Solar Flare Tracking Using Image Processing Techniques

Ming Qu\textsuperscript{1}, Frank Y. Shih\textsuperscript{1}, Ju Jing\textsuperscript{2}, Haimin Wang\textsuperscript{2}

\textsuperscript{1}College of Computing Sciences
\textsuperscript{2}Department of Physics
New Jersey Institute of Technology
Newark, NJ 07102
shih@njit.edu

1. Introduction.

Solar flares are intense, abrupt releases of energy which occur in areas where the magnetic fields are changing due to flux emergence or sunspot motion. Large amounts of high-energy electrons are then accelerated. Most like, as a result of magnetic reconnection, the electrons generate intense X-ray and radio bursts \cite{1}. In an earlier paper we discussed the application of the Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), and Support Vector Machine (SVM) to detect solar flares in real time \cite{2}.

A two-ribbon flare is usually observed after flare onset. With the study of the motion of two-ribbon flares, we yield interesting insight into the flare-CME (Corona Mass Ejection) relationship \cite{3}. In this paper, we develop a new method to track automatically the expanding two-ribbon flare motion using the H\textalpha{} solar image.

This paper is organized as follows. In Section 2, the previous work and current work are given. In Section 3, solar flare property measurement using image segmentation techniques is presented. In Section 4, we demonstrate automatic solar flare motion tracking. In Section 5, the experimental results are shown. Finally, conclusions are made in Section 6.

2. Previous work and current work.

2.1. Automatic solar flare detection.

Automatic solar flare detection plays a key role in real-time space weather monitoring. In our previous paper, we developed a set of nine features for solar images, and use RBF and SVM in addition to MLP to perform classification. Our classification is applied to the H\textalpha{} images. The classification rate is more than 95\%, and running time is less than 0.3 second using our method. Using our classification program, we can detect the beginning and ending of a flare. After a solar flare is detected by the SVM, we obtain the position of a flare using the key pixel. The key pixel is the maximum gray level difference between the current and the previous images. An image may have several key pixels if the image has several flares. Each key pixel is a point in the flare region.

2.2. Automatic solar flare characterization.

In this paper, we demonstrate our current work on automatic solar flare characterization. After detecting the position of a flare, we want to measure the following properties for a flare: center position \((x,y)\) for a flare on the full-disk H\textalpha{} images, number of pixels for each flare; lifetime of each flare tracked by comparing consecutive images; and the expanding orientation and speed calculated by comparing differences between consecutive images. In the first step, we filter noises and align the center of the full disk solar image. In the second step, we use the region growing method, the boundary-based method, the morphology method, small part removing and hole filling to obtain the boundary of a flare. In the third step, we adopt component labeling and model matching to obtain differences between consecutive images.
images. We correlate each pixel on the current model to a pixel on the previous model, and track motion orientation and speed for each pixel pair. By statistical study for a sequence of flare images, we can find the direction, and furthermore find the speed in this direction.

3. Solar flare properties measurement using image segmentation techniques.

3.1. Preprocessing.

In the preprocessing step, we use image enhancement and filtering techniques to obtain high quality images. We use a median filter to remove additive noises, and apply recursive soft morphological filters that possess the desirable property of being less sensitive to additive noises and small variations [4]. Then we detect the center of the solar disk in each image, and align images using their centers. A solar flare is a small feature on the full-disk solar image. In our experiments, we pick a 400 × 400 window which is centered at the key pixel of a solar flare.

3.2. Region growing and adaptive boundary-based method.

We have combined region-based method and adaptive boundary-based methods [5], in addition to morphological image processing techniques and hole filling techniques, to obtain the accurate boundary and the structure of flares.

We use region growing to include the bright part of a flare. The result of region growing is shown in Figure 1b. The region-based method may lose details near the edges of an object since the threshold for the region growing is hard to decide. The boundary-based method is used to detect the edge, such as with the Sobel edge detector [6]. In our experiment, the first-order and second-order derivatives have been used for detecting the boundary of flares.

First-order derivatives of a digital image are based on various approximations of the 2-D gradient. The gradient of an image \( f(x,y) \) is defined as the vector

\[
\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}
\]

(1)

where \( G_x \) is the gradient in x coordination, and \( G_y \) is the gradient in y coordination.

Computation of the gradient of an image is based on obtaining the partial derivatives at every pixel location. For a 3 × 3 area, we used Sobel operator to obtain \( G_x \) and \( G_y \). The second-order derivative is defined as

\[
\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]

(2)

An edge of an object has two sides: dark and bright. If a pixel is not on an edge, we have \( |\nabla f| < T \) (\( T \) is a threshold); if a pixel is on the dark side, we have \( |\nabla f| > T \) and \( \nabla^2 f \geq 0 \); if a pixel is on the bright side, we have \( |\nabla f| > T \) and \( \nabla^2 f < 0 \). The boundary is difficult to be found by a single global threshold. An approach for handling such a situation is to divide the original image into subimages and then utilize a different threshold for each subimage. We adopt the 10 × 10 subimage, and set a different threshold \( T \) according to its brightness and standard deviation. We set the output gray level of the bright side pixels to 3, of the dark side pixels to 2, and of the background to 1. The result of boundary thresholding is shown in Figure 1c.

We use the region growing method to expand the bright side area. The initial seeds for region growing are pixels on the bright side, and the growing criterion is that the gray level of the neighbors is greater than or equal to the seed. Then we add this result image to the result of simple region growing method to have the final segmented image. The result is shown in Figure 1d.

3.3. Morphology, small part removing and hole filling.

There are two important morphological operations: opening and closing. Closing tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in the contour [7]. We apply morphological closing on Figure 1d to erase gap and smooth contour. The 3 × 3 structuring element with all one is used.

In order to remove a small part, we count each individual component in the binary image. If the number of pixels in a component is less than our threshold, we will remove it. In order to fill the small hole inside the flare, we check the pixel number of each individual dark region. If the pixel number is less than our threshold, we remove this dark hole.
Figure 1e shows the result of morphology closing, small part removing and hole filling.

4. Motion Tracking.

4.1. Previous method.

As we discussed in the introduction, tracking ribbon separation speed is very important to understand the physics of magnetic reconnection for flares. The current method for tracking the motion of two-ribbon flares was not automatic. First, researchers made a movie for a two-ribbon flare to obtain the orientation of the flare motion. Second, they drew a short line on each side of the two ribbons. Finally, they calculated the distance between these two lines. Using the distance obtained from each image, they obtained the speed of the two-ribbon flare. This method is accurate but it is not automatic procedure.

4.2. Component labeling and model matching method.

There are two major techniques to estimate the motion field: differential techniques and matching techniques [8]. Our solution is to build a flare model for each image and calculate differences between consecutive images using the previous segmented image results. In order to track the motion automatically, our program gives the same label to the same object on consecutive images according to the size and position of the objects. For the first flare image, we compute the smallest distance between each region. If the smallest distance between two components is less than the threshold, we let them merge; otherwise, we label them as different objects. Then we select the largest object to be the flare region. The result is shown in Figure 2a.

After we build a model for the first image. For the following image, if a region overlaps with the model on the previous image, we count this region to this model. Then we expand and connect the inner edge and keep the outer edge of this flare object. The result is shown in Figure 2b.

4.3. Differences tracking and pixel corresponding.

We obtain differences (D1) using the current model minus the previous model. The result image of differences is shown in Figure 2c. After having the current model, the previous model, and differences between the current model and previous model, we can calculate the expanding orientation and distance for each pixel using the following method:

a) Let \( A = \) the edge of the current model, \( B = \) the edge of the previous model, and \( C=3A+2B \). We obtain the gray level 5 on \( C \), where \( A \) and \( B \) are overlapped, \( C=3 \) where pixels on the current model, and \( C=2 \) where pixels on the previous model. We track the corresponding pixel \( (C=2) \) on the previous model for each pixel \( (C=3) \) on the current model. For example, a pixel \( (x1, y1) \) on the current model, we can get the nearest pixel \( (x2, y2) \) on the previous model to it.

\[ d = \sqrt{(x2-x1)^2 + (y2-y1)^2} \]  

and the expanding direction by

\[ \theta = \tan^{-1}\left(\frac{y1-y2}{x1-x2}\right) \]  

b) We can obtain the expanding direction for each pair of corresponding pixels by

\[ \theta = \tan^{-1}\left(\frac{y1-y2}{x1-x2}\right) \]  

and the expanding distance by

\[ d = \sqrt{(x2-x1)^2 + (y2-y1)^2} \]  

We record the number of pixels and median of expanding distance \( d \) for each angle interval \( j \), where \( j=1 \) denotes the angle between \([-90^0, -80^0]\), \( j=2 \) denotes between \([-80^0, -70^0]\), and so on.

\[ d = \text{median}(d_{i,j}) \]

where \( i \) is the sequence number of a flare image, and \( j \) denotes the flare motion direction.

c) When a flare is over, we calculate the two-ribbon flare motion using pixels’ motion. First, we need to calculate the direction of expanding motion. We check the number of pixels in each direction, and pick the direction where the number of pixels is maximal. Then we can obtain the expanding speed by

\[ s_{i,j} = d_{i,j} - d_{(i-1),j} \]  

5. Experimental results.

We develop the programs in Interactive Data Language (IDL) by Research Systems, Inc. The program runs on a DELL Dimension L733r with CPU time 733 Mhz and memory of 256 Mbytes under Windows 2000. The process of image segmentation spends less than 10 seconds for each image, and the process of motion tracking spends less than 3 seconds for each image. See the computation time for the following methods in Table 1: (1) reading image and preprocessing, (2) region growing and adaptive boundary process, (3) morphology closing process, (4) small part removing and hole filling process, (5) component labeling and model
matching (6) pixel corresponding and motion tracking. We select Hα two-ribbon flare images observed on May 21, 2002. Data were obtained at BBSO. In the following section, we demonstrate the results of flare detection and characterization around the time of 20:40:21UT.

### Table 1: Computation time (seconds).

<table>
<thead>
<tr>
<th>Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>3.9</td>
<td>6.2</td>
<td>0.8</td>
<td>4.7</td>
<td>6.3</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Figure 1: Image segmentation results for flare peaked at 20:40:21 on 21 May 2002. a) Original image, b) result of region growing, c) result of adaptive threshold boundary method, d) result of boundary growing method, e) result of morphological closing, small part removing and hole filling.

Figure 2: a) Result of component labeling, b) result of the final model, c) result image of differences between the current and previous images.

### 6. Conclusions.

In this paper, automatic solar flare detection and characterization are presented. Region growing and boundary-based method can be combined to find detailed properties of solar flares. Moreover, we use the morphology technique, small part removing and hole filling to further improve the performance. Component labeling and model matching techniques are used to characterize the main region of a flare. We have also proposed a motion tracking method to compute the orientation and speed of two-ribbon flares. The experimental results show that we can obtain accurate results. The process of image segmentation and motion tracking take less than 30 seconds for each image. Our automatic process is valuable for the studies of solar flares since this process dramatically improves efficiency and accuracy comparing with the old method by human interactive.

### References.
