Using UV Event Localization to Constrain Magnetic Reconnection Geometry in Solar Flares

Research by Nathaniel Balmain
Goals

• Introduce our research and findings for a solar event
  • Date October 09, 2014
  • Time 01:35:52 – 02:00:54
• Explain how the created code extracts data from solar flares
• Summarize results
• Importance of our source selection and the process behind it
  • No overlapping is important
Abstract

• Hard X-ray and ultraviolet emissions in solar flares have been shown to be temporally correlated for complete flare regions (Alexander and Coyner, 2005). For larger, more complex flares, the temporal correlation of the full flaring region persists, however individual sources may be uncorrelated or only correlated to a portion of the event. Emission evolves in a series of localized sources.

• One side effect of the sun’s plasma behaving like a fluid, is magnetic field lines warp and bend. When these field lines warp, it creates a kink point, or an X point which serves as a magnetic reconnection site. When the magnetic field lines reconnect, it creates X-ray emissions through the release of high energy electrons. When these particles are jettisoned, they slam back down into the sun with an incredible force. This impact with more dense plasma creates ultraviolet emissions in response. If these are correlated and coming from the same source, it is indicative of UV and hard X-ray emission having the same physical origin, the flare reconnection event. We show evidence of an event where the reconnection sites evolve throughout the flare.

• To prove our hypothesis, we created a versatile code framework that is capable of data acquisition for the 1600Å and 1700Å wavelengths specifically. Our code is also able to handle data from any wavelengths of the Solar Dynamics Observatory’s Atmospheric Imaging Assembly instrument (Lemen, et. al, 2012, Pesnell, Thompson, & Chamberlin, 2011). This allows for more complete pictures of the emission response in localized regions.
X-Ray components

- Obtained from Reuven Ramaty High Energy Solar Spectroscopic Imager (RHESSI)
- Selected 2 energy bands for analysis
  - 6-12 keV (Soft X-Ray)
  - 25-50 keV (Hard X-Ray)
- Hard X-Rays show primary event particle emission
- Soft X-Rays show thermal response to the primary and secondary event particle emission
RHESSI Initial Data

RHESSI Photons vs Time

- 9 Oct 2014
- Detectors: 1F, 3F, 4F, 5F, 6F, 8F, 9F
- 25.0 to 50.0 keV

Photons vs Time (09-Oct-14, 01:30:00 to 02:00:00)
RHESSI data

- Photon count value for 25-50 keV
- Two peaks can be seen around 01:42 and 1:58
- Large spike at 02:00 is an attenuation switch, not connected to the flare.
- By comparing all UV data to this X-Ray data we can see how the flaring region is behaving, and how the particle emission is connected to reconnection.
Hard X-Ray vs Soft X-Ray

- Hard X-Ray emissions primarily present in the primary solar event with a partial presence in the later event.
- Soft X-Rays 6-12 keV are present for primary event and secondary event. In addition, the middle event that peaks around $T=1:51$ is indicative of thermal fluctuation in the solar plasma.
Regions of interest

- Regions 0, 1 and 2 are most of note producing the most particles.
- Regions 3-6 were not a significant source of particle emission.
- Region 4 has some coincidence due to its special relation to the primary event region, though not enough to indicate magnetic reconnection.
Golden image was created with the AIA 1600 color table.

Right image was created with The STD GAMMA-II color table (loadct, 5).

The data presented in regions 3-6 produced graphs that indicated that they were not primarily correlated with the solar event.

It is likely that these regions, while in the local event region, are not a primary source of UV Radiation.
• Region 4 (1600Å in yellow and 1700Å in white) is minorly connected with the event due to its relatively close proximity.
Region 0

• This is the region of interest that has the single largest data point value right around the center of the diamond.

• For 1600Å
  • 6336.25 Counts per Pixel

• For 1700Å
  • 17509.14 Counts per Pixel

• The max value in the code is very important.
  • Used for growing all regions quickly.

• Max value can be viewed as counts per second on that particular pixel
• 1600Å and 1700Å Curves correlate primarily to the peak event.

• Clear X-Ray coincidence with both early onset and subsequent flare.

• Thermal fluctuation events in the middle peaks at T= 1:45 and 1:51
Both UV curves correlate primarily to the first event

Soft X-Ray (blue) Curves indicate there is some thermal fluctuation coinciding with event release around 1:51

Apparent correlation with primary event, less so with secondary.
Region 1
Region 1 Data

- Region 1 has a stronger correlation with the hard X-Rays than region 0.
- Initial hard X-Ray burst (blue) at the main peak with some thermal response on the shoulder. The second peak increases when the X-Rays increase again.
- Thermal response indicated from the smaller middle peaks of the 1700Å graph.
Region 1 Data

- More correlation in 1700Å for secondary event than for 1600Å
Region 2 Image
Region 2 data

- Region 2 has more correlation with the secondary event than region 0 or region 1.
- Likely indicates a magnetic field reconnecting to a different part of the plasma.
Program Goals

• Extract Data provided from SDO in the form of .fits files
• Select regions of interest from the flare
• Extract total counts per pixel value from each unique region of interest.
• Save the extracted data for further analysis.
Main

- Main body of the program.
- Includes 2 sub functions
  - Grow_me
  - Count_per_pixel
- This part of the program is responsible for passing and handling of data, primarily in the form of Maps in this case.
- For each file we want to extract the total counts per pixel value for the regions of our solar flare.
Main

- Load the .fits files into memory
- Load the data from the .fits files
- Prep the color table
- Convert the data to maps
- Derotate the maps
  - Very important for spatial correlation of values.
- Display a macro view of the overall region of interest at the point in time when largest emission was found
  - This image is used for region selection.

Get some water because this takes some time
Main

- The user clicks on a particular region they want to examine
  - To save time regions 3-6 were hard coded. This has since been fixed, but the hard coded data was used for analysis, due to time constraint.
- The program expands the region out by using the function Grow_me.

Flowchart:
- Get click position from user
- Expand out region box
- Grow_Me
- Create a new map from these points
- Plot counts per pixel against times
- Save variable and plot
Grow_Me Function Overview

1. Scan out in the four cardinal directions. The criteria are the same for every direction.
2. Find the current and the next pixel value.
3. If the difference between these points is greater than a value, break.
4. Add 1 to the direction.
5. If current is in range, look for the next point.
6. Add buffer to the direction.
7. For the size of the region do:
   - Look for the next point.
Grow_Me

- This process is the same for each cardinal direction, only the direction we scan is different.
  - for +x, add 1 to the x part of the array
  - For –y subtract 1 from the y part of the array
  - Etc.
- Treats the data like a 2 dimensional array in the form of (x,y) with x values corresponding to their position in the X direction, likewise for the y direction.
- Count values are stored in (x,y) format.
Grow_Me

• Effectively this function is 4 single for loops with logic that tells it when to quit.
• For this event, the size of the regions never exceeded a value of 20 for our scanning direction (exit condition 1).
  • This value is changeable for other flares with larger or smaller regions.
• Increment the for loops by adding the pixel spacing (dx or dy) to the iterator for the particular map.
  • Different sub_maps this function receives can have different dx/dy values, this ensures a consistent function.
Grow_Me

- The program finds the data values at the current point and at the next point.
  - Represented by data(x,y)

- If the current value is in range, it will add 1 to our spacing counter if not, this function ends (exit condition 2)

- It also checks the difference between the current and the next value, as for this event in 1600 Å is always under 1000 for our most notable regions of interest: 0, 1 and 2 (exit condition 3).
  - This value is flexible, and the user can change it for their solar event.
Grow_Me

- The program then adds a small buffer to all directions and returns the array of our newfound “grown region”
- The x/y buffer is also a variable that the user can change if necessary.
  - Also serves to combat irregularities in very small regions or irregularly shaped regions.
Grow_Me Function Overview

Scan out in the four cardinal directions. The criteria are the same for every direction.

Find the current and the next pixel value.

Add buffer to the direction.

For the size of the region do:

Look for the next point.

If current is in range:

Add 1 to the direction.

If the difference between these points is greater than a value, break.

Return array of our "grown region".
Main

- Grow_me returns an array that matches the form of the sub input for the index2map built in IDL function, this is intentional.

- If we think about this in the cardinal direction point of view
  - \([-x, +x, -y, +y]\)
  - These values are the range that will be used to create a sub_map of the region.

- Now that we have a sub_map of our region, we can start examining it much closer.

```
index2map, index, data2, regionmap1, /double, /normalize,$
sub = [sub_args(0), sub_args(1), sub_args(2), sub_args(3)]
```
Main

• By using the amount of files as the iterator, we are able to keep the program scalable.

• For each image the program will acquire the total counts per pixel in order to graph it against the X-Ray data for the same solar event.
  • Data by RHESSI satellite
Count_per_pixel

For Each map File do
  Subtract background from each pixel
  Find where data is inside threshold
  Sum all pixels
  Divide by the number of pixels that match our criteria
  Add counts per pixel value to array

Return Counts per pixel array
Count_per_pixel

- For each file in the given map region:
  - subtract the background from each pixel value.
- For this flare, the following values were used:
  - 1600Å: 42 Counts Per Pixel
  - 1700Å: 393.395 Counts Per Pixel
- find where the data is above the threshold.
  - $2 \times$ the background value is the minimum threshold default value; changeable.

```
For Each map File do
  Subtract background from each pixel
  Find where data is inside threshold
  Sum all pixels
  Divide by the number of pixels that match our criteria
  Add counts per pixel value to array
Return Counts per pixel array
```
Count_per_pixel

- Sum all values found for the region
- Divide the summed value by the number of pixels
  - Note this is the number of pixels within our threshold, NOT the total number of pixels in the region itself.
- Add this value to an array with index corresponding to the file
- Return the array containing the total counts / pixel.

Flowchart:
- Sum all pixels
- Divide by the number of pixels that match our criteria
- Add counts per pixel value to array

Subtract background from each pixel
For Each map File do
Find where data is inside threshold
Sum all pixels
Divide by the number of pixels that match our criteria
Add counts per pixel value to array
Return Counts per pixel array
counts per pixel

For Each map File do
  Subtract background from each pixel
  Find where data is inside threshold
  Sum all pixels
  Divide by the number of pixels that match our criteria
  Add counts per pixel value to array

Return Counts per pixel array
Main

- The base program only plots the UV data for each region alone.
- It does, however, save the variables for manual plotting.
- Such a change could be easily implemented by adding another helper function that plots the data created by this program as well as the hessi object.
  - See appendix plot code chart
- Plotting of the graphs in the presentation were created manually by using utplot and outplot.
Main

- This code does not check for duplicate values due to time constraint.
- For the solar event selected, the code does not generate a region that overlaps with another because of their respective locations.
- Future implementations would need to check for a duplicate value before obtaining Counts_per_pixel.
Sample manual plot chart

;region 0 16 and 1700 Å vs hard x-rays
1 utplot, anytime(time, /ext), data.countrate[3,*], color = 50, thick = 1.5, xstyle = 1, title = '
2 outplot, mytimes1700[*], region0_1700_sum[*]*.02, psym=10, color = 255, thick =2
3 outplot, mytimes1600[*], region0_1600_sum[*]*.1, psym=10, color = 110, thick=2
4 imd_legend, ["Hard XRAYS", "1600Å", "1700Å"], color = [50,110,255], position = 12

;region 1 16 and 1700 Å vs hard x-rays
5 utplot, anytime(time, /ext), data.countrate[HARD,*], color = 50, thick = 1.5, xstyle = 1, title = '
6 outplot, mytimes1700[*], region1_1700_sum[*]*.02, psym=10, color = 255, thick =2
7 outplot, mytimes1600[*], region1_1600_sum[*]*.1, psym=10, color = 110, thick=2
8 imd_legend, ["Hard XRAYS", "1600Å", "1700Å"], color = [50,110,255], position = 12

;region 2 16 and 1700 Å vs hard x-rays
9 utplot, anytime(time, /ext), data.countrate[HARD,*], color = 50, thick = 1.5, xstyle = 1, title = '
10 outplot, mytimes1700[*], region2_1700_sum[*]*.02, psym=10, color = 255, thick =2
11 outplot, mytimes1600[*], region2_1600_sum[*]*.1, psym=10, color = 110, thick=2
12 imd_legend, ["Hard XRAYS", "1600Å", "1700Å"], color = [50,110,255], position = 12
Conclusions

• The code extracts data that can be further analyzed to find evidence of magnetic reconnection.

• The M1.3 Primary flare and the M1.4 secondary flare on October 09, 2014 from 01:35:52 to 02:00:54 has data that indicates strong correlation for magnetic reconnection points moving from one part of the solar plasma to another.
  • Regions 0 or 1 to region 2.
References


Acknowledgements

• The material contained in this document is based upon work supported by a National Aeronautics and Space Administration (NASA) grant or cooperative agreement. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of NASA.