ABSTRACT

AUTOMATIC SOLAR FEATURE DETECTION USING IMAGE PROCESSING AND PATTERN RECOGNITION TECHNIQUES

by
Ming Qu

The objective of the research in this dissertation is to develop a software system to automatically detect and characterize solar flares, filaments and Corona Mass Ejections (CMEs), the core of so-called solar activity. These tools will assist us to predict space weather caused by violent solar activity. Image processing and pattern recognition techniques are applied to this system.

For automatic flare detection, the advanced pattern recognition techniques such as Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), and Support Vector Machine (SVM) are used. By tracking the entire process of flares, the motion properties of two-ribbon flares are derived automatically. In the applications of the solar filament detection, the Stabilized Inverse Diffusion Equation (SIDE) is used to enhance and sharpen filaments; a new method for automatic threshold selection is proposed to extract filaments from background; an SVM classifier with nine input features is used to differentiate between sunspots and filaments. Once a filament is identified, morphological thinning, pruning, and adaptive edge linking methods are applied to determine filament properties. Furthermore, a filament matching method is proposed to detect filament disappearance. The automatic detection and characterization of flares and filaments have been successfully applied on Hα full-disk images that are continuously obtained at Big Bear Solar Observatory (BBSO). For automatically detecting and classifying CMEs, the image enhancement, segmentation, and pattern recognition techniques are applied to Large Angle Spectrometric Coronagraph (LASCO) C2 and C3 images.

The processed LASCO and BBSO images are saved to file archive, and the physical properties of detected solar features such as intensity and speed are recorded in our

database. Researchers are able to access the solar feature database and analyze the solar data efficiently and effectively. The detection and characterization system greatly improves the ability to monitor the evolution of solar events and has potential to be used to predict the space weather.

AUTOMATIC SOLAR FEATURE DETECTION USING IMAGE PROCESSING AND PATTERN RECOGNITION TECHNIQUES

by
Ming Qu

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

January 2006

APPROVAL PAGE

AUTOMATIC SOLAR FEATURE DETECTION
USING IMAGE PROCESSING AND PATTERN RECOGNITION TECHNIQUES

Ming Qu

Dr. Frank Y. Shih, Dissertation Advisor — Date
Professor of Computer Science, Director of Computer Vision Laboratory, NJIT

Dr. Haimin Wang, Dissertation Co-advisor — Date
Distinguished Professor of Physics, NJIT

Dr. Alexandros Gerbessiotis, Committee Member — Date
Associate Professor of Computer Science , NJIT

Dr. Cristian Borcea, Committee Member — Date
Assistant Professor of Computer Science , NJIT

Dr. Qun Ma, Committee Member — Date
Assistant Professor of Computer Science , NJIT

Dr. Carsten Denker, Committee Member — Date
Assistant Professor of Physics, NJIT

Dr. Phil Goode, Committee Member — Date
Distinguished Professor of Physics, NJIT

BIOGRAPHICAL SKETCH

Author: Ming Qu

Degree: Doctor of Philosophy

Date: January 2006

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, New Jersey, 2006
- Master of Science in Computer Science,
  New Jersey Institute of Technology, Newark, New Jersey, 2002
- Bachelor of Science in Computer Science,
  Beijing Polytechnic University, Beijing, China, 1997

Major: Computer Science

Publications in Refereed Journals:

Qu, M., Shih, F. Y., Jing, J. and Wang, H.: 2003, Automatic Solar Flare Detection using MLP,RBF and SVM, *Solar Phys.*, 217, 157.

Qu, M., Shih, F. Y., Jing, J. and Wang, H.: 2004, Automatic Solar Flare Tracking Using Image Processing Techniques, *Solar Phys.*, 222, 137.

Qu, M., Shih, F. Y., Jing, J. and Wang, H.: 2005, Automatic Solar Filament Detection Using Image Processing Techniques, *Solar Phys.*, 228, 121.

Cao, W., Denker, C. J., Wang, H., Ma, J., Qu, M., Wang, J. and Goode, P. R.: 2004, Characteristic Evaluation of a Near-Infrared Fabry-Perot Filter for the Infrared Imaging Magnetograph (IRIM), *Proceedings of the SPIE*, 5171, 307.

Jing, J., Qiu, J., Lin, J., Qu, M., Xu, Y. and Wang, H.: 2004, Magnetic Reconnection Rate and Flux-Rope Acceleration of Two-Ribbon Flares, *Astrophys. J.* , 204, 3903.

Liu, L., Lim, H C., Qu, M., Federici, J. F., Thomas, G. A., Gleskova, H. and Wagner, S.: 2004, Resistance to Cracking of a Stretchable Semiconductor: Speed of Crack Propagation for Varying Energy Release Rate, *Materials Research Society Symp. Proc.*, 795.

Presentations at Scientific Conferences and Meetings:

Qu, M., Shih, F. Y., Jing, J. and Wang, H.: 2003, Automatic Solar Flare Detection using MLP, RBF and SVM, *34th Meeting of the Solar Physics Division*, John Hopkins University, Columbia, MD.

Qu, M., Shih, F. Y., Jing, J. and Wang, H.: 2004, Solar Flare Tracking Using Image Processing Techniques, *IEEE International Conference on Multimedia and Expo (ICME)*, Taiwan.

Qu, M., Shih, F. Y., Jing, J., Wang, H. and Rees, D.: 2004, Automatic Solar Flare Tracking, *Eighth Inter. Conf. on Knowledge-based Intelligent Information and Engineering Systems*, New Zealand.

Wang, H., Qu, M., Shih, F. Y., Denker, C., Gerbessiotis, A., Lofdahl, M., Rees, D. and Keller, C.: 2004, Innovative Information Technology For Space Weather Research, *204th American Astronomical Society Meeting*, Denver, CO, 36, 755.

Qu, M., Shih, F. Y., Jing, J. and Wang, H.: 2005, Automatic Detection of Corona Mass Ejections Using Pattern Recognition Techniques, *Joint Assembly AGU and AAS Solar Physics*, New Orleans, Louisiana.

This dissertation is dedicated to Ju

ACKNOWLEDGMENT

I would like to thank my dissertation advisor Dr. Frank Y. Shih. He had given me excellent guidance since I was a master student. His many suggestions on how to deal with difficult technical issues as well as his vast knowledge were invaluable to me.

I thank my co-advisor Dr. Haimin Wang with the deepest of respect. His help, stimulating suggestions and encouragement helped me in all the time of research. I believe no words could fully encompass the amount of gratitude I have for his supervision and support for imparting wisdom that has inspired me to be a life-long learner and a passionate seeker of knowledge.

I thank Dr. Carsten Denker for helping me with my several papers. His help and advice improved the quality of my papers by orders of magnitude. I would like to thank Dr. Phil Goode in the Department of Physics, Dr. Alex Gerbessiotis, Dr. Cristian Borcea and Dr. Qun Ma in the Department of Computer Science for taking time to be committee members. They spent a lot of precious time reviewing my dissertation and helping me to improve it. I appreciate their valuable help.

During the years 2002-2005, I have worked for more than 3 years in the group of Innovative Information Technology for Space Weather Research (ITR) project which is supported by National Science Foundation (NSF) under grants IIS-0324816. I appreciate the opportunity of being a member of this project and give thanks to remarkable people who provide support, knowledge, and friendship. I give my thanks to: Drs Wenda Cao, Guo Yang, Jeongwoo Lee, Jiong Qiu in the Center for Solar and Terrestrial Research; and Chao-fa Chuang, Yi-Ta Wu, Kai Zhang in the Computer Vision Lab. Dr. Yan Xu deserves my warmest thanks for many discussions on the research and application. Also, I thank Dr. Tom Spirock for helping me with some of my paper revision. I thank the BBSO observing staff for their assistance and support.

I am most grateful to my parents Guangli Qu and Shuqing Zhang. Without their

strength and courage I would not have been able to complete my studies. Especially, I would like to give my special thanks to my wife Ju Jing whose patience and love enabled me to complete this work.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Facilitated by the major advance in solar observation, high quality and large quantity of solar images are obtained by space-based and ground-base observatories. Solar events seen in those images are used by scientists to study and extend scientific results. To process a large amount of solar images efficiently and effectively, many applications based on advanced image processing and pattern recognition techniques have been developed. In the following sections, the work on solar flares, filaments and Corona Mass Ejections (CMEs) are introduced.

This work provides a tool to rapidly scan through a large amount of solar data and efficiently study statistical properties of solar events including solar flares, filaments and CMEs. Hα full-disk images from Big Bear Solar Observatory and Large Angle and Spectrometric Coronagraph Experiment (LASCO) images from Solar and Heliospheric Observatory (SOHO) are used. This application can detect these phenomena of solar activities in either off-line or real-time, and also provides a graphical user interface using Data Base Management System (DBMS) and web site. In addition to feature detection and extraction on the images, the classification of solar flares, sunspots and CMEs is first proposed by this work. Pattern recognition techniques such as neural networks and Support Vector Machines (SVMs) are used in the solar feature classification.

1.1 Image Processing and Pattern Recognition

The field of digital image processing refers to processing digital images by means of a digital computer (Gonzalez and Woods 2002). The image processing includes preprocessing the image such as image restoration and image enhancement, extracting the individual regions such as region-based and edge-based image segmentation, and describing the indi-

vidual regions such as morphology and motion tracking. Image processing becomes more and more important in today's life. Using advanced digital cameras, people can capture a large amount of images and videos. In some scientific fields, scientists are facing a large amount of digital images. There is no doubt that automatic processing of those images are crucial to advance the research. In the last decade, image processing invokes a large of interests in the Solar Physics field. Researchers in the Solar Physics field use different image processing application to obtain accurate results and fasten their research.

Pattern is the universal concept in intelligence and discovery (Looney 1997). Pattern Recognition (PR) partially belongs to Artificial Intelligence techniques space(AI). In order to recognize the patterns from 1D, 2D and multi-dimensional data, PR analyzes priori knowledge to obtain statistical information and categorizes the data according to experiential knowledge base. PR is widely used for automatic feature extraction and comparison systems such as human-face recognition, voice identification and handwriting analysis. Neural Networks (NNs) and Support Vector Machines (SVMs) are two popular tools for the PR study. In this dissertation, NNs and SVMs are used for the solar feature classification.

1.2 Space Weather

Space weather in the near earth space has societal effects on human systems such as safety of spacecrafts and astronauts, navigation of Global Positioning System (GPS), and transmission of electricity grids. The change of space weather is the result of some sudden solar activities. The activities of the Sun change the radiative and particle outputs of the Sun, producing corresponding changes in the near-earth space environment as well as the earth surface (Wang *et al.* 2004). The importance of solar events such as filament eruptions, solar flares and Coronal Mass Ejections (CMEs), specially earth-direct CMEs, has been proven to affect space weather, and furthermore the earth (Jing *et al.* 2004; Brueckner *et al.* 1998; Webb *et al.* 2000; Cane *et al.* 2000).

A typical CME is a high-brightness enhanced and fast moving feature seen in coronagraph images (Howard *et al.* 1985). When a large amount of erupted mass along with magnetic field passes through the geo-space, the ensuing geomagnetic storms are produced. In order to forecast the geomagnetic storm, it is essential to forecast earth-direct CMEs in real-time. Therefore, monitoring solar flares and filaments is very important for establishing the forecast model. When this model is built, scientists are able to forecast CMEs by using real-time detection and characterization results of filaments and flares, and eventually forecast geomagnetic storms promptly and accurately.

1.3 Innovative Information Technology for Space Weather Research

Innovative information technology for space weather research (ITR) that is funded by National Science Foundation (NSF). Figure 1.1 shows the structure of ITR project. Our project consists of three modules: artificial intelligence group for automatic solar feature detection, real-time imaging for high quality of images, forecast modeling and data publication. This project involves professors, researchers, graduates, and undergraduates in physics and computer science from U.S. and Europe.

There are three goals in the ITR project.

1. Use phase diversity speckle techniques to obtain high quality images in near real-time. Parallel computing techniques are used to expedite the image processing.

2. Use advanced image processing and pattern recognition techniques including neural networks, support vector machines, image segmentation and motion tracking to detect and characterize three important solar events: filament disappearances, flares, and CMEs, in near real-time.

3. Develop a tool to forecast solar activities and orientation of magnetic clouds in interplanetary space. Setup a database and a web site to publish our image data, feature detection and characterization results, and forecasting results. Provide the advanced

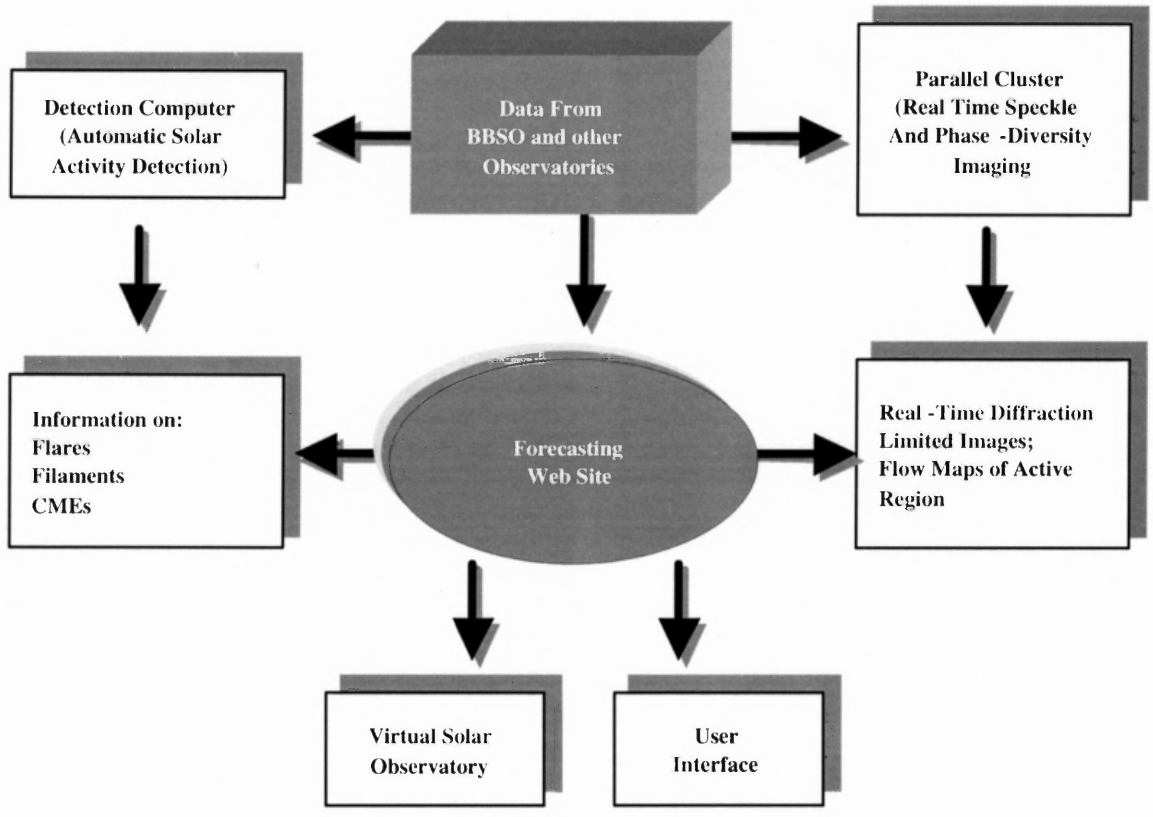

Figure 1.1 Infrastructure of ITR project.

user interface and high speed server. Further, integrate with solar activity report and space weather prediction web pages at Big Bear Solar Observatory (BBSO).

My dissertation focuses on the automatic solar feature detection (part 2), and database and web portal (part 3) in this project. The primary data sources are full-disk Hα images from Big Bear Solar Observatory (BBSO) for solar flare and filament detection and Large Angle and Spectrometric Coronagraph Experiment (LASCO) images from Solar and Heliospheric Observatory (SOHO) for the CME detection. Figure 1.2 shows a picture of BBSO.

1.4 Automatic Solar Flare Detection and Characterization

A solar flare is the intense, abrupt release of energy which occurs in areas on the sun where the magnetic field is changing due to flux emergence or sunspot motion (Zirin 1988). The

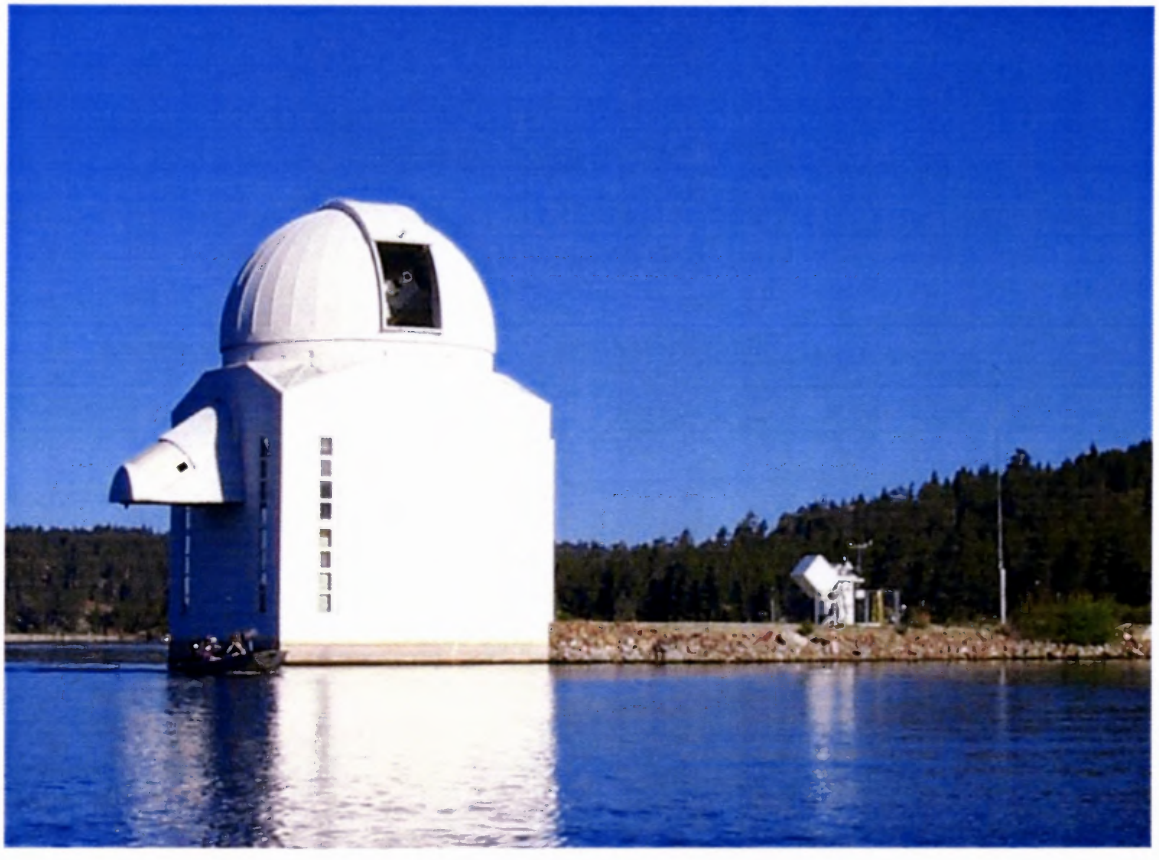

Figure 1.2 Big Bear Solar Observatory (BBSO) is located in the Big Bear mountain lake in the southern California, and operated by New Jersey Institute of Technology. BBSO is an ideal solar observational site because of the stable clear weather condition and geographical location (URL: http://bbso.njit.edu).

high-energy electrons, accelerated by solar flares, generate intense X-ray and radio bursts. Typically, the brightness increases for several minutes, followed by a slow decay which lasts between 30 minutes to 1 hour. Figure 1.3 shows a Full disk Hα (Hydrogen-Alpha) image with a flare detected on the low-left corner. Flares are often associated with CMEs.

Automatic solar flare detection is the key to monitor space weather. It is very challenging since the features of the solar flares are complicated. We need a system that can handle the solar flares' complexity and scalability. Veronig et al. (2000) proposed a method for automatic flare detection by applying a combination of region-based and edge-based segmentation methods. Simple region-based methods are applied for a tentative assignment of flare activity, making use of one of the decisive flare characteristics: the high

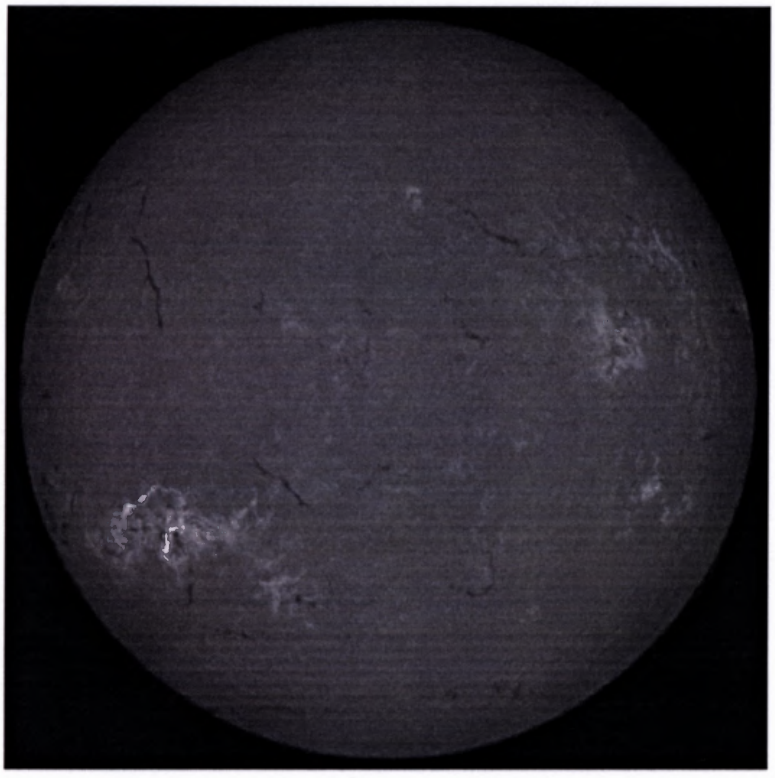

Figure 1.3 A two-ribbon solar flare appears on the left side of the Hα image.

intensities. Borda et al. (2001) presented a method for automatic detection of solar flares using the neural network technique. The network used is the multi-layer perceptron (MLP) with back-propagation training rule. They used a supervised learning technique and required a lot of iterations to train the network. In 2003, we developed a set of nine features for solar images and use Radial Basis Function (RBF) and Support Vector Machine (SVM) in addition to MLP to perform classification (Qu et. al, 2003). We found that SVM can obtain the best classification rate.

Automatic solar flare characterization is also important for flare study. Because of the difficulty of flare image processing, the flare study becomes a time-consuming job. Veronig et al. (2000) proposed a method for flare property measurement using region-based and edge-base segmentation methods. The tracking ribbon flares separation speed is very important to understand the physics of magnetic reconnection for flares. In the previous work, the motion of two-ribbon flares is not measured automatically (Qiu et al., 2003).

First, researchers outlined manually the trajectories of the flare ribbon fronts. Then, the velocity of the ribbon expansion is determined as the time derivative of the distance between the centroids of successive trajectories. They also take a running mean over several time frames. The uncertainties generated by this method are discussed by Qiu et al. (2003). A major disadvantage is that the method is not entirely automatic. In 2004, we developed solar flare characterization and two-ribbon flare tracking program using image segmentation, morphology and motion tracking techniques.

1.5 Automatic Solar Filament Detection and Characterization

Filaments are located in the corona, the outer layer of the Sun. Filaments are low temperature and high density feature compared with the local corona values. In Hα images, filaments are seen as dark ribbons against the bright solar disk which is shown in Figure 1.4. Filament eruptions, flares and Corona Mass Ejections (CMEs) are the most important solar events as far as space weather effects are concerned linking to solar eruptions, major interplanetary disturbance and geomagnetic storms (Gosling *et al.* 1991). Increasing observational evidence that there is an association between filament eruptions, flares and CMEs confirms that they are different manifestations of one physical process at different evolutionary stages (Gilbert *et al.* 2000; Gopalswamy *et al.* 2003). In order to gain a better understanding of CMEs and furthermore, to develop the geomagnetic storm predictions, it is essential to find early manifestations of CMEs in the solar atmosphere (i.e., filament eruptions, flares) (Jing *et al.* 2004). Therefore, detecting and tracking filaments are important for developing the prediction model.

Previous work on the detection of filaments is based on thresholding methods. Gao et al. (2002) detected the filament using global thresholds which are chosen by median values of the image intensity. This method can not handle the low contrast filaments and produce unstable results of filaments. Shih and Kowalski (2003) adapt local thresholds which are chosen by median values of the image intensity to extract filaments. The local

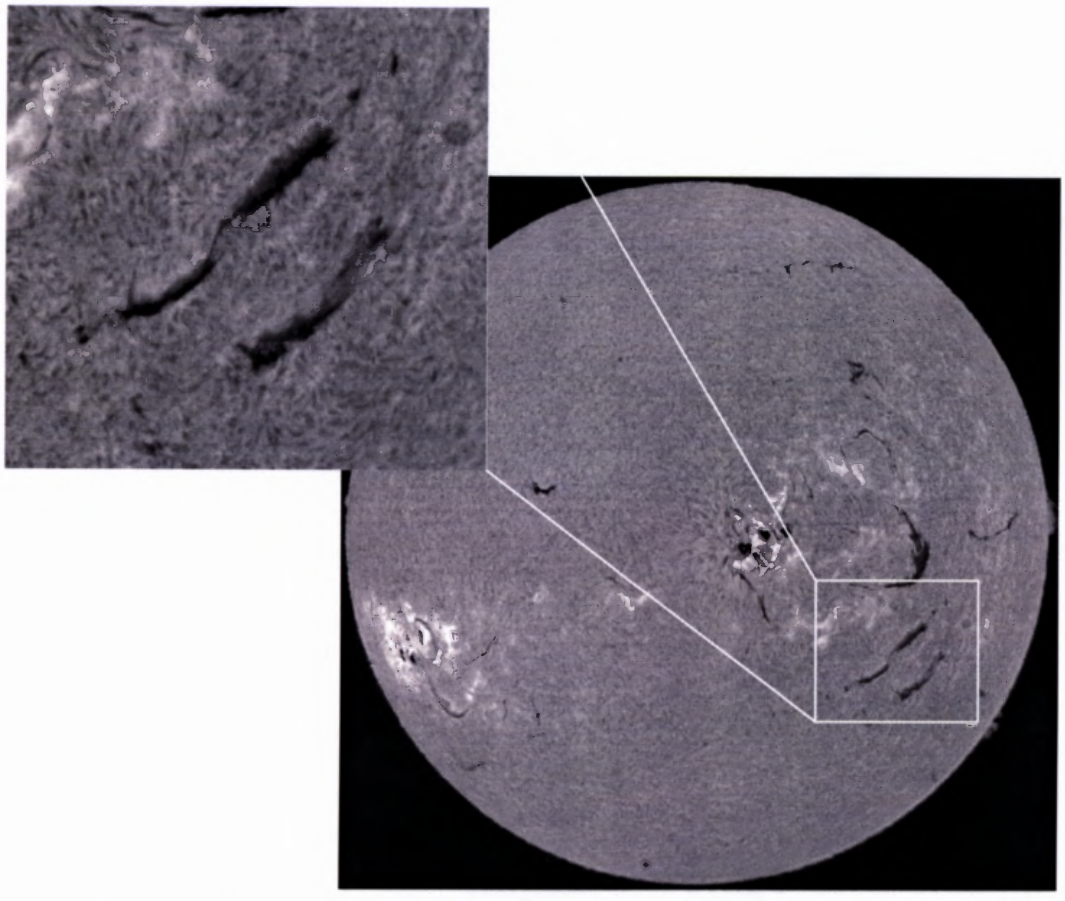

Figure 1.4 Filaments in an Hα image.

method is effective in measuring the low contrast filaments and small features. However, the observational background may change from time to time. It is difficult to set the local thresholds for each sub-image. The median criterion for the threshold selection cannot guarantee robust results in that the bright features on images can significantly affect the value of thresholds associated with the median value. In order to distinguish sunspots from filaments, Shih and Kowalski (2003) proposed the sunspot removal method based on darkness. But the single darkness classifier is not able to yield accurate result because of the limb darkening and nearby bright region effects.

A procedure for the automatic detection of filaments and their disappearance is proposed. Full-disk Hα images are used as the data set for the proposed procedure. During the last few years, the Big Bear Solar Observatory (BBSO) has developed a new generation of well-calibrated, photometric Hα full-disk observations (Denker *et al.* 1999), which correct the limb darkening to enhance features on the disk and above the limb.

Generally speaking, the procedure consists of the following steps.

1. In the first step, the `stabilized inverse diffusion equation` (SIDE), that preserves the high frequencies of the image via the iteration of nonlinear partial differential equations (PDEs) (Pollak et al., 2000), was applied to emphasize and sharpen the features (i.e., filaments) for the further processing.

2. In the second step, a new algorithm is proposed for automatic threshold selection from the result of edge detection. The edges of the filaments were detected by Sobel operator. Then compute the segmented regions for each potential threshold. The difference regions made by two successive thresholds were computed iteratively. The optimal threshold is the one that best matches with the edge detected previously.

3. In the third step, a new and efficient feature-based classifier, the Support Vector Machine (SVM), was used for distinguishing sunspots from filaments. For the present study, a sunspot is represented by nine features. As far as we know, it is the first

time that SVM is applied for sunspot detection. Experimental results show that this method significantly improved the classification rate in comparison with other methods (Shih & Kowalski, 2003).

4. Next, because some isolated regions might be the parts of the main body of one filament, the relationship between the segmented regions is checked to determine if they should be integrated. This was achieved by morphological closing, thinning, pruning and adaptive edge linking methods. Consequently, the spines and the footpoints of filaments were obtained.

5. Finally, the filament disappearance is detected by comparing the results, obtained from the step 4, on two successive days.

1.6 Automatic CME Detection and Characterization

A CME is a large-scale rearrangement of the Sun's magnetic structure which can lead to large amount of ejected material from the Sun's outer atmosphere (Wang 2005). Each CME may carry away a mass of up to 10^{13} kg and release up to 10^{25} J energy from coronal magnetic fields (Harrison 1994, 1995).

CMEs have been observed from Large Angle and Spectrometric Coronagraph Experiment (LASCO) images by Solar and Heliospheric Observatory (SOHO) which is shown in Figure 1.5. The European Space Agency (ESA) and the US National Aeronautics and Space Administration (NASA) operate the SOHO project. LASCO imaging provides three different coronagraphs: C1, C2, and C3. The observational capability of C1, C2 and C3 images are ranged from 1.1 to 3, from 1.5 to 6 and from 3.7 to 30 solar radii, respectively. A more detailed description of LASCO can be found in Brueckner et al. (1995).

Berghmans (2002) introduced his automatic CME detection procedure using image processing methods based on LASCO images. He also demonstrated the difficulty of detection as no CME catalog could have 100% success rate. Furthermore, Robbrecht and

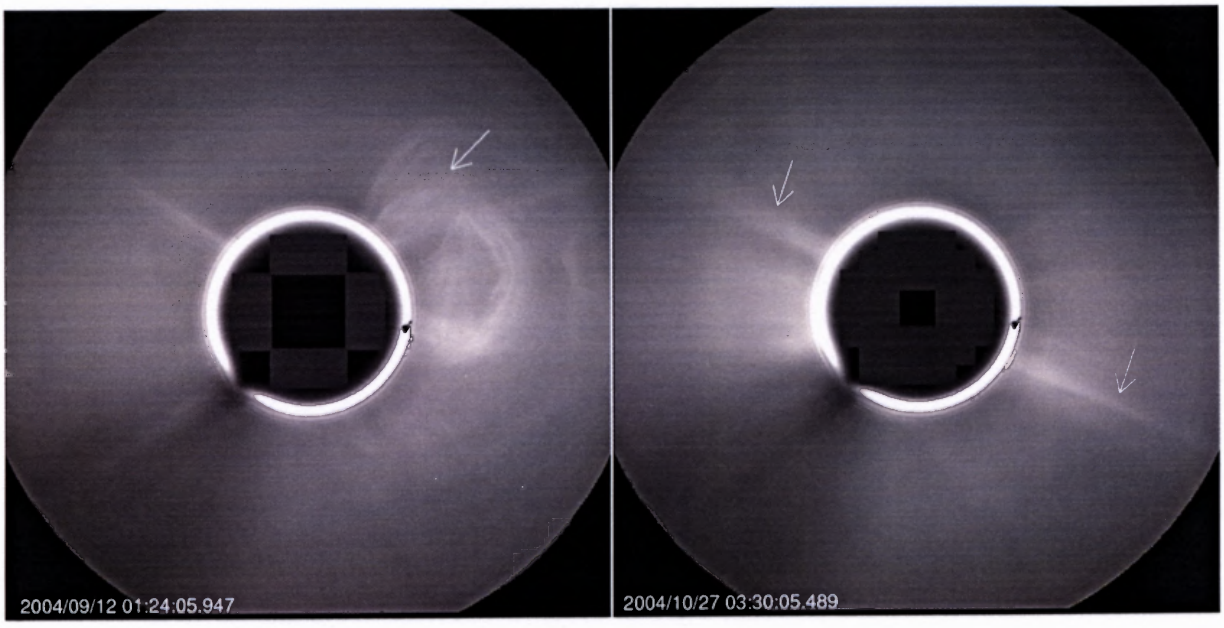

Figure 1.5 CMEs on LASCO C2 images

Berghmans (2004) improved their previous method and presented their first application for automatically detecting CMEs in image sequences from LASCO. They used Hough transform to detect lines on the running difference images.

In this dissertation, an automatic procedure to detect and characterize CMEs is proposed using LASCO C2 and C3 images. First step is to normalize the LASCO images, remove noises and use a reference image to produce running differences. The second step is designed to characterize CME events. CME regions are segmented using running differences, and are further reformed to angular images along the occulting disk limb. By analyzing three consecutive LASCO images and their differences, the physical properties of CME events such as intensity, height, angular width of span, and speed are derived. The last step is to distinguish the strong, medium, and weak CMEs. In this study, Support Vector Machine (SVM) classifiers are incorporated with the CME properties to classify strong CMEs. The real-time CME detection and classification results are recorded in a database to be available to solar physicists. By comparing other CME catalogs including LASCO and CACTUS, our CME catalogs are proven to be more complete and accurate. The CME detection and classification will be helpful for the CME study and forecast of geo-magnetic storms.

1.7 Solar Feature Database and Web Portal

The third module of our ITR project is to build the space weather forecast module and data publish. I work on the data archive and data publish as a principle member. We build a web site where users can access, download, compare and analyze the solar data through Internet. In the future, our database will become a part of sources for Virtual Solar Observatory (VSO) (Hill *et al.* 2004). NASA Goddard Space Flight Center has built a portal of the VSO where users can search the solar data by observation time, physical observable, instrument and data provider, and spectral range. The data are stored locally in VSO database or remotely in other data providers' database and organized by VSO web

site with a search engine.

Instead of storing a large amount of solar data, our database is focused on the properties of solar feature. The web site of this project has been setup to publish our data. This web site provides users an efficient portal to access the database. Through our web site, users not only can search and download raw data, but also view the processed data and feature detection results. The processed data is the result of feature segmentation that can reduce the time on analysis of the solar feature. The complete detection results can help researchers analyze solar activity statistically and efficiently. There are three steps to build our database. First, the automatic detection procedures are invoked to detect solar filaments, flares and CMEs in near real-time. The BBSO full-disk images and SOHO LASCO images are automatically downloaded by our Unix shell programs. Second, the properties of solar features are tracked, then are stored into our MySQL 5.0 database under the Linux machine. Third, the data are published to all Internet users through the web server of Tomcat 5.0. Figure 1.6 shows a picture of our web portal.

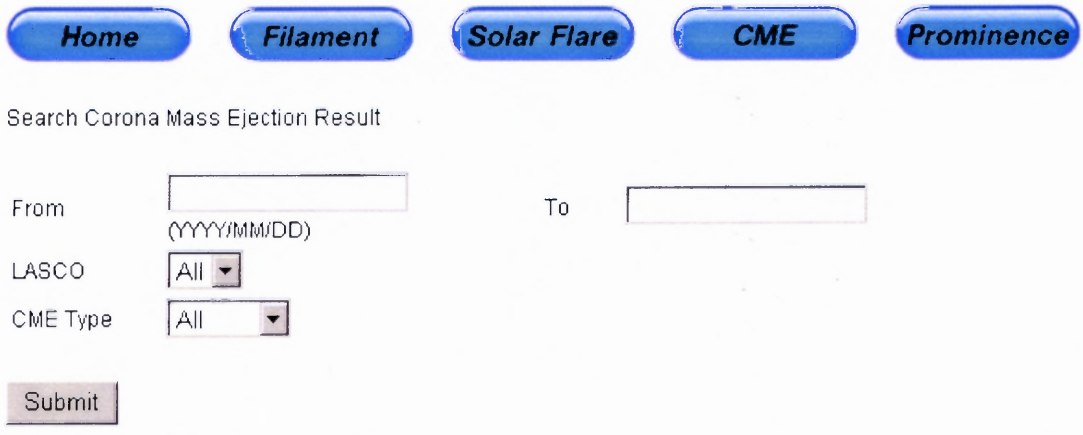

Figure 1.6 The web page for searching CMEs (URL: http://filament.njit.edu).

1.8 Summary

The goal of this study is to develop automatic procedures to detect the solar features in real-time and characterize the features using solar images. This work adopts advanced image enhancement, segmentation, mathematical morphology and pattern recognition methods. The real-time data results will be part of the source to the Virtual Solar Observatory (VSO) where researchers are able to analyze the solar data more efficiently and effectively.

In Chapter 2, image enhancement, segmentation and morphological methods are presented. In Chapter 3, an introduction of Pattern Recognition techniques is presented. The detail procedure and results for solar flare detection and characterization are shown in Chapter 4. The detail procedure and results for solar filament and filament eruption detection and characterization are shown in Chapter 5. The detail procedure and results for CME detection and characterization are shown in Chapter 6. Finally, the summary and future work are presented.

CHAPTER 2

IMAGE PROCESSING TECHNIQUES

2.1 Image Enhancement

The idea of image enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image (Gonzalez and Woods 2002). In the filament detection, some low contrast filaments are difficult to be segmented. To enhance the contrast of the filament, an edge enhancement technique, called *stabilized inverse diffusion equation* (SIDE) which was introduced by Pollak et al. (2000), is adopted.

2.1.1 Linear Multiscale Filtering

The basic idea of multiscale filtering is to simplify a 1D signal or a 2D image via iterative filtering in such a way that the output of each filtering stage is a "coarser" version of its input (Pollak *et al.* 2000). The iteration number is called the *scale*. The structure of multiscale filtering is shown in Figure 2.1 . The input signal u^0 goes through the filter A_1 producing u^1, and then repeat the same procedure producing u^2, u^3 and so on. The input signal is a sequence of real numbers at all time instances: $u^0(-1)$, $u^0(0)$, $u^0(1)$, etc., which is denoted by a vector $\mathbf{u}^0$. In a linear scale space, the filter A_t is linear. The 1-D multiscale filter is used to process signals using a Partial Differential Equation (PDE) of the following form:

$$\begin{aligned} u_t &= A_t(u, u_x, u_{xx}) \\ u(0,x) &= u^0(x). \end{aligned} \tag{2.1}$$

The variable t is called *scale* or *time*, u_x is the first partial derivative of u with

respect to x, and u_{xx} is the second partial derivative of u with respect to x. This equation is called a *linear diffusion* equation or *heat* equation.

In the 2D case, an image u^0 that has two variables, x and y, is processed using a PDE of the following form:

$$\begin{aligned} u_t &= A_i(u, u_x, u_y, u_{xx}, u_{yy}, u_{xy}) \\ u(0,x,y) &= u^0(x,y), \end{aligned} \tag{2.2}$$

which generates the scale space $u(t,x,y)$, for $0 \leq t < \infty$. Two useful operators of the PDEs are gradient and divergence. The gradient of $u(t,x,y)$ is a vector consisting of two partial derivatives of u with respect to the spatial variables x and y:

$$\nabla u \stackrel{def}{=} (u_x, u_y)^T, \tag{2.3}$$

where T denotes the transpose of a vector. The norm of the gradient is:

$$|\nabla u| \stackrel{def}{=} \sqrt{u_x^2 + u_y^2}. \tag{2.4}$$

The divergence of a vector function $(u(x,y), v(x,y))^T$ is:

$$\vec{\nabla} \cdot \begin{pmatrix} u \\ v \end{pmatrix} \stackrel{def}{=} u_x + v_y. \tag{2.5}$$

The principle of image multiscale filtering is to do "edge detection" by differentia-

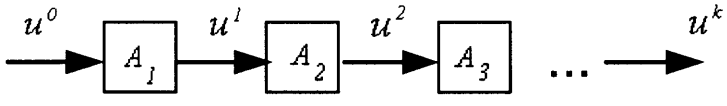

Figure 2.1 Multiscale filtering: the input signal u^0 goes through a series of filters $A_1, A_2, ..., A_k$, producing coarser versions $u^1, u^2, ..., u^k$.

tion; the image must be filtered (or smoothed) before differentiation. Thus, edge detection becomes multiscale filtering, and edge detection theory is directly deduced from multiscale filtering theory (Perona and Malik 1990). In the discrete setting, the solution of Equation (1) is a low-pass filtered version of the initial data (Weickert *et al.* 1997). More specifically,

$$u(t,x) = u(0,x) * G_t(x), \tag{2.6}$$

where "$*$" is a convolution operator and $G_t(x)$ is a kernel function. For example, $G_t(x) = \frac{1}{2\sqrt{\pi t}} \exp(-\frac{x^2}{4t})$ is a Gaussian kernel of variance $2t$. This kernel function such as Gaussian can also be applied to 2D images. A linear filter has been known that it can remove noise, but at the same time blur edges.

2.1.2 Nonlinear Multiscale Filtering

To preserve high frequencies, nonlinear PDEs are developed for processing images. Perona and Malik in 1990 improved the linear multi-scale paradigm with a accurate edge detection. The main idea of Perona and Malik is to introduce a part of edge detection in the filtering itself, allowing an interaction between scales from the beginning of the algorithm. They proposed to replace the heat equation by a nonlinear equation. The Perona-Malik equation can be expressed as follows.

$$\begin{aligned} u_t &= \overrightarrow{\nabla} \cdot \{G(|\nabla u|)\nabla u\}, \\ u(0,x,y) &= u^0(x,y), \end{aligned} \tag{2.7}$$

where $\overrightarrow{\nabla}$ and ∇ are the divergence in Equation (5) and the gradient in Equation (3), respectively. The nonlinear diffusion coefficient $G(|\nabla u|)$ is chosen to suppress diffusion in high-gradient regions which are identified as edges and to encourage diffusion in low-

gradient regions which are identified as noises (Pollak 2002). G is small near edges, so as to minimize the smearing effect found in the linear heat equation. Such an equation affects a segmentation in the sense that high-gradient edges are enhanced, and small-gradient edges become more uniform (Kichenassamy 1997).

The 1-D equation can be rewritten as:

$$\begin{aligned} u_t &= \frac{\partial}{\partial x}\{F(u_x)\}, \\ u(0,x) &= u^0(x), \end{aligned} \tag{2.8}$$

where $F(u_x) = G(|u_x|)u_x$, and F is odd and tends to zero at infinity. Similarly, F can be applied to 2-D.

The large edges with $|u_x| \geq K$ will be enhanced, while the small edges with $|u_x| < K$ will be diffused. Note that the Perona-Malik equation is apparently not implementable. An existing and unique solution is used to obtain the close results at finite time intervals for any initial condition. To achieve this, a semi-discrete version of Perona-Malik equation is proposed. The 2-D semi-discrete version of Perona-Malik equation is:

$$\begin{aligned} \dot{u}_{ij} = \frac{1}{m_{ij}}(&F(u_{(i+1)j} - u_{ij}) - F(u_{ij} - u_{(i-1)j}) \\ &+F(u_{i(j+1)} - u_{ij}) - F(u_{ij} - u_{i(j-1)})), \end{aligned} \tag{2.9}$$

where $i, j = 1, 2, ..., N$, and $u_{0,j} = u_{1,j}, u_{N+1,j} = u_{N,j}, u_{i,0} = u_{i,1}$ and $u_{i,N+1} = u_{i,N}$ for the absence of neighbors.

2.1.3 Stabilized Inverse Diffusion Equations

The parameter of the Perona-Malik scale-space is K. If K is much larger than the dynamic range of an image, the results of Perona-Malik filtering will be similar to the linear averaging filter. The smaller K will preserve more edges across scale. When $K = 0$, a

discontinuous force function F is obtained. Such a system is called *stabilized inverse diffusion equation* (SIDE) (Pollak *et al.* 1997). It has been proved that the SIDE filter is able to enhance the features and obtain the better performance than other non-linear multi-scale filters (Pollak *et al.* 2000).

The detailed mathematical treatment of the SIDE is given by (Pollak *et al.* 2000). Here the basic concept is reviewed. An image of $M \times N$ pixels is denoted as u_{ij} with $i \in [1,\ldots,M]$ and $j \in [1,\ldots,N]$. The pixel subscripts are conveniently dropped when refer to the entire image. If u^0 is an original image, then $u^1, u^2, \ldots, u^k$ are the successively coarser images after a series of iterative filters $A^1, A^2, \ldots, A^k$ have been applied to u^0. In this iterative filtering, the iteration number k is referred to as the scale and the succession of filtered images is part of a k-dimensional scale-space. The algorithm is based on the semi-discrete SIDE, which is continuous with respect to the temporal evolution in the k-dimensional scale-space but discrete with respect to spatial scales.

The PDE has a simple mechanical analog, where particles with mass m_{ij} are coupled to their four neighbors by a force $F(u)$. The implementation of the SIDE begins with the 2D PDE as

$$\dot{u}_{ij} = \frac{1}{m_{ij}} \Big(F(u_{i+1,j} - u_{ij}) - F(u_{ij} - u_{i-1,j}) + F(u_{i,j+1} - u_{ij}) - F(u_{ij} - u_{i,j-1}) \Big) \quad , \tag{2.10}$$

where $\dot{u} = (u^{k+1} - u^k)/\Delta t$ is the temporal derivative representing the evolution in the temporal scale space. The boundary conditions for peripheral particles are satisfied if they remain stationary. The case of a linear force, e.g., Hooke's law, results in a heat-equation-like behavior.

In this implementation, the force equation is

$$F(u) = \text{sign}(u) - \frac{u}{L} \quad , \tag{2.11}$$

where $\text{sign}(u) = +1$ if $u \leq 0$, and -1 otherwise. This equation was proposed by Pollak *et al.* (1997). In order to satisfy the conditions of a force function, L has to be larger than the dynamic range of an image. Initially, each pixel is considered as a unique region. Equation 2.10 is applied with a force function given by Equation 2.11. Once some neighboring pixels become identical, they will remain identical throughout the iteration. Neighboring pixels are merged together, and progressively larger areas with identical values generate coarser approximations of the original image. This recursive region merging leads to a modification of Equation 2.10 to account for the growing blobs of pixels with identical values

$$\dot{u}_{n_i} = \frac{1}{m_{n_i}} \sum_{n_j \in A_{n_i}} F(u_{n_j} - u_{n_i})\, p_{ij} \quad , \tag{2.12}$$

where m_{n_i} is the number of pixels in region n_i, A_{n_i} is a set of indices of all neighbors of n_i, and p_{ij} denote the force connecting regions n_i and n_j. Once the region merging is concluded, the next evolution starts again with Equation 2.12. The previous steps are repeated until the quiet Sun background is sufficiently suppressed.

2.2 Image Segmentation

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in image processing. The simplest segmentation method is thresholding method. A familiar example of segmentation is when a pixel whose value is less than the threshold is set to be background and others is set to be foreground. In the solar feature segmentation, region growing, edge detection and automatic thresholding methods are combined to produce robust and accurate results.

2.2.1 Region Based and Edge Based Segmentation

Region growing is a popular method used for image segmentation. This method is to merge pixels or pixel-groups into larger regions based on predefined criteria. The basic approach

is to start with an initial set of small areas are iteratively merged according to similarity constraints. Region is grown from the seed pixel by adding in neighboring pixels that are similar, increasing the size of the region. When the growth of one region stops, simply choose another seed pixel which does not yet belong to any region and start again. This whole process is continued until no pixels can make the further merging.

Edge detection is the most common approach for detecting meaningful discontinuities in gray level. A point is defined in an image as being an edge point if its two-dimensional first-order derivative is greater than a specified threshold. First-order derivatives of a digital image are based on various approximations of the 2-D gradient (Gonzalez and Woods 2002). The gradient of an image $f(x,y)$ is defined as a vector

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{2.13}$$

where G_x and G_y denote the gradients in x and y coordinates, respectively. Computation of the gradient is based on obtaining the partial derivatives at every pixel location. The Roberts, Prewitt and Sobel edge operators are widely used first-order derivative operators.

Edge operation can also be applied by the second-order derivative operators. The second-order derivative is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{2.14}$$

The Laplacian edge operator is a second-order derivative operator. As a second-order derivative, the Laplacian typically is sensitive to noise. The magnitude of the Laplacian produces double edges because it complicates segmentation.

2.2.2 Automatic Thresholding

Otsu(1979) proposed an algorithm for automatic threshold selection from a histogram of image. This algorithm minimizes the classification errors to yield the suitable thresholds. Generally speaking, the ideal of Otsu thresholding algorithm is to separates the gray-level values of the image into two clusters according to between variances and within variances. The Otsu thresholding is not supposed to return the minimum between two peaks in a histogram. The Otsu method is to divide the histogram into two clusters such that between-class variance is maximized.

Specifically, let us suppose that the pixels were dichotomized into two classes C0 and C1, which denote pixels with levels [0, t-1] and [t, 255], respectively. And then we can define the within-class variance σ_W^2 as the weighted average of the variances of each cluster

$$\sigma_W^2 = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{2.15}$$

where $q_1(t) = \sum_{i=0}^{t-1} p(i)$ is the probability of the first cluster occurrence, $q_2(t) = \sum_{i=t}^{N-1} p(i)$ is the probability of the second cluster occurrence, $\sigma_1^2(t)$ is the variance of the first cluster, and $\sigma_2^2(t)$ is the variance of the second cluster. If subtract the within-class variance from the total variance of the population, the between-class variance is obtained as follows:

$$\sigma_B^2 = \sigma^2 - \sigma_W^2(t) = q_1(t)[\mu_1(t) - \mu]^2 + q_2(t)[\mu_2(t) - \mu]^2 = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2 \tag{2.16}$$

where σ^2 is the total variance of the population, μ_1 is the mean of the first cluster,μ_2 is the mean of the second cluster, and μ is the mean of the population.

The generic procedure is to separate the pixels into two clusters, find their mean value and between-class variance and within-class variance. The optimal threshold is the one that is able to maximize the between-class variance or minimize the within-class variance (Otsu 1979).

2.3 Morphological Processing

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

2.3.1 Basic Morphological Methods

Mathematical morphology, which is based on set-theoretical concept, can extract object features by choosing a suitable structuring shape as a probe. Two basic morphological operators are dilation and erosion. A binary image can be represented as a set, where "1" denotes the object pixel and "0" denotes the background pixel. Let A and B respectively denote the image and structuring element sets in Euclidean space. The reflection of set B, denoted $\hat{B}$, is defined as

$$\hat{B} = \{w | w = -b, b \in B\} \tag{2.17}$$

Both A and B as sets in X^2,the dilation of A by B, denoted by $A \oplus B$, is defined as

$$A \oplus B = \{x | (\hat{B})_x \cap A \neq \emptyset\} \tag{2.18}$$

The erosion of A by B, denoted by $A \ominus B$, is defined as

$$A \ominus B = \{x | (B)_x \subseteq A\} \tag{2.19}$$

Dilation expands an image and erosion shrinks it. There are two other important morphological operators: opening and closing. Opening generally smooths the contour of an object, breaks narrow isthmuses and eliminates the protrusions. 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 (Shih and Mitchell 1989).

The opening of A by B, denoted by $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B \tag{2.20}$$

Thus, the opening of A by B is the erosion of A by B, followed by the dilation of the result by B. Similarly, the closing of A by B, denoted by $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B \tag{2.21}$$

The closing of A by B is simply the dilation of A by B, followed by the erosion of the result by B.

2.3.2 Directional Morphological Closing

Shih and Kowalski (2003) introduced the eight directional linear 11×11 structuring elements with 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5° slopes, respectively (Shih and Gaddipati 2003). The closing of A by B denoted by $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B \tag{2.22}$$

where $\ominus$ denotes the morphological erosion and $\oplus$ denotes the morphological dilation.

2.3.3 Morphological Thinning and Pruning

The morphological thinning is defined as:

$$A \otimes B = A - (A \circledast B) \tag{2.23}$$

where $A \circledast B$ denotes the morphological hit-and-miss transformation of A by B (Gonzalez and Woods 2002). For thinning A symmetrically, use a sequence of structuring elements:

$B = B^1, B^2, B^3, ..., B^n$, where B^i is a rotated version of B^{i-1} which is shown in Figure 2.2. The entire process is repeated until no further changes occur.

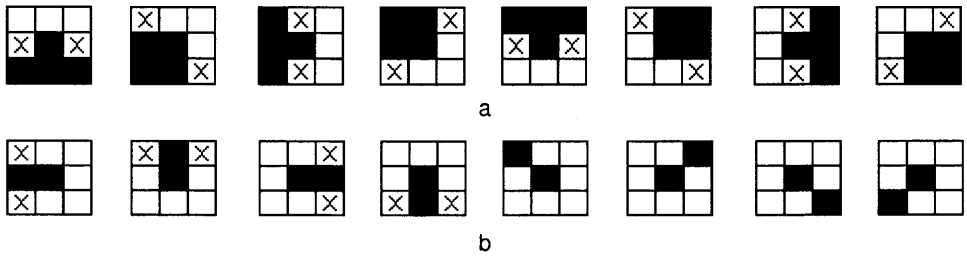

Figure 2.2 a. Structuring elements B for thinning; b. Structuring elements B for pruning.

There are four steps in the morphological pruning which uses four steps to yield X_1, X_2, X_3 and final pruning result X_4 by Equation 2.24, 2.25, 2.26 and 2.27 respectively (Gonzalez and Woods 2002).

$$X_1 = A \otimes \{B\} \tag{2.24}$$

where A is an input set and B is pruning structuring element sequence shown in Figure 2.2.

$$X_2 = \cup(X_1 \circledast B) \tag{2.25}$$

$$X_3 = (X_2 \oplus H) \cap A \tag{2.26}$$

where H is a 3 × 3 structuring element of 1's.

$$X_4 = X_1 \cup X_3 \tag{2.27}$$

The pruning method is used to remove the fix length of a parasitic component.

CHAPTER 3

PATTERN RECOGNITION TECHNIQUES

3.1 Neural Networks

A Neural Network (NN) is a massively parallel distributed processing paradigm that is able to store experiential knowledge and make decision on classification (Haykin 1994). Neural networks, inspired by the nervous systems to learn meaning from complicated or imprecise data, can be used to recognize patterns and detect trends (Bishop 1995).

3.1.1 Multi-Layer Perceptron Neural Network

Multi-layer perceptron (MLP) is a type of feed-forward networks. The learning rule is back-propagation (Ripley 1996). The feed-forward network has three layers: input layer, hidden layer, and output layer. The hidden layer is between the input and the output layers. For example, a simple feed-forward network is shown in Figure 3.1, where there are two nodes (neurons) in the input layer, two nodes in the hidden layer, and one node in the output layer. Weights are incorporated into the connections from input nodes to hidden nodes and from hidden nodes to the output node.

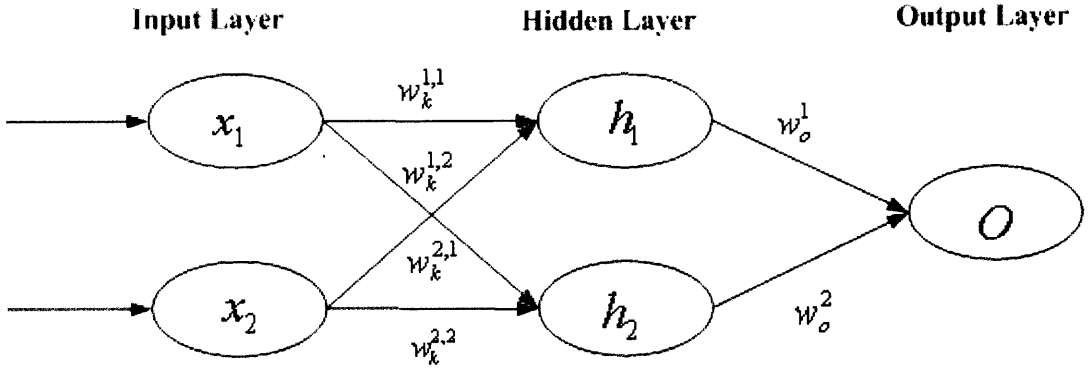

Figure 3.1 The architecture of MLP.

The behavior of an NN depends on both the weights and the input-output function

that is specified for the neurons. There are three types of the activation function: linear, threshold and sigmoid. In this work, the sigmoid function is adopted as the activation function

$$h_j = \text{sigmoid}(\sum_{i=1} w_k^{i,j} x_i),$$
$$O = \text{sigmoid}(\sum_{j=1} w_o^j h_j) \tag{3.1}$$

where x_i is the input feature at the input node, $w_k^{i,j}$ is the weight connecting the input node with the hidden node, w_o^j is the weight connecting the jth hidden node with the output node, and O is the output value. The sigmoid is defined as

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{3.2}$$

.

To use an MLP as a pattern classifier, the training of the correct weights are needed. MLP networks can be trained by a method as iterative weight adjustments: back-propagation training algorithm. Given a sample feature vectors to the initial stage of the network and a set of desired output identifier (target) t. We can compute the Mean Square Error (MSE) between the actual output o and the desired output for the given input in the training set. The error function E can be obtained by

$$E = (t_j - o_j)^2 \tag{3.3}$$

To reduce the MSE, it is necessary to calculate the gradient of the error function with respect to each weight. One may then move each weight slightly in the opposite direction to the gradient. The gradient function for the weights in the output layer is shown below.

$$\delta_j = \frac{\partial E}{\partial w_o^j} = \frac{\partial (t-o)^2}{\partial w_o^j} \tag{3.4}$$

The new values for the network weights from output units to hidden units are calculated by multiplying the negative gradient with a step size of parameter ξ' (called the learning rate) which is shown in Equation 3.5. The ideal learning rate is to use relatively high values to start with and then reduce them as the training progresses (Zaknich 2003).

$$w_o^{j+1} = w_o^j - \xi' \delta_j. \tag{3.5}$$

The weights in the hidden layer are updated using the same procedure. After all the correct weights are calculated, the neural network is completely constructed.

3.1.2 Radial Basis Function Neural Network

The basic idea of Radial basis function (RBF) is covered with overlapping circular regions in a high-dimensional space. From this viewpoint, learning is accomplished by finding a surface in a multi-dimensional space. This surface is used to interpolate the test data. RBF network is a fully connected network and generally is used as a classification tool (Broomhead and Lowe 1988). RBF are becoming popular because of their quick training and elegance. They originated as potential functions in Bashkirov et al. (1964) and Sprecht (1968).

In a RBF model, the layer from input nodes to hidden neurons is unsupervised and the layer from hidden neurons to output nodes is supervised (Bishop 1995). The transformation from the input to the hidden space is nonlinear, and the transformation from the hidden to the output space is linear. The hidden neurons provide a set of "functions" that constitute an arbitrary "basis" for the input patterns. These are the functions known as called radial basis functions. Through careful design, it is possible to reduce a pattern in a high-dimensional space at input units to a low-dimensional space at hidden units.

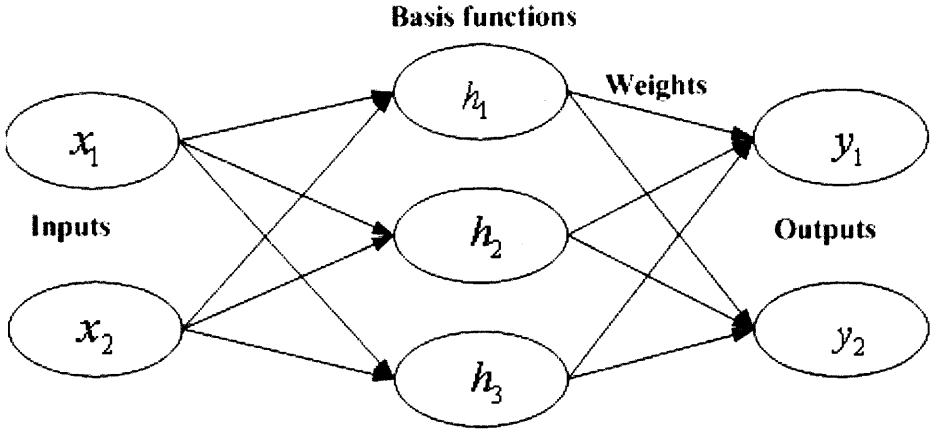

Figure 3.2 The architecture of a radial basis function neural network. The input vector has d nodes and the outputs vector has c nodes. It is a mapping from $R^d \rightarrow R^c$.

In Figure 3.2, the network forms the following equation:

$$y_k(x) = \sum w_{kj} \cdot h_j(x) + w_{k0} \tag{3.6}$$

where $h_j(x)$ is a Gaussian function typically, w_{kj} is the weight from hidden units to output units, w_{k0} is the bias or threshold. The Gaussian basis function is used as an activation function. That is

$$h_j(x) = \exp(-\|x - \mu_j\|^2 / 2\sigma_j^2) \tag{3.7}$$

where x is the d-dimensional input vector with element x_i, μ_j and σ_j are respectively the center and the standard deviation of the Gaussian basis function. Since the first and second layers of RBF network are unsupervised and supervised respectively, a two-stage training procedure is used for training the RBF model. In the first stage, the input data set is used to obtain the parameters of the activation functions (like μ_j and σ_j). In the second stage, the optimal weights between hidden neurons and output nodes are obtained by minimizing a sum-of-square error function. The following procedures describe the steps of obtaining the optimal weights.

By absorbing the bias parameter, w_{k0} , into the weights, Equation 3.6 can be revised

as

$$h_j(x) = \exp(-\|x-\mu_j\|^2/2\sigma_j^2) \tag{3.8}$$

where $h_j(x)$ is an extra basis function with the activation value 1. Equation 3.8 can be rewritten using the matrix notation as

$$Y(x) = W\Phi \tag{3.9}$$

where $W = [w_{kj}]$ and $\Phi = [h_j(x)]$.

The sum-of-square error function, E, can be described as

$$E = \frac{1}{2}\sum\sum[y_k(x^n) - t_k^n]^2 \tag{3.10}$$

where x^n is the input data set, and t_k^n is the target value for the output unit k.

By differentiating E with respect to w_{kj} and setting derivative to zero, the optimal weights can be obtained. The solutions of weights can be expressed using the matrix notation as

$$(\Phi^T\Phi)W^T = \Phi^T T \tag{3.11}$$

where $T = [t_k^n]$ and $\Phi = [h_j(x^n)]$.

By multiplying By multiplying $(\Phi^T\Phi)^{-1}$ to Equation 3.11, the solution for the weights is given as

$$W^T = (\Phi^T\Phi)^{-1}\Phi^T T \tag{3.12}$$

$$W^T = \Phi^+ T \tag{3.13}$$

where $\Phi^+ = (\Phi^T\Phi)^{-1}\Phi^T$ being the pseudo-inverse of Φ. After computing the optimal weights, the RBF network can be used as a classifier to segment the test data into the corresponding classes.

RBF is strongly dependent on the quality of the employed learning strategy and the quantity of training images. The aim of an adaptive learning RBF network is to reduce the required knowledge of the system parameters with a minimum amount of performance loss. The RBF network requires knowledge of three different parameters per neuron: the center vector μ_j, the weights w_{kj} and the radius σ_j.

One unsupervised learning strategy is the self-organizing feature map. When the algorithm has converged, prototype vectors which corresponding to nearby points on the feature map grid have nearby locations in input space. However, the imposition of the topographic property, particularly if the data is not intrinsically two-dimensional, may lead to a sub-optimal placement of vectors. Here the K-Means unsupervised learning strategy is used as follows: let μ_j be the mean of the data points in set S_j given by

$$\mu_j = \frac{1}{N_j} \sum_{n \in S_j} x^n \tag{3.14}$$

The initial centers are randomly chosen from the data points and the nearest μ_j is updated using

$$\Delta\mu_j = \eta(x^n - \mu_j) \tag{3.15}$$

where η is the learning rate parameter.

The second parameter of the RBF network is the weight w_k of the output layer. It can be done by using the Least Mean Square (LMS) algorithm. The LMS algorithm was originally developed by Widrow and Hoff in 1960 (Widrow and Hoff 1960) and is also known as the Widrow-Hoff rule (Mitra and Poor 1994). The weights can be summarized as follows

$$\Delta W = 2\eta(y_k(x^n) - t_k^n)h(x^n) \tag{3.16}$$

where η is the learning rate of LMS, and $y_k(x^n)$ and t_k^n are the responses of the RBF network and the desired response. The vector $h(x^n)$ contains the unweighted responses of all

neurons in the hidden layer.

The last parameter of the RBF network is the radius or spread of the radial function. We may use an average of the center spread of all RBFs to calculate the radius. After the centers μ_j are established, σ^2 can be derived from the center as

$$\sigma^2 = \frac{1}{M}\sum_{j=1}^{M} \|y_k(x^n) - \mu_j\|^2 \tag{3.17}$$

where M is the number of hidden nodes.

3.2 Support Vector Machines

Support Vector Machine (SVM) is a generation learning system based on advances in statistical learning theory (Vapnik and Chervonenkis 1991). It is a powerful tool for data classification. SVM Classification can be achieved by a linear or nonlinear separating surface in the input space of the dataset. SVM has been successfully applied in text categorization, hand-written character recognition, image classification, biosequences analysis, etc.

Before go to the detail of SVM, it is necessary to introduce how the learning machine such as NNs and SVMs works. Suppose l observations are given, and each observation consists of a pair: a vector $x_i \in R^n$, $i = 1,...,l$ and the corresponding "target" y_i. y_i can be obtained by existed knowledge base. In the hypothesis space, there is a density distribution $p(x,y)$ which derived from these data, for example, the data are assumed independently put and identically distributed. The learning machine is to discover the mapping $x_i \rightarrow y_i$ by utilizing a set of possible mappings $x_i \rightarrow f(x,\alpha)$. α is the adjustable parameter in the function $f(x,\alpha)$. In a fixed structure MLP neural network, the α can be treated as the weights and biases. $f(x,\alpha)$ function could represent a set of Radial Basis functions or MLPs with a certain number of hidden neurons (Vapnik 1982). The expected risk for a

learning machine can be obtained by Equation 3.18

$$R(\alpha) = \int \frac{1}{2}|y - f(x,\alpha)|p(x,y)dxdy. \tag{3.18}$$

The empirical risk $R_{\text{emp}}(\alpha)$ is defined in Equation 3.19. It is the mean error based on learning of l observations.

$$R_{\text{emp}}(\alpha) = \frac{1}{2l}\sum_{i=1}^{l}|y_i - f(x_i,\alpha)|. \tag{3.19}$$

The Empirical Risk Minimization (ERM) inductive principle is to minimize the risk functional on the basis of empirical data (x_i , y_i). The ERM principle is based on the law of large numbers converges in probability to the expected risk (strategy followed by MLP and RBF neural networks). By choosing some η such that $0 \leq \eta \leq 1$, the following bound holds (Vapnik 1998).

$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{(\frac{d(\log(2l/d)+1) - \log(\eta/4)}{l})} \tag{3.20}$$

where d is called the Vapnik Chervonenkis (VC) dimension. The right hand side of Equation 3.20 is the "bound on the risk", and the second addition term is called the "Confidence interval". When l/d is large, the Confidence interval is small. When the Confidence interval is close to zero, the total risk is then close to the value of the empirical risk. When l/d is small, a small $R_{\text{emp}}(\alpha)$ does not guarantee a small value of the actual risk. In this case, to minimize the total risk has to consider the confidence interval and d. To minimize the right-hand side of the bound risk in Equation 3.20, VC dimension is considered to be a controlling variable.

In the previous section, classical neural network is considered, which implement the first strategy: Keep the confidence interval fixed and minimize the empirical risk. Below a new type of universal learning machine, the Support Vector Machine, implements

the second strategy: Keep the value of the empirical risk fixed and minimize the confidence interval (Vapnik 1995). SVM is based on the structural risk minimization (SRM) inductive principle. The SRM principle considers and minimizes the actual risk with respect to the both terms, the empirical risks, and the confidence interval (Vapnik and Chervonenkis 1974). Therefore, SVM is a better strategy than classical neural network such as MLP neural network. The basic idea of a support vector machine is to separate the fixed given input pattern vectors into two classes using a hyperplane in the high dimensional space (Vapnik and Chervonenkis 1991). This means we always increment the VC-dimension d by one if we fail to separate the input pattern vectors. The process is not stopped until the separation is satisfied. SVM intends to separate the input pattern using the minimum VC-dimension d, which minimizes the total risk by considering both empirical risk and confidence interval. Figure 3.3 shows the SVM with a linear hyperplane.

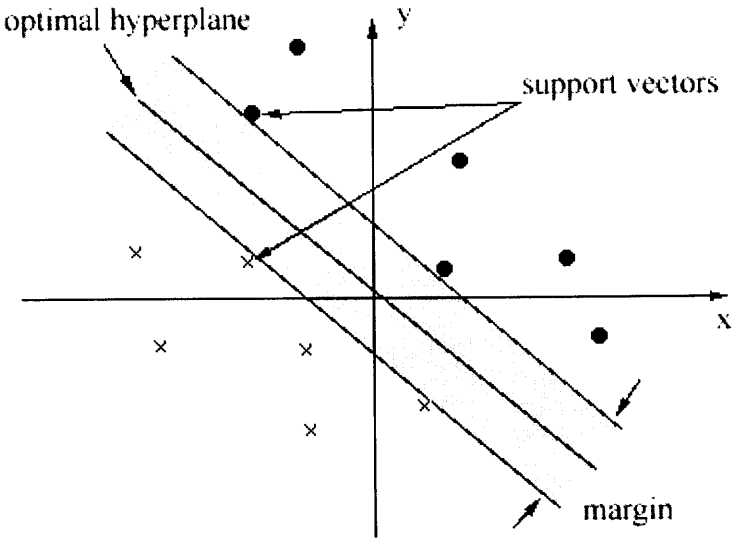

Figure 3.3 Support vector machine classification with a linear hyperplane that maximizes the separating margin between the two classes.

Consider a two-class classification problem where patterns are represented as an N-dimensional vector x. For each corresponding vector, there exists a value $y \in \{-1, +1\}$.

$$x = \{x_1, x_2, x_3, ..., x_m\} \in R^N,$$
$$y = \{y_1, y_2, y_3, ..., y_m\} \in \{-1, +1\} \tag{3.21}$$

The problem is to find a decision function $g(x)$ such that the class label y of any example x can be accurately predicated. Using mathematical terms, we have g: $R \rightarrow -1, +1$ (Guyon and Stork 2000). For linear support vector classifier

$$f(x) = (x \cdot w) + b \tag{3.22}$$

where $w \in R^N$ and $b \in R$.

The set of labelled training patterns

$$(y_1, x_1), ..., (y_l, x_l), \quad y_i \in -1, 1 \tag{3.23}$$

It is linearly separable if there exist a vector w and a scalar b such that the following inequalities are valid for all elements of the training set.

$$w \cdot x_i + b \geq 1 \quad \text{if} \quad y_i = 1,$$
$$w \cdot x_i + b \leq -1 \quad \text{if} \quad y_i = -1. \tag{3.24}$$

The optimal hyperplane is the unique one which separates the training data with a maximal margin. The distance $\rho(w, b)$ is given by

$$\rho(w, b) = \min_{\{x:y=1\}} \frac{x \cdot w}{\|w\|} - \max_{\{x:y=-1\}} \frac{x \cdot w}{\|w\|} \tag{3.25}$$

Because the optimal hyperplane (w_0, b_0) is the arguments that maximize the dis-

tance $\rho(w,b)$, we have Equation below.

$$\rho(w,b) = \frac{2}{\|w_0\|} = \frac{2}{\sqrt{w_0 \cdot w_0}} \tag{3.26}$$

It means that the optimal hyperplane is the unique factor that minimizes the actual risk under the constraints 3.24. This problem is usually solved by means of the classical method of Lagrange multipliers. If we denote the N nonnegative Lagrange multipliers as $\alpha = (\alpha_1, \alpha_2, ..., \alpha_N)$, the solution is equivalent to determining the saddle point of the function (Minoux 1986).

$$L(w,b,\alpha) = \frac{1}{2} w \cdot w - \sum_{i=1}^{N} \alpha_i \{y_i (w \cdot x_i + b) - 1\} \tag{3.27}$$

At the saddle point, L has a minimum for $w = w_0$ and $b = b_0$ and a maximum for $\alpha = \alpha^0$. We have

$$\begin{aligned} \frac{\partial L(w_0, b_0, \alpha^0)}{\partial b} &= \sum_{i=1}^{N} y_i \alpha_i^0 = 0, \\ \frac{\partial L(w_0, b_0, \alpha^0)}{\partial b} &= w_0 - \sum_{i=1}^{N} \alpha_i^0 y_i x_i = 0. \end{aligned} \tag{3.28}$$

Substituting Equations 3.28 into the right hand side of Equation 3.27, it is reduced to the maximization of the function

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j). \tag{3.29}$$

From Equation 3.28, it follows that

$$w_0 = \sum_{i=1}^{N} \alpha_i^0 y_i x_i. \tag{3.30}$$

The parameter b_0 can be obtained from the Kuhn-Tucker condition.

$$b_0 = y_j - w_0 \cdot x_j. \tag{3.31}$$

The problem of classifying a new data point x is now simply solved by looking at the sign of $w_0 \cdot x + b_0$.

For the nonlinear case, the kernel functions are used to transfer input data to feature space. By choosing a non-linear mapping, the SVM constructs an optimal separating hyperplane in the higher dimensional space. The idea exploits the method of Aizerman et al. (1964) which, enables the curse of dimensionality to be addressed. Most commonly employed functions are acceptable to be the kernel function. Among acceptable mappings are polynomials, radial basis functions and certain sigmoid function. For example, Polynomial support vector classifier

$$k(x,x') = (x,x')^d. \tag{3.32}$$

and Gaussian RBF kernel classifier

$$k(x,x') = \exp(-\|x-x'\|^2/2\sigma_j^2). \tag{3.33}$$

The optimization problem of Equation 3.29 becomes

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \tag{3.34}$$

where $K(x_i \cdot x_j)$ is the kernel function performing the non-linear mapping into feature space and the constraints are unchanged.

CHAPTER 4

AUTOMATIC SOLAR FLARE DETECTION

4.1 Introduction

Measurement of the evolution properties of solar flares through their complete cyclic development is crucial in the studies of Solar Physics. From the analysis of solar Hα images, the Support Vector Machine (SVM) is used to automatically detect flares and apply image segmentation techniques to compute the properties of solar flares. The application also automatically tracks the apparent separation motion of two-ribbon flares and measure their moving direction and speed in the magnetic fields. From these measurements, with certain assumptions, we can further infer the reconnection of the electric field as a measure of the rate of the magnetic reconnection in the corona. The measured properties of the flare evolution can be compared with associated events such as coronal mass ejections (CMEs)(Yurchyshyn *et al.* 2000). Statistical properties of flares can only be derived by the study of a large number of events. This can only be achieved by an automatic computer program.

Automatic solar flare detection plays a key role in real-time space weather monitoring. The application uses a set of nine features for solar images and used RBF and SVM in addition to MLP to perform classification. A key pixel is detected on a flare which is the one with the maximum gray level difference between current and previous images. A solar image may have several key pixels when the image contains several flares. The classification rate of the method is more than 95% and the running time is less than 0.3 second. The detection also provides the start time, end time and position of a flare. The diagram of the automatic solar flare detection is shown in Figure 4.1.

The center of a solar flare is considered to be located at the key pixel. After detecting the position of a flare, the following properties are measured: center position (x,y) of a

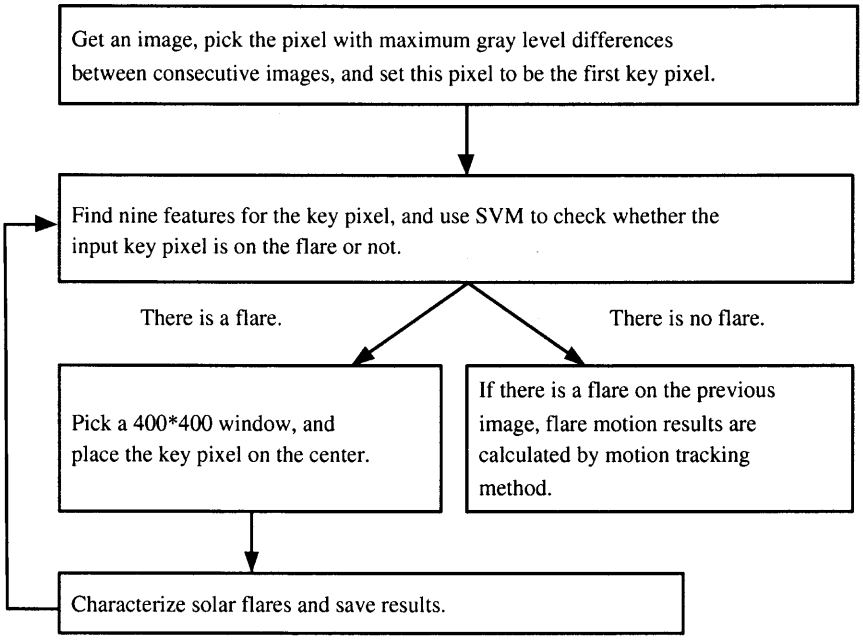

Figure 4.1 Automatic procedure to detect and characterize flares.

flare on the full-disk Hα images, the 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. The first step of automatic characterization is preprocessing; the second step is to segment the solar image to obtain structure properties of a flare; the third step is to track the motion. 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. Each pixel on the current model is correlated to a pixel on the previous model, and track the motion orientation and speed for each pixel pair. The maximum number of pixels along a motion orientation and the speed in this direction is computed. The process is shown in Figure 4.1.

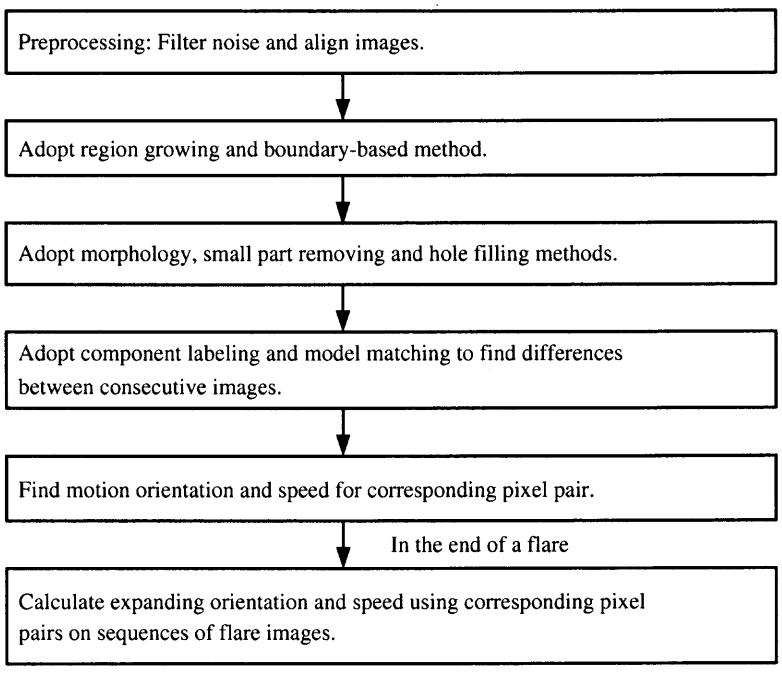

Figure 4.2 Automatic procedure to compute flare region and flare motion.

4.2 Automatic Solar Flare Detection

4.2.1 Previous Work

Veronig et al. (2000) proposed a method for automatic flare detection by applying a combination of region-based and edge-based segmentation methods. Simple region-based methods are applied for a tentative assignment of flare activity, making use of one of the decisive flare characteristics: the high intensities. Borda et al. (2001) presented a method for automatic detection of solar flares using the neural network technique. The network used is multi-layer perceptron (MLP) with back-propagation training rule. They used a supervised learning technique and required a lot of iterations to train the network.

4.2.2 Feature Analysis and Preprocessing

Borda et al. (2001) used seven features for solar flare detection. After extensive investigations, our application uses nine features as described below.

Feature 1: Mean brightness of the frame. Usually, when flares happen, the image

becomes brighter. Let x_j denote the gray level of a pixel and N denote the number of pixels. The mean brightness $\hat{x}$ is represented as

$$\hat{x} = \sum_{j=0}^{N-1} x_j \tag{4.1}$$

Feature 2: Standard deviation of brightness. Usually, when flares happen, the standard deviation of the brightness becomes large.

$$\begin{aligned} \sigma &= \sqrt{V}, \\ V &= \sum_{j=0}^{N-1} (x_j - \hat{x})^2 \end{aligned} \tag{4.2}$$

where *std* denotes the standard deviation, and V denotes the variance.

Feature 3: Variation of mean brightness between consecutive images. This is the difference of mean values between the current image and the previous image.

Feature 4: Absolute brightness of a key pixel. By comparing consecutive images, the key pixel that has the maximum gray level difference is found. Usually, when flares happen, the absolute brightness of the key pixel becomes large. The position of the key pixel k is represented as

$$k = \text{where}(\max(G_{pre} - G_{current})) \tag{4.3}$$

where G_{pre} denotes the previous image, and $G_{current}$ denotes the current image.

Feature 5: Radial position of the key pixel.

Features 1-5 focus on the global view of the consecutive images. But these features are insufficient to obtain the information for small solar flares. Therefore, it is necessary to include the local view of the solar images.

Feature 6: Contrast between the key pixel and the minimum value of its neighbors in a 7 by 7 window. When flares occur, this contrast becomes large.

Feature 7: Mean brightness of a 50 by 50 window, where the key pixel is on the center. Since we assume that the key pixel is one of the pixels on the flare, the 50 by 50 window will include most of the flare. This region is the most effective for the solar flare detection.

Feature 8: Standard deviation of the pixels in the aforementioned 50 by 50 window.

Feature 9: Difference of the mean brightness of the 50 by 50 window between the current and the previous images.

4.2.3 Results

Data-set and Design

The solar Hα images of 2032×2032 pixels were obtained from BBSO. The images of flare events starting from January 1, 2002 to December 31, 2002 are selected. The first step taken was to carefully divide the images into two classes: flare state and non-flare state. All the flare events that can be recognized by human eyes are covered. Corresponding images are labelled as "flare state" and included the same number of "non-flare state" images. Because some micro-flares are difficult to be recognized even by human eyes, they are regarded as "non-flare state". Secondly, the application computed the nine features for a given solar image and put it into either the training or the testing data set. The application used the training data set to train the networks and used the testing data set to obtain the classification rate. Finally, the application obtained flare properties using image segmentation techniques, since we are also interested in obtaining the properties of flares such as size, lifetime, and motion.

The programs are developed in Interactive Data Language (IDL) by Research Systems, Inc. The program runs on a DELL Dimension L733r with 733 MHz CPU and memory of 256 Mbytes under Windows 2000.

For the experiments of solar flare detection, we capture one image per minute and

process the current image with comparisons of previous image to obtain the nine features. First, the application filters the noise of Full-disk images using a Gaussian function and center these two images using the IDL FIT_LIMB function. Second, the application computes the mean brightness and standard deviation for the two images. Third, application divides them to obtain the pixel with the maximum gray level difference. It is regarded as the key pixel to compute other features. After preprocessing steps, the program calculate the result of flare or non-flare using the neural network and SVM classifiers with the nine features of the key pixel.

Classification Rate

The MLP, RBF and SVM methods are tested based on the 240 events including pre-flare, main phase and post-flare and half of them are used as the training set and half of them are used as the testing set. For MLP, after about 1000 iterations for the training set of 120 events, the application tests this network with 120 events. The fraction of misclassified events is about 5%. MLP uses supervised training and needs a lot of iterations to obtain the satisfied classification rate. For RBF, the application obtains a better classification rate and faster training speed. For SVM, the best performance is obtained among the three methods, such that the misclassification rate is 3.3%. The results are shown in Table 4.1.

The MLP architecture is to perform a full non-linear optimization of the network. Therefore, a lot of iterations to train the system to get the best result are needed. In Table 4.2, it is demonstrated that the result performs better with more iterations. The number of hidden nodes determines the complexities of the neural networks. We can adjust the

Table 4.1: Classification report of solar flares on 120 training events and 120 testing events.

Methods based on events	Classification rate	Training time	Testing time
MLP with 1000 iterations	94.2%	60.20 secs	0.01 sec
RBF	95%	0.27 sec	0.01 sec
SVM	96.7%	0.16 sec	0.03 sec

number of hidden nodes to obtain the best performance. In Table 4.2, the comparison of MLP with 11 hidden nodes and MLP with other number of hidden nodes is shown. Test correctness of MLP with 11 hidden nodes is comparable to MLP with more hidden nodes. MLP with 11 hidden nodes has better performance than MLP with more or less hidden nodes.

RBF neural network, as compared with the MLP, is the possibility of choosing suitable parameters for each hidden unit without having to perform a full non-linear optimization of the network (Bishop 1995). Usually, the number of hidden nodes equals the number of classes. For flare detection, we want to use RBF classifier to separate two classes. The application tries RBF with different number of hidden nodes and find that RBF with two hidden nodes is good for the pattern classification. The parameters of RBF are center vector, the weight and the radius. The training architecture can be quickly constructed by K-mean algorithm.

SVM is based on the idea of "minimization of the structural risk". To obtain the best classification rate, we consider the confidence interval and empirical risk. Complex system has high confidence interval and low empirical risk. The application tests the flare events using linear SVM and non-linear SVMs respectively with Polynomial kernel classifier and Gaussian RBF kernel classifier. Experimental results are shown in Table 4.3.

Table 4.3 shows that linear SVM is better than non-linear kernel SVM for the events. Non-linear SVM can reduce the empirical risk using complex systems, but it has

Table 4.2: Comparison of different MLP iterations and hidden nodes on 120 training events and 120 testing events.

Methods based on events	Classification rate	Training time
MLP with 1000 iterations and 11 hidden nodes	94.2%	60.20 secs
MLP with 100 iterations and 11 hidden nodes	69.2%	6.2 sec
MLP with 3000 iterations and 11 hidden nodes	94.2%	179.3 secs
MLP with 1000 iterations and 11 hidden nodes	94.2%	60.2 secs
MLP with 1000 iterations and 6 hidden nodes	92.5%	46.7 secs
MLP with 1000 iterations and 20 hidden nodes	94.2%	115.6 secs

higher confidence interval. Using linear SVM with the nine features of the solar flare, we have both low empirical risk and confidence interval to achieve the best total risk. Therefore we prefer linear SVM to classify the flare patterns.

4.3 Automatic Solar Flare Characterization

4.3.1 Previous Work

In the previous work the motion of two-ribbon flares is not measured automatically (Qiu *et al.* 2004). First researchers outlined manually the trajectories of the flare ribbon fronts. Then the velocity of the ribbon expansion is determined as the time derivative of the distance between the centroids of successive trajectories. They also take a running mean over several time frames. The uncertainties generated by this method are discussed by Qiu et al. (2004). A major disadvantage is that the method is not entirely automatic.

4.3.2 Preprocesssing

In the preprocessing step image enhancement and filtering techniques are used to obtain high quality images. A median filter is used to remove additive noise and a recursive soft morphological filter is applied to be less sensitive to additive noise and small variations (Shih and Puttagunta 1995). The application detects the center of the solar disk in each image using an IDL program, called fit limb, to align images using their centers. A solar flare is only a small feature on the full-disk solar image. In this step a 400×400 window is picked centered at the key pixel of a solar flare.

Table 4.3: Comparison of different SVM training strategies on 120 training events and 120 testing events.

Methods based on events	Classification rate	Training time	Testing time
Linear SVM	96.7%	0.16 secs	0.03 sec
SVM with Polynomial kernel	95%	0.34 sec	0.03 sec
SVM with RBF kernel	90.83%	0.44 sec	0.03 sec

4.3.3 Region Growing and Adaptive Boundary-based Method

Both region-based and adaptive boundary-based methods (Castleman 1996), in addition to morphological image processing techniques and hole filling techniques, are combined to obtain the accurate boundary and the structure of flares. The application use region growing to include the bright part of a flare since the biggest part of a flare is the highly brightening area on the sun. The maximum gray level pixel is selected in a 400×400 image as the seed. The growing pixels have to be 8-connected to one of the pixels in seed regions and the neighbors' gray level must be greater than 80 percent of the seed. When a pixel satisfies the criteria, it is included in the seed regions. The result of region growing is shown in Figure 4.3.

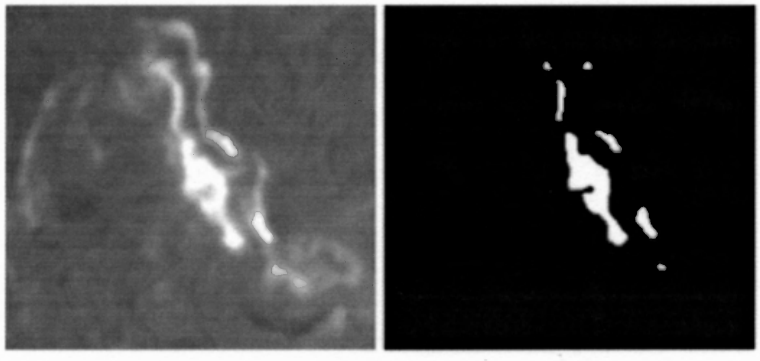

Figure 4.3 A flare peaked at 21:40:21 on 21 May 2002 on the BBSO Hα image. Left: the original image; right: the result of region growing.

The region-based method may lose detail near the edges of an object since it is hard to decide the threshold for region growing. Thus boundary-based methods such as the Sobel edge detector (Jähne 1997) are adopted to enhance the boundary of a solar flare. In our experiment the first-order and second-order derivatives have been used for detecting the boundary of flares. The first-order derivatives are computed using gradient, and the second-order derivatives are obtained using Laplacian.

Considering the first-order and second derivatives are defined as

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{4.4}$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{4.5}$$

The edge of an object can be split into two sides: dark and bright. When a pixel is on the edge, we have $|\nabla \mathbf{f}| \geq T$, where T is a threshold; in addition, when a pixel is on the dark side, we have $\nabla^2 f \geq 0$, and on the bright side, we have $\nabla^2 f < 0$.

The application uses 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. The result of region growing is added to the result of the adaptive boundary method to have the best possible result as shown in Figure 4.4.

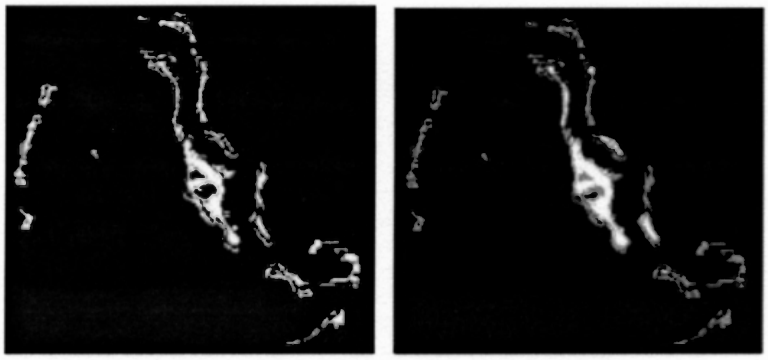

Figure 4.4 Left: the result of the adaptive boundary method; right: the addition result of the adaptive boundary method and the region growing method.

4.3.4 Morphological Processing

The application applies the morphological closing to the image to erase gaps and to smooth contours. The result is shown in Figure 4.5, where a 3×3 structuring element with gray

level of all ones is used.

In order to remove some small regions, each individual component in the binary image is counted. When the number of pixels in a component is less than a threshold, it is removed.

In order to fill the small hole inside the flare, a simple and efficient method is proposed. The segmented flare in the binary image is located at the bright region where the gray level is 1. A dark hole inside a flare is an individual dark region with a limited number of pixels. The pixel number of each individual dark region is checked. When the pixel number is less than the threshold, this dark hole is removed. The results show that a good threshold is 150. Figure 4.5 shows the result of small part removing and hole filling.

Figure 4.5 Left: the result of the morphological closing; right: the result of the small part removing and hole filling.

4.3.5 Component Labeling and Model Matching

There are two major techniques to estimate the motion in the field: differential techniques and matching techniques (Trucco and Verri 1989). The solution is to build a flare model for each image and calculate differences between consecutive images using the previous segmented image results. For the first flare image, the application uses the component labeling technique to set the first three largest components as the main body of a flare and build a model based on the component. Then the application matches the flare on the

current image with the previous model to obtain the flare components for the current image. To ensure the differences between consecutive images are selected, gray level differences on the original image are also taken into account. Thus a pixel pair is removed from the result of differences when the pixel pair difference is very small.

To track the motion automatically, the same object is labelled on consecutive images. For the first image, the smallest distance is obtained between regions. When the smallest distance between two components is less than the threshold, let them merge; otherwise, label them as different objects. Then the first three largest objects are selected to be the flare region.

The separation of the two-ribbon flares can grow larger after certain time. The following model matching method is proposed for matching flares. A model is built for each image and match each object in the current image with the previous model. When an object overlaps with the previous image, this object is included to the current model. The model is created for the flare image as follows:

1. For the first image, a morphological closing is used with a 15×15 structuring element to fill in the inside gap. For the images thereafter, the previous model is used as the reference.

2. Obtain the differences between the segmented result and model result.

3. Remove the outer differences where the outer edge is recovered with the segmented result and keep the inner differences.

Differences ($D1$) are obtained using the current model minus the previous model. The result of component labeling and model matching methods are shown in Figure 4.6.

4.3.6 Difference Tracking and Pixel Corresponding

After having the current model, the previous model, and differences between the current and previous models, the expanding orientation and distance are calculated for each pixel

Figure 4.6 Left: result of component labeling; middle: result of the final model; right: result of differences between the current and previous images.

using the following method:

1. Let A = the edge of the current model, B = the edge of the previous model, and $C = 3A + 2B$. The application obtains the gray level 5 on C, where A and B are overlapped, $C = 3$ where the pixels are on the current model, and $C = 2$ where the pixels are on the previous model. The application tracks the corresponding pixel ($C = 2$) on the previous model for each pixel ($C = 3$) on the current model. For instance, a nearest pixel (x_p,y_p) is detected on the previous model for a pixel (x_c,y_c) on the current model.

2. Obtain the expanding direction for each pair of corresponding pixels by

$$\theta_{c,p} = \arctan\left[\frac{y_c - y_p}{x_c - x_p}\right] \tag{4.6}$$

and the expanding distance by

$$d_{c,p} = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2} \tag{4.7}$$

Let us assign an increasing sequence number i to each flare image. For each flare image, we record the number of pixels n and the mean of the expanding distance D on each flare's motion direction j, where $j = 1$ denotes the angle between $[-90^o, -80^o]$,

$j = 2$ denotes between $[-80^o, -70^o]$, and so on.

$$D_{i,j} = \text{mean}(d_{c,p}) \tag{4.8}$$

where d is the resulting distance, i denotes the sequence number of a flare image, and j denotes the flare motion direction.

3. Calculate the two-ribbon flare motion using the pixels' motions. First estimate the principle direction of the expanding motion by checking the number of pixels in each orientation. Then compute the expanding speed s by

$$s_{i,j} = \frac{D_{i,j} - D_{(i-1),j}}{t} \tag{4.9}$$

where t denotes the time interval of consecutive images.

4.3.7 Results

The programs is developed in Interactive Data Language (IDL) by Research Systems, Inc. The programs run on a DELL Dimension L733r with CPU 733 MHz and memory of 256 Mbytes under Windows 2000. The process of image segmentation takes less than 20 seconds for each image, and the process of motion tracking takes less than 10 seconds. The computational time for the following processing steps is given in Table 4.4: (1) image preprocessing, (2) region growing and adaptive boundary processes, (3) morphological closing process, (4) small part removing and hole filling processes, (5) component labeling and model matching, and (6) pixel corresponding and motion tracking.

We select an Hα two-ribbon flare event observed on May 21, 2002 from BBSO

Table 4.4: Computational time (seconds) for each processing step.

Step	1	2	3	4	5	6
Time	3.9	6.2	0.8	4.7	6.3	1.8

to demonstrate the characterization results. In the following section the results of flare detection and characterization around 21:40:21UT is demonstrated.

Using the SVM classifier, the beginning and the ending time of a flare are obtained. An interval of two minutes is used to track speed. First the application uses component labeling and model matching to set the flare components. After building this model, the application computes outer difference pixels for consecutive models, and only keep difference pixels for each model. The application corresponds each pixel on the current model to each pixel on the previous model for computing the orientation and speed of pixel's motion. The application records the number of pixels and the median speed in each direction range. When a flare is over, the application sums up all the pixels to estimate the maximum number of pixels along a motion orientation. The speed for consecutive images is obtained using the pixel pairs' motion in this orientation. The comparison of the two-ribbon flare's expanding distance between Qiu's method and the new method is shown in Figure 4.7. The results of the two methods are quite similar but the new result is obtained from automatic procedure.

The magnetic reconnection rate is measured in term of electric field E in reconnection current sheet. Methods and uncertainties of these measurements are extensively discussed by Qiu et al. (2004) and Jing et al. (2005). Using these methods, one can compute the expansion velocity V_r of flare ribbons, and then align the ribbon with the magnetic fields B_n they sweep through. The new method detects all the moving pixels and obtain accurate B_n by using the average B of moving pixels. E is obtained by

$$E = V_r \times B_n \tag{4.10}$$

The comparison of the E field between Qiu's method and the new method is shown in Figure 4.7. Compared with the previous method which tracks the moving front of ribbons manually, the new method has the following advantages: 1) it avoids the difficulty

in determining the positions of the moving front which usually do not have a regular and clear shape, and 2) one can obtain an accurate average by considering all the B_n's in the newly brightened area. However because noise affects in the image processing, our results also contain errors. We estimate the confidence interval of both methods by computing the standard deviation of parameters in the background level. The confidence interval are shown in Figure 4 using error bars. In general, the deviations between two sets of results are acceptable. For solar flare study, we are most interested in the total separation distance shown in Figure 4a and the peak value of the electric field shown in Figure 4b. Figure 4b shows that the new method can obtain a better result than the method in Qiu et al. since the peak of the E field from the new result is closer to the peak of the light curve of the flare as observed in the radio.

4.4 Summary

This chapter shows the comparisons of flare detection using the three advanced techniques which are Multi-Layer Perceptron (MLP), Radial Basis Function (RBF) and Support Vector Machine (SVM). Through experiments, it has been proven that SVM is the best for the solar flare detection because it offers the best classification result and the training and testing speed are relatively fast. The second choice is RBF. Because its training data are presented only once, it is the fastest neural network. MLP is not a well-controlled learning machine. The sigmoid function has a scaling factor that affects the quality of the approximation, and the convergence of the gradient-based method is rather slow. The result of MLP is not as good as the RBF and SVM.

After the solar flare detection, the morphological properties of flares are analyzed using region growing and boundary-based methods. Moreover, the application uses 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. A motion tracking method is implemented to compute the orien-

tation and speed of two-ribbon flares automatically. The processes of image segmentation and motion tracking take less than 30 seconds for each image and the results are accurate. Moveover the application obtains the E field of magnetic reconnection current sheet with this automatic program and verify the result of Qiu et al (2004).

Our automatic process is valuable for the forecasting and studies of solar flares since this process dramatically improves efficiency and accuracy. The method allows us to study the evolution of a large number of solar flares efficiently, which will help space weather forecasting.

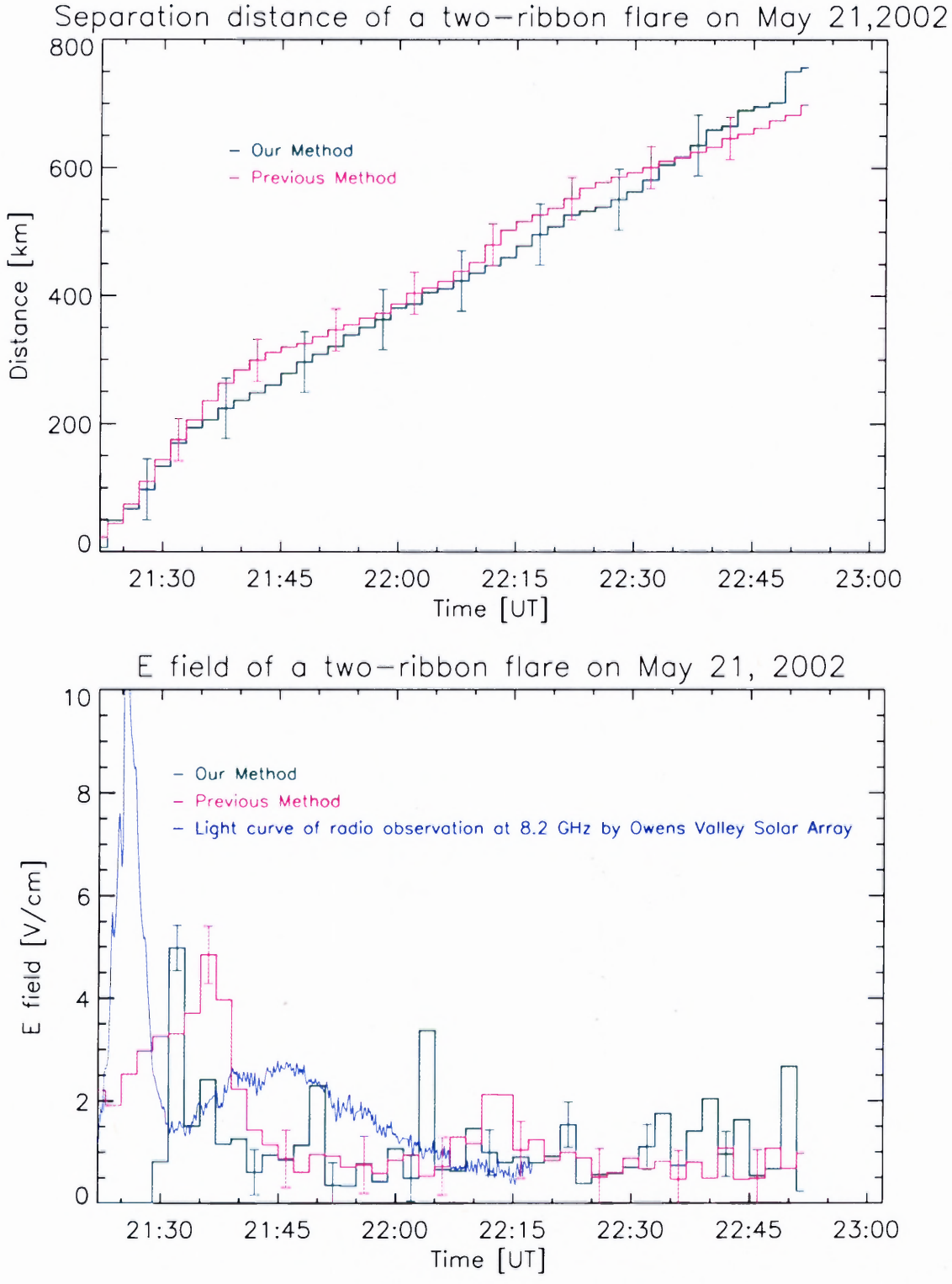

Figure 4.7 a. Comparison of the two-ribbon separation distance as a function of time for the flare happened on May 21, 2002. Red curve denotes the result of previous method, green curve denotes the result of the new method. b. comparison of derived electric field of the above two-ribbon flare. The light curve of 8.2 GHz radio emission is overplotted to indicate onset of the flare.

CHAPTER 5

AUTOMATIC SOLAR FILAMENT DETECTION

5.1 Introduction

The data volume of ground- and space-based solar observations has dramatically increased over the last decade with the advance of large-format digital detector systems. We have reached a point where it has become impossible for an observer or a scientist to scrutinize all data for energetic events such as flares, filament eruptions, and CMEs. The solar physics and space science community is currently focussing its efforts on making their data available to a variety of communities in research, technology, and education. These efforts include the Virtual Solar Observatory (VSO; Hill *et al.*, 2004), the European Grid of Solar Observations (EGSO; Bentley *et al.*, 2004), and the Virtual Solar-Terrestrial Observatory (VSTO; Fox *et al.*, 2004). The Active Region Monitor (ARM, Gallagher *et al.*, 2002) represents the sole autonomous analysis and archiving system for solar/space weather data currently operating.

Image processing and pattern recognition techniques are being actively developed to extract physical properties of solar features from vast amounts of data. The following references provide an (incomplete) overview of these efforts for several classes of solar features. Turmon *et al.* (2002) presented a Bayesian image segmentation technique to identify quiet Sun regions, faculae, sunspot umbra and penumbra in magnetograms, and white-light images obtained by the Michelson Doppler Imager (MDI) onboard the Solar and Heliospheric Observatory (SoHO). Statistical models based on *a priori* knowledge of feature labels provided by scientists are the starting point for the subsequent objective and automated classification of active regions and quiet Sun characteristics with the ultimate goal to trace solar irradiance variations with high photometric accuracy. Optimization methods such as Mean Field Fast Annealing (MFFA, Bratsolis and Sigelle (1998)) have been applied

to Hα images to automatically determine sunspot properties. Borda *et al.* (2001) used neural networks and back-propagation methods for pattern recognition and flare detection in Hα images. Support Vector Machine (SVM) based flare detection algorithms have been presented by Qu *et al.* (2003). Finally and closely related to this work, Gao *et al.* (2002) and Shih and Kowalski (2003) developed filament detection methods based on local and global thresholds and mathematic morphology.

The filament detection algorithm is based on the Stabilized Inverse Diffusion Equation (SIDE) and the SVM. The SIDE belongs to a family of Partial Differential Equations (PDEs; Pollak et al., 2000) and has the intriguing properties of effective noise suppression while at the same time preserving or even sharpening edges that are present in the input data. Pollak et al. (2002) presents a comprehensive summary of image segmentation and restoration via non-linear multiscale filtering and its applications in biomedical imaging, radar ranging, and computer vision. The SVM has been used in solar flare forecasting (Qu *et al.* 2003) and in space physics for the prediction of substorm events from solar wind and interplanetary magnetic field data (Gavrishchaka and Ganguli 2002). These authors point out the potential of SVM for space weather models, which rely on a multitude of disparate, high-dimensional, and multiscale data sets. The importance of the emerging science of space weather is illustrated in the monograph by Carlowicz and Lopez (2002). The ever-increasing number of (real-time) space weather data motivate this study of filament disappearance and the development of appropriate tools for real-time space weather predictions.

5.2 Observations

The 15 cm aperture Singer telescope takes one image per minute, high-resolution (2032×2032 pixels and 14-bit) Hα images (Denker *et al.* 1999). It is part of the Global Hα Network (GHN, Steinegger *et al.* (2001)), which is a collaboration of the Observatoire de Paris, Meudon (France), Kanzelhöhe Solar Observatory (Austria), Catania Astrophysical Obser-

vatory (Italy), Yunnan Astronomical Observatory (China), and Huairou Solar Observatory (China). The scientific objectives of the Hα full-disk observations include, among others, flare patrol, active region evolution, chromospheric differential rotation, and space weather forecasting and predictions (Gallagher *et al.* 2001). In combination with space and ground-based observations, Hα full-disk observations of high temporal and spatial resolution have proven to be a key diagnostic to determine the magnetic field topology among the photosphere, chromosphere, transition region, and corona, which provides the boundary conditions and environment for energetic events such as flares, filament eruptions, and CMEs (Denker *et al.* 1999).

A test data-set consisting of 50 randomly selected images was extracted from the daily 1-minute cadence Hα full-disk images covering the time period from 1 January 1999 to 1 September 2004. This data-set is only a small fraction but a fair representation of the entire data-set, which is comprised of more than 500,000 images. Both low and high contrast filaments are present and the samples include the images at solar maximum as well as the images at moderate to low solar activities. The overall performance evaluation is based on these 50 images. For the illustration of the algorithm, an Hα full-disk image obtained on 24 October 2004 (see Figure 5.1) is used. Denker *et al.* (1999) presented dark and flat-field correction as well as a precise photometric calibration and limb darkening subtraction to enhance the contrast of solar features on and off the disk. To compare images obtained at different time and correct them for the solar differential rotation, the application makes use of Interactive Data Language (IDL) routines provided in the Solar SoftWare (SSW) IDL library (Freeland and Handy 1998).

5.3 Filament Detection

In this section, the various steps of filament detection, which are summarized in the flow chart depicted in Figure 5.2 are discussed. The dark background of the Sun is first removed. Then the SIDE is used to suppress the Hα fibrils in the quiet Sun regions and enhance

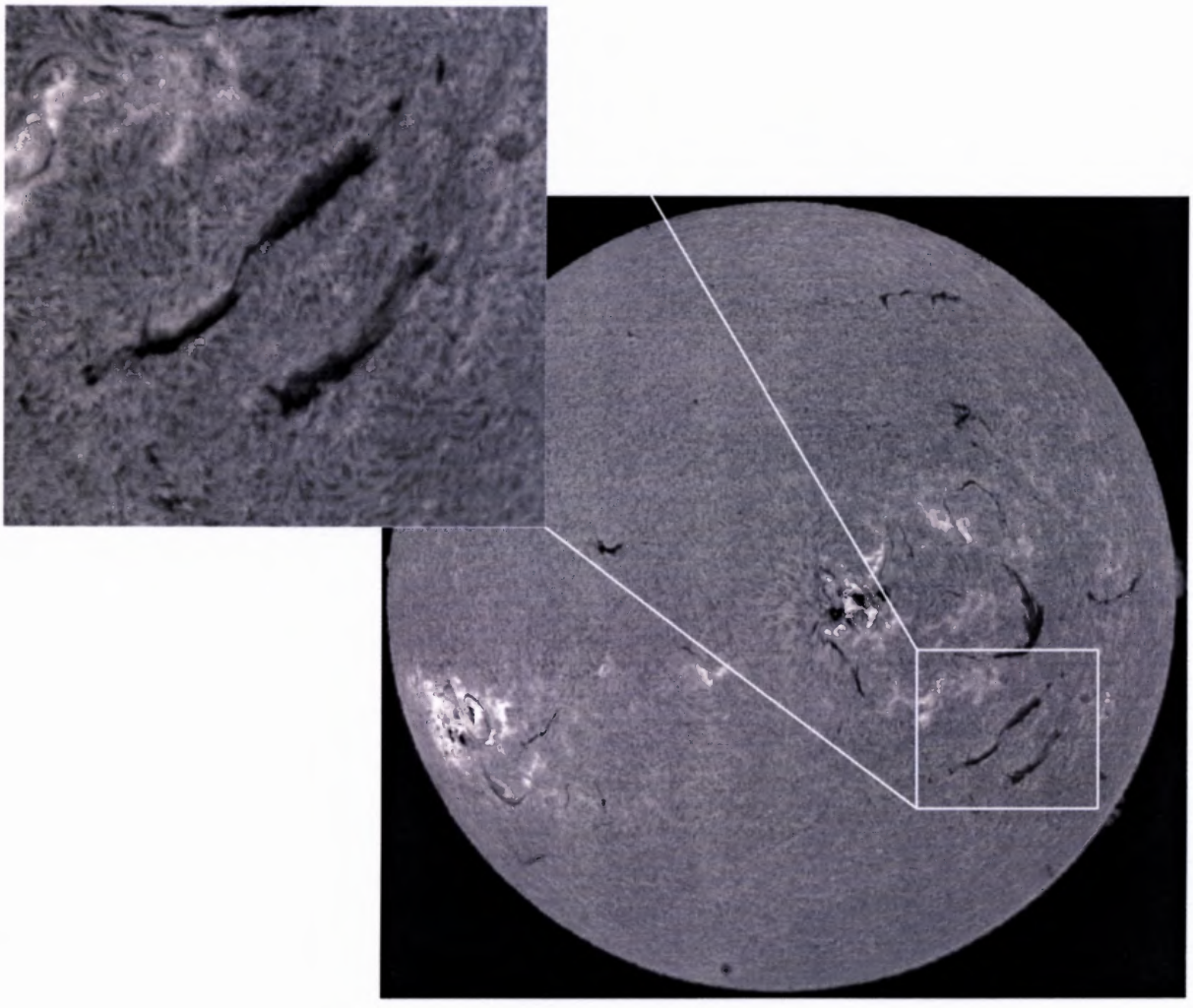

Figure 5.1 Hα filtergram obtained with the Singer full-disk telescope at the Big Bear Solar Observatory on October 24, 2003. A region shown in the upper left at full resolution was selected to illustrate the various steps of the filament detection procedure throughout the remainder of this study.

solar filaments. After filaments are segmented using the adaptive thresholding method, sunspots are distinguished from filaments using the SVM classifier with nine features. The filament morphology is obtained by several mathematical morphology methods. Finally, the filament disappearance is reported.

5.3.1 Stabilized Inverse Diffusion Equation (SIDE)

Pollak et al. (2000) provides a comprehensive introduction to nonlinear multiscale filtering and its application to image segmentation and restoration. The SIDE (Pollak *et al.* 2000) is used to transform a highly detailed Hα full-disk image (fibrils, filaments, sunspots, plage regions, etc.) to a coarser image with less detail that allows an easier detection of filaments.

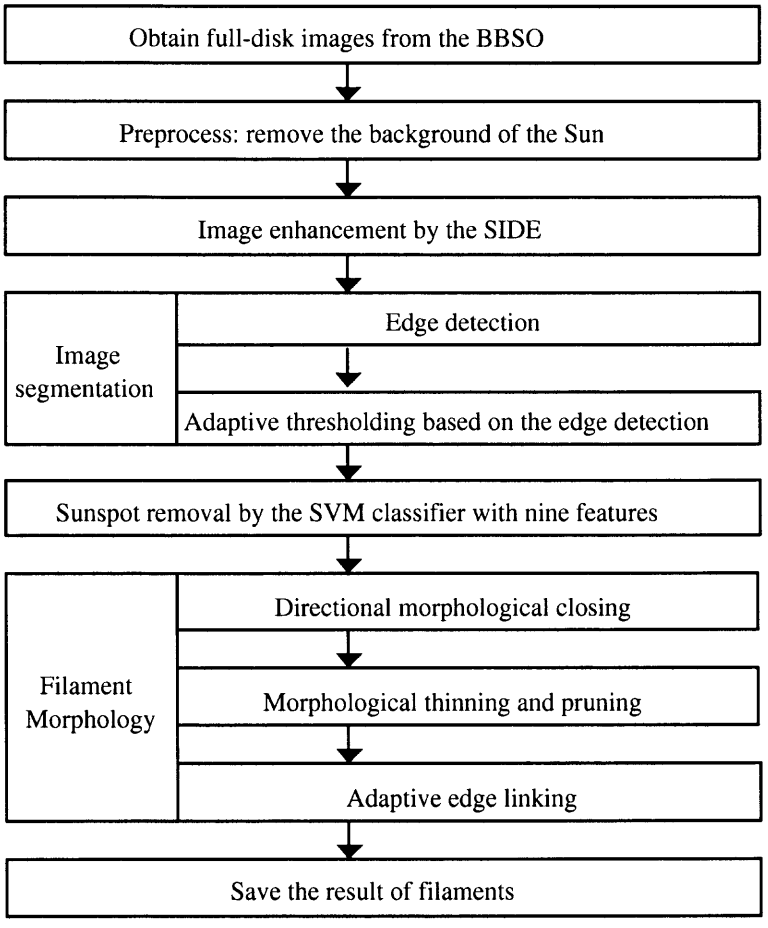

Figure 5.2 Flow chart of the filament detection.

Plage regions are exceptionally bright in Hα and cannot be confused with dark filaments. Filaments are extracted from the quiet Sun background, which is dominated by dark Hα fibrils. These fibrils have typical sizes of about $10''$. Their orientations often trace the local field of geometry; i.e., they are not randomly oriented. However, their distributions are very uniform. Therefore, Hα fibrils share many characteristics with noise that is present in images. This motivates the use of the SIDE, which has been successful in restoring images in the presence of pervasive, large-amplitude noise while preserving or even sharpening edges. The SIDE is a fast and robust image segmentation and restoration algorithm. It is robust against image blur, which affects the full-disk images due to atmospheric turbulence.

My first goal is to suppress the Hα fibrils in the quiet Sun regions. This can be achieved by linear shift-invariant filters such as a Gaussian filter to suppress high spatial frequencies and thus smooth sharp edges. What we need is a stabilized inverse diffusion near edges and a heat-equation-like behavior in homogeneous regions without edges. Therefore, the SIDE filter is utilized to enhance edges and remove unimportant small features and has been successful used to enhance low contrast filaments.

5.3.2 Edge Detection and Adaptive Thresholding

A new method is proposed for adaptive threshold selection from the results of edge detection. The edges of filaments are detected by Sobel operator which performs a 2D spatial gradient measurement on an image and emphasizes the regions of high spatial gradient that correspond to edges. It is used to find the approximate absolute gradient magnitude at each pixel (Jähne 1997).

My adaptive thresholding method contains two steps. In the first step, the global threshold is computed for the entire Hα image to select filaments that have visible high contrasts relative to background. In the second step, the local thresholds are calculated for sub-images of 100×100 pixels to select filaments that have locally high contrasts relative to background. The threshold selection method is described as follows.

A sequence of self-adaptive thresholds $t_1 < t_2 < ... < t_n$ ranging from 0 to the median of pixel intensities of the image are chosen for segmenting dark regions. Suppose r_i is an individual segmented region by adopting the threshold t_i, and r_i' is the new expanded region where $r_i' = r_i - r_{i-1}'$. Initially, we have $r_1' = r_1$. A set of gradients g_i is calculate by Equation 5.1.

$$g_i = \frac{\text{Sobel}(r_i')}{n_{r_i'}}, \tag{5.1}$$

where $n_{r_i'}$ is the number of pixels in the region r_i'. The best global threshold T is computed by Equation 5.2.

$$T = t_{i_{\max}}, \tag{5.2}$$

where $i_{\max}$ is the index of the maximum of g_i.

For example, we utilize thresholds $t_1 < t_2 < ... < t_n$ to obtain regions $r_1 < r_2 < ... < r_n$ respectively. New expanded regions are $r_1', r_2', ..., r_n'$. The average spatial gradient g_i for each pixel on r_i' has been calculated by Sobel operator. When a new expanded region r_i' meets the edges of filaments, the gradient g_i reaches the maximum. By finding the maximum gradient for each new expanded regions g_i, the optimal global threshold T for segmenting solar filaments is obtained.

In the second step, the local threshold is selected using the same threshold selection method with the following two criteria: The local threshold $t_{loc} = T \pm 30$; the size of the region obtained by t_{loc} is less than three times of the region obtained by T. After segmentation, the small regions whose areas are less than 10 pixels are considered as noises and are removed.

5.3.3 Sunspots Removal

An efficient feature-based classifier, called the Support Vector Machine (SVM), is used to distinguish sunspots from filaments. We represent a sunspot by nine features and use the SVM for sunspot classification. Experimental results show that this method significantly

improved the classification rate in comparison with other methods (e.g., Shih & Kowalski, 2003).

The SVM applied in the solar flare detection is superior to the neural network classifier (Qu *et al.* 2004). In the experiment, the SVM classifier is used with the linear kernel. The comparisons between the linear kernel and other kernels can be found in Qu et al. (2003). For further information regarding the SVM, readers can refer to Guyon and Stork (2002) and Vapnik (1998).

According to the physical nature of sunspots, we represent them by nine features to be the input of the SVM. These nine features can be computed from the segmented results and the original Hα image. We denote the segmented regions for the whole image as A, a window of 100×100 pixels which is centered on a region S as $W1$, and a window which just encloses a region S as $W2$. The nine features of S are described below.

1. The number of pixels in S, say n_S.

2. The radial position of the center of S. When the radial position is far away from the center of Sun, the size and darkness of sunspots could be quite different.

3. The distance from the center of S to the equator of Sun. It is calculated as

$$d = y - y_c, \tag{5.3}$$

   where y and y_c denote the respective Y-coordinates of S and the equator of Sun.

4. The ratio of mean brightness between S and A is defined by

$$ratio_1 = \frac{\bar{x}_S}{\bar{x}_A}, \tag{5.4}$$

   where $\bar{x}$ denotes mean brightness.

5. The ratio of standard deviation between S and A is defined by

$$ratio_2 = \frac{\sigma_S}{\sigma_A}, \tag{5.5}$$

where std denotes standard deviation.

6. The mean brightness of the window $W1$.

7. The standard deviation of $W1$.

8. The shape feature h_1 is defined by

$$h_1 = n_S / n_{W2}, \tag{5.6}$$

where n_{W2} denotes the number of pixels of $W2$.

9. The shape feature h_2 is defined by

$$h_2 = L_{W2} / H_{W2}, \tag{5.7}$$

where L_{W2} and H_{W2} denote the length and height of $W2$, respectively.

5.3.4 Filament Morphology

Because some adjacent regions are considered to be one component, it is necessary to check the spatial relationship between regions to determine if they should be integrated. This is achieved by morphological closing, thinning, pruning and adaptive edge linking methods.

The original pruning method is used to remove the fixed length of a parasitic component. For filament pruning, an adaptive pruning method is implemented by monitoring X_3 which has described in Chapter 2.3.3. Step 1 to step 3 for pruning are repeated to remove pixels until X_3 has two points left. Then the fourth step is invoked to compute the

final results. The pruned results are lines which represent the spines of filaments. The two end-points in X_3 are called the foot-points of a filament.

Big gaps of broken filaments could not be filled in by directional morphological closing. Therefore the application adopts the adaptive edge linking method developed by Shih and Cheng (2004). The basic idea is to connect edges based on the orientation of spines which is obtained by the edge slope direction of the foot-points. Adaptive edge linking is performed in an iterative manner, so that the spines of broken filaments can be linked up gradually and smoothly. The final results of spines and foot-points of filaments are used for the detection of filament disappearance.

5.3.5 Detection of Filament Disappearance

Finally, the application detects filament disappearance by comparing the results of two successive days. A computer program has been developed to detect filament disappearance automatically and send the report to the researchers as described in Figure 5.3. The Hα images of two consecutive days are loaded. They are matched after the differential solar rotation correcting (SSW; Freeland and Handy, 1998). My enhancement, segmentation, and morphology methods are applied to obtain the spines and foot-points of filaments. To detect the filaments that have disappearance on the two images, the following methods are proposed:

1. The spines of filaments are dilated with a structuring element of 20×20 pixels.

2. When two filaments are overlapped, the application matches them according to the size of spines and the intensity of filaments. The intensity of a filament is calculated by the mean brightness of the filament. If the size of a spine on the current day is greater than 40% of the previous day, it is considered as their sizes are matched. If the intensity of a filament on the current day is less than 1.5 times of the previous day, it is considered as their intensities are matched.

3. Filaments are matched if both sizes and intensities are matched. Unmatched filaments on the previous day are reported as disappearance.

5.4 Results

In this section, the results of image enhancement, image segmentation, sunspot removal, filament morphology, and filament disappearance are demonstrated respectively. In the SIDE experiment, the image pixel is 8 bits and L in Equation 2.11 is set to 100. In Figure 5.4a, it shows a small Region-Of-Interest (ROI), which is located close to west limb in the southern hemisphere of the Sun as indicated in Figure 5.1. The ROI is selected since it includes a variety of Hα features such as a plage region, quiet Sun Hα fibrils, and two filaments. The results of soft morphological filter (Shih and Puttagunta 1995), Perona-Malik filter (Perona and Malik 1990) and the SIDE are shown in Figure 5.4. it observes that low contrast filaments are best enhanced by the SIDE. Besides, background and dark Hα fibrils

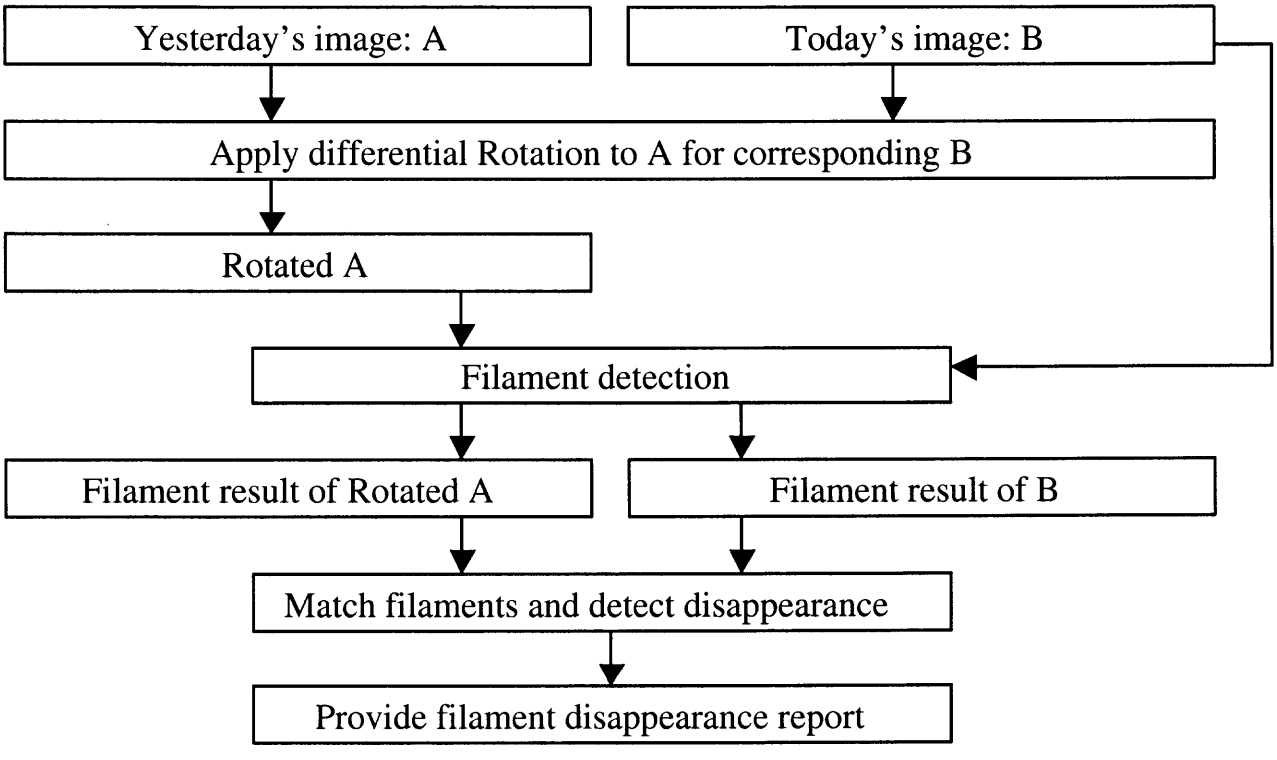

Figure 5.3 Flow chart of the detection of filament disappearance.

are suppressed without blurring the edges of solar filaments.

The results of enhancement and segmentation are shown in Figure 5.5. After segmentation, the SVM classifier is applied to classify sunspots. 50 sunspots and 50 non-sunspots are collected, and 25 sunspots and 25 non-sunspots are randomly selected for training and the remaining for testing the SVM classifier. The sunspot removal result is shown in Figure 5.5. The detection rate of filaments and sunspots is shown in Table 5.1.

Morphology methods are used to connect broken filaments and find spines and footpoints of filaments. Figure 5.6 shows the results of directional morphological closing, thinning, pruning, and directional edge linking.

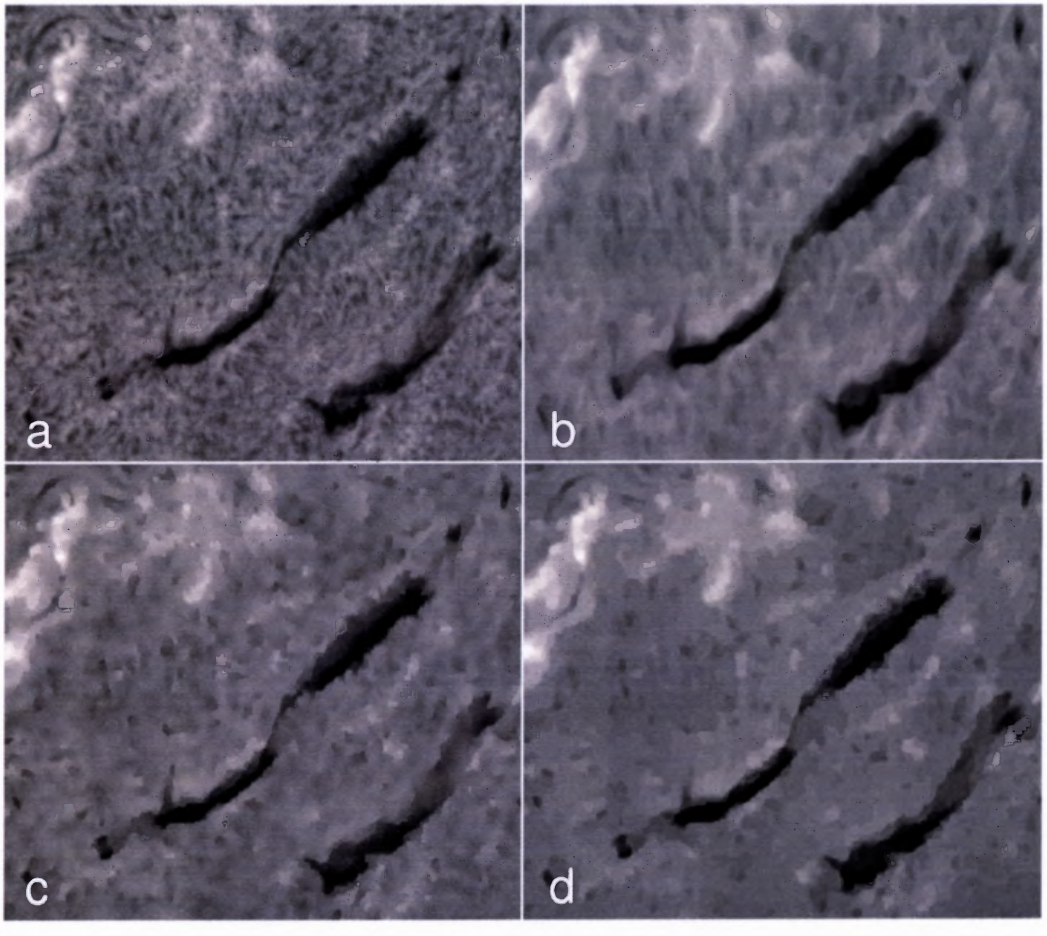

Figure 5.4 (a) The ROI with quiet Sun Hα fibrils, a plage region, and two filaments. For comparison, (b) recursive soft morphological filter, (c) Perona-Malik filter, and (d) the SIDE have been applied to the ROI to illustrate feature suppression and edge enhancement/preservation.

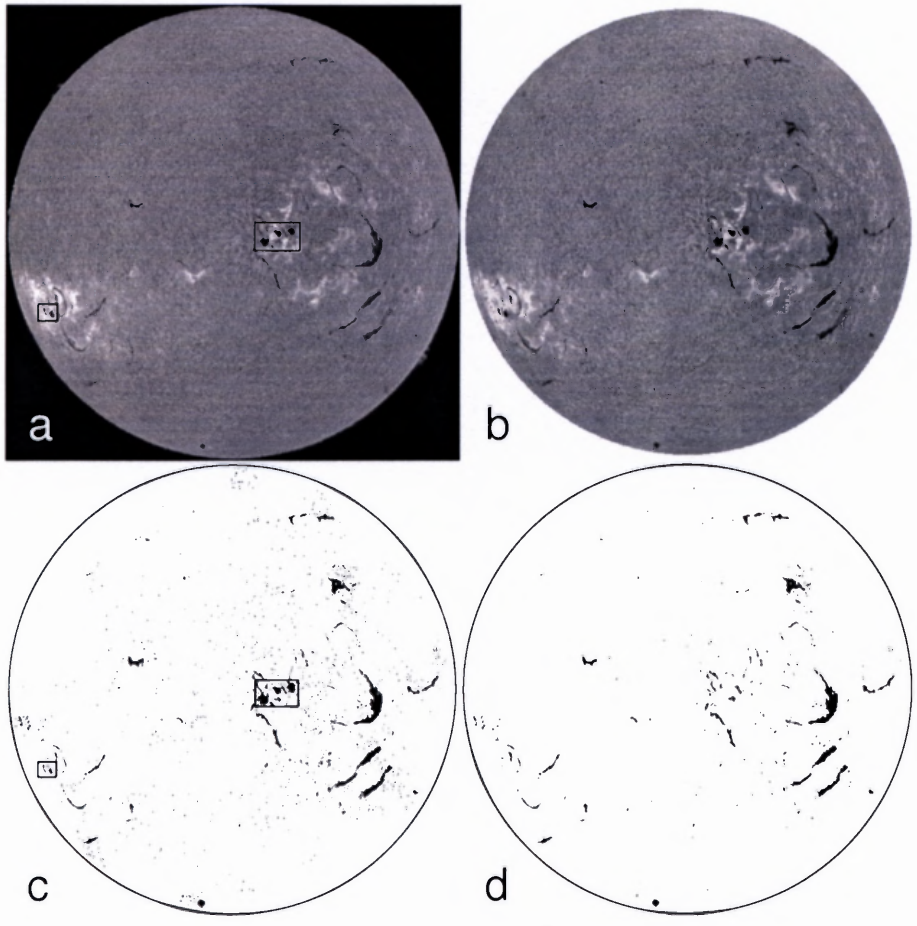

Figure 5.5 (a) Original Hα full-disk image, (b) result of the dark background removal and the SIDE, (c) corresponding bi-level image after image segmentation using the adaptive thresholding method, and (d) final result without sunspots (as enclosed by two black boxes in (c)) and small regions comprised of less than ten pixels.

Method	LF	SF	CS
Gao *et al.* (2002)	50.2%	5.1%	0.0%
Shih and Kowalski (2003)	94.3%	54.9%	60.0%
Ming *et al.* (2005, this study)	100.0%	75.0%	94.0%

[a]LF - Large filament whose length is larger than or equal to 200 pixels;
[b]SF - Small filament whose length is less than 200 pixels;
[c]CS - Sunspot.

Table 5.1: Filament and sunspot detection rate.

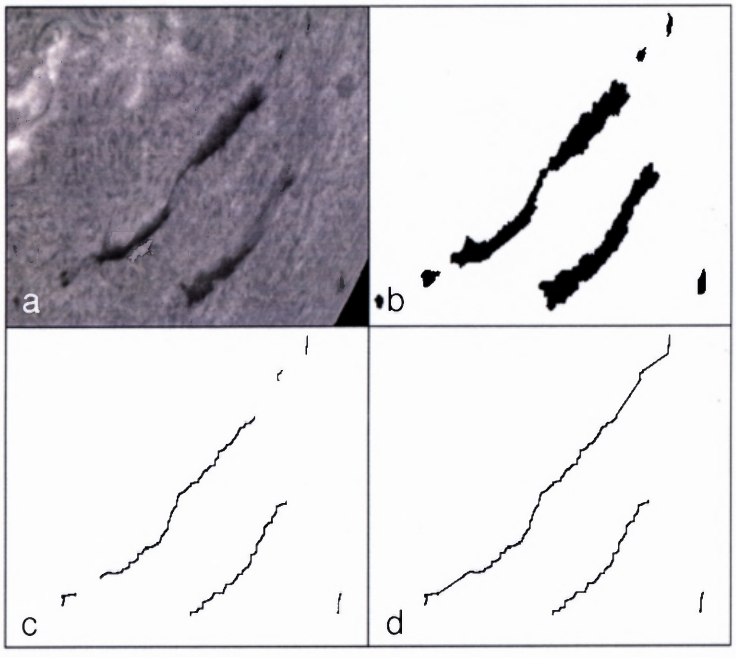

Figure 5.6 Filament morphology. (a) Original image, and results of (b) directional morphological closing, (c) morphological thinning and closing, and (d) adaptive edge linking.

Finally, filament disappearance is detected. Since the research on filament disappearance is focused on large filaments, the disappearance of small filaments whose spines are less than 60 pixels is not reported. Figure 5.7 shows the results of filament disappearance, where the filament on the right side disappears.

A computer with an Intel Celeron Processor (2.40GHz) was used for the filament detection and the development language is IDL. A break-down of computational time in

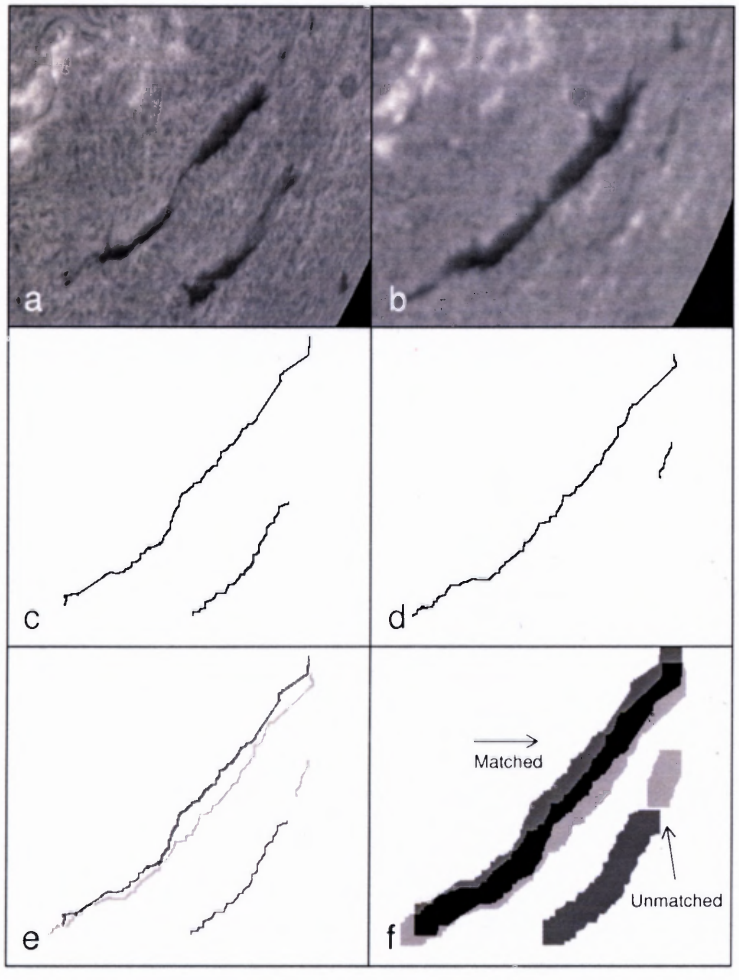

Figure 5.7 Detection of the filament disappearance. (a) ROI on 24 October and (b) 25 October 2003 while approaching the solar limb, (c, d) corresponding spines of the filaments, (e) superposition of the filament spines, and (f) superposition of the filaments. A differential rotation correction has to be applied to the ROI in (a) to match the filament on the following day.

each method is given in Table 5.2 for a full-disk image of 2032 × 2032 pixels with 8 bits per pixel. The total time is less than one minute, which is negligible if only one image is processed per day. The real detections would be considered real-time if using 1 minute cadence.

5.5 Summary

A new automatic filament detection algorithm based on image enhancement, segmentation, pattern recognition, and mathematical morphology methods has been successfully applied to Hα full-disk filtergrams. For the first time, the SIDE has been adopted to enhancing filaments from the quiet Sun background. Filaments are extracted using an adaptive threshold selection method. The filament segmentation method is efficient and robust compared to other methods Gao *et al.* (2002); Shih and Kowalski (2003). The SVM is an essential tool to distinguish between filaments and sunspots using nine features. Filament properties such as foot-points, spines, positions, and disappearances are subsequently retrieved by directional morphological closing, thinning, pruning, directional edge linking, and component matching methods. These properties will be stored in a database for context-based data query and BBSO's space weather forecast and prediction.

Method	**Time**
Image enhancement by the SIDE	20.68 s
Adaptive thresholding	6.62 s
Sunspot removal	0.63 s
Directional morphological closing	0.42 s
Morphological thinning and pruning	4.02 s
Directional edge linking	5.51 s
Filament disappearance detection	2.45 s
Total	**40.33 s**

Table 5.2: Computational time of the filament detection algorithm.

CHAPTER 6

AUTOMATIC DETECTION OF CORONA MASS EJECTIONS

6.1 Introduction

CMEs were statistically analyzed by Howard et al. (1985) who categorized them into nine structural classes, known as Halo, Curved Front, Loop, Spike, Double Spike, Multiple Spike, Streamer Blowout, Diffuse Fan, and Complex. Halo CMEs are the most energetic, and diffuse fan CMEs are the least energetic. A halo CME appears as a diffuse halo around the surface of the Sun and may fill the entire or a part of corona. If a CME is an Earth-directed event, it usually shows the halo structure in the LASCO image. It has been proven that halo CMEs have strong impact on the Earth and cause geomagnetic storms Brueckner *et al.* (1998). The detection and classification of CMEs in real-time can provide an early warning of the occurrence of earth-directed CMEs and the geo-effectiveness of CMEs. By combining with the detection of filaments and flares in our previous work (Qu *et al.*, 2003, 2005), we can establish the relationship among CMEs, filament eruptions, and flares.

Based on Howard et al. (1985)'s CME classification, Dai et al. (2002) further classified the aforementioned nine classes into three categories: strong, medium, and weak CMEs. A strong CME is defined as having a large amount of masses moving with a high speed, such as a halo, curved front, and most complex CMEs. By their studies, the speed of a strong CME is greater than 500 km/s, the span is greater than 60°, and the mass is greater than 6.0×10^{30} ergs. The real velocity of a CME may be larger than the calculated speed because of the projection effect. In Figure 6.1, examples of a strong and weak CMEs in the LASCO C2 image are shown.

CMEs can be observed by tracking a sequence of SOHO LASCO images. The LASCO group provides a CME catalog based on the detection by human operators. Before human visual detection, the LASCO images are preprocessed to enhance the CME struc-

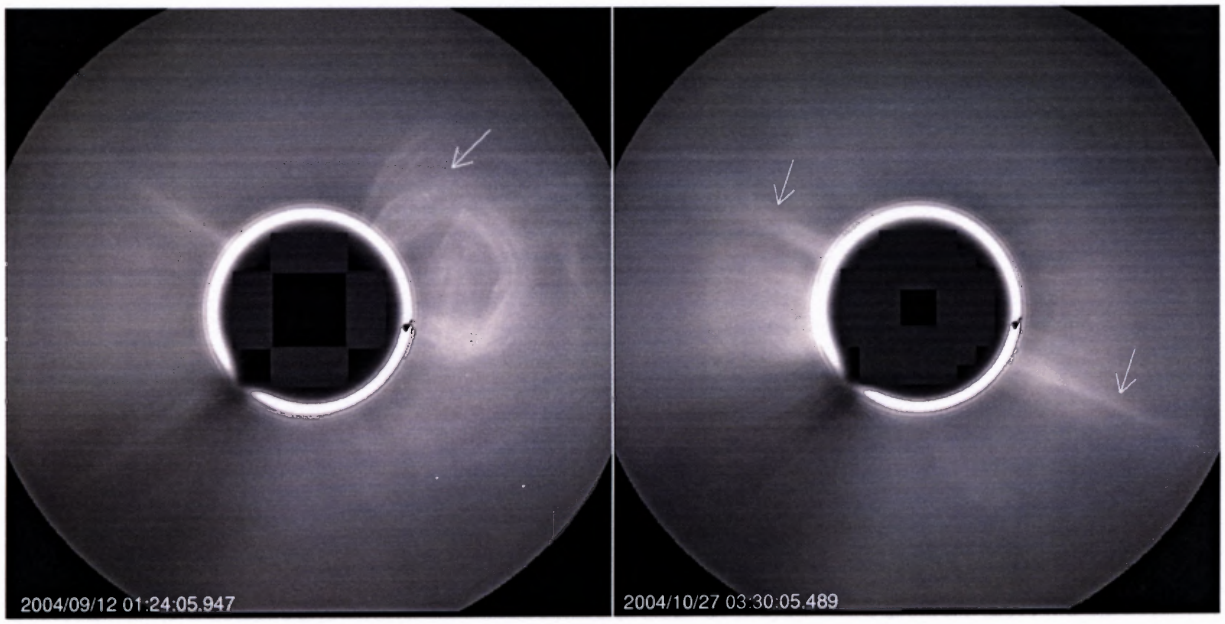

Figure 6.1 Left: a strong CME on the right side of the LASCO C2 image, Right: weak CMEs on the LASCO C2 image.

tures. CMEs can be easily enhanced by computing the differences between two consecutive LASCO images because CMEs are visible as a sudden increase of brightness. Stenborg *et al.* (2003) presented their CME enhancement method using wavelet transform which may improve the accuracy of human detection. The drawback is that human detection is slow and its results are subjective. The decision rules may vary from one operator to another.

Berghmans (2002) introduced an automatic CME detection procedure using image processing methods on LASCO images. He demonstrated the difficulty in detection since no CME catalog could have a 100% success rate. Furthermore, Robbrecht and Berghmans (2004) presented automatic detection of CMEs in sequences of LASCO images which uses Hough transform to detect CMEs on the running differences.

In this chapter, we present a new detection and classification algorithm to track each CME region. There are three steps in our algorithm: preprocessing, detection, and classification. The preprocessing is used to calibrate LASCO images, filter noises, and produce running differences. The detection is used to segment the differences and obtain CME characteristics such as angular width of span, height, and speed. The classification is used to distinguish the strong CMEs from other CMEs using the SVM classifier.

6.2 Preprocessing

The LASCO C2 and C3 images in an FITS format are obtained from the SOHO web site. The X and Y coordinates of solar centers, extracted from image headers, are used to correlate a sequence of LASCO images. The images have two different sizes: 512×512 and 1024×1024. All images are resized to 512×512. In Figure 1.5, an inner ring, which indicates the edge of a coronagraph occulting disk, is located at the center of a LASCO image. In order to remove the inner ring, a dark circle is used to cover the center of the image. The radii of the circles used for LASCO C2 and C3 images are 100 and 40 pixels, respectively. After that, brightness normalization is applied. The brightness is adjusted according to the ratio of each exposure time extracted from image headers. If the gray level ranges

of mean brightness between two consecutive images are different after the exposure time adjustment, the images are normalized using the mean brightness ratio of the current image to the previous image. If it is greater than 1.1 or less than 0.9, Equation 6.1 is applied to correct the brightness of the current image.

$$M_c' = M_c \times \frac{\bar{x}_p}{\bar{x}_c}, \tag{6.1}$$

where M_c and M_c' denote the current images before and after the normalization, and $\bar{x}_p$ and $\bar{x}_c$ denote the mean brightness of no-CME regions on the previous and current images, respectively. When a CME appears on the current image frame, the mean brightness normalization may reduce its brightness. To avoid this problem, the mean brightness ratio is calculated based on the regions with no CME. The pixel with a gray level between the minimum and the median value of the image is considered as the one with no CME.

On the LASCO image, streamers are steady bright structures, and therefore are different from CMEs as having sudden brightness changes. To distinguish a CME from other stable bright structures, all the CME detections are based on the running difference, which is defined as the difference between the current and the reference images. It is also useful for the measurement of the CME's moving-front (i.e. the top moving edge) and foot-point (i.e. the lower moving bottom). The running difference D can be obtained by Equation 6.2.

$$D = M_c - G, \tag{6.2}$$

where G is the reference image and M_c is the current image. The running difference is obtained by subtracting the previous image from the current image. There are two disadvantages in using the previous image frame as the reference. One is if some regions of the previous image were badly captured, the running difference becomes unstable. The other is when a CME is evolved on a sequence of image frames, the running difference will succeed in detecting the moving front of the CME, but will fail in detecting its beginning foot-point

because some parts of CME regions overlap in a sequence of images. In this paper, a stable reference image G' is generated by using the combination of current and previous images, as shown in Equation 6.3.

$$G' = G \times c_1 + M_c \times c_2, \tag{6.3}$$

where c_1 and c_2 denote the percentages of effectiveness on the reference and the current images. Based on our experiments, we set $c_1 = 90\%$ and $c_2 = 10\%$. However, if a CME is detected in the previous image frame, $c_1 = 100\%$ and $c_2 = 0\%$ are used.

For detecting the regions that have a large brightness increase in the running difference, but are relatively darker than other regions on the original LASCO image, the partial division image V in Equation 6.4 is adopted.

$$V = \frac{D_2}{G_2}, \tag{6.4}$$

where G_2 denotes the region on the reference image having pixels brighter than the median value of the original LASCO image and D_2 denotes the region corresponding to G_2 on the difference image. By using the relatively bright regions for G_2, the division of dark regions is avoided. The resulting running difference and division images are used for the segmentation of CMEs in the next step.

Some of LASCO images have missing blocks which appear as dark or bright squares in image frames. Three criteria are proposed to find the blocks. When a dark missing block first appears in the current image, its appearances in the current and the running difference images are both dark. Its gray level on the current image is close to the minimum value of the image, and its gray level on the running difference is less than the negative threshold t_1. Because a missing block contains a larger number of pixels as compared to small noises, the number of pixels for an individual block region is set to be greater than the threshold t_2. The thresholds t_1 and t_2, obtained from our experiments, are -400 in brightness and 50 in pixel, respectively. A bright missing block can also be detected similarly. Those missing

blocks can be replaced by the corresponding pixels in the previous images.

6.3 Automatic Detection of CMEs

In order to detect the CME on the current image, the current and two previous images are used as inputs. Two running differences are obtained from these three inputs, and are segmented into two binary differences by applying a thresholding method. The motion of a CME can be obtained by comparing the consecutive binary differences.

6.3.1 Segmentation of CMEs

The running differences are constructed by using the consecutive LASCO images. To produce the binary differences which separate CME regions from background, a thresholding method is used. The fixed value of the threshold would produce an unstable result because the contrasts of running differences vary from time to time. Therefore, the threshold is computed for an image using the median and standard deviation. The threshold is greater than the median value of the difference image because CME regions are contrast enhanced. Based on extensive experiments, the thresholds for the images of the C2 difference D and the C2 division V are chosen as $m+s$, where m and s are the median and standard deviation of the C2's D and V images, respectively. Similarly, we use the thresholds for the C3's D and V images as $m+1.5s$ based on experiments. These automatic thresholds are robust with respect to different contrasts. The final segmentation result is the summation of the two segmentation results from the difference and the division images.

After segmentation, morphological closing Shih and Mitchell (1989); Shih and Wu (2004) is applied to the binary segmentation image to eliminate small gaps. A 5×5 structuring element with the gray level of all ones is used to perform the binary closing. The

closing of A by B, denoted by $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B. \tag{6.5}$$

The closing of A by B is simply the dilation of A by B, followed by the erosion of the result by B. In order to calculate the features of CMEs easily, the binary segmentation image is reformed to a $360 \times r$ binary angular image, where r is the maximum distance from any pixel on the LASCO image to the edge of the C2 or C3 occulting disk. The results of the segmented and the angular images are shown in Figure 6.2.

6.3.2 Features of CMEs

In the detection, all the segmented regions in a CME frame are treated as CME candidate regions, but only continuously moving regions on consecutive images are treated as real CME regions. A CME candidate region is classified as a CME region if it occurs in the current image frame as well as in the previous image. The region corresponding rules are given as follows: the distance of the two centers of corresponding regions is less than t_1, where t_1 is the threshold for the maximum movement, and the span difference between two overlapping regions is less than t_2, where t_2 is the threshold for the maximum degree span change. For the LASCO C2 image, t_1 is chosen as 100 pixels; for the C3 image, as 40 pixels. t_2 is chosen as a half of the span width. In addition, the speed and the new increasing part for the CME region are computed by comparing the corresponding CME regions on the two binary angular images. Let us denote a CME region in the current image and its corresponding region in the previous image as A and A_p, respectively. The CME properties of A in an image frame are listed in Table 6.1.

The detection reports for a CME on 2002 September 1 are shown in Figures 6.3 and 6.4. The classification of strong, medium, and weak CMEs is presented in the next section. Figures 6.5 and 6.6 show the height and velocity for the CMEs on 2002 September 1 in C2

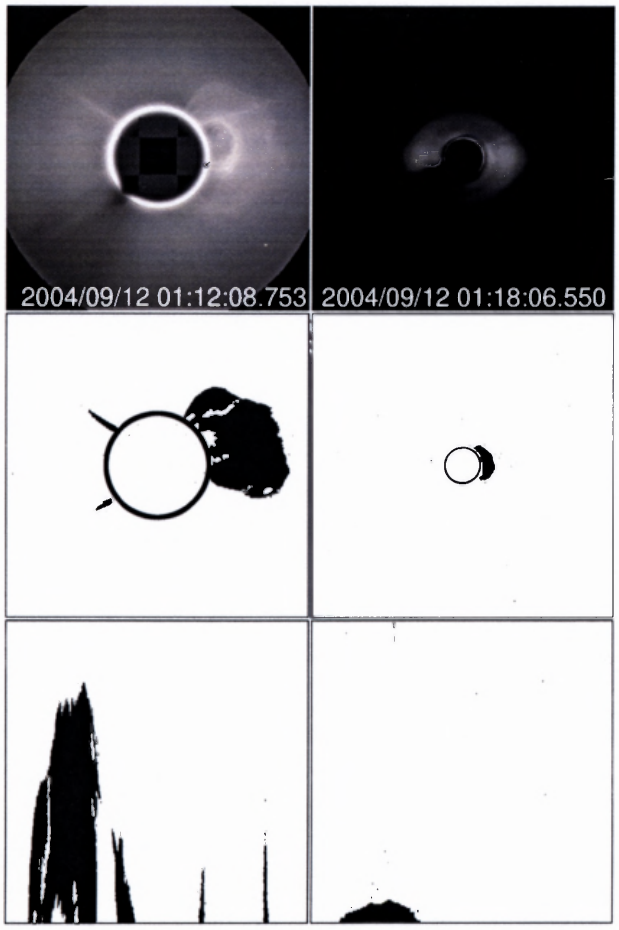

Figure 6.2 Upper: LASCO C2 and C3 images with CMEs observed on 2004 September 12. The size of images is 512×512 pixels. Middle: The binary results are obtained by our threshold method. The dark region on the right-hand side of the image is a curve-front CME. The size is 512×512 pixels. Bottom: The angular images are obtained. The degree of angle [0,359] is counted from north clockwise. The original size of the C2 angular image is 360×156 pixels, and of the C3 angular image is 360×216 pixels.

No.	Description of the CME properties
1	The exposure time of the LASCO image.
2	The pixel size of the LASCO image.
3	The mean brightness value for A on the reference image.
4	The mean brightness value for A on the current image.
5	The mean brightness value for A on the running difference.
6	The standard deviation for A on the running difference.
7	The number of pixels for A on the running difference.
8	The threshold for segmenting A from the running difference.
9	The maximum of Y-coordinate for A.
10	The center of Y-coordinate for A.
11	The minimum of Y-coordinate for A.
12	The left side of X-coordinate for A.
13	The center of X-coordinate for A.
14	The right side of X-coordinate for A.
15	The angular width of span of A.
16	The height of A subtracts the height of A_p: h_1.
17	The height of the subtraction region between A and A_p: h_2. It may differ from h_1.
18	The speed that is computed using the h_1 divided by the interval time cadence.
19	The speed that is computed using the h_2 divided by the interval time cadence.
20	The span width for the new increase.
21	The center X-coordinate for the new increase.

Table 6.1: The detected CME properties of a CME region. The features 16-21 are obtained using the corresponding regions.

and C3 images, respectively. The speed of the CME is around 300 km/s which is almost equal to the CME speed listed in the LASCO catalog.

6.4 Classification of Strong, Medium, and Weak CMEs

A strong CME consists of a large amount of fast moving masses. It is likely to be a halo, curved front, or complex CME. In our CME classification step, all halo and curved front

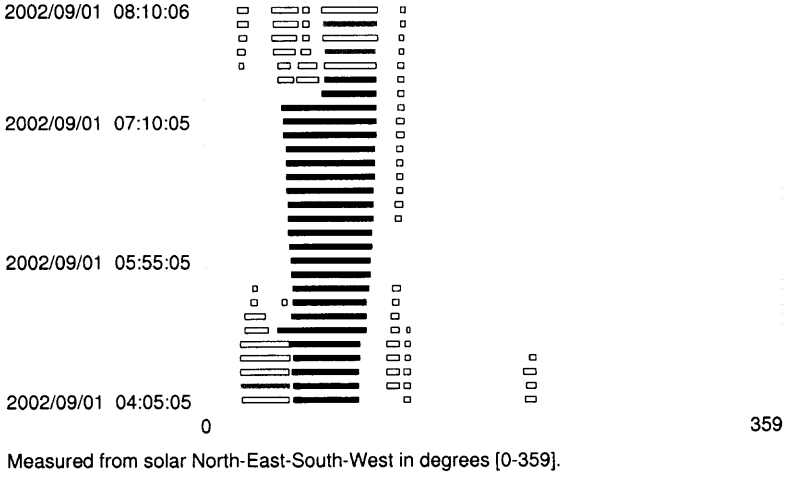

Figure 6.3 Detection and classification report for a detected CME using LASCO C2 images on 2002 September 1. Apparently, there is a strong CME on East from 04:05 to 08:10UT. Solid dark, solid gray, and empty rectangle denote strong, medium, and weak CMEs, respectively.

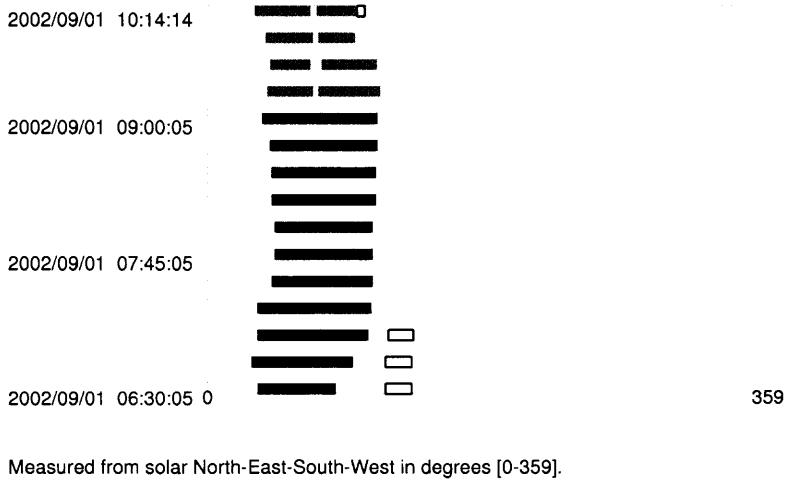

Figure 6.4 Detection and classification report for a detected CME using LASCO C3 images on 2002 September 1. There is a strong CME on East from 06:30 to 10:14UT. Solid dark, solid gray, and empty rectangle denote strong, medium, and weak CMEs, respectively.

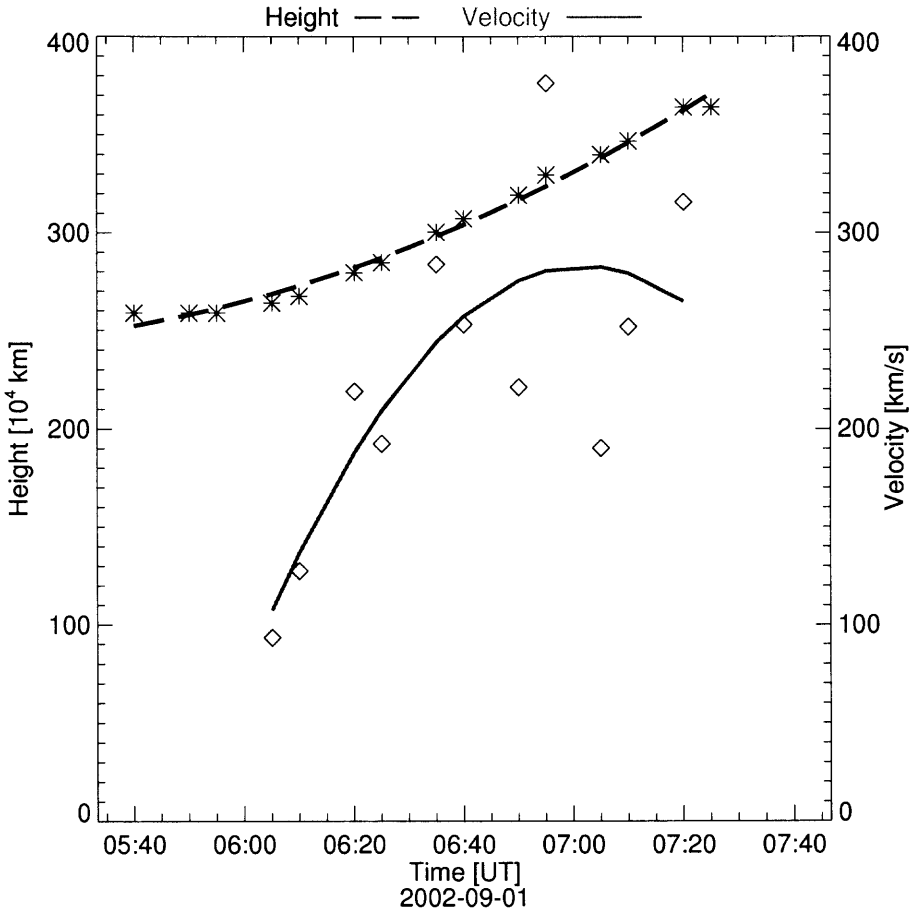

Figure 6.5 Detected height and velocity profile of a CME using LASCO C2 images on 2002 September 1.

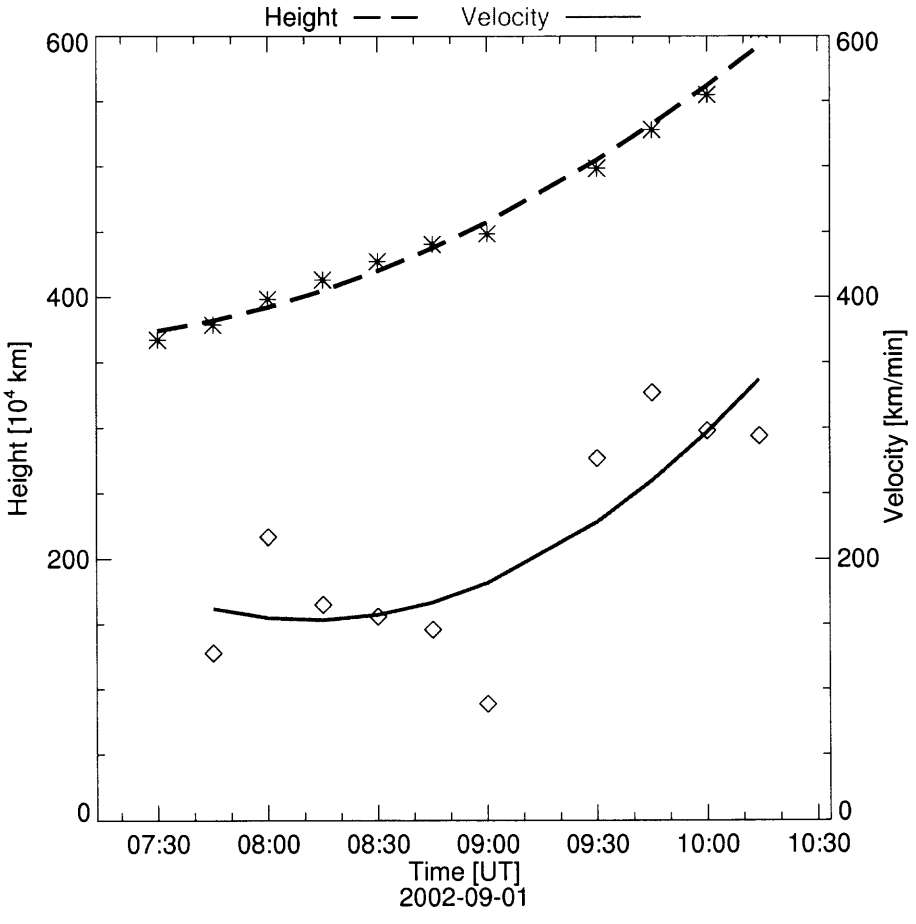

Figure 6.6 Height and velocity profile of a CME using LASCO C3 images on 2002 September 1.

and most of the complex CMEs are considered to be 'strong'. For the present study, a CME can be represented by twenty-one features which are obtained in the previous step. The SVM training provides the corresponding weights for each feature. Six features with significant weights are selected as the inputs for classification. The six input features are proven to be robust based on our experiments of feature combinations and comparisons. In this paper, the SVM classifier with a linear kernel is used for distinguishing the strong CME from others. This is the first time that the SVM is applied for the CME classification. The comparisons between the linear kernel and others can be found in Qu et al. (2003). For further information regarding the SVM, readers can refer to Vapnik (1998).

The six inputs for A are: the mean brightness in the running difference, the number of pixels in the running difference, the angular width of span, the center Y-coordinate of A, the span width for the new increase, and the speed described in Feature 19 of Table 6.1. After strong CMEs are distinguished from other CMEs, further classification is needed to distinguish medium from weak CMEs according to the studies by Howard et al. (1985) and Dai et al. (2002). The speed of medium CMEs is greater than 300 km/s, while the speed of weak CMEs is less than 300 km/s.

A list of CMEs are randomly selected in 2004. Assuming that human classification for strong CMEs is 100% accurate, we can select 50 strong (halo, curved front and most complex) and 50 other (medium and weak) CMEs by searching through a sequence of running differences. A SVM classifier is used to train and test the same group of 100 CMEs based on the aforementioned six features. After training, the SVM classifier is able to classify strong CMEs from others automatically in the testing. The classification rate for the testing group is shown in Table 6.2.

6.5 Comparisons for CME detections

We have developed the software to detect and characterize CMEs. The programs were developed in Interactive Data Language (IDL) by Research Systems, Inc, and run on a

DELL Dimension 4600 PC with CPU time 2.8 Ghz and memory of 512 Mbytes under Linux. The computational time for detecting a CME using three LASCO images is about 5 seconds which is far less than the observational interval. The results in our catalog include a list of CMEs, a sequence of image frames for each CME, the classification type of each CME frame, and the properties of each CME frame, such as height, velocity and angular width. The properties and classification results of CMEs are saved in our database available through our web site.

Previously, there are two CME detection catalogs available to the public. From the LASCO web site at http://cdaw.gsfc.nasa.gov/cme_list/, one can find the CME catalog created by visual inspection. Robbrecht and Berghmans presented their results at http://sidc.oma.be/cactus/. Our CME results are currently shown at http://filament.njit.edu. It is difficult to compare the CME catalogs. The detection rate of CMEs is difficult to determine since there is no comprehensive catalog to be used as a reference to conduct the comparisons Berghmans (2002). A CME in one catalog may be considered as two CMEs in another catalog, and the beginning and ending time for a CME is hard to determine. The reasons are that the preprocessing methods and detection rules are different and human decision is subjective. The reference image and threshold selection may affect the final decision of a CME detection.

We select results between 2004 August 1 and 2004 August 31 to perform comparisons among the three catalogs. The LASCO catalog, used as the reference, listed 65 CMEs in this period. Our catalog missed one weak CME which happened in West 316 in degree on 2004 August 26 at 16:54UT. CACTUS missed three CMEs which happened

LASCO	Strong CMEs	weak and medium CMEs
C2	94%	96%
C3	96%	96%

Table 6.2: Success rates of the strong CME classification and of tge weak and medium CMEs classification based on 50 strong CMEs and 50 non-strong CMEs. We assume the detection by human operators to be 100% success rate.

in East 77 in degree on 2004 August 2 at 23:06UT, East 94 in degree on 2004 August 22 at 17:30UT, and East 111 in degree on 2004 August 26 at 21:54UT. The missed CMEs in our catalog and in CACTUS are weak CMEs. On the other hand, some CMEs missed in LASCO can be detected in CACTUS and our catalog. For example, the CMEs happened in East on 2004 August 6 at 22:30UT, in East on 2004 August 7 at 18:54UT, and in Southeast on 2004 August 9 at 21:30UT. By combining the results of CACTUS and our catalog, the CME detection results are more complete and accurate than the LASCO catalog which is based on human eye detection. Compared with CACTUS which is based on detection of lines, our detection based on regions is easier for computing the properties of CMEs such as height and velocity.

6.6 Summary

An automatic algorithm to detect and categorize Corona Mass Ejections (CMEs) is presented. The preprocessing step intends to normalize the images, remove the missing blocks, and obtain the running differences. Three consecutive images are used to produce two running differences, two binary, and two angular images. The properties of CMEs such as intensity, height, span, and velocity are measured using automatic thresholding and morphology methods. Strong CMEs are distinguished from other kinds of CMEs using the SVM classifier. Compared with other catalogs, our results are proven to be more accurate and efficient for strong CMEs.

CHAPTER 7

SUMMARY AND FUTURE WORK

This dissertation have presented the comparisons of flare detection using the three advanced techniques which are MLP, RBF, and SVM. The SVM classifier is the best for the solar flare detection because it offers the best classification result and the training and testing speed are relatively fast. The second choice is RBF. Because its training data are presented only once, it is the fastest neural network. MLP is not a well-controlled learning machine. The sigmoid function has a scaling factor that affects the quality of the approximation, and the convergence of the gradient-based method is rather slow. The result of MLP is not as good as the RBF and SVM. After the solar flare detection, region growing and boundary-based methods are combined to obtain the detailed properties of solar flares. Moreover, the morphology technique, small part removing and hole filling are used to further improve the performance. Component labeling and model matching techniques are used to characterize the main region of a flare. A motion tracking method is proposed to compute the orientation and speed of two-ribbon flares automatically. The processes of image segmentation and motion tracking take less than 30 seconds for each image and the results are accurate. The proposed automatic process is valuable for the forecast and studies of solar flares since this process dramatically improves efficiency and accuracy. The method allows us to study the evolution of a large number of solar flares efficiently, which will help space weather forecasting.

A new automatic filament detection algorithm based on image enhancement, segmentation, pattern recognition, and mathematical morphology methods has been successfully applied to Hα full-disk filtergrams. For the first time, the SIDE has been adopted to enhancing filaments from the quiet Sun background. Filaments are extracted using an adaptive threshold selection method. The filament segmentation method is efficient and

robust compared to other methods (Gao *et al.* 2002; Shih and Kowalski 2003). The SVM is an essential tool to distinguish between filaments and sunspots using nine features. Filament properties such as foot-point, spine, position, and disappearance are subsequently retrieved by directional morphological closing, thinning, pruning, directional edge linking, and component matching methods.

An automatic algorithm for detection and classification of Corona Mass Ejections (CMEs) is developed. There are three steps in the detection and classification. Normalize the images and remove missing blocks, and use a reference image to produce the running differences. Obtain CME regions using the running differences, and reform the binary segmentation results to angular images. The properties of CMEs such as intensity, height, angle width of span, and speed are obtained using the running differences and angular images. Finally, strong CMEs such as halo, curved front, and complex CMEs are distinguished from other medium and weak CMEs. In this step, SVM classifiers are used to LASCO C2 and C3 images. The input features for the SVM classifiers are the CME properties obtained in the second step.

In the study of solar flares, filaments and CMEs, the image processing and pattern recognition techniques are used to detect and classify the solar features automatically. It is the first time that the SIDE and the SVM are applied for the solar feature enhancement and classification. In addition, the automatic thresholding, region and edge based segmentation, modified edge linking, model matching, and region corresponding methods are used for the solar feature detection. The results of the detection and classification are saved to our database system. The real-time solar feature detection and classification report is posted through http://filament.njit.edu.

In the future research of the classification, numerical analysis can be used to solve the following questions. How to optimize the number of solar features used in classification? What features should we use to perform better results? Are these input features correlated with each other? Are they or a smaller set of features the optimal representation

of solar images in order for necessary classification? Can efficiency of input features be proved theoretically and empirically? In the feature evaluation, Linear Discriminant Analysis (LDA) is a classical statistical approach for classifying samples of unknown classes, based on training samples with known classes (Hakak *et al.* 2001). Another famous approach is Principal Component Analysis (PCA). It involves a mathematical procedure that transforms a number of correlated variables into a smaller number of uncorrelated variables called principal components. The principal components with the larger eigenvalue have the more strong correlation with the outputs. We can use the PCA and/or LDA to find the relationship coefficient between the input features and results.

In addition to feature analysis, the image processing and pattern recognition methods can be used to other solar features such as solar prominences and active regions. The database will store the complete detected data, and the web site will provide users a graphic user interface to access the data and view their analysis figures. The complete catalog of solar features would improve the efficiency and accuracy of solar study, and help in building the relationship among solar events and forecast model of solar events.

REFERENCES

Aizerman, M. A., Braverman, E. M., and Rozeonoer, L. I.: 1964, *Automation and Remote Control*, **25**, 821.

Bashkirov, O. A., Braverman, E. M., and Muchnik, I. B.: 1964, *Automation and Remote Control*, **25**, 629.

Bentley, R. D., Csillaghy, A., and Scholl, I.: 2004, *Proc. SPIE*, **5493**, 170.

Berghmans, D.: 2002, *In: Solar variability: from core to outer frontiers. The 10th European Solar Physics Meeting*, Prague, Czech Republic. Ed. A. Wilson. ESA SP-506, **1**, 85.

Bishop, C. M.: 1995, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, p. 164.

Borda, R. A., Mininni, P. D., Mandrini, C. H., Gómez, D. O., Bauer, O. H., and Rovira, M. G.: 2001, *Solar Phys.*, **206**, 347.

Bratsolis, E. and Sigelle, M.: 1998, *Astron. Astrophys. Suppl. Ser.*, **131**, 371.

Broomhead, D.S. and Lowe, D.: 1988, *Multivariate Functional Interpolation And Adaptive Networks*, Complex Systems, **2:321**, 355.

Brueckner, G. E.; Delaboudiniere, J.-P., Howard, R. A., Paswaters, S. E., St. Cyr, O. C., Schwenn, R., Lamy, P., Simnett, G. M., Thompson, B., and Wang, D.: 1998, *Geophysical Research Letters*, **25**, 15, 3019.

Cane, H. V., Richardson, I. G., and Cyr, O. C. St.: 2000, *Geophysical Research Letters*, **27**, 21, 3591.

Carlowicz, M. and Lopez, R.: 2002, *Storms From the Sun: The Emerging Science of Space Weather*, National Academies Press, Washington, D. C.,

Castleman, K. R.: 1996, *Digital Image Processing*, Prentice Hall.

Dai, Y., Zong, W., and Tang, Y.: 2002, *Chinese Astronomy and Astrophysics*, **26**, 183.

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

Fox, P. A., Middleton, D., Solomon, S. C., Garcia, J., Cinquini, L., and West, P.: 2004, *Am. Geophys. Union*, Fall Meeting 2004, SA54A-07.

Freeland, S. L. and Handy, B. N.: 1998, *Solar Phys.*, **182**, 497.

Gallagher, P. T., Denker, C., Yurchychyn, V., Spirock, T., Qiu, J., Marquette, W. H., Wang, H., and Goode, P. R.: 2001, *Ann. Geophys.*, **20**, 1105.

Gallagher, P. T., Moon, Y. J., Wang, H.: 2002, *Solar Phys.*, **209**, 171.

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

Gavrishchaka, V. V. and Ganguli, S. B.: 2001, *J. Geophys. Res.*, **106**, 29911.

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

Gonzalez, R. C. and Woods, R. E.: 2002, *Digital Image Processing*, Prentice Hall, Upper Saddle River, New Jersey.

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

Gosling, J. T., McComas, D. J., Phillips, J. L., and Bame, S. J.: 1991, *J. Geophys. Res.*, **96**, 7381.

Guyon, I. and Stork, D. G.: 2000, *Linear Discriminant and Support Vector Machine*, MIT Press, Cambridge, Massachussetts.

Hakak, Y., Walker, J. R., Li, C., Wong, W.H., Davis, K. L., Buxbaum, J. D, Haroutunian, V., and Fienberg, A. A: 2001, *Proc. Natl. Acad. Sci.*, **98**, 4746.

Harrison, R. A.: 1994, *Advances in Space Research*, **14**, 23.

Harrison, R. A.: 1995, *Astrophys. J.*, **304**, 585.

Haykin, S.: 1994, in J. Griffin (ed.), *Neural Networks*, Macmillan Publishing Company, NY, USA.

Hill, F., Bogart, R. S., Davey, A., Dimitoglou, G., Gurman, J. B., Hourcle, J. A., Martens, P. C., Suarez-Sola, I., Tian, K., Wampler, S., and Yoshimura, K.: 2004, *Proc. SPIE*, **5493**, 163.

Howard, R. A., Sheeley, N. R., Jr., Michels, D. J., and Koomen, M. J.: 1985, *Journal of Geophysical Research*, **90**, 8173.

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

Jing, J., Qiu, J., Lin, J., Qu, M., Xu Y., and Wang, H.: 2005, *Astrophys. J.*, **620**, 1085.

Jähne, B.: 1997, *Digital Image Processing*, Springer, New York.

Kichenassamy, S.: 1997, *SIAM. J. Appl. Math.*, **57**, 1328.

Looney, G. C.: 1997, *Pattern Recognition Using Neural Networks*, New York Oxford, New York.

Martin, S.F.: 1989, *Journal of Solar Physics*, **121**, 215.

Minoux, M.: 1986, *Mathematical Programming: Theory and Algorithms*, John Wiley and Sons.

Mitra, U. and Poor, H.V.: 1994, *IEEE Journal on Selected Areas in Communications*, **12(9)**, 1460.

Otsu, N.: 1979, *IEEE Trans.Syst. Man. Cyber. SMC*, **9(1)**, 62.

Perona, P. and Malik, J.: 1987, *Proc. of IEEE Computer Soc. Workshop on Computer Vision.*

Perona, P. and Malik, J.: 1990, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629.

Pollak, I., Willsky, A.S. and Krim, H.: 1997, *Proc. of the First Int. Conf. on Scale-Space Theory in Computer Vision*, Utrecht, Netherlands.

Pollak, I.: *PhD Thesis LIDS-TH-2461*, Laboratory for Information and Decision Systems, MIT, August 1999.

Pollak, I., Willsky, A. S. and Krim, H.: 2000, *IEEE Trans. Image Proc.*, **9**, 2.

Pollak, I.: 2002, *IEEE Signal Proc. Mag.*, **19/5**, 26.

Robbrecht, E. and Berghmans, D.: 2004, *A & A*, **425**, 1097.

Qiu, J., Wang, H., Cheng, C.Z., Gary, D.E.: 2004, *Astrophysical Journal*, **604**, 900.

Qu, M., Shih, F. Y., Jing, J., and Wang, H.: 2003, *Solar Phys.*, **217**, 157.

Qu, M., Shih, F. Y., Jing, J. and Wang, H.: 2004, *Solar Phys.*, **222**, 137.

Qu, M., Shih, F. Y., Jing, J. and Wang, H.: 2005, *Solar Phys.*, **228**, 121.

Ripley, B. D.: 1996, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, p. 143.

Shih, F.Y. and Mitchell, O.R.: 1989, *IEEE Trans. Pattern Analysis and Machine Intelligence*, v11, 31.

Shih, F. Y. and Puttagunta, P.: 1995, *IEEE Trans. Image Proc.*, **4**, 1027.

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

Shih, F. Y. and Gaddipati, V.: 2003, *Pattern Recognition*, **36**, 1489.

Shih, F. Y. and Cheng, S.: 2004, *Information Sci.*, **167**, 9.

Shih, F. Y. and Wu, Y.: 2004, *IEEE Trans. Image Processing* **13**, 3, 1078-1091.

Sprecht, D. F.: 1968, *A Practical Technique for Estimating General Regression Surfaces*, Report LMSC-6-79-68-6, Lockheed Missile and Space Co, Inc., Palo Alto, CA.

Steinegger, M., Denker, C., Goode, P. R., Marquette, W. H., Varsik, J., Wang, H., Otruba, W., Freislich, H., Hanslmeier, A., Luo, G., Chen, D., and Zhang, Q.: 2001, *The Solar Cycle and Terrestrial Climate*, A. Wilson (ed.), *ESA SP* **463**, 617.

Sobotka, M., Brandt, P. N., and Simon, G. W.: 1997, *Astron. Astrophys.*, **328**, 682.

Trucco, E. and Verri, A.: 1989, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall.

Turmon, M., Pap, J., and Mukhtar, S.: 2002, *Astrophys. J.*, **568**, 396.

Vapnik, N. V. and Chervonenkis, A. J.: 1974, *Theory of Pattern Recognition*, Nauka, Moscow.

Vapnik, N.V.: 1982, *Estimation of Dependences Based on Empirical Data*, Addendum 1, New York, Springer-Verlag.

Vapnik, V.N. and Chervonenkis, A.: 1991, *Pattern Recognition and Image Analysis*, 1(3), p. 283.

Vapnik, N.V.: 1995, *The Nature of Statistical Learning Theory*, Springer, New York, p. 126.

Vapnik, N. V.: 1998, in: S. Haykin (ed.), *Statistical Learning Theory*, John Wiley & Sons, New York.

Veronig, A., Steinegger, M., Otruba, W., Hanslmeier, A., Messerotti, M., Temmer, M., Brunner, G., and Gonzi, S.: 2000, ESA. **463**, 455.

Wang, H., Qu, M., Shih, F.Y., Denker, C., Gerbessiotis, A., Lofdahl, M., Rees, D., Keller, C.: 2004, *204th American Astronomical Society Meeting*, Denver, CO.

Wang, H.: 2005, *Astrophysical Journal*, **618**, 1012.

Webb, D. F., Lepping, R. P., Burlaga, L. F., DeForest, C. E., Larson, D. E., Martin, S. F., Plunkett, S. P., and Rust, D. M.: 2000, *Journal of Geophysical Research*, **105**, 27251.

Weickert, J., Ishikawa, S. and Imiya, A.: 1997, *Gaussian Scale-Space Theory*, Kluwer, Germany.

Widrow, B., and Hoff, M. E., Jr.: 1960, *Adaptive switching circuits, in 1960 IRE WESCON Convention Record*, Part 4, New York: IRE, p. 96.

Yurchyshyn, V. B., Wang, H., Qiu, J., Goode, P. R. and Abramenko, V. I.: 2000, *Astrophysical Journal*, 540.

Zaknich, A.: 2003, *Neural Networks for Intelligent Signal Processing*, World Scientific Publishing Co. Pte. Ltd, Singapore.

Zirin, H.: 1988, *Astrophysics of the Sun*, Cambridge University Press, Cambridge.