

Automation of the Filament Tracking in the Framework of the HELIO project

X. Bonnin¹ · J. Aboudarham¹ · N. Fuller¹ ·
A. Csillaghy² · R. Bentley³ ·

Received: date / Accepted: date / Published online: date

Abstract We present a new method to automatically track filaments over the solar disk. The filaments are first detected on Meudon Spectroheliograph H α images of the Sun, applying the technique developed by Fuller, Aboudarham, and Bentley (*Solar phys.* **227**, 61, 2005). This technique combines cleaning processes, image segmentation based on region growing, and morphological parameter extraction, including the determination of filament skeletons. The coordinates of the skeleton pixels, given in a heliocentric system, are then converted to a more appropriate reference frame that follows the rotation of the Sun surface. In such a frame, a co-rotating filament is always located around the same position, and its skeletons (extracted from each image) are thus spatially close, forming a group of adjacent features. In a third step, the shape of each skeleton is compared with its neighbours using a curve-matching algorithm. This step will permit us to define the probability [P] that two close filaments in the co-rotating frame are actually the same one observed on two different images. At the end, the pairs of features, for which the corresponding probability is greater than a threshold value, are associated using tracking identification indexes. On a representative sample of filaments, the good agreement between automated and manual tracking confirms the reliability of the technique to be applied on large data sets. Especially, this code is already used in the framework of the *Heliophysics Integrated Observatory* (HELIO) to populate a catalogue dedicated to solar and heliospheric features (HFC). An extension of this method to others filament observations, and possibly the sunspots, faculae, and coronal holes tracking can be also envisaged.

Keywords: Solar filaments, Halpha observations, automated tracking, image processing, virtual observatory, HELIO, HFC

¹ LESIA, Observatoire de Paris, CNRS, UPMC, Université Paris-Diderot, 5 place Jules Janssen, 92195 Meudon, France
email: xavier.bonnin@obsppm.fr email: jean.aboudarham@obsppm.fr email: nicolas.fuller@obsppm.fr

² Institute of 4D Technologies, FHNW, Steinackerstrass 5, CH-5210 Windisch, Switzerland
email: andre.csillaghy@fhnw.ch

³ MSSL, University College London, Hombury St. Mary, Dorking, Surrey RH5 6NT, U.K. email: rdb@mssl.ucl.ac.uk

1. Introduction

The heliophysics community embraces a number of existing disciplines – solar and heliospheric physics, magnetospheric and ionospheric physics for the Earth and other planets – and thus must deal with very large sets of heterogeneous data from ground- and space-based instruments. The *Heliophysics Integrated Observatory* (HELIO: <http://www.helio-vo.eu>) is a virtual observatory dedicated to the solar physics and heliophysics (Bentley *et al.*, 2011). HELIO provides a distributed network of services, which helps the researchers to easily mine relevant information and data. Especially, the *Heliospheric Feature Catalogue* (HFC: <http://voparis-helio.obspm.fr/hfc-gui/index.php>) is a database-oriented service that allows access to a large amount of solar and heliospheric features data. Extraction of feature information stored in the HFC is realized using an increasing number of recognition codes (Fuller, Abouadarham, and Bentley, 2005; Zharkov *et al.*, 2005; Barra *et al.*, 2009; Krista and Gallagher, 2009; Lobzin *et al.*, 2009; Higgins *et al.*, 2011).

We present here a new algorithm for the solar-filament tracking. It has been initially developed in the framework of the HELIO project in order to provide tracking data, as a supplementary information, to the description of filaments already available in the HFC. The filaments are large-scale structures of relatively dense and cool plasma suspended in the hot and thin corona. They are particularly visible on $H\alpha$ observations, where they appear as elongated dark features with several barbs on the solar chromosphere (Tandberg-Hanssen, 1995; Mackay *et al.*, 2010). During their lifetime, their shape and intensity can be subject to a number of modifications, especially, part or all of a filament can sometimes undergo sudden disappearances (which may be followed in some cases by reappearances) at some wavelengths, probably due to changes in their background characteristics – temperature, pressure, *etc.* – or caused by eruptive process. Hence, this sudden and unpredictable behaviour can make filaments difficult to track over the Sun surface. However, following these features over time is relevant, notably for space weather, since erupting disappearances can play a role in the triggering of coronal mass ejections (CMEs) (Gilbert *et al.*, 2000; Gopalswamy *et al.*, 2003).

Many automated methods have been successfully developed during the last ten years to detect filaments, including Gao, Wang, and Zhou (2002), Fuller, Abouadarham, and Bentley (2005), Bernasconi, Rust, and Hakim (2005), Zharkova and Schetinina (2005), and more recently Joshi, Srivastava, and Mathew (2010). In addition to their recognition algorithms, Gao, Wang, and Zhou (2002); Bernasconi, Rust, and Hakim (2005); Joshi, Srivastava, and Mathew (2010) also propose tracking capabilities that allow to identify disappearances by following feature locations day by day. In particular, Gao, Wang, and Zhou (2002) have tracked filament disappearances over a full year, and the code of Bernasconi, Rust, and Hakim (2005) is currently applied to detect and track filaments on Big Bear Solar Observatory (BBSO) images, providing among others data to the *Heliophysics Events Knowledgebase* (HEK: <http://lmsal.com/hek/>). In all cases the basics of the methods are quite similar: starting from heliocentric positions of

the filaments detected at a given time (*i.e.*, on a given image), they estimate the coordinates of their respective centroids on the previous and/or next adjacent images using equations that correct for the solar rotation (Cox, 2000). Then, they check if detected filament lies within a given circle area from the predicted locations; if it does, the two filaments are considered to be the same. In the case of Bernasconi, Rust, and Hakim (2005), where no filament is found they then extend the search up to three days to confirm that the filament actually disappears or not. As highlighted by the authors, the 3-day search is motivated by the fact that filaments sometimes change shape so much between times of observation, that their location may fall out of the search area, temporarily losing the tracking. A consequence is that during this period of time it is not possible to know if the filament actually disappears, or if its shape is just strongly deformed due to splitting or partial disappearance. A tracking algorithm based on time, position, but also shape matching would permit to distinguish between the different possible behaviour. This capability appears to be essential to detect, among others, erupting disappearance.

In our method, filaments are first detected on $H\alpha$ images by an automated recognition code that extracts, among other things, the pruned skeletons of filaments. A pruned skeleton can be defined as the thickest curve inside the feature area that preserves the shape topology; in terms of image processing it results of a thinning followed by a pruning operation (Gonzalez and Woods, 2002) applied on the feature pixels. Then, the coordinates of the resulting skeleton pixels are converted to a more appropriate reference frame that rotates with the solar surface. In such a frame, the skeleton of a filament observed on successive images appears as a cluster of close curves (Mouradian, 1998). Hence, a comparison in this frame of each skeleton curve with its closer neighbours using a curve-matching algorithm, allows us to calculate probabilities that all of these features actually belong to the filament. The algorithm demonstrates its ability to track filaments on large data sets, limiting the number of false tracking detections. Section 2 will introduce the observations used to test the technique, and briefly describe the filament detection process; Section 3 is devoted to the explanation of the tracking algorithm; Section 4 will present the resulting performances of the code. Finally applications will be discussed in Section 5.

2. Detection of Filaments

2.1. Observations

The Meudon spectroheliograph of the Observatoire de Paris performs daily observations of the solar photosphere and chromosphere at three wavelengths: blue wing of the Ca II K 393.4 nm line, or K_{1v} ; centre of the Ca II K 393.4 nm line, or K_3 ; and $H\alpha$ 656.3 nm. More than fifteen years of data are accessible from the BASS2000 website (<http://bass2000.obspm.fr/home.php>), which permits us to apply the tracking algorithm over a large period of time.

The images produced are 2D heliocentric projections of the full solar disk as

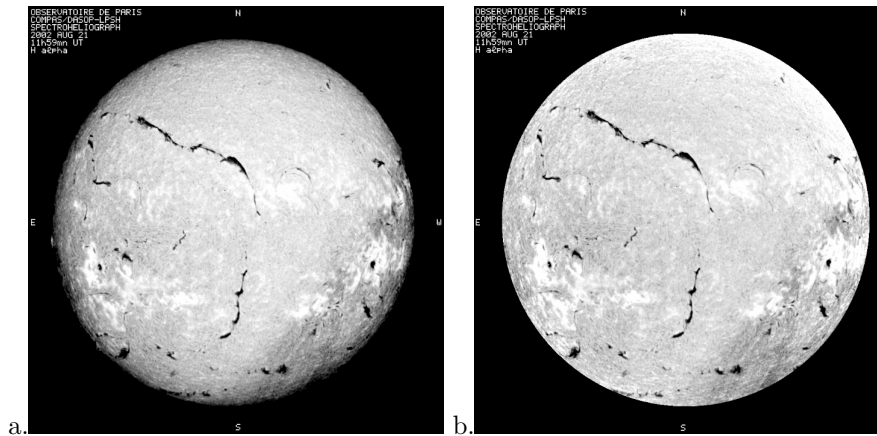


Figure 1. a. Gray-scale image of the Sun observed in $H\alpha$ by the Meudon spectroheliograph on 21 August 2002 at 11:59:00 UT. The image is centred on the solar disk, and modified to have x - and y -axis respectively aligned with the Equator and rotation axis. Due to their cooler temperature, the filaments are seen like dark features on the solar surface. Centre-of-limb effects as well as dark faint lines, caused by dust particles on the optics, are also visible. b. Same image after cleaning processes.

seen from the Earth, with a typical size of 1024 by 1024 pixels, and a spatial resolution of ≈ 2.28 arcsecs. The origin of the reference frame corresponds to the disk centre, x -axis is aligned with the Sun's Equator and points towards the west limb, and y -axis is aligned with the rotation axis and points towards the North. Figure 1 shows such an image before (left panel) and after (right panel) cleaning processes (Fuller, Aboudarham, and Bentley, 2005); these pre-processes are required to avoid false detections, but also to optimize the efficiency of the region-growing process used by the detection algorithm, by increasing the contrast between the filaments and quiet-sun intensities.

2.2. Recognition Code

In order to retrieve the location and morphology of filament skeletons required to perform tracking, an automated recognition code is first run over the full data set available. We use here an algorithm developed in the framework of the *European Grid of Solar Observations* (EGSO: <http://www.egso.org>) project, and successfully applied on $H\alpha$ data from the Meudon Spectroheliograph and BBSO (Fuller, Aboudarham, and Bentley, 2005). This code is now also used in HELIO to provide to the HFC, a description of the filaments detected on Meudon images. (Filaments detected on BBSO observations are planned to be added in HFC in few months.)

The recognition method requires three main steps to be completed:

- i*) Since images are sometimes blurred, which can reduce the efficiency of the filament segmentation, a Laplacian spatial filter (Russ, 2002) is first used to enhance the clearness of filament contours.
- ii*) Then, a region-growing algorithm (Gonzalez and Woods, 2002) permits us to group pixels of a same filament together. Starting from a seed region (found

using an appropriate threshold value), the procedure searches the connected pixels for which the intensities are within a range defined by the mean and standard deviation of their neighbours.

- iii) A morphological-closing operator (Gonzalez and Woods, 2002) will finally merge resulting nearby regions that could be considered as a single filament on the segmented images.

From the boundary detection, several parameters can be extracted concerning the intensity, location, or morphology of filaments. In particular, the algorithm uses thinning and pruning methods (Gonzalez and Woods, 2002) in order to compute skeleton shapes (results of a skeleton computation is illustrated on Figure 2). Two conditions are required to produce such shapes: all skeleton parts should be connected, and the full extent of the region must be contained in the skeletonized representation. Knowing this characteristic will permit us to define, among others, the length, the centre, and the curvature of the filaments. At the end, all of these parameters will be ingested in the HFC from where they can be downloaded using dedicated query interfaces.

For our purpose, the shape of the skeletons will serve as a matching criterion to identify a filament from an image to the following one.

3. Tracking of Filaments

3.1. Description of the Algorithm

In this section we present in detail the tracking algorithm. In addition to the coordinates of skeleton pixels provided by the recognition code, two types of indices are also required to follow co-rotating features over time. The first index, called feature identification number $[\nu_i]$, is allocated to each filament $[i]$ extracted by the recognition code. All of its values are unique (the values of the feature identification number follow typically the order of detection of the filaments) in order to identify and retrieve in the HFC, a feature detected at a given time on a given image. The second index, called tracking identification number $[\tau_i]$, is assigned to each filament $[i]$ by the tracking code. As opposed to ν_i several filaments can have the same value for τ_i , which will be an indication that these filaments are actually components of the same co-rotating feature observed on successive images. The values of τ_i are set by the tracking code, comparing and associating filaments two by two (as explained in Section 3.1.4). At the beginning of the execution, all of the tracking indexes $[\tau_i]$ are initially equal to the feature ones $[\nu_i]$ by default.

The main steps of the tracking method can be summarized as follows:

- i) We select the time $[t_i]$ of the image $[l]$ for which we want to perform tracking.
- ii) We then define a time window $[t_0, t_1]$ centred on t_i and spanning one Carrington rotation of the Sun, such that $t_0 = t_i - 0.5T_{\text{carr}}$ and $t_1 = t_i + 0.5T_{\text{carr}}$, where $T_{\text{carr}} = 27.2753$ days. The choice of this time range makes sure that all of the filaments detected at t_i will be fully tracked over a period of $0.5T_{\text{carr}}$ days (which is actually the average time required to cross the solar disk).

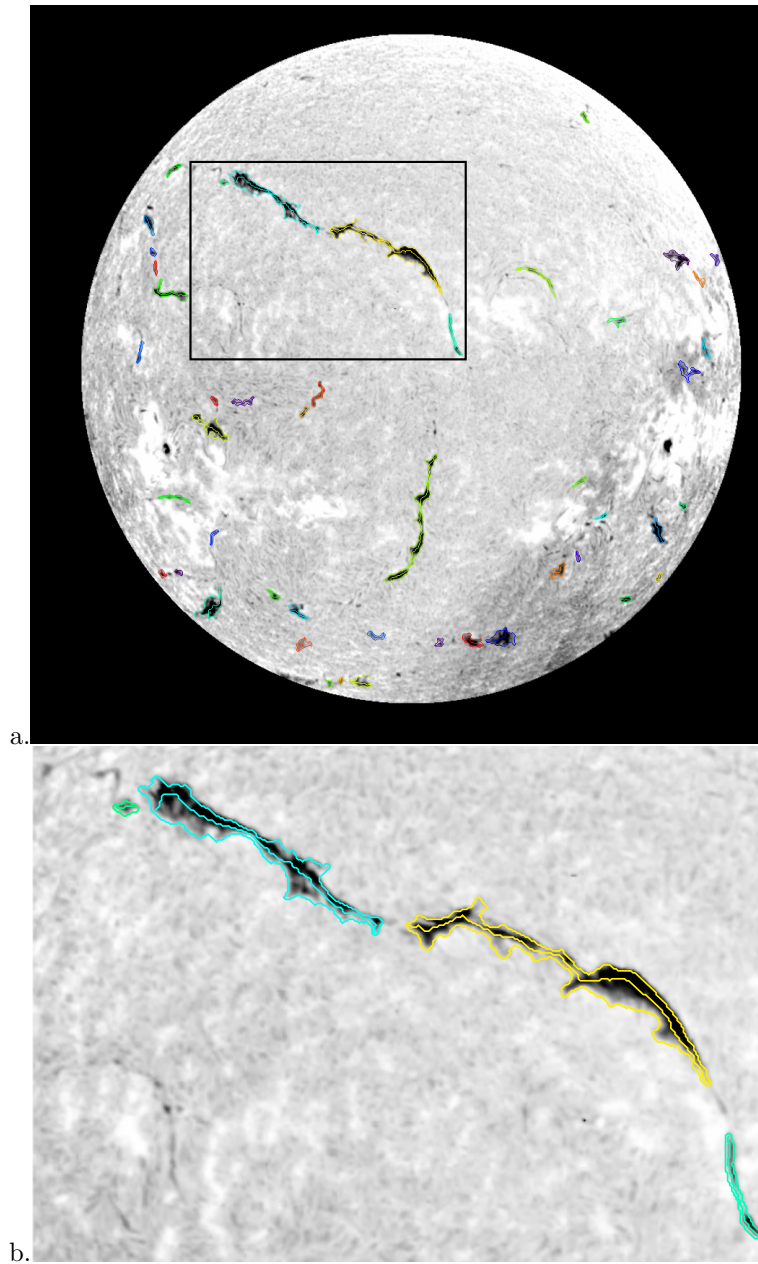


Figure 2. a. Same image as in Figure 1b, but over-plotted with coloured contours and skeletons extracted from the recognition code. (The colors used here are randomly distributed to distinguish individual features detected.) b. Zoom on the area marked by the black box on the image displayed in Panel a.

- iii) We download the HFC information about filaments detected between t_0 and t_1 ; parameters required are typically the pixel coordinates of the pruned skeletons, but also their corresponding feature identification numbers $[\nu]$. (In practice, the corresponding tracking identification numbers $[\tau]$ are also loaded to ensure a continuity with possible previous tracking runs.)
- iv) Since the pixel coordinates are given in the reference frame of images (see Section 2.1), we convert them to a more appropriate coordinates system as explained in the following section.
- v) We apply a curve-matching algorithm in the new frame. The purpose of this algorithm is to compare the shape of all of the possible pairs of skeletons: if two skeletons have similar shapes, then they are assumed to be two distinct parts of the same filament moving over the Sun's surface.
- vi) When the comparison succeeds, we associate the two corresponding skeletons by allocating them the same tracking identification number.
- vii) The same process is then repeated on the next image $[\iota + 1]$ taken at time $t_{\iota+1}$, saving all of the tracking numbers $[\tau]$ defined during previous runs on the image $[\iota]$. This operation allows us to conserve tracking information from a processed image to the following.

3.1.1. Reference Frame used for Tracking

The tracking for filaments detected at time $t = t_\iota$ is performed on a specific reference frame \mathfrak{R} that follows the solar rotation between t_0 and t_1 . Such a frame has the advantage to: i) stabilize the coordinates of co-rotating filaments around the same position, by correcting the translation of heliocentric longitudes for the rotation, ii) make appear all the segments of a filament visible between t_0 and t_1 , which may suffer full or partial disappearance during its lifetime, iii) since along the first dimension time and space are intermingled, offer fast computation time by working in two dimensions instead of three. (The computation for all of the filaments detected over one solar rotation takes less than ≈ 30 seconds on average on a 3.06 GHz Intel duo core machine.)

For each image processed the detection code returns the heliocentric coordinates (X, Y) of skeleton pixels (as defined in Section 2.1), which must be converted in the proper coordinates system associated to \mathfrak{R} before being usable by the matching algorithm. To achieve this goal, the heliographic longitudes and latitudes (φ_h, λ_h) of skeleton pixels on the Sun's surface are first computed using an appropriate transformation matrix (Thompson, 2006), then the coordinates (φ, λ) in the co-rotating frame \mathfrak{R} are deduced using the relations:

$$\lambda = \lambda_h, \quad (1)$$

$$\varphi = \Omega_{\text{carr}}(t_0 - t) + (\varphi_{t_0} - \varphi_h) + \varphi_{DRC}(\lambda_h), \quad (2)$$

where Ω_{carr} is the Carrington rotation speed in degrees.day^{-1} , t is the time when the filament was detected in decimal days (*i.e.*, the time of the observation), t_0 is the start time of the tracking period in decimal days, $\varphi_{t_0} = 360^\circ$ is the longitude of the central meridian in \mathfrak{R} at t_0 , and $\varphi_{DRC}(\lambda_h)$ is a term that corrects the

effects caused by differential rotation (Ulrich and Boyden, 2006). These effects are not significant here, but will become important when the algorithm will be run to track filaments from one solar rotation to the following (see Section 3.2). We note that the coordinates system defined here is quite similar to the Carrington one, but starting at φ_{t_0} .

The steps of calculation of the skeleton coordinates in the reference frame \mathfrak{R} are illustrated on Figure 3. In this example, we want to track filaments detected on a Meudon spectroheliogram on 21 August 2002 at 11:59:00 UT; date that defines here the time of observation $[t_i]$. Thus, according to the discussion in Section 3.1, the tracking process concerns all of the filaments extracted between 7 August 2002 at 20:40:47 UT and 4 September 2002 at 03:17:12 UT, which correspond to the start time $[t_0]$ and the end time $[t_1]$ of the tracking period respectively. Panels a, b, c, d, and e, show a sample of five images of the Sun obtained during this time range; the date of observation is indicated in the upper-left corner of each image (for more clarity only a few images produced during this period are displayed). The heliocentric coordinates of skeleton pixels as well as the identification numbers of each detected filament are also plotted using a coloured scale. Panel f presents an example of corresponding coordinates (φ, λ) , calculated for the co-rotating filament visible inside the white bounding rectangles on the previous panels. In \mathfrak{R} , all of the skeletons belonging to this filament form a structured cluster around the same location, giving the full shape of the filament over the disk crossing.

Figure 4 presents the resulting synoptic map of skeleton pixel's coordinates converted in \mathfrak{R} for the previous example over the full period $[t_0, t_1]$. The longitude axis uses the Carrington convention starting at $\varphi_{t_0} = 360^\circ$, and ending at $\varphi_{t_1} = 0^\circ$, with $\varphi_{t_i} = 180^\circ$, corresponding to the longitude of the central meridian on 21 August 2002 at 11:59:00 UT. At this stage of the procedure, the heliocentric and co-rotating pixel coordinates of features, as well as their corresponding identification numbers $[\nu$ and $\tau]$, are stored by the tracking code.

3.1.2. Curvilinear Interpolation

To be consistent, the curve-matching algorithm must be applied on two curve segments having the same number of points and the same length. However, due at least to the coordinates' projections, the distribution of points along the skeleton curves is not necessarily uniform, hence, we need to interpolate the curves in such a way that the distance $[ds]$ between two consecutive points is constant. At the same time, such a procedure can be seen as a smoothing operation that will highlight the main shape of the skeletons. It results that the value in degrees of $[ds]$ to use in the co-rotating frame has to be a compromise between the spatial resolution (*i.e.*, minimum distance between two points) and the length of the skeletons: too small a value can significantly increase the computation time, but also the possible spatial fluctuations along the curve at small scales, whereas too large a value decreases the number of points along the curve, and so the spatial resolution used to define its shape.

Once an acceptable value of $[ds]$ is found, starting from the location of the first

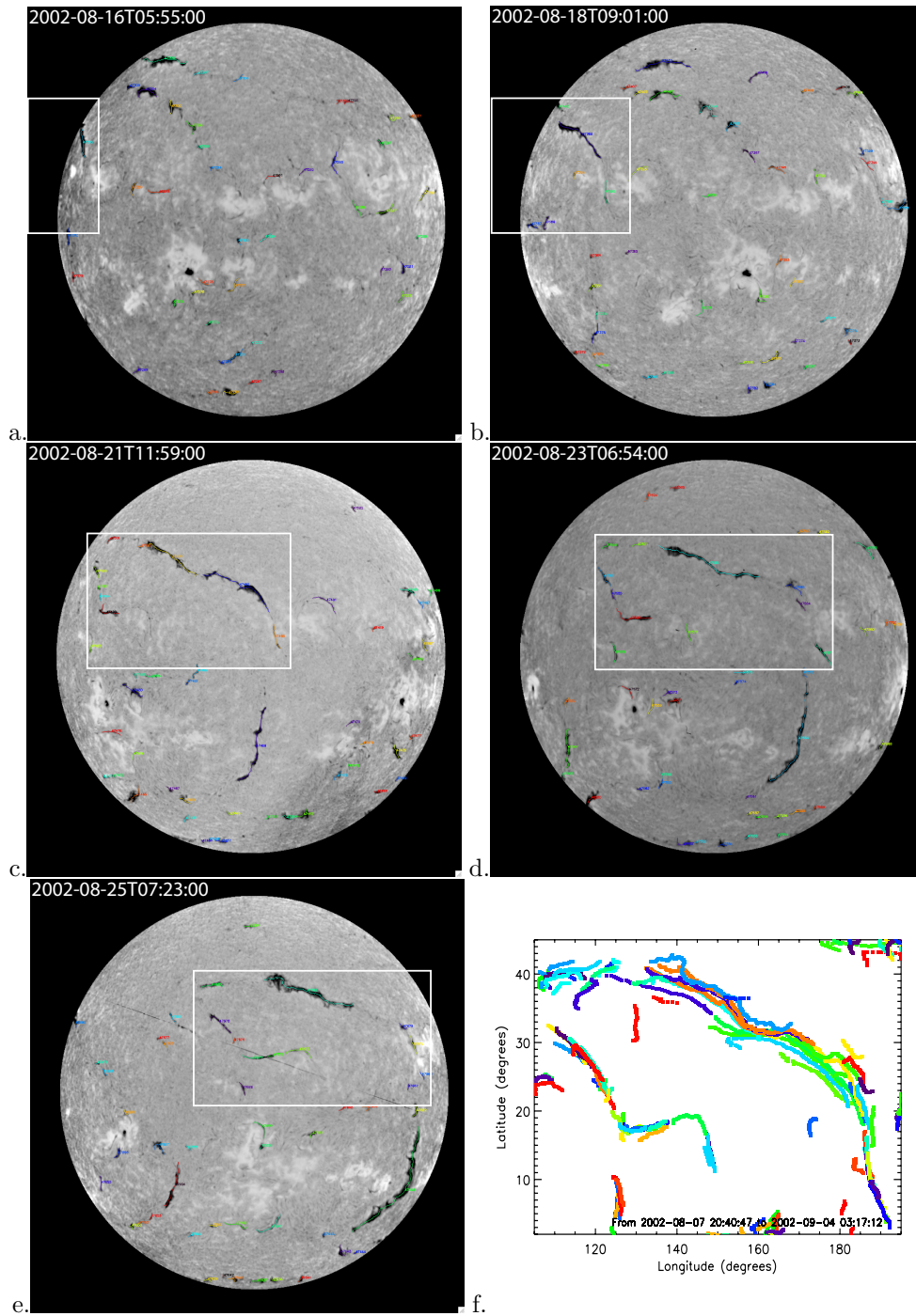


Figure 3. Illustration of the calculation of the skeleton pixel's coordinates from the heliocentric coordinates system to the co-rotating one. Panels a, b, c, d, and e show a sample of five images of the Sun produced between 7 August 2002 at 20:40:47 UT and 4 September 2002 at 03:17:12 UT (on each image the date of observation is written in the upper-left corner). Example of skeleton pixel's coordinates (φ, λ) calculated in the co-rotating frame \mathcal{R} are plotted on Panel f; the region displayed corresponds to the white bounding rectangles on each image.

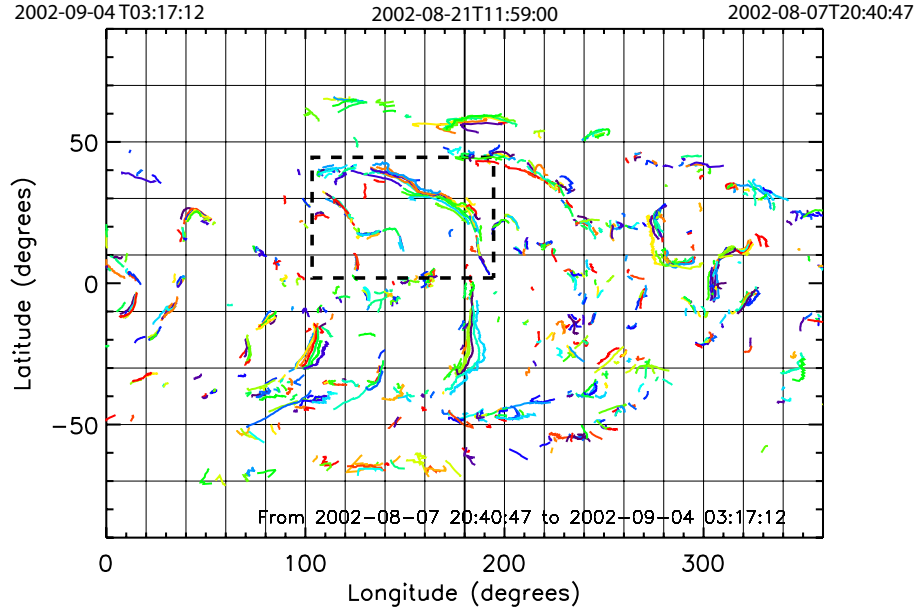


Figure 4. Synoptic map of the skeleton pixel's coordinates in \mathfrak{R} between 07 August 2002 at 20:40:47 UT (*i.e.*, t_0) and 04 September 2002 at 03:17:12 UT (*i.e.*, t_1). Carrington convention is used to define the longitude axis. The vertical darker line at the centre indicates the longitude $\varphi_{t_i} = 180^\circ$, which corresponds the longitude of the central meridian on 21 August 2002 at 11:59:00 UT. The bounding rectangle delimited by black dashed line shows the area displayed on the panel f in Figure 3.

point (φ_1, λ_1) at one of the ends of the curve (the order of the points along the curve goes from one end to the other), the interpolation method consists in the determination of the intersection point $(\varphi_{k_{int}}, \lambda_{k_{int}})$ between the circle of radius $[ds]$ centred on (φ_1, λ_1) , and the line passing through the two points (φ_k, λ_k) and $(\varphi_{k+1}, \lambda_{k+1})$ of the curve, which satisfy the condition:

$$\sum_{l=2}^k \sqrt{(\varphi_l - \varphi_{l-1})^2 + (\lambda_l - \lambda_{l-1})^2} \leq ds \leq \sum_{l=2}^{k+1} \sqrt{(\varphi_l - \varphi_{l-1})^2 + (\lambda_l - \lambda_{l-1})^2}. \quad (3)$$

When the intersection point $(\varphi_{k_{int}}, \lambda_{k_{int}})$ is known, the process is then repeated replacing (φ_1, λ_1) by $(\varphi_{k_{int}}, \lambda_{k_{int}})$ as the new circle centre, and calculating the next intersection point $(\varphi_{k_{int+1}}, \lambda_{k_{int+1}})$ with the curve. The interpolation finally stops when the other end of the curve is reached.

At the same time, the lengths of the skeletons are estimated by simply summing the distances $[ds]$ found between each interpolated points (*i.e.*, $(\varphi_{k_{int}}, \lambda_{k_{int}})$ and $(\varphi_{k_{int+1}}, \lambda_{k_{int+1}})$). This technique will slightly under-estimated the actual length, which appears to be not really significant on the tracking results.

3.1.3. Curve Matching Algorithm

In order to significantly reduce the computation time, the matching algorithm is only applied on skeletons that are close enough on the reference frame. To achieve this goal, for each coordinates $(\varphi_{k_i}, \lambda_{k_i})$ of a first skeleton i , we search the coordinates $(\varphi_{k_j}, \lambda_{k_j})$ of others skeletons j ($j \neq i$) for which the distance $r_{k_i k_j} = \sqrt{(\varphi_{k_i} - \varphi_{k_j})^2 + (\lambda_{k_i} - \lambda_{k_j})^2}$ is less than a maximum value r_{\max} (defined in degrees in \mathfrak{R}). If the condition $r_{k_i k_j} < r_{\max}$ is fulfilled, then the curve matching algorithm is applied. In practice the value of the input parameter r_{\max} does not affect significantly the efficiency of the algorithm, because it will automatically dissociate features that are too much distant.

Given two curves C_1 and C_2 of close skeletons 1 and 2 from the same filament observed on two different images at two different times, there should be a Euclidean transformation E , such that EC_1 matches C_2 . However, since during its lifetime a filament can be subject to shape deformation, segmentation, but also translation and/or rotation, no exact match is likely to occur, and so we look for a transformation E giving the best match in the least-squares sense. Let C_1 and C_2 be represented by the vectorial sequences $(\mathbf{u}_{\mathbf{k}})_{\mathbf{k}=1}^n$ and $(\mathbf{v}_{\mathbf{k}})_{\mathbf{k}=1}^n$ respectively, where n is the number of points of the curves (we assume here that the two curves have the same length). Matching consists of finding a Euclidean transformation E of the plane that will minimize the distance $[l^2]$ between the sequences $(E\mathbf{u}_{\mathbf{k}})_{\mathbf{k}=1}^n$ and $(\mathbf{v}_{\mathbf{k}})_{\mathbf{k}=1}^n$, such that:

$$\Delta = \min_E l^2 = \min_E \sum_{\mathbf{k}=1}^n |E\mathbf{u}_{\mathbf{k}} - \mathbf{v}_{\mathbf{k}}|^2. \quad (4)$$

To simplify the calculation, the curve C_1 is first translated such as $\sum_{\mathbf{k}=1}^n \mathbf{u}_{\mathbf{k}} = 0$. Then, if we write E as $E\mathbf{u}_{\mathbf{k}} = R_\theta \mathbf{u}_{\mathbf{k}} + \mathbf{a}$, where R_θ denotes a counter-clockwise rotation by an angle θ , and \mathbf{a} a translation; the best match is obtained when [Ayache and Faugeras \(1986\)](#); [Wolfson \(1990\)](#)

$$\mathbf{a} = \frac{1}{n} \sum_{\mathbf{k}=1}^n \mathbf{v}_{\mathbf{k}}, \text{ and } \theta = - \sum_{\mathbf{k}=1}^n u_{\mathbf{k}} \overline{v_{\mathbf{k}}}, \quad (5)$$

where $u_{\mathbf{k}}$ and $\overline{v_{\mathbf{k}}}$ are complex representations of the vectors $\mathbf{u}_{\mathbf{k}}$ and $\mathbf{v}_{\mathbf{k}}$ respectively.

At the end, for each pair (C_i, C_j) of skeleton curves analysed, the matching process returns three best-fitted parameters

$$\left(a = |\mathbf{a}| + \delta a, \theta = |\theta|, l = \sqrt{\Delta} \right)_{ij}, \quad (6)$$

where δa is a term that corrects the $\sum_{\mathbf{k}=1}^n \mathbf{u}_{\mathbf{k}} = 0$ translation introduced previously. These parameters, which represent respectively the degrees of translation, rotation, and deformation between the two curves, will be used to compute in a

second step the probability $[P_{ij}]$ that the skeletons $[i$ and $j]$ belong to the same filament.

If two skeletons have different lengths, $l_i \neq l_j$, the matching is also performed but between the smallest of the two, say i , and all of the possible sub-segments k of the other one j that have the same length $l_k = l_i$. In this case, only the probability $[P_{ik}]$ for the sub-segment returning the best match is finally retained. This technique becomes actually consistent only if the lengths of the two curves do not differ significantly, we therefore introduce an additional condition to the minimum on maximum lengths ratio: if this ratio is too small, then the matching algorithm is not applied.

3.1.4. Confidence of Tracking

For each set of best-fitted parameters $(a, \theta, l)_{ij}$, we calculate a normalized probability $[P_{ij}]$ defined by:

$$P_{ij} = W_a P_a + W_\theta P_\theta + W_l P_l \quad (0 \leq P_{ij} \leq 1), \quad (7)$$

where P_a , P_θ , and P_l are the probability terms respectively related to the translation $[a]$, rotation $[\theta]$, and deformation $[l]$ parameters, and W_a , W_θ , and W_l are the corresponding probability weights, which satisfy $W_a + W_\theta + W_l = 1$. P_{ij} can be seen as the level of trust of the tracking; the larger it is, the more the automated code is confident about the fact the two skeletons $[i$ and $j]$ are actually two segments of the same filament.

Since no formal relation exists, the P_a , P_θ , and P_l terms are simply described here using linear expressions, nevertheless, under the condition that the probability $[P_{ij}]$ decreases when the three parameters a , θ , and l increase:

$$P_a = 1 - \frac{a}{a_0}(1 - P_\Theta) \text{ if } a \leq \frac{a_0}{1 - P_\Theta}, \text{ and } P_a = 0 \text{ otherwise,} \quad (8)$$

$$P_\theta = 1 - \frac{\theta}{\theta_0}(1 - P_\Theta) \text{ if } \theta \leq \frac{\theta_0}{1 - P_\Theta}, \text{ and } P_\theta = 0 \text{ otherwise,} \quad (9)$$

$$P_l = 1 - \frac{l}{l_0}(1 - P_\Theta) \text{ if } l \leq \frac{l_0}{1 - P_\Theta}, \text{ and } P_l = 0 \text{ otherwise,} \quad (10)$$

where a_0 , θ_0 , and l_0 are input parameters that satisfy $P_{a=a_0} = P_\Theta$, $P_{\theta=\theta_0} = P_\Theta$, and $P_{l=l_0} = P_\Theta$ respectively. At the same time, the three conditions $a \leq \frac{a_0}{1 - P_\Theta}$, $\theta \leq \frac{\theta_0}{1 - P_\Theta}$, and $l \leq \frac{l_0}{1 - P_\Theta}$ will ensure that P_a , P_θ , and P_l have always positive values. The determination of these three parameters' values will be explained later in Section 4.2. Concerning P_Θ , we take $P_\Theta = 0.5$ which is actually also the value of the respective probability $[P_{ij}]$ above which two skeletons are assumed to match, and are thus associated.

The association step is very important since it will permit us to follow a filament on successive images, assigning the same tracking identification number to its skeletons. An example of such a process is described on Figure 5. Where two skeletons $[i$ and $j]$ are not associated together (*i.e.*, $\tau_i \neq \tau_j$), but both are associated with a third one $[k]$ (*i.e.*, $\tau_i = \tau_k$ and $\tau_j = \tau_k$), then all three will be

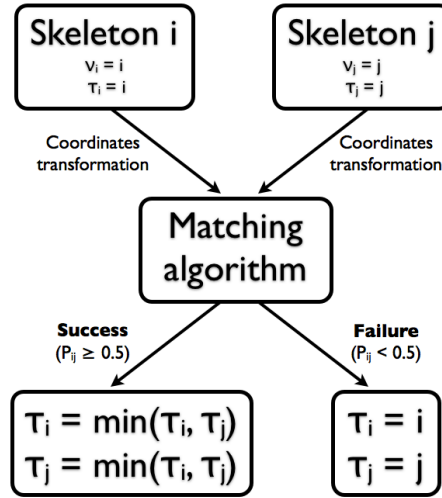


Figure 5. Given two skeletons $[i$ and $j]$ of feature and tracking identification numbers initially equal to $(\nu_i = i, \tau_i = i)$ and $(\nu_j = j, \tau_j = j)$ respectively, where $i \neq j$, the association consists of assigning the same tracking identification number to both skeletons when the matching succeeds. In practice the minimum value between τ_i and τ_j is conserved. In the case where at least one of the skeletons compared has already received a different tracking number, from a previous matching with another skeleton k for instance, $\tau_i = \tau_k \neq i$, then we assign to all the features $[i, j, \text{ and } k]$ the same number $\tau_i = \tau_j = \tau_k = \min(\tau_i, \tau_j, \tau_k)$.

related, and will obtain the same tracking identification number (i.e., $\tau_i = \tau_j = \tau_k$). This condition appears to be very efficient for merging several segments of a filament that suffers significant shape modifications.

Once all of the associations between the pairs of curves are done, for each skeleton $[i]$ we finally calculate the average value

$$\langle P_i \rangle = \frac{1}{N_j} \sum_{j=1}^{N_j} P_{ij}, \quad (11)$$

where N_j is the number of matchings realized between the skeleton $[i]$ and the skeletons $[j]$ ($j \neq i$) for which $\tau_j = \tau_i$. This value will be used as an average tracking confidence level for the corresponding filament $[i]$.

3.2. Tracking Filaments over Two Successive Solar Rotations

The algorithm can also be adapted to track filaments that perform two consecutive crossings on the solar disk. If we assume that a given filament is observed around a time t_ι , but also around $t_\iota - T_{\text{carr}}$, then we can extend the previous time range $[t_0, t_1]$ such that $t_0 \approx t_\iota - 1.5T_{\text{carr}}$ and $t_1 \approx t_\iota + 0.5T_{\text{carr}}$, where the average coordinates of the filament skeletons in the current and in the previous rotations will be respectively $(\langle \varphi \rangle, \langle \lambda \rangle)$ and $(\langle \varphi \rangle + 360^\circ, \langle \lambda \rangle)$ (using the reference convention defined by Equation (1) and Equation (2)). Figure 6 shows the skeleton pixel's coordinates in \mathfrak{R} between t_0 and t_1 , taken t_ι at 11:59:00 UT

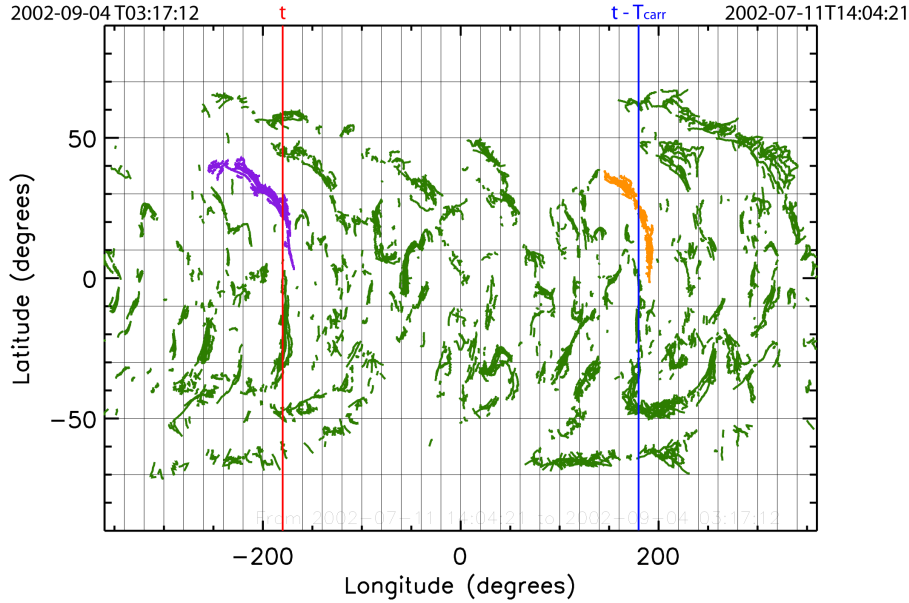


Figure 6. Synoptic map of the new skeleton pixel's coordinates between t_0 at 14:04:21 UT on 11 July 2002, and t_1 at 03:17:12 UT on 4 September 2002. Carrington convention is still used here to define the longitude axis. The vertical red line on the left indicates the longitude $\varphi_t = -180^\circ$, which corresponds the longitude of the central meridian at t , on 21 August 2002 at 11:59:00 UT. The vertical blue line on the right indicates the longitude $\varphi_{t-T_{\text{carr}}} = +180^\circ$, which coincides with the longitude of the central meridian at $t - T_{\text{carr}}$. The two groups of skeletons coloured respectively in purple and orange belong to the same filament observed over two solar rotations (all the others skeletons are plotted here in green for more clarity). In this example, the matching process is first used to group together the purple skeletons on one side, and the orange skeletons on the other. Then, when each group has its own tracking identification number, the matching algorithm is run to compare the curves of the two clusters using Equation (14). In this case the matching process can be done because both group gravity centres are separated by 360° .

on 21 August 2002.

At this stage, we first apply the matching algorithm in order to set the tracking identification numbers (as described on previous sections), then from a first group $[g_i]$ of skeletons $[i]$ having the same tracking identification number $\tau_i = \tau_{g_i}$ and located around $(\langle \varphi_i \rangle, \langle \lambda_i \rangle)$, we look for a possible group $[g_j]$ of skeletons $[j]$, satisfying $\tau_j = \tau_{g_j}$ and situated around the coordinates $(\langle \varphi_j \rangle \approx \langle \varphi_i \rangle + 360^\circ, \langle \lambda_j \rangle \approx \langle \lambda_i \rangle)$. If such a group $[g_j]$ is found, then the matching algorithm is run between the skeletons of both groups only. In practice, to start the matching process, the average gravity center $(\langle \varphi_j \rangle, \langle \lambda_j \rangle)$ of the group $[g_j]$ must be inside the circle of radius r_{max} and centred on $(\langle \varphi_i \rangle + 360^\circ, \langle \lambda_i \rangle)$.

However, since the chance to observe the same filament over several successive rotation periods decreases with time, we introduce here a new probability term $[P_{\Delta t}]$ that is proportional to the difference Δt of the observation times between the two skeletons to match. (Since one solar rotation period separates the two features, Δt should be normally around T_{carr} .) Hence, the resulting probability

$[P'_{ij}]$ that we calculate for each pair (i, j) of skeletons of the two groups will be slightly different:

$$P'_{ij} = P_{\Delta t} P_{ij}, \quad (12)$$

where P_{ij} is the probability defined in Equation (7).

In order to reproduce the filament's average lifetime, the term $P_{\Delta t}$ must be maximal (*i.e.*, $P_{\Delta t} = 1$) when $\Delta t = 0$, and must tend towards zero when Δt increases. As for P_a , P_θ , and P_l (see Equation (8)), we use here a linear expression defined by:

$$P_{\Delta t} = 1 - \frac{\Delta t}{\Delta t_0}(1 - P_\Theta) \text{ if } \Delta t \leq \frac{\Delta t_0}{1 - P_\Theta}, \text{ and } P_{\Delta t} = 0 \text{ otherwise,} \quad (13)$$

where Δt_0 is an input parameter above which the probability $[P_{\Delta t}]$ falls below a given value $P_\Theta = 0.5$. The value of Δt_0 is also estimated empirically as explained in the next section.

Finally, the code will associate the two groups $[g_i]$ and $[g_j]$ of skeletons only if the average value

$$\langle P' \rangle = \frac{1}{2} \frac{1}{N_i} \frac{1}{N_j} \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} P'_{ij}, \quad (14)$$

where N_i and N_j are respectively the number of skeletons in the first and second group, is greater than 0.5. In this case, all of the skeletons $[i]$ of the first group $[g_i]$ will receive a third identification number $[\omega_i]$, which will be equal to the tracking identification number of the group $[g_j]$ such as $\omega_i = \tau_j = \tau_{g_j}$.

At the end of the tracking computation, a filament will be specified by three identification numbers $[\nu_i, \tau_i, \text{ and } \omega_i]$ to be identified on a image, to be tracked over the solar disk, and to be possibly linked to another feature on the previous rotation respectively. This information will be written in a dedicated tracking table of the HFC.

3.3. Filaments Behaviour

In addition to the tracking algorithm, a module has been implemented in the code to characterize the behaviour of the filaments during their crossing the solar disk. To achieve this goal, the procedure checks, using the tracking identification numbers, that each co-rotating filament is well detected on every image between its first and last times of observation. If during this period the filament disappears (*i.e.*, is not seen on one or more successive images), then the times corresponding to the last observation before its disappearance and the first observation after its re-appearance are saved. Moreover, if the filament appears once it has crossed the east limb, and/or disappears before reaching the west limb, the information is also returned by the code. To proceed in this case, average heliographic coordinates $\langle \varphi_h \rangle$, $\langle \lambda_h \rangle$ of the filament skeleton centres on the first/last image are used to calculate the predicted longitudes on the observation just before/after;

if these longitudes are less than 70° (in absolute value), then we assume that there is an appearance/disappearance after/before the limb. As for the tracking data, the results of the analysis are stored in the tracking table of the HFC.

4. Performances of the Algorithm

4.1. Assessments

To optimize the tracking code, we need to define the combination of input parameters $[a_0, \theta_0, l_0, \Delta t_0]$ that gives the best matching rate. This set will serve as a reference for tracking the filaments on the Meudon $H\alpha$ spectroheliograms. At the same time, the assessment of these parameters will also give an indication of the performance of the algorithm, providing a *positive* tracking rate $[R_p]$ after each run (as explained below).

To find the best set of input parameters, we compared the code results with a representative sample of filaments manually tracked. This sample contains approximately 2000 filaments, detected by the recognition code between 2000 and 2009 on Meudon spectroheliograms. The tracking was then performed identifying "by-hand" the co-rotating filaments on synthetic synoptic maps, such as the one displayed on Figure 4. Direct checking on corresponding $H\alpha$ images was also done to confirm the first choice. In order to keep a track of this selection, the results are written in ASCII format files using identification numbers $(\nu_i^{\text{man}}, \tau_i^{\text{man}}, \omega_i^{\text{man}})$ as in the automatic code. In practice, since all filaments were loaded from the HFC, the identification numbers $[\nu_i^{\text{man}}$ and $\nu_i]$ are actually identical; this criterion will permit us to identify individual features between both sets of results.

Once the sample was generated, we then proceeded as follows:

- i)* The values of the input parameters ds , r_{max} , W_a , W_θ , W_l , P_Θ , and Θ were first fixed to a given value. (The effects of these parameters on the code performance were also tested, but the results of this evaluation are not presented here to not overload the discussion.). We took respectively, $ds = 1^\circ$, $r_{\text{max}} = 5^\circ$, $W_a = W_\theta = W_l = 1/3$, $P_\Theta = \Theta = 0.5$, and the ratio of minimum to maximum lengths of the compared filaments had to be greater than $1/3$. Since the application of the tracking algorithm to too small filaments is neither relevant nor efficient, we thus decided to process only the filaments for which the length is equal or greater than 5° .
- ii)* We carried out a series of executions of the code, using different combinations of values for the input parameters a_0 , θ_0 , l_0 , and Δt_0 (reasonable ranges of values were previously defined for each parameter). After each computation, a *positive* tracking rate $[R_p]$ was systematically calculated to evaluate the performance of the automated process compared to the manual one. In order to estimate R_p , we compare two by two the groups of filaments from both manual and automatic sets, which can be identified by their own tracking identification numbers τ_i^{man} and τ_i respectively. For two filaments $[i$ and $j]$ belonging to each type of groups, if the condition $\nu_i^{\text{man}} = \nu_j$ is fulfilled, then R_p was incremented by $1/n$ where n is the total number of comparisons.

Table 1. Results of the assessments for the tracking over one crossing on the solar disk: the *positive* rate R_p in percent (second line) and the corresponding number of filaments N_{fil} (third line), using the best input parameters, are indicated for different lengths of filament l_{ske} (first line).

l_{ske}	$\geq 5^\circ$	$\geq 6^\circ$	$\geq 8^\circ$	$\geq 10^\circ$	$\geq 15^\circ$
R_p	86%	89%	92%	94%	95%
N_{fil}	1991	1698	1263	924	479

iii) Finally, we keep the combination of values of a_0 , θ_0 , l_0 , and Δt_0 that gives the highest *positive* rate.

In the first computation series, we did not take account of the results of the tracking over two rotations including ω_i , then in a second series, we performed the same operation including both τ_i^{man} and ω_i^{man} to compute R_p . Results of the computed series are presented in the following section.

4.2. Results

Table 1 shows the results of a first series of comparisons between the automated and the manual processes. As explained in the previous section, this first series does not take account of the tracking from one rotation to another using ω_i^{man} . The values of the input parameters given the highest rate R_p are in this case: $a_0 = 5^\circ$, $\theta_0 = 30^\circ$, and $l_0 = 4^\circ$. R_p and the corresponding numbers of tracked filaments, which have a length of skeleton $[l_{\text{ske}}]$ greater than 5° , 6° , 8° , 10° , and 15° respectively, are indicated. (R_p is provided in percent.)

According to Table 1, the performance of the code increases when the length l_{ske} increases: passing from 86% of successful tracking for the $N_{\text{fil}} = 1991$ filaments that have a length greater than 5° , to 95% for the $N_{\text{fil}} = 479$ filaments for which $l_{\text{ske}} \geq 15^\circ$. This is the consequence of the algorithm sensitivity to the length of the skeletons, since it is more difficult for the curve-matching to compare filaments that have too small lengths.

Table 2 shows the results of the second series of assessment, including here the tracking results from a rotation to the following. In this case, the input parameters that give the best *positive* rate R_p are : $a_0 = 5^\circ$, $\theta_0 = 20^\circ$, $l_0 = 2^\circ$, and $\Delta t_0 = 112$ decimal days. The total *positive* rate $[R_p]$ (given in percent) and the corresponding numbers of tracked filaments, which have a length of skeleton $[l_{\text{ske}}]$ greater than 5° , 6° , 8° , 10° , and 15° respectively, are provided.

The method also offers good results for the tracking over successive rotations (*i.e.* 90% for $l_{\text{ske}} \geq 15^\circ$), but the performance decreases more rapidly when l_{ske} decreases. We note that the best values of the input parameters are smaller than for the first series, because the algorithm tends to reduce the false tracking occurrences by using more restrictive values on larger features matched.

Table 2. Results of the assessments for two successive rotation tracking: the total *positive* rate R_p in percent (second line) and the corresponding number of filaments N_{fil} (third line), using the best input parameters for different lengths of filament l_{ske} (first line).

l_{ske}	$\geq 5^\circ$	$\geq 6^\circ$	$\geq 8^\circ$	$\geq 10^\circ$	$\geq 15^\circ$
R_p	74%	77%	81%	88%	90%
N_{fil}	1991	1698	1263	924	479

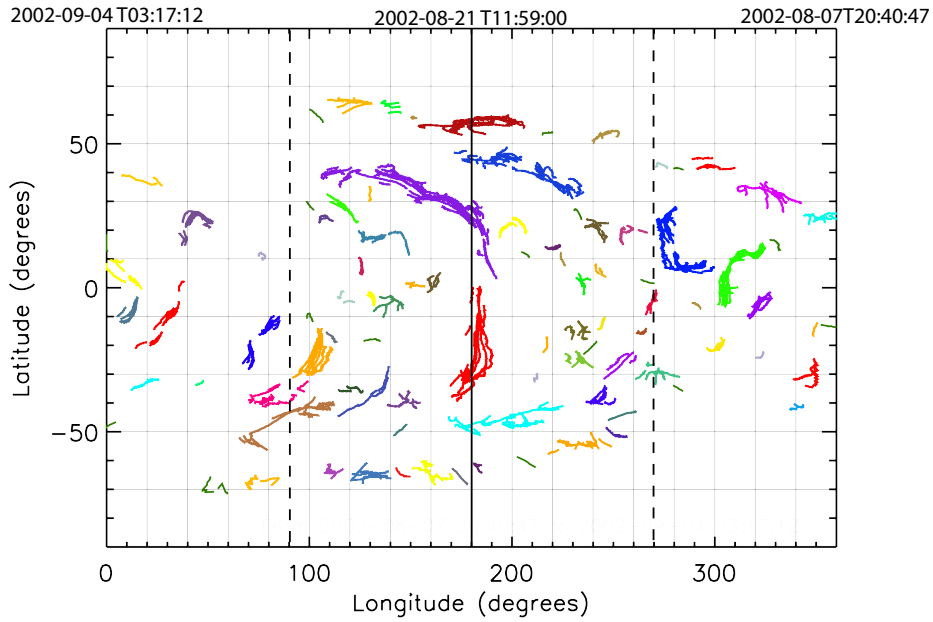


Figure 7. Same synoptic map as Figure 4, but after the execution of the tracking code using the best input parameters found in the first series. Each group of skeletons that actually correspond to the same co-rotating filament (*i.e.*, which have the same tracking identification number) are plotted with one specific color. Only filaments that have a length greater than 5° are tracked, and so plotted on the map. The two black dashed lines delimit the longitude range of the image taken at the time t , and for which tracking is initially performed.

An example of a synoptic map, produced after a run of the tracking code using the best set of inputs parameters found in the first series, is shown on Figure 7. The colors of skeletons are proportional to the corresponding tracking identification numbers, in such a way that associated features appear with the same color on the plot. In most cases, we can notice that the larger features are well grouped by colors.

5. Conclusions

We have presented here a new method to track filaments on solar images. This technique is based on a curve-matching algorithm, applied to the skeletons of filaments in a reference frame rotating with the Sun's surface. The comparison of the automated code results with a representative sample of filaments tracked manually confirms the good performance (close to $\approx 90\%$; depending on the length of the skeletons) of the method. Moreover, the results show the good aptitude of the process to identify the main parts of a segmented filament (as a complementary tool to the detection codes, that often offer such a process on each image). Application of the code to track features over two successive rotations gives also good results, but appears to be less efficient, especially for the smaller filaments; improvements to include smaller filaments cases are in progress.

For now, a version of the code is already used to provide tracking information to the filaments table of the HFC. Stored content only concerns filaments detected on the Meudon H α spectroheliograms, but efforts are currently made to also include BBSO observations. Joint use of these data sets will allow us to improve the ability of the code to follow filaments, by increasing the temporal resolution (Meudon Observatory provides only one image per day on average), but also will refine the analysis of the filament behaviour, since a low cadence limits the capability of the algorithm to accurately detect the occurrence time of events such as filament disappearances.

In addition, several applications of this method are envisaged in the future. Firstly, automated tracking will be a primary step to the creation of filaments synthetic synoptic maps (Mouradian, 1998; Aboudarham *et al.*, 2007) ; these maps provides useful information about solar features to the community. Since the detection and tracking codes can be run independently (and insofar as a dedicated detection code can previously provide the pruned skeletons), investigations are in progress to apply this technique to filament extraction codes working on other wavelengths observations (Buchlin *et al.*, 2010). Finally, an extension to others solar features such as active regions or coronal holes, could be reasonably envisaged using appropriate matching algorithms.

Acknowledgements This work has been supported under the Heliospheric Integrated Observatory HELIO project. HELIO is a Research Infrastructures funded under the Capacities Specific Programme within the European Commission's Seventh Framework Programme (FP7; Project No. 238969). The project started on 1 June 2009 and has a duration of 36 months. In addition, the authors want to thank Z.Mouradian from LESIA for the his valuable help, and the referee for helpful comments. The final publication is available at www.springerlink.com.

References

- Aboudarham, J., Scholl, I., Fuller, N., Fouesneau, M., Galametz, M., Gonon, F., Maire, A., Leroy, Y.: 2007, Automation of Meudon Synoptic Maps. In: Heinzl P., Dorotovič I., Rutten R. J. (ed.) *The Physics of Chromospheric Plasmas, Astron. Soc. Pacific* **368**, San Francisco, 505.

- Ayache, N., Faugeras, O.D.: 1986, HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects. *Pattern Analysis and Machine Intelligence, IEEE Trans. on* **8**, 44–54. doi:10.1109/TPAMI.1986.4767751.
- Barra, V., Delouille, V., Kretzschmar, M., Hochedez, J.-F.: 2009, Fast and robust segmentation of solar EUV images: algorithm and results for solar cycle 23. *Astron. Astrophys.* **505**, 361–371. doi:10.1051/0004-6361/200811416.
- Bentley, R.D., Csillaghy, A., Abouadarham, J., Jacquey, C., Hapgood, M.A., Bocchialini, K., Messerotti, M., Brooke, J., Gallagher, P., Fox, P., Hurlburt, N., Roberts, D.A., Duarte, L.S.: 2011, HELIO: The Heliophysics Integrated Observatory. *Advances in Space Research* **47**, 2235–2239. doi:10.1016/j.asr.2010.02.006.
- Bernasconi, P.N., Rust, D.M., Hakim, D.: 2005, Advanced Automated Solar Filament Detection And Characterization Code: Description, Performance, And Results. *Solar Phys.* **228**, 97–117. doi:10.1007/s11207-005-2766-y.
- Buchlin, É., Mercier, C., Engin, S., Parenti, S., Vial, J.: 2010, Automated detection of filaments in SDO data. In: Boissier S., Heydari-Malayeri M., Samadi R., Valls-Gabaud D. (ed.) *SF2A-2010: Ann. meeting of the French Society of Astronomy and Astrophysics*, Marseille, 297.
- Cox, A.N.: 2000, *Allen's astrophysical quantities*, 4th edn. AIP Press; Springer, New York.
- Fuller, N., Abouadarham, J., Bentley, R.D.: 2005, Filament Recognition and Image Cleaning on Meudon H α Spectroheliograms. *Solar Phys.* **227**, 61–73. doi:10.1007/s11207-005-8364-1.
- Gao, J., Wang, H., Zhou, M.: 2002, Development of an Automatic Filament Disappearance Detection System. *Solar Phys.* **205**, 93–103.
- Gilbert, H.R., Holzer, T.E., Burkepile, J.T., Hundhausen, A.J.: 2000, Active and Eruptive Prominences and Their Relationship to Coronal Mass Ejections. *Astrophys. J.* **537**, 503–515. doi:10.1086/309030.
- Gonzalez, R.C., Woods, R.E.: 2002, *Digital image processing*, 2nd edn. Prentice Hall, Inc., Upper Saddle River, New Jersey.
- Gopalswamy, N., Shimojo, M., Lu, W., Yashiro, S., Shibasaki, K., Howard, R.A.: 2003, Prominence Eruptions and Coronal Mass Ejection: A Statistical Study Using Microwave Observations. *Astrophys. J.* **586**, 562–578. doi:10.1086/367614.
- Higgins, P.A., Gallagher, P.T., McAteer, R.T.J., Bloomfield, D.S.: 2011, Solar magnetic feature detection and tracking for space weather monitoring. *Adv. Space Res.* **47**, 2105–2117. doi:10.1016/j.asr.2010.06.024.

- Joshi, A.D., Srivastava, N., Mathew, S.K.: 2010, Automated Detection of Filaments and Their Disappearance Using Full-Disc H α Images. *Solar Phys.* **262**, 425–436. doi:10.1007/s11207-010-9528-1.
- Krista, L.D., Gallagher, P.T.: 2009, Automated Coronal Hole Detection Using Local Intensity Thresholding Techniques. *Solar Phys.* **256**, 87–100. doi:10.1007/s11207-009-9357-2.
- Lobzin, V.V., Cairns, I.H., Robinson, P.A., Steward, G., Patterson, G.: 2009, Automatic recognition of type III solar radio bursts: Automated Radio Burst Identification System method and first observations. *Space Weather* **70**, S04002. doi:10.1029/2008SW000425.
- Mackay, D.H., Karpen, J.T., Ballester, J.L., Schmieder, B., Aulanier, G.: 2010, Physics of Solar Prominences: II-Magnetic Structure and Dynamics. *Space Sci. Rev.* **151**, 333–399. doi:10.1007/s11214-010-9628-0.
- Mouradian, Z.: 1998, The New ”Solar Activity Synoptic Maps” of Observatoire de Paris-Meudon. In: K. S. Balasubramaniam, J. Harvey, & D. Rabin (ed.) *Synoptic Solar Physics, Astronomical Society of the Pacific Conference Series* **140**, 197–+.
- Russ, J.C.: 2002, *The image processing handbook*.
- Tandberg-Hanssen, E. (ed.): 1995, *The nature of solar prominences, Astrophysics and Space Science Library* **199**.
- Thompson, W.T.: 2006, Coordinate systems for solar image data. *Astron. Astrophys.* **449**, 791–803. doi:10.1051/0004-6361:20054262.
- Ulrich, R.K., Boyden, J.E.: 2006, Carrington Coordinates and Solar Maps. *Solar Phys.* **235**, 17–29. doi:10.1007/s11207-006-0041-5.
- Wolfson, H.J.: 1990, On curve matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **12**(5), 483–489. doi:10.1109/34.55108.
- Zharkov, S., Zharkova, V.V., Ipson, S., Benkhalil, A.: 2005, Technique for Automated Recognition of Sunspots on Full-Disk Solar Images. *EURASIP Journal on Applied Signal Processing* **2005**(15), 2573–2584. doi:10.1155/ASP.2005.2573.
- Zharkova, V.V., Schetin, V.: 2005, Filament Recognition In Solar Images With The Neural Network Technique. *Solar Phys.* **228**, 137–148. doi:10.1007/s11207-005-5622-1.