

Image Enhancement of Extreme Ultra-Violet Solar Images

R. M. Caplan

Abstract—Here I present my efforts to reproduce similar results to the new image processing technique used in a recently published paper to enhance the extreme ultra-violet solar images taken with the EIT instrument aboard the SOHO satellite. The main procedure of the paper is to use the a-trous wavelet (otherwise known as the stationary wavelet transform) to bring out structural details in the images not previously observed. In order to do this, the original images need to be pre-processed and then background light models and noise masks are developed for use with the a-trous algorithm. This report is organized as follows. In Section 1, I introduce the problem and the paper used in the study. In Section 2, I show my attempts at preprocessing the original raw data and compare it to the official processed data from the SOHO website. Then, in Section 3, I describe the a-trous algorithm and its application to the enhancement of the solar images. In Section 4, I display my results from using the a-trous wavelet. In Section 5, I describe the additional steps taken in the paper and their results, and conclude in Section 6.

Index Terms—A-trous wavelet, EIT images, SOHO.

I. INTRODUCTION

SOHO, the Solar and Heliospheric Observatory, is a project of international cooperation between ESA and NASA to study the Sun, from its deep core to the outer corona, and the solar wind [1]. One of its many instruments is the Extreme ultraviolet Imaging Telescope (EIT), which provides full disc images of the Sun at four selected colors in the extreme ultraviolet (EUV), mapping the plasma in the low corona and transition region at temperatures between 80,000°C and 500,000°C.

The EIT can image active regions, filaments and prominences, coronal holes, coronal "bright points," polar plumes, loops, and arcades, as well as dynamical events such as flares and mass ejections [2].

However, the multiscale nature of the observed solar features has not been fully exploited so far. Guillermo Stenborg, Angelos Vourlidas, and Russell A. Howard have come up with a wavelet-based processing technique that enhances the EUV images based on their multiscale nature, and reveals features not seen with standard image-processing techniques [3]. They have processed the entire EIT data set with their technique, and have made it available to solar physicists.

In this report, I document my attempt to reproduce their

results using techniques that are similar, but less complicated. I first try to pre-process the raw data to try to recreate the 'official' processed images from the SOHO database, and then apply an a-trous wavelet algorithm similar to the one used in [3].

II. PREPROCESSING OF EIT IMAGES

A. Data Types from EIT

There are generally two types of EIT solar images that are available for use. These are the raw data images that have had no processing done to them, and the images that are produced after standard processing techniques (these include dark current subtraction, de-gridding, flat field, degradation correction, contrast stretching, as well as others). The raw images are in uint16 format (0->65,535), while the processed images are in RGB (0->255,0->255,0->255) format. The raw data can be accessed at [4], while the processed images can be accessed at [1].

B. Image Rescaling

According to [3], the a-trous wavelet can bring out noise and grid patterns in the images because they were not completely eliminated in the pre-processing. To solve this, they add noise masks to their algorithm as will be explained in Sec. 5. Since these masks are complicated to form, I decided to try to pre-process the raw data myself to rid my chosen image of as much gridding as possible, so that my a-trous wavelet will not bring it out.

When loading the raw image from EIT into MATLAB, it is noticed that there are a minority (about 70 out of a million pixels) intensity values that are way higher than the rest of the image as shown in Fig. 1.

To eliminate this (which can cause problems in rescaling and imaging methods), I use the following thresholding:

$$\text{if}(im_{i,j} > T) \rightarrow im_{i,j} = T + \left[100 \frac{im_{i,j}}{\max(im_{i,j})} \right]$$

This keeps the shape of the high-valued pixels, but rescales them to within a value 100 higher than the new maximum. In my example image, I set $T=2400$.

Next, I convert the raw image into uint8(0->255) grayscale format in order to make compatibility with different functions easier, and to make easy comparisons to the 'official' preprocessed images which I flatten to grayscale as well. This is done by using MATLAB's *im2uint8.m* function along with the *imadjust.m* function. Fig. 2 shows the raw data in its original form, and then after the thresholding and

rescaling are applied to it.

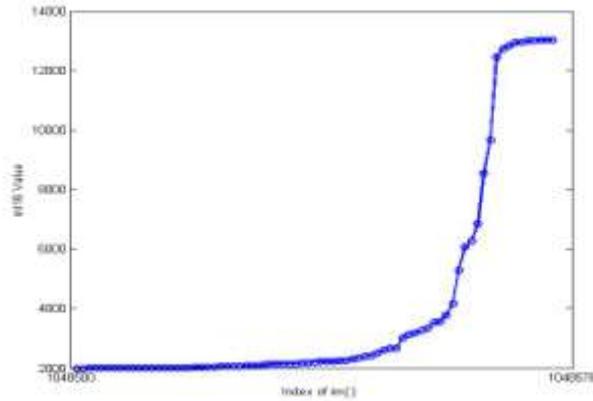


Figure 1 Intensity values of tail-end of unwrapped sorted image showing around 70 pixels whose intensities are very disproportionate to the others in the image.

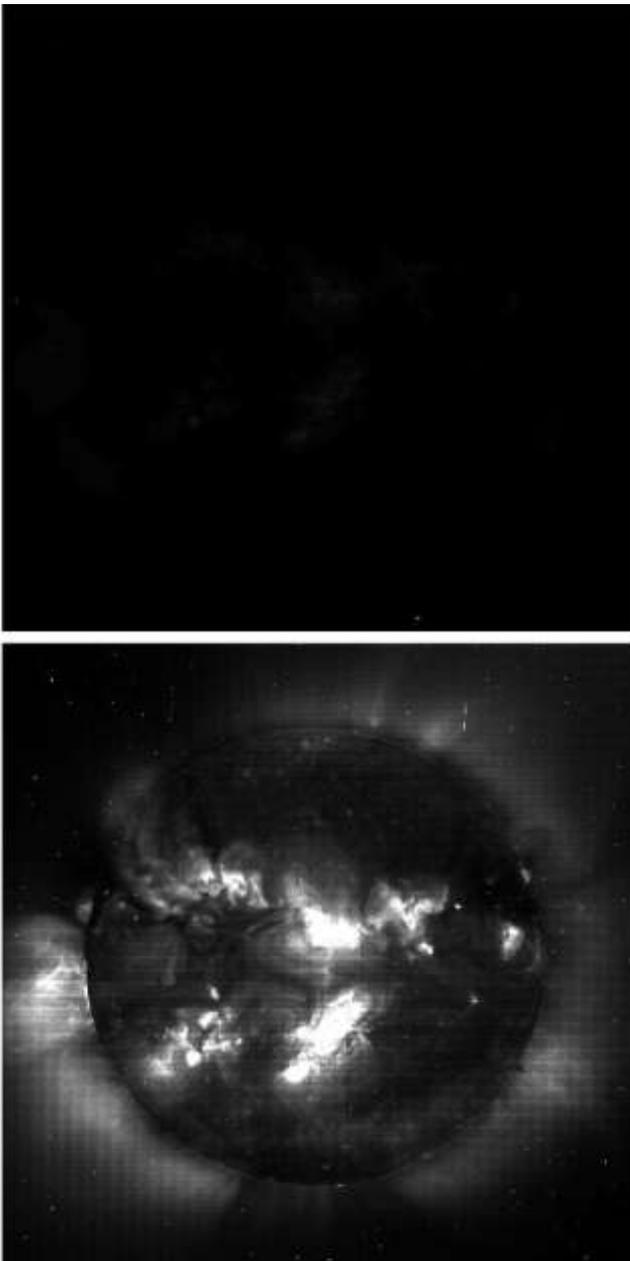


Figure 2 Top: Raw data image. Bottom: Image after thresholding and rescaling.

It can easily be seen that the image suffers from an artificial grid pattern caused by the EIT CCD imaging instrument.

C. Image De-gridding

There are many approaches to de-grid an image. One intuitive one is to manipulate the fast fourier transform (FFT) of the image. Since the grid pattern is periodic with a specific period, it should show up as spikes in the FFT of the image. Indeed, the FFT of my sample image as depicted in Fig. 3 shows a series of spikes as expected.

In order to eliminate these spikes, I developed a manual notch-filter program. This program allows the user to select a point on the FFT (computed using *fft2.m*) and then a threshold value is computed using the maximum value in a radius, r , around the chosen point. Any point within r that is above the threshold is either set to 0 or some fraction of the threshold value. The image is then recovered using *ifft2.m*. For my image, I use:

$$T = 0.1 \max[\text{im}(x - r : x + r, y - r : y + r)],$$

and set the thresholded values to 0. Fig. 4 shows my degrided image along with the ‘official’ processed image for comparison. In terms of the grid, it seems that my method has degrided the image to the same level as the ‘official’ image. However, the ‘official’ image seems to have a more equalized histogram, and therefore less contrast than my image.

III. THE A-TROUS ALGORITHM (STATIONARY WAVELET TRANSFORM) AND ITS USE IN IMAGE ENHANCEMENT

The A-trous algorithm, otherwise known as the Stationary Wavelet Transform (SWT) is an altered form of the classic Discrete Wavelet Transform (DWT). The difference is that instead of down-sampling the image at each level, the low-pass and high-pass filters are up-sampled by padding them with zeros in an interwoven manner (i.e. a zero is added to every other entry in the filter) [5], [6]. There are many filters one can choose from. In [3], a B3-Spline filter is used, while in my results, a biorthogonal 3.3 filter is used. A schematic of the a-trous algorithm is given in Fig. 5.

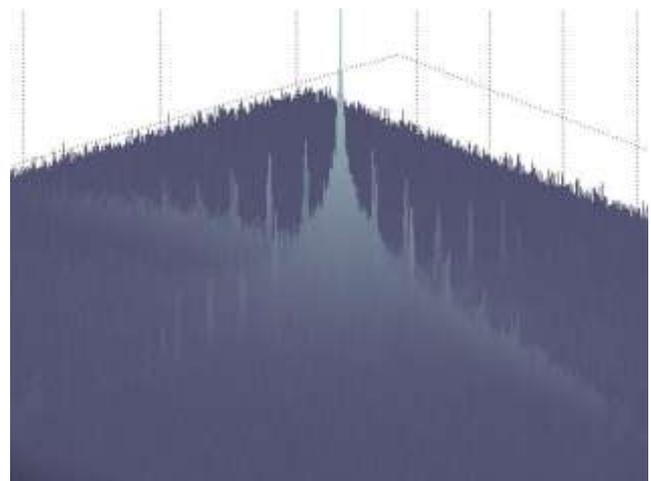


Figure 3 Surface plot of FFT of sample image. Spikes corresponding to the grid pattern are discernable.

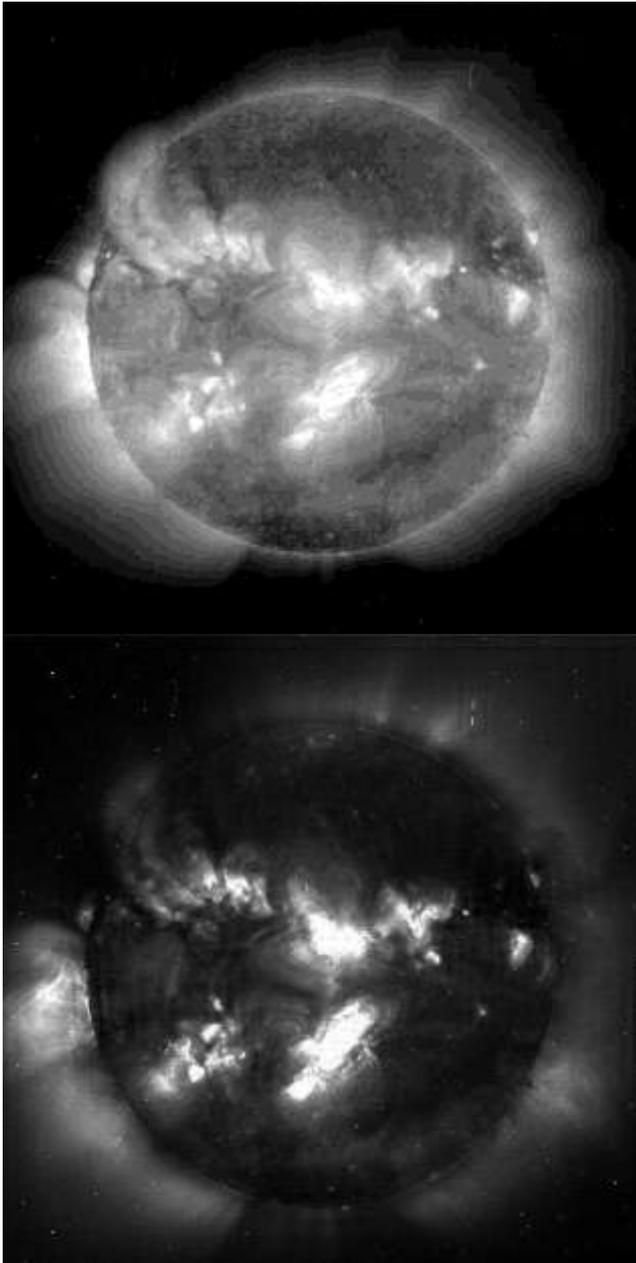


Figure 4 Top: 'Official' processed image from EIT. Image courtesy of SOHO/[EIT] consortium. SOHO is a project of international cooperation between ESA and NASA. Bottom: My rescaled image after degriding using manual notch filter.

The process creates three N by N detail coefficient matrices along with an N by N approximate (blurred) image at each level. Each new level is formed by performing the next level transform on the approximate image at the previous level. The coefficients for a level 1 decomposition of my test image are shown in Fig. 6.

To enhance images, one can re-weight the detail coefficients of each level and then compute the inverse SWT to bring out structure that may not be apparent in the original image. The resulting image will differ greatly depending on the weighting chosen. One issue that [3] mentions is that when such details are brought out using a-trous, that noise and grid patterns may be enhanced as well. This effect can be seen in my results.

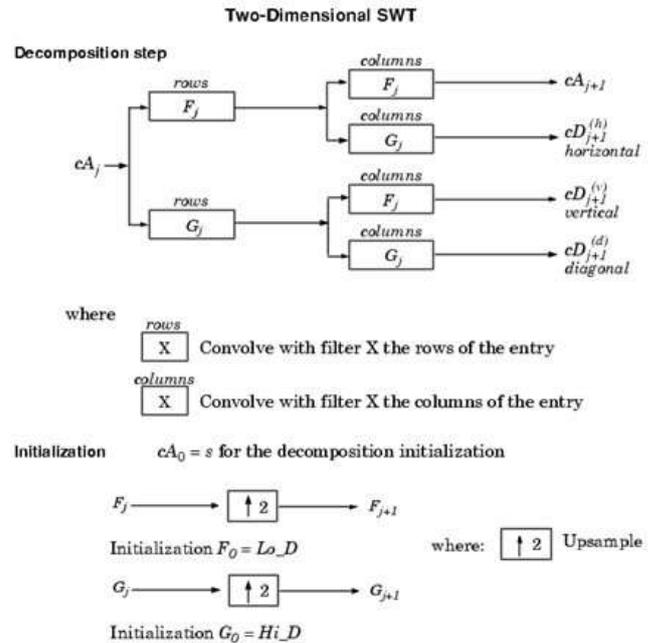


Figure 5 Schematic diagram of the a-trous algorithm. Image courtesy of MATLAB help file for swt2.

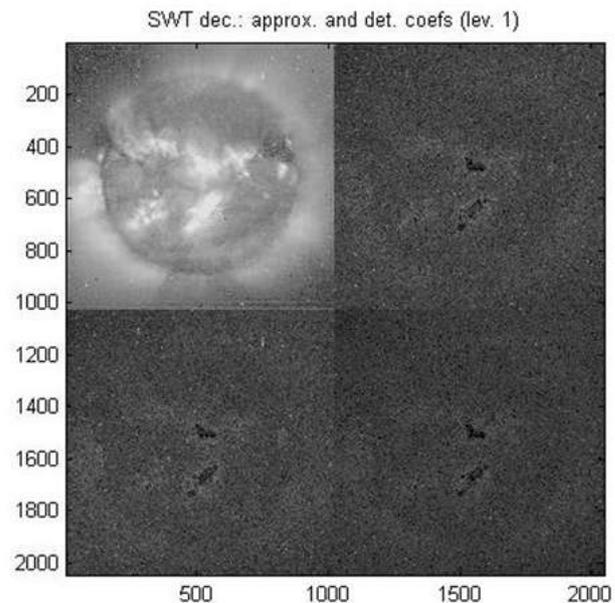


Figure 6 Wavelet coefficients for a 1-level decomposition of the test image using the filter 'bior3.3'.

IV. MY RESULTS

To enhance my test image, I use a five-level a-trous decomposition, with a detail coefficient weighting of $w = [2 \ 5 \ 4 \ 3 \ 2]$, where w_i is the weight given to the detail coefficients of level i . My results starting with the 'official' image and also using my degrided image are shown in Fig. 7. It can be seen that starting with each image, the a-trous process brings out structures much clearer than in the original image. The enhanced noise can also be observed. In [3], this process is enhanced to eliminate the noise problem.

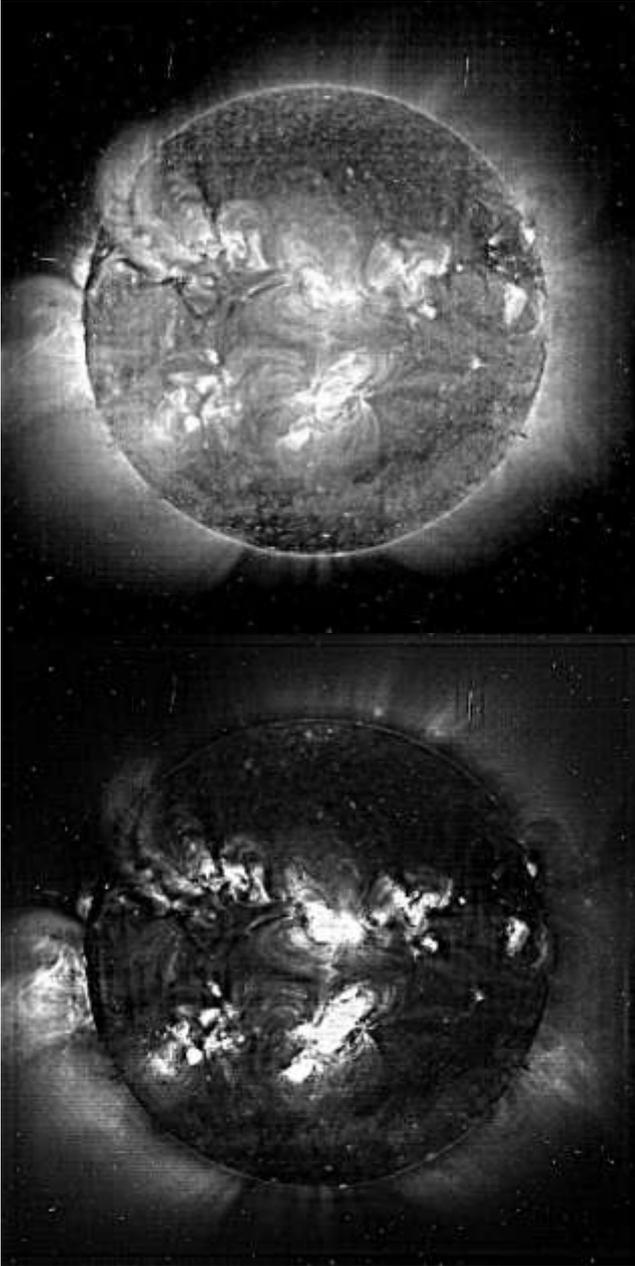


Figure 7 Result of a-trous weighting reconstruction using five levels and weighting of $w=[2\ 5\ 4\ 3\ 2]$. Top: Result starting with the 'official' image. Bottom: Result using my degraded image.

V. FULL ALGORITHM AND RESULTS FROM PAPER

In [3], the authors develop a way to prevent the a-trous algorithm from enhancing noise and gridding. They formulate noise masks, W , from taking the median of 4 weeks' worth of median images for each desired level of the a-trous wavelet. In addition to this, they develop a model for the scattered light residual, R , by again taking 4 weeks' worth of median images, but taking the minima, instead of the medians. With these two tools, they formulate their full algorithm as follows:

$$I_k^{\text{treated}} = [\log(I_k > 0) - \log(R^i > 0)] + \sum \alpha_j (w_j - W_j^i)$$

where the alphas are the weight coefficient scales. Using this algorithm on a sample image (not the same image I used), and for weight scaling of $\alpha = [1\ 30\ 5]$, they get the results depicted in Fig. 8.

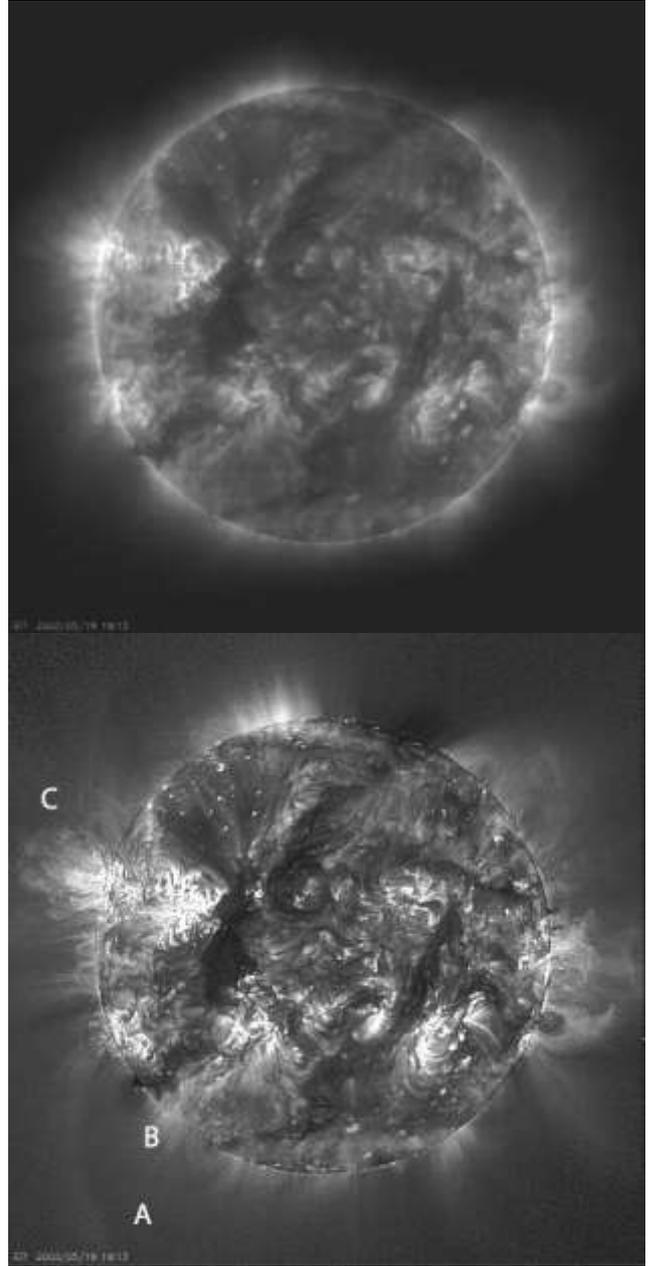


Figure 8 Results from using the a-trous algorithm developed in [3]. A, B, and C are structures not readily seen in the original image.

VI. CONCLUSION

In conclusion, I see that the papers algorithm does an excellent job at bringing out solar features not readily seen in the original EIT images. However, from my own programs, I see that using a similar procedure without the use of residual light models or noise masks, and using a weaker filter can produce results that mimic somewhat the full algorithm in terms of bringing out structural detail.

ACKNOWLEDGMENT

I would like to thank Dr. Angelos Vourlidis and Guillermo A. Stenborg for offering to send me images and filters, and giving me links to data and suggestions. I also would like to thank Professor Morris for advice on degridding techniques.

REFERENCES

- [1] SOHO Official Website, <http://sohowww.nascom.nasa.gov/>
- [2] Official EIT Website, <http://umbra.nascom.nasa.gov/eit/>
- [3] G. Stenborg, A. Vourlidas, and R. A. Howard, "A fresh view of the extreme-ultraviolet corona from the application of a new image-processing technique" *The Astronomical Journal*, vol. 624, Feb. 2008, pp. 1201-1206.
- [4] The Virtual Solar Observatory, <http://sdac.virtualsolar.org>
- [5] MATLAB swt2 Help File, <http://www.socsci.umn.edu/doc/matlab/toolbox/wavelet/swt2.html>
- [6] G. P. Nason and B. W. Silverman, "The Stationary wavelet transform and some statistical applications" *Lecture Notes in Statistics*, 103, pp. 281-299.