Fitted Lines to Bundles

Jeren Suzuki

Last Edited August 8, 2013

Contents

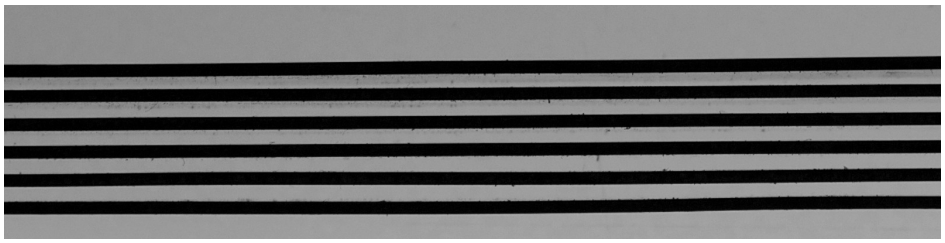
1 Introduction

With pictures taken with my camera, then fitted lines to edges of slats.

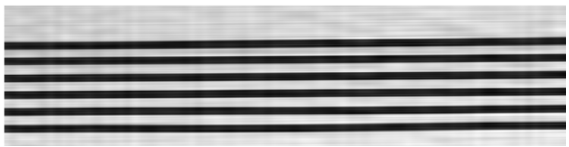
2 Starting Image

Figure 1: Starting Image, red channel

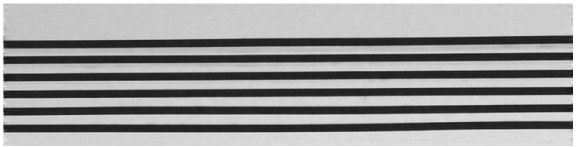
Before we threshold the image to isolate the darkened slats, we have to de-noise the image. There is an option to de-noise the image in third-party post-processing software but it *should* be done in IDL for the sake of completeness and control over data manipulation. FFT filtering is used, but it takes a long time. An option is to crop a region out of just the slats so that the analysis can be done quicker.



(a) No filtering



(b) FFT'd on whole image, then cropped



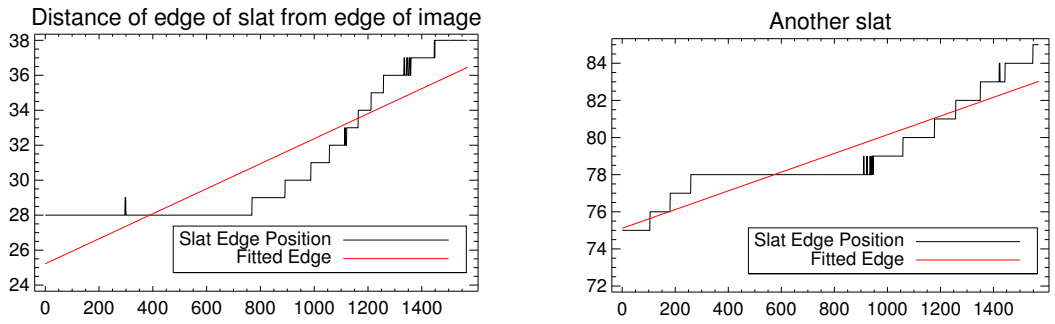
(c) Cropped, then FFT'd

Figure 2: FFT Test

As you can see, using the FFT operator on the whole image resulted in some ringing and artifacts in the cropped region.

3 Fitted Lines

With a filtered image, I set a threshold to find the edges of the slats. The process is scalable to any number of slats. In Figure 3a, a particularly curved edge of a slat results in a weird shape but upon closer inspection, the variance from the edges of the slat only differs by 10 pixels. With all slats edge-fitted, a color check on the processed image is performed, as in Figure 4.



(a) Linear fitting the edge of a slat

(b) Linear fitting the edge of another slat

Figure 3

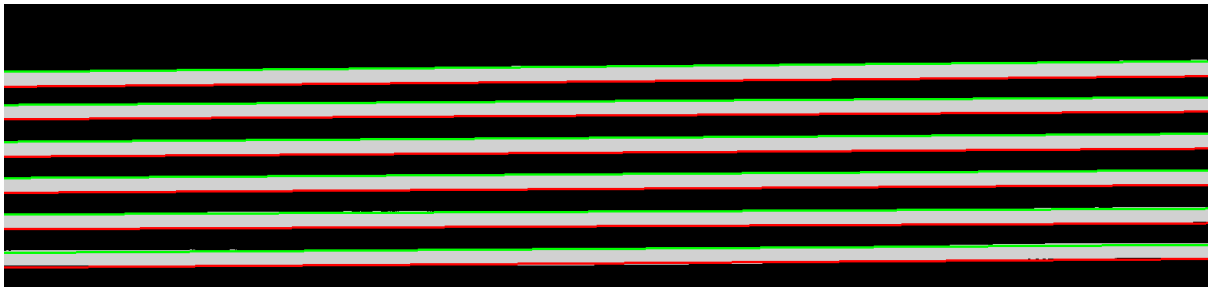


Figure 4: Cropped region of slats with edges of slats marked in color

4 Bundle Analysis Steps

To start off with, I use FFT filtering to get rid of the dust and high spatial noise in our image. Once I get Figure 2c, I take vertical slices and threshold the slats below a certain value. In Figure 5, the top plot illustrates the slat structure while the bottom plot is thresholded to values lower than 20.

With these edges, I find where values of `array - shift(array,1)` equal 1 and where `array - shift(array,-1)` equal -1, which correspond to the edges of each pillar structure. These positions are stuck into two 6xN length matrices, one for each edge of the slat. Now I move on to the next column, do a `shift()` check and append the next 6 pairs into the 6xN arrays. Once finished, I fit each row of the 6xN array to a line and overplot the line position into the 2D starting image, as per Figure 4.

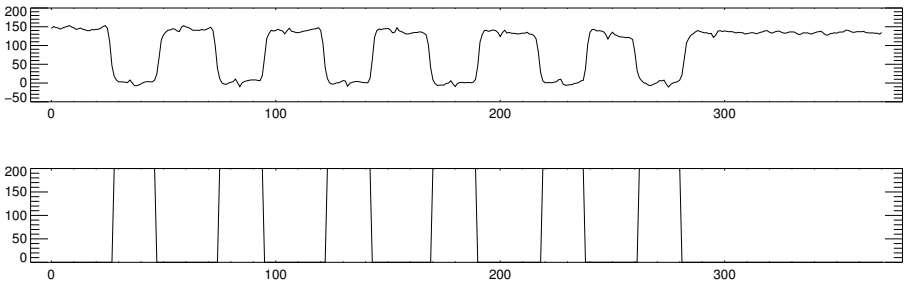


Figure 5