Solar Phys (2013) 286:385–404
DOI 10.1007/s11207-013-0285-9

Developing an Advanced Automated Method for Solar Filament Recognition and Its Scientific Application to a Solar Cycle of MLSO Hα Data

Q. Hao · C. Fang · P.F. Chen

Received: 1 June 2012 / Accepted: 24 March 2013 / Published online: 16 April 2013

Abstract We developed a method to automatically detect and trace solar filaments in Hα full-disk images. The program is able not only to recognize filaments and determine their properties, such as the position, the area, the spine, and other relevant parameters, but also to trace the daily evolution of the filaments. The program consists of three steps: First, preprocessing is applied to correct the original images; second, the Canny edge-detection method is used to detect filaments; third, filament properties are recognized through morphological operators. To test the algorithm, we successfully applied it to observations from the Mauna Loa Solar Observatory (MLSO). We analyzed Hα images obtained by the MLSO from 1998 to 2009 and obtained a butterfly diagram of filaments. This shows that the latitudinal migration of solar filaments has three trends in Solar Cycle 23: The drift velocity was fast from 1998 to the solar maximum, after which it became relatively slow. After 2006, the migration became divergent, signifying the solar minimum. About 60 % of the filaments with latitudes higher than 50° migrate toward the polar regions with relatively high velocities, and the latitudinal migrating speeds in the northern and the southern hemispheres do not differ significantly in Solar Cycle 23.

Keywords Prominence, formation and evolution, quiescent · Automatic detection · Butterfly diagram

Q. Hao (✉) · C. Fang · P.F. Chen
School of Astronomy and Space Science, Nanjing University, Nanjing 210093, China
e-mail: hq_turtle@163.com

C. Fang
e-mail: fangc@nju.edu.cn

C. Fang · P.F. Chen
Key Laboratory for Modern Astronomy and Astrophysics (Nanjing University), Ministry of Education, Nanjing 210093, China

1. Introduction

Solar filaments, called prominences when they appear above the solar limb, are important magnetized structures containing cool and dense plasma embedded in the hot solar corona. Typically, a filament is 100 times cooler and denser than the surrounding corona. They are particularly visible in Hα observations, where they often appear as elongated dark features with several barbs (Tandberg-Hanssen, 1995; Labrosse *et al.*, 2010). Filaments are always aligned with photospheric magnetic-polarity inversion lines (Martin, 1998) and are located at a wide range of heliocentric latitudes. This characteristic makes filaments suitable for tracing and analyzing the solar magnetic fields (McIntosh, 1972; Mouradian and Soru-Escaut, 1994; Minarovjech, Rybansky, and Rusin, 1998; Rusin, Rybansky, and Minarovjech, 1998). Moreover, filaments sometimes undergo large-scale instabilities, which break their equilibria and lead to eruptions; therefore they are often associated with flares and coronal mass ejections (CMEs) (Gilbert *et al.*, 2000; Gopalswamy *et al.*, 2003; Jing *et al.*, 2004; Chen, 2008, 2011; Zhang, Cheng, and Ding, 2012). This makes case studies and statistical analyses of filaments equally important and significant.

With the rapid development of telescopes, both temporal cadences and the spatial resolutions of observations are increasing. This in turn generates vast amounts of data, which can be efficiently searched with automated detection to derive the features of interest in the observations. Several automated detection methods and algorithms have been developed in the past decade for solar filaments. For example, Gao, Wang, and Zhou (2002) combined the intensity threshold and region-growing methods to develop an algorithm to automatically detect growth and disappearance of filaments. Shih and Kowalski (2003) adopted local and global thresholding and employed morphological-closing operations to identify filaments. Fuller, Aboudarham, and Bentley (2005) used a morphological hit-or-miss transformation and calculated the Euclidean distance to obtain the filament spines. Bernasconi, Rust, and Hakim (2005) developed an algorithm based on a geometric method, which was recently updated by Martens *et al.* (2012), to determine the filament chirality in addition to the locations, where they confirmed the hemispheric rule of filament chirality. Based on the Sobel operator, Qu *et al.* (2005) applied an adaptive-threshing method for detecting and deriving various filament parameters. Wang *et al.* (2010) employed morphological methods, while Labrosse, Dalla, and Marshall (2010) applied the support vector machine (SVM) method to detect a EUVI 304 Å prominence above the solar limb. Yuan *et al.* (2011) designed a cascading Hough circle to determine the center location and the radius of the solar disk, and also to find the filament spines based on graph theory.

In this article, we present an efficient and versatile automated detecting and tracing method for solar filaments. It is able not only to recognize filaments, determine features such as position, area, spine, and other relevant parameters, but also to trace the daily evolution of the filaments. In Section 2 we describe image preprocessing before detecting filaments. The filament-detection algorithm based on the Canny edge-detection method and the connected-components process are given in Section 3. We describe the feature-recognition algorithm in detail in Section 4. The tracing algorithm is explained in Section 5. The performance of our program is described in Section 6. Finally, statistical results of the filament latitudinal distribution based on the Hα archive of the Mauna Loa Solar Observatory (MLSO) are presented in Section 7 before we draw our conclusions in Section 8.

2. Preprocessing

The raw image preprocessing consists of five steps, which are explained one by one in the following subsections.

2.1. Hα Data Acquisition and Analysis

The full-disk Hα images that we processed were mainly downloaded from the MLSO website (http://mlso.hao.ucar.edu). Each image has a size of 1024×1024 pixels and was acquired with the *Polarimeter for Inner Coronal Studies* (PICS). The pixel size of the image is $2.9''$. The MLSO Hα data archive provides two types of images: one has a limb-darkening correction applied along with contrast enhancement, the other is the raw data. Our program can process the FITS format as well as web image formats such as GIF and JPEG.

2.2. Limb-Darkening Removal

The limb-darkening effect, *i.e.* the effect produced when the intensity decreases toward the solar limb, may cause false detections. This is to be removed first. Some observatories such as those from MLSO also provide limb-darkening-corrected images. The polynomial fitting method of Pierce (Cox, 2000) was adopted to remove the limb-darkening effect,

$$I_{\mathrm{H}\alpha}^{\mathrm{cor}}(\theta) = I_{\mathrm{H}\alpha}^{\mathrm{raw}}(\theta)/\left(1 - u_2 - v_2 + u_2 \cos\theta + v_2 \cos^2\theta\right), \tag{1}$$

where $I_{\mathrm{H}\alpha}^{\mathrm{cor}}(\theta)$ is the corrected intensity and $I_{\mathrm{H}\alpha}^{\mathrm{raw}}(\theta)$ is the intensity in the raw images, θ is the angle between the local radial direction and the line of sight, and $u_2 = 0.88$ and $v_2 = -0.23$ are the fitted constants for the Hα wavelength at 6563 Å.

2.3. Solar Disk Extraction

Since the solar disk is only a part of the entire image, surrounded by a large part of the sky background, we need to remove the background to process the solar disk only, which can reduce processing time and storage space. The method for disk extraction is simple: we just find the left, the right, the top, and the bottom ranges of the solar disk. Then we obtain the subimage according to these ranges. An example is shown in Figure 1(b).

2.4. Top-Hat Filter for Enhancement

Morphological image processing is a type of processing in which the spatial forms or the structures of the objects within an image are modified (Haralick and Shapiro, 1992; Pratt, 2001). Erosion, dilation, opening (erosion followed by dilation), and closing (dilation followed by erosion) are the basic operators in the morphological concepts that have been extended to work with gray-scale images for image segmentation and enhancement. Sometimes we obtain the images where the filament boundaries are fuzzy. To more accurately segment the filament structure, it is necessary to enhance the image to increase the intensity contrast between the filament and non-filament structures. We used the morphological top-hat transformation to enhance the images. The algorithm is composed of three steps:

i) Computing the morphological opening of the image with the top-hat filtering and subsequently subtracting the result from the original image.
ii) Computing the morphological closing of the image with the bottom-hat filtering and subsequently subtracting the result from the original image.

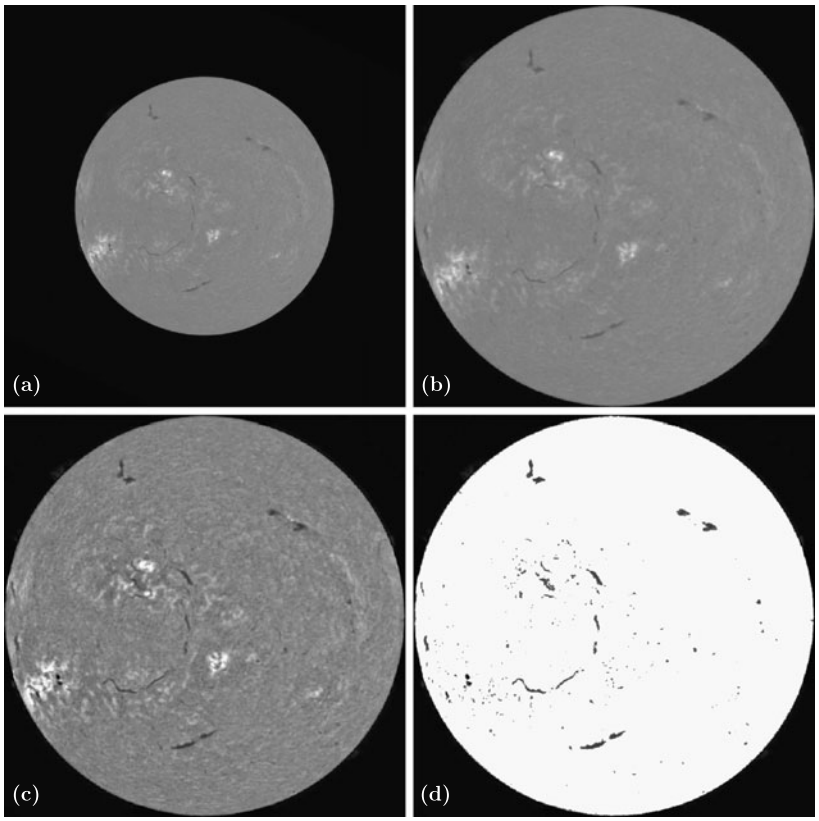


Figure 1 An example of the Hα image at 18:08:52 UT, 20 April 2001, obtained by the Mauna Loa Solar Observatory (MLSO). (a) The original corrected image downloaded from the MLSO web site. (b) The sub-image extracted from panel (a). (c) The enhanced image after the top-hat transformation. (d) The image after threshold filtering.

iii) Adding the top-hat filtered image to the original image and subsequently subtracting the result from the bottom-hat filtered image.

This produces an enhanced image, as shown in Figure 1(c).

2.5. Threshold Filter

The solar disk in Hα shows some other bright features such as plages and flares in addition to dark features such as filaments and sunspots. Many authors used a local-threshold method to filter these features out when they were not the focus of their studies. After limb-darkening removal and top-hat filtering, we can easily distinguish the filaments from these non-filament structures in the gray-scale images with the global-threshold method. We tested several hundred images to find the appropriate global-threshold value. For MLSO images we found that the threshold value is about 95 – 100. The algorithm is simple: if the pixel value exceeds the threshold value, we assigned it to be 255, which means white in the image. An example is shown in Figure 1(d).

3. Filament Detection

3.1. Canny Edge Detection

Segmenting an image entails dividing or separating the image into regions of similar attributes (Pratt, 2001). The most basic attribute for segmentation is the intensity level for a gray-scale image and color components for a color image. The image edge is another useful attribute for segmenting. It is possible to segment an image into regions of common attributes by detecting the boundary of each region across which there is a significant change in intensity. We adopted the most powerful edge-detection method, *i.e.* the Canny (1986) method, to identify filaments. The Canny method differs from other edge-detection methods in that it uses two different thresholds: one for detecting strong edges and the other for weak edges. The weak edges are included in the output only if they are connected to strong edges. Compared with others, this method is therefore less distracted by noise and is more likely to detect true weak edges (Lim, 1990). The Canny method works in a multi-step process:

Step 1: Noise reduction. Because the Canny edge detector is susceptible to noise present in the image data, the image needs to be smoothed first. In our method, after preprocessing, each image is smoothed by Gaussian convolution as follows:

$$G(x,y)=\frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{2}$$

$$f'(x,y)=\sum_{x'}\sum_{y'}f(x,y)G\big(x-x',y-y'\big), \tag{3}$$

where $G(x,y)$ is the 2D Gaussian filter, $f(x,y)$ is the input image, and $f'(x,y)$ is the output image that is convolved with the 2D Gaussian filter, (x,y) is the position in the x–y plane of the image. We chose $\sigma=1$ in our processing.

Step 2: Finding gradients. The edges can usually be found at places where the gray-scale intensity changes drastically. This means that we can find them by checking the gradient at each pixel in the image. The first step is to obtain the gradient in the x-direction $[g_x(x,y)]$ and y-direction $[g_y(x,y)]$, by applying the derivative of a Gaussian filter:

$$g_x(x,y)=\sum_{x'}\sum_{y'}f(x,y)g_1\big(x-x',y-y'\big), \tag{4}$$

$$g_y(x,y)=\sum_{x'}\sum_{y'}f(x,y)g_2\big(x-x',y-y'\big), \tag{5}$$

where

$$g_1(x,y)=-\frac{1}{\pi\sigma^2}x\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{6}$$

$$g_2(x,y)=-\frac{1}{\pi\sigma^2}y\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{7}$$

Then we use the following two equations to determine the gradient magnitude and the direction of the edge:

$$g(x,y)=\sqrt{g_x(x,y)^2+g_y(x,y)^2}, \tag{8}$$

$$\theta(x,y)=\arctan\frac{g_y(x,y)}{g_x(x,y)}. \tag{9}$$

Step 3: Non-maximum suppression. For an image array, the edge direction angle is rounded to one of four angles representing the vertical, the horizontal, and the two diagonals (*i.e.* 0, 45, 90, 135, 180, 225, 270, 315, and 360 degrees), corresponding to using an eight-connected neighborhood. Then, for each pixel of the gradient image, we compare the edge gradient magnitude of the current pixel with the edge gradient magnitude of the pixel along the gradient direction. For example, if the gradient direction is toward the Northeast, the pixel should be compared with pixels toward the Northeast and the Southwest. If the edge gradient magnitude of the current pixel is the steepest one, we mark it as one part of the edge. If not, we suppress it, *i.e.* it is ignored.

Step 4: Edge tracing by hysteresis. After step 3, many of the remaining edge pixels are probably the true filament edges, but some may be caused by noise. The Canny method uses thresholding with hysteresis to determine whether the edges obtained in step 3 are true or not. The algorithm adopts two, high and low thresholds: If the edge pixel's gradient magnitude is higher than that of the high threshold, the pixel is marked as a strong pixel; if the edge pixel's gradient magnitude is lower than the low threshold, the pixel will be suppressed; if it is between the two thresholds, the pixel will be marked as weak.

To find the thresholds, we used an automatic method: First, we provided a probability of the pixels that are not the edge points and calculated the number of pixels that may not be the edge points in the entire image by the probability. Then we increased the gradient threshold until the total number of the pixels whose gradient was lower than the threshold was just higher than the probability value, then the current gradient threshold was chosen as the high threshold. The low threshold is about half that of the high threshold. In our process the probability is chosen to be about 0.98, and the low threshold is 0.4 times the high threshold.

After tracing through the entire image we obtained strong and weak pixel arrays, which could be treated as a set of edge curves. The weak edges were included if, and only if, they were connected to strong edges. We scanned the entire binary image to find the pixels where strong and weak edges overlapped each other and finally obtained the edge map. Then, the morphological-thinning operation (Lam, Lee, and Suen, 1992) was applied to minimize the connected lines to obtain accurate and fine edges.

After applying the Canny edge-detection method, we obtained the edge of each filament, as shown in Figure 2(a).

3.2. Connected Component Process

After segmenting, the image is still an array of pixels in the computer vision. We are not interested in each pixel but in the special region (*i.e.* the filament) constituted by the pixels. These regions are called the connected components of the binary images, which are more complex and have a rich set of properties (*i.e.* shape, position, and area). We used a classical method (Haralick and Shapiro, 1992) to realize the connected component labeling. This means that the pixels in a connected component are given the same identity label. After connected-component labeling, we obtain a product that is changed from pixels to the regions that we are interested in. The input binary images are of eight-connectivity, and the algorithm consists of the following two steps:

In the first step, the algorithm proceeds through each pixel from left to right and from top to bottom, as indicated by the arrows shown in Figure 3(a). It checks the labels of four neighboring pixels that are Northeast, North, Northwest, and West of the current pixel. For example, assuming the current pixel is (i, j) as shown in Figure 3(a), the code checks the labels of the four pixels that are $(i-1, j+1)$, $(i-1, j)$, $(i-1, j-1)$, and $(i, j-1)$:

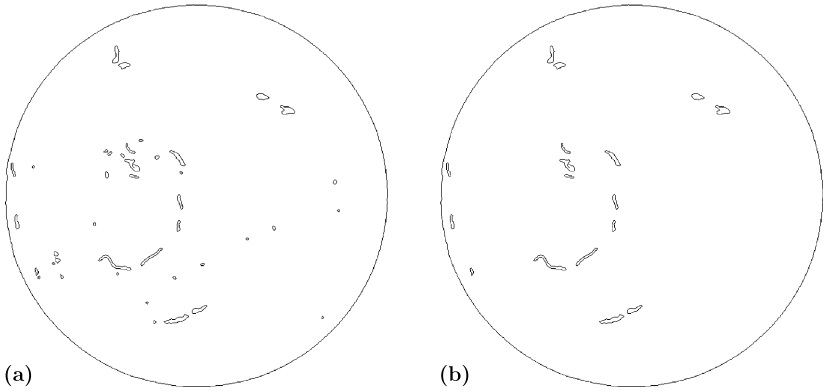
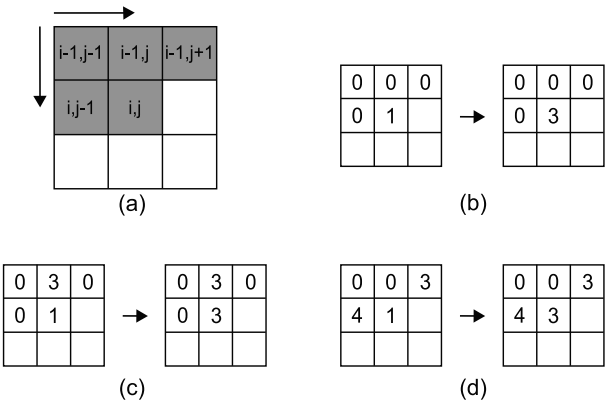


Figure 2 Examples after filament detection processing. (a) After Canny edge detection, the filaments were segmented; the result shows the edge of each filament or fragment. (b) The image after sunspot removal. Compared with (a), small cycles (*i.e.* the edges of sunspots) were removed. To obtain clear results, the foreground and background colors are exchanged, *i.e.* the filament features become black and the background becomes white. The same procedure was applied in Figure 4.

Figure 3 Scanning sequence (panel a) and three examples for the connected-components label method (panels b, c, and d).



i) If none of the four neighbors are assigned, a new label is assigned to the current pixel. An example is shown in Figure 3(b): Assuming that the value of the current pixel is 1 and the values of the four neighbors are 0 (0 means this pixel is a background pixel, and 1 means the pixel is the foreground), this means a new filament is encountered. If the previous label is 2, we assign label 3 to the new filament.

ii) If one of the four neighbors has been labeled before, we assign the neighbor's label to the current pixel. An example is shown in Figure 3(c): One of the four neighbors has been labeled, *i.e.* the north neighbor has been labeled 3, so we assign the same label to the current pixel.

iii) If more than one of the neighbors have been labeled before, we assign the lower label to it. An example is shown in Figure 3(d): Two neighbors have been labeled, *i.e.,* the northeast and west neighbors. The label of the northeast neighbor is 3, which is lower than the west neighbor's, so we use 3 to assign the current pixel.

After completing the scanning, the equivalent label pairs are sorted into equivalence classes. A unique label is assigned to each class.

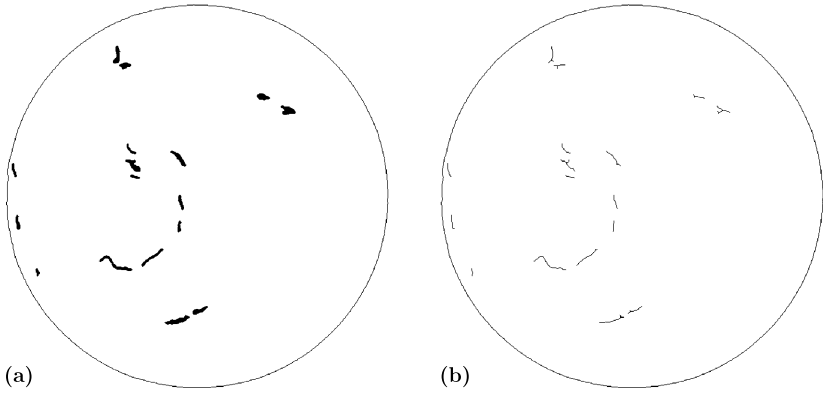


Figure 4 Examples of filament feature recognition. (a) In the image after morphological reconstruction processing, the edge curve is full with white pixels, from which we obtain the filament area. (b) The filament spines after morphological skeletonization and barb-removal processing.

In the second step, the algorithm again processes the data and replaces each label by the label assigned to its equivalence class. After completing the scanning, a unique label is assigned to each equivalence class. In other words, we have assigned each filament a unique label.

3.3. Sunspot Filter

It is challenging to distinguish between sunspots and filaments by gray-scale levels. The labeled "filaments" so far may include some sunspots. Therefore, we have to separate real filaments from sunspots by use of geometric structures: the size and the long-to-short-axis ratio of the filament. A candidate whose size (*e.g.* the perimeter) is larger than the threshold is considered to be a filament. If the size of the labeled object is smaller than that of the threshold, the candidate is treated as a small filament only if the ratio of the long to short axis is higher than a given value; otherwise it is removed. We obtained the edge of each filament whose length can be treated as the filament perimeter. We set the perimeter threshold to be 25 pixels, and the long-to-short-axis ratio threshold is 2 in our procedure. An example is shown in Figure 2(b). The filament label needs to be updated after removing the sunspots. Then each filament is labeled with a unique number. An example is shown in Figure 7(a). This method is used to filter out sunspots. With the same method, we can filter out other features. In other words, we can adopt the method to automatically detect sunspots, which will be implemented in our future work.

4. Filament Feature Recognition

4.1. Perimeter

As mentioned above, we have the filament edge, which can be easily used to derive the filament perimeter after the connected-component process. This is achieved by integrating the distance that connects neighboring pixels along the edge of each filament.

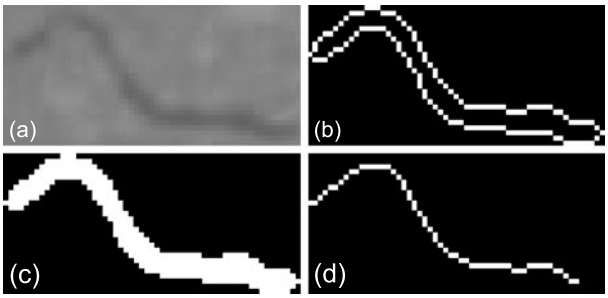


Figure 5 An example of a filament feature recognition. (a) A filament extracted from the original image (Figure 1(a)). (b) The filament edge (the perimeter). (c) The filament edge was filled with white pixels, from which we obtained the filament area. (d) The filament skeleton (spines).

4.2. Position

We chose the geometric center of each filament, *i.e.* the centroid, as the location of the filament. First, we found the centroid of a filament (x_c, y_c), *i.e.* we calculated the average of the abscissa and the ordinate of all filament pixels. Since filaments follow the solar rotation and the rotation axis wobbles with time, the position of a certain filament has an elliptic orbit in the plane of the sky. To obtain the longitude and latitude of the filament in the heliographic coordinates, we used the SolarSoftWare routine xy2lonlat.pro.

The center of the solar disk that we used here is not that provided in the header of the "FITS" file. After the Canny edge detection, we obtained the boundary of the solar disk in addition to the filament locations, which is the biggest connected component after the connected component processing. We fit the circle and calculated the geometric center of the fitted circle, which is the exact center of the solar disk.

4.3. Area

In our process, the area of a filament is the integration of the pixel area divided by the cosine of the heliocentric angle, where the integration is taken in the area enclosed by the edge curve. We used the foreground color (white) to fill the edge curve so that the pixels inside the curve become white, then the white pixels constitute the filament area. The algorithm is based on morphological reconstruction (Soille, 1999). The edge curve with holes is filled with white pixels. The hole is a set of background pixels (*i.e.* black pixels) surrounded by foreground pixels (white pixels). An example is given in Figure 4(a) and an extracted filament example is shown in Figure 5(c).

4.4. Spine

The morphological-skeletonization method was adopted to derive the filament spine, which is a stick-like skeleton, spatially placed along the middle region of the filament. The skeleton is a unit-wide set containing only pixels that can be removed without changing its topology (Kong and Rosenfeld, 1996). The skeleton method employs iterations of the morphological-thinning operation (Haralick and Shapiro, 1992). Assuming that the input binary image is I (*i.e.* a matrix only has values of 0 and 1), the details of the method are as follows:

First, the thinning processing of I by structuring-element pair (J, K) is defined as

$$I \oslash (J, K) = I - I \otimes (J, K), \tag{10}$$

where $\oslash$ is the morphological thinning operator, J and K must satisfy $J \cap K = \emptyset$, where the symbol $\emptyset$ represents the empty set. $I \otimes (J, K)$, *i.e.* the hit-and-miss transformation of set I by (J, K) is defined by

$$I \otimes (J, K) = (I \ominus J) \cap \left(I^c \ominus K\right), \tag{11}$$

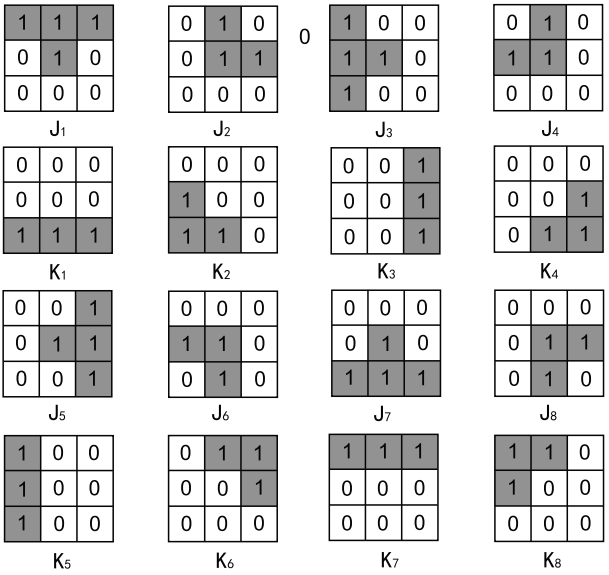


Figure 6 Structuring-element pair (J_i, K_i) used to determine the filament skeleton. J_i and K_i must satisfy $J_i \cap K_i = \emptyset$.

where $\otimes$ is the hit-and-miss transformation operator, $\cap$ the intersection operator, I^c the complement of I, and the symbol $\ominus$ denotes the morphological-erosion operator.

Then, one uses the sequence of eight structuring-element pairs as shown in Figure 6 to iteratively process the thinning operation:

$$I_{N+1} = \left(\cdots\left\{\left[I_N \oslash (J_1, K_1)\right] \oslash (J_2, K_2)\right\} \oslash \cdots \oslash (J_8, K_8)\right). \tag{12}$$

We let $I_0 = I$, which is firstly thinned by the structure-element pair (J_1, K_1), and then by $(J_2, K_2), \ldots, (J_8, K_8)$; the thinned result is defined as I_1. This process is repeated to derive $I_2, I_3, \ldots, I_N$.

Finally, the thinning process is repeated until $I_N = I_{N+1}$, *i.e.* the filament skeleton is the final structure that cannot be thinned any more. Some of the resulting filament skeletons still contain small barbs. We used the morphological hit-and-miss transformation to find the endpoints that constitute the barbs, and then removed the barbs. This process may be iterated several times because some of these points may not be removed in one iteration. In our process it is iterated four times. The filament spines on the whole Sun are shown in Figure 4(b) and an extracted filament example is shown in Figure 5(c).

4.5. Tilt Angle

The tilt angle is defined as the fitted filament spine orientation with respect to the solar Equator. After we derived the spine, we used a linear polynomial to fit it, with which we calculated slope A. The tilt angle is $\arctan(A)$.

4.6. Feature Update

A filament may consist of several fragments. Thus, the filaments that we obtained after the previous processing may not be the real individual filaments, with some being fragments of one filament. We adopted a distance criterion to find the fragments belonging to a single filament. The method is the same as the labeling criterion for the filament tracking method

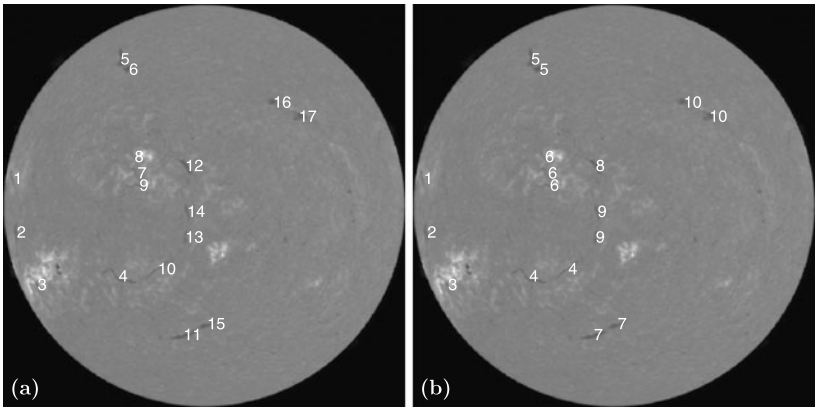


Figure 7 An example of the detection results. (a) After filament detection, each filament or fragment is labeled with a unique number. There are 17 filaments or fragments in total; each of their centers is labeled with a unique number. (b) Each candidate filament that we obtained after the filament detection processing may not be a true one but a filament fragment. After the feature-update processing, the fragments belonging to a single filament are labeled with the same number. For example, filament fragments numbers 11 and 15 in (a) were recognized as one filament and are accordingly updated with the same label number 7 in panel (b).

used by Joshi, Srivastava, and Mathew (2010), which is explained as follows: We compared a certain filament or filament fragment with all other fragments. The fragments were recognized to belong to a common filament if the two fragments lie within the distance threshold. The process was iterated until all fragments were checked. The filaments in the new image were compared with those in the previous image. The experiential distance threshold in our processing was taken to be 60 pixels for the MLSO data. After this process, the fragment labels were updated; if several fragments belonged to a common filament, the label was unified, as shown in Figure 7(b). For example, filament fragments 11 and 15 in Figure 7(a) were recognized as one filament, thus they were updated with the same label number 7. For the area update, the perimeter, and the spine length, we calculated the sum of each fragment with the same label, while the position and the tilt angle needed to be reprocessed.

Another way to identify a broken filament is to compare the tilt angle of the fragments, *i.e.* if the neighboring candidates have similar tilt angles, they can be considered as one big filament. This method works fine for the magnetic inversion lines that are not strongly curved, and will be incorporated in our future version.

5. Filament Tracing

Tracing the evolution of the filament is important for understanding the physical nature and the solar-cycle variation of the filaments. In this section we present a tracing method. Here we used the filament label, position, and area that have been obtained in Section 4 as input parameters to trace the daily filament evolution.

We defined the two input images as I_{old} (the image observed at the old time) and I_{new} (the image observed at the new time). The main steps of the tracing method are these:

i) We obtain the observation time of the two images I_{old} and I_{new}, then calculate the time interval T_{interval}.

ii) Using the latitude from the position features of each filament in $I_{\rm old}$ (or $I_{\rm new}$) to calculate the rotation velocity at this latitude $\omega_{\rm old}$ (or $\omega_{\rm new}$), then calculate the possible longitude $PLO_{\rm old}$ (or $PLO_{\rm new}$) with the time interval $T_{\rm interval}$. Here the $PLO_{\rm new}$ is calculated by assuming that the Sun rotates backward,

$$PLO_{{\rm old}(i)} = CLO_{{\rm old}(i)} + \omega_{{\rm old}(i)} T_{\rm interval} \quad (i = 1, 2, \ldots, n), \tag{13}$$

$$PLO_{{\rm new}(j)} = CLO_{{\rm new}(j)} - \omega_{{\rm new}(j)} T_{\rm interval} \quad (j = 1, 2, \ldots, n), \tag{14}$$

where $CLO_{{\rm old}(i)}$ (or $CLO_{{\rm new}(j)}$) is the current longitude of i-th (or j-th) filament in $I_{\rm old}$ (or $I_{\rm new}$). Here, we adopt the solar-rotation angular velocity formula (Balthasar, Vazquez, and Wöehl, 1986) to determine $\omega_{{\rm old}(i)}$ and $\omega_{{\rm new}(j)}$:

$$\omega(\theta) = (14.551 \pm 0.006) - (2.87 \pm 0.06) \sin^2 \theta, \tag{15}$$

where ω is the angular velocity (degrees per day) and θ the latitude.

iii) We obtain the possible position $[PP_{\rm old} : (PLA_{\rm old}, PLO_{\rm old})]$ of the filament after $T_{\rm interval}$ via the differential rotation formula, then calculate the distance $D_{\rm old\to new}$ between $PP_{\rm old}$ and real current position $[CP_{\rm new} : (CLA_{\rm new}, CLO_{\rm new})]$ of the filament in $I_{\rm new}$. Because the drift velocity of the filament is much lower than the solar-rotation velocity, we assume the filament latitude does not change in $I_{\rm old}$ (*i.e.* $PLA_{\rm old} = CLA_{\rm old}$, $PLA_{\rm old}$ is the possible latitude and $CLA_{\rm old}$ is the current latitude in $I_{\rm old}$). The distance $D_{\rm old\to new}$ between $PP_{\rm old}$ and $CP_{\rm new}$ is

$$D_{\rm old\to new} = \sqrt{(PLA_{\rm old} - CLA_{\rm new})^2 + (PLO_{\rm old} - CLO_{\rm new})^2}. \tag{16}$$

iv) We assume that the Sun rotates backward, then obtain the possible position of the filament $[PP_{\rm new} : (PLA_{\rm new}, PLO_{\rm new})]$ after $T_{\rm interval}$. A processing similar to iii) is used and the distance $D_{\rm new\to old}$ between $PP_{\rm new}$ and $CP_{\rm old}$ is

$$D_{\rm new\to old} = \sqrt{(PLA_{\rm new} - CLA_{\rm old})^2 + (PLO_{\rm new} - CLO_{\rm old})^2}. \tag{17}$$

v) For each filament in $I_{\rm new}$, we check all filaments in $I_{\rm old}$. Only if it meets the following three conditions, the filament in $I_{\rm old}$ is considered to be the same filament and marked with the same label as in $I_{\rm new}$:

Condition 1: $D_{\rm old\to new} \le ED_1 T_{\rm interval}$;
Condition 2: $D_{\rm new\to old} \le ED_2 T_{\rm interval}$;
Condition 3: $Area_{\rm old(current)}/Area_{{\rm new}(j)} \ge ratio$ $(j = 1, 2, \ldots, n)$,

where $ED_1 T_{\rm interval}$ and $ED_2 T_{\rm interval}$ are the distance threshold. In our program we take $ED_1 = ED_2 = 100$ pixels day^{-1}. $Area_{\rm old(current)}$ and $Area_{{\rm new}(j)}$ are the filament area at the old and new times, respectively. Condition 3 gives the maximum proportion of the deformation, *i.e.* less than the specified *ratio* of the size of the previous filament. Here we set $ratio = 50\,\%$. If the three conditions are not satisfied, the filament at the new time is identified as a new filament and given a new label. This step continues until all filaments in $I_{\rm new}$ are treated. Finally, the filament labels in $I_{\rm new}$ are updated. Figure 8 gives an example of our tracing method, where panel (a) shows the detected result of the earlier image with each filament labeled with a unique number, panel (b) depicts the filament fragments after detection, panel (c) shows the updated filaments where several fragments are merged into one filament, and panel (d) shows the tracing result based on the earlier image in panel (a). For example, filament 2 in the earlier image (panel a) is split into three fragments (labeled 2, 4, and 5 in the later image (panel b)). After using the tracing method, they were labeled with the same number as in the earlier image (panel a). Filament 13 in the later image (panel d) was not detected in the earlier image (panel a), therefore it was given a new number.

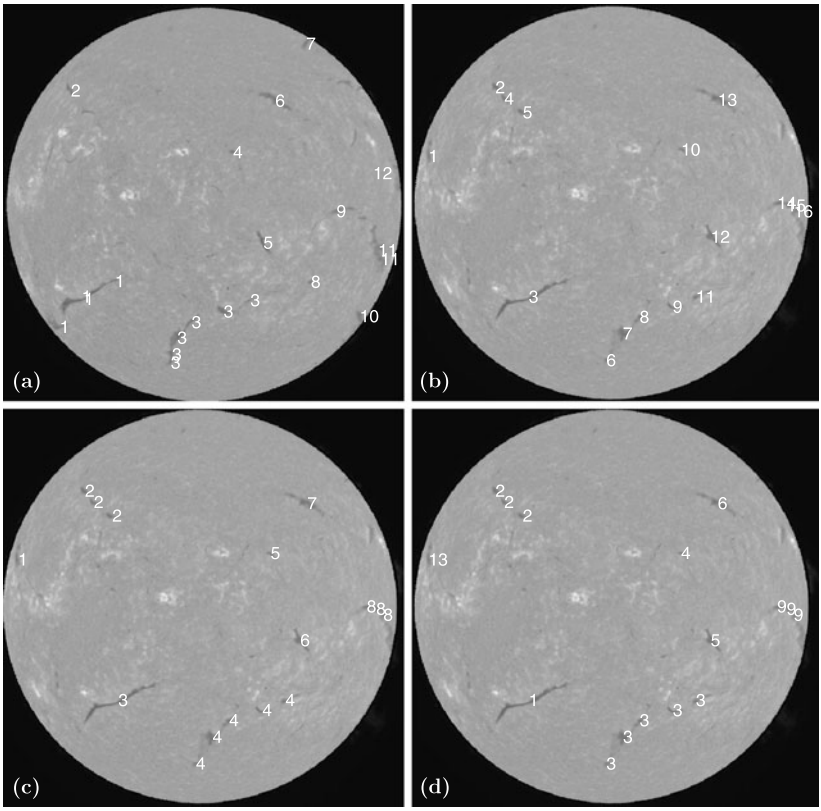


Figure 8 An example of the filament tracing method between two images at 18:41:38 UT, 17 February 2001 and at 18:41:40 UT, 18 February 2001 obtained by MLSO. (a) The detected result of the earlier image, each filament is labeled with a unique number. (b) The detected filament fragments are shown individually. (c) The updated filaments where fragments are labeled to be one filament. (d) Final result, where the labels of the traced filaments are updated.

6. Performance

Our code was developed with the MATLAB® Desktop Tools and Development Environment on a desktop computer (CPU: Intel® Core™ Duo 3.00 GHz). After processing each image file, the result (such as the label, the position, the area, and other features) are written into a text file. The average processing speed is 1 second for the filament detection in a single image and 3.5 seconds for the filament detection and tracing in two images. We randomly selected 100 images from the MLSO Hα archive for testing and compared the automated result with the manual ones. For filaments and filament fragments, the two methods are overlapping by 85 ± 2 % and 88 ± 4 %, respectively. The error includes two types of false recognition: One is that there is a manually recognized filament, but the automated method cannot detect it. The other is that the automated method detects a filament, but it does not appear on the real solar disk. This latter case is rarely seen in our method. If a filament splits into several small fragments, and the criterion of the long-to-short axes ratio is not satisfied (*i.e.* being recognized as a sunspot), our method may miss these fragments. The accuracy of the filament-fragment number is a little higher than that of the filament, which is

due to the prescribed distance criterion. Sometimes several filaments or filament fragments in one active region are so close to each other and within the distance criterion that they are recognized as one filament. This kind of false recognition does appear in our filament-fragment detection, and we have to improve the filament-fragment-merging method in the future. For other features such as position, perimeter, area, and spine, there are no standard criteria to test the accuracy of the results processed by our codes. However, we defined two indices to test the performance of our method, *i.e.* the "edge-closed rate" and the "area fully filled rate". The edge-closed rate is defined as the number of detected filaments whose edge curve is closed as a percentage of the total number of detected filaments among the 100 test images. We found that the rate is 91 %. This rate mainly depends on the selection of the threshold in the threshold filter processing and the two thresholds in the Canny edge-detection method. If the filament edge curve is not closed, it leads to a low detection accuracy and affects the subsequent processing. The area fully filled rate is defined as the number of the detected filaments whose edge curve is fully filled with foreground pixels as a percentage of the total number of detected filaments among the 100 test images; this rate is 75 %. After the edge detection is finished, the morphological object-filling method cannot fully fill in the area enclosed by the edge curve if the edge curve is not closed. This leads to the decrease in the area fully filled rate and the detection-accuracy rate. The filament spine is also affected by the area problem, *i.e.* if the area is not fully filled, our method may produce a wrong filament topology after the morphological skeletonization processing. Furthermore, if the barbs are located near the end of the filament spine or if filament size is relatively small, the recognized spine may be shorter than the real length after the morphological barb-removal processing. The shorter the time interval, the higher the tracing accuracy. Here, we set the default time interval to be about one day, the accuracy of the tracing method is about 80 %. In addition, we also tested images from the Big Bear Solar Observatory (BBSO) Hα archive (ftp://ftp.bbso.njit.edu/pub/archive/) to validate the versatility of our method. The results are similar and satisfactory.

7. Statistical Results of the Filament Latitude

We analyzed with our automated method 3470 images obtained by the MLSO from January 1998 to December 2009. In this section, we present the statistical results of the evolution of the filament latitudinal distribution because of its relatively high accuracy. Furthermore, from a statistical point of view, the results can significantly help in understanding the cyclic migration of solar filaments.

7.1. Butterfly Diagrams

For the period from January 1998 to December 2009, we processed one image per day and detected 13 832 filaments. The temporal evolution of the latitudinal distribution of these filaments is depicted as the scatter plot in Figure 9 (each dot represents a single observation), where we can clearly see a butterfly diagram, similar to sunspots. Because of the lack of observations in some periods, there are several white vertical gaps in the butterfly diagram. From the diagram we can see the distribution and the migration of the filaments. This butterfly diagram indicates that the formation of the filaments mainly migrates toward the Equator from the beginning to the end of Solar Cycle 23.

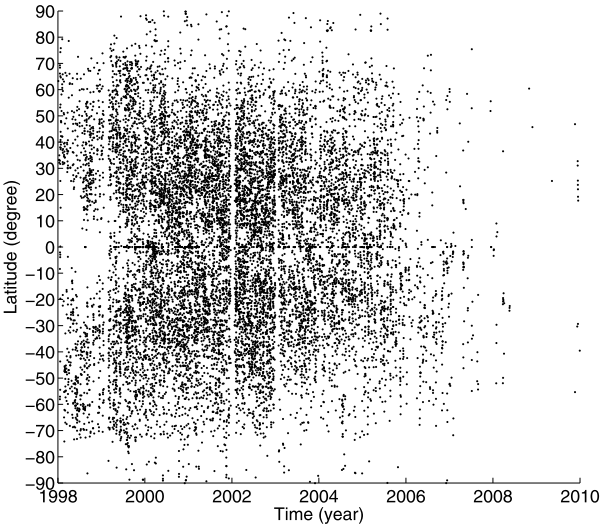


Figure 9 Butterfly diagram of filaments from January 1998 to December 2009 in Solar Cycle 23. Each dot represents a single observation.

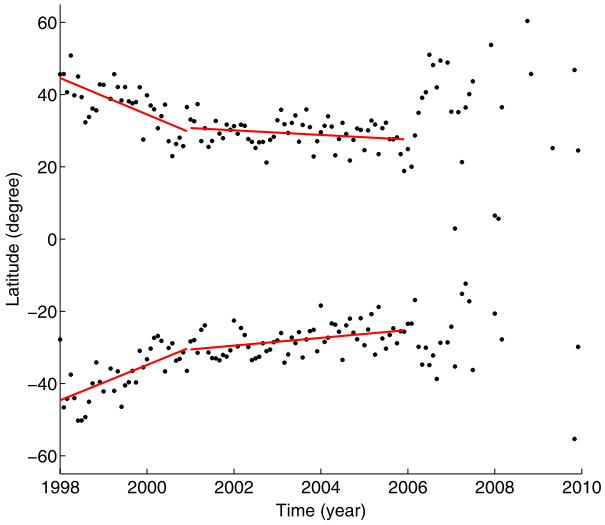


Figure 10 Temporal evolution of the monthly mean latitudinal distribution of filaments from January 1998 to December 2009. Linear fittings are shown by solid-red lines.

7.2. Drifting Velocity

From the butterfly diagram we only obtain qualitative results, as mentioned by Li (2010). To make a quantitative analysis, we adopted the monthly mean latitude of the filaments in the northern and southern hemispheres, respectively. The calculated results are plotted in Figure 10. It can be seen that the monthly mean latitudinal distribution of the filaments has three drift trends: from 1998 to the solar maximum (2001) the drift velocity is very fast. After the solar maximum it becomes relatively slow. After 2006, the drift velocity becomes divergent. A linear fitting was used for the data points in different periods, resulting in an average drift velocity of 0.0138 degree day^{-1}, or 1.86 m s^{-1}, during 1998 – 2001, and 0.0017 degree day^{-1} or 0.23 m s^{-1} during 2002 – 2006 in the northern hemisphere. It is 0.0134 degree day^{-1}, or 1.80 m s^{-1} during 1998 – 2001 and 0.0029 degree day^{-1}, or 0.39 m s^{-1} dur-

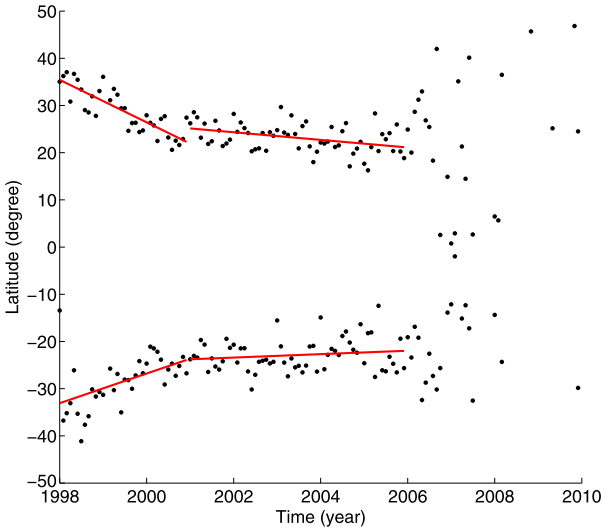


Figure 11 Temporal evolution of the monthly mean latitudinal distribution of filaments with a latitude lower than 50° from January 1998 to December 2009. Linear fittings are shown by solid-red lines.

ing 2002 – 2006 in the southern hemisphere. Here, we did not fit the monthly mean filament distribution after 2006, because it becomes divergent near the solar minimum.

Since the normal solar activity is usually applied to events with latitudes lower than 50° (Sakurai, 1998; Li *et al.*, 2008), we analyzed the filaments with latitudes lower than 50°. The calculated result is plotted in Figure 11. It can be seen that the monthly mean latitudinal distribution of these filaments again has three drift trends: From 1998 to the solar maximum (2001) the drift velocity is fast, *i.e.* 0.0123 degree day^{-1} or 1.66 m s^{-1} in the northern hemisphere and 0.086 degree day^{-1} or 1.16 m s^{-1} in the southern hemisphere. After the solar maximum, the drift velocity becomes relatively slow, *i.e.* 0.0022 degree day^{-1} or 0.29 m s^{-1} in the northern hemisphere and 0.0010 degree day^{-1} or 0.13 m s^{-1} in the southern hemisphere, respectively. After 2006 it becomes divergent. These results are similar to those of the entire latitudinal distribution. The reason is easy to understand: among the 13 832 filaments that we detected, only 1130 filaments have latitudes higher than 50°. In other words, the detected filaments are mainly distributed in latitudes lower than 50°. There is no obvious difference between the northern and southern hemispheres. These results are similar to the statistical results of Li (2010). Similarly, we plot the monthly mean latitude of the filaments with latitudes higher than 50° in Figure 12. However, no clear trend is discernible.

To find the migration of the individual filaments above 50°, we employed our tracing method and set three additional conditions for tracing:

Condition 1: The filament positions are higher than 50° at the first detection.

Condition 2: The time interval is shorter than two days. In other words, in three consecutive days, observations are available during at least two days. The purpose of this condition is to improve the accuracy.

Condition 3: The total time lapse should be less than ten days, because one specific filament observation can be clearly visible in less than a half-period of the solar rotation. If the time lapse is longer, the accuracy of the tracing method is lower.

We plot the tracing results of the latitude *versus* normalized time in Figure 13. Here the normalized time means that we assumed the dates of the first filament detection as the start time. The drift-velocity distribution histograms in the northern and southern hemisphere are

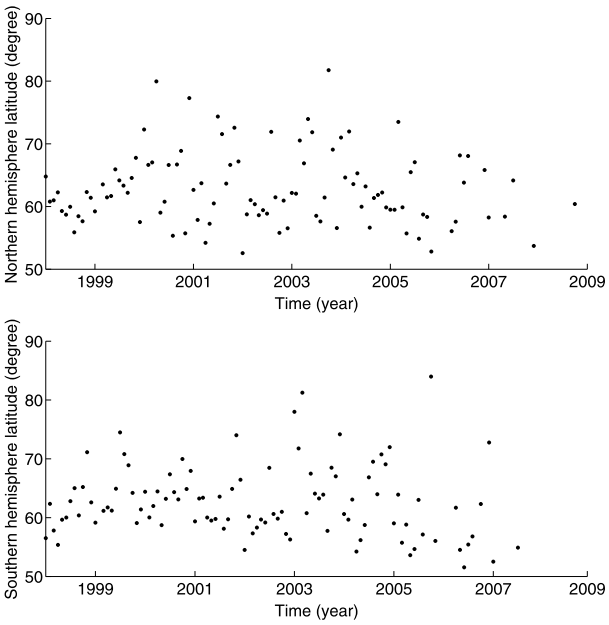


Figure 12 Temporal evolution of the monthly mean latitudinal distribution of the filaments with latitude higher than 50° from January 1998 to December 2009. The upper panel plots the northern hemisphere and the bottom panel the southern hemisphere.

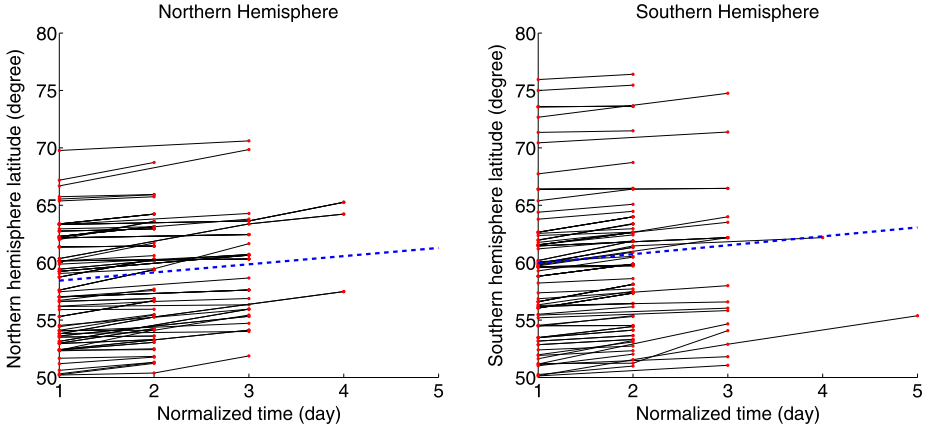


Figure 13 Traced filament latitude variation *versus* normalized time. Here the normalized time means that we assumed the dates of the first filament detection as the start time. The left and right panels depict the northern and southern hemisphere, respectively. The solid-black line represents the temporal and spatial variation of the traced filament latitudes, and the red dot indicates the filament where and when it was detected and traced. The dashed-blue line is the linear fitted average of all traced temporal and spatial filament variations.

shown in Figure 14. In the northern hemisphere, there are 103 filaments (which is 57 % of all filaments that satisfy the conditions and are traced with a latitude higher than 50°), which migrate toward the polar region. The average drift velocity is 0.7126 degree day^{-1} (96.2 m s^{-1}). In the southern hemisphere, there are 97 filaments (which is 61 % of all filaments that satisfy the conditions and are traced with a latitude higher than 50°), which migrate toward the polar region. The average drift velocity is 0.7771 degree day^{-1} (104.9 m s^{-1}). From Figure 14, we found that the drift velocities of filaments with latitudes higher than 50° are divergent,

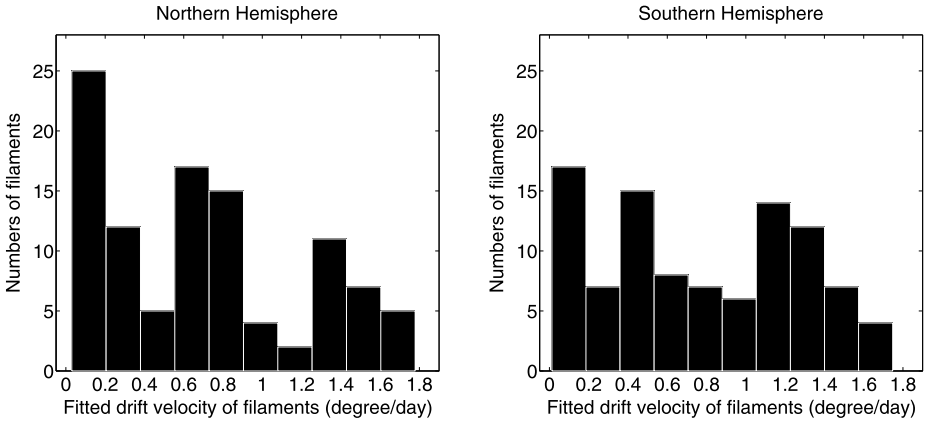


Figure 14 Histograms of the drift-velocity distribution of the filaments whose latitudes are higher than 50°. The left panel plots the northern hemisphere with 103 filaments in total and the right panel the southern hemisphere with 97 filaments in total.

while most of these filaments migrate toward the polar region with relatively high velocities. This result is similar to that of Topka *et al.* (1982). However, these authors found that the poleward drift velocity is about 10 m s^{-1}, which is much slower than ours.

8. Conclusions

We have developed a method to automatically detect and trace solar filaments from Hα full-disk images. The program consists of three parts: First, a preprocessing module is applied to correct the original images. Top-hat enhancement enables us to clearly distinguish the filaments from non-filament features. Second, we introduce the Canny edge-detection method to segment and detect filaments. This method gives us a precise filament edge. Third, our program routines recognize filament features through the morphological operators. We randomly selected 100 images from MLSO observations to test our method, which is demonstrated to be robust and efficient. For the filament detection, the similarity between the machine recognition and human vision is 85 ± 2 %. The solar rotation, the filament position, and the deformation of the filament were considered to trace the filament evolution. The accuracy of the tracing method is about 80 % when the time interval is about one day. In addition, our program can process images not only in different file formats, but also from different observatories.

We used our method to automatically process and analyze 3470 images obtained by MLSO from January 1998 to December 2009. A butterfly diagram of filaments was obtained, in which we can clearly see that filaments move mainly toward the Equator in both hemispheres. To obtain more quantitative results, we calculated the monthly mean latitudes of the filaments whose latitudes are within $0° - 50°$ or higher than 50° in the northern and southern hemispheres. Furthermore, we used our tracing method to trace the evolution of the individual filaments with a latitude higher than 50°. Our main conclusions are these:

- The latitudinal migration of solar filaments have three trends in Solar Cycle 23: from 1998 to 2001 (the solar maximum) the drift velocity is fast. From solar maximum to the year 2006 the drift velocity becomes relatively slow. After 2006, *i.e.* near solar minimum, the migration becomes divergent.

- About 60 % of the filaments with latitudes higher than 50° migrate toward the polar region with relatively high velocities in the northern and southern hemispheres.
- The difference of the latitude migration of the filaments between the northern and southern hemispheres is not obvious in Solar Cycle 23.

We will improve our method to be more reliable and efficient, and apply it to the observational data from our *Optical & Near Infrared Solar Eruption Tracer* (ONSET) in Nanjing University (Fang *et al.*, 2012).

Acknowledgements The authors thank the Mauna Loa Solar Observatory team for making the data available and J.Q. Sun for his help in identifying filaments. We also thank the referee very much for the constructive suggestions, which greatly improved the article in various ways. This work is supported by the National Natural Science Foundation of China (NSFC) under the grants 10221001, 10878002, 10403003, 10620150099, 10610099, 10933003, 11025314, and 10673004, as well as a grant from the 973 project 2011CB811402.

References

Balthasar, H., Vazquez, M., Wöehl, H.: 1986, *Astron. Astrophys.* **155**, 87.
Bernasconi, P.N., Rust, D.M., Hakim, D.: 2005, *Solar Phys.* **228**, 97. doi:10.1007/s11207-005-2766-y.
Canny, J.: 1986, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**(6), 679. doi:10.1109/TPAMI.1986.4767851.
Chen, P.F.: 2008, *J. Astrophys. Astron.* **29**, 179. doi:10.1007/s12036-008-0023-0.
Chen, P.F.: 2011, *Living Rev. Solar Phys.* **8**, 1. www.livingreviews.org/lrsp-2011-1, doi:10.12942/lrsp-2011-1.
Cox, A.N.: 2000, *Allen's Astrophysical Quantities*, Springer, Berlin, 355.
Fang, C., Chen, P.F., Ding, M.D., Dai, Y., Li, Z.: 2012, In: *EAS Publ. Series* **55**, 349. doi:10.1051/eas/1255048.
Fuller, N., Aboudarham, J., Bentley, R.D.: 2005, *Solar Phys.* **227**, 61. doi:10.1007/s11207-005-8364-1.
Gao, J., Wang, H., Zhou, M.: 2002, *Solar Phys.* **205**, 93. ADS:2002SoPh..205...93G, doi:10.1023/A:1013851808367.
Gilbert, H.R., Holzer, T.E., Burkepile, J.T., Hundhausen, A.J.: 2000, *Astrophys. J.* **537**, 503. doi:10.1086/309030.
Gopalswamy, N., Shimojo, M., Lu, W., Yashiro, S., Shibasaki, K., Howard, R.A.: 2003, *Astrophys. J.* **586**, 562. doi:10.1086/367614.
Haralick, R.M., Shapiro, L.G.: 1992, *Computer and Robot Vision*, 1st edn., Addison-Wesley, Longman. ISBN 0201569434.
Jing, J., Yurchyshyn, V.B., Yang, G., Xu, Y., Wang, H.: 2004, *Astrophys. J.* **614**, 1054. doi:10.1086/423781.
Joshi, A.D., Srivastava, N., Mathew, S.K.: 2010, *Solar Phys.* **262**, 425. doi:10.1007/s11207-010-9528-1.
Kong, T.Y., Rosenfeld, A.: 1996, *Topological Algorithms for Digital Image Processing*, Elsevier, Amsterdam, 300.
Labrosse, N., Dalla, S., Marshall, S.: 2010, *Solar Phys.* **262**, 449. doi:10.1007/s11207-009-9492-9.
Labrosse, N., Heinzel, P., Vial, J.-C., Kucera, T., Parenti, S., Gunár, S., Schmieder, B., Kilper, G.: 2010, *Space Sci. Rev.* **151**, 243. doi:10.1007/s11214-010-9630-6.
Lam, L., Lee, S.W., Suen, C.Y.: 1992, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(9), 869.
Li, K.J.: 2010, *Mon. Not. Roy. Astron. Soc.* **405**, 1040. doi:10.1111/j.1365-2966.2010.16508.x.
Li, K.J., Li, Q.X., Gao, P.X., Shi, X.J.: 2008, *J. Geophys. Res.* **113**, 11108. doi:10.1029/2007JA012846.
Lim, J.S.: 1990, *Two-Dimensional Signal and Image Processing*, Prentice Hall, New York, 478.
Martens, P.C.H., Attrill, G.D.R., Davey, A.R., Engell, A., Farid, S., Grigis, P.C., Kasper, J., Korreck, K., Saar, S.H., Savcheva, A., Su, Y., Testa, P., Wills-Davey, M., Bernasconi, P.N., Raouafi, N.-E., Delouille, V.A., Hochedez, J.F., Cirtain, J.W., Deforest, C.E., Angryk, R.A., de Moortel, I., Wiegelmann, T., Georgoulis, M.K., McAteer, R.T.J., Timmons, R.P.: 2012, *Solar Phys.* **275**, 79. doi:10.1007/s11207-010-9697-y.
Martin, S.F.: 1998, *Solar Phys.* **182**, 107. doi:10.1023/A:1005026814076.
McIntosh, P.S.: 1972, *Rev. Geophys. Space Phys.* **10**, 837.
Minarovjech, M., Rybansky, M., Rusin, V.: 1998, *Solar Phys.* **177**, 357. ADS:1998SoPh..177..357M, doi:10.1023/A:1004948832097.
Mouradian, Z., Soru-Escaut, I.: 1994, *Astron. Astrophys.* **290**, 279.
Pratt, W.K.: 2001, *Digital Image Processing*, 3rd edn., Wiley, Hoboken. ISBN 0471221325.
Qu, M., Shih, F.Y., Jing, J., Wang, H.: 2005, *Solar Phys.* **228**, 119. doi:10.1007/s11207-005-5780-1.

Rusin, V., Rybansky, M., Minarovjech, M.: 1998, In: Balasubramaniam, K.S., Harvey, J., Rabin, D. (eds.) *Synoptic Solar Physics* **CS-140**, Astron. Soc. Pacific, San Francisco, 353.
Sakurai, T.: 1998, In: Balasubramaniam, K.S., Harvey, J., Rabin, D. (eds.) *Synoptic Solar Physics* **CS-140**, Astron. Soc. Pacific, San Francisco, 483.
Shih, F.Y., Kowalski, A.J.: 2003, *Solar Phys.* **218**, 99. doi:10.1023/B:SOLA.0000013052.34180.58.
Soille, P.: 1999, *Morphological Image Analysis: Principles and Applications* **11**, Springer, Secaucus, 173.
Tandberg-Hanssen, E. (ed.): 1995, *The Nature of Solar Prominences*, *Astrophys Space Science Lib.* **199**, Springer, Dordrecht.
Topka, K., Moore, R., Labonte, B.J., Howard, R.: 1982, *Solar Phys.* **79**, 231. doi:10.1007/BF00146242.
Wang, Y., Cao, H., Chen, J., Zhang, T., Yu, S., Zheng, H., Shen, C., Zhang, J., Wang, S.: 2010, *Astrophys. J.* **717**, 973. doi:10.1088/0004-637X/717/2/973.
Yuan, Y., Shih, F.Y., Jing, J., Wang, H., Chae, J.: 2011, *Solar Phys.* **272**, 101. doi:10.1007/s11207-011-9798-2.
Zhang, J., Cheng, X., Ding, M.-D.: 2012, *Nat. Commun.* **3**. doi:10.1038/ncomms1753.