Determination of the 3D Trajectory and Velocity of Coronal Mass Ejections using Stereoscopy

Vahab Pournaghshband
University of California, Berkeley
vahab@berkeley.edu

Paulett C. Liewer
Jet Propulsion Laboratory
paulett.liewer@jpl.nasa.gov

Abstract

Coronal mass ejections (CMEs) are powerful eruptions that can blow up to ten billion tons of the Sun's atmosphere into interplanetary space. Major CMEs which impact the Earth are capable of creating major disturbances in the outer atmosphere, causing geomagnetic storms and interruptions to communications systems. Identifying those CMEs that will hit the Earth, which appears as "Halo events" in coronagraph images, enables us to significantly reduce the potential damages caused by CMEs. NASA's two-spacecraft STEREO mission, to be launch in the summer of 2006, will take time series stereoscopic image pairs of CME. We developed and implemented a time efficient algorithm that intelligently detects and tracks the CMEs on the stereo images taken by STEREO. We have then tested our software on six synthetic stereo image sets created by our collaborator. By applying the implemented program to data received by STEREO, Halo events will be predicted early enough to take necessary actions in order to reduce CMEs' severe damages.

1. Introduction

Coronal Mass Ejections (CMEs) or solar storms (figure 1) are powerful eruptions that can blow up to ten billion tons of the Sun's atmosphere into interplanetary space. They travel away from the Sun at speed of approximately one million mph (1.6 million kph). Major CMEs that hit the Earth (Halo event) are capable of creating major disturbances in the interplanetary medium and trigger severe magnetic storms when they collide with Earth's magnetosphere. The potential disturbances caused by these large solar storms directed towards the Earth fall into a variety of capacities: They can damage and even destroy satellites; they are extremely hazardous to Astronauts when outside of the protection of the Space Shuttle performing Extra Vehicular Activities (EVAs) [1]; and they have been known to cause severe electrical power outages. By all means, reasons for detecting and tracking Halo events are compelling.

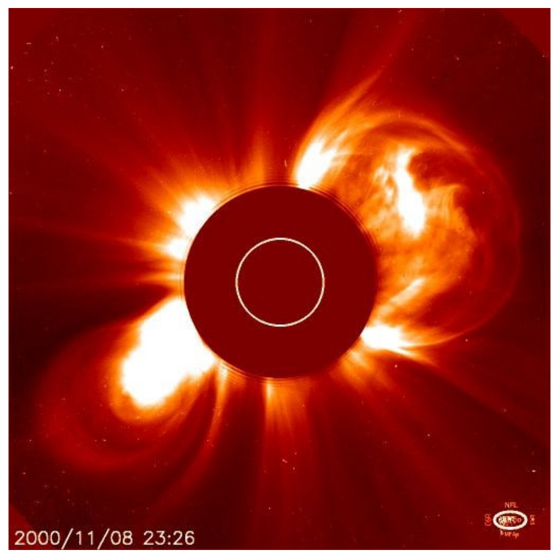


Figure 1. Coronal Mass Ejection as seen by a coronagraph on
SOHO (Solar and Heliospheric Observatory)

To better understand how CMEs originate and how they change as they travel to Earth, National Aeronautics and Space Administration (NASA) is planning to launch the twin STEREO (Solar TErrestrial RElations Observatory) spacecraft (figure 2) in near future . STEREO's unique stereoscopic images of the structure of CMEs will enable scientists to determine their fundamental nature and origin. The images taken by STEREO can also be used to determine the 3D trajectory of CMEs to verify whether a Halo event is about to take place at early stage. The main objective of this research is to automate this verification process.

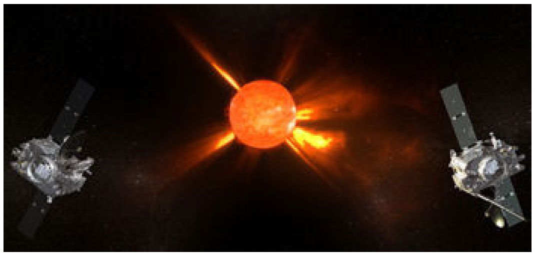


Figure 2. STEREO Twin Spacecraft (courtesy of NASA)

2. Approach

In order to achieve the objectives of this project, we developed and further implemented a software program using Interactive Data Language (IDL). The high-level architecture of the software is illustrated in figure 3.

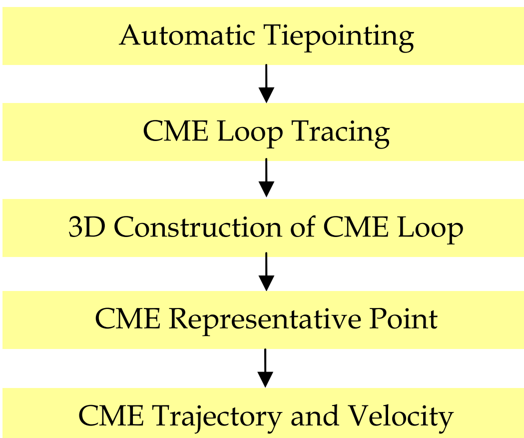


Figure 3. High-level Architecture

As shown in the architecture, our software consists of five modules and follows a pipe-and-filter architecture with no feedback. Details about implementation of these modules are given below.

1. Automatic Tiepointing

The automatic tiepointing module is intended to receive a pair of stereo images containing a CME and return one point on each image (preferably on the CME leading edge) that correspond to one another. This module consists of four sub-modules - Masking, Skeletonizer, Closing, and Slope Analyzer.

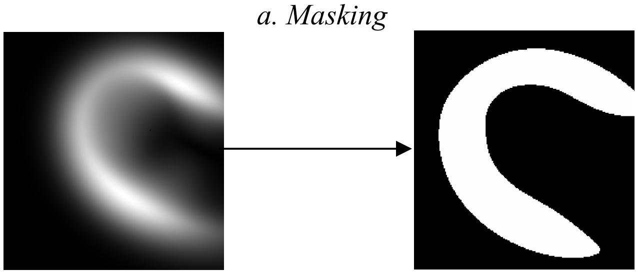
1.1. Masking. Initially, each image is passed through the masking filter to retrieve a dataset containing only zero and ones. Masking filter, also known as two-tone converter, eliminates the unnecessary complexity that a long-range image would introduce for later manipulation.

1.2. Skeletonizer. Here, automatic tiepointing would then (i.e., after masking) extract the skeleton of the CME by utilizing the Stentiford's thinning algorithm [2]. The use of the thinning algorithm results in a single-point thickness and relatively smooth but discontinuous curve representing the skeleton of any object in the original image. See figure 4 (b).

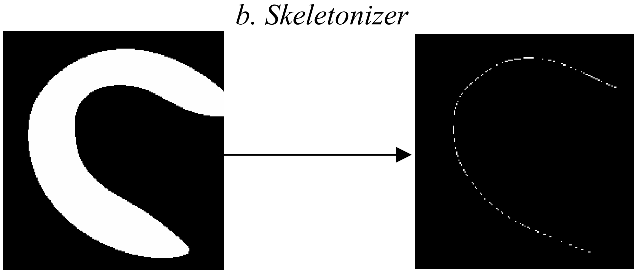
1.3. Closing. Since a continuous curve is what we are specifically interested in, using a closing algorithm that fills the gaps in the skeleton of the CME is necessary. Even tough there exists a handful robust closing algorithms [3,4], we have chosen to develop our own version which is more efficient than those available algorithms. The closing algorithm that we present here, searches for an endpoint (i.e. neighborhood of one) in every four direction (bottom up, top down, right to left, left to right) as its starting point. It then traces the curve segment until it reaches another endpoint. The goal then is to find the correct endpoint to connect them and fill the gap. Our closing algorithm searches in its neighborhood of 0 to some threshold τ and if point p is an endpoint in the neighborhood of less than τ and the endpoints belong to separate curve segments, they will be connected using the simple line algorithm. The algorithm will then traces the connected segment and repeats the procedure. If such an endpoint could not be found, the algorithm looks for another point that can begin with. If this also fails, τ will be incremented. The process will be repeated until either the maximum τ is reached or there is no more gaps left to fill. The before-and-after pictures at each stage are illustrated in figure 4.

1.4. Slope Analyzer. The fact that both spacecrafts are constantly positioned at the same elevation results in a vertical mapping for any pair of images. Hence, if the drawn horizontal line at a given elevation results in single intersection in each image, the intersections map one another in a good approximation. In case of multiple intersections, the inner-most intersections are assumed to map one another. Now, that given an elevation, we are able to identify a corresponding pair of points, the only question that remains unanswered is what elevation is desirable. The later modules assume that the chosen points represent a 3D point on the CME's leading edge. To accomplish this, Slope Analyzer draws consecutive secant lines for the entire curve and assigns the sign of the slope to each segment. Since a bell-curved shape is always assumed, the presence of a peak (or turning point) is accordingly assumed. By looking at the change in signs, the region that contains the leading edge is identified, and by picking the median point in that region the presence of the point on the leading edge is ensured. Figure 5 illustrates this process pictorially.

a. Masking



b. Skeletonizer



c. Closing

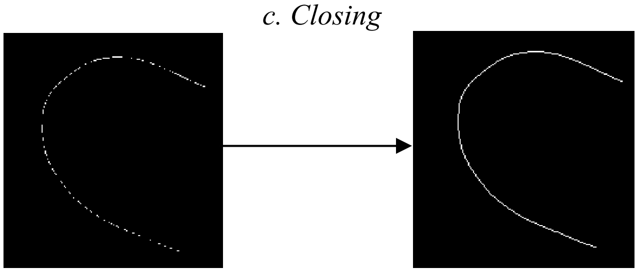


Figure 4. Preprocessing Filters

2. CME Loop tracing

An existing module that traces the edge of the CME extracts the loop on each image of the stereo pair.

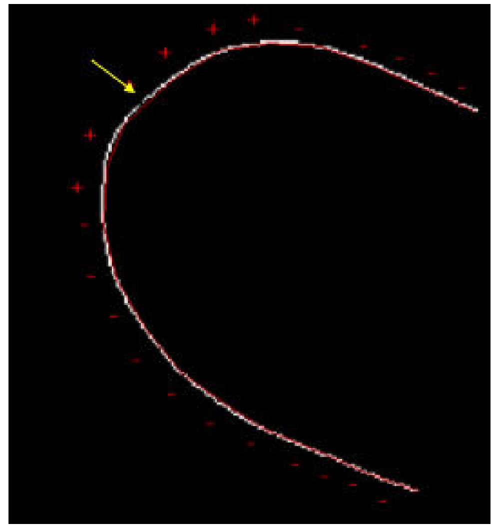


Figure 5. Slope Analyzer Process

3. 3D Construction of CME Loop

An existing module that employs the triangulation concept to match up the points traces the edge the loop on the two images and locates the points in the 3D space. Given the angles between each spacecraft and the matched points on the CME leading edge and the known distance between the spacecrafts, the triangle is reconstructed.

4. Detection of the CME's Representative Point

In order to reduce the complexity that tracking the entire CME would introduce, this module attempts to identify the most representative point on the CME. This representative point is assumed to be the farthest point on the CME from the Sun[1].

5. Determination of the CME Trajectory and Velocity

The input to this module is a set of 3D points that are ideally representing the leading edge of the CME at each stereo image pair. Our software will then tracks these points to determine the CME's trajectory and velocity. This module also rejects the incomplete data before further calculation to avoid noise in the result. Ultimately, the outcome would be summarized as the following three plots:

1. CME's are assumed to move radially, that is their angular deviation is negligible.

1. Distance vs. Time: indicates the CME's displacement at each moment
2. Velocity vs. Time: indicates the average velocity of the CME[1]
3. Longitude vs. latitude: illustrates the direction of the CME at certain times

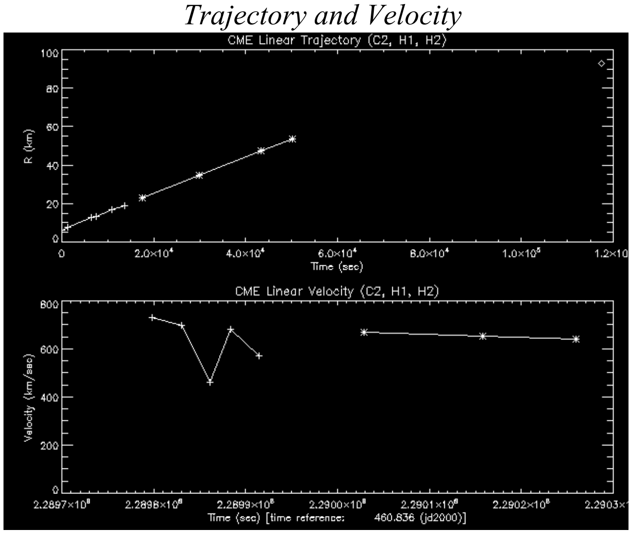
3. Testing

Six synthetic image sets were provided to us by one of our collaborators to test our software. The result of our software on the first simulated data along with its actual characteristics is provided in the next section for comparison.

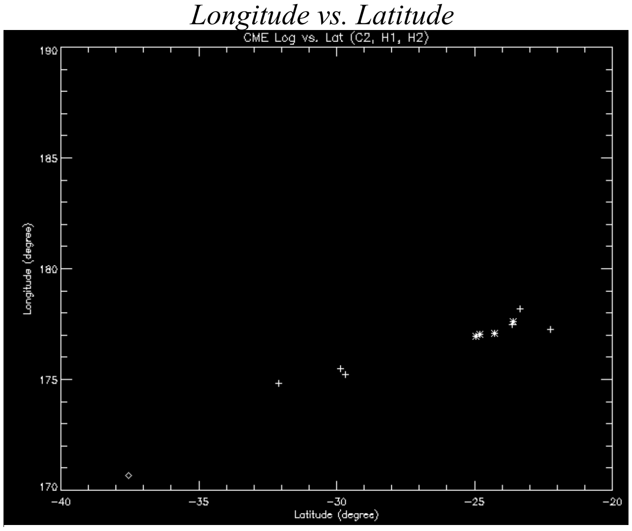
4. Result

Here are the final plots of one of the simulations produced by our software.

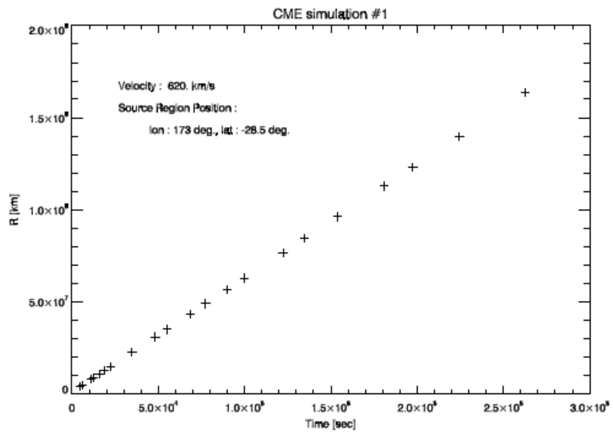
Trajectory and Velocity



Longitude vs. Latitude



And here we provide the actual characteristics of the same simulated CME for comparison.



5. Acknowledgment

The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with National Aeronautics and Space Administration (NASA). This work was supported by JPL Undergraduate Scholar program.

References

[1] I. Nourmohammadi, H. Ahmadvand, and M. Taghikhani, Evaluation of Levels of Macro- and Micro Nutrient in Workers Exposed to Electromagnetic Fields and Comparison with Levels of Patints with Leukemia, 1996 Iranian Biomedial Journal.
[2] J. R. Parker. Algorithms For Image Processing and Computer Vision. John Wiley and Sons, Inc., 1997.
[3] R. Deriche, J. P. Cocquerez, and G. Almouzny. An Efficient Method to Build Early Image Description, IEEE, 1988.
[4] O. Monga, R, Deriche, G. Malandian, and J. P. Cocquerez, Recursive Filtering and Edge Tracking: two primary tools for edge detection, Butterworth-Heinemann Ltd, 1991.

1. Average velocity is computed using the simple average velocity formula: $V_n = \dfrac{R_{n+1} - R_{n-1}}{t_{n+1} - t_{n-1}}$