

Some Notes on Albert's Code

Jeren Suzuki

Last Edited May 21, 2013

Contents

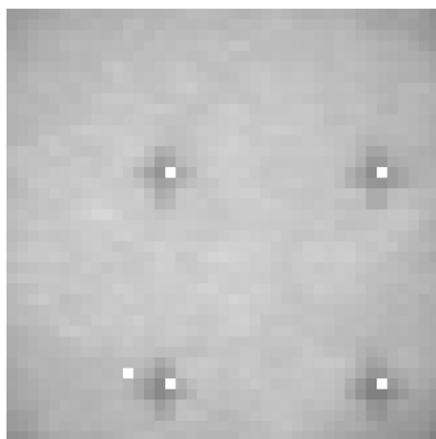
1	Introduction	1
2	Rough Center vs Sub-Pixel Center	1
3	Information Depth	1
4	Cut off Fiducials	2

1 Introduction

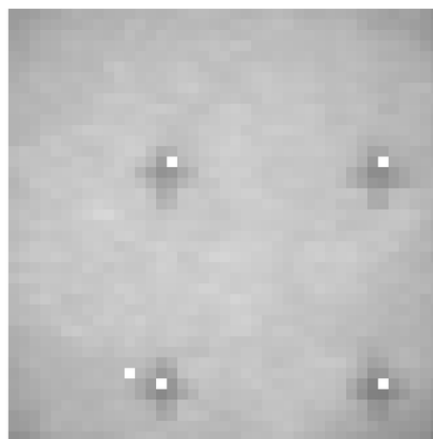
Using Albert's C++ as a guide, I've integrated a few of his procedures into IDL. One of the major improvements was the fiducial-finding software. Instead of using two edge-detection filters, we instead use a single convolution filter and check local maxima, then fit them to a parabola for a sub-pixel centroid.

2 Rough Center vs Sub-Pixel Center

If we fit a parabola to the local maxima, we can get sub-pixel values for the center, although their accuracy is questionable. See Figure 1. The location of the fiducials just happen to be directly over the pixels so it looks like the not-sub-pixel-method is better, but we need some other solar images to test it on.



(a) The center of the convolution filter's local maxima



(b) The center of the parabolic fit to the local maxima.

Figure 1 Ignore the white pixel near the bottom left fiducial, that is a bright outlier pixel meant for debugging purposes. It is not characteristic to a fiducial.

3 Information Depth

Albert approaches approximate center-finding in a different way:

1. Find pairs of rising/falling edges
2. Remove pairs if along fiducial
3. Remove pairs if too close to limb
4. Remove pairs if across fiducial
5. If sun is cut off, create artificial rising/falling edge to force pair
6. Fit chord edges with a linear fit
7. Find midpoints of edge-fit chords

There are several major differences than our code. Instead of finding N chords centered around the known approximate center, the chords are made at `n_pix` intervals in the entire solar cropped image. This incorporates the need to check if the chord is too close to a limb, which is a step we don't have. The choice of a linear fit instead of a 2nd order polynomial is probably for the sake of speed.

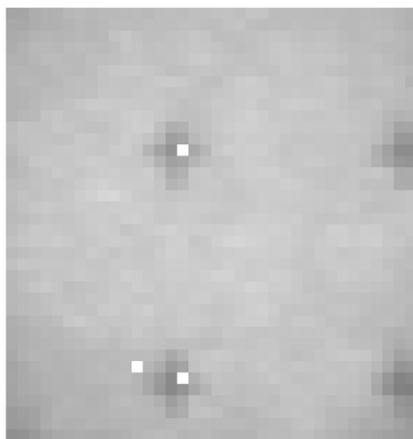
Albert's code takes into account the scenario where the sun may be cropped out of part of our image. Our current approach is to use the approximate masking method to find the rough center and leave it at that. We don't use the chord-centering approach if the sun we are looking at is cut off. Not sure if we need to add this condition in our code, but it should be noted.

Albert chooses to remove the chords if they are too close to a fiducial, a step that I *should* incorporate rather soonish. The "problem" that arises is that since we have a fixed number of chords, removing a chord may be a problem. It will be problematic if, say, we have 5 chords in 1 axis and 2 in another. The chord centers may be biased. Albert's choice to simply remove the chords, however, leads me to believe that this bias is acceptable.

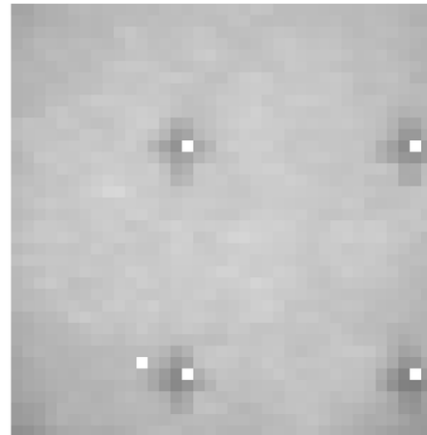
4 Cut off Fiducials

The acid test for the new method of centroiding the convolution will be if it can properly detect cut off fiducials.

At first, there was a problem with the check-if-your-neighbor-is-higher-than-you part of the code incorrectly finding local maxima strewn throughout the image. The problem was rectified by modifying the threshold value to a stricter value. The result is the fiducial finding code requires 5 fiducial pixels to be inside the image, but other than that, it robustly finds them. See Figure 2.



(a) The raw image used for fiducial finding



(b) If 5 of the 6 fiducial pixels are inside the image, the center is found

Figure 2 Again, ignore the bright pixel near the bottom left fiducial.

The convolution filter already has the `/edge_truncate` keyword to try and deal with edge fiducials and I even tried padding the image but the latter produced some artificial ringing artifacts strewn throughout the image. Hopefully this will be the last we deal of cut off fiducials.