ADVANCED AUTOMATED SOLAR FILAMENT DETECTION AND CHARACTERIZATION CODE: DESCRIPTION, PERFORMANCE, AND RESULTS

PIETRO N. BERNASCONI, DAVID M. RUST and DANIEL HAKIM
Johns Hopkins University, Applied Physics Laboratory, 11100 Johns Hopkins Road, Laurel, MD 20723, U.S.A.
(e-mail: pietro.bernasconi@jhuapl.edu)

(Received 15 January 2005; accepted 19 February 2005)

Abstract. We present a code for automated detection, classification, and tracking of solar filaments in full-disk Hα images that can contribute to Living With a Star science investigations and space weather forecasting. The program can reliably identify filaments; determine their chirality and other relevant parameters like filament area, length, and average orientation with respect to the equator. It is also capable of tracking the day-by-day evolution of filaments while they travel across the visible disk. The code was tested by analyzing daily Hα images taken at the Big Bear Solar Observatory from mid-2000 until beginning of 2005. It identified and established the chirality of thousands of filaments without human intervention. We compared the results with a list of filament proprieties manually compiled by Pevtsov, Balasubramaniam and Rogers (2003) over the same period of time. The computer list matches Pevtsov's list with a 72% accuracy. The code results confirm the hemispheric chirality rule stating that dextral filaments predominate in the north and sinistral ones predominate in the south. The main difference between the two lists is that the code finds significantly more filaments without an identifiable chirality. This may be due to a tendency of human operators to be biased, thereby assigning a chirality in less clear cases, while the code is totally unbiased. We also have found evidence that filaments obeying the chirality rule tend to be larger and last longer than the ones that do not follow the hemispherical rule. Filaments adhering to the hemispheric rule also tend to be more tilted toward the equator between latitudes 10° and 30°, than the ones that do not.

1. Introduction

Solar filaments are large-scale structures of relatively dense and cool plasma suspended in the hot and thin corona. They can be easily observed in chromospheric Hα images as long, dark threads when seen against the solar disk, and as bright, fuzzy arches at the solar limb (in this case they are usually called prominences). Filaments are always found above polarity inversion lines (PILs) of the photospheric magnetic field and occur within so-called "filament channels", bands in which the chromospheric fibrils are aligned with the PIL (Martin, 1998). This suggests that the magnetic field in filaments channels is mostly parallel to the PIL.

Quiescent filaments often exhibit "barbs" or "feet" extending in an acute angle from the side of the raised dark core, or spine, of the main filament body and descending towards the chromosphere (Martin, Bilimoria, and Tracadas, 1994). Similar to the ramps of an elevated highway, barbs can be categorized as "right-bearing" or "left-bearing" depending on their orientation with respect to the filament spine (Martin, Marquette, and Bilimoria, 1992; Martin, Bilimoria, and Tracadas, 1994; Pevtsov, Balasubramaniam, and Rogers, 2003). Martin and colleagues called "sinistral" the filaments exhibiting a majority of left-bearing barbs and "dextral" the ones showing a majority of right-bearing barbs.

Several authors have suggested that the plasma in filaments is supported by a twisted magnetic flux rope and resting at the bottom of the magnetic coils (e.g., Priest, Hood, and Anzer, 1989; van Ballegooijen and Martens, 1989; Rust and Kumar, 1994; Low and Hundhausen, 1994; Aulanier and Démoulin, 1998). In those models the barbs are explained as dips in the flux rope resulting from the interaction of the low-lying portion of the transversal fields in the twisted flux rope with photospheric parasitic polarities on either side of the filament channel (Aulanier and Démoulin, 1998; Aulanier, Srivastava, and Martin, 2000; van Ballegooijen, 2004). The models also show that dextral filaments are embedded in a *left-handed* helical flux rope, while for sinistral filaments the flux rope has a *right-handed* twist. Therefore, by observing the direction and number of barbs of a filament, it is possible to assign a "chirality" to the filament itself and to infer the sign of the twist in its flux rope.

A number of authors have shown that filaments have the tendency to be sinistral, i.e. the magnetic flux rope has a right-handed twist, in the southern hemisphere and dextral (left-handed twist) in the northern hemisphere (Martin, Bilimoria, and Tracadas, 1994; Rust and Martin, 1994; Martin, 1998; Pevtsov, Balasubramaniam, and Rogers, 2003). This is in agreement with the so-called "hemispheric helicity rule" stating that generally solar magnetic fields have positive helicity, i.e. right-handed twist of the fields, in the southern hemisphere and negative helicity (left-handed twist) in the northern hemisphere.

It is now fairly well established that sudden disappearance (or eruption) of filaments or prominences is usually well associated with coronal mass ejections (CMEs) (Gilbert *et al.*, 2000; Gopalswamy *et al.*, 2003; Jing *et al.*, 2004). Being able to determine the sign of magnetic helicity in erupting flux ropes is important because it can be used to predict the orientation of the magnetic field in the associated CMEs and the probability of a geomagnetic storm if the CME is heading towards Earth (Yurchyshyn *et al.*, 2001; Rust *et al.*, 2005).

In this paper we present an advanced computer code for fast, automated detection, characterization, and tracking of solar filaments from full-disk Hα images. In particular, the code is able to determine filament chirality without human intervention. Our principal goal is to develop a tool that can be used in space weather forecasting to quickly warn about the disappearance of filaments and to help determine the magnetic structure of the associated CMEs. This code can also be used to gain more insight into the structure and evolution of filaments.

A number of groups have successfully developed codes and algorithms for the automated detection of filaments (e.g., Gao, Wang, and Zhou, 2002; Shih and Kowalski, 2003; Zharkova *et al.*, 2004). For example, Jing *et al.* (2004) used the code from Gao, Wang, and Zhou (2002) to automatically select a large number of filament disappearances from a data set spanning 4 years. However, to our knowledge, we are the first to develop a code that not only automatically detects filaments but also determines their intrinsic properties and in particular their chirality, which is of high relevance for space weather studies.

In Section 2 we give a detailed description of the code algorithm, while in Section 3 we describe the test we performed to evaluate the code performance and we discuss its results. In Section 4 we present some first results from a statistical analysis of the data obtained after applying the code to daily Hα images recorded during 4 years. Finally, in Section 5 we present the conclusions and outline our future plans.

To avoid confusion we must note that in the remainder of this paper when talking about the chirality of filaments we refer to the actual *chirality of the magnetic flux rope* in which they are embedded. Therefore, for us a *right-handed filament* is what other authors have frequently defined as a *sinistral* filament, and respectively a *left-handed filament* is what others call a *dextral* filament. We decided to adopt this definition because we believe that it reflects more accurately the actual physical meaning of a filament's chirality and is more closely related to the definition of the sign of magnetic helicity (Berger, 1999).

2. Algorithm for Hα Filaments Detection and Characterization

The algorithm that we have developed is composed of four main modules, which are as follows: (1) image acquisition, (2) image processing, (3) filament detection and characterization, and (4) filament tracking. The nominal procedure uses the four modules in sequence, but they can also be called separately, depending on what the operator wishes to do. Most of the of code is written in Interactive Data Language (IDL) from Research Systems Inc. A few routines, used for calculating solar ephemeris, are written in C language. The entire package runs under a Linux operating system and can be either automatically run daily via a script or accessed via a IDL user interface screen. In the following sections we will present a detailed description of the four modules.

2.1. Image acquisition

The full-disk Hα images are automatically downloaded from the Big Bear Solar Observatory (BBSO) ftp archive (*ftp://ftp.bbso.njit.edu/pub/archive*). See Denker *et al.* (1999) for a description of the instrument, data collection, and calibration. The images are in 'fits' format, 2023 × 2023 pixels in size, and have a resolution of a little less than 1 arcsec per pixel. The solar radius varies from about 890 to 915 pixels depending on the day of the year. BBSO provides two types of images: one without limb darkening correction (so called low-contrast images) and one with the limb darkening removed (high-contrast images), meaning that the chromospheric background level is uniform across the entire face of the Sun. To minimize the

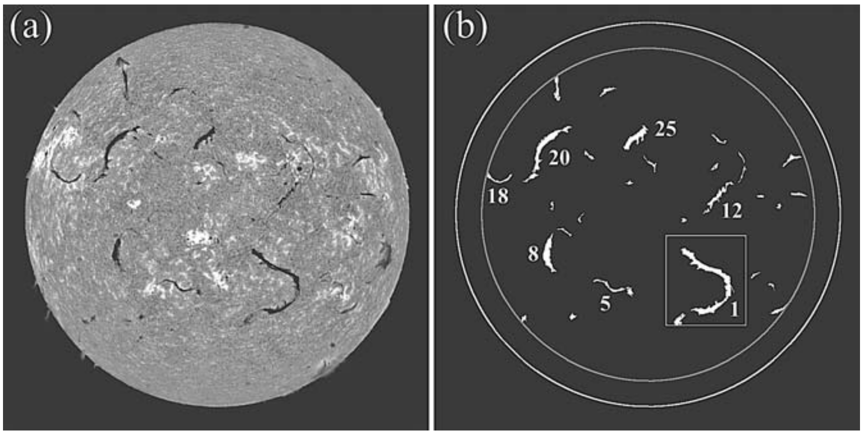

Figure 1. (a) Example of solar Hα image observed at BBSO on February 9, 2002. (b) Filament detection mask produced by the code at the end of the image processing stage. The numbers next to some filaments are their daily identifications numbers assigned to them by the code. The gray inner circle indicates the 60° heliocentric latitude circle, while the rectangle in the low right highlights filament no. 1 that will be used in Section 2.3 as an example for the filament characterization algorithm.

computational load and to take advantage of the already preprocessed data provided by BBSO, the code downloads only the images with the limb darkening already removed. When for the selected day no BBSO images are posted, the code downloads the images taken at the Kanzelhöhe Solar Observatory (Germany), which have the same size even if of somewhat lower spatial resolution. The Kanzelhöhe images are also available in the same BBSO ftp archive as part of the Global High-Resolution Hα Network (*http://www.bbso.njit.edu/Research/Halpha/*). In either case, the code always uses the full-resolution images, i.e., it does not reduce the image size.

After the module has downloaded the images into the local file system, they can be either directly passed to the other modules for immediate processing or processed at a later time. Figure 1a shows an example of a solar Hα image, recorded on February 9, 2002, downloaded from the BBSO ftp archive.

2.2. Image processing

The image-processing module is responsible for standardizing the intensity levels of the downloaded image and for creating a binary *filament mask* in which pixels labeled with 1 are considered part of a filament and the rest is filled with zeroes. The two main techniques used to isolate filaments from the rest of the image are thresholding and morphological filtering. Shih and Kowalski (2003) have described in detail the application of these two techniques for the extraction of filaments from Hα images. In our code we follow similar algorithms, although with some differences. We refer to their article and the references therein for a detailed description on how to perform thresholding and morphological filtering.

2.2.1. *Image Standardization*

To perform a reliable and stable automatic detection of filaments over periods of years, it is critical to have a standardized set of images that is as little dependent as possible on the constantly varying observing conditions. These variations are mostly caused by weather but also by occasional changes in the instrumentation.

The first step is to remove any residual localized inhomogeneities in the background intensity level. In many images we noticed that even though the limb darkening was removed, the background intensity level was still not sufficiently flat to allow a proper application of the thresholding technique for isolating filaments. We believe that most of these inhomogeneities are due to thin clouds in the Earth's atmosphere that locally dimmed the solar disk intensity. To remove them, the code degrades the image resolution by a factor of two and then performs a surface fitting of the solar intensity with a two-dimensional, fourth-order polynomial. The surface fit is then subtracted from the original image, thus, removing quite efficiently most of the inhomogeneities.

The second step is to standardize the image intensity scale. This is necessary because the images do not always exhibit exactly the same contrast. It is done by determining the histogram intensity distribution of the full disk image under consideration and by comparing it with a reference intensity histogram that was obtained from a BBSO image recorded on August 8, 2003. The choice of the reference image was merely dictated by good image quality and relatively low solar activity. The code multiplies each pixel value in the image by a scaling factor that is chosen so to make the intensity histogram match the reference one.

These two image correction steps produce a fairly well-standardized image that is passed on to the next stages of image processing.

2.2.2. *Creation of the Filament Mask*

The code starts by limiting its search for filaments to within a circle of 60° heliocentric latitude from Sun center. Beyond that limit the shape of filaments becomes too distorted because of projection effects, thus rendering their characterization unreliable.

The second step is to remove all sunspots from the mask. We follow a method suggested by Shih and Kowalski (2003). Sunspots are usually darker than filaments and can be easily detected via thresholding with a *spot threshold* that is lower than the lowest possible intensity level that filaments can have. After some experimenting, we determined that, in general, pixels with an intensity value below -3000 only belong to sunspots. The spot pixels marked in this way are used as seeds for a region-growing morphological operation. With this operation each spot area is grown by adding adjacent pixels until the pixel brightness in the standardized image exceeds a value that is *higher* than the threshold level for detection of filaments (the *filaments detection threshold*, discussed later). All values within the spot pixels areas in the standardized image are set to a value higher than the *filament detection threshold*. This method of sunspot elimination works fairly well, although it is not free from problems. We noticed that particularly large and thick filaments can have their lowest intensity levels actually below the *spot threshold* and therefore they can be wrongly identified as spots. To avoid this problem, after the region growing process, the code checks the size of each spot found. If the area of a *spot* is larger than 2000 pixels then the code recognizes that it is too large to be a spot (even very large spots are always smaller than 2000 pixels) and resets its pixel intensities to the original values, therefore allowing it to be later correctly detected as a filament. The second problem is that if the sunspot is very small it will not be deep enough to reach the *spot threshold* and it will not be detected as a spot. This problem will be handled later when all detected filaments smaller than 300 pixels are eliminated from the mask. Usually all these small spots are smaller than 300 pixels.

After spot elimination the code can proceed to the actual creation of the *filament mask*. To extract the filaments the code once again applies the thresholding technique. First, all pixels with a brightness level below the *filament detection threshold* of -600 are identified and given the value of 1 in a mask that originally is filled with zeroes. Unfortunately, this operation detects not only all the relevant filaments in the image, but also several spurious pixels and small areas that are not actual filaments. These are mainly places where there are large localized fluctuations of the background intensity. To filter out the unwanted regions the code applies an *advanced* morphological filtering operation that was also proposed by Shih and Kowalski (2003). The main idea is that filaments are by nature elongated shapes. These shapes can be efficiently isolated from other small and rounder structures that are not filaments, by separately applying to the filament mask eight opening morphological operations with the eight *linear structuring elements* depicted in Figure 2. Pixels that "survive" at least two of such opening operations are considered belonging to real filaments (see Shih and Kowalski, 2003 and references therein for a detailed description of this technique). These "surviving" pixels are used as seeds in a region-growing morphological filter operation. The regions identified as filaments are expanded until the pixel intensity level in the standardized Hα image surpasses a *second filament threshold* value of -500.

Once all filaments are detected and grown to their full shape, the code checks their pixel area and removes from the mask the ones whose area is less than 300 pixels. This is done to avoid the further processing of filaments that are either too small for a meaningful characterization or actually false detections, like occasional large clumps of dust or other imperfections in the optics of the telescope. This area filtering also takes care of removing small sunspots that were not filtered out by the

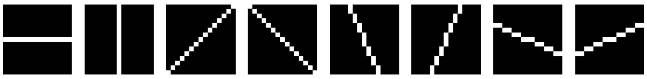

Figure 2. The eight directional 15 × 15 pixels linear structuring elements used by the advanced morphological filter to remove spurious pixels from the filaments mask. White pixels have value 1 and back are zeroes.

spot removal algorithm. The final result is a mask in which all pixels belonging to filaments within a heliocentric angle θ of 60° have value 1 and elsewhere is filled with zeroes. Figure 1b shows the filament mask obtained from the Hα image in Figure 1a. The numbers next to a few of the large filaments in the figure are their daily ID numbers that the code automatically assigns to them (see also Table I).

2.3. Filament detection and characterization

This module is responsible for determining the following main filament characteristics: position, length, area, average tilt of axis with respect to the Sun's equator, and most importantly chirality of the magnetic flux rope in which it is embedded.

A filament is identified as such by considering the cluster of all adjacent pixels in the mask, i.e., all the pixels touching each other either on the sides or on the corners. Each separate cluster receives an identification number and is considered as an individual filament and processed as such. In Section 2.3.6 we will discuss the technique used to identify segmented filaments and to merge their components.

We have segmented the algorithm into six separate steps that are applied in sequence to each individual filament found in the *filament mask*.

2.3.1. *Determination of the Filament's Boundary*

At first, the code determines the boundary of the cluster of pixels. It is simply an array that stores the Cartesian coordinates of each pixel along the outline of the cluster. It is determined by starting at the pixel in the lowest left end of the boundary and then by finding the next adjacent boundary pixel moving around in a counterclockwise direction until back at the starting pixel. The boundary array will be used later to find the location and direction of the filament's barbs (see Section 2.3.3).

2.3.2. *Determination of the Filament's Spine*

To trace the filament's spine the code applies a principal curve algorithm. Originally, we used a simplified version of the Kégl *et al.* (2000) algorithm for finding the principal curves defined by sets of points. However, during the development phase of this project, to make this algorithm faster and more suitable to our needs we have modified it so much that the final version retains only minor similarities to the original one presented in Kégl *et al.* (2000).

The algorithm uses a multi-step iterative technique. As an example we show in Figure 3 the algorithm applied to the long filament marked as no. 1 inside the box in Figure 1b.

The first iteration starts by determining a first guess of the location of the two spine end points V_1 and V_2. This is done by considering a rectangular box that tightly encloses the filament. The filament main axis is assumed to run roughly parallel to the longest side of the box. The first two vertices defining the spine are

TABLE I

Filament summary table generated by the code after processing the Hα image shown in Figure 1.

No.	Area	Xp	Yp	Lat	Lng	Ang	Len	Tb	#*R*	#*L*	Ch
1	13515	285	−256	−22.41	19.65	86.0	656	13	1	10	1
2	474	−590	−493	−36.67	−53.47	58.2	47	3	1	2	0
3	541	−356	−482	−37.53	−29.34	20.3	50	1	1	0	0
4	964	627	−454	−33.63	55.36	27.8	99	1	0	1	0
5	2399	−173	−357	−29.34	−12.50	−18.5	228	5	2	3	0
6	304	595	−339	−26.34	46.46	−89.6	47	2	2	0	−1
7	375	510	−284	−23.33	37.31	25.1	56	0	0	0	0
8	4398	−483	−180	−16.84	−33.41	−85.3	206	3	1	1	0
9	549	−394	−78	−10.80	−25.94	−24.4	79	1	1	0	0
10	328	168	−29	−8.27	10.67	34.2	39	3	2	1	0
11	302	−322	−14	−7.02	−20.72	72.5	56	2	1	0	0
12	2971	337	77	−1.29	21.57	40.8	233	12	4	7	1
13	304	−454	41	−3.14	−29.73	85.7	41	2	1	0	0
14	326	625	38	−2.42	43.06	86.3	33	2	1	1	0
15	318	535	87	0.12	35.72	−12.1	35	0	0	0	0
16	723	719	97	2.03	51.79	−12.9	79	2	1	1	0
17	330	−89	152	3.00	−5.58	2.2	43	1	1	0	0
18	1382	−716	157	5.84	−51.82	−2.2	150	2	0	1	0
19	802	459	225	8.56	30.43	−87.2	158	1	0	1	0
20	7030	−521	301	13.89	−35.86	48.6	383	10	6	4	−1
21	732	10	246	8.99	0.63	−55.6	107	3	2	1	0
22	486	318	225	8.06	20.50	43.4	50	1	0	1	0
23	1212	680	267	12.75	49.59	28.7	107	2	1	1	0
24	610	−283	281	11.62	−18.37	−41.1	62	2	0	2	1
25	3988	−53	362	16.70	−3.46	52.8	161	9	3	5	1
26	643	330	356	16.78	22.09	−23.6	86	3	2	1	0
27	1795	−432	603	35.96	−35.66	−86.7	124	5	4	0	−1
28	964	−195	595	34.17	−14.91	25.1	94	4	3	0	−1

The example filament from Figure 3 is listed in the first row (no. 1). Area: area in pixels, (Xp, Yp): location in Cartesian coordinates (pixels), (Lat, Lng): location in heliographic coordinates, Ang: average tilt angle with respect to equator (°), Len: total length of spine (pixels), Tb: total number of barbs detected, #*R*: number of right-bearing barbs, #*L*: number of left-bearing barbs, Ch: filament chirality.

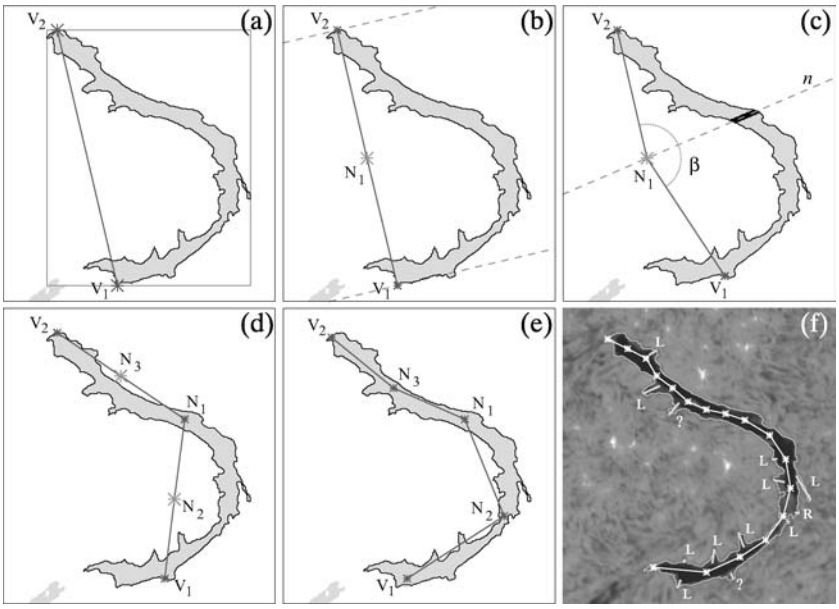

Figure 3. Algorithm to determine the filament's spine. (a) Determination of first two vertices V_1 and V_2. (b) Introduction of a new vertex N_1 and optimization of the end-points locations. (c) Optimization of the new vertex N_1. β is the angle between vertices $V_1N_1V_2$, and the dashed line *n* bisects β and passes through N_1. (d) Second iteration: two new vertices N_2 and N_3 are added. (e) End of second iteration. (f) Spine at the end of final iteration. Part (f) also shows the location and type of the barbs detected (see Section 2.3.3). All the steps are described in the text.

therefore set to be the two pixels touching the shortest sides of the box, as shown in Figure 3a.

Next, a new vertex N_1 is added in the middle of the segment connecting the two vertices. The code then optimizes the location of the first three vertices so far determined. It starts by adjusting the location of the two spine end points V_1 and V_2. For each end point it locates all the pixels belonging to the filament that lie on the side of a line perpendicular to the direction of the spine segment end point facing "away" from the spine (see Figure 3b). The new end-point location is the centroid of those pixels. In the example shown in Figure 3, end point V_2 migrates to the right, while V_1 does not move because no pixels "outside" the spine are found (see Figure 3c). Even though V_2 at first moves in the "wrong" direction, in subsequent iterations it will migrate towards the correct location, thanks to a better estimation of the entire spine.

To optimize the location of the middle vertex N_1 the code calculates a line *n* that bisects the angle β between the two segments connecting N_1 and its nearest vertices (V_1 and V_2 in this case) and passing through the vertex N_1 itself (see Figure 3c). It then calculates the centroid of all the pixels inside the filament that are located to within a distance of three pixels from that line (see black area inside the filament in Figure 3c). The centroid location becomes the new vertex position and this ends the first iteration (Figure 3d).

The next iterations repeat the process by first adding new vertices in the middle of each spine segment (e.g., N_2 and N_3 in Figure 3d) and then by re-optimizing *all* vertices, end points included (Figure 3e). The iterative process stops when the length of the smallest segment in the spine is less than 20 pixels (Figure 3f). The coordinates of the locations of the spine vertices are stored in a *spine array*.

2.3.3. *Detection of the Filament's Barbs*

To detect the barbs, the code first calculates the distance of each pixel in the *boundary array* from the spine. As the plot in Figure 4b shows there are several peaks and dips in the distance from the spine. We interpret the highest peaks as possible barbs because they indicate the presence of a structure along the boundary that extends away from the average filament body.

The second step is to register the location of all peaks in the distance plot that are higher than 80% of the median distance of the boundary pixels from the spine. For each peak, the code then follows down the distance curve on each side until it reaches the two lowest points. We call those the barb's base points and the line connecting them the *base* of the barb. We define as the barb *axis* the line connecting the barb peak with the middle point of the barb base (Figure 4a).

Once the barb axis and base are determined, the code proceeds with two tests to establish whether the isolated structure can indeed be accepted as a barb or instead is something else, like a broad bend in the filament for instance. The first test is quite simple: if the barb axis is less than three pixels long, the structure is considered too small and immediately rejected. The second test relies on the empirical observational fact that barbs are elongated structures in which the barb's base can not be too broad compared to the barb's height. The test is passed if

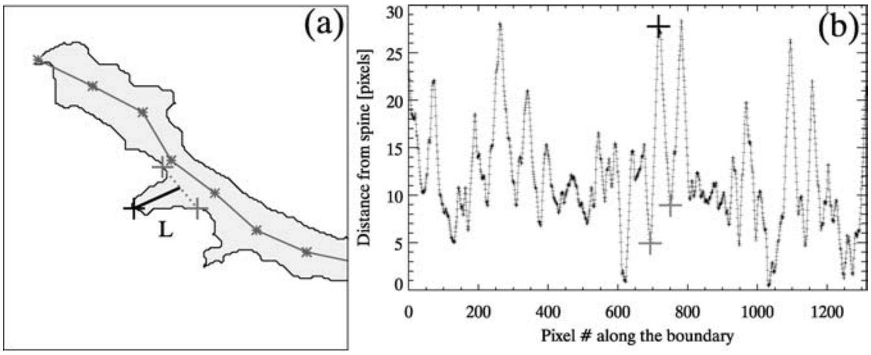

Figure 4. (a) Detail of the upper part of the example filament highlighting the location and direction of a barb, with its axis (*thick, dark line*) and the three points used to determine it. (b) Plot showing the distance (in pixels) between each boundary pixel and the spine. The *three large crosses* indicate the location in the plot of the three points highlighted in (a).

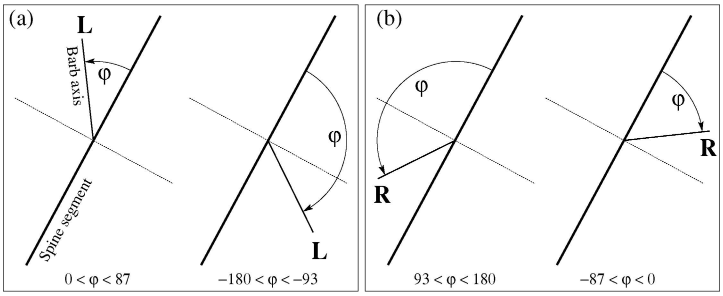

Figure 5. Sketch describing the rules used to determine a barb direction with respect to the spine. (a) For *left*-bearing barbs. (b) For *right*-bearing barbs. The angle φ is measured starting from the spine segment direction towards the barb axis and is defined positive in the counter-clockwise direction.

the ratio *axis-length/base-length* is greater than 0.3. After several trials and after analyzing the properties of hundreds of barbs we have determined that these two simple tests, with the limit values mentioned earlier, are very effective and reliable in determining whether the detected structure is indeed a barb or not. However, it is quite probable that more sophisticated tests can be applied to achieve even more reliable results. We will continue investigating this aspect in an attempt to improve the performance of the algorithm.

Once the code has positively identified a barb, it proceeds in determining whether it extends from the filament with a bear-right or a bear-left direction. The code calculates the angle φ of each barb's axis with respect to the segment of the spine closest to the barb as depicted in Figure 5. A positive φ is defined as counter-clockwise from the spine. If φ is within $\pm 3^\circ$ from the normal to the spine the code declares it too-close-to-call and marks such barb as undetermined. If $0^\circ < \varphi < 87^\circ$ or $-180^\circ < \varphi < -93^\circ$ (quadrants I and III; see Figure 5a) the barb bears to the *left*. Instead if $93^\circ < \varphi < 180^\circ$ or $-87^\circ < \varphi < 0^\circ$ (quadrants II and IV; see Figure 5b) the barb bears to the *right*. In the example shown in Figure 4 the barb is clearly bearing to the left and the code correctly identifies it as such. This is done for each positively identified barb.

2.3.4. *Estimation of the Filament Chirality*

To determine the chirality of the flux rope in which the filament is embedded the code computes the difference between the number of right-bearing barbs and the number of left-bearing barbs and it assigns the chirality according to the following truth table:

$$\begin{aligned}
&\#R - \#L \geq 2 && \Rightarrow \text{LEFT-handed helix} && \Rightarrow \text{chirality} = -1,\\
&\#R - \#L \leq -2 && \Rightarrow \text{RIGHT-handed helix} && \Rightarrow \text{chirality} = 1,\\
&|\#R - \#L| < 2 && \Rightarrow \text{UNDETERMINED} && \Rightarrow \text{chirality} = 0,
\end{aligned} \tag{1}$$

where $\#R$ and $\#L$ is the number of right-, and left-bearing barbs, respectively.

If the difference is less than 2, the code declares the case unclear and it assigns an undefined chirality to the filament. Note that here we define the chirality simply as a number that if positive (negative) it means *right-handed* (*left-handed*) and if zero no specific chirality is assigned. It does not, by any means, imply the *actual* amount of twist in the magnetic flux rope. The amount of twist can not be determined with this simple technique. Only the flux rope handedness can be established. For our example, the filament in Figure 3, the code found a majority of left-bearing barbs and correctly identified it as a right-handed filament (see Figure 3f and Table I, first row).

The number of right- and left-bearing barbs, and the chirality value derived from Equation (1) are entered in a table that summarizes all relevant filament parameters (Table I).

2.3.5. *Determination of Other Filament Parameters*

The filament summary table mentioned earlier also contains a number of other parameters: the filament area which is simply the number of pixels within the filament boundary; the filament length which is the sum of the length in pixels of all the spine segments; the average orientation (or tilt) of the filament spine with respect to the Sun's equator, which is the average of the angles of all spine segments. Finally, the code calculates the "center" of the filament. It is determined by first establishing the center of the box tightly enclosing the filament and then by locating the point along the spine that is closest to it. In the table it is given in both Cartesian pixel coordinates and in heliographic latitude and longitude coordinates.

2.3.6. *Merging of Segmented Filaments*

The last step in the filament detection and characterization module is to check whether there are segmented filaments, and if so, to merge them into one.

The code uses two approaches. The first is to calculate the distance of *each* spine end point of *each* filament to *all* other filaments spine end points. If two different filament end points are found to be less than 25 pixels apart, the two filaments to which they belong are considered the same and they are merged. If the distance between different filaments end points is more than 25 pixels but less than 100 pixels the second approach is applied. The code tries to see if the end-segments of the two filaments have an orientation similar to the line connecting the two end points themselves. The idea is that if such a case arises the two filament segments are probably pointing at each other and therefore are part of the same structure. This test is executed by calculating the angle between the last segments of the two spines relative to the line connecting the two end points. If for *both* segments this angle is less than 22.5° then they are considered segments of the same filament and are merged. This double test technique is not perfect but in more than 70% of the cases it correctly detects segmented, or even multi-segmented, filaments and it is conservative enough to avoid any false detection.

The actual merging is done by adding a spine segment between the two nearby end points and by recalculating (via summation or average) all filament parameters.

Finally, the code saves in the computer file system a table in ASCII format with all the filament parameters. Optionally, it also generates an image in PNG format showing the original Hα image with for all the detected filaments a trace of their boundary, spine, detected barbs, and a print of the barbs statistics (#*L*, #*R*, and chirality). Table I shows the output table generated by the code after processing the image in Figure 1. It detected in total 28 filaments. The filament example shown in Figure 3 is listed in the first column (no. 1). The filaments with daily ID numbers 5, 8, 12, 18, 20, and 25 are also marked in the mask in Figure 1b.

With our "rather slow" machine (with a Pentium III processor running at 600 MHz clock speed) the typical processing time is about 2 min. This does not include the time to download the fits file from the ftp archive. Our current entire database of about 1400 images, which covers a span of almost 4 years, can be fully processed in approximately 48 h.

2.4. Filament tracking

This is the last module called by the code. Its purpose is to track the day-by-day evolution of the detected filaments and to compose a *filaments tracking table* that stacks in subsequent rows all the daily entries relative to a specific tracked filament.

The module can track filaments in either temporal direction, into the future or into the past. Here, to describe the algorithm, we use only the case of tracking into the future. In this example, we assume that this module is used *after* having run the other three modules for a series of several adjacent daily images and having obtained all the relative filament summary tables. We want to track the day-by-day evolution of the filaments detected in the first image of the series.

First, the code calculates the time difference ΔT (in days) between the first and the second table. Second, it reads from the first table the location of each filament in heliocentric latitude and longitude and it calculates where each filament should be located after the time ΔT is passed according to the following formula for synodic rotation of filaments (Allen, 2000):

$$\delta' = \delta + \Delta T(14.48 - 2.16\,\sin^2(\phi)). \tag{2}$$

ϕ and δ are the latitude and longitude (in degrees), respectively, of a point on the Sun, and δ' is the new longitude location of the same point after the time ΔT (in days) has passed. The code proceeds by searching the second filament table to see if any filament location lies within a 5° circle from the predicted location (α, δ'). If a match is found the two filaments are considered the same and the data of both filaments are entered in a *tracking table*, an ASCII file stored in the computer file system. The code continues by looking for matches between the filaments in the second and third table. If something is found it adds a new entry in the tracking

TABLE II

Fragment of the *tracking table* showing the day-by-day evolution of filament number 4278, which is used as an example in Section 2.3.

Date	Time (UT)	Num	Area	Lat	Lng	Len	Ang	Ch
02/04/2002	18:16:39	6	6684	−19.62	−48.16	323	−30.40	1
02/05/2002	17:32:06	8	7559	−20.11	−34.65	359	−31.50	0
02/06/2002	16:59:49	7	9031	−20.71	−20.97	399	−34.70	1
02/07/2002	17:53:50	6	8727	−20.28	−10.98	539	62.40	1
02/08/2002	17:28:59	3	12230	−20.90	3.28	630	−32.10	1
02/09/2002	18:15:38	1	13515	−22.41	19.65	656	86.00	1
02/10/2002	17:40:11	6	6770	−22.72	32.18	324	−51.20	1
02/11/2002	17:45:23	4	6051	−22.45	47.63	251	−51.80	0

Num in the third column means the temporary number the code gave to the filament for that specific day. It changes from day-to-day depending on how many filaments were detected on each day.

table under the row relative to the filament being tracked. The process continues until no further matches are found in the next filament tables.

If for a specific filament no matches are found between the first and second table the code searches in the third table to see if a match is found there. The search is extended to up to 3 days in the future, or until the predicted location falls beyond the 60° filament detection limit. If after three days the filament is not found again, it is considered disappeared. The 3-day search is motivated by the fact that sometimes filaments change shape so much between times of observation that their location may fall out of the 5° search circle temporarily loosing the tracking.

The usual operation mode is to run the tracking module once all the images for a specific period are processed. When a filament is first seen, a unique filament identification number is assigned to it to distinguish it from any other filament detected in the time series, and to help its day-by-day tracking.

Table II is a small fragment of the huge tracking table generated by the tracking module. It shows the evolution of the filament used as an example in Section 2.3. When the code first detected it on February 4, 2002 near the eastern limb of the solar southern hemisphere, it received the ID number 4278 and was entered in the tracking table. It kept growing until it reached a maximum size around February 9, 2002. After that it rapidly decayed until on February 12, 2002, it dropped out of the 60° detection window. The filament had consistently a right-handed chirality, as the majority of filaments in the southern hemisphere do. There are however a few days when the code was unable to determine its chirality. This is only a fraction of the tracking table, which contains entries for almost 9500 filaments.

3. Performance Evaluation

To test the code performance we compared its output with a human-generated list of filament properties compiled by Pevtsov, Balasubramaniam, and Rogers (2003), which from now on we will designate as Pevtsov's list. The list spans from July 15, 2000 to June 22, 2002. It contains daily information about filaments locations and how many right- and left-bearing barbs they have. The list does not contain any tracking information, meaning that if a filament appears in multiple days it will also have multiple entries in the table.

We have considered almost 1 year's worth of data: from July 15, 2000 to June 15, 2001. We manually compared daily results for filaments matching the same location in both Pevtsov's list and the computer-generated list. We found 997 filaments present in both lists. Five hundred of these were located in the northern hemisphere and 497 in the southern hemisphere. To determine the chirality of the filaments in Pevtsov's list, we applied the same truth table used by our automated code (Equation (1)).

The code could not determine the chirality of 339 filaments, while in Pevtsov's list only 17 had undetermined chirality. Of the 658 filaments to which the code assigned a chirality, 473 had a positive match with Pevtsov's list. If we assume that Pevtsov's list is 100% accurate, the code has a 72% success rate in assigning the correct chirality. However, a man-made list is susceptible to inaccuracies mainly due to person-to-person differences in judging how many barbs a specific filament has and in which direction they are pointed. For example, a close inspection of Pevtsov's list revealed some clear differences in different parts of it, suggesting that the list may have been compiled by multiple people.

Table III summarizes the results of the comparison between the two lists. The table separates the results into the two hemispheres. In general, the most striking difference is that the code finds significantly more filaments with undetermined

TABLE III

Summary of the results from the comparison between the filament list generated by the automated code and Pevtsov's list.

	Northern hemisphere					Southern hemisphere				
	?	R	L	$R+L$	$\%L$	?	R	L	$R+L$	$\%R$
Automated code	170	99	231	330	70	169	216	112	328	66
Pevtsov	8	91	401	492	82	9	422	66	488	86

Totals are separated by hemisphere. Columns marked with "?" denote the total number of filaments without a defined chirality. "R" and "L" indicate total number of right- and left-handed filaments, while "$R + L$" is the sum of the two values. "$\%L$" is $L/(R + L) \times 100$ and "$\%R$" is $R/(R + L) \times 100$.

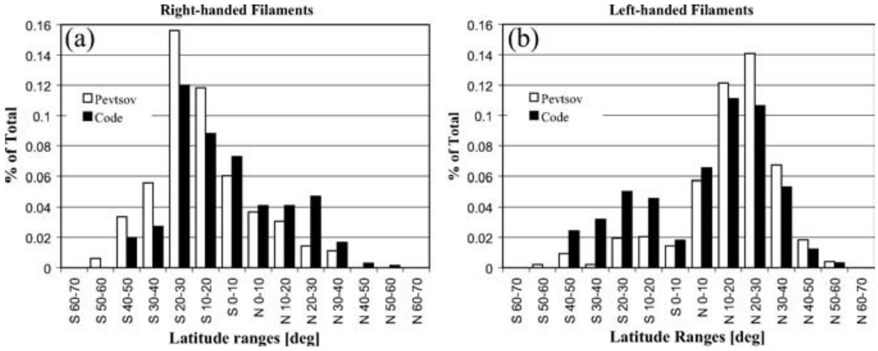

Figure 6. (a) Hemispheric distribution in 10° latitude bands of the fraction of *right*-handed filaments with respect to the total, as derived from the code filaments list (*black*) and Pevtsov's list (*white*). (b) Same as in "a" but for the *left*-handed filaments.

chirality than Pevtsov's list. We believe that one reason is because the code is only able to detect clearly visible barbs, while a human eye is capable to see finer, more subtle barbs. This limitation greatly affects the barbs statistics especially for small, skinny filaments for which the code can detect only a few barbs. In those cases, the result is almost always an undetermined chirality (see Section 4 and Figure 9 for a more detailed discussion). We believe another reason for this discrepancy may once again be due to a human tendency to be biased. An operator aware of the hemispheric helicity rule may be induced to for example "see" more right-bearing barbs in a filament located in the northern hemisphere also in not so clear cases. On the other hand, the code, even with all its limitations, is totally unbiased.

Table III also shows that both lists confirm the hemispheric helicity rule, i.e., filaments embedded in a magnetic flux rope with a left-handed twist are predominantly observed in the northern hemisphere and vice versa for the opposite twist. For Pevtsov's list this was already pointed out in Pevtsov, Balasubramaniam, and Rogers, (2003). It is worth noting, however, that for the code-generated list the rule is less clear. Figure 6 demonstrates that the hemispheric distribution of right- and left-handed filaments derived from the code list, and Pevtsov's list are very similar. Once again, the code's distribution tends to show the hemispheric helicity rule somewhat less clearly.

4. Results

We let the code automatically download and process BBSO Hα filtergrams for the period from July 6, 2000 to January 9, 2005 (1411 images). It detected and analyzed a total of 19 211 filaments without human intervention. After running the tracking module, the total number of individual filaments actually seen during the aforementioned period reduced to 9459. For these tracked filaments, we computed their individual average properties as well as their lifetime, i.e., the total time each

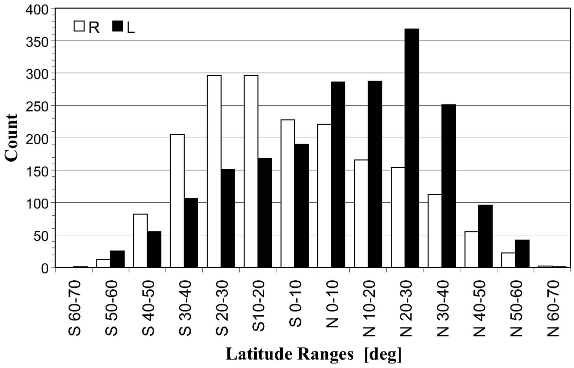

Figure 7. Hemispheric distribution of the chirality for the 9459 filaments detected by our automated code during the period from July 6, 2000 to January 9, 2005. *White* (*black*) columns denote the number of right-handed (left-handed) filaments within each 10° latitude band.

single filament was seen (or detected) on the visible side of the Sun. If a filament was detected only once, we assigned no lifetime to it. To determine the average chirality of a specific filament, we calculated the sum of all the chirality values (+1, −1 or 0) for all the days in which it was seen and we assigned to it a chirality value of +1 if the result was a positive number, respectively −1 if the result was negative, and zero otherwise.

Figure 7 shows the hemispheric distribution of the right-handed (white columns) and left-handed (black columns) filaments during the four and a half years period mentioned earlier. Once again, the hemispheric helicity rule is quite obvious even though less clear than what is reported in Pevtsov, Balasubramaniam, and Rogers, (2003).

Figure 8 shows the hemispheric variation of the filaments lifetimes averaged over latitude bands of 10° for the three different chirality types. From the plot it

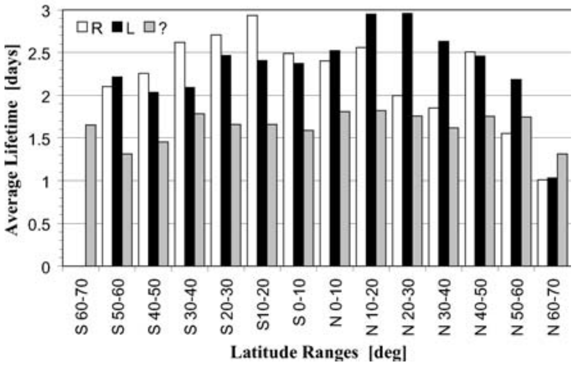

Figure 8. Observed lifetime averaged on each 10° latitude band for right-handed (*white*), left-handed (*black*), and undetermined (*gray*) filaments. Only filaments that were seen in at least two consecutive images are considered.

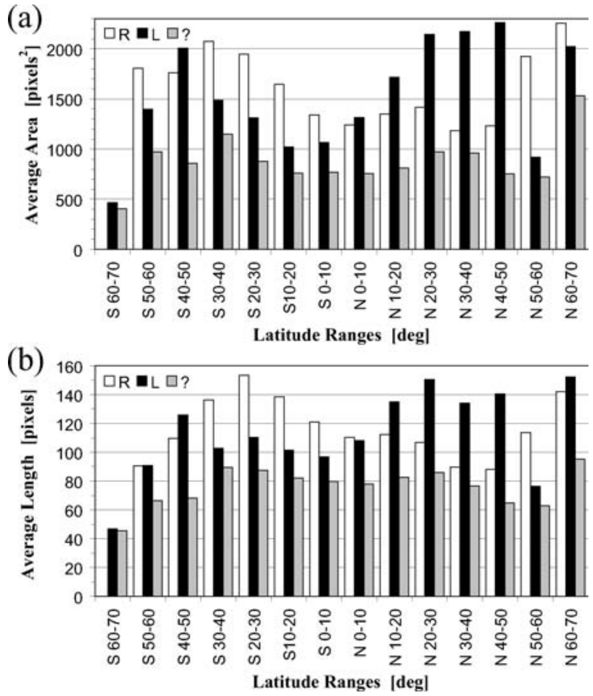

Figure 9. Hemispheric distribution of filament areas (a) and lengths (b) averaged over bands of 10° latitude for right-handed (*white*), left-handed (*black*) and undetermined (*gray*) filaments.

is clear that the filaments with undetermined chirality consistently have a shorter lifetime than the ones with a definite chirality. The filament lifetime peaks in a band between latitudes 10° and 30° for both hemispheres. More interestingly though, the distribution shows quite clearly that filaments in the southern hemisphere with right-handed chirality (the ones obeying the hemispheric magnetic chirality rule) have the tendency to be more persistent than the ones having left-handed chirality in the same hemisphere. The behavior is reversed in the northern hemisphere.

We can see something similar also when we consider the size (area or length) of filaments averaged over the same 10° latitude bands. Figure 9 shows that right-handed filaments in the southern hemisphere tend to cover a larger area and to be longer than the left-handed filaments in the same hemisphere. The opposite is true in the northern hemisphere. Also noteworthy is that filaments with undetermined chirality are on average smaller than the ones with a defined chirality at all latitude bands. We believe that this is in part an artifact caused by the limitations of our automated detection code. If filaments are very small or skinny the code has a hard time seeing the small barbs, and this causes problems in the determination of the filament's chirality.

By comparing Figure 9a and b one can see that the size discrepancy between filaments with clear chirality and the ones without is much larger for the area,

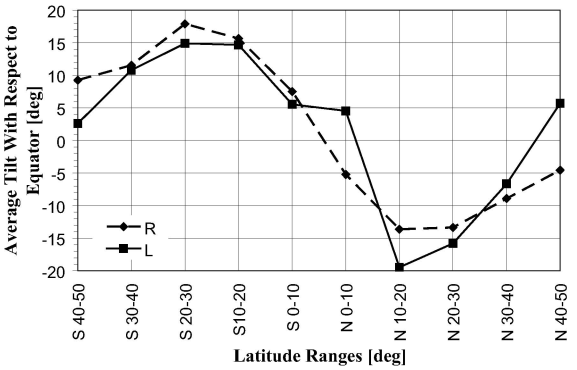

Figure 10. Filament tilt with respect to the equator averaged over all filaments within the different 10° latitude bands. *Diamonds* are for right-handed filaments, and *squares* for left-handed filaments.

rather than for the length. This leads us to conclude that indeed the majority of the filaments with undetermined chirality are the skinny ones. Upon visual inspection we noted that filaments in this class usually show one or no barbs at all.

Incidentally, the long and skinny filaments appear mostly above neutral lines in active regions, as opposed to the so-called quiescent filaments that are not associated with a particular active region. Quiescent filaments are, in general, fatter and more structured than active region filaments. Our code is presently not able to distinguish between active region filaments and quiescent ones.

Figure 10 shows the hemispheric distribution of the filaments tilt angle when averaged over the different 10° bands. A positive tilt angle means that on average the filament leading end is pointing northward, and a negative angle indicates that the leading end is pointed southward. For both types of chirality the highest tilt is reached by filaments in the mid-latitudes between 10° and 30°. From the configuration of the magnetic switchbacks in either hemisphere it is to be expected that the leading end of a filament points on average towards the equator. However, it is interesting to notice that, once again, also for the tilt angle there is an asymmetry between right- and left-handed filaments. At mid-latitudes in the southern hemisphere right-handed filaments have a tendency of being *more* tilted towards the equator than the left-handed filaments. The opposite happens in the northern hemisphere.

5. Conclusions and Future Work

We have developed a computer code for the automatic detection, characterization, and tracking of filaments from full-disk Hα images. For each detected filament the code provides location, area, length, average spine tilt with respect to the equator, and most importantly its chirality. The code is also capable of tracking the day-by-day evolution of each filament.

We have tested the code's performance by comparing the automatically generated filament properties list with a man-made list provided by Pevtsov, Balasubramaniam, and Rogers, (2003). The automated code can determine the correct chirality in more than 70% of the cases, assuming that Pevtsov's list is absolutely accurate. The result confirms the hemispheric magnetic helicity rule. The main difference is that the code finds considerably more filaments with undetermined chirality. This may be in part due the code's difficulty to detect all small barbs extending from the core of small or skinny filaments. It could also be attributed to a tendency of human operators to be biased, thereby assigning a chirality in less clear cases, while the code is totally unbiased.

We believe to be the first to discover a new hemispheric rule similar to the magnetic helicity one but applicable to filaments lifetime, size and tilt. The new rule may read like this: *Right-handed filaments located in the southern hemisphere have the tendency to be larger, live longer, and be more tilted towards the equator than the left-handed filaments in the same hemisphere. The opposite is true in the northern hemisphere*. This interesting and intriguing new finding was made possible by the development of a reliable automated filament detection and characterization code.

We continue expanding our database by running the code twice a day.

For the future, we plan to first develop an online interface where anyone can view the latest Hα image processed by the code, check the current filament tracking status, and browse through all the past processed data. We also plan to develop a filament disappearance, detection, and warning system that we will eventually incorporate in a Web site. This capability can be used by space weather forecasters when trying to determine the magnetic field direction of an erupted flux rope before it reaches Earth. We will also continue to develop techniques to improve the code performance and reliability. In particular, we will develop more sophisticated methods for detecting barbs.

The capability to detect filaments when they appear and to track their evolution can provide not only early warnings of potentially hazardous conditions but also improve our understanding of solar filaments and their implications for space weather at 1 AU.

Acknowledgements

We thank the Big Bear Solar Observatory/New Jersey Institute of Technology and the Global High Resolution Hα Network for providing and maintaining the ftp data archive. We are grateful to Alexei Pevtsov for kindly making available to us his list of filaments properties. We thank Kiri Wagstaff for her great contribution in the initial development of the code. We also thank Valentina Abramenko for reading the manuscript and for her helpful comments and suggestions.

References

Allen, C. W.: 2000, *Allen's Astrophysical Quantities*, Springer-Verlag, New York, p. 362.

Aulanier, G. and Démoulin, P.: 1998, *Astron. Astrophys.* **329**, 1125.

Aulanier, G., Srivastava, N., and Martin, S. F.: 2000, *Astrophys. J.* **543**, 477.

Berger, M. A.: 1999, *Plasma Phys. Control. Fusion* **41**, B167.

Denker, C., Johannesson, A., Marquette, W., Goode, P., Wang, H., and Zirin, H.: 1999, *Solar Phys.* **184**, 87.

Gao, J., Wang, H., and Zhou, M.: 2002, *Solar Phys.* **205**, 93.

Gilbert, H. R., Holzer, T. E., Burkepile, J. T., and Hundhausen, A. J.: 2000, *Astrophys. J.* **537**, 503.

Gopalswamy, N., Shimojo, M., Lu, W., Yashiro, S., Shibasaki, K., and Howard, R. A.: 2003, *Astrophys. J.* **586**, 562.

Jing, J., Yurchyshyn, V. B., Yang, G., Xu, Y., and Wang, H.: 2004, *Astrophys. J.* **614**, 1054.

Kégl, B., Krzyżak, A., Linder, T., and Zeger, K.: 2000, *IEEE Trans. Pattern Anal. Machine Intel.* **22**, 281.

Low, B. C. and Hundhausen, J. R.: 1994, *Astrophys. J.* **443**, 818.

Martin, S. F.: 1998, *Solar Phys.* **182**, 107.

Martin, S. F., Bilimoria, R., and Tracadas, P. V.: 1994, in R. Rutten and C. Schrijvers (eds.), *Solar Surface Magnetism*, Kluwer Academic Publishers, Dordrecht, Holland, p. 303.

Pevtsov, A. A., Balasubramaniam, K. S., and Rogers, J. W.: 2003, *Astrophys. J.* **595**, 500.

Priest, E. R., Hood, A. W., and Anzer, U.: 1989, *Astrophys. J.* **344**, 1010.

Rust, D. M. and Kumar, A.: 1994, *Solar Phys.* **155**, 69.

Rust, D. M. and Martin, S. F.: 1994, in Y. Uchida, T. Kosugi, and H. S. Hudson (eds.), *Magnetodynamic Phenomena in the Solar Atmosphere: Prototypes of Stellar Magnetic Activity*, ASP Conference Series, Vol. **68**, p. 337.

Rust, D. M., Anderson, B. J., Andrews, M. D., Acuña, M. H., Russell, C. T., Schuck, P. W., and Mulligan, T.: 2005, *Astrophys. J.* **621**, 524.

Shih, F. Y. and Kowalski, A. J.: 2003, *Solar Phys.* **218**, 99.

van Ballegooijen, A. A.: 2004, *Astrophys. J.* **612**, 519.

van Ballegooijen, A. A. and Martens, P. C. H.: 1989, *Astrophys. J.* **343**, 971.

Yurchyshyn, V. B., Wang, H., Goode, P. R., and Deng, Y.: 2001, *Astrophys. J.* **563**, 381.

Zharkova, V. V., Aboudarham, J., Zharkov, S. I., Ipson, S. S., Benkhalil, A. K., and Fuller, N.: 2004, in *American Geophysics Union, Fall Meeting 2004*, abstract #SH52A-04.