SUBMITTED TO APJ (NOVEMBER 14, 2005)
Preprint typeset using LaTeX style emulateapj v. 11/26/04

A METHOD TO REMOVE FRINGES FROM IMAGES USING WAVELETS

PATRICIO M. ROJO
516 Space Sciences Bldg, Center for Radiophysics and Space Research
Cornell University, Ithaca, NY 14853-6801

AND

JOSEPH HARRINGTON
326 Space Sciences Bldg, Center for Radiophysics and Space Research
Cornell University, Ithaca, NY 14853-6801
Submitted to ApJ (November 14, 2005)

ABSTRACT

We have developed a new method that uses wavelet analysis to remove interference fringe patterns from images. This method is particularly useful for spectroscopic flat fields in the common case where fringes vary between the calibration and object data. We analyze the efficacy of this method by creating fake flats with fictitious fringes and removing them. The method removes 90% of the fringe pattern whose amplitude is equal to the random noise level and 60% if the fringe amplitude is $\approx 1/10$ of the noise level. We also present examples using real flat field frames. A routine written in the Interactive Data Language (IDL) that implements this algorithm is available from the authors and as an attachment to this paper.

Subject headings: methods: data analysis — technique: image processing

1. INTRODUCTION

Images in astronomical detector arrays sometimes contain unwanted fringe patterns that are produced by the interference of light reflecting between parallel surfaces in the instrument. Usually, the fringe pattern is ignored with the hope of correcting it when dividing by the flat field. However, flexure and variations in the illumination geometry (e.g., from movable instrument parts) can change the fringe pattern. Hence, observations requiring high sensitivity necessitate explicit removal of these patterns. Fringe correction methods found in the literature are either specific to the instrument or assume a global fringe period (e.g., Malumuth et al. 2003a,b; Mellau & Winnewisser 1995). However, in most cases the pattern's period varies over the image, making global techniques like Fourier filtering less than satisfactory.

Here we present an algorithm that uses the wavelet transform, a local spectral technique (e.g., Starck & Murtagh 2002; Torrence & Compo 1998). This transform is linear. Hence, we can isolate the fringe pattern in wavelet space, do an inverse transform, and then obtain a clean image by subtracting this reconstructed fringe pattern. The challenge is to do this correctly in the presence of noise.

We implemented our algorithm in the Interactive Data Language (IDL, a product of Research System Inc., Boulder, Colorado). The code (named "defringeflat") is not specific to any instrument and includes tutorial documentation. It is available under the GNU General Public License from our websites[1] or as an electronic attachment to this paper.

Section 2 describes the algorithm. Section 3 discusses performance in the presence of noise. Finally, Section 4 summarizes the benefits and limitations, and presents our conclusions.

2. ALGORITHM

Electronic address: pato@astro.cornell.edu
Electronic address: jh@oobleck.astro.cornell.edu
[1] http://www.das.uchile.cl/∼pato/sw/ or http://oobleck.astro.cornell.edu/jh/ast/software.html

The main steps in our procedure are listed in Table 1. Figures 1 – 7 illustrate the steps of the algorithm using an example flat field. Their captions contain details regarding the example array, while the main text only refers to the algorithm in general. The example flat field is included in the defringeflat package.

We assume that the change in fringe period is smooth across the columns. The period must be at least several pixels, but may not exceed $\sim 1/4$ of the image width. Prior to applying the algorithm, all borders whose values are not consistent with the image must be cropped, and the image should be oriented such that the nodes of the fringe pattern align mostly with the columns.

STEP I. ENHANCED ROW AND WAVELET TRANSFORM

For each image row we combine several surrounding rows to suppress random noise and remove bad pixels. To do this, we replace each pixel in the row with the median of a $1 \times n$ subimage centered on the pixel and traversing n rows (bin width, hereafter). We then subtract a linear fit from the median-averaged row to obtain an *enhanced row* (Figure 2). This significantly lowers the wavelet amplitudes at small periods, allowing the next step to proceed more efficiently.

We then compute the wavelet transform of each enhanced row. There are several wavelet basis functions to choose from, but this algorithm requires a complex basis. We used the Morlet wavelet because its functional form makes it well suited for smoothly varying periods and because it is compact in the frequency domain. The result for each row is a two-dimensional (2D), complex array, whose two dimensions are column number and period. Steps II and III are computed over the complex array amplitudes (wavelet array, hereafter). The phases of the complex array must be stored for use in step IV. We now have 3 dimensions: row, column, and period.

STEP II. PARAMETRIC FIT OF FRINGE TRANSFORM

At the period of the fringe pattern, the wavelet array will contain a prominent *fringe transform* pattern traversing the columns. Its amplitude depends on the amplitude of the fringe

pattern (Figure 3). This algorithm's success rests on our ability to distinguish this feature from the background noise level of the wavelet array. The fringe transform may vanish at particular positions, but it should be clearly distinguishable in most of the array. Improved sampling in period can be obtained by interpolating or by decreasing the spacing between discrete scales in the wavelet transform. The latter approach is more accurate but demands more computer resources. Hence, a compromise should be chosen.

We next fit the fringe transform (Figure 4). Starting from a reference column, the fringe profile is isolated by finding the first local minima on both sides of the reference period. Then, either the actual data (*trueshape*) or a parametric fit can be used to represent the profile within the minima. Only the latter approach will allow execution of the optional Step III, though. The values of the profile must be zero outside the fringe transform. Inside, on the other hand, it is recommended that the fringe transform profile exclude a background level (attributable to non-fringe image components). The highest point in the profile is used as the new reference period for the neighbor column. The procedure is repeated for the whole fringe transform, extending in both directions from the reference column to the cone of influence (COI) boundary, beyond which the wavelet values are contaminated by edge effects.

To fit the profile we have experimented with an extraction of the true shape of the profile, plain Gaussian fits with variable center (GVC), and Gaussian functions in which the center is fixed at the maximum height (GFC). Both Gaussian alternatives were considered without a constant background parameter (*noback*), and with this parameter. In the latter case, the background value can be kept or not when reconstructing (*keep* and *nokeep*, respectively). In total, we tried 6 parametric fits (that can be smoothed or not in step III) and 2 trueshape fits, for a total of 14 fits. The Gaussian shape is chosen not only because it is a natural choice to fit a peak, but also because it is the frequency-domain representation of the Morlet wavelet. The relative fringe-removal efficiency of these fits and of trueshape is discussed in Section 3.

STEP III. OPTIONAL PARAMETER SMOOTHING

If a functional parametric fit was used in the previous step, one can reduce the effects of noise by forcing the reconstructed fringe's parameters to vary smoothly. After repeating Steps I and II for every row, a 2D array is obtained for each of the fit parameters. First, we "patch" each of the parameter arrays by finding outliers that are beyond a given number of standard deviations from the neighborhood median and replacing them by that median value. Then, we smooth the array with a boxcar filter. Figure 5 shows the results.

STEP IV. RECONSTRUCTION OF FRINGE PATTERN

We next evaluate the parameters to obtain the fringe's wavelet amplitudes (Figure 3). Far from the reconstructed fringe transform, the amplitude must be zero, because any non-zero value there will cause unwanted noise in the reconstructed fringe. In particular, if a keep method is chosen, the reconstructed amplitude is set to zero outside the local minima. Finally, we apply an inverse wavelet transform to the reconstructed wavelet amplitude and the corresponding phases from Step I.

We repeat these steps for every row to obtain the image's isolated fringe pattern (Figure 6). Due to the optional smoothing, the method to obtain the enhanced rows, and the COI boundary, the recovered fringe pattern will have smaller borders than the original image. The fringe pattern can now be subtracted from the original image (Figure 7).

3. PERFORMANCE TESTS

The ratio of fringe pattern amplitude to the pixel-to-pixel variation (or noise) level varies among different instruments. We tested the algorithm's performance at different noise levels by using a synthetic image consisting of a fringe pattern, a background intensity, and random noise with a Gaussian distribution that mimics pixel-to-pixel flat-field variations and photon noise.

The fringe pattern was created using an analytic function that mimics the pattern in our example image. Its functional form is

$$F(x,y) = A\sin(\nu(x,y)x + \phi(y)),$$

where x,y are the position indices in the array, A is the amplitude (which we keep constant), and $\phi()$ and $\nu()$ are linear functions fitted to the phase and frequency, respectively, of our example's fringe. Note that there is no reason for A to be constant in a real image. The background level is a double-linear function in both x and y directions and has an edge taper.

We define *noise strength* as the standard deviation of the Gaussian noise divided by the standard deviation of the noiseless fringe pattern ($2^{-1/2}A$, due to its sinusoidal nature). Figure 8 shows the fraction of remnant fringe after running the algorithm on simulated data with different fitting functions and varying noise strength. The remnant fringe level is not strongly dependent on noise strength and all methods show very similar behavior with slight numerical differences when the noise strength is below ≈ 11 for noback methods and below ≈ 14 otherwise. However, GFC consistently gives the best results in all cases, even improving at high noise levels when smoothing. Above noise strength of ≈ 11 (or $approx14$), and only in some of the test runs, spikes begin to appear in the remaining fringe fraction for methods other than trueshape. Most of the methods remove over 95% of the fringe at noise strength ≈ 0.5 and over 55% at noise strength ≈ 9 (equivalent to Figure 1's noise strength). The lower plot of Figure 8 confirms the intuitive result that the method yields better absolute results for smaller initial fringe amplitudes.

Figure 9 shows the effect of varying the bin width. If the width is too small when computing the enhanced row, the noise is insufficiently suppresed. For low noise, a bin width that is too large will begin to average out the fringe.

The algorithm is mainly limited by the degree to which the analytic profile fit function mimics the data. Figure 10 shows an example of a difficult profile, which gives very different fits when using the different fitting functions. Another source of error is the potential for the algorithm to miss the correct trace in the presence of high noise in the wavelet array (Figure 11).

Due to the factors listed in Step IV, the reconstructed fringe pattern is smaller than the input data. For the example of Figure 1 this area is $\approx 85\%$ of the cropped input image, or over 90% if only considering the pixels lost for each surviving row, on average.

4. CONCLUSIONS

We have developed an algorithm that removes fringe patterns from 2D images while preserving other patterns. The algorithm is especially useful for cleaning flat fields if the fringe pattern varies between the flat and the object data.

TABLE 1
STEPS OF DEFRINGING ALGORITHM

Step	Description	Figure
	Original image with fringe	1
	FOR EACH ROW	
I	$\Rightarrow$ Compute enhanced row	2
	$\Rightarrow$ Compute wavelet transform	3
	$\Rightarrow$ FOR EACH PIXEL IN ROW	
II	$\Longrightarrow$ Fit fringe transform's profile	4
	FOR THE WHOLE ARRAY	
III	$\Rightarrow$ Smooth fit parameters (optional)	5
	FOR EACH ROW	
IV	$\Rightarrow$ Reconstruct wavelet array	3
	$\Rightarrow$ Inverse transform	3, 6
	FOR THE WHOLE ARRAY	
	$\Rightarrow$ Subtract fringe pattern to obtain clean image	7

There are three main limitations of this procedure. First, the shape of a fringe in wavelet space may be much more complicated than any reasonable fitting function, resulting in a partially-corrected fringe. Second, to be able to follow the trace, the change in the fringe's period must be smooth. Finally, there is a region along the borders where the fringe pattern cannot be recovered.

A potential for improvement exists if a method can be found to fit the entire fringe transform pattern simultaneously in the 3D wavelet space of row, column and period. The 2D wavelet transform may be more appropriate for this approach.

Our IDL implementation of this algorithm and its documentation appear as an electronic supplement to this article. Updated versions are available on the authors' websites or by email request.

This material is based upon work supported by the National Aeronautics and Space Administration under Grant No. NAG5-13154 issued through the Science Mission Directorate. The example flat field was obtained from public archives of the European Southern Observatory.

Facilities: VLT:Antu (ISAAC)

REFERENCES

Malumuth, E. M., Hill, R. J., Cheng, E. S., Cottingham, D. A., Wen, Y., Johnson, S. D., & Hill, R. S. 2003a, in Proceedings of the SPIE., Vol. 4854, 567

Malumuth, E. M., Hill, R. S., Gull, T., Woodgate, B. E., Bowers, C. W., Kimble, R. A., Lindler, D., Plait, P., & Blouke, M. 2003b, PASP, 115, 218

Mellau, G. C. & Winnewisser, B. P. 1995, in ASP Conf. Ser. 81: Laboratory and Astronomical High Resolution Spectra, 138

Starck, J.-L. & Murtagh, F. 2002, Astronomical image and data analysis (Berlin: Springer)

Torrence, C. & Compo, G. P. 1998, Bull. of the Am. Meteor. Soc., 79, 61

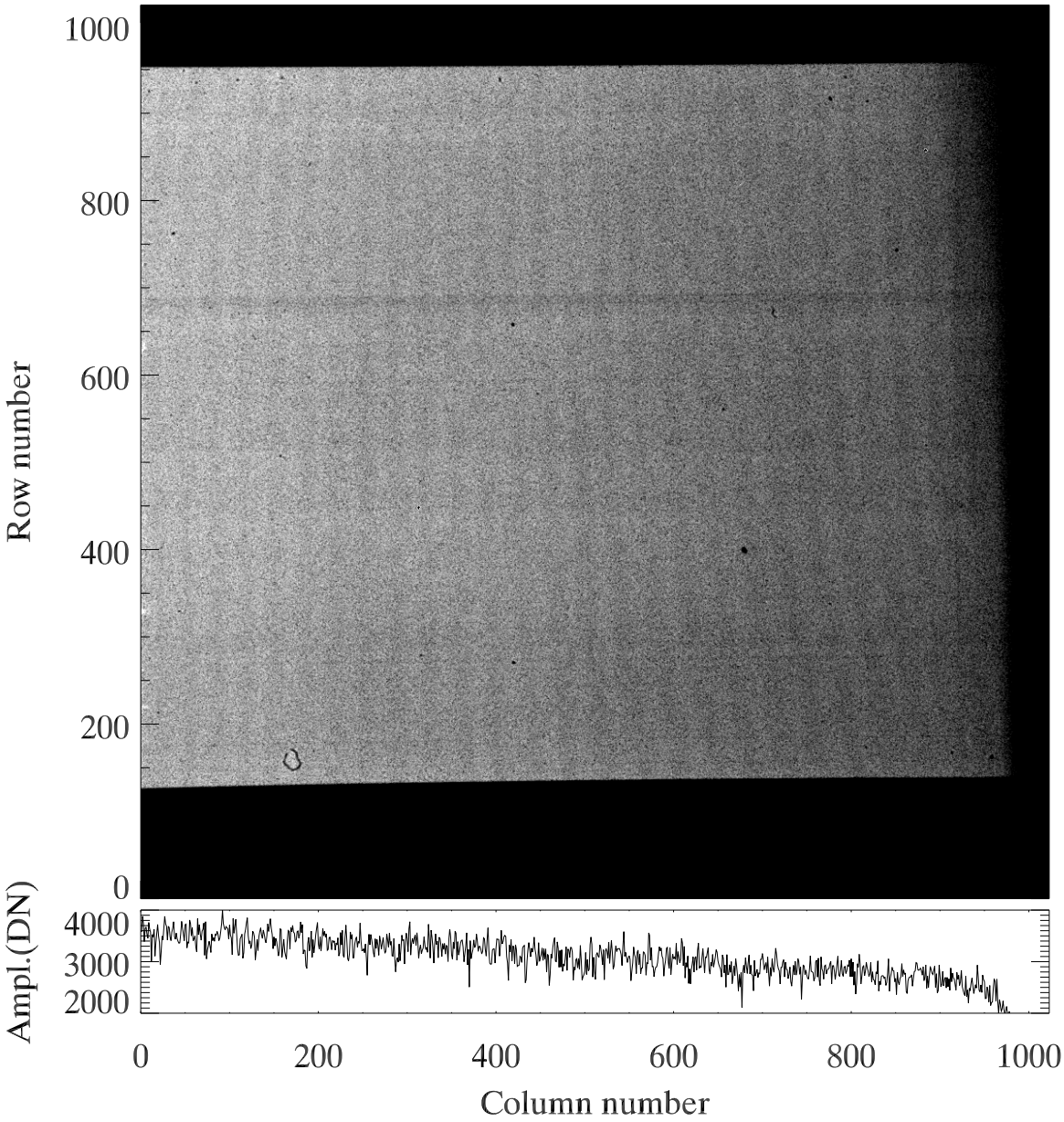

FIG. 1.— Top: Sample image with fringes. Flat field was obtained with the ISAAC instrument at the VLT. Each of the numeric parameters indicated in the captions from Figures 1 to 7 were found to be the most appropiate for this particular example, but will need to change for different images. Columns 901 to 1024, and rows 0-149 and 951-1024, are vignetted and thus are cropped before analysis. Periodicity can be estimated by eye at $\sim$ 40 pixels in the center of the image. Bottom: Middle (512th) row.

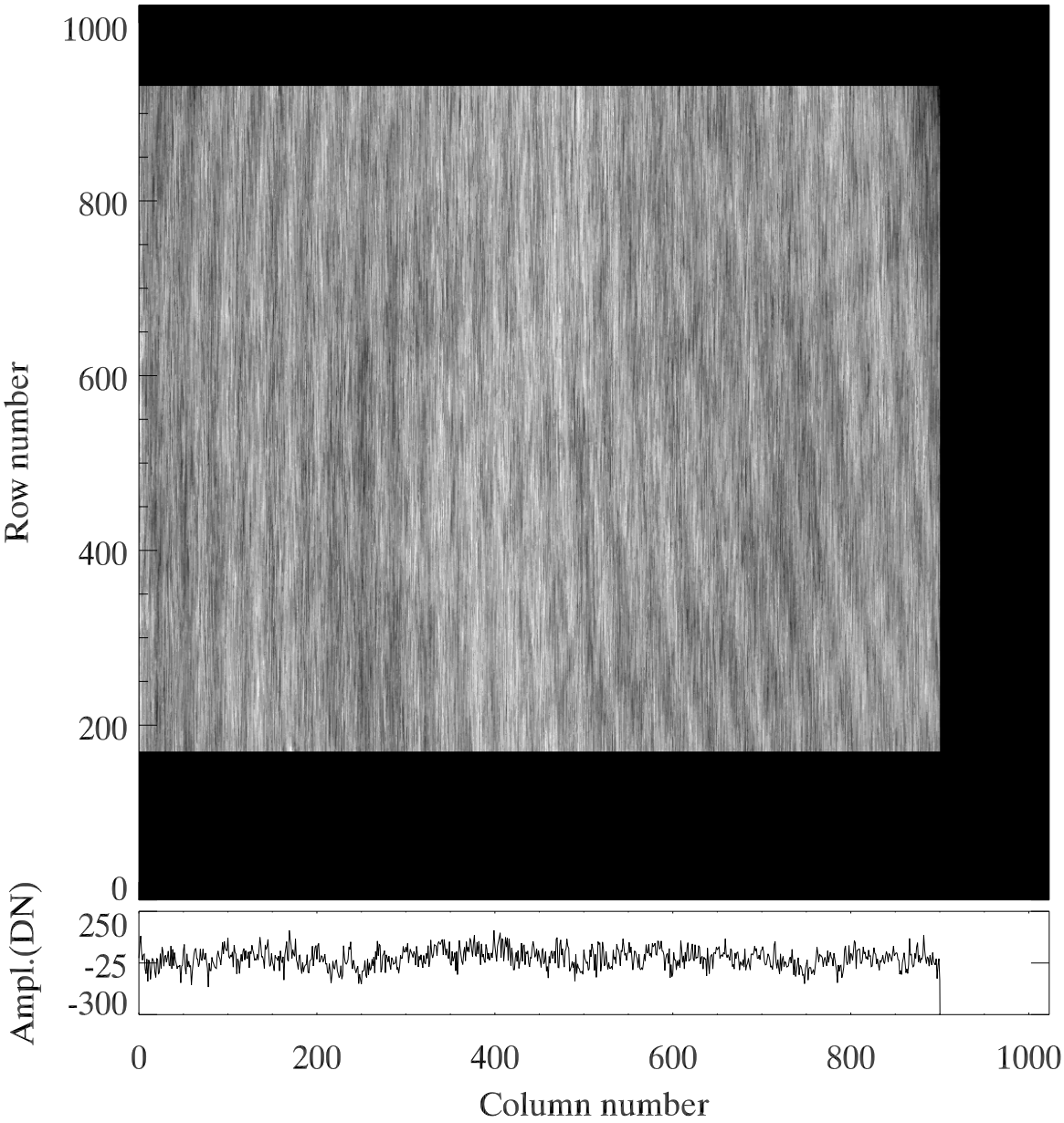

FIG. 2.— Top: Enhanced rows. Each pixel is first replaced by the median average of the 41 closest pixels in the vertical direction. A linear fit to each row is then subtracted. The fringe pattern is enhanced and some bad pixels have been removed. Note that the usable data area is reduced by 20 rows on the top and bottom because of the averaging. Bottom: Middle (512th) row.

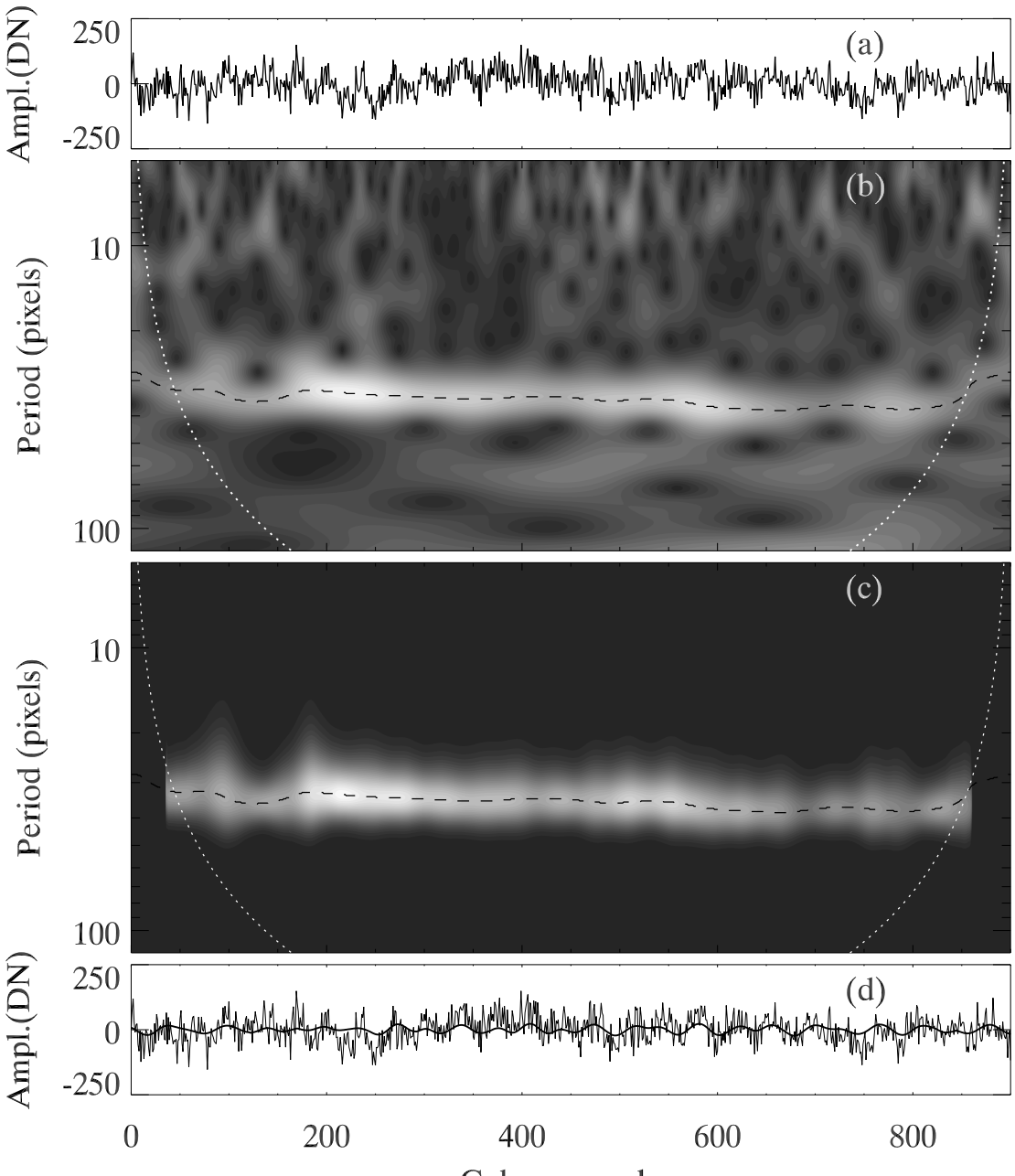

FIG. 3.— Wavelet transform of an enhanced row. In the center plots, the dotted line marks the cone of influence (COI); wavelet values below this boundary should not be trusted. The dashed line shows the fitted trace. Plot a: Middle enhanced row of Figure 2. Plot b: Amplitude of the Morlet wavelet transform of plot (a). The wavelet is interpolated in period by a spline from the period sampling of the transform, and the fringe transform, a coherent pattern corresponding to a fringe with a period of 35 pixels, is clearly visible. Plot c: Reconstructed fringe transform using a Gaussian fit (c.f. Figure 4). Plot d: Fringe pattern after applying an inverse wavelet transform to plot (c), plotted over the input data.

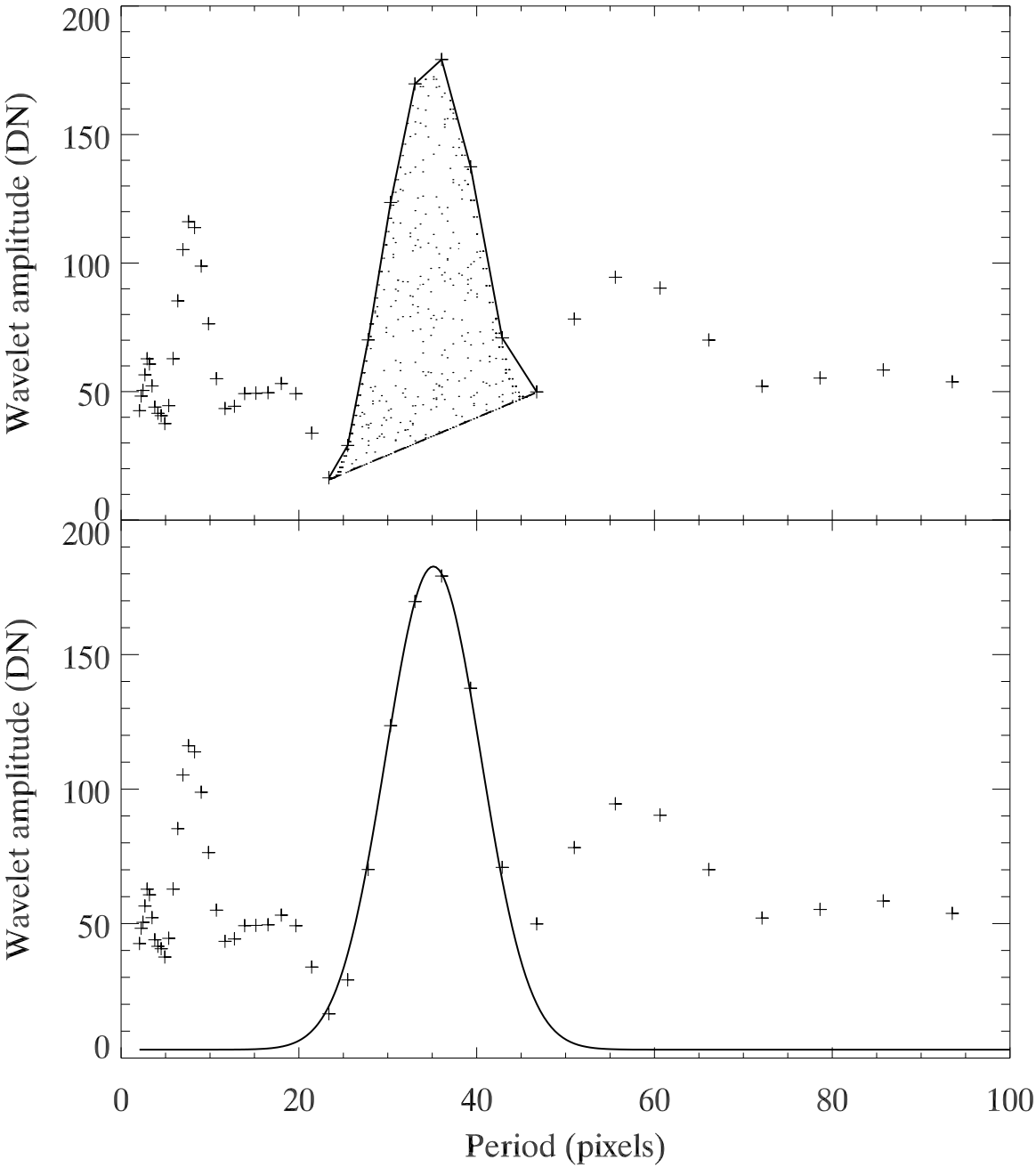

FIG. 4.— Profile fitting that is repeated for each column. Crosses are the amplitude of the wavelet points, solid lines are the fitted profiles, and dashed line is the background. Top: Trueshape profile; dotted area shows the profile that will be reconstructed after subtracting the indicated background. Bottom: Gaussian fit with background.

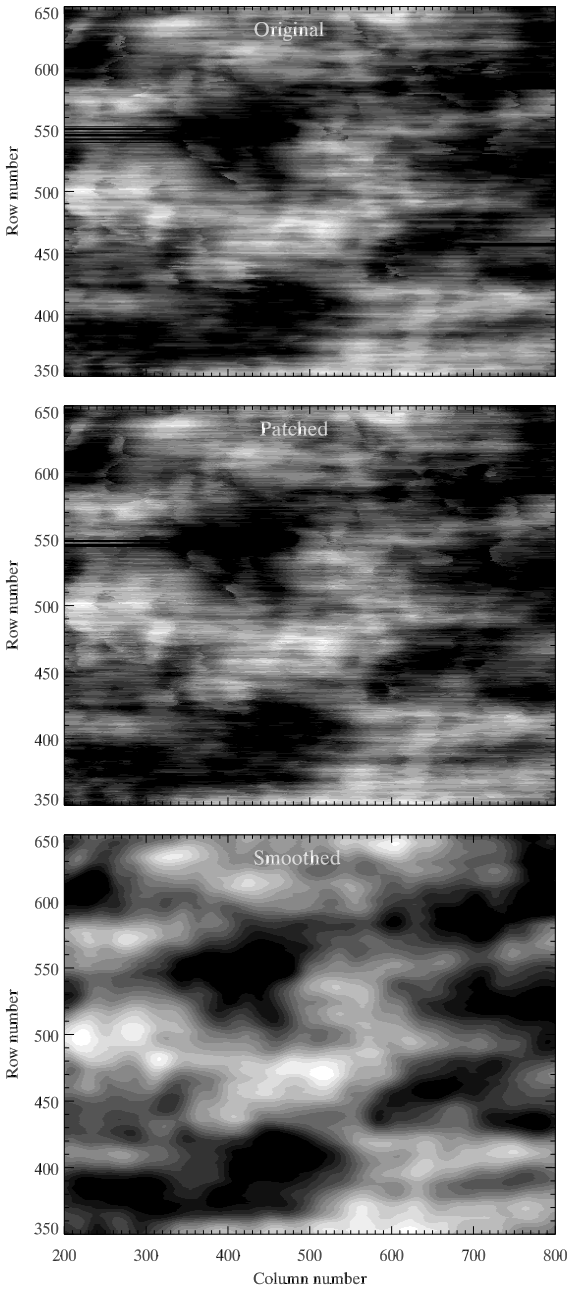

FIG. 5.— Gaussian height parameter smoothing. Top: Gaussian fit height parameter for central portion of example image. Middle: Parameter after replacing all values more than $\pm 1.5\sigma$ from the local median level (patched array). Bottom: Patched array after smoothing with a 19-pixel boxcar filter. This procedure is repeated for each of the other Gaussian fit parameters.

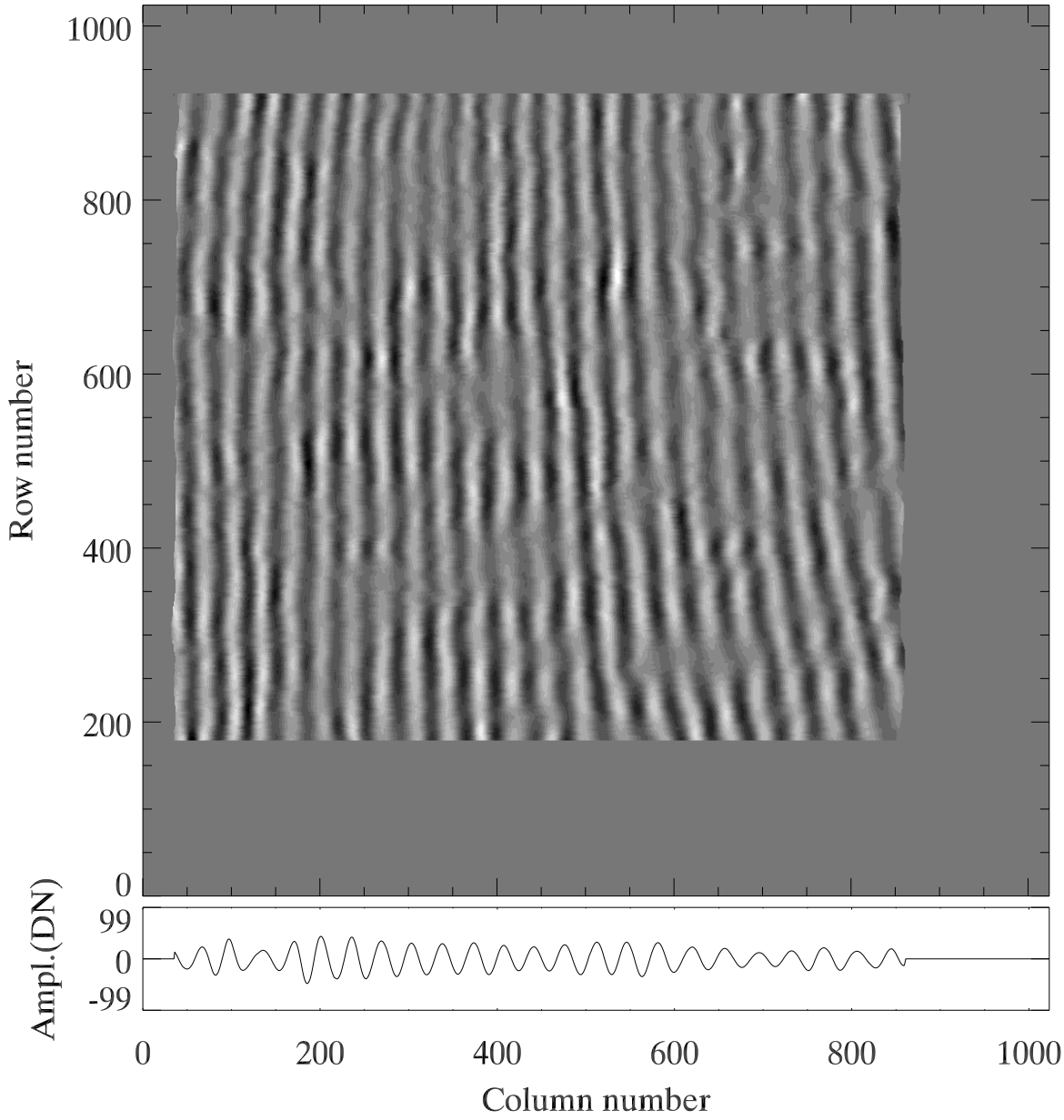

FIG. 6.— Top: Reconstructed fringe pattern (c.f. bottom panel of Figure 3). Bottom: Middle (512th) row.

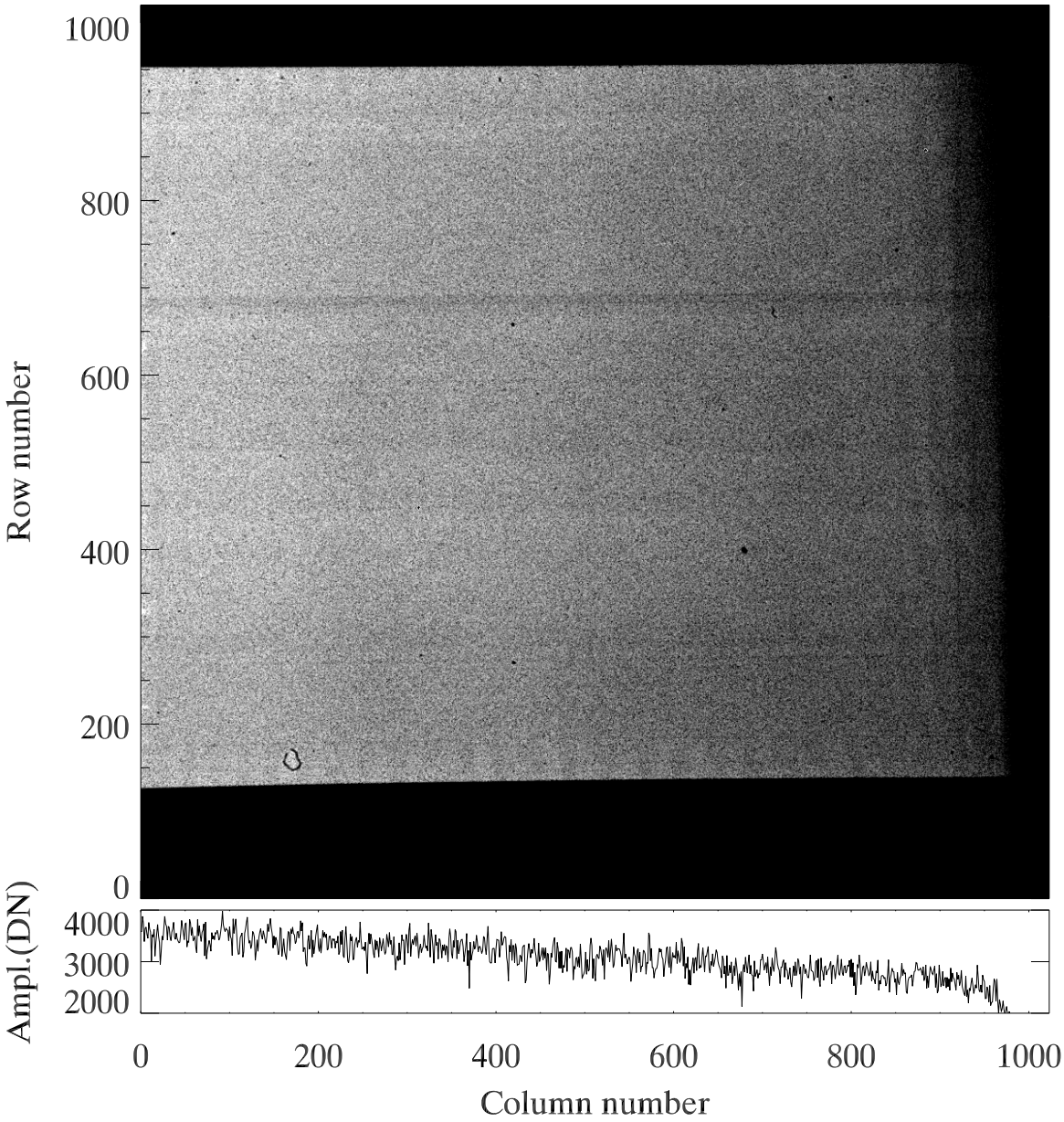

FIG. 7.— Top: Cleaned image. Flat field of Figure 1 minus the fringe pattern of Figure 6. Note that some of the edges remain uncorrected (see text). Bottom: Middle (512th) row.

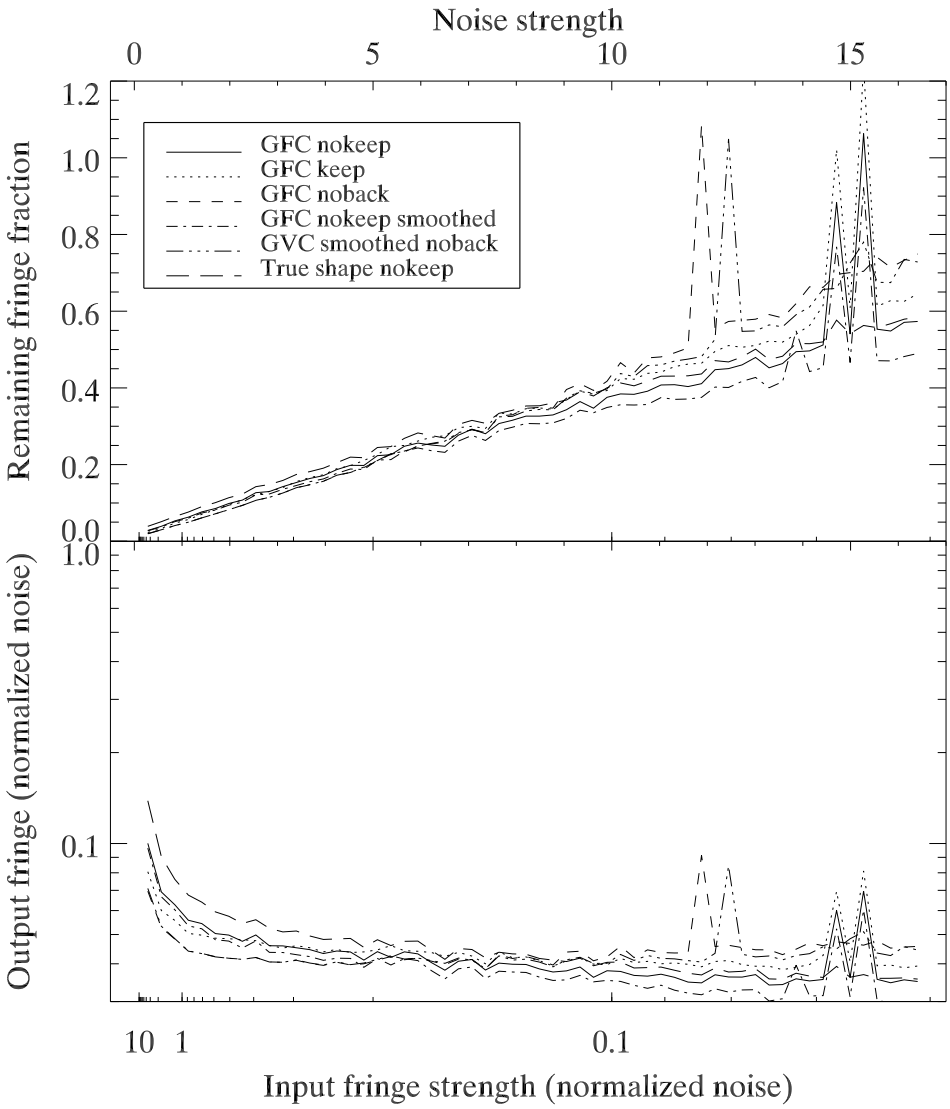

FIG. 8.— Remaining fringe for varying noise strengths. Only 6 of the 14 methods are shown above, for clarity. Omitted methods are similar to plotted methods and fall within the range of traces shown. The two nokeep methods give the best results, with GFC being more consistent at high noise. In general, methods without background give similar results for low noise strength, but are worse than their counterparts with background for high noise strength. Spikes at high noise levels are from random noise that tricks different methods, as shown in Figures 10 and 11. For a given method and noise level, spikes only appear in some of the test runs. Trueshape never shows spikes. Top: Fraction of fringe remaining. Bottom: Absolute fringe remaining when noise level is scaled to 1.

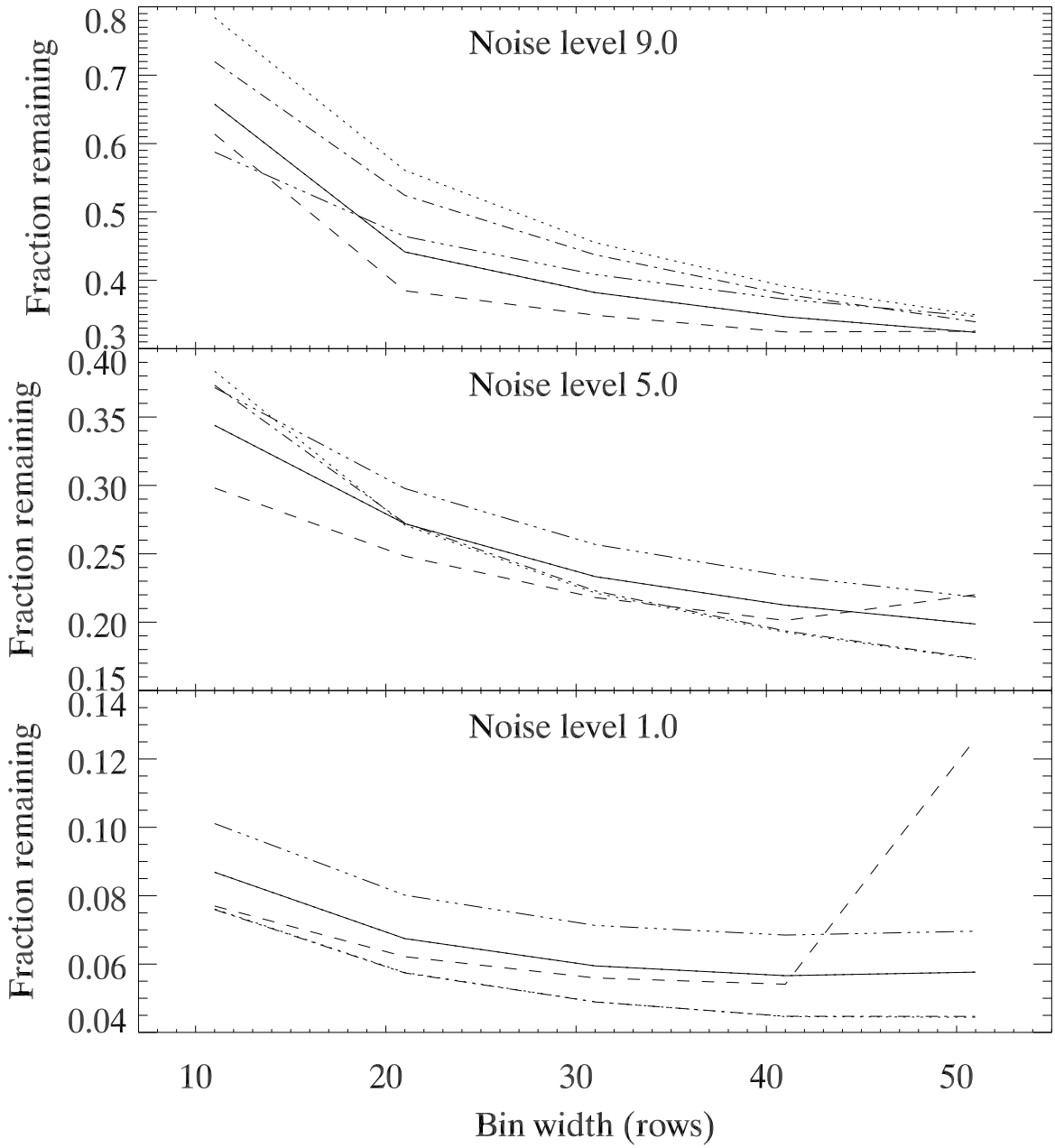

FIG. 9.— Remaining fringe for different enhanced-row bin widths. The panels show same synthetic fringe pattern as in Figure 8 at three selected noise levels. Same line styles as in Figure 8.

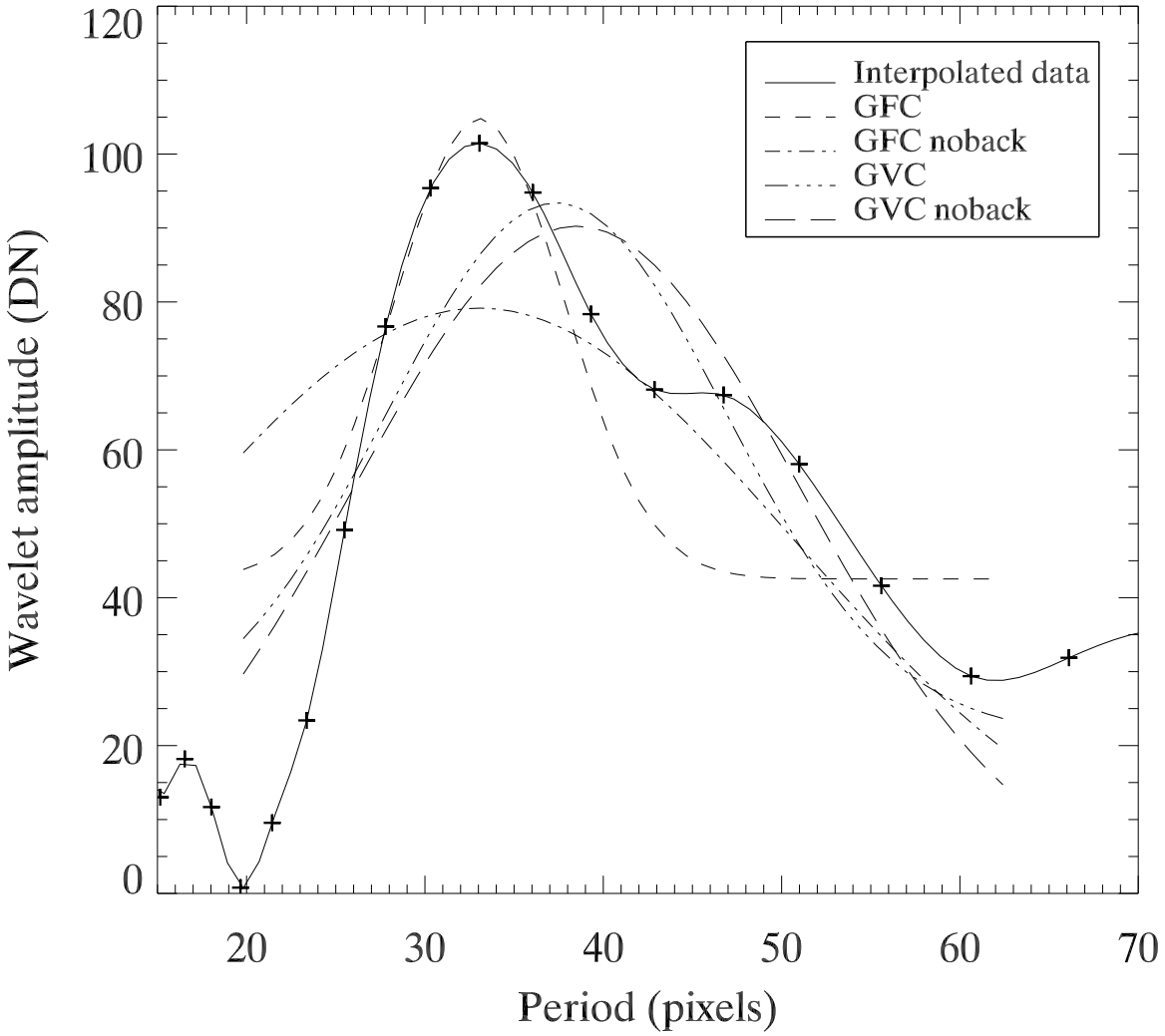

FIG. 10.— Different fitting methods applied to a profile with a complicated shape. Such shapes are due to noise and are the main limiting factor for this algorithm. This profile comes from the fringe transform for row 798, column 627 of our example array. Crosses show the data points, while the solid line is the interpolated profile. Other lines are explained in the key. Note that the profiles are only fit within the local minima at both sides of the reference period.

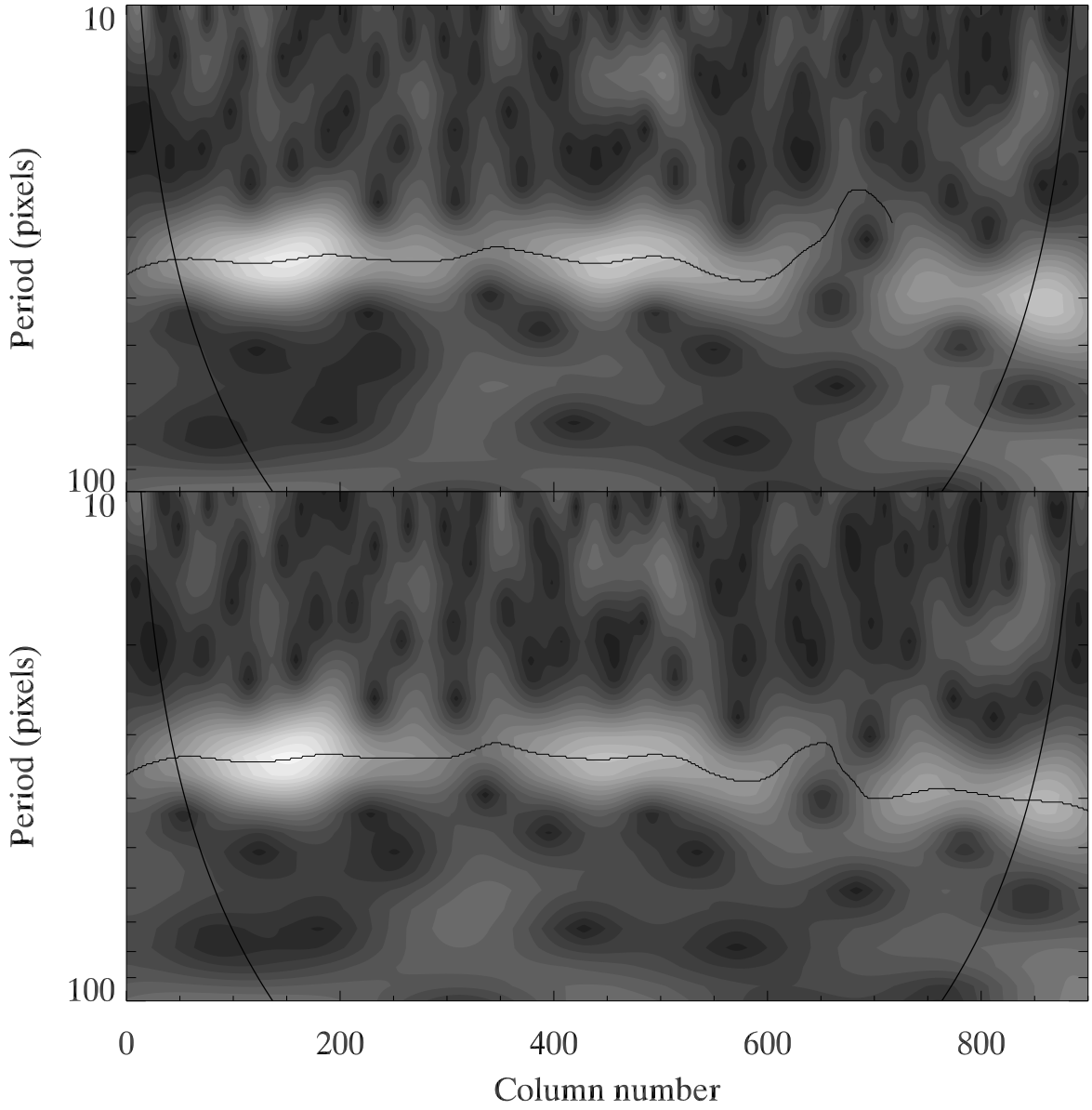

FIG. 11.— Example of missed trace. Top: Wavelet array from row 798 of Figure 2. Around column 650 the trace goes in the wrong direction, towards short period, and disappears around column 720. Bottom: Wavelet array from row 799 of Figure 2. Array is similar to top plot but now the trace is correct through the last column.