

Detection and Tracking of Coronal Mass Ejections (CMEs) by Means of the Watershed Segmentation and Hough Transform

Mariusz Nieniewski *

Abstract—Detection and tracking of CMEs is an important issue in solar physics. The appropriate images are continuously collected and parameters describing the extent, position, and time of occurrence of the CMEs are put in a catalogue. Because of the huge amount of data, it is imperative to process these images automatically. One of the best techniques for object extraction is the watershed segmentation. It works well for individual images. It is shown in this paper how this technique can be extended to a sequence of images showing the temporal development of the CMEs. The detection and tracking of the CME is done automatically via the Hough transform which points to at least some pixels located on the front of the moving CME. An approach has been developed which allows one to obtain the initial masks of the CMEs and subsequently merge the regions obtained from the watershed segmentation into the full masks of the CMEs so that in response to the sequence of CME input images a sequence of masks is generated.

Keywords: *watershed segmentation, Hough transform, region merging, CMEs detection and tracking, motion detection and tracking*

1 Introduction

CMEs are of major concern for research in astrophysics since they expel huge masses from the Sun which propagate outwards through the heliosphere and often interact with the Earth's magnetosphere [1]. In the following only a few basic facts concerning the CME images are given, indispensable for a conscious analysis of sequences of these images.

The images of the CMEs are routinely obtained from the Large Angle Spectrometric Coronagraph (LASCO) mounted on the board of the Solar and Heliospheric Observatory (SOHO). Details regarding the satellite SOHO mission as well as the results obtained from the LASCO experiment are described in [2]-[5]. The LASCO instru-

ment consists of one internally occulted coronagraph, C1, and two externally occulted coronagraphs, C2 and C3. The C1 coronagraph observes the corona from about 1.1 to 3 R_{\odot} , where R_{\odot} is the radius of the Sun and uses the wavelength of the Fe XIV emission line. This paper is concerned with two other coronagraphs C2 and C3. The field of view of C2 extends from about 1.5 to 6 R_{\odot} , the inner radius of this field being determined by the occulter mounted in front of the camera. The field of view of C3 similarly extends from 4 to 30 R_{\odot} . The signal in C2 and C3 is formed by a broad-band visible photospheric light scattered by electrons in the corona. The classical CME has a 'light bulb' structure and consists of three parts: a bright coronal loop that is a front part of the CME, a darker void, and an eruptive prominence closer to the surface of the Sun.

The CME events are being continuously catalogued and are available for inspection at the following web site http://cdaw.gsfc.nasa.gov/CME_list. The catalogue includes major data on the CMEs, such as the time of occurrence, angular span, velocity, acceleration. Strictly speaking, the appropriate C2 and C3 images should be merged into a single image in order to observe a time evolution of the CME in a wider range of distances from the center of the Sun [6]. However, this is a secondary technical problem, which is outside the scope of this paper, and we use C3 images as an example. In principle there should be no problem with using the proposed approach for a larger image in which data from C2 and C3 are combined.

Up to now a couple of papers have been published on the detection and tracking of the CMEs [6]-[8] – mainly from the point of view of astrophysics. At the same time this problem can also be treated as an image processing task in its own right.

The purpose of the research described in this paper is to find the way of obtaining the mask of the moving CME by means of the watershed segmentation. The watershed segmentation of separate images does not relate directly to the motion of the CMEs. The motion of individual pixels in time is detected by the Hough transform in the auxiliary time-height (also called time-radius) planes. The

*Institute of Fundamental Technological Research, Warsaw, Poland 00049. Phone: 48-22-8261281, ext. 321, Fax: 48-22-8269815. Email: mnieniew@ippt.gov.pl The research described was supported by the grant from the (Polish) Ministry of Science and Higher Education. The solar images were obtained by courtesy of NASA.

two kinds of information: from the watershed segmentation and from the Hough transform are put together with the aim of receiving a complete mask of a CME developing in time. This paper proposes the watershed segmentation for the detection and tracking of the CME, where the process of watershed region merging is initiated by the Hough transform and subsequently continued in such a way that a mask of the CME is formed by a connected set of pixels differing in brightness from the neighborhood. Once this mask has been obtained, the appropriate statistics of the CME can be calculated.

It seems quite possible that the general idea of motion tracking presented in this paper may also be used in other applications, for example where motion of poorly defined objects is of concern.

2 Watershed Segmentation

An example of the CME image obtained from the catalogue is shown in Fig. 1(a). This is an image of the running difference, that is the difference between the current image and the immediately previous one. In this way, changes from image to image are clearly visible. The running difference images are of size 512×512 pixels and have 11 gray levels uniformly dispersed between 0 and 255. The image in Fig. 1(a) is extremely noisy, primarily due to cosmic rays hitting the camera. The result is white pixels or longer white lines. We would like to use the watershed segmentation for extraction of the masks of the CMEs. The main reason for this is that the watershed algorithm is a largely recognized method of image segmentation [9], [10], which is attractive since it often gives regions with borders nicely fitting the expected contours of the objects in the image. Probably the most comprehensive review of the watershed algorithms is given in [11]. In the following only a very short description is included. The gray value in an image pixel can be regarded as the elevation above some reference level. In this way we can consider an image as a topographical map. A higher gray level corresponds to a higher elevation. Each drop of the rain falling on such a terrain flows down to some catchment basin. All pixels for which water accumulates in a common basin form a watershed region, and the borders between regions are called watershed lines. Most algorithms actually simulate flooding the terrain from the bottom instead by a rain falling from above, but in principle the results are similar.

In our case, the watershed segmentation of the image in Fig. 1(a) cannot give meaningful results since there would be too many regions due to excessive number of noisy pixels. In order to reduce noise, the low pass Gaussian filtering is carried out following the procedure presented in [12]. In particular, in [12] several Matlab functions are given for calculating the Discrete Fourier Transform (DFT) of the image, Gaussian filtering in the frequency

domain, and the inverse DFT. The chosen Gaussian filter has the cutoff frequency equal to 60. This frequency gives a reasonably smooth image shown in Fig. 1(b), in which the shape of the CME is maintained and noise is removed to a large extent. In order to carry out the watershed segmentation we follow the usual procedure and first calculate the morphological gradient of the image in Fig. 1(b). This gradient is obtained using a flat structuring element of size 3×3 pixels. The watershed segmentation is carried out using the function available in the Image Processing Toolbox in Matlab. The result of the watershed segmentation of the filtered image is shown in Fig. 1(c). Fig. 1(c) demonstrates that the segmentation of an individual image can be carried out correctly; however, there are many regions and they have to be clustered, or merged, appropriately in order to obtain a meaningful mask of the CME. Furthermore, the results of the segmentation for successive images have to be adjusted so that a sequence of CME masks and the sequence of original running difference images would be comparable. The task of finding the appropriate pixels on the moving front of the CME for successive images of the running difference is carried out by the Hough transform.

3 Hough Transform for Detecting Straight Lines in Time-Height Plane

In preparation to the calculation of the Hough transform we transform the original images to polar coordinates. The original running difference images exhibit a circular structure. There is an invisible center of the Sun, and the CMEs move away from this center approximately along the radii in all spatial directions. The result observed is a sequence of images shown in the left column of Fig. 5 below. The rectangular coordinate system for the original images is (x, y, t) , where (x, y) are spatial coordinates in the image plane, and t denotes time. For each (x, y, t) point there is some gray level in the image. We call the collection of the original images an (x, y, t) data box.

Taking the point corresponding to the center of the Sun in the image as the origin we introduce a polar coordinate system in which r is the distance from the origin and ϕ is the angle. The point (x, y, t) can then be recalculated to (r, ϕ, t) . In this way we obtain a new (r, ϕ, t) data box, corresponding to the original (x, y, t) data box. A program for coordinate transformation is written in Matlab. The result of the transformation of Fig. 1(b) into the polar coordinate system is shown in Fig. 2. This is an image of size 215×360 pixels. The angle ϕ , measured in degrees, is shown along the horizontal axis. The radius r , measured in pixels is shown along the vertical axis. 256 integer values of the radius have been assumed, but the first 41 lines in Fig. 2 have been removed since the gray values of the pixels covering the occulter are irrelevant. The row coordinate of the Sun center in the Matlab coordinates system is assumed 230, and the col-

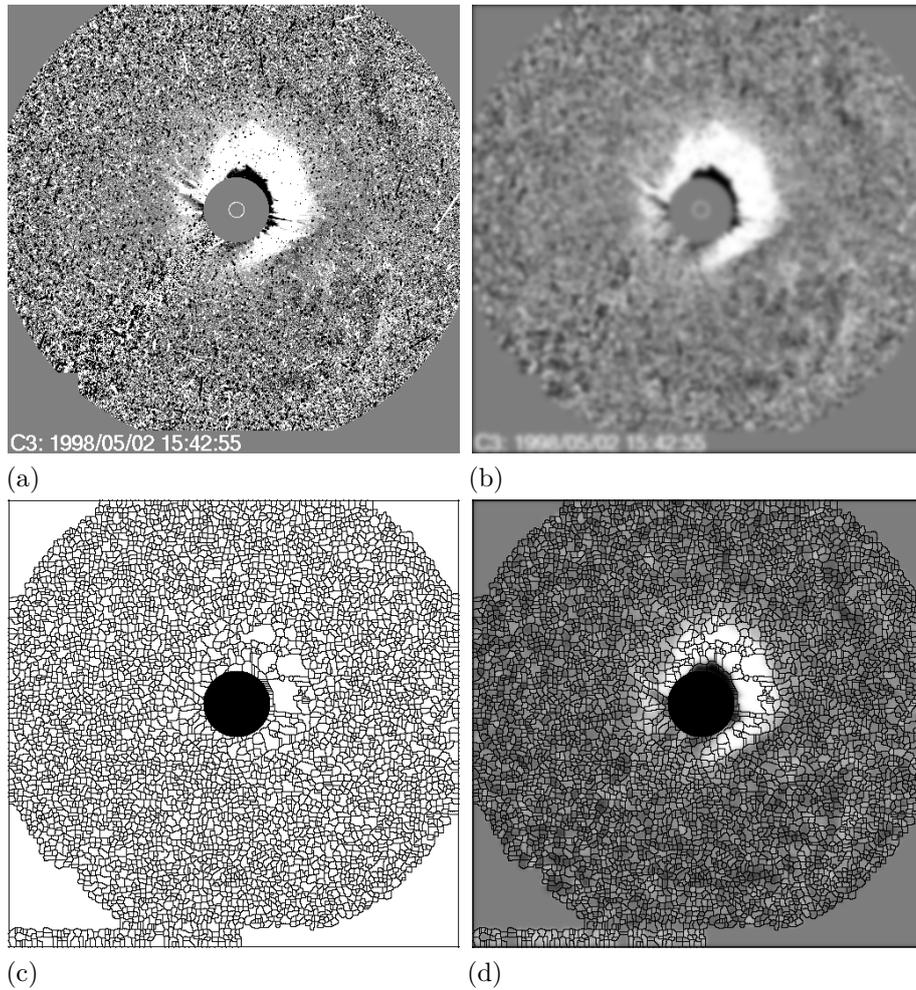


Figure 1: (a) Original running difference image. (b) Filtered image. (c) Watershed segmentation of the gradient of the filtered image. (d) Superposition of the watershed lines on the filtered image.

umn coordinate is 260. The program uses the bicubic interpolation for calculation of the gray level at a given (r, ϕ) point. The angles are measured counterclockwise starting from the horizontal axis.

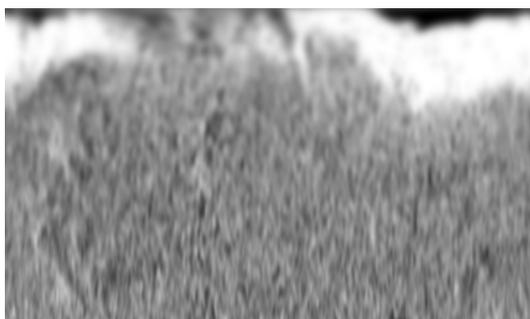


Figure 2: (a) Filtered image from Fig. 1(b) after transformation to polar coordinates.

The (r, ϕ, t) data box in the example described in this paper contains 24 images corresponding to images available for the date of 1998/05/02 with the time distance

between successive image equal to approx. 1 hour. Intersecting the (r, ϕ, t) data box with the plane $\phi = \text{const.}$ we obtain a set of 360 (r, t) images of size 215×24 pixels. The plane (r, t) is called the time-height plane.

An example of a set of time-height planes is shown in Fig. 3(a), which depicts the images obtained for $\phi = 301, 311, \dots, 351^\circ$ for the data box under consideration. The gray-level images in Fig. 3(a) illustrate the motion of the bright CME front as a function of time, with the time growing horizontally to the right and the height growing downwards. The images in Fig. 3(a) are thresholded with a threshold equal to 220. This value was chosen experimentally and strikes a balance between high sensitivity of detecting the CMEs and susceptibility to noise. However, it was found that changing the threshold in a relatively wide range practically does not influence the results. The obtained binary images are shown in Fig. 3(b). The Hough transform is now applied to the images in Fig. 3(b).

The Hough transform is well described in textbooks [12]

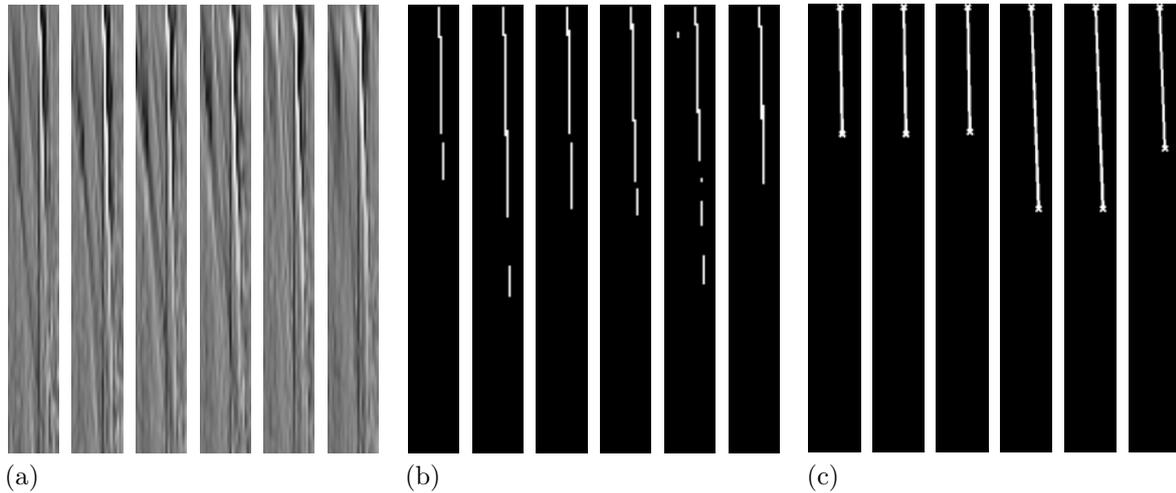


Figure 3: (a) Original time-height image. (b) Thresholding of (a). (c) Straight line segments detected by the Hough transform.

and only a general idea is mentioned here. Given a set of points in a binary image, we want to find a subset of points which lie on straight lines. Each point with spatial coordinates (x, y) in the image plane determines a large number of straight lines which pass through this point. It is convenient to describe these lines in terms of the parameters (ρ, θ) specifying the offset and the angle of a line. For a large number of points lying on the same straight line we obtain a large number of votes for the same (ρ, θ) coming from individual points. The maxima of votes determine the parameters of the detected straight lines. The points dispersed all over the plane and not forming straight lines do not contribute significantly to the maxima.

The straight line segments in images in Fig. 3(b) are found using the functions: *hough*, *houghpeaks*, *houghpixels*, and *houghlines* described in [12]. Most of these functions use default parameters whenever possible, except that *houghpeaks* admits 10 peaks to be detected. Examples of the lines found for the images in Fig. 3(b) are shown in Fig. 3(c). For our purposes, a graphical representation is less important. The most critical output is an array which for all 360 values of ϕ specifies the following items: (a) the number of detected straight-line segments for a given ϕ , (b) the offset ρ and the angle ϕ for each straight-line segment, (c) coordinates of the end points of the segments. This information is subsequently used for the generation of another array which for each angle ϕ specifies the (integer) coordinates of the points of intersection of each segment with the vertical lines corresponding to successive values of time t in Fig. 3(c). The obtained points of intersection can be collected for any value of time and shown in images such as in Fig. 4(b).

The results from the Hough transform are subsequently combined with the watershed segmentation previously described.

4 Region Merging for Generation of the Masks of CMEs

As shown in Fig. 4(b) the Hough transform indicates where the CME is, but it hardly gives any information on the actual shape of the CME. It can be observed in Fig. 1(d) that by appropriately merging the watershed regions one could obtain the shape of the CME nicely agreeing with human expectation. The region merging has been a subject of several papers [13], [14]. The approach we propose for the CME detection and tracking is as follows.

Based on the image of watershed lines obtained as described above we generate two arrays (a C++ program doing this is mentioned in [13]). One of the arrays contains information on the pixels in the image, that is the coordinates of all pixels together with the labels (region numbers) assigned to these pixels. The black watershed lines in Fig. 1(c) have a special label which is not used for any watershed region. In Fig. 1(c) a slight modification of the watershed result is shown since all pixels covering the occulter have been assigned to watershed lines. This has been done in order to exclude the area of the occulter from calculations. The functions described below process exclusively white pixels or their closest black neighbors. The other array contains the information on all the regions in the image, that is the list of labels (region numbers) together with the area and the average gray value of each region.

The first step in combining the information from the Hough transform and the watershed segmentation is to generate the so-called seed pixels. Comparing Fig. 4(b) with the pixel array mentioned in the previous paragraph we assign the labels to the pixels indicated by the Hough transform. As a result we obtain an array containing pixel coordinates together with assigned labels. Some of

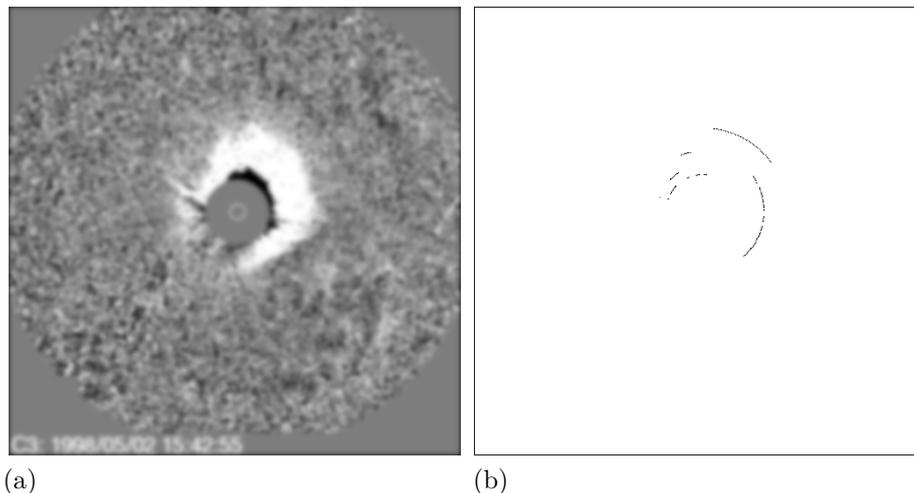


Figure 4: (a) Repetition of Fig. 1(b). (b) The corresponding result obtained from the Hough transform.

the labels are repeated a number of times because quite often multiple pixels indicate the same region. In order to avoid multiple entries the above array is reduced so that each region appears only once with a valid pixel chosen arbitrarily. These calculations are carried out by a function written in Matlab. In addition to generating the seed pixels, this function generates the initial masks, that is masks covering the regions indicated by the seed pixels. An example of initial masks is shown in the middle column in Fig. 5.

The second step is the actual region merging carried out for individual initial masks. For this purpose a program in C++ has been written based on the one described in [13]. This program is used together with several auxiliary functions written in Matlab for processing a large number of subdirectories created under way. Each initial mask is processed separately. Because the Hough transform shoots inaccurately, it may happen that the indicated region is too dark to be included in the mask of a CME. It is assumed that if the average gray value of the indicated region is below the threshold value 180, then the appropriate initial mask is excluded from further computations. The value of 180 was chosen experimentally, but in any case the threshold can be changed in a large range without modification of the results. Assuming that the initial mask has been accepted, the following operations are carried out. We find all neighboring regions of the initial mask. These neighbors are sorted according to the decreasing average gray value and are merged with the initial mask region as long as their gray values are higher or equal to the average gray value of that region. There is a finite number of such regions and the computations terminate in a relatively short time. We call the mask obtained in this way the intermediate mask. The result of the current step is a set of intermediate masks, each mask obtained for an initial mask. However, some of the intermediate masks may partly overlap.

The third step is the generation of the final masks. The intermediate masks are put together by the operation of logical OR, forming the final mask. An example of the final masks is shown in the right column in Fig. 5.

The masks presented in Figs. 5(a)-(s) were obtained for the date 1998/05/02. One of the running difference images (Fig. 5(e)) is missing in the catalogue for that day and was reconstructed by interpolation. There are only 19 masks shown in Fig. 5 since no CMEs were detected in the last five running difference images for that day. It can be observed that some final masks do not include the initial masks (compare Fig. 5(d)). The reason for this is that the appropriate initial mask has been excluded because a region it covers is too dark.

It may be of interest to look at some numbers related to the conducted experiments. For the sequence in Fig. 5, the maximum number of seed points (and initial masks) is 35 and is obtained for Fig. 5(q). The maximum number of iterations required to obtain an intermediate mask for this image is 24. However, these are extreme numbers. In most cases the number of seed pixels, and particularly the number of iterations are considerably smaller. Obviously, the largest numbers are obtained for the halo CMEs, that is for CMEs surrounding the whole (or almost whole) solar disk [5]. The variable number of seed pixels and of the iterations leading to the intermediate masks, as well as the variable number of intermediate masks contribute to the considerable overhead in the functions performing the calculations. A minor improvement can be made by adding an extra function that provides for inclusion of a few missed regions to the final mask. In particular, in case of Figs. 5(p) - (r) one or two regions neighboring the final mask and having the average gray value higher than that of the final mask were included in the mask. Using a more sophisticated approach to region merging, for example such as based on the maximization of the

contrast between the CME region and the surroundings [13] would lead, in fully automatic CME processing, to excessive programming burden. Similarly, removal of the overlap of intermediate masks would be rather complicated.

5 Conclusions and Future Work

The presented method of CME detection and tracking is fully automatic. A considerable part of the software described includes creation and manipulation of various directories with variable number of images. No attempt was made to make this software homogeneous because more important was the use of the available functions.

The proposed method could be extended to extraction of vaguely defined or diffuse objects, for example explosions or blasts appearing across a sequence of images.

References

- [1] Webb, D. F., "Coronal Mass Ejections: Origins, Evolution, and Role in Space Weather," *IEEE Transactions on Plasma Science*, V28, N6, pp. 1795-1806, 2000.
- [2] Moses, D., et al., "EIT Observations of the Extreme Ultraviolet Sun," in *The First Results from SOHO*, edits. B. Fleck and Z. Švestka, Kluwer, pp. 571-599, 1989. Reprinted from *Solar Physics*, V170, N1, 1997.
- [3] Dere, K. P., et al., "EIT and LASCO Observations of the Initiation of a Coronal Mass Ejection," in *The First Results from SOHO*, edits. B. Fleck and Z. Švestka, Kluwer, pp. 601-612, 1989. Reprinted from *Solar Physics*, V170, N1, 1997.
- [4] Webb, D. F., "CMEs and the Solar Cycle Variation in Their Geoeffectiveness," *Proc. SOHO 11 Symposium, 'From Solar Min to Max: Half a Solar Cycle with SOHO'*, Davos, Switzerland, pp. 409-419, 3/02 (ESA SP-508).
- [5] Michalek, G., Gopalswamy, N., Yashiro S., "A New Possibility to Estimate the Width, Source Location, and Velocity of Halo CMEs," *Proc. SOHO 11 Symposium, 'From Solar Min to Max: Half a Solar Cycle with SOHO'*, Davos, Switzerland, pp. 453-456, 3/02 (ESA SP-508).
- [6] Berghmans, D., Foing, B. H., Fleck, B. "Automatic Detection of CMES in LASCO Data," *Proc. SOHO 11 Symposium, 'From Solar Min to Max: Half a Solar Cycle with SOHO'*, Davos, Switzerland, pp. 437-440, 3/02 (ESA SP-508).
- [7] Robbrecht, E., Berghmans, D., "Automated Recognition of Coronal Mass Ejections (CMEs) in Near-Real-Time Data," *Astronomy and Astrophysics*, V425, pp. 1097-1106, 2004.
- [8] Robbrecht, E., Berghmans, D., "A Broad Perspective on Automated CME Tracking: Towards Higher Level Space Weather Forecasting," in *Solar Eruptions and Energetic Particles*, edits. N. Gopalswamy et al., Geophysical Monograph Series, V165, American Geophysical Union, 2006.
- [9] Soille, P., *Morphological Image Analysis*, Springer, 2004.
- [10] Frucci M., Longo G., "Watershed Transform and the Segmentation of Astronomical Images," *Astronomical Data Analysis III, Electronic Workshops in Computing - eWiC*, The British Computer Society, 2004.
- [11] Roerdink, J. B. T. M., Meister, A., "The Watershed Transform: Definitions, Algorithms, and Parallelization Strategies," *Fundamenta Informaticae*, V41, pp. 187-228, 2000.
- [12] Gonzalez, R. C., Woods, R. E., Eddins, S. L., *Digital Image Processing Using MATLAB*, Pearson Prentice Hall, 2004.
- [13] Nieniewski, M., "Extraction of Diffuse Objects From Images by Means of Watershed and Region Merging: Example of Solar Images," *IEEE Transactions on Systems, Man, and Cybernetics*, B, V34, N1, pp. 796-801, 2004.
- [14] Hansen, M. W., Higgins, W. E., "Watershed-Based Maximum Homogeneity Filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, V8, N7, pp. 982-988, 1999.

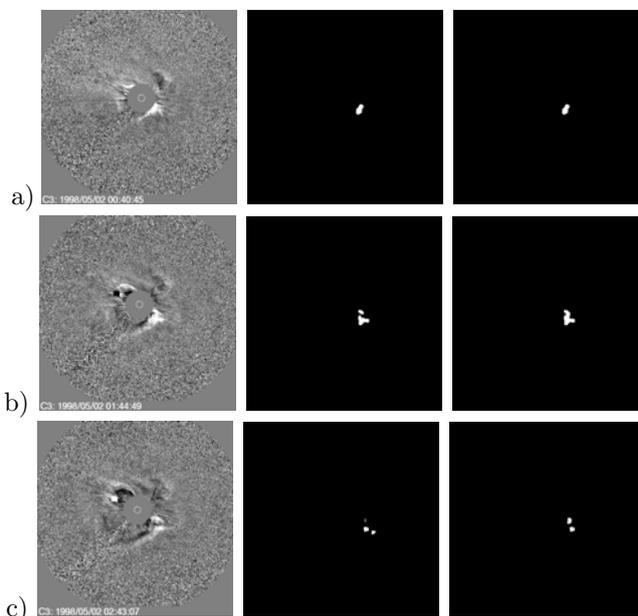


Figure 5: Original difference images, initial masks, and final masks of the CMEs (to be continued).

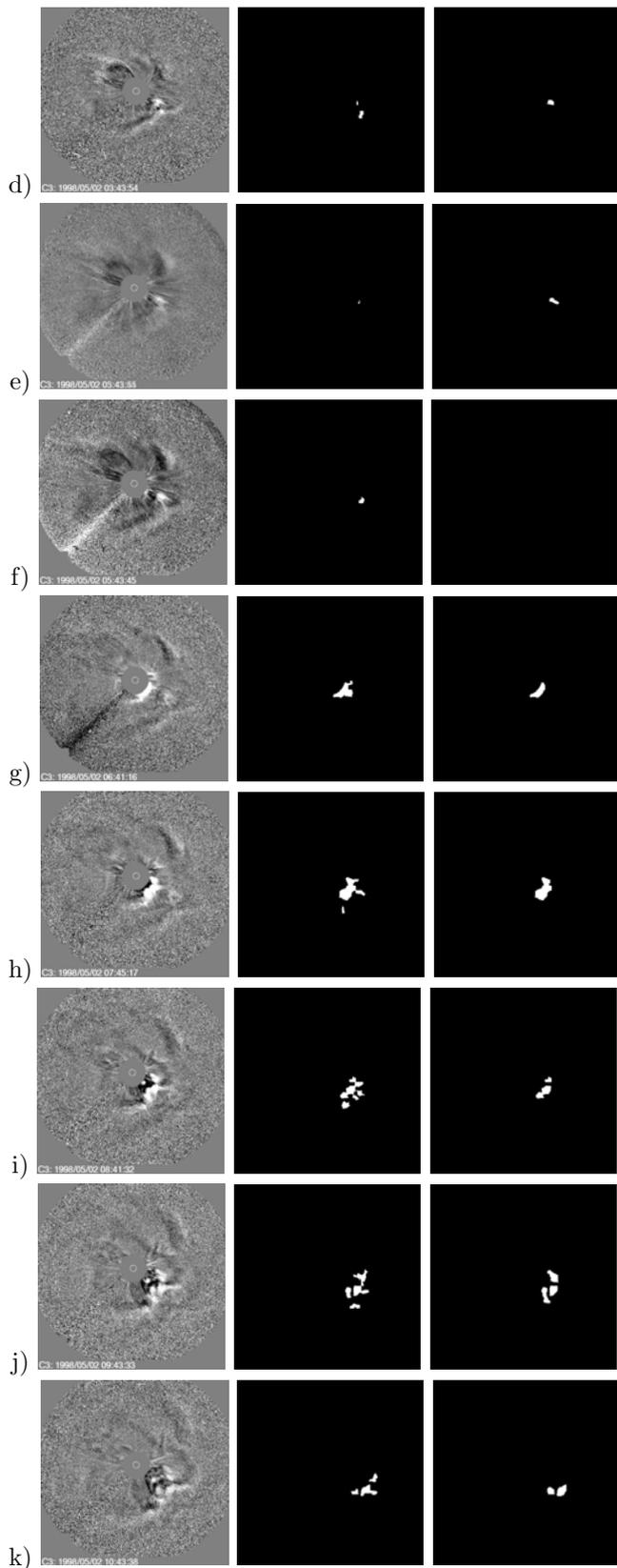


Figure 5: (continuation) Original difference images, initial masks, and final masks of the CMEs.

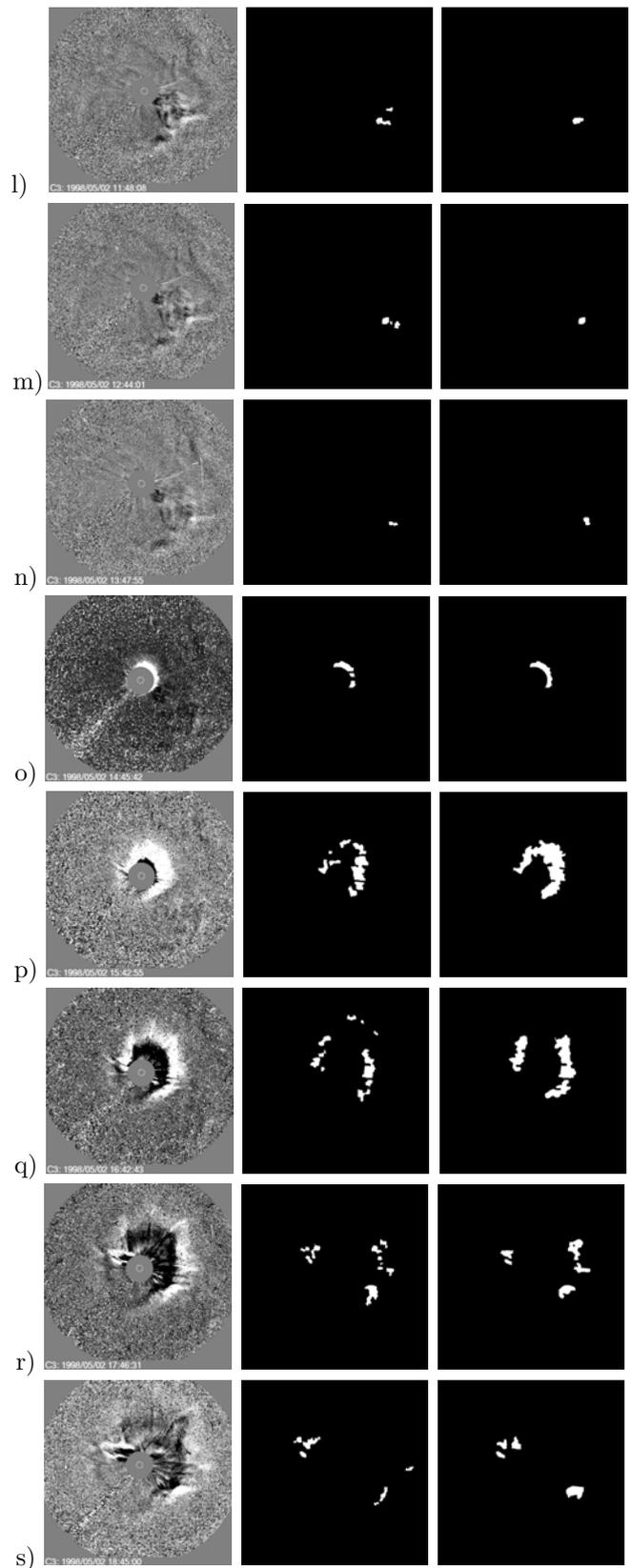


Figure 5: (continuation) Original difference images, initial masks, and final masks of the CMEs.