’ button to open a file selection menu. Note this folder should also contain a file named ‘save_image’ and ‘mask’, as well as ‘stable’ if a stable ground correction is required. See instructions above for more details on these masks.

2 Minimum latitude

The latitude of the southernmost corner of the input images. Note that in the southern hemisphere 'minimum lattitude' will be the larger negative number.

3 Maximum latitude

The latitude of the northernmost corner of the input images.

4 Minimum longitude

The latitude of the westernmost corner of the input images. Note that in the W hemisphere 'minimum lattitude' will be the larger negative number.

5 Maximum latitude

The latitude of the easternmost corner of the input images.

6 Time oversampling factor

This defines whether image pairs are run in series (1-2,2-3,3-4 for a value of 1) or with increase pairing (1-2,1-3,1-4,2-3,2-4,2-5, etc for a value of 3). You may run the 'Analyse image pairs' script (press the blue button below) in order to determine how many image pairs will be chosen. Note that this is also sensitive to the minimum and maximum image time separation chosen.

7 File name

The name of this file run. Note that successive runs on the same image dataset will overwrite each other unless the name is chosen. Note also that all input parameters are automatically saved so there is no need for a very long descriptive name.

8 Parallelize code

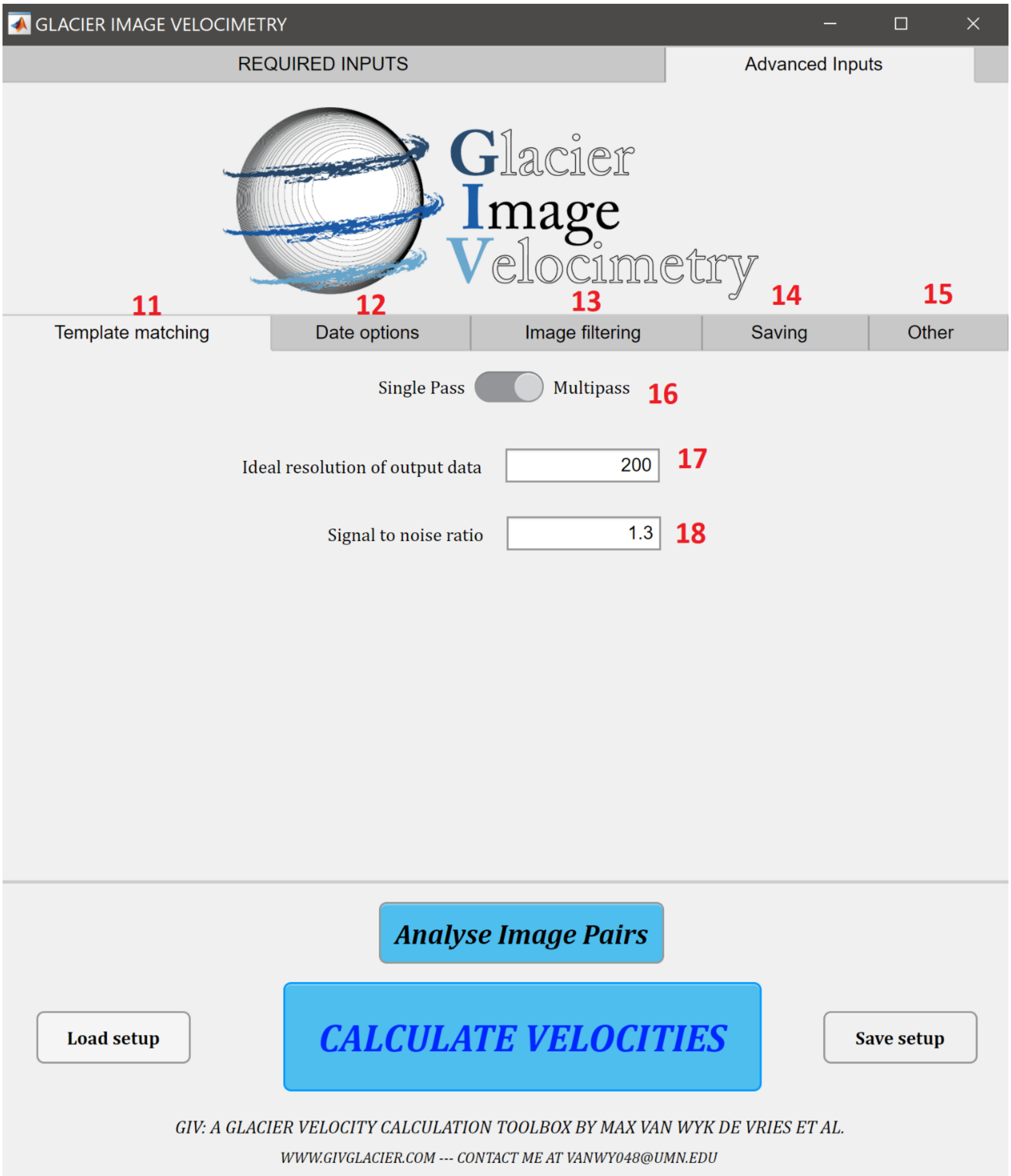
Run different image pairs in parallel. This will improve runtime in a majority of cases, and significantly improve it in machines with 6+ cores (factor or 2 or more speedup). If you are running GIV through MATLAB this requires the parallel computing toolbox to be installed. The default is to run with the maximum number of cores on your computer, if you do not wish to do so please create a different parallel pool prior to running GIV (use command parpool(*number of cores you wish to use*))

9 Required inputs tab

This tab defines the inputs that must be entered in order for GIV to successfully run.

10 Advanced inputs tab

Advanced model options. These may be left at the defaults in most cases, but adjusting them may result in better results. This section will present these inputs. They are sorted into five different tabs.



11 Advanced: Template matching tab

This advanced options tab shows options related to the template matching/feature tracking algorithms.

12 Advanced: Date options

This advanced options tab shows options related to the timing of images and temporal filtering of the satellite imagery timeseries.

13 Advanced: Image filtering

This advanced options tab shows options related to the pre-filtering of satellite images.

14 Advanced: Saving options

This advanced options tab shows options related to the saving of images after velocity maps have been calculated.

15 Advanced: Other

This advanced options tab shows a variety of other options.

16 Template matching: Single or Multipass

Choose whether to run the feature tracking as a single pass or multi-pass. See the The Cryosphere paper for more advanced discussion on the distinction. In a majority of cases Multipass will be more suitable.

17 Template matching: resolution of output data

Choose the desired resolution of the output velocity maps. Note that resolution may not be higher than the original resolution of the satellite imagery, and may differ slightly from the value suggested here. Default resolution is 50m.

18 Template matching: signal to noise ratio

Determines the signal to noise level for preliminary filtering of outlier pixels during the feature tracking. A value of 1.3 is usually suitable for the multipass, while a higher value is necessary for the single pass (3-5 typically). Signal to noise ratio is computed differently in the two methods.

19 Date options: Minimum Year

First year searched for in the image timeseries. If correctly labelled in YYYYMMDD format this will be automatically extracted from the file name. Default is 1900 (i.e. all images counted).

20 Date options: Maximum Year

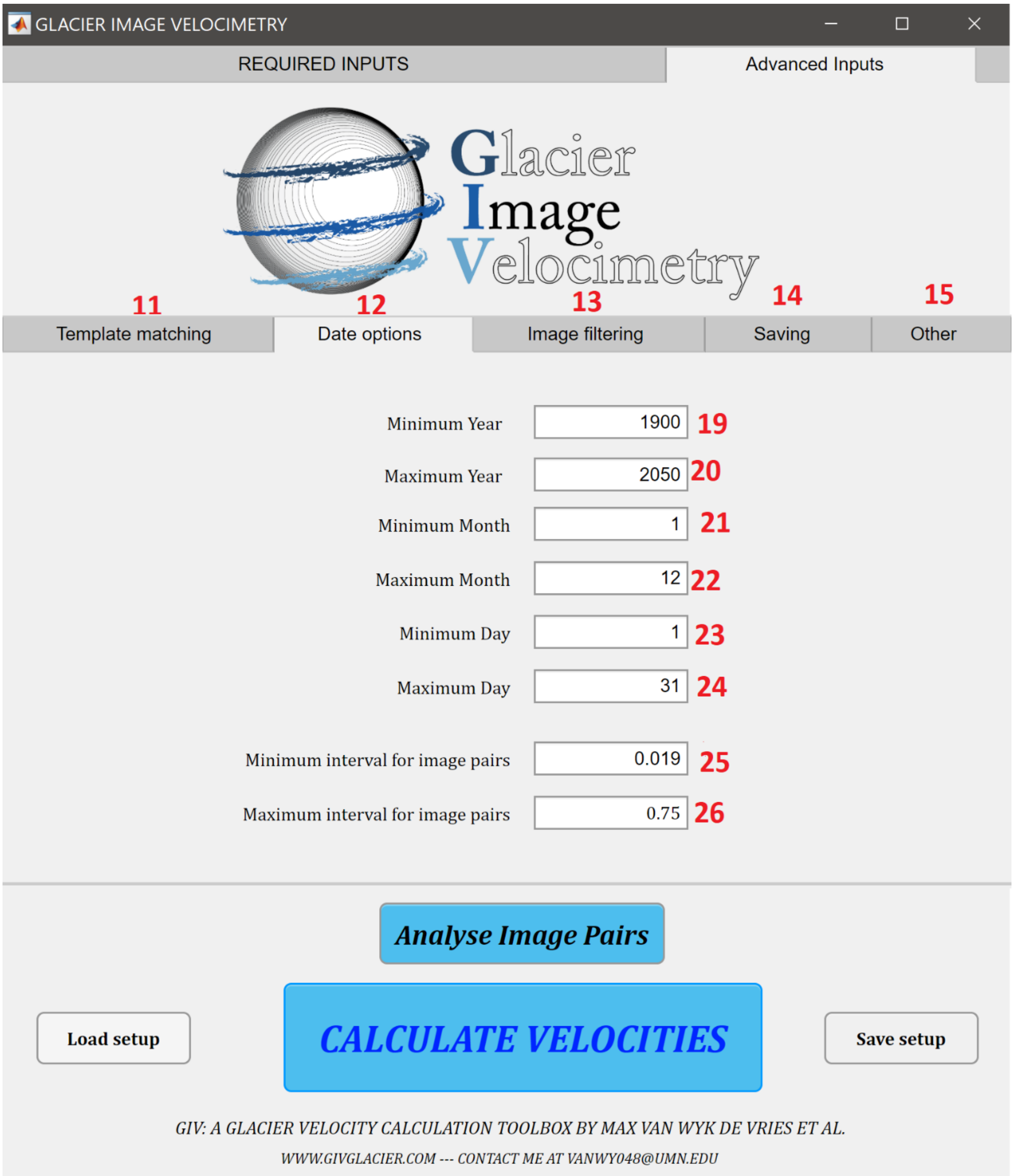
Last year searched for in the image timeseries. If correctly labelled in YYYYMMDD format this will be automatically extracted from the file name. Default is 2050 (i.e. all images counted).

21 Date options: Minimum month

First month searched for in the image timeseries. If correctly labelled in YYYYMMDD format this will be automatically extracted from the file name. Default is 1 (i.e. all images counted).

22 Date options: Maximum month

Last month searched for in the image timeseries. If correctly labelled in YYYYMMDD format this will be automatically extracted from the file name. Default is 12 (i.e. all images counted).



23 Date options: Minimum day

First day searched for in the image timeseries. If correctly labelled in YYYYMMDD format this will be automatically extracted from the file name. Default is 1 (i.e. all images counted).

24 Date options: Maximum day

Last day searched for in the image timeseries. If correctly labelled in YYYYMMDD format this will be automatically extracted from the file name. Default is 31 (i.e. all images counted).

25 Date options: Minimum interval between image pairs

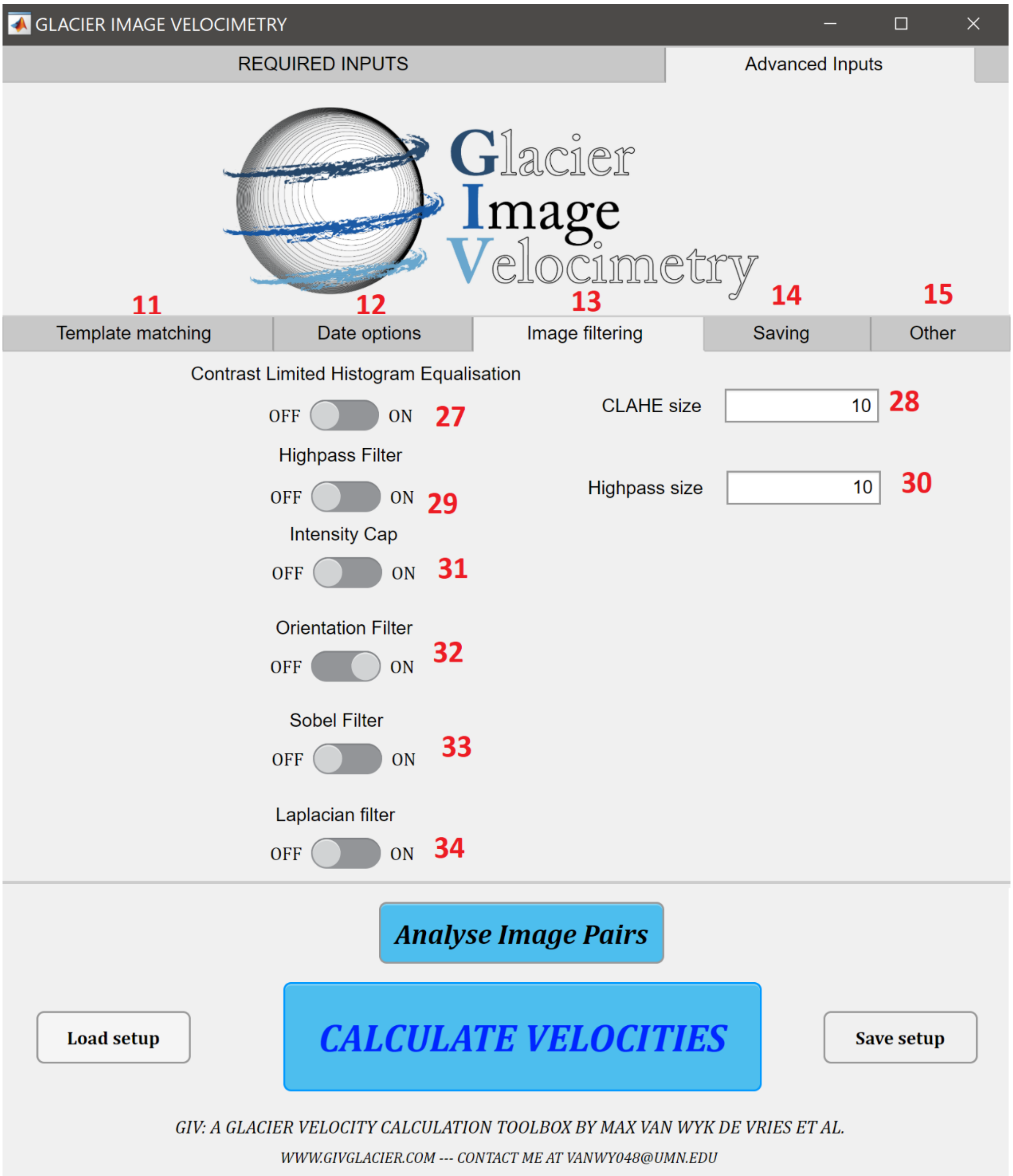
This option sets a minimum temporal spacing between images. Image pairs more closely spaced than this will not be counted. Value in years, default = 0.019 is one week. Setting a minimum image spacing helps minimise the effect of georeferencing and image distortion errors as these effects will be lower relative to the movement of the feature. Note for very slow moving glaciers this value should likely be increased.

26 Date options: Maximum interval between image pairs

This option sets a maximum temporal spacing between images. Image pairs more distantly spaced than this will not be counted. Value in years, default = 0.75 is 9 months. Setting a maximum image separation is often necessary to maintain coherence of the surface of the glacier, such that features may be found in the second image. Note for very slow moving glaciers this value should likely be increased.

27 Image filtering options: Contrast limited histogram equalisation

The contrast limited histogram equalisation is an image filter that reduces the local contrast in pixels throughout the entire image. It may be advantageous to reduce the effect of outlier or 'extreme bright' pixels in the cross-correlation steps. Filter based on CLAHE algorithm from PIVlab (Thielicke and Stamhuis, 2014).



28 Image filtering options: Contrast limited histogram equalisation size

Size (in pixels) of the CLAHE filter. Larger sizes will normalize to a larger number of local pixels.

29 Image filtering options: Highpass filter

Edge emphasizing highpass filter. This may improve the number of correct matches by emphasizing the high frequency ‘features’ of interest and reducing lower frequency background colour variation.

30 Image filtering options: Highpass filter size

Highpass filter size (pixels).

31 Image filtering options: Intensity capping filter

This filter caps pixel intensity at a given value. This may enable better contrast in the low intensity portions of an image and better matches.

32 Image filtering options: Orientation filter

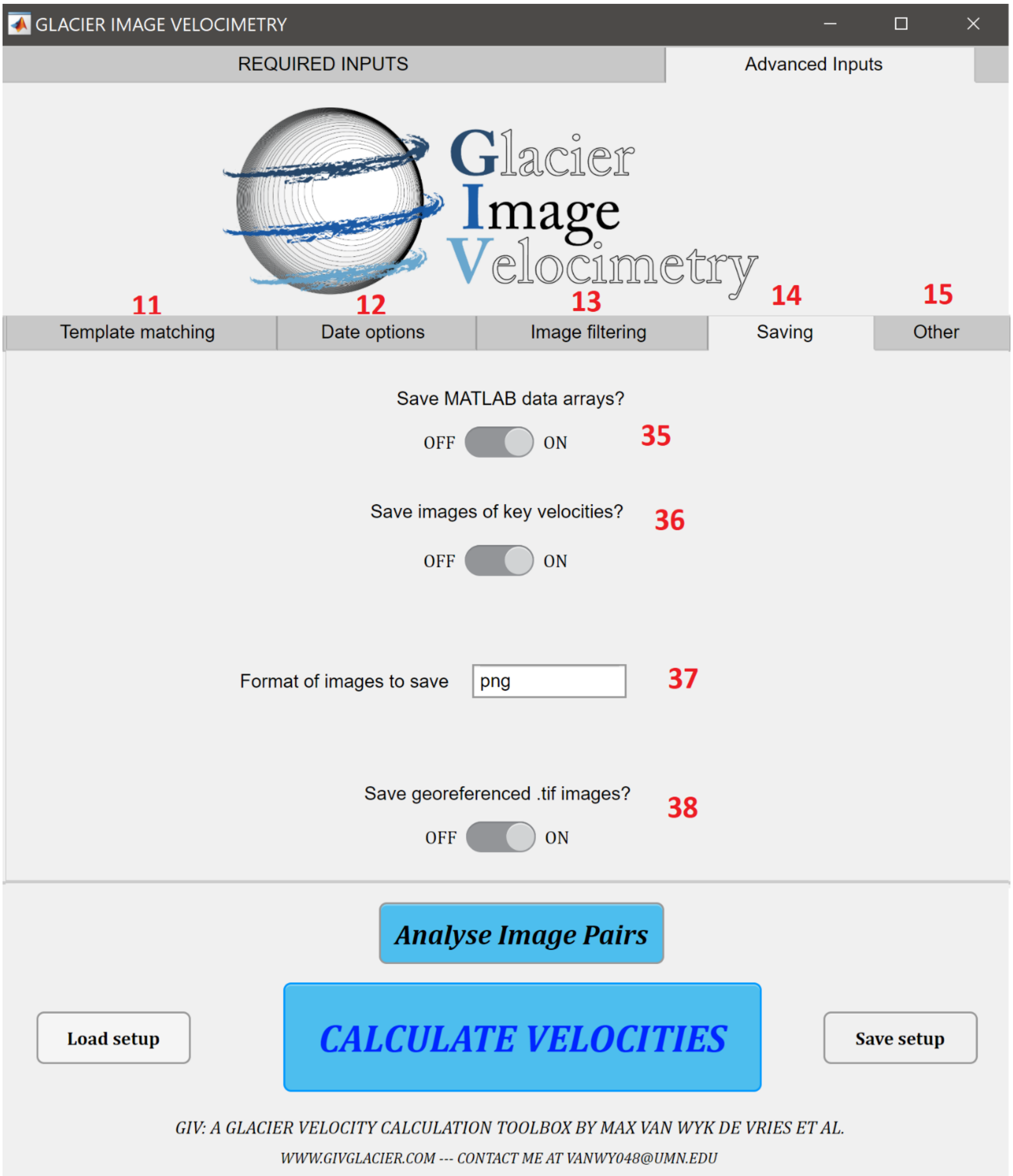
Custom made orientation filter, termed ‘Near Anisotropic Orientation Filter’. This is the default image filter, and will usually lead to the best match results. This filter converts features into ‘orientation filters’ based simply on the direction of pixel intensity gradients, discarding magnitudes. It will normalize intensities in cloud free, cloudy and shadowed regions and increase the magnitude of features to track. See the The Cryosphere 2020 paper for the exact equations involved.

33 Image filtering options: Sobel size

Applies a Sobel filter to the image. This is one type of edge emphasizing filter and may improve the number of successfully tracked features. In most cases the orientation filter or highpass filter will be best adapted.

34 Image filtering options: Laplacian filter

Applies a Laplacian filter to the image. This is one type of edge emphasizing filter and may improve the number of successfully tracked features. In most cases the orientation filter or highpass filter will be best adapted.



35 Saving: Save matlab arrays?

This option gives you the choice of saving or not saving the raw matlab input and output arrays. You may wish to not save them if 1) storage space is an issue or 2) you are running GIV through the app and do not have access to matlab. In most cases these should be saved so as to retain a copy of the original inputs file for future scrutiny.

36 Image filtering options: Save velocity images.

This option gives you the choice of saving or not saving images of the output velocities. You may wish to not save them if you wish to generate your own plots with a different program. The file sizes are small and in most cases these provide a good initial check of the data.

37 Saving: Image format

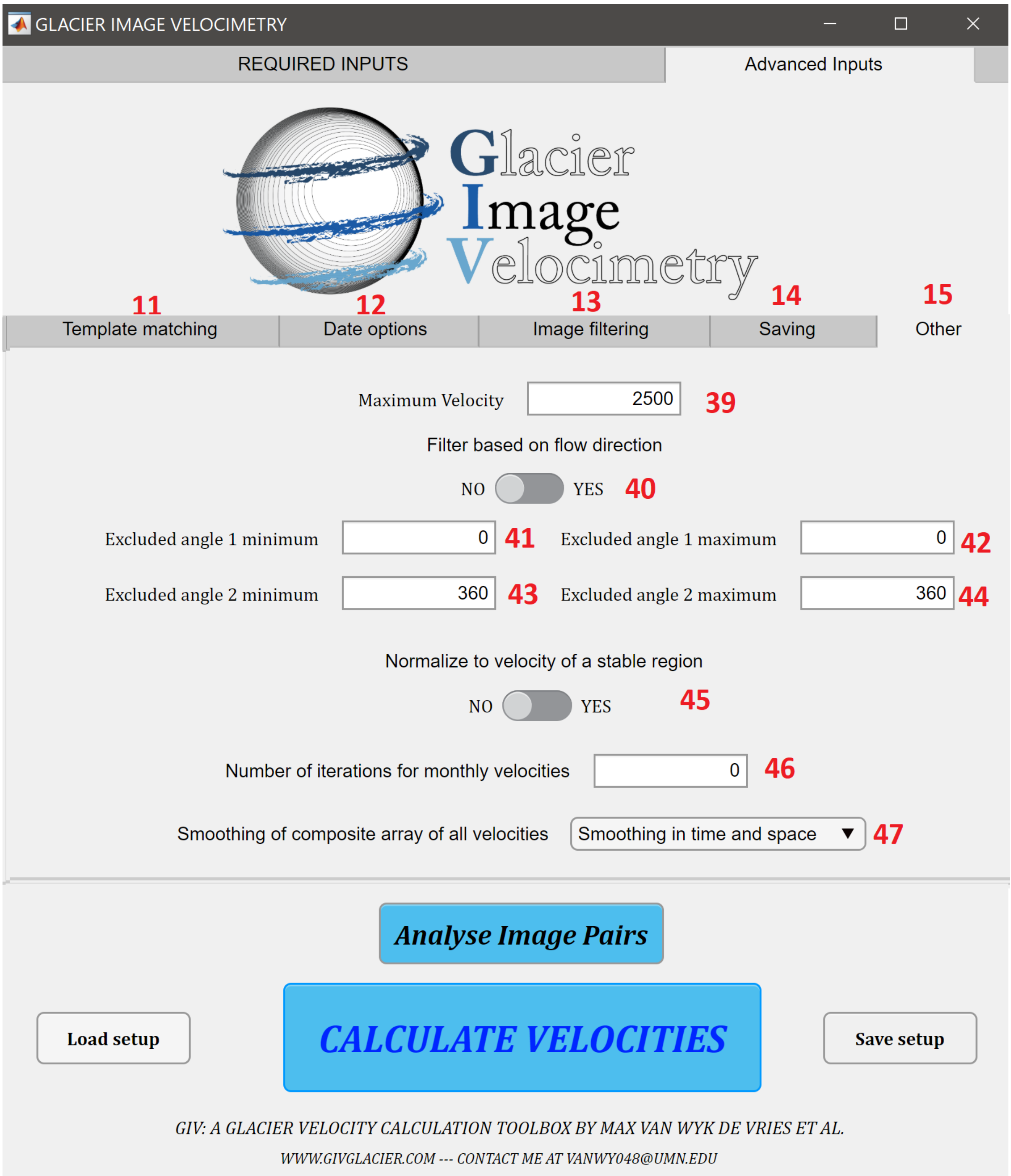
Choose the format of output images. A range of common formats may be chosen, including .jpg, .png, .tiff, .pdf and more.

38Image filtering options: Save georeferenced velocities

This option gives you the choice of saving or not saving georeferenced rasters of the velocities. The default is to save them.

39 Other: Maximum Velocity

Input a maximum velocity expected for these glaciers. Measured displacements larger than this will be considered outliers and excluded.



40 Other: Flow direction filter?

Do you want to exclude velocities based on their flow directions? If yes, inputs 41-44 will be used to define directions ice is known to not flow in. This may be used for instance where a glacier is known to flow due south, and pixels flowing N are known to be outliers.

41-42 Other: Excluded angle 1 min and max

Minimum and maximum for the first angle excluded (000 represents N, 090 represents E and so on). If the minimum is 0 and the maximum is 180, this will exclude all velocities flowing to the E. Leave both at 000 to not exclude any angle.

43-44 Other: Excluded angle 2 min and max

Minimum and maximum for the first angle excluded (000 represents N, 090 represents E and so on). If the minimum is 0 and the maximum is 180, this will exclude all velocities flowing to the E. Leave both at 360 to not exclude any angle.

45 Other: Stable region correction

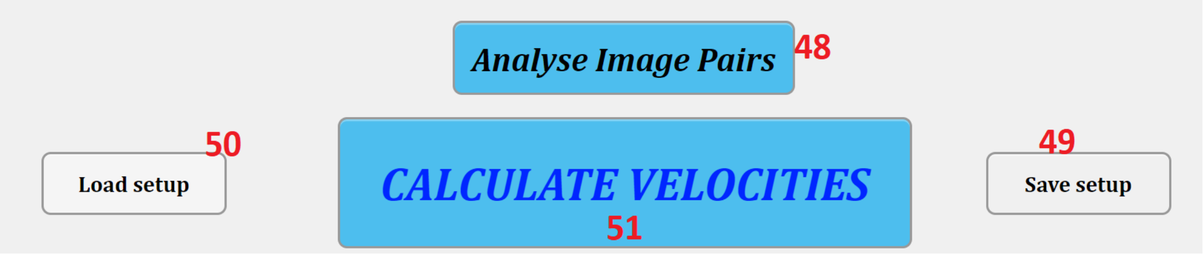
Do you wish to normalize the velocities to those of a stable region (e.g. bedrock)? This is particularly useful for slow moving glaciers where small georeferencing errors may be important. This requires the 'stable' file to be in the same folder as the other images.

46 Other: Iterations for monthly velocities

Number of iterations for the monthly velocity averaging script. 0 will calculate a simple average of all velocities covering that time period, higher numbers will attempt to correct for velocities covering several months. 5 to 10 iterations should generally be sufficient.

47 Other: Smoothing of all velocity maps

The combined array of all velocity maps may be smoothed, either in space or in space and time (e.g. values from the neighbouring times may be used to infill missing values). They may also not be smoothed.



48 ANALYSE IMAGE PAIRS

Script that calculates the number of valid image pairs for given input parameters. Thus may be run to help decide which 'temporal oversampling value' (6) is used.

49 SAVE SETUP

Save a given input parameter set-up. This parameter set-up can then be automatically loaded and re-run at any time in the future.

50 LOAD SETUP

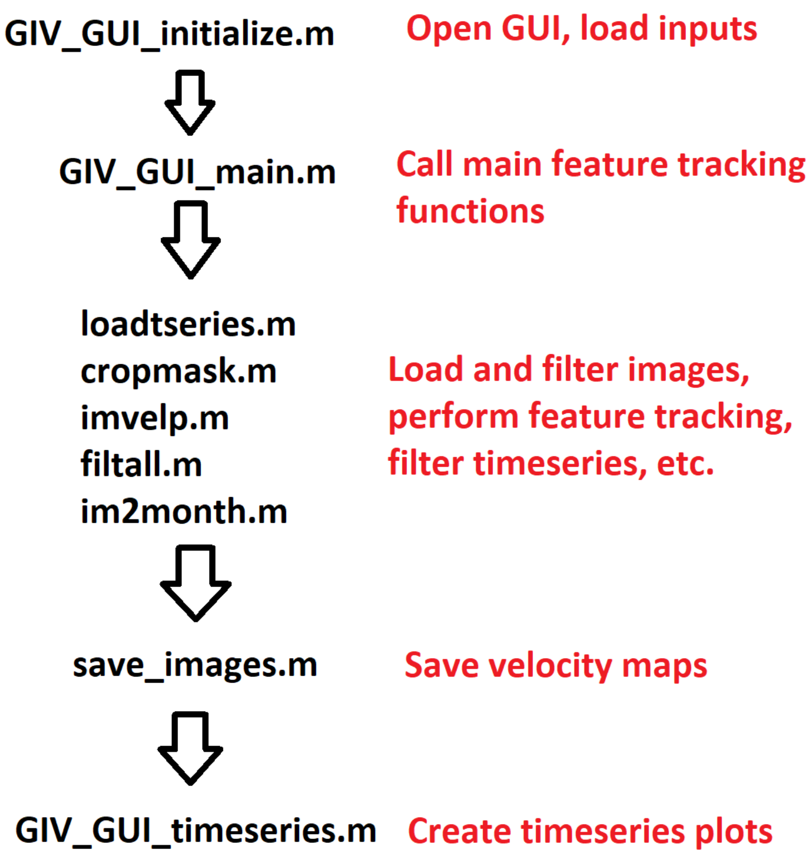
Load and run a previously saved input file.

51 CALCULATE VELOCITIES

This button initiates the main GIV script and feature tracking algorithm. Press this once you have adjusted all input parameters to your needs.

5 MATLAB FUNCTIONS AND CODE

This function describes the main GIV functions and MATLAB files included in this toolbox. The relationship between these functions is described in the following flowchart:



GIV_GUI_initialize.m

This function simply launches the graphical user interface, and feeds the inputs from this interface into GIV_GUI_main.m

GIV_GUI_main.m

This function is a wrapper from which the other functions are loaded. It runs through the workflow of this toolbox: loading the imagery, filtering the images, calculating the displacement between pairs, filtering and saving the results.

loadtseries.m

This function loads a series of .png or .jpg images into a single cell array, and extracts the metadata from the images titles (in yyyymmdd format). This array, named '*images*' is then passed through all functions in this toolbox.

cropmask.m

This function converts the initial array of .png or .jpg to pixel intensity maps by summing the RGB values. It then crops these images to the user inputted mask, and performs a series of image pre-filtering steps.

Included functions: imageprefilter.m, LaplaceFilter.m, SobelFilter.m

imvelp.m

This function performs the feature tracking on all image pairs. It is set up to loop through the different image pairs, and uses a slightly different loop architecture depending on whether the run is parallelised or not. These loops will call on different functions to perform single pass, multipass feature tracking and a number of outlier detection and correction filters. This is the bulk of the toolbox, and will typically account for the majority of the computational time. Note that all functions have at least a brief description, and most are commented.

Included functions: fillnan.m, fillnan2.m, GIVtrack.m, GIVtrackmulti.m, GIVtrackmultifinal.m, GIVtrackmultifirst.m, GIVtrackmultipeak.m, neighbourfilter.m, xcorrelate.m, coordtom.m, filtall.m, myfilter.m, nanfillsm.m, smooth_snr.m, xytoV.m

save_raw_array.m

This function saves the raw '*images*' array at various stages in the program, includeing after running imvelp.m. This allows the raw velocity maps to be filtered differently without needing to be recalculated, and prevents data loss in case of incorrect inputs in the filtering functions.

filtall.m

This function filters the dataset of velocity maps calculated by imvelp.m. It detects values that differ significantly from the median velocity and flow direction, excludes extreme values and interpolates over newly created gaps.

Included functions: filtall.m, myfilter.m, nanfillsm.m, nanfill_time.m, nanfill_timeandspace.m

im2month.m

This function splices the raw ice velocity dataset into a monthly timeseries. This may be done via simple averaging of all velocities covering a certain time period, or via iterative portioning of all velocities crossing the boundary between months.

save_images.m

This function saves the raw MATLAB input and output files, as well as various representations of the velocity maps. Depending on user inputs, .png images of the maps and georeferenced rasters may also be generated.

GIV_GUI_timeseries.m

This function is automatically called once a run of GIV is finished, and may be called at any point later in time. It post-processes the velocity maps and generates a timeseries of velocity at a given point in space.

Included functions: velocitytimeplot.m

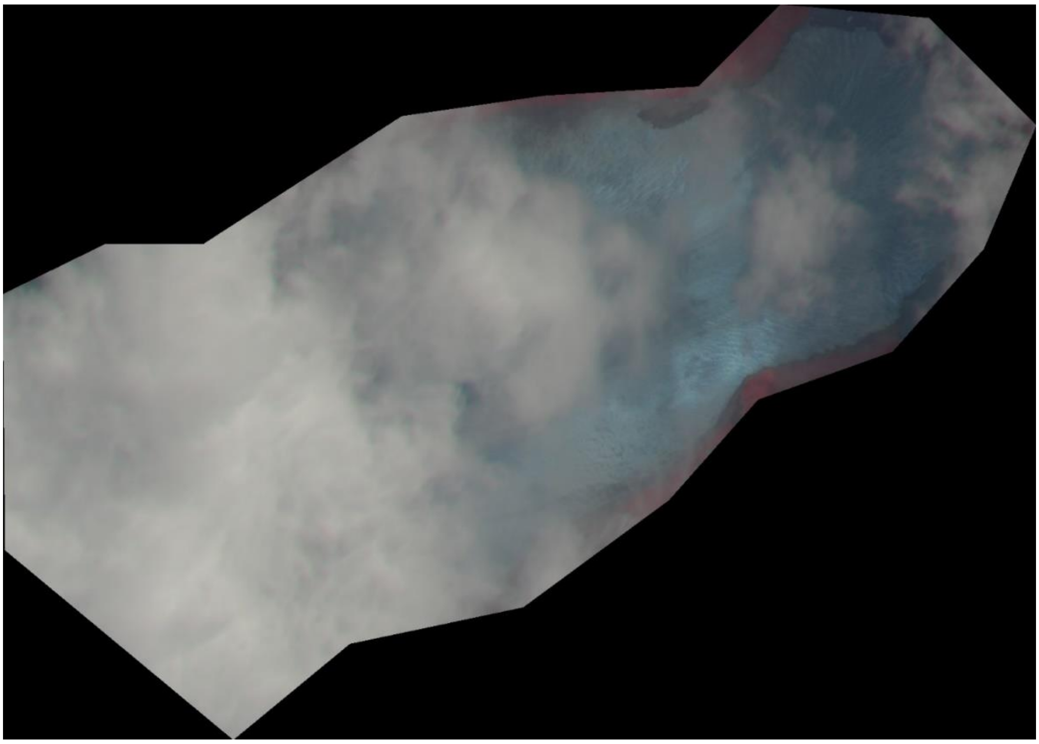
GIVruntime.m

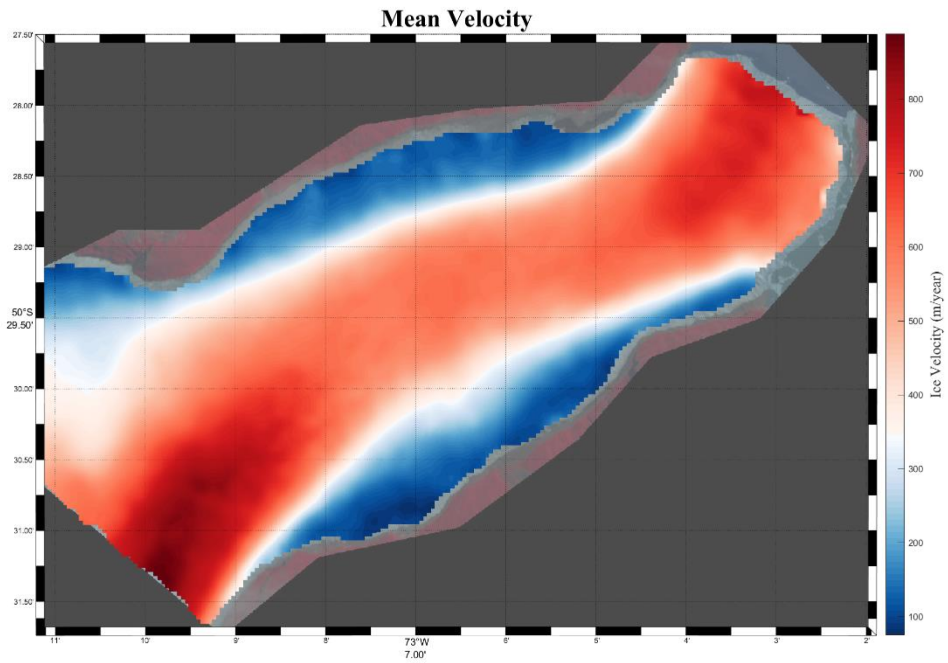
This function is called when the user presses the 'Analyse image pairs' button on the main GUI. It calculates the number of valid image pairs for a given set of input parameters.

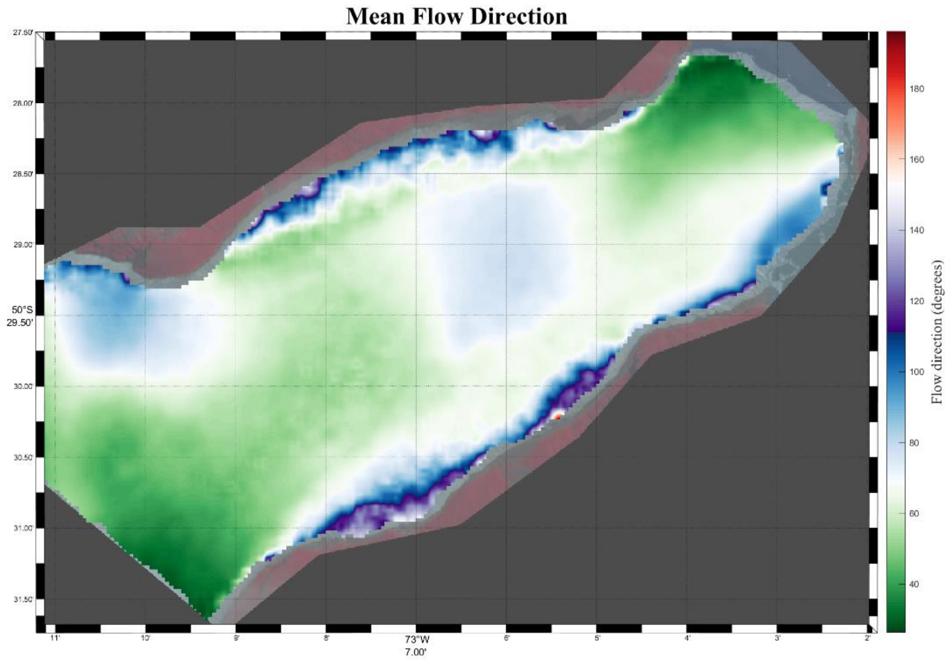
A variety of sub-functions are included that perform duties such as converting x and y displacements to magnitude and direction of displacement, detecting and extrapolating outliers and more.

6 EXAMPLE: GLACIAR PERITO MORENO

Glaciar Perito Moreno is an iconic glacier in Southern Patagonia, famous for periodically damming a branch of the lake it terminates in. As a large, heavily crevassed and fast flowing glacier, it is an ideal location to test GIV. An image folder named ‘Perito_Moreno’ is included in with GIV’s folder. The folder contains 23 .jpg Sentinel 2 images covering the period of Jan-Mar 2020, a mask file and a save_image file. The dataset contains a number of imperfect images, including heavily clowded and shadowed images (see below). Follow the instructions listed in section 3 to run GIV on this dataset using the default parameters. Set the temporal oversampling value to 10, set the coordinates (min lat = -50.964, max lat = -50.892; min long = -73.746, max long = -73.473) , and sit back and let GIV calculate velocity maps. The velocity map images generated can be found in the folder Perito_Moreno/Results/[*name you inputted*]/ Data Figures (Images)/ Mean, Standard Deviation and other statistics. The mean velocity and mean flow direction plots should look similar to those shown on the next two pages. If not, try double checking your input parameters (see section 4 if needed). You are now ready to download your own imagery timeseries and calculate velocity fields for a different glacier!




Mean Velocity
27.50'
28.00'
28.50'
29.00'
50°S
29.50'
30.00'
30.50'
31.00'
31.50'
11'
10'
9'
8'
73°W
7.00'
6'
5'
4'
3'
2'
800
700
600
500
400
300
200
100
Ice Velocity (m/year)


Mean Flow Direction
27.50'
28.00'
28.50'
29.00'
50°S
29.50'
30.00'
30.50'
31.00'
31.50'
11'
10'
9'
8'
73°W
7.00'
6'
5'
4'
3'
2'
180
160
140
120
100
80
60
40
Flow direction (degrees)

7 FURTHER REMARKS AND CREDITS

Operating systems: GIV has been tested on Mac, Windows and Linux. At the present day, only the Windows version is available as a precompiled app, although versions for different operating systems will be generated if there is demand.

Running GIV as a standalone app requires the prior download of MATLAB runtime, that may be found at the following link: https://www.mathworks.com/products/compiler/matlab-runtime.html. MATLAB runtime is free, although the file size is relatively large (~1.5GB).

Credit is given to a number of different open PIV and mapping toolboxes that helped contribute to this work: IMGRAFT (Messerli and Grinstead, 2015), PIVlab (Thielicke and Stamhuis, 2014), matpiv (Sveen, 2004), M_map (Pawlowicz, 2020) and cbrewer (Charles, 2020). The acknowledgements are given in each function where appropriate as well.

Ben Popken contributed to the early stages of GIV's design and testing on Upsala glacier in Patagonia. Michele Guala from the University of Minnesota's St Anthony Falls lab provided advice on particle image velocimetry tools used in fluid dynamics research. Emi Ito, Kelly MacGregor, Shanti Penprase, Matias Romero, and Jabari Jones provided comments on early drafts of this work and associated manuscript.