The cmeDetect Computer Code for CME Analysis

Ahmed A. Selman and Zeinab F. Hussein

Abstract—In the present work, computer detection code is introduced to isolate CME from SOHO/LASCO C3 images utilizing a Matlab code. The detection program, cmeDetect, contained various functions with detection method based on bulk detection of CMEs. Measurements included height, speed, acceleration, area, masse, and direction. This paper is dedicated to outline the cmeDetect code with its initial results. The code has many subfunctions one of which is that used for corrections of the solar size from LASCO C3 images. The first results showed that most height results were satisfactory but their differences, which were used in speed and acceleration, were fluctuating.

Keywords – Coronal Mass Ejections, Matlab Computer Code, Solar and Space Physics.

1 Introduction

Coronal Mass Ejection (CME) is an interesting phenomenon in both scientific and technological areas since it reveals an important solar activity. In particular it is thought to remove built-up magnetic energy and plasma from the solar corona[1]. It is defined as a large eruption of a hot plasma mass confined with magnetic field from the corona of the Sun. Various shapes are found for CMEs while they emerge as dense objects leaving the Sun. The mass can reach 10^{15} to 10^{16} gm with speed that can reach up to 3000 km/sec [2]. Earlier work of Mc Queen et.al. [3,4] and later Harrison [5] discussed CME transients and the availability to distinguish transient driving mechanisms, and the urgent need to develop CME detection and analysis supported by theoretical models. Gosling [6] pointed out variety of solar wind observations are consistent with the concept of sustained 3-dimensional reconnection within the magnetic legs of CMEs close to the Sun. Then in 2000, David [7] reviewed some of the well-determined coronal properties of CMEs, what we know about their source regions, and what their manifestations are in the solar wind. One exciting new type of observation is of halo-like CMEs, which suggest the launch of a geo-effective disturbance toward earth. Andrews and Howard [8] pointed out that the causes and origins of CMEs remain among the outstanding questions in space physics. Developments of CME detection and origin continued by Wu et al. [9], and Webb et al. [10-12] and others [13,14] using various techniques that extended from MHD simulation models to analyzing coronagraph data only or utilizing other SOHO detectors—see ref. [1] for the full survey.

There is an evident relation of CME to the important solar magnetic field development mechanism since it is the dominating factor in their shape. For example, a recent study of the solar magnetic field showed that at regions about 5 Rs the poloidal component of the solar magnetic field decreases to about 50% [15].

In such studies, the use of near real-time data has a significance. SOHO is ideal for this task since it holds 12 functional detectors some of which have more than a single band of solar activity detection. The Large Angle Spectrometric Coronagraph (LASCO) is a wide-field white light and spectrometric coronagraph consisting of three optical systems having nested fields of view (FOV) [2]. The three optical systems are named C1 (FOV 1.1 to 3.0 solar radius, Rs), C2 (2.0 to 6.0 Rs) and C3 (3.7 to 32 Rs), with C1 being the smallest-although this particular detector stopped functioning since 1998. LASCO C3 instrument was the main source of data in the present work.

2 Aim

In this paper, the aim was to write a computer code that automatically detects CMEs from SOHO/LASCO C3, 512^2 pixel jpeg images, using Matlab. After detection was made, the required code must be able to measure the basic properties of these CMEs, namely: their height, time of evolution, speed, acceleration, total area and direction. The results are then to be compared with the well-known CME catalog library [16].

3 Numerical Method

Few image processing techniques were used in the cmeDetect code. The code cmeDetect, written using Matlab program ver. 7.8 (R 2009 a). It uses many subroutines and makes use of few Matlab built-in commands such as cell data type, region properties function (regionprops) and black-white boundary trace function (bwtraceboundary), among other functions. See the next paragraph for details of this code and the flowchart.

4 Outline, Results and Discussions of cmeDetect

In general, this code uses colored images as its input, provided from the library of SOHO/LASCO C3 coronagraph with resolution 512 x 512 pixels. Then the

- *Ahmed A. Selman is currently at the Department of Space and Astronomy, College of Science, University of Baghdad, Baghdad-Iraq, E-mail: aaselman@scbaghdad.edu.iq*
- *Zeinab F. Hussein is currently at the Department of Space and Astronomy, College of Science, University of Baghdad, Baghdad-Iraq,*

ISSN 2229-5518

program performs the process of image conversion to grayscale utilizing an automatic thresholding procedure that is based on the information obtained from the same image. This conversion with thresholding is performed using the Matlab function (rgb2gray). The conformation results give indication about image quality and file arrangement, and the preparation task provides determining actual solar position and size. After that, another conversion is made from grayscale to logic (black-white) scale using (im2bw) function.

In order to grip and maintain as much useful information, black-white images are converted first to grayscale images. The threshold value was computed from a global level in each image. The global level is assumed as a normalized intensity value that lies within the range (0, 1). This was the only normalization made in this work.

It should be mentioned here that the present work is based on a code that does not perform image normalization based on accumulation time (which is calculated from the shutter speed of the camera of C3 coronagraph). This is mainly because the images were collected from the SOHO/LASCO's website image library in the form of (jpg) format, rather than from the raw images one.

Beside this, a reference image was chosen from the LASCO C3 library (with the same size as mentioned above). This reference image was chosen such that it contains no CME and the minimal possible background noise. Such a reference image is to be used for two reasons in the main calculating code:

1-To detect the actual size of the Sun.

2-To be considered as a reference for background initial elimination in other images when a CME is detected.

In the present work the CME was determined from its main bulk mass region as appeared in LASCO images and its properties were measured from the motion of its center of mass, as described below. Although the present method may bring considerable error in few cases, it is an method that can be programmed with relative ease. Error generates mainly due to the inhomogeneous nature of CMEs. The main body of the CME may become more dense than its boundaries because CMEs usually hover out of the Sun as a plasma with large mass and turbulent density. Thus, CME main body usually becomes more dense than its leading and following boundaries. Detecting the CME based on its main bulk mass may lead to ignoring the leading boundary edge and this may accumulate the error of calculations. This problem was approximately corrected in the present code by detecting the major axis length—see below.

Any detected bulk is assumed as a region of interest for the code (ROI). Then ROI that represents the possible CME event is traced, and its various properties are calculated. Calculations include height-time profile for each CME, speed, acceleration, main position angle and mean area. In order to perform these measurements from each image, few parameters must be found first such as the time of each image and the solar size. Solar size is measured first in pixels and the height-time profile is compared with the solar size which was previously determined in the same code. The code flowchart is given in Figure (1).

4.1 Measuring the Solar Radius

Since used images were with resolution (512 x 512), then it would be expected that the center of the Sun lies at the center of the image, i.e. at pixel (256,256). However, this was not the case. From careful inspection of any image from LASCO of the year 2002 using the Matlab image tool (imtool) it was found that the center of the Sun lies exactly at x=247, y=260 pixels; with intrinsic (measuring) error of ± 1 pixels for both dimensions as given in Figure (2). In this figure, the center of any image from LASCO is magnified by the ratio 1600%. The *green semi-circle* is the detected boarder of the white one. This shape was detected using the main program using the boundary trace function (bwtraceboundary) as described below. This time the radius of the detected (green) circle is calculated from the previous step as well as the center of the green circle (xc, yc), then data were regenerated using the equation of the circle:

ISSN 2229-5518

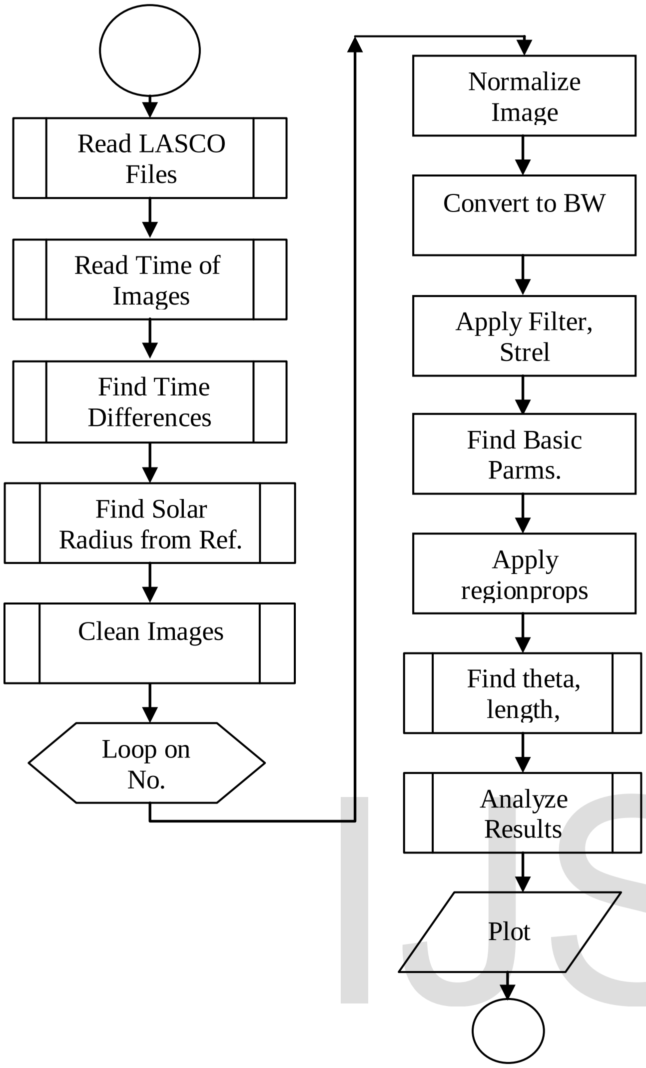

Figure (1). The flowchart of the code cmeDetect.

$(x+xc)^2+(y+yc)^2=r^2$, where r is the radius. The *red crossed lines* are the lines representing the center of the Sun, found at xc=247, yc=260 pixels. The *yellow crossed lines* are the lines plotted exactly at the center of the image, x=256, y=256 pixels. Only the white circle exists in LASCO images, while all other (colored) shapes were plotted using the Matlab code.The detection of the white circle shape was made by detecting any continuous shape near the center of the image, i.e., around the pixel point (256,256). After the program senses this shape it starts to trace the outer boundaries of it. Here, the main program uses a function found in Matlab called the boundary trace function (bwtraceboundary). This function is able to determine the geometrical properties of any continuously connected bulk shape in a black-white image. In the case when this shape is a rough circle, this function calculates the circle's center and radius. These information were not directly used in the program, but they were used to perform a simple curve fit to find the parameters of a smooth circle that is concentric with the original white circle and with the same radius. The fitted circle is shown in Figure (4-1) as the thin red circle laying at about the boarders of the original white circle.

The error (err) in determining the position (xc, yc) is obvious from the difference between the yellow dot at the center and the point of the red-crossed lines. That is ~ 1 pixel, and it exists due to the fact that points must occupy at least a single pixel, and during the calculation the center (yellow dot) is at (xc, yc) while the lines cross at (xc+err, yc+err). Although the error seen in this is apparently less than 1 pixel, but it is assumed as (1 pixel) since this is the least possible error value in determining a single point in an image.

The solar radius is measured from the reference image and it can be assumed that its size does not change in other images-see the next paragraph. The solar radius is an important parameter used during the calculations performed by the code, since this code aims to detect CME different spatial features based on their distances and sizes compared to the solar size. Thus, the solar size is first detected in the code to find its value in pixels. Then, since the solar diameter is previously well known, one can calculate the conversion parameters from pixels to kilometers (or to meters, or even solar radii). After this any measurement that is made by the code is easily converted into distance units as required. This is shown in the example of Figure (3-a and b).

Regarding the pervious assumption about fixed measured solar radius, however, it was found that selecting other reference images resulted in values that were as large as 9.9707 pixels. Selecting few images for this procedure showed that no radius values were less than 9.6700 pixels, and no values were higher than 9.9707 pixels. Therefore, the error assumed in this part is between 3.11 % to 3.02 %. The mean value is assumed as 9.8204 pixels, and it is the value used in the rest of this work. This value comes with a round-off error 1.53% from both maximum and minimum detected solar radius values. This error value is small enough to be neglected, therefore in practical applications the previous assumption about fixed measured radius can be considered. However, in this work, this error will not be neglected, and the mean value of 9.8204 pixels will be used to represent the solar radius with error ±1.53%.

ISSN 2229-5518

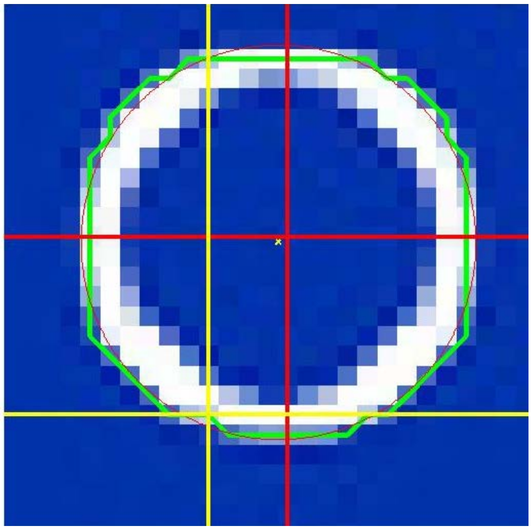

Figure (2). The center of any image from LASCO, magnified 1600%. The white circle exists originally in all LASCO images and this represents the actual circumference of the Sun. The green semi-circle is the detected boarder of the white one from the main program. The thin red circle is the perfect circle plotted from the green semi-circle smoothed points. The red crossed lines are the lines representing the center of the Sun, found at x=247, y=260 pixels. The yellow crossed lines are the lines plotted exactly at the center of the image, x=256, y=256 pixels. Only the white circle exists in LASCO images, while all other (colored) shapes were plotted using the Matlab code. Furthermore, the diameter of this circle was also roughly measured using the Matlab's (imtool), as seen in Figure (3-c). The shown value of the diameter was 18.91 pixels, indicating that the radius is 9.4550 pixels, which is less than the minimum detected value from the code.

Since are images are treated as matrices in Matlab, then it should be kept in mind that the entire program uses integer pixel values. In fact any image processing technique takes this as granted fact. This means that practically there is no value of 9.8204 pixels in the image, but this value is assumed for length conversion because it reflects the program ability to define lengths in meters rather than in pixels. This assumption is considered in order to minimize the error that associates with the program.

4.2 Filtering the Images

A CME is recognized as a distinguished, large and bright object leaving the Sun. Thus in order to develop a computer program that is able to detect a CME event it must be able to isolate any large and bright objects leaving the Sun. The isolated object must be recognized from the background found in the image. Therefore, careful isolation of the region of interest should be made by removing any unneeded details in each image. This task is made in four steps in the present program as described below:

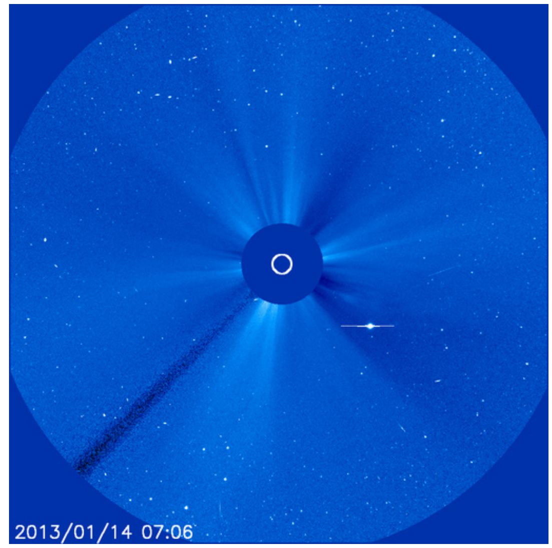

Figure (3-a). The image 20130114_0706_c3_512.jpg. This is assumed as a reference image to detect the solar radius.

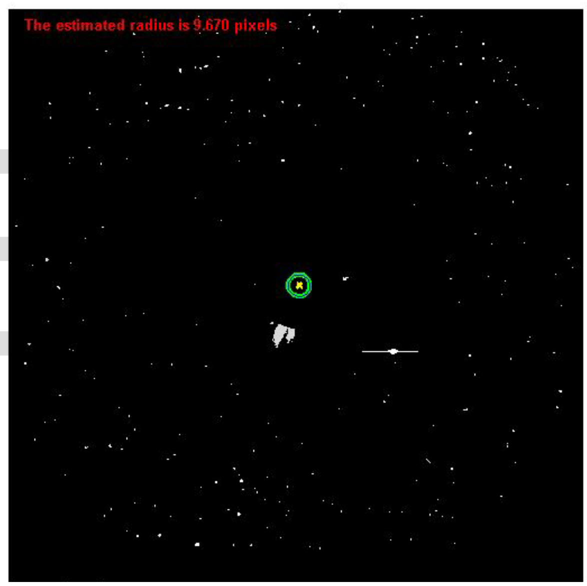

Figure (3-b). The estimated solar radius is 9.670 (in pixels) from the same image 20130114_0706_c3_512.jpg. This value corresponds to the solar radius 6.958x108 m. However, the value 9.8204 pixels is used-see below. The solar measured radius from the reference image was 9.6700 pixels.

First: Erasing of Descriptive Information Areas

Each C3 image has descriptive information such as the location of the Sun and the date of the image. The solar position is annotated by a circle at the center and the date at the lower left side of each image. Both of these areas are completely erased in the grayscale image of C3. The circle at center of the image by (248,230) , (280,230), (280,260) and (240,260). The region of date at the lower left side of the images by (1,480), (512,480), (512,512), (1,512).

ISSN 2229-5518

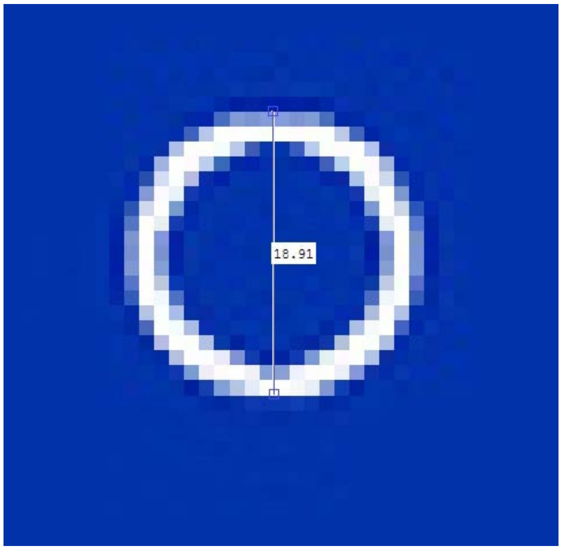

Figure (3-c). Visually-measured solar diameter using imtool of Matlab indicated that it is 18.91 (in pixels) from the same image used in Figures (3-a and b).

Second: Images Main Cleaning

A cleaning procedure is performed for all the images. In this step, each one of the images is subtracted from a reference image as mentioned in the aforementioned paragraph. This reference image is chosen previously for when there are no significant CME events.

It should be mentioned here that in this work it was found that the current method of choosing a reference image to subtract the selected images from is better than the method of running difference for selected regions around the occulting disk, as described elsewhere [17]. The present method performs faster and more accurately, and it assumes equal weight for the entire image since it is assumed that the whole image represents an area of interest. The examples illustrated in Figures (4) below show the results of cleaning procedure from this work. In order to achieve fast performance, the code was written such that it uses an initial background subtraction from as clean as possible reference image from LASCO C3. Any extra noise in the targeted image is compared with that noise in the reference image, if there is a difference and if this difference was large, the noise is assumed as a useful data and left out in the targeted image. See more details below. However, if the noise existed and it was not large it is then deleted from the targeted image before filtering the image. Such a process guarantees fast removing of fixed points in the background (far stars) and eases the removal of the occulting disk in the center of the image and date in the bottom left side.

The example given in Figure (4-a) shows how an image is easily and quickly cleaned up after subtracting it from a reference image. An important thing to be careful about is that after this process, all pixels are logically tested such that any pixel with a value not in the (0,1) category is rounded to the zero level. By this it can be ensured that the resultant image will still has a logical format consisting of (0,1) values only.

Figure (4-a). The resultant image from the cleaning procedure used in the present work for the image 20130110_0054_c3_512.jpg. No filtering nor thresholding were performed yet, but the middle circle and date were erased.

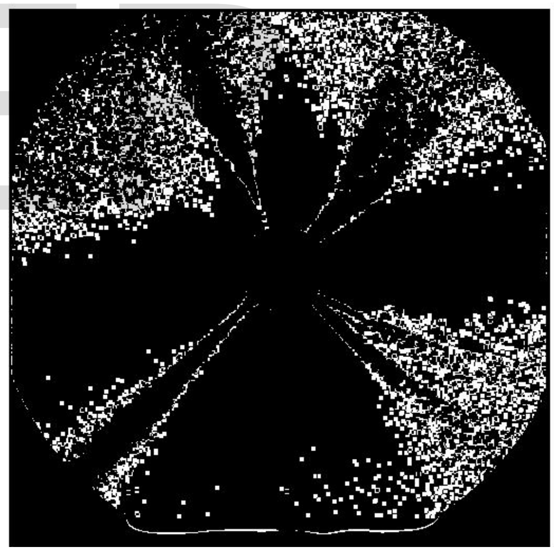

Figure (4-b). The resultant from the treating the same image using run difference for selected areas around the occulting disk. No filtering nor thresholding were performed yet, but the middle circle and date were erased.

Third: Spatial Filter Application

The resultant image is then filtered using spatial filter, then eroded to clean up small isolated objects. The filter used in the present code was an average filter. The resultant image from the previous step is stored in a matrix K. This is first tested, pixel by pixel, to detect any possible value of (-1). This worked if K(row,col)==1 and tested as if they are (-1), if so the pixel value is reset to zero. The result is stored in a vector, e.g.,

ISSN 2229-5518

B1(imgNumber, row, col) of the binary image. This is the vector which holds the pixels of difference for each image, considering the entire image sequence. This matrix is made using a matrix concatenation. After this the basic filter procedure is made. This is achieved by detecting where (1) is present, then test if the following six pixels (up and down) are (0)'s. If so, then the code replaces the (1) with a (0) in the matrix K, otherwise it aborts because it may mean that this block of pixels is a large block which might be a CME.

After this the code uses STREL and IMERODE functions with specified square shape, (2x2) pixels. The STERL function of Matlab creates a morphological structuring element that is proper for image erosion. The selected vector was chosen as a square one. This vector is made to put a small square of zeros centered at the remaining pixels of value (1). When experimenting few other procedure, it was seen that this method works smoothly with the best final results. Although Matlab comes with a verity of filters, the present method showed much better results than the pre-programmed Matlab filters.

Fourth: Fast Normalization

A simple normalization procedure was finally made for each image in the sequence. The resultant image was ran through a thresholding procedure and tested for isolated objects. This step is described before with enough details.

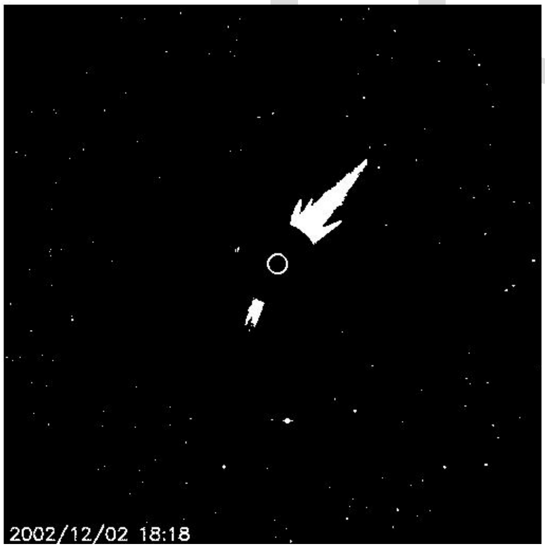

Figure (5-a). Image 20021202_1818_c3_512.jpg after black-white conversion only. No other treatment was applied to this image.

4.3 CME Detection

This is the key aim of the cmeDetect code. Each one of the image files is now assumed to be sorted in ascending manner, has its time stored, properly filtered and clean of unnecessary details, and has a proper (normalized black-white) format. Then each of these images are being input to the main procedure of the code where the CME is to be detected.

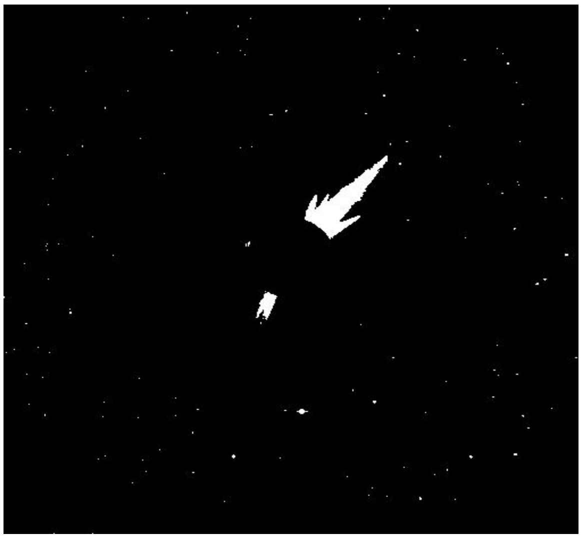

Figure (5-b) Same 20021202_1818_c3_512.jpg after black-white conversion and subtraction from the reference image. The middle circle and date were erased.

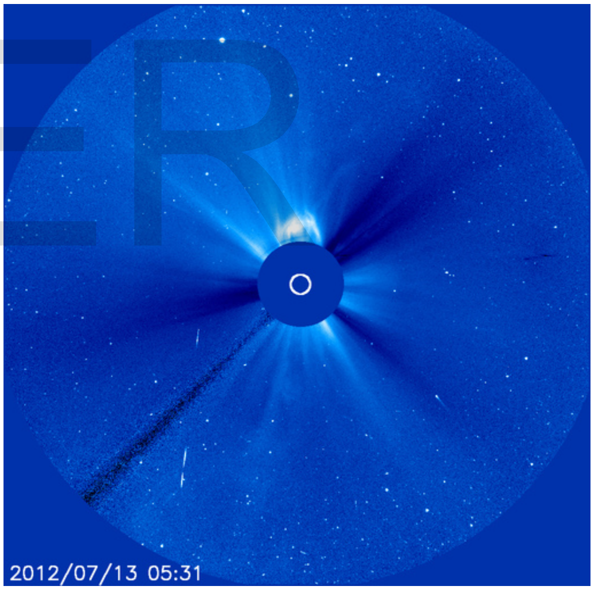

Figure (6-a). One of LASCO C3 images of the event of early day 13-7-2012. This is the 6th image in the sequence.

Detecting CME from LASCO images is an operation that requires special attention. For example, Vršnak et al. [18] discussed that white-light emissions that appear in coronagraph images might have a blob-like shape with speed range 10^2-10^3 km/sec; yet they may not be always generated from CME events. Thus the present code used a selection criterion from which the user can determine what size of the feature is considered and which one is ignored.

ISSN 2229-5518

Figure (6-b). The same image after filtering and preparation.

As mentioned before, the detection is made from the isolation and tracking of bulk mass areas in the image. In Figure (6-a) the event started in the early day of 13-7-2012 is shown, and the image after processing and preparation is shown in Figure (6-b). In this example it is seen how the program focused on the only bright objects representing a possible CME event. The detection of the present code is sensitive to few parameters as the size of the strel vector, the horizontal and dimension of the object and thresholding of the image.

Once each of these images is fed to the main procedure of the code, any possible CME is detected. If any image contains such an object, the program starts to follow its time development and measures simultaneously the following parameters. The following parameters are measured from each image for the entire filenames of a single CME event:

1. Time development of each image. Time is stored in minutes. Difference time Δt is calculated as $\Delta t = t_{i+1} - t_i$ was also calculated and stored, where t_i is the i^{th} time of each image obtained as described before. The maximum value of i represents the number of CME event images.

2. The height of the CME edge from the solar center. These values are corrected by subtracting one solar radius value in order to obtain the height from the solar surface rather than the solar center. The height measured in the present code, H, is assumed as $H = H_{c.m.} + C/2$, where $H_{c.m.}$ is the center of mass height of the detected object and C is the length of the semi-major axis of the detected object. The value of $H_{c.m.}$ alone, apparently, contains a considerable error in height determination because the wanted value must be from the outer (leading) edge of the detected CME to the solar surface (or center). Therefore, the object semi-major axis, C, is also measured in the main code for the detected object and half its value is added with $H_{c.m.}$ See the Figure (7).

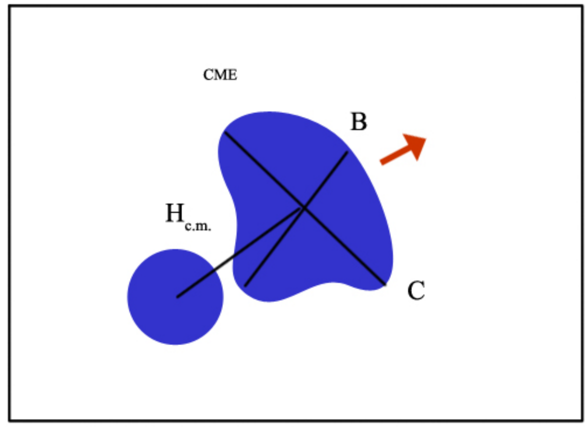

Figure (7). Any detected object in the LASCO C3 images is assumed to have semi-major and semi-minor axes, C and B respectively. The distance of the leading CME edge is approximated by adding C/2 to the center of mass height.

3. The direction of the line between center of mass of the detected CME and solar center point is calculated. This angle is then transformed such that it is being measured from the solar north and with direction of counter clockwise. This transformation is made in order to obtain comparable results with the CDAW catalog [45].

4. The area of each CME is measured. The value of this parameter is measured with (pixels^2) and then converted into (kilometers^2) using the same conversion parameter mentioned before.

After these steps, the main code directs the results into subfunctions to calculate the following parameters:

5. Speed. The speed is calculated from dividing the matrix containing CME heights by the matrix containing time difference. The resultant speed would also be stored in a matrix form. CME speed is calculated as km/sec.

6. Acceleration. The speed matrix goes through a difference procedure by the time difference matrix and the result is stored into the acceleration matrix. CME acceleration is calculated as km^2/sec^2.

Next, the numerical results of the code are presented for a single example and discussed with a comparison to CDWA catalog.

5 CME EVENT OF 04/12/2002 AT TIME 1:33:00

The results are shown in the Figures (8-a to d). The figures are arranged as before. In Figure (8-a), it can be seen that after about 3 hours the CME height raised suddenly from 7Rs to about 9 Rs. CME height linearity this time is less dependent on time specially at high altitudes where from the present results it is shown that speed decreased at about this point of time as shown in Figure (8-b).

Another remark is that te height of CME resulted in decreased values at about the end of the calculations,

ISSN 2229-5518

which obviously declares that there is an inconsistent results. This behavior is due to the nature of the present code which, as described before, depends on determining the bulk mass of the CME as a detection method.

In this case it can be noticed that CME height developped almost linearly with time till time ~ 150 minutes then almost constant value was reached at time between 150 to 250 minutes, followed by a sudden acceleration that took place till time of about 300 minutes. This remark, combined with the results of the date 01-12-2002 given above, indicates that the region that lays at about 5 to 15 R_s acts as an acceleration region for CME. This effect was suggested earlier [14] where the reason was suspected to be either due to solar flare associated with CME eruptions or with interplanetary medium. Alsawad [14] mentioned that "if the bulk of acceleration takes place in interplanetary medium, then the interplanetary shock driven ahead of the CME is the major accelerator and hence the flare part in acceleration will be a minor or even not at all an accelerator, according to some studies."

The persent height values were, in general, much higher than correspondent CDAW values till about time ~ 300 minutes, then dropped below that value as seen in Figure (8-e). From this figure a linear fitting showed that the speed value was calculated as 262 km/sec while CDAW value was 301 km/sec, giving an error value to present result of about 13%. This was by no means better than the mean average value of CME speed taken from Figure (8-b) which was 140 km/sec and error 53%. Acceleration results also indicated this generally, as shown in Figure (8-c). The mean average value of calculated acceleration was about 2.1 m/sec^2, and that from CDAW was 2.65 m/sec^2; thus the present calculation was in general agreement with the standard catalog value within an error about 20%. The mean position angle measured in this example was 315 degrees and that of the CDAW catalog was 339 degrees, which means that the error in the present calculations was about 5% only.

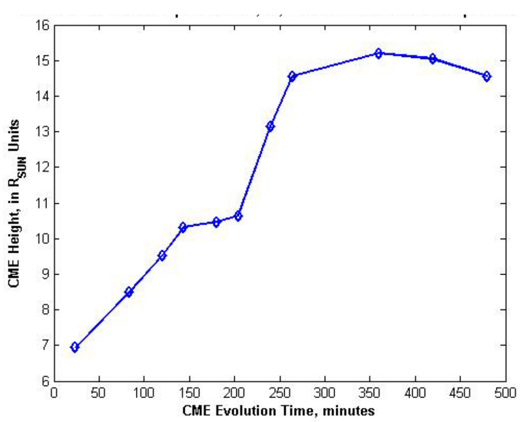

Figure (8-a).The present results of CME height of the event 04/12/2002 at time 1:33:00.

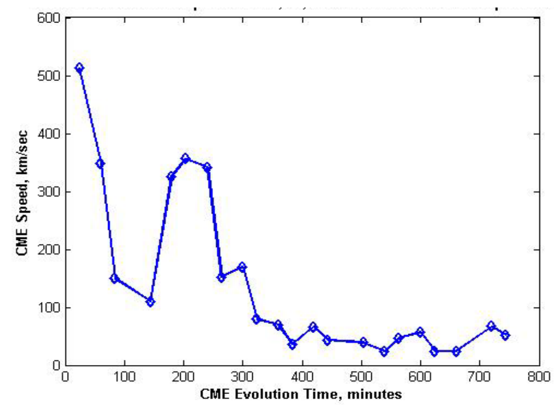

Figure (8-b).The present results of CME speed of the event 04/12/2002 at time 1:33:00.

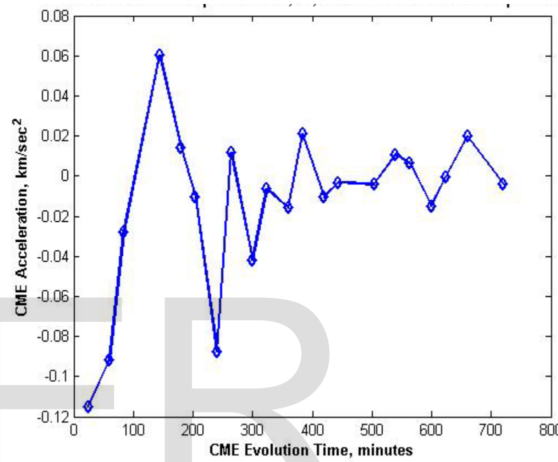

Figure (8-c).The present results of CME acceleration of the event 04/12/2002 at time 1:33:00.

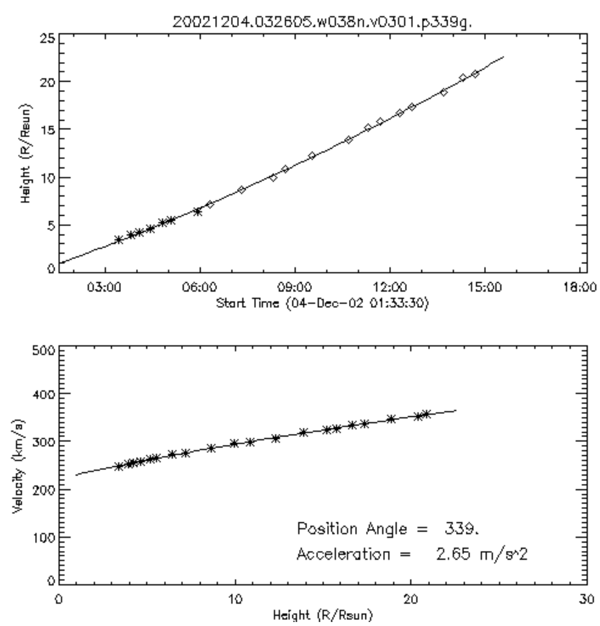

Figure (8-d). Height and its derivative from CDAW [16].

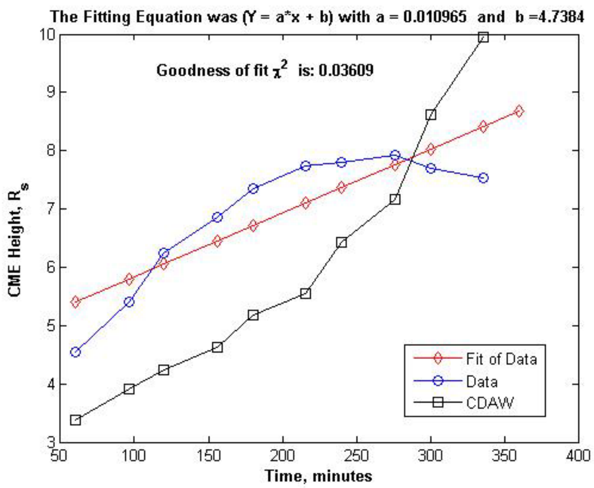

Figure (8-d). Height and its derivative results of the event 04/12/2002 at time 1:33:00 compared with CDAW results.

REFERENCES

[1] Jason P. B., "The Kinematics and Morphology of Solar Coronal Mass Ejections", Ph.D. Thesis, School of Physics University of Dublin, Trinity College, 2010.

[2] The official website of SOHO from NASA (accessed 2-12-2014): http://sohowww.nascom.nasa.gov/

[3] R. Mc Queen et al., "Initial results from the high altitude observatory white light coronagraph on skylab – a progress report", phil. Trans. R. Soc. Lond. A. 281, 405 – 414, 1976.

[4] R. Mc Queen et al., "Coronal transients: A summary", phil. Trans. R. Soc. Lond. A 297, 605 – 620, 1980.

[5] Richard, A. H., "Coronal mass ejection", New York, 1991.

[6] Gosling, J. T., "Magnetic Topologies of Coronal Mass Ejection Events: Effects of 3-Dimensional Reconnection" Los Alamos National Laboratory, 1995.

[7] David, F. W., " Coronal Mass Ejections: Origins, Evolution, and Role in Space Weather", IEEE Transactions On Plasma Science, Vol. 28, 2000.

[8] Andrews M. D. and Howard R. A., "A two-Type Classification of LASCO Coronal Mass Ejection", Space Science Reviews, Volume 95, Issue 1-2, pp 147-163, 2001.

[9] Y.Q. Wu et al., "The Solar Origin of the 6 January 1997 Coronal Mass Ejection" Solar Physics, Volume 207, Issue 1, pp 159-171, 2002.

[10] Webb, D.F., "CMEs and the solar cycle variation in their geoeffectiveness", vol. SP-508 of ESA Special Publications, pp. 409–419, 2002.

[11] D.F. Webb, T.A. Howard, C.D. Fry , T.A. Kuchar, D. Odstrcil, B.V. Jackson, M.M. Bisi, R.A. Harrison, J.S. Morrill, R.A. Howard, and J.C. Johnston; "Study of CME Propagation in the Inner Heliosphere: SOHO LASCO, SMEI and STEREO HI Observations of the January 2007 Events", Solar Phys. 256: 239–267, 2009.

[12] Webb D.F., "CMEs observed in the heliosphere by the Solar Mass Ejection Imager (SMEI)", USA, 2004.

[13] Goussies N. et al., "Detection of Coronal Mass Ejections", IEEE International Conference on Image Processing ISSN: 15224880, Pages: 1065-1068, 2008.

[14] Al-Sawad A., "Multi –Eruption solar energetic particle event observed by SOHO/ERNE", PH.D thesis, ISBN: 978-4165-0, Department of Physics and Astronomy, University of Turku, Turku, Finland, 2009.

[15] Selman, A. A. and Thabit, S. A., "Analysis of Solar Magnetic Field Using the Dynamo Model", International Journal of Scientific & Engineering Research (IJSER), Vol.5, Issue 6, June-2014, p.383.

[16] CDAW Catalog from ALSCO/SOHO website (accessed 2-12-2014): http://cdaw.gsfc.nasa.gov/CME_list/

[17] Rafael C. G., Richard E. W., "Digital Image Processing", 2nd Ed., 2001.

[18] B. Vršnak, G. Poletto, E. Vuji´c, A. Vourlidas, Y.-K. Ko, J. C. Raymond, A. Ciaravella, T. Vic, D. F.Webb, A. Bemporad, F. Landini, G. Schettino, C. Jacobs, and S. T. Suess, "Morphology and density structure of post-CME current sheets", Astro. Astrophys. 499, 905–916, 2009.