Zvezda

Jaime Ernesto Forero Romero
Nicolas Rocha Pacheco

2016

Contents

List of Figures

1 Introduction

Figure 1: This image was used to test if the algorithm was working

2 Algorithm

Since we have been looking for a way to detect the filamentary structures in the sun we had to start with an image of the sun and then do some analysis of that image through the algorithm. The mathematical analysis of the image was done keeping in mind an algorithm developed and previously used by Professor Forero to find filamentary structures on sun's chromosphere. Since we are working with an image we must use values for each pixel as they were part of an associated function $F(x, y)$ even we can not define them in x or y. The first step was calculating the 2×2 Hessian matrix for each pixel as it follows:

$$H = \begin{bmatrix} \dfrac{\partial^2 F}{\partial x^2} & \dfrac{\partial^2 F}{\partial x \, \partial y} \\ \dfrac{\partial^2 F}{\partial y \, \partial x} & \dfrac{\partial^2 F}{\partial y^2} \end{bmatrix} \tag{1}$$

The problem was that as the associated function $F(x, y)$ can not be differentiated with common mathematical rules as it is only defined on integers, we had to use the following numerical approximations of second order derivatives:

$$\frac{\partial^2 F}{\partial x^2} = \frac{F(x+h, y) - 2F(x, y) + F(x-h, y)}{h^2} \tag{2}$$

$$\frac{\partial^2 F}{\partial y^2} = \frac{F(x, y+j) - 2F(x, y) + F(x, y-j)}{j^2} \tag{3}$$

$$\frac{\partial^2 F}{\partial x \partial y} = \frac{F(x+h, y+j) - F(x+h, y-j) - F(x-h, y+j) + F(x-h, y-j)}{4hj} \tag{4}$$

Since we are working with differences between pixels h and j are 1 for every numerical approximation except for the pixels from the edge as we can not calculate the derivatives as those pixels do not have surrounding pixels. Once we have the Hessian matrix for each pixel we are encouraged to calculate the eigenvalues of this matrix. During the development of the algorithm we plotted the derivatives as we were able to calculate them.

This is a summary of all the steps in the algorithm:

1. **Hessian Matrix** The first step was calculate the Hessian matrix for each pixel, so we used finite derivatives as a numerical method to approximate second-order derivatives. As each pixel had it's own Hessian matrix we had to save each Hessian matrix in another matrix that was used later to do some calculations. The Hessian matrix calculation was defined as a function inside the Python script that used the FITS image data as parameter and resulted in a matrix containing the Hessian matrix for each pixel.
2. **Eigenvalues** Once we had the Hessian matrix for each pixel, we calculated the eigenvalues of each Hessian matrix. With those eigenvalues we created a new matrix where each pixel had it's two associations. Then we made two graphs from that matrix, one containing the first eigenvalue and a second one containing the second eigenvalue. This step was also defined as a function inside Python where the parameter was each Hessian matrix.
3. **Histogram** Our final step was plotting the histograms of each eigenvalues, also, we plotted both of them in the same graphic. Finally we took different sections of the image we used to see the difference between different patterns.

3 Results

3.1 Full image analysis

The first test we made was passing the whole image through the algorithm. We started by cutting the edges out of the image as can be seen in Figure **??**. As we were not interested on analyzing the sunspot, this was just a test on the eigenvalue function.

Figure 2: Part of the image that was analyzed

Figure 3: Resulting image from calculating the first eigenvalues

Figure 4: Resulting image from calculating the second eigenvalues

3.2 Lower-left corner

Figure 5: Resulting image from calculating the first eigenvalues

Figure 6: Resulting image from calculating the first eigenvalues

Figure 7: Resulting image from calculating the second eigenvalues

4 Future work

5 References