Gain Calibration for the Precision Solar Photometric Telescope at the Mauna Loa Solar Observatory

R.W. Meisner, M.P. Rast, and T. Holzer
High Altitude Observatory, National Center For Atmospheric Research, Boulder, CO

1. Introduction

Since the introduction of the Kuhn, Lin, & Loranz algorithm in 1991 for computing a gain calibration image using only the image data, there have been many variations and improvements on this technique, depending on desired accuracy and rate of convergence (e.g., Wild 1997; Dalrymple, Bianda, & Wiborg 2003; Toussaint, Harvey, & Toussaint 2003; Chae 2004). At the High Altitude Observatory (HAO), we have developed a new variation on the procedure for the Precision Solar Photometric Telescope (PSPT) at the Mauna Loa Solar Observatory to achieve a relative accuracy of 0.1%, and an increased rate of convergence in comparison to the Kuhn, Lin, & Loranz 1991 algorithm. The new HAO algorithm implements a simple iterative refinement technique on the image scale while computing the gain for each iteration on a pixel by pixel basis.

2. Algorithm

The HAO algorithm has several advantages and improvements over previous implementations: it achieves the relative accuracy required by the PSPT, has a fast rate of convergence, does not leave residual artifacts in the computed gain image (i.e., linear gradient), and is simple to implement. Given a sequence of images, offset around the field of view of the detector, an outline of the procedure is as follows:

1. Given a reference image from the sequence of offset images, determine the pixel offset of each sequence image, relative to the reference image.
2. For each frame of the flat-field sequence, determine an activity mask which masks out pixels with strong gradients, or intensity which is outside the linear range of the detector.
3. Shift the activity masks to the reference image coordinates.
4. Normalize the intensity values of each offset image (divided by the current gain image) relative to the reference image.
5. Read in all of the raw offset images into a data cube, divided by the current gain image (begin with the gain image set to a value of one for each pixel), and shifted to the reference image coordinates.
6. Step through the data cube in (x, y) coordinates and for each (x, y) pixel coordinate determine from the activity masks if there is a valid pixel from any of the sequence images (scan the z component of the data cube at each (x, y) pixel coordinate).
7. For each (x, y) pixel coordinate with valid pixels, extract the column of valid data pixels (in the z direction) from the data cube and perform the following:
   • Compute the mean of the data pixels.
   • Compute the gain of each data pixel relative to the mean.
   • Store the number of data pixels used in computing the gain for a weighted mean later
in the algorithm.

- Store each computed pixel gain in a separate data cube.

8. For each z-level gain image, shift back to its original data offset coordinate.
9. For each pixel in the (x, y) dimensions of the gain data cube, compute the weighted mean of
the computed gains in the z-direction and store in a two dimensional gain correction image
array.
10. Multiply the gain correction by the previous gain to obtain a new gain image.
11. Check for stopping criteria – If not met, repeat from step one.

The algorithm amounts to the following. The intensity of a given location on the Sun is sampled
by from 1 to N different pixels using N offset images. Gain values are determined for each pixel
so that the gain times the intensity is equal to the mean intensity measured for that location.
Since the same detector location samples different locations on the Sun in the offset images, this
procedure yields more than one gain value for a given pixel. The weighted (by the number of
images contributing to a particular gain value) average value of the gain is used to correct the
offset images and the procedure is repeated. Iteration proceeds until the the standard deviation of
the difference between subsequent gain corrections changes by less than a specified amount.

3. Implementation & Testing

In order to check the validity, accuracy, and rate of convergence of the HAO algorithm, we
generated a fake gain image and fake full disk solar data. The gain image shows large scale
variation with random pixel to pixel noise add to it, both at levels much greater than a typical
CCD (Figure 1). The simulated full disk solar data was generated by smoothing an average of
numerous limb darkening profiles of the PSPT blue continuum images. Copies of the simulated
solar image were then offset around the field of view and multiplied by the fake gain image to
produce our simulated raw flat-field sequence data. Figure 2 shows the simulated raw reference
image used in the flat-field sequence.

![Figure 1: Simulated gain image used in testing the HAO flat-field algorithm](image1)

![Figure 2: Simulated raw data image used in testing the HAO flat-field algorithm.](image2)
As mentioned by Kuhn, Lin, & Loranz 1991, and stressed by Toussaint, Harvey, & Toussaint 2003, the accuracy and the rate of convergence of this type of algorithm are both increased if the displacements between sequence frames do not contain any common factors, thereby “linking” all adjacent pixels. Therefore, the HAO algorithm was tested with two flat-field sequences: one with common factor offsets, and one with prime number offsets. For the algorithm testing, we used image arrays of size 2048x2048, the same as produced by the PSPT, and 18 offset images with the ninth image being the centered reference image (the PSPT currently uses 16 offset images).

3.1 Results

Figure 3 shows the relative accuracy of the common factor offset sequence and the prime number offset sequence for the HAO algorithm. The relative accuracy shown here is the standard deviation of over 3200 pixel to pixel ratios throughout the image, compared to the same ratios in the original gain image.

To further stress the importance of using non-common factor offsets, we have constructed surface plots of the relative error of the HAO algorithm for the common factor and prime factor flat-field sequences after 25 iterations. Figure 4 is the relative error surface plot for the sequence with the common factor offsets and illustrates the periodic regions of unacceptable high error values. The white area is where the relative error is less than, or equal to, 0.1%, and the areas in black are where the relative error is greater than 0.1%. Figure 5 is the same surface plot for the prime number offset sequence and shows considerably better accuracy across the entire array, and demonstrates the confinement of the greatest error to locations where there is little or no overlap of the offset images. Figure 6 is the same as Figure 5, but with the segmentation level set to 0.05%.
Figure 4: Relative error surface plot for the flat-field sequence with common factor offsets. The white area is where the relative error is less than, or equal to, 0.1%, and the black areas are where it is greater than 0.1%.

Figure 5: Relative error surface plot for the flat-field sequence with prime number offsets. The white area is where the relative error is less than, or equal to, 0.1%, and the black areas are where it is greater than 0.1%.

Figure 6: Relative error surface plot, segmented at 0.05%, for the prime number offset flat-field sequence.
References


