

Fast Correlation Technique for Glacier Flow Monitoring

by Digital Camera and Space-borne SAR Images

Flavien Vernier* Renaud Fallourd*[†] Jean Michel Friedt[‡] Emmanuel Trouvé*
Jean-Marie Nicolas[†] Luc Moreau[§]

Submitted to **EURASIP Journal on Image and Video Processing**, special issue on “**Remote and Proximal Sensing of the Environment for the Detection and Monitoring of Natural Risks**”

Abstract

Most of the image processing techniques have been first proposed and developed on small size images and progressively applied to larger and larger data sets resulting from new sensors and application requirements. In geosciences, the monitoring of glacier evolution by image analysis is expected to provide useful information regarding the local impact of global changes and associated risks in the surrounding areas. Digital cameras or remote sensing images can be used to measure the glacier surface velocity by different techniques. However, the image size and the number of acquisitions to be processed to analyze time series become a critical issue to derive displacement fields by the conventional correlation technique. A fast correlation computation implies an optimization study of a correlation algorithm. This paper describes the mathematical optimization of the classical normalized cross-correlation and its implementation. This implementation overcomes the computation time and window size limitations. To reduce the computation time, the algorithm is studied so as to eliminate most of the temporary result re-computations. Consequently, the proposed implementation is performed with a specific memory management. The software resulting of this optimization is used to compute the displacement between two optical images of a serac fall and between two Synthetic Aperture Radar (SAR) images of Alps glaciers. The optical images are acquired by proximal sensing – a digital camera installed near a serac fall – and the SAR images are acquired by remote sensing: TerraSAR-X satellite. The results illustrate the potential of this implementation to derive dense displacement fields with a computational time compatible with the camera operating around the Argentière glacier and the size of the 2 m resolution TerraSAR-X scenes covering $30 \times 50 \text{ km}^2$. The computation results highlight the movement of the glacier and the speedup of the optimization.

*Université de Savoie - Polytech Annecy-Chambéry - LISTIC, BP 80439, 74944, Annecy-le-Vieux cedex, France. Email: (flavien.vernier,renaud.fallourd, emmanuel.trouve)@univ-savoie.fr

[†]Institut TELECOM, TELECOM ParisTech, CNRS LTCI, 75013, Paris, France. Email: jean-marie.nicolas@telecom-paristech.fr

[‡]Institut FEMTO-ST, Département LPMO, 25044, Besançon, France. Email: jean-michel.friedt@femto-st.fr

[§]EDYTEM, CNRS Université de Savoie, F-73376, Le Bourget du Lac, France. Email: moreauluc@club-internet.fr

1 Introduction

In the last decades, the warmer climate together with less precipitation in the glacial accumulation areas has resulted in a spectacular retreat of most of the monitored temperate glaciers [1]. If confirmed in the coming years, this evolution will have important consequences in terms of water resources, economical development and risk management in the surrounding areas. To monitor glacier displacements and surface evolutions, two main sources of information are available:

- in-situ data collected for instance by using accumulation/ablation stakes, GPS stations, or digital cameras installed near the glaciers to acquire regular images of specific areas such as serac falls, unstable moraines...
- remote sensing data acquired by airborne or space-borne sensors such as multispectral optical images or Synthetic Aperture Radar (SAR) images.

Both sources are complementary: in situ data are usually more accurate and specifically produced for glacier observation, but very few temperate glaciers are monitored by ground measurements because of the cost, the access difficulty and the risks associated to ground missions in high mountain areas where alpine glaciers are usually located. Remotely sensed data have the advantage of a more global observation potential ($60 \times 60 \text{ km}^2$ in SPOT images for instance), but they are more dependent on the sensor/satellite availability: several days often separate repeat passes and conflicts between requested acquisitions may occur. Those images, especially SAR data also require complex processing chains to derive the sought-after information.

Optical data sets are often used to observe changes and allow the computation of high resolution (HR) information such as the surface elevation or glacier displacement fields during the summer [2, 3], but they cannot be regularly acquired along the year and efficiently used because of clouds or snow cover uniformity. Space-borne SAR data and especially the recently launched HR satellites such as TerraSAR-X, COSMO-SkyMed or Radarsat-2, are a new source of information which may allow global evolution monitoring and provide regular measurements thanks to the all-weather capabilities of SAR imagery. They are used to derive surface changes and velocity fields [4], or to detect and track rocks and crevasses [5]. To monitor geophysical phenomena such as glaciers or volcanos, it is necessary to acquire series of images at different time (along the year for slow process, up to several images per day for fast moving areas). The so called "image time series" allow to derive the evolution parameters by image processing techniques.

With the increase of the sensor spatial resolution, the data transmission and storage possibilities, the use of image time series for Earth observation is facing computational challenges which can be separated in two groups: the need to develop new signal/image processing methods to extract information from huge amount of data (by information fusion or data mining approaches for instance), but also the need to improve existing robust techniques applied at the early processing stages in order to be able to apply them in a reasonable computation time on very large images (often more

than 20000×20000 pixels for HR satellite data) and on large number of images to explore temporal evolution. Image co-registration is one of the first tasks to be performed to handle time series of images acquired by a sensor in similar conditions. The difficulty depends on the sensor and acquisition configuration, and also on the amount of change between subsequent images. When motion-free areas and moving features can be distinguished, this co-registration stage also provides displacement information which is useful to derive surface displacement fields. This task is often performed by the well known correlation technique, which can be applied in different ways depending on window sizes and density of the points to correlate between images.

Several tools have been developed to solve the classical correlation problem. For optical imagery, a software like COSI-Corr (Co-Registration of Optically Sensed Images and Correlation) [6, 7] is widely used in the geoscience community, but it is not developed for fast computation. Due to its integration to ENVI, COSI-Corr is easy to use and offers classical and Fast Fourier Transform (FFT) techniques to compute correlation. However, its use for large images is limited by computation time. For SAR imagery, accurate co-registration is necessary for interferometric applications (InSAR) which are based on the phase difference between co-registered complex images. The well-known software called ROI-PAC (Repeat Orbit Interferometry Package) [8] is not dedicated to the correlation problem but it offers a solution for this problem. In the case of ROI-PAC, a two steps strategy has been adopted: a first global co-registration of the two images on a sparse grid, followed by the refined computation of the correlation on a regular grid. Also, the disadvantage of ROI-PAC is the computation time that can dramatically increase with the image size and the number of correlation points in the image.

There are many different techniques developed for image co-registration [9]. Those based on sub-image correlation operate either in the temporal domain (the spatial domain for the image 2 dimensional (2D) signal) by directly computing the values of the cross-correlation function and searching for its peak, or in the spectral domain after the computation of the discrete Fourier transform of the two sub-images. The methods developed in the spectral domain are meant to speedup the computation by using the FFT algorithm proposed with optimized implementation in signal/image processing libraries [10]. They derive the sub-image shift either from the phase of the cross-spectrum [11], or by computing its inverse Fourier transform and identifying the correlation peak in the spatial domain [12]. A basic computation of the cross-correlation in the spatial domain requires $N \times M$ products between N samples of the sub-images for M positions where the cross-correlation is estimated. By taking $N = M$, the number of operations is proportional to N^2 . Whereas with an implementation in the spectral domain, it is proportional to $N \log N$. A speedup of the process is expected when the window size increases, with the constraint of being a power of 2 in both directions in order to benefit of the FFT optimizations.

Compared to the conventional implementation of the correlation in the spatial domain, the benefit of the spectral approach depends on the window sizes. An efficient implementation in the spatial domain also presents some

advantages. It is more flexible since there is no constraint on window sizes, which allows to take the limitation of the local stationarity hypothesis into account. It has also the advantage of being more generic since it allows the choice of different similarity criteria according to the statistics of the images. Several alternatives to the conventional “cross-correlation function” have been proposed for image co-registration [13], especially in the case of SAR images which are affected by the speckle effect for distributed targets. The properties of the “true correlation function” in the Fourier domain cannot be transposed for more complex criteria derived for instance from a maximum likelihood approach [14, 15].

In this paper, an implementation strategy of the correlation function in the spatial domain is proposed. The objective is to preserve the flexibility and the genericness of the spatial domain approach, and to benefit from the computation efficiency of parallel or distributed processing architectures which become more and more common on conventional computers. The originality of this approach is to be able to efficiently compute the disparity measure at the initial resolution and to derive a dense displacement field. To our knowledge, no tool exists for such fast computation over large remote sensing (or proximal sensing?????????) images. In our point of view, this kind of tool is essential (important — crucial ??????) to manage the new data sets from HR sensors, time series and large scenes. The potential and the performances of this approach are illustrated on two kinds of data: remote sensing data with repeat pass acquisitions of HR TerraSAR-X images over fast moving glaciers in the Alps, and proximal sensing image time series from a digital camera installed in front of a serac fall of the Argentière glacier in the Mont-Blanc area.

This paper is organized as follows: Section 2 details the Normalized Cross-Correlation (NCC) algorithm, its optimization and its implementation, so as to obtain an efficient correlation software. In the next sections, Sections 3 and 4, the correlation software is applied to a realistic problem. Section 3 is dedicated to the computation of the displacement of serac falls in front of the Argentière glacier. The results show a set of serac displacements and highlights the impacts of the optimized software. Section 4 illustrates the computation of glacier flow by correlation of SAR images. This section confirms the results obtained with optical images and shows the impact of master window size on the computation time. Finally, Section 5 concludes this paper and projects future work.

2 Implementation Techniques for Fast Correlation

2.1 Similarity Function

The objective of the correlation consists in finding the best match between sub-images, a slave image I' compared to a master image I . To simplify the algorithms in this paper, the sizes of images I and I' are the same and given by the number of rows I_r and the number of columns I_c . Figure 1 illustrates the algorithm and the chosen notations.

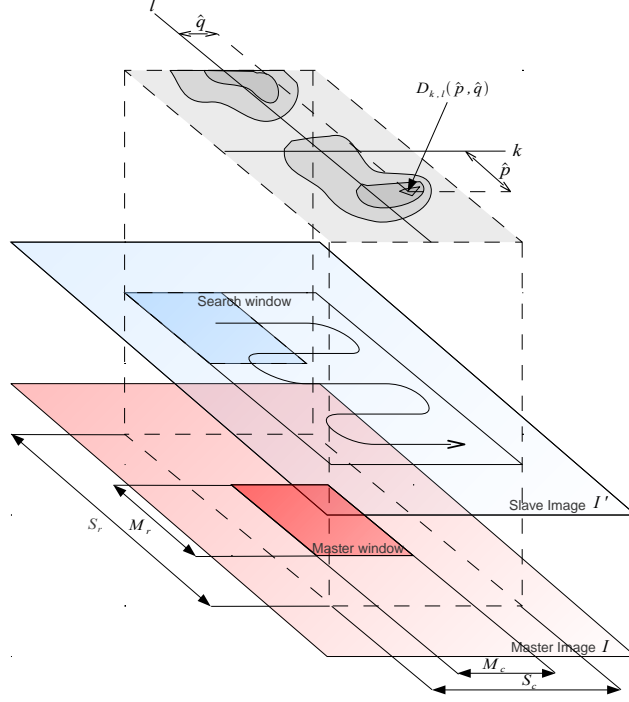


Figure 1: Schematic illustration of the correlation algorithm with used notations.

The master window M is defined by its size $M_r \times M_c$, where M_r and M_c are respectively the number of rows and the number of columns. Like the master window, the search window S in the slave image is defined by its number of rows S_r and its number of columns S_c . To simplify the notations and to make the presentation easier, M_r , M_c , S_r and S_c are odd. In this manner, the correlation objective is finding for each point (k, l) of the master image, the best position of the window $M_{k,l}$ centered on (k, l) in $S_{k,l}$, according to a similarity function $D(p, q)$, where p and q are the displacements of $M_{k,l}$ in $S_{k,l}$. The search window definition implies that $S_r \geq M_r$ and $S_c \geq M_c$. The best position (\hat{p}, \hat{q}) is defined by the maximum of the similarity function for a given couple $(M_{k,l}, S_{k,l})$:

$$D_{k,l}(\hat{p}, \hat{q}) = \max_{p,q} (D_{k,l}(p, q)) \quad (1)$$

As $M_{k,l}$ and $S_{k,l}$ always depend on the position (k, l) , they will be denoted by M and S respectively in the rest of this paper (thereafter????????????). The similarity function $D(p, q)$ is not fixed and depends on the user needs. In this paper, the classical NCC defined by Equations 2, 3 and 4 is used.

$$D_{k,l}(p, q) = \frac{\sum_{i,j} P_{k,l}(p, q, i, j)}{N_{k,l}(p, q)}, \quad (2)$$

where

$$P_{k,l}(p, q, i, j) = M(i, j) \cdot S(i - p, j - q), \quad (3)$$

and

$$N_{k,l}(p, q) = \sqrt{\sum_{i,j} M(i, j)^2 \cdot \sum_{i,j} S(i - p, j - q)^2}. \quad (4)$$

The correlation result is the computation of (\hat{p}, \hat{q}) for all (k, l) such that $\frac{S_r}{2} \leq k \leq I_r - \frac{S_r}{2} + 1$ and $\frac{S_c}{2} \leq l \leq I_c - \frac{S_c}{2} + 1$. Thus, for each point, the result is defined by \hat{p} , \hat{q} and $D_{k,l}(\hat{p}, \hat{q})$. The value of \hat{p} , \hat{q} are respectively the displacement on the lines and the displacement on the columns of the point (k, l) , and $D_{k,l}(\hat{p}, \hat{q})$ is the cross-correlation level for these displacements, which varies between 0 and 1.

2.2 Optimized Algorithm

To optimize the algorithm and reduce the computation time, the correlation algorithm must be rewritten to highlight the computation dependencies. The first objective is to avoid re-computing an already computed value. The second one is to introduce a flow computation technique to reduce the number of operations of the algorithm: like a rolling average where the sum of values is never recomputed but the outgoing value is subtracted and the incoming value is added to the previous sum. According to these points the correlation equations given in Section 2.1 can be rewritten as follows:

For a given master point k, l :

$$D_{k,l}(p, q) = \frac{U_{k,l}(p, q)}{\sqrt{V_{k,l} \times W_{k,l}(p, q)}},$$

where the computation of U , V and W can depend on their previous computation. For the first master point (k_0, l_0) given by $(\frac{S_r}{2}, \frac{S_c}{2})$, no optimization can be used, thus U , V and W are computed according to Equations 2, 3 and 4:

$$\begin{aligned} U_{k,l}(p, q) &= \sum_{i,j} M(i, j) \cdot S(i - p, j - q), \\ V_{k,l} &= \sum_{i,j} M(i, j)^2, \\ W_{k,l}(p, q) &= \sum_{i,j} S(i - p, j - q)^2. \end{aligned}$$

For the points (k, l) such that $k \neq k_0$ or $l \neq l_0$, the values of U , V and W can be expressed depending on the previous point $(k - 1, l)$ or $(k, l - 1)$. If the point (k, l) is not on the first column - $(l \neq l_0)$ - U can be computed like Equation 5,

$$\begin{aligned} U_{k,l}(p, q) &= U_{k,l-1}(p, q) \\ &\quad - \sum_i M(i, j_0 - 1) \cdot S(i - p, j_0 - 1 - q) \\ &\quad + \sum_i M(i, j_n) \cdot S(i - p, j_n - q), \end{aligned} \quad (5)$$

or depending on the point on the previous line if $k \neq k_0$:

$$\begin{aligned}
U_{k,l}(p, q) = & U_{k-1,l}(p, q) \\
& - \sum_j M(i_0 - 1, j) \cdot S(i_0 - 1 - p, j - q) \\
& + \sum_j M(i_n, j) \cdot S(i_n - p, j - q).
\end{aligned} \tag{6}$$

Let us note that j_0 and j_n are respectively the indices, in the master image, of the first and the last column of the current master window. j_0 and j_n are given by $l - \frac{M_c}{2}$ and $l + \frac{M_c}{2}$. The indices of the first and the last line are given by $i_0 = k - \frac{M_r}{2}$ and $i_n = k + \frac{M_r}{2}$. The values of $V_{k,l}$ and $W_{k,l}(p, q)$ can be computed in the same way. If $l \neq l_0$:

$$\begin{aligned}
V_{k,l} = & V_{k,l-1} \\
& - \mathcal{M}_{k,l}(j_0 - 1) \\
& + \mathcal{M}_{k,l}(j_n),
\end{aligned} \tag{7}$$

where

$$\mathcal{M}_{k,l}(j) = \begin{cases} \sum_i M2_{k,l}(i, j) & \text{if } j = M_c \vee (k = k_0 \wedge l = l_0), \\ \mathcal{M}_{k,l-1}(j) & \text{otherwise,} \end{cases} \tag{8}$$

and

$$M2_{k,l}(i, j) = \begin{cases} M(i, j)^2 & \text{if } j = M_c \vee (k = k_0 \wedge l = l_0), \\ M2_{k,l-1}(i, j) & \text{otherwise.} \end{cases} \tag{9}$$

Respectively for $W_{k,l}(p, q)$:

$$\begin{aligned}
W_{k,l}(p, q) = & W_{k,l-1}(p, q) \\
& - \mathcal{S}_{k,l}(p, q, j_0 - 1) \\
& + \mathcal{S}_{k,l}(p, q, j_n),
\end{aligned} \tag{10}$$

where

$$\mathcal{S}_{k,l}(p, q, j) = \begin{cases} \sum_i S2(p, q, i, j) & \text{if } j = M_c \vee (k = k_0 \wedge l = l_0), \\ \mathcal{S}_{k,l-1}(p, q, i, j) & \text{otherwise.} \end{cases} \tag{11}$$

and

$$S2_{k,l}(p, q, i, j) = \begin{cases} S(i - p, j - q)^2 & \text{if } j = M_c \vee (k = k_0 \wedge l = l_0), \\ S2_{k,l-1}(p, q, i, j) & \text{otherwise.} \end{cases} \tag{12}$$

If $k \neq k_0$:

$$\begin{aligned}
V_{k,l} = & V_{k-1,l} \\
& - \mathcal{M}_{k,l}(i_0 - 1) \\
& + \mathcal{M}_{k,l}(i_n),
\end{aligned} \tag{13}$$

where

$$\mathcal{M}_{k,l}(i) = \begin{cases} \sum_j M2_{k,l}(i, j) & \text{if } i = M_r \vee (k = k_0 \wedge l = l_0), \\ \mathcal{M}_{k-1,l}(i) & \text{otherwise,} \end{cases} \quad (14)$$

and

$$M2_{k,l}(i, j) = \begin{cases} M(i, j)^2 & \text{if } i = M_r \vee (k = k_0 \wedge l = l_0), \\ M2_{k-1,l}(i, j) & \text{otherwise.} \end{cases} \quad (15)$$

Respectively for $W_{k,l}(p, q)$:

$$\begin{aligned} W_{k,l}(p, q) = & W_{k-1,l}(p, q) \\ & -\mathcal{S}_{k,l}(p, q, i_0 - 1) \\ & +\mathcal{S}_{k,l}(p, q, i_n), \end{aligned} \quad (16)$$

where

$$\mathcal{S}_{k,l}(p, q, i) = \begin{cases} \sum_j S2(p, q, i, j) & \text{if } i = M_r \vee (k = k_0 \wedge l = l_0), \\ \mathcal{S}_{k-1,l}(p, q, i) & \text{otherwise,} \end{cases} \quad (17)$$

and

$$S2_{k,l}(p, q, i, j) = \begin{cases} S(i - p, j - q)^2 & \text{if } i = M_r \vee (k = k_0 \wedge l = l_0), \\ S2_{k-1,l}(p, q, i, j) & \text{otherwise.} \end{cases} \quad (18)$$

Let us note that this optimization strongly reduces the number of operations compared to a naive implementation. As the number of operations is one of the most critical criteria for the efficiency, the correlation algorithm must be implemented according to this optimization. The optimization is not the unique critical point to reduce the computation time, the implementation is another one.

2.3 Implementation

For the implementation, one of the main problem is the memory to be used. The input and output images can be too big to be stored in the memory, and hard drive access can be very time consuming. Moreover, the optimizations presented in Section 2.2 need memory to store the pre-computed values. Thus, an important point is to manage the required memory according to the available memory to execute the correlation algorithm as fast as possible.

The common point about the optimization given by Equations 8 to 18 is the use of rolling vectors or rolling matrices. A rolling vector is a vector of $N + k$ elements where N is the common size of the vector and k the number of slide steps. At each slide step, the head of the vector is increased by 1 and only the last element is re-computed.

$$\begin{array}{cccc|cccc} 1 & 2 & 3 & 4 & - & - & - & - \\ \hline 1 & 2 & 3 & 4 & 5 & - & - & - \end{array}$$

To create a rolling matrix, a data vector of the size of the matrix plus the number of slide step is allocated. In the following example a 3×3 matrix is allocated and 3 slide steps are planned.

11	12	13	21	22	23	31	32	33	-	-	-
----	----	----	----	----	----	----	----	----	---	---	---

Next, each start point of lines is correctly placed to have the following matrix:

11	12	13			
21	22	23			
31	32	33	-	-	-

To slide the matrix on the right, each start point of lines is incremented by 1:

	11	12	13	21		
		22	23	31		
		32	33	-	-	-

After that, the last column can be recomputed:

	11	12	13	14		
		22	23	24		
		32	33	34	-	-

With a rolling matrix, only the last column is recomputed, this is equivalent to a condition to the re-computation of Equations from 9 to 18. Thus these equations are rolling matrices.

The optimizations presented in Section 2.2 give two approaches, one based on lines dependency and the other on columns dependency. Both approaches are necessary. In our case, a point that is not on the first column is computed depending on the point on the previous column. A point that is on the first column, except on the first line, is computed depending on the previous line. In this way, the memory corresponding to the pre-computation of two points must be allocated, one for the next point on the same line and one to start the next line.

The input/output required memory depends on the size of the couple of input images. In most cases, the actual computers have enough memory to store all the computations, but sometimes, with the huge images used (more than 4GB per image), the available memory can be insufficient. That is why the implementation of the algorithm must manage the computation lines block. This kind of implementation has two advantages. Firstly, it allows the distribution of the algorithm. If N CPU are available, the images can be split in N blocks and each CPU computes the correlation on its block. Secondly, if on a machine there is not enough memory to compute the correlation, the implementation computes on a first block that can be stored in the memory, saves the results and then computes

the next block, and so on. This approach can be realized due to the fact that the needed memory for each part of the algorithm can be predicted according to the previous optimizations. Any Equations from 9 to 18 depending on j require a rolling vector of M_c elements plus $I_c - S_c$ slide steps, depending on i, j require a rolling matrix of $M_r \times M_c$ elements plus $I_c - S_c$ slide steps, depending on p, q, j require $S_r \times S_c$ rolling vectors of M_c elements plus $I_c - S_c$ slide steps, and so on.

This optimized implementation is available in the EFIDIR Tools under GPL License. These tools can be download from the EFIDIR web site (see Acknowledgment).

3 Experiments and results on digital video camera images

In this section, the performances of the implementation proposed in Section 2 are assessed and illustrated on the processing of optical images from a digital camera installed for glacier monitoring. In the literature, two types of camera have been used to measure glacier flow: the analog and the digital cameras. Initially, the traditional analog technology has been used [16, 17, 18, 19]. At the beginning of the 21st century, digital photography development has made the glacier flow monitoring with HR digital camera possible. Up to now, only few experiments have been reported with HR digital cameras, as for example in Greenland polar glaciers [20, 21]. To our knowledge, no experiment on Alpine temperate glacier has been performed.

3.1 Digital camera data set

Since 2007, in the framework of the project ANR (French National Research Agency) project Hydro-Sensor-FLOWS, HR automated digital cameras were developed and installed around the Mont-Blanc massif (see Table 1). In this paper, one of the Argentière cameras is used: the camera installed at 2300 *m* Above Sea Level (ASL) in the summer of 2008 which is focused on "Lognan serac falls" (see Figure 2).

Camera(s)	Location	Installation Date
2	Argentière glacier	autumn 2007 and summer 2008
2	Mer de Glace	summer 2008
1	Tacul glacier (the Géant seracs falls)	September 2008
1	Bionnassay glacier	summer 2009
1	Trient glacier	summer 2010

Table 1: Cameras installed around the Mont-Blanc massif.

The HR automated digital cameras installed around the Mont-Blanc massif are based on customized Leica DLux 3 and DLux4 units. These cameras were selected for their excellent optics, HR and off the shelf availability. They were heavily modified to allow a custom low-power microcontroller based board to control all basic functions, including

switching on and off the camera, focusing and triggering the shutter. A software-defined real time clock wakes up every second to increment a counter. When the user-defined alarm condition is met, the camera triggering sequence is started and a pre-defined amount of time (about 12 seconds) is provided for the camera to focus and grab the picture before power is switched off to save battery life. All functions provided by the camera manufacturer for operator handling are simulated through analog switches. A custom software allows for the user to define on the field the wake up hour, time interval between images and number of images taken every day. The default configuration is to wake up at 8 AM local time and grab 6 pictures every day, with 2 hours interval between images. Due to harsh environmental conditions, the local quartz oscillators are expected to drift within the manufacturer tolerance of ± 30 ppm in the operating range of $-20 \sim +70^\circ\text{C}$, *i.e.* a maximum drift of ± 15 min/yr.



Figure 2: Automated digital camera installed near the Argentière glacier in the summer of 2008.

The digital electronic control board was designed:

- to update most of the capabilities of the automated cameras on the field, *i.e.* 16:9 HR mode with 10 Mega pixels (image of 4224×2376 pixels),
- to acquire long time series without moving the setup and hence keeping the same field of view over time.

The digital camera takes a photograph of the glacier 6 times per day and stores images on a Secure Digital (SD) card, accessible to the user without breaking the seal keeping the camera dry. The “image time series” starts on the 3rd, Aug. 2008. The angle of view of the camera was calibrated to 65 degrees. The 10 Mega pixel mode generates pictures with a width of 4224 pixels so that the angle of view of a single pixel is 0.015 degrees (aperture angle).

3.2 Processing

All images are stored as HR JPEG images: this format was selected as a compromise between storage efficiency (since the cameras are running autonomously for up to 6 months without supervision) and data quality. The selected cameras do not provide raw format storage, which would have been too inefficient for our autonomous application. However, the JPEG format is not compatible with mono-band fast correlation approach presented in this paper.

Weather conditions are often extreme above 2000 *m* ASL in mountain areas such as the Alps. Wind and strong temperature variations might move the camera between two image acquisitions, as observed previously on a similar setup [17]. In such a case, a translation is observed between the two images. As an example of this effect, the image pair 2008-10-09/2008-10-10 exhibits a shift of 4 and -2 pixels respectively in the line and column directions.

According to these two previous points, the digital images acquired over "Lognan serac falls" are processed in 3 steps:

1. The initial JPEG images I_{jpeg} , i.e. Red-Green-Blue colour format, are converted in grayscale images I_{gray} to obtain mono-band images. This conversion is processed according to the following formula:

$$I_{gray} = 0.30 \times I_{jpeg}(Red) + 0.59 \times I_{jpeg}(Green) + 0.11 \times I_{jpeg}(Blue).$$

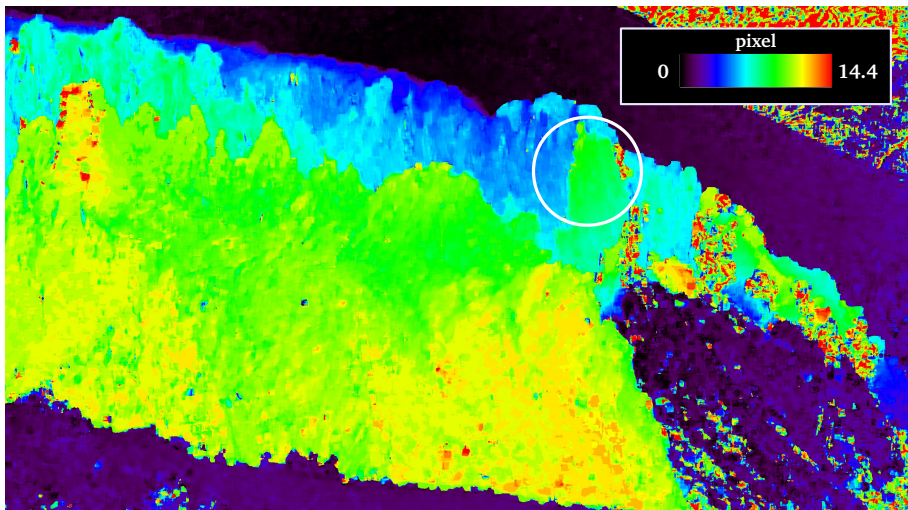
The resulting image is typically called luminance in digital image processing domain [22].

2. An initial co-registration between the images is made on the motion-free part of the images. In practice, the motion-free parts, i.e. mountains on the background, are used to perform it. This initial image co-registration on motion-free areas is realized by a translation without applying sub-pixel offsets.
3. The proposed fast correlation technique is applied on the image pair with a search window of 51×51 pixels, corresponding to the maximum offset visually observed. On motion-free areas, the sub-pixel offsets provide an accurate estimation of the remaining offset due to the camera instability. On the glacier, the measured offset is the sum of the displacement offset and the geometrical offset which has not been compensated for at step 2.

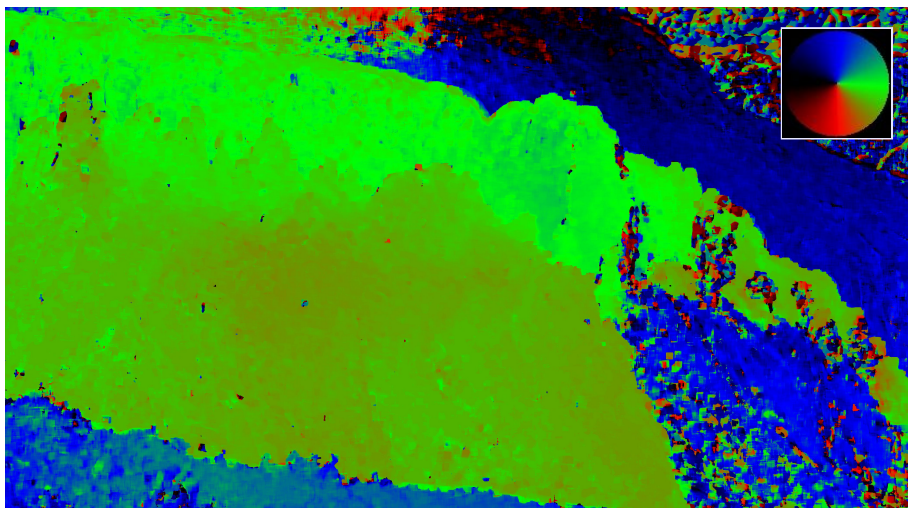
The correlation results obtained with 31×31 master window (i.e. $M_r \times M_c$) and 51×51 slave window (i.e. $S_r \times S_c$), on digital camera images are illustrated by the magnitude and the orientation of the pixel offset vector in Figure 3. The values close to zero (in black) on magnitude correspond to the motion-free parts around the Argentière glacier which are well co-registered by the initial translation (step 2). The areas in purple correspond to the motion-free parts where there are remaining offset variation due to the camera motion. In fact, the camera motion creates a stereo configuration between the two images with a very small baseline. Consequently, the obtained magnitude map on motion-free parts can be seen as a disparity map [23]. The glacier displacement appears with stronger magnitudes



(a) JPEG image 2008-10-09 (4224 × 2376 pixels).



(b) Magnitude of the 2D displacement measured in the two directions of image plane.



(c) Orientation of the 2D displacement (color reference on the top right corner).

Figure 3: Displacement field over the Argentière glacier derived by NCC between digital camera image pair (2008-10-09/2008-10-10)

in blue, green and yellow colors. The heterogeneity is due to either the glacier flow physics or the scene configuration: the nearest parts of the glacier appear to be flowing faster than the farthest parts. The displacement map of Figure 3-b highlights the differences between the ice blocks in the foreground (mostly in yellow), in the middle distance (mostly in green) and in the background (mostly in blue). One can notice a large ice bloc in green color on the right part of the blue background, corresponding to a larger displacement: this ice block is about to fall. There are also a few parts where the magnitude and the orientation maps look like noisy and inconsistent. These parts correspond to ice falls which happened between the two image acquisitions.

3.3 Computation Speedup

To highlight the effect of the optimization, the correlation is executed with and without optimization, using 1 to 8 CPU. The objectives of this execution are to illustrate the speedup given by the optimization and the number of used CPU. This experiment, and the following one, are computed on an octo-core Intel(R) Core(TM) i7 3GHz with 24Go memory. In the experiment, this machine is considered as 8 independent CPU with 3Go of memory by CPU. Figure 4 shows these results: the computation time without optimization (T) and with optimization (T_{opt}) depending on the optimization and the number of used CPU. A first observation of Figure 4 shows that the benefit of the optimization is

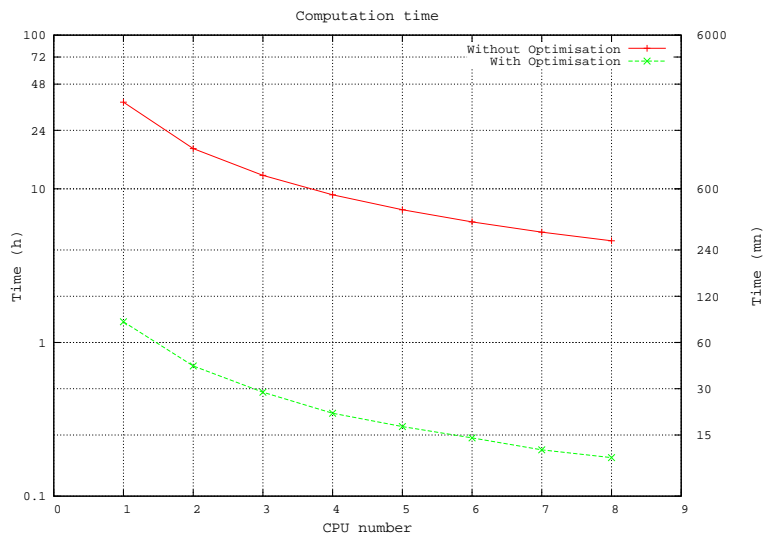


Figure 4: Computation time depending on the optimization and CPU number. (displayed with y – time – log scale)

very important. According to the delay between two image captures (2 h in our case), it is interesting to observe that the correlation can be computed between two captures only if the optimized method is used. Moreover, the impact of the number of used CPU gives an almost linear speedup. When the number of used CPU doubles, the computation

time is divided by two. Thus, the combination of optimization and distribution reduces the computation time, in our context from more than 36 hours to 10 minutes, or even less time if more than 8 CPU are used.

Figure 5 illustrates the gain obtained by the optimization. The first curve named “absolute gain” shows the difference of computation time without and with optimization ($T - T_{opt}$), for each number of used CPU. The second curve named “relative gain” shows the ratio between the previous curve and the computation time without optimization ($\frac{T - T_{opt}}{T}$). From Figure 5, the relative gain can be considered constant for our experiment and it is very significant: more than

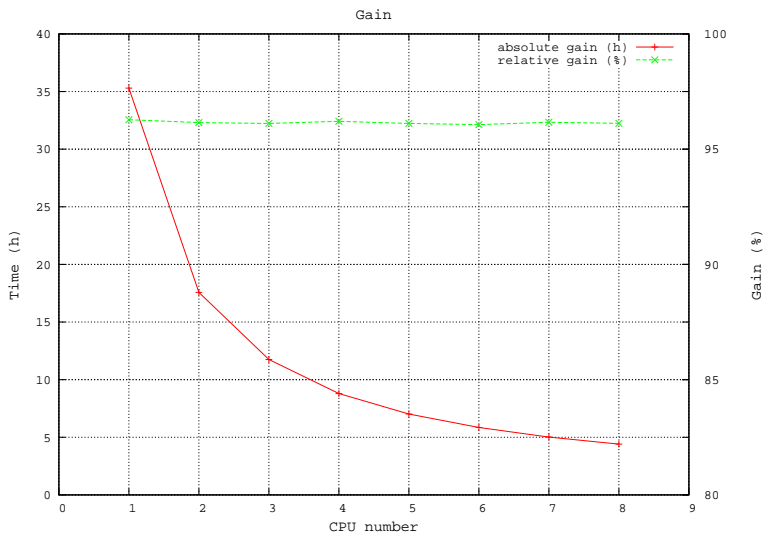


Figure 5: Gain given by the optimization depending on the CPU number.

96%. Since the computation without optimization can be very long - more than 1 day - the absolute gain can change the work habits. The prospects with many days of computation are not the same as with a few hours. The computation time and the absolute benefit decrease when the number of used CPU increases, but even with 8 CPU, several hours are saved thanks to the optimization.

This first experiment highlights the benefit of the optimization and the distribution of the correlation algorithm for optical images. It is important to note that this benefit permits to decrease the interval between two image captures. Consequently, a real time glacier flow monitoring can be projected. With the appropriate computation system, an acceleration of the glacier and an important loss of correlation corresponding to serac falls can be quickly detected.

4 Experiments and results on SAR images

Despite of improved acquisition, transmission and processing performances, the proximal sensing by ground based optical cameras, as illustrated in Section 3, is limited to specific parts of a few glaciers. In this section, the proposed fast correlation technique is applied to remote sensing data which can cover large areas: spaceborne images allow the whole glacier surface, and even all the glaciers of a mountain area, to be observed simultaneously. The feasibility in a reasonable computation time and the interest of the dense correlation measurements of this fast correlation technique are illustrated on HR SAR images which can be regularly acquired by satellite repeate passes.

4.1 TerraSAR-X data set

In the framework of the TerraSAR-X science project MTH0232 [24], 35 stripmap TerraSAR-X images have been acquired on the Mont-Blanc test site (see Table 2). There are 3 time series in descending configuration (orbit 25) and 1 in ascending configuration (orbit 154). Most images have been acquired in a single polarization mode (HH), except a winter series in the dual polarization mode (HH/HV) for the analysis of the snow backscattering properties. Ascending and descending measurements provide 4 different projections of the surface displacement which can be combined to retrieve the 3 components (Est, North, Up) of the 3D displacement field [25].

Date	Polarization	Orbit	N ^o . of image	Comments
2007-10-24 to 2007-11-04	HH and HH/VV	Descending 5h44 UTC	2	1 pair with $\Delta t = 11$ days
2008-01-09	HH	Descending 5h44 UTC	1	-
2008-01-11	HH	Ascending 17h25 UTC	1	-
2008-09-29 to 2008-10-21	HH	Descending 5h44 UTC	3	2 pairs with $\Delta t = 11$ days
2009-01-06 to 2009-03-24	HH/HV	Descending 5h44 UTC	8	7 pairs with $\Delta t = 11$ days
2009-05-29 to 2009-08-25	HH	Descending 5h44 UTC	7	5 pairs with $\Delta t = 11$ days
2009-05-31 to 2009-08-27	HH	Ascending 17h25 UTC	9	8 pairs with $\Delta t = 11$ days
2009-09-18 to 2009-10-21	HH	Ascending 17h25 UTC	4	3 pairs with $\Delta t = 11$ days

Table 2: Temporal series of TerraSAR-X images acquired on the Chamonix Mont-Blanc test site.

In this paper, the cross-correlation results are illustrated on the single polarization (HH) descending images which are acquired with an incidence angle of 37° and a spacing of 1.36 m per pixel in range and 2.04 m per pixel in azimuth direction. The range and azimuth image axes correspond to the radar line of sight (LOS) and the sensor displacement direction respectively. The stripmap mode has been chosen because it supplies a large scene coverage (about 30×50

km²) and HR at the same time. Higher resolution images could be used (less than 1 m per pixel in Spotlight mode) but such images are limited to a few kilometers in the azimuth direction. Figure 6 shows a whole strip-map image which covers almost the whole Mont-Blanc area, i.e. French, Italian and Swiss parts.

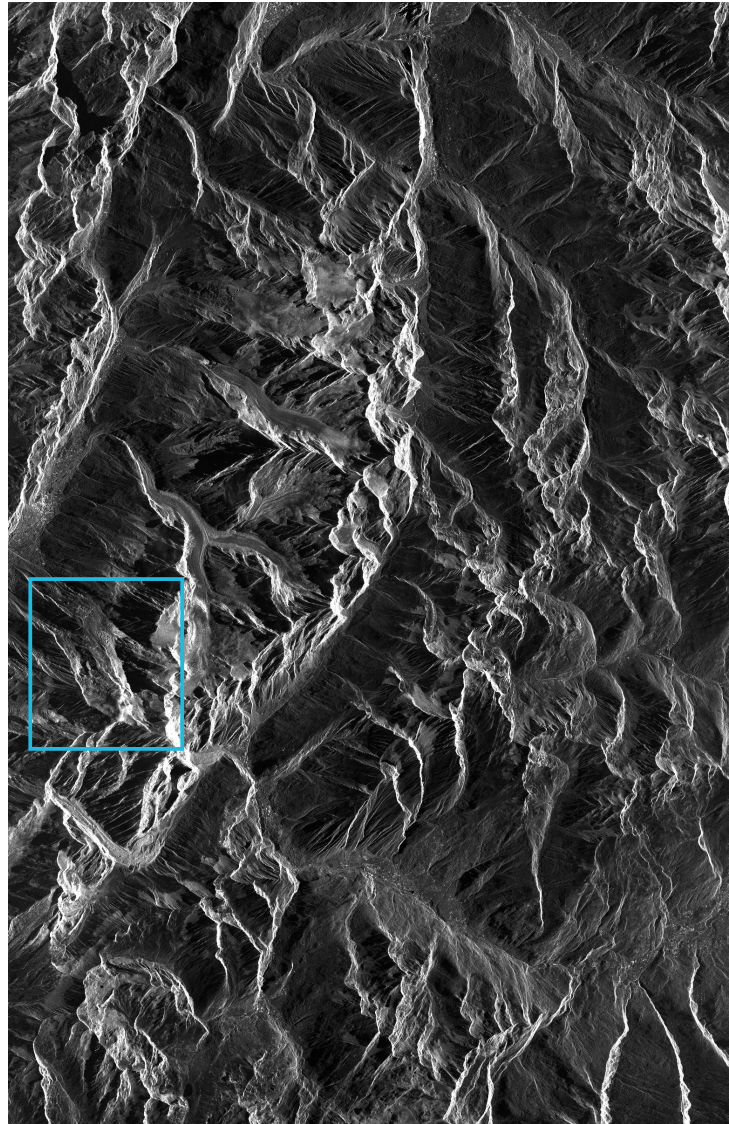


Figure 6: TerraSAR-X amplitude strip-map image (15790×24183 pixels), Mont-Blanc area, 2008-09-29 (averaged by 10×10 for display). The blue rectangle corresponds to the sub-image chosen to illustrate glacier displacements in Figure 8.

4.2 Processing

In the mountainous areas where most of the Alpine glaciers are located, the “range sampling” of SAR images introduces strong geometrical distortions. To avoid geocoding artefacts, the SAR images of the Mont-Blanc test site have been ordered in their initial geometry. The offsets measured in range direction between two images are sensitive to

the position along the swath (near range¹ / far range²), to the topography, as well as to the surface displacement occurred between the two acquisition dates. The offsets measured in azimuth direction mainly depend on the surface displacement (a linear correction is sufficient to remove along-track registration variations over long scenes). The range variations due to the topography depend on the perpendicular baseline between the two orbits as in a stereo configuration. These variations can be predicted by using a digital elevation model (DEM) of the area and the orbital data (antenna state vectors) which are provided together with the images.

In the studied area, the altitude varies between 1000 m ASL (in the Chamonix valley) up to 4800 m ASL (on the Mont-Blanc). For the image pair (2008-09-29/2008-10-10) whose perpendicular baseline is around 138 m, the range registration offsets due to this baseline vary between 28.9 and 82.4 pixels in near and far range respectively.

The glaciers of this test site might move up to 1,5 m per day in the fastest areas, according to in situ measurements. The glacier displacements vary between 0 to 16 m in 11-day (32 m in 22-day), hence 0 to 8 pixels with the resolution of the TerraSAR-X images used in this paper.

According to these a priori displacement information, the TerraSAR-X data acquired over the Mont-Blanc test site are processed in 3 steps:

1. An initial co-registration by a simple translation (without resampling) is applied by matching an area of the image located at an intermediate elevation of about 2000 m ASL.
2. The proposed fast correlation technique is applied on the whole image with a search window of 77×77 pixels, corresponding to an offset of ± 16 m in each direction. On motion-free areas, the sub-pixel offsets provide an accurate estimation of the remaining offset due to the SAR geometry. On the moving glaciers, the measured offset is the sum of the displacement offset and the geometrical offset which has not been compensated for at step 1.
3. Depending on the variations of the geometrical offset along the glaciers, a post-processing step can be necessary to deduce the offsets due only to the glacier movement. The remaining geometrical offset can be subtracted by using either the predictions from the DEM and the orbits, or the results of the sub-pixel correlation around the glaciers.

The correlation results obtained with 61×61 master window (i.e. $M_r \times M_c$) and 77×77 slave window (i.e. $S_r \times S_c$), on the whole TerraSAR-X image presented in Figure 6 are illustrated by the magnitude of the offset vector in Figure 7. The values close to zero (in black) correspond to the motion free areas which are well co-registered by the initial translation (step 1). The remaining offset variation due to the SAR geometry can be observed in the dark

¹The portion of a radar image near the satellite during acquisition.

²The portion of a radar image far away from the satellite during acquisition.

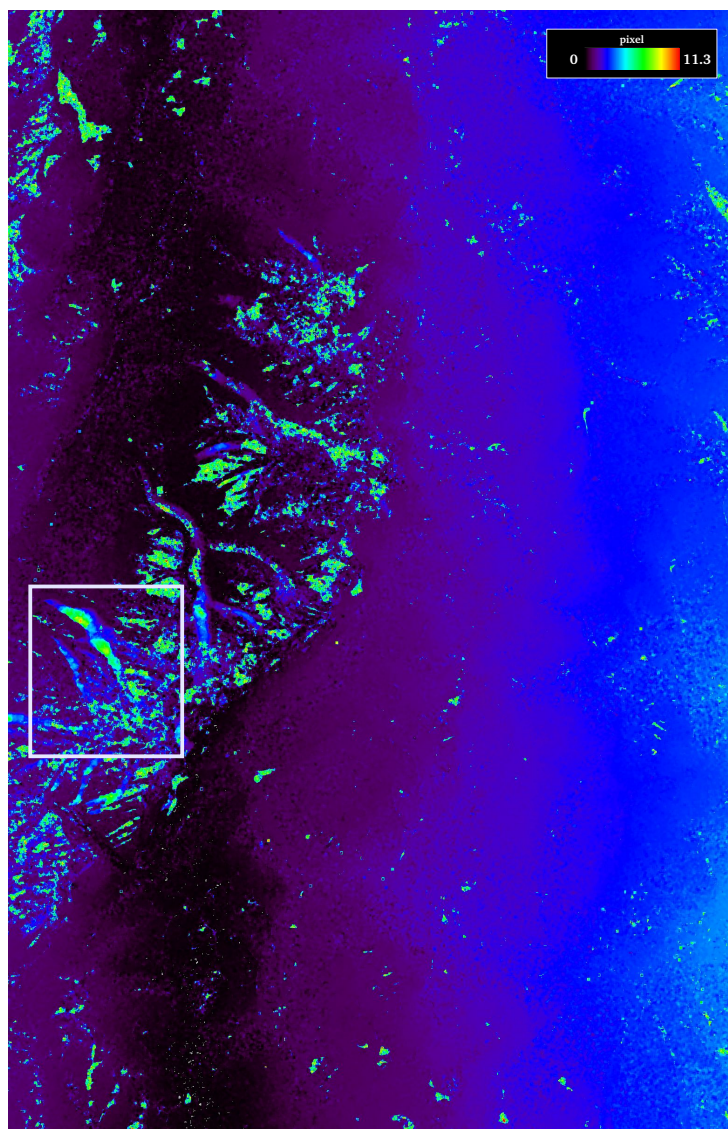


Figure 7: Magnitude of the offset vector derived by NCC between two TerraSAR-X strip-map images (2008-09-29 and 2008-10-10) over the Mont-Blanc area. (averaged by 10×10 for display).

and light blue areas. The shapes of moving glaciers (Mer de Glace, Argentière, Les Bossons...) appear with stronger magnitude in green. Some of the stronger magnitude are due to mis-registration when the correlation technique fails, in areas with strong decorrelation between the 2 images because of surface changes.

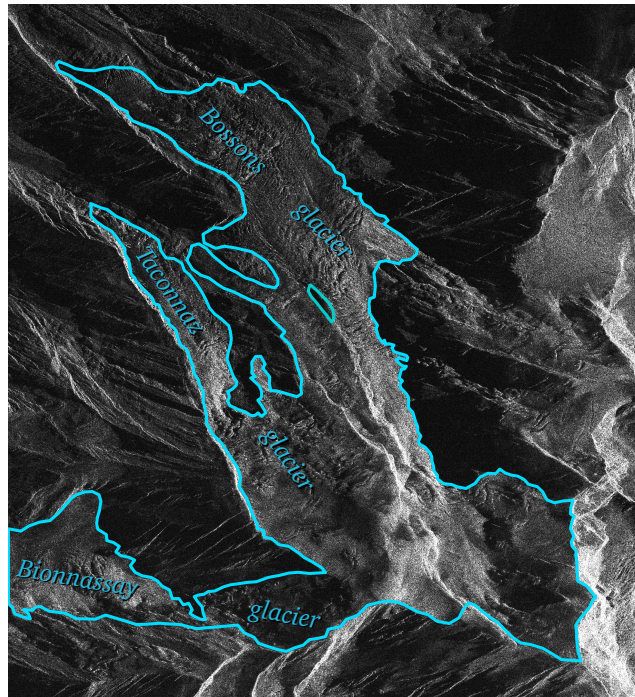
The results obtained on moving glaciers are illustrated with the Tacconnaz, the Bossons and the Bionnassay glaciers in Figure 8. The displacement magnitude and orientation show that the motion is not uniform: the velocity is higher in the center of the glacier, and two acceleration areas appear on the Bossons glacier. These results are consistent with the glacier behaviors, but there is no ground truth available since crevasses and seracs make in situ measurements too dangerous. On the higher part of those glaciers, the magnitude and orientation are very noisy: the correlation technique does not provide reliable results. Larger window size could improve the results on poorly correlated areas, but the window size cannot be very large since the displacement field is not homogeneous over the glaciers and the border discontinuity should be preserved. A flexible choice of window size is useful to find a good trade-off between reducing the “false alarms” (wrong match) and preserving the spatial resolution (displacement field heterogeneity).

4.3 Computation Speedup

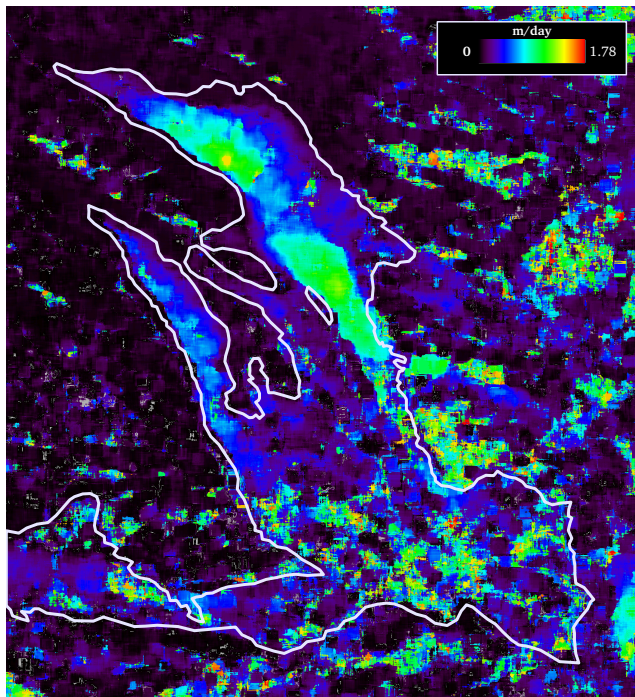
This second experiment on SAR images is realized in the same context as the experiment presented in Section 3: the same algorithms and the same CPU configuration are used. Figure 9 shows the speedup given by the optimization and the number of used CPU. These results confirm those obtained with optical images. The only difference is that the computation time is longer with SAR images. This is mainly due to the master and search window sizes that are larger for SAR images. It is well known that the computation time of the correlation algorithm is strongly linked to the master and search windows sizes and their differences. For the whole SAR image of Figure 6, the computation time without optimization is 30 times longer than this experiment. For the computation of Figure 7, it takes 15 hours with optimization and 8 CPU against 18 days without optimization.

As in Section 3, the gain given by the optimization is very important. Figure 10 illustrates these results. One curve illustrates the difference between the computation time without and with optimization and the other illustrates the gain relatively to the execution without optimization. The relative gain is close to that obtained with the optical images: more than 96%. As the computation time without optimization is very long – many days – in the case of SAR images, the benefit can be expressed in computation days. Thus, the impacts of the optimization and distribution are more important than for the smaller images of the digital camera.

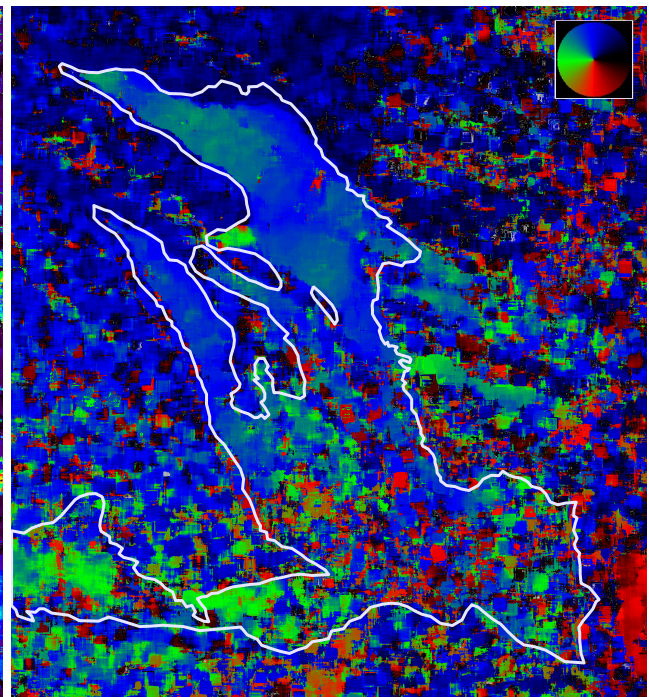
This second experiment in the context of space-borne SAR images, confirms the results obtained with optical in-situ images. To extend these results, another experiment is realized. The objective is to highlight the impact of the master window size on the computation time and the optimization. For this experiment the master window size is increased from 41 to 81 with a step of 4 pixels. The search window size is 16 pixels larger than the master window



(a) Amplitude sub-image (3400 × 3700 pixels).



(b) Magnitude of the 2D displacement measured in the range and azimuth directions.



(c) Orientation of the 2D displacement (color reference on the top right corner).

Figure 8: Displacement field over the Bossons, Taconnaz and Bionnassay glaciers derived by NCC between TerraSAR-X image pair (2008-09-29/2008-10-10).

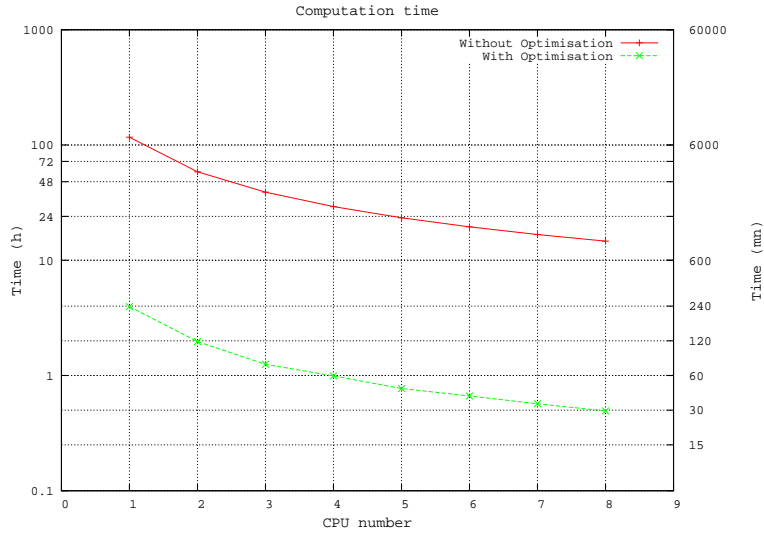


Figure 9: Computation time depending on the optimization and CPU number. (displayed with y – time – log scale)

size. The computation time depending on the optimization and the master window size are given by Figure 11. On one hand, this figure shows that the computation time dramatically increases when the master window size increases. In our case, when the size doubles the computation time is multiplied by more than 3. Despite this impact, this time stays reasonable with the optimized implementation. Thus, the “best” master window size can be searched by experiments. On the other hand, the impact of master window size is quantified in an absolute and a relative point of view as shown in Figure 12. Let us note that the absolute gain increases with the master window size and several hours are saved. It is also important to note that the relative gain increases with the size of the master window. In other words, the larger the master window size is, the more efficient the optimization is.

5 Conclusion and future work

This paper details an optimized implementation of the NCC algorithm. The objective is to reduce the computation time of the correlation technique in order to handle large data set for Earth change monitoring. The saved time induced by the optimization has multiple impacts. The computation on each point of the image can be achieved in a reasonable time: n min/Mega pixel instead of n min/Mega pixel with a conventional approach. High resolution remote sensing images covering large scenes can be processed in a few hours. This fast correlation technique is very useful to extend experimental researches. For example, it allows researcher to experiment different processing parameters and to analyze larger data sets than the conventional techniques.

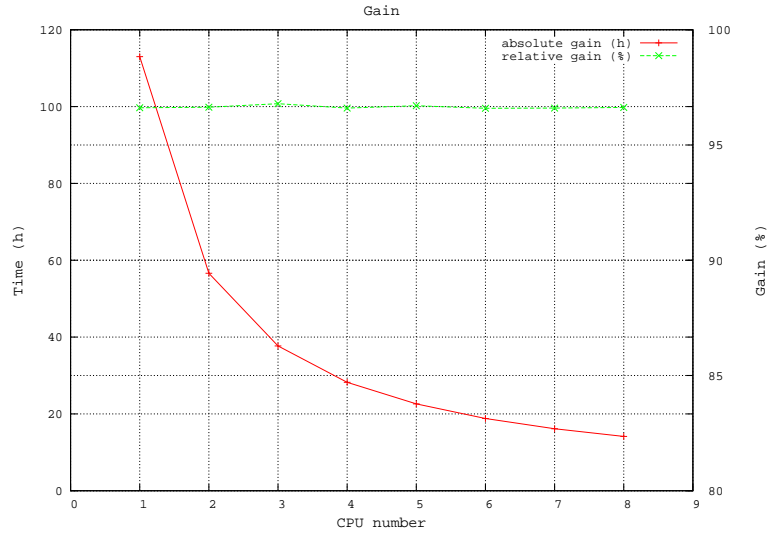


Figure 10: Gain given by the optimization depending on the CPU number.

Two experiments illustrate the benefits of the proposed approach. The evolution of serac falls is studied with optical images and the whole glacier surface evolution can be observed with SAR images. On the Mont-Blanc area, the correlation reveals particular areas like glaciers, lakes or other changing features that can be studied. These experimental results highlight the potential of proximally and remotely sensed images to monitor the glacier flow and to contribute to risk assessment : the Taconnaz glacier is for instance an important source of risk for the access road to the Mont-Blanc tunnel.

Future work includes a comparison between this optimization and different implementations of the FFT approach to illustrate the advantage and limitation of those techniques. As the NCC is only one of the available similarity functions, the study and the optimization of new criteria, different from the NCC, will also be investigated.

Acknowledgment

The authors wish to thank the French Research Agency (ANR) for supporting this work through the Hydro-Sensor-FLOWS project and the EFIDIR project (ANR-2007-MCDC0-04, <http://www.efidir.fr>). They also wish to acknowledge the German Aerospace Agency (DLR) for the TerraSAR-X images (project MTH0232) and the Société d'électricité Emosson SA for their support.

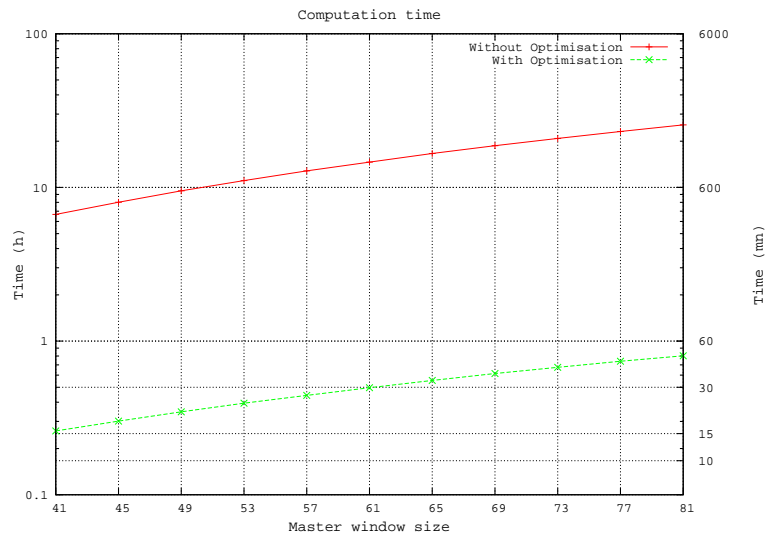


Figure 11: Computation time depending on the optimization and the master window size.

References

- [1] C. Vincent, A. Soruco, D. Six, and E. Le Meur. Glacier thickening and decay analysis from 50 years of glaciological observations performed on glacier d’Argentière, Mont Blanc area, France. *Annals of Glaciology*, 50:73–79, 2009.
- [2] E. Berthier, H. Vadon, D. Baratoux, Y. Arnaud, C. Vincent, K.L. Feigl, F. Rémy, and B. Legrésy. Mountain glaciers surface motion derived from satellite optical imagery. *Remote Sensing Environ.*, 95(1):14–28, 2005.
- [3] D. Scherler, S. Leprince, and M. R. Strecker. Glacier-surface velocities in alpine terrain from optical satellite imagery-accuracy improvement and quality assessment. *Remote Sensing of Environment*, 2008.
- [4] E. Trouvé, G. Vasile, M. Gay, L. Bombrun, P. Grussenmeyer, T. Landes, J.M. Nicolas, P. Bolon, I. Petillot, A. Julea, L. Valet, J. Chanussot, and M. Koehl. Combining airborne photographs and spaceborne SAR data to monitor temperate glaciers. Potentials and limits. *IEEE Transactions on Geoscience and Remote Sensing*, 45(4):905–923, 2007.
- [5] R. Fallourd, O. Harant, E. Trouvé, J.-M. Nicolas, F. Tupin, M. Gay, G. Vasile, L. Bombrun, A. Walpersdorf, J. Serafini, N. Cotte, L. Moreau, and P. Bolon. Monitoring temperate glacier : Combined use of multi-date TerraSAR-X images and continuous GPS measurements. In *MULTITEMP’2009: Fifth International Workshop on the Analysis of Multi-temporal Remote Sensing Images*, pages CDROM, 8 pages, Groton, Connecticut, USA, 2009.

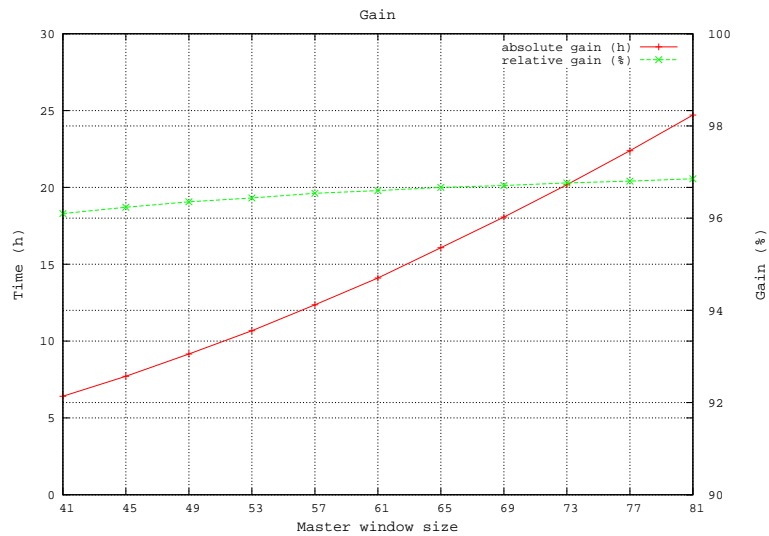


Figure 12: Gain given by the optimization depending on the master window size.

- [6] S. Leprince, S. Barbot, F. Ayoub, and J. P. Avouac. Automatic and precise ortho-rectification, coregistration, and subpixel correlation of satellite images, application to ground deformation measurements. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6):1529–1558, 2007.
- [7] S. Leprince, F. Ayoub, Y. Klinger, and J.P. Avouac. Co-registration of optically sensed images and correlation (COSI-Corr): an operational methodology for ground deformation measurements. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2007)*, Barcelona, Spain, July 2007.
- [8] P.A. Rosen, S. Hensley, G. Peltzer, and M. Simons. Updated repeat orbit interferometry package released. *The Earth Observation System, Transactions, American Geophysical Union, Electronic Supplement*, <http://www.agu.org>, 85(5), 2004.
- [9] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.
- [10] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on “Program Generation, Optimization, and Platform Adaptation”.
- [11] H. Stone, M. Orchard, C. Ee-Chien, and S. Martucci. Fourier-based algorithm for subpixel registration of images. *IEEE Transactions on Geoscience and Remote Sensing*, 39(10):2235–2243, 2001.
- [12] H. Foroosh, J. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing*, 11(3):188–200, 2002.

- [13] C. Collet, J. Chanussot, and K. Chehdi. *Multivariate Image Processing*. WILEY, 2010.
- [14] E. Erten, A. Reigber, and O. Hellwich. Glacier velocity monitoring by maximum likelihood texture tracking. *IEEE Transactions on Geoscience and Remote Sensing*, 47(2):394–405, 2009.
- [15] O. Harant, L. Bombrun, G. Vasile, M. Gay, L. Ferro-Famil, R. Fallourd, E. Trouvé, J.-M. Nicolas, and F. Tupin. Fisher pdf for maximum likelihood texture tracking with high resolution polsar data. In *EUSAR 2010 Proceedings, Aachen, Germany*, pages 418–421, june 2010.
- [16] Adrian N. Evans. Glacier surface motion computation from digital image sequences. *IEEE Transactions on Geoscience and Remote Sensing*, 38(2):1064–1071, 2000.
- [17] W. D. Harrison, K. A. Echelmeyer, and D. M. Cosgrove. The determination of glacier speed by time-lapse photography under unfavorable conditions. *Journal of Glaciology*, 38(129):257–265, 1992.
- [18] R.-M. Krimmel and L.-A. Rasmussen. Using sequential photography to estimate ice velocity at the terminus of columbia glacier, alaska. *Annals of Glaciology*, 8:117–123, 1986.
- [19] W. D. Harrison, C.-F. Raymond, and P. Mackeith. Short period motion events on variegated glacier as observed by automatic photography and seismic methods. *Annals of Glaciology*, 8:82–89, 1986.
- [20] H.-G. Maas, E. Schwalbe, R. Dietrich, M. Bäessler, and H. Ewert. Determination of spatio-temporal velocity fields on glaciers in West-Greenland by terrestrial image sequence analysis. In *IAPRS, Beijing, China, XXXVII, Part B8*, pages 1419–1424, 2008.
- [21] J.-M. Friedt, C. Ferrandez, G. Martin, L. Moreau, M. Griselin, E. Bernard, D. Laffly, and C. Marlin. Automated high resolution image acquisition in polar regions. In *European Geosciences Union, Vienna, Austria*, 2008.
- [22] William K. Pratt. *Digital image processing*. John Wiley & Sons, Second edition, 1991.
- [23] E. Arce and J. L. Marroquin. High-precision stereo disparity estimation using HMMF models. *Image and Vision Computing*, 25:623–636, 2007.
- [24] TerraSAR-X science service system: Proposals Pre-launch. <http://sss.terrasar-x.dlr.de>.
- [25] R. Fallourd, F. Vernier, Y. Yan, E. Trouvé, Ph. Bolon, J.-M. Nicolas, F. Tupin, O. Harant, M. Gay, G. Vasile, L. Moreau, A. Walpersdorf, N. Cotte, and J.-L. Mugnier. Alpine glacier 3D displacement derived from ascending and descending TerraSAR-X images on Mont-Blanc test site. In *EUSAR 2010 Proceedings, Aachen, Germany*, 4 pages, june 2010.