



Instrument Science Report WFC3 2010-13

WFC3/IR Bad Pixel Table: Update Using Cycle 17 Data

B. Hilbert and H. Bushouse
August 26, 2010

ABSTRACT

Using data collected during Servicing Mission Observatory Verification (SMOV) and Cycle 17, we have created an updated bad pixel mask for WFC3's IR channel. The bad pixel table contains flags that mark the position of pixels which are dead, unstable, have a bad zeroth read value, or are affected by "blobs". In all, 20,700 of the science pixels (2.0%) are flagged as bad. Observers are encouraged to dither their observations as a means of lessening the effects of these bad pixels.

Introduction

During the various ground testing campaigns for the WFC3 IR channel, efforts were made to identify and flag any pixels on the detector which were bad in ways that made them scientifically useless (Hilbert, 2003, 2007). With WFC3 successfully installed in HST, we wished to update this list of bad pixels to reflect any changes since ground testing. During SMOV and the early stages of Cycle 17 a large amount of calibration data were obtained, allowing for an on-orbit bad pixel search. This was accomplished using a combination of dark current and internal flat field observations.

Data

Observations used in the creation of this bad pixel table included dark current and tungsten lamp flat field ramps from both SMOV and Cycle 17 calibration programs. Details of the observations used are listed in Table 2 in Appendix 1. As noted in previous work (Hilbert and McCullough, 2009, Hilbert 2009, and Baggett 2009) a light leak in the

IR channel contaminated a fraction of both the dark current and flat field on-orbit observations. We ignored all ramps affected by light leak in this investigation.

Raw data files were processed through a modified version of *calwf3* v2.0, the standard data reduction pipeline. As with normal versions of *calwf3*, it performed reference pixel subtraction to remove bias offsets from the data, and zeroth-read subtraction to remove pixel-to-pixel variations in the data. This special version of *calwf3* however, contained a modification to the basic processing of WFC3/IR ramps which was very helpful in the search for bad pixels. Bad pixel searches performed with standard versions of *calwf3* netted a population of pixels affected by what appeared to be negative cosmic rays. In a given observation, these pixels would begin accumulating signal as expected. Then at some point in the ramp there would be a sharp decrease in signal of several tens to several hundreds of counts between two consecutive reads. Subsequent reads would return to the nominal rate of signal accumulation. Figure 1 shows an example of a negative cosmic ray. A more detailed examination and characterization of this effect will be performed at a later time. Just as with normal cosmic rays though, we did not want pixels affected by these negative signal jumps to be interpreted as bad. The version of *calwf3* that we used searches for and removes the effects of these negative signal jumps during processing, just as it does for the more normal positive CR jumps. With several thousand of these negative signal jumps removed from each input ramp, this addition to *calwf3* removed a large number of incorrectly identified bad pixels from the bad pixel search. This feature will be added to a future public release of *calwf3*. Release notes for new versions of *calwf3* can be found at: http://www.stsci.edu/hst/wfc3/pipeline/wfc3_pipeline.

We also made one other change to the nominal *calwf3* processing steps while preparing these data for the bad pixel search. During the final processing step, when *calwf3* fits a line to the signal up the ramp for each pixel in order to determine a signal rate, it uses the cosmic ray rejection parameters reference table (listed under the CRREJTAB keyword in each data file) to decide which data quality (DQ) values assigned to each pixel should be considered as bad and therefore rejected from the line fitting. The default behavior is that any pixels flagged as dead, with a bad zeroth read value, unstable, or saturated will not have line-fitting performed, and hence have a value of zero in the final *flt* image. As our goal was to identify bad pixels, we did not want to rely on previous identifications of bad pixels. We therefore created our own CRREJTAB file, where we changed the values in the BADINPDQ column to be zeros. This tells *calwf3* to ignore all bad pixel flags, and to perform line-fitting on all pixels except those that are saturated for the entire duration of the ramp. After running all observations listed in Table 2 through *calwf3*, we had a signal rate image (*flt* file) for each dark current and flat field observation.

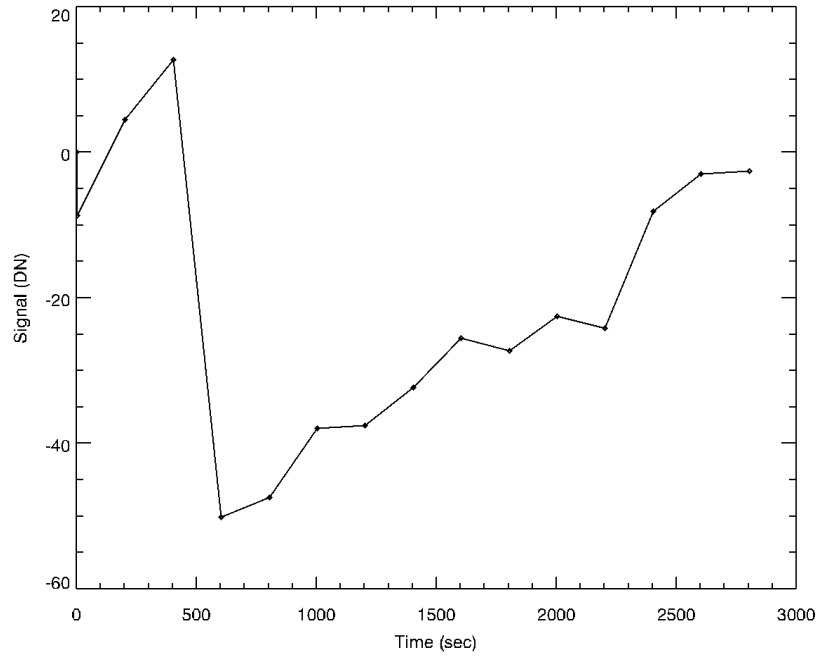


Figure 1: An example of a “negative cosmic ray” observed in a dark current ramp. Note the large drop in signal between the consecutive reads at roughly 500 seconds, followed by a return to a nominal dark current rate for the remainder of the ramp. In publicly available versions of *calwf3* (up through at least version 2.1), these negative signal jumps were not being identified, causing *calwf3*’s line-fitting routine to underestimate the true dark current rate.

Analysis

Using the flat field and dark current observations, we searched for four types of bad pixels. These included: “dead”, “unstable”, “bad zeroth read”, and those affected by “blobs”. Our goal was to produce a table that lists pixels which are permanently bad. For example, a “dead” pixel, which exhibits little or no sensitivity to illumination, is unlikely to someday gain sensitivity and wake from the dead. By focusing on permanently bad pixels, our aim is to have a table that changes little over time and can be applied to any WFC3/IR observation. Pixels which are bad in potentially more transitory ways, such as those with higher than average dark current, are flagged in other types of reference files, such as dark current files.

Bad Zeroth Read Pixels

We performed a separate search for each of the four types of bad pixels to be listed in the bad pixel table. First, we identified “bad zeroth read” pixels. These are pixels which

exhibit anomalous signals in the zeroth read of a data ramp (the read that immediately follows the detector reset at the beginning of an IR exposure sequence), usually due to being shorted or unbonded (Hilbert, 2003). We used the zeroth reads of the raw dark current ramps listed in Table 2 for this search. Using the zeroth reads from all of the 13 input ramps, we created a sigma-clipped mean zeroth read image. We then created a histogram of the values in this mean zeroth read for each quadrant of the detector. Since each quadrant of the detector is read out using a different amplifier and we are working with raw data that are not corrected for differing bias levels, it was important to treat each quadrant separately. A Gaussian was fit to the histogram from each quadrant, and pixels falling more than 3σ from the mean value were flagged as bad. In order to catch any pixels which may have been inconsistent from ramp-to-ramp, we also repeated the above process on each input ramp individually. Any pixels which were more than 3σ from the mean in some ramps but not others were caught with this additional step. With this two-step method, we identified 4,990 pixels (0.5% of the detector's light-sensitive pixels) which are bad in the zeroth read. Figure 3 shows a map of these bad pixels, which are flagged with a value of 8 in the bad pixel table. Figure 2 shows an example of a histogram and gaussian fit.

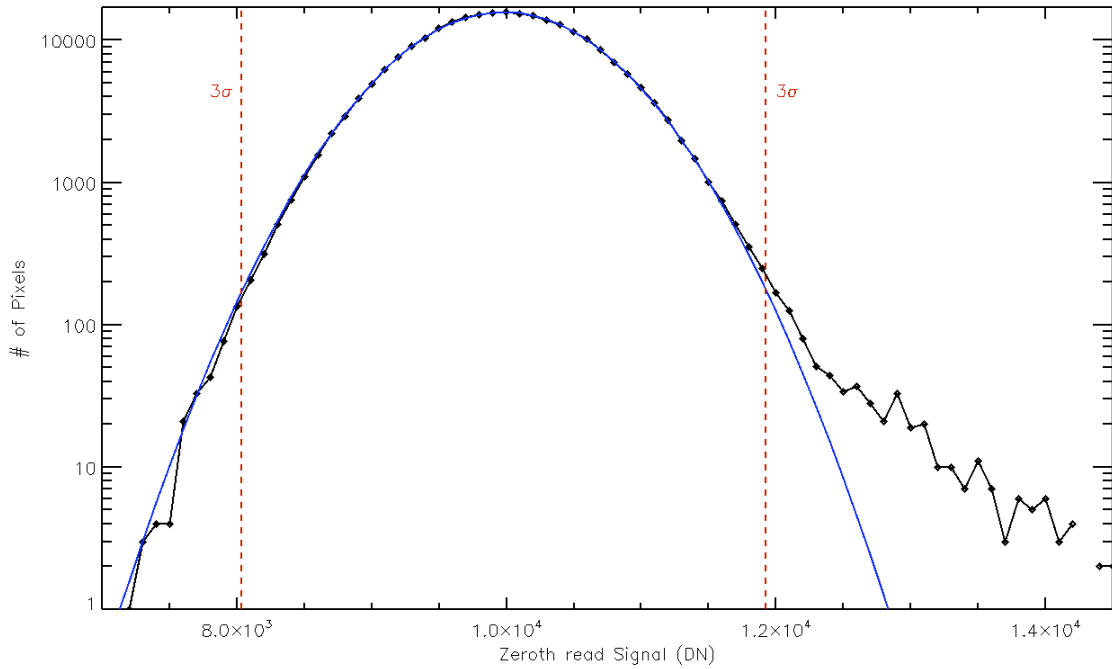


Figure 2: Histogram of zeroth read signal values in quadrant 1 (upper left) of the detector. The black line shows the histogram, on a log scale. The blue line is a Gaussian fit to the histogram. The dashed red lines show the 3σ limits of the distribution. Pixels outside of these limits were flagged as bad in the zeroth read.

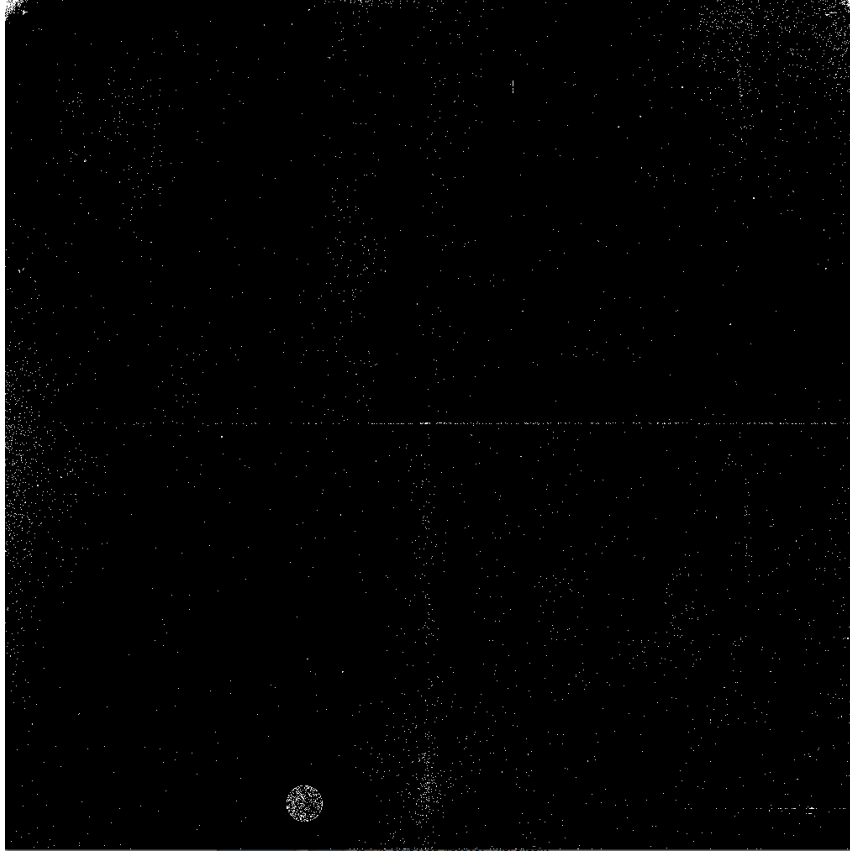


Figure 3: Bad zeroth read pixel mask. These pixels are flagged with an 8 in the final bad pixel table.

Dead Pixels

We next searched for dead pixels. These are pixels with a very low quantum efficiency and measure little or no signal when illuminated. As with the bad zeroth read pixels, we used a two step process to find dead pixels. First, we created a mean flat field image, using the final read from each of the flat field files listed in Table 2. In order to combine data from different filters, we created a mean flat field image for each filter. We then normalized each input file by the appropriate mean image for that filter. Using all of the normalized flat field images, we calculated a sigma-clipped mean value for each pixel, resulting in a single mean flat field image. First, any pixels exhibiting zero or negative signal in the mean flat field were flagged as dead. Next, we looked for pixels with very low, but not necessarily zero, quantum efficiency. For this, we moved pixel-by-pixel across the detector. For each pixel, we calculated the sigma-clipped mean value

in the surrounding 50 by 50 pixel box. If the pixel had a signal less than 30% of the local mean signal, we flagged it as dead. By using this strategy, we hoped to compensate for the non-uniformity of the flat field illumination level. By comparing each pixel's signal to that of its neighbors, we avoided marking a pixel as bad simply because it was located on a portion of the detector with a lower illumination level compared to other areas. Finally, we manually marked the pixels comprising the “death star” as dead. Using these three steps, we found 3,910 dead pixels (0.4% of the detector's light-sensitive pixels), which were marked with a 4 in the bad pixel table.

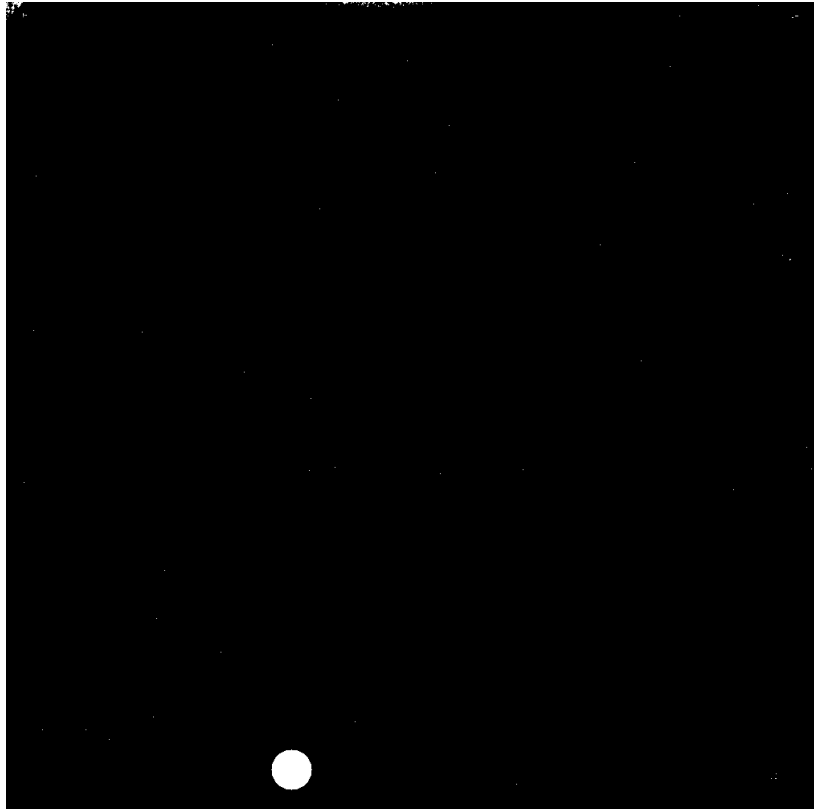


Figure 4: Dead pixel mask. These pixels are flagged with a 4 in the final bad pixel table. Dead pixels are currently largely confined to the “death star” and small areas in the upper corner and along the edge of the detector.

“Blobs”

The third type of bad pixel added to the bad pixel mask was pixels which were affected by “blobs”. WFC3 ISR 2010-06 (Pirzkal, Viana & Rajan 2010) provides details on these blobs. Essentially, these blobs are due to particulates resting on the Channel Select Mechanism (CSM). Areas covered by blobs predictably show a decrease in throughput and increased photometric uncertainties. As a result we take a list of blob

positions and diameters, provided by Pirzkal, and mark the blob-affected pixels with the value of 512 in the bad pixel table. In all, 4,534 pixels (0.44% of all science pixels) are distributed among 19 blobs.

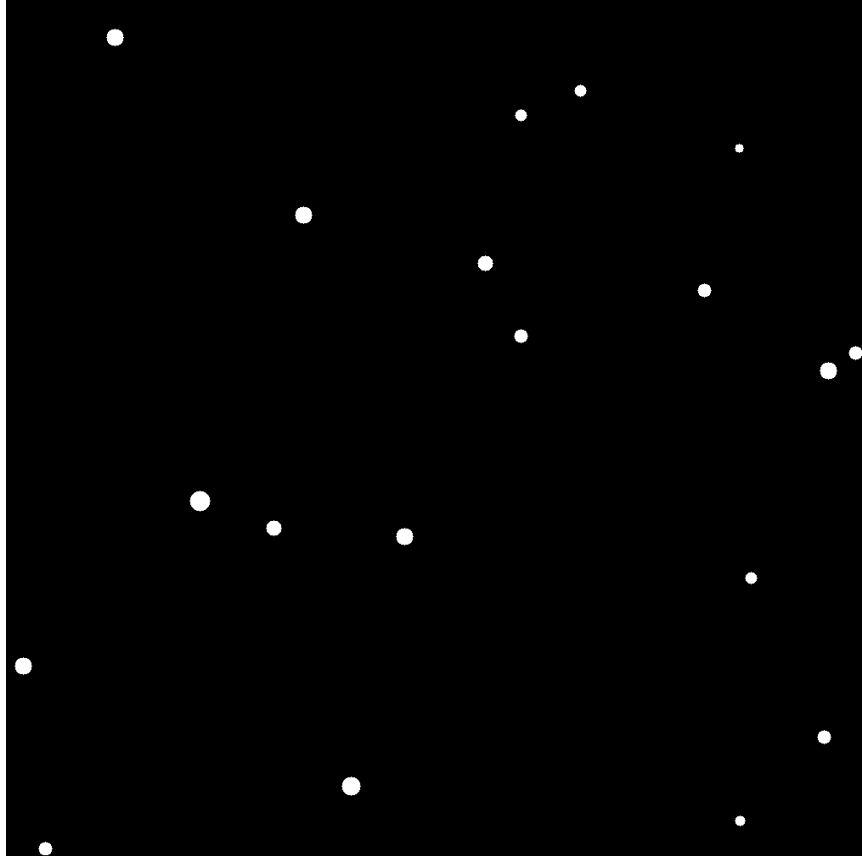


Figure 5: The IR channel blob mask. Pixels impacted by blobs are marked with a value of 512 in the final bad pixel table.

Unstable Pixels

The final type of bad pixel cataloged in the bad pixel table is the “unstable pixel”. These pixels display an inconsistent measurement of signal in a set of nominally identical ramps. Unstable pixels are characterized more thoroughly in Hilbert (2007). The search methods with which unstable pixels were identified here changed substantially from those used in the past analysis in order to work around the higher rate of cosmic rays present in on-orbit data.

Unstable pixels observed on the WFC3/IR detector display a wide range of behaviors. Some unstable pixels appear stable and repeatable in almost all of a data set, but will then measure appreciably different signal values in one or two ramps. Other unstable pixels display signal values that vary wildly from ramp to ramp in all observations of a data set. Investigations across a wide range of IR data as part of this study also revealed some

pixels which were unstable in a set of dark current ramps but stable in a set of flat field ramps, and others with the opposite behavior. See Appendix 2 for plots of various types of unstable pixels found on the IR detector.

From these observations, we developed a dual-search method for finding unstable pixels, and applied these searches separately to a set of flat field data and a set of dark current data. Pixels flagged by these searches were all flagged with a value of 32 in the final bad pixel mask.

Search Method 1

As in the dead pixel search, the first step in the unstable pixel search was to create a mean flat field or dark current image, depending on the data set being investigated. For this test, we wished to create a mean flat field from a set of *flt* files returned by *calwf3*. For the flat field investigation, we also wished to combine data from several filters when creating the mean flat field in order to make a more exhaustive search for unstable pixels.

In order to create a complete mean flat field, the first task was to ensure that each pixel in each *flt* image had a non-zero value. By default, *calwf3* will not perform line-fitting on certain types of bad pixels, leaving those pixels with a value of zero in the final output image. As mentioned previously, we wished to avoid depending on flags in the previous bad pixel mask due to the differences in the ways we identified bad pixels during this search. As a result, we manually performed line-fitting on any pixels which *calwf3* ignored. Despite using a CRREJTAB parameter file specifying that *calwf3* should reject no pixels flagged as bad and perform line-fitting on all pixels, there was a population of 218 pixels across the detector where one or more input *flt* files had a value of zero. These zeros imply that *calwf3* did in fact skip the line-fitting step for these pixels. Examination of this small population of pixels reveals a wide range of data quality flags. Most, but not all, of these pixels contained the 256 flag in the data quality array of their respective *flt* file, meaning that they were saturated in all reads of the *ima* file. For each of these 218 pixels, we manually fitted the signals in each of the 54 input flat field *ima* ramps, even if only one input *flt* file had a value of zero in that pixel, in order to arrive at a consistent set of *flt* values for that pixel.

Next we created a mean flat field for each filter. This was done on a pixel-by-pixel basis, using IDL's *resistant_mean.pro* to calculate a sigma-clipped mean value in each pixel. Each individual *flt* file was then normalized by the appropriate filter's mean flat field image. All normalized *flt* files were then combined into an overall mean flat field image. Again, sigma-clipped mean values were calculated on a pixel-by-pixel basis with *resistant_mean.pro*. When the searches were run on dark current data, we obviously did not need to normalize data ramps in order to combine them. In that case, we simply created a mean dark current image on a pixel-by-pixel basis from all the input dark current ramps listed in Table 2.

To make the following discussion more clear, we introduce the following notation. In the overall mean flat field (or mean dark current) image, the signal in pixel i,j is $\mu_{i,j}$, with an uncertainty of $\sigma_{i,j}$. For the same pixel, the measured signals in the individual flat field or dark current images are given as $f^1_{i,j}, f^2_{i,j}, f^3_{i,j}$, etc, with corresponding uncertainties of $s^1_{i,j}, s^2_{i,j}, s^3_{i,j}$, etc.

With the mean flat field image in hand, we performed the first of the two searches for unstable pixels. In the first search, we looked for pixels that are subtle in their instability. For example, for a given set of identical flat field images, pixels caught by this search method may exhibit consistent signal measurements in all but one or two images. An example of a pixel found to be unstable using this search method is shown in Figure 6. This behavior results in a small value of $\sigma_{i,j}$ in the mean image, which we can therefore use as a filter.

We compared a pixel's value in the mean flat field image, $\mu_{i,j}$, versus its values in all of the individual flat field images, after each of which was normalized by the mean flat field image for its filter. We took into account the uncertainty in both the mean flat field, $\sigma_{i,j}$, as well as those in the individual flat fields. If a pixel's value in any of the individual flat field images was more than N times the measured uncertainty away from that in the mean flat field, the pixel was flagged as unstable. That is, pixel i,j was flagged as unstable if Equation 1 was found to be true for any of the input flat field images (f^1 through f^{54}).

$$\left(f_{i,j} + N \times s_{i,j}\right) \leq \left(\mu_{i,j} - N \times \sigma_{i,j}\right) \text{ or } \left(f_{i,j} - N \times s_{i,j}\right) \geq \left(\mu_{i,j} + N \times \sigma_{i,j}\right) \quad 1)$$

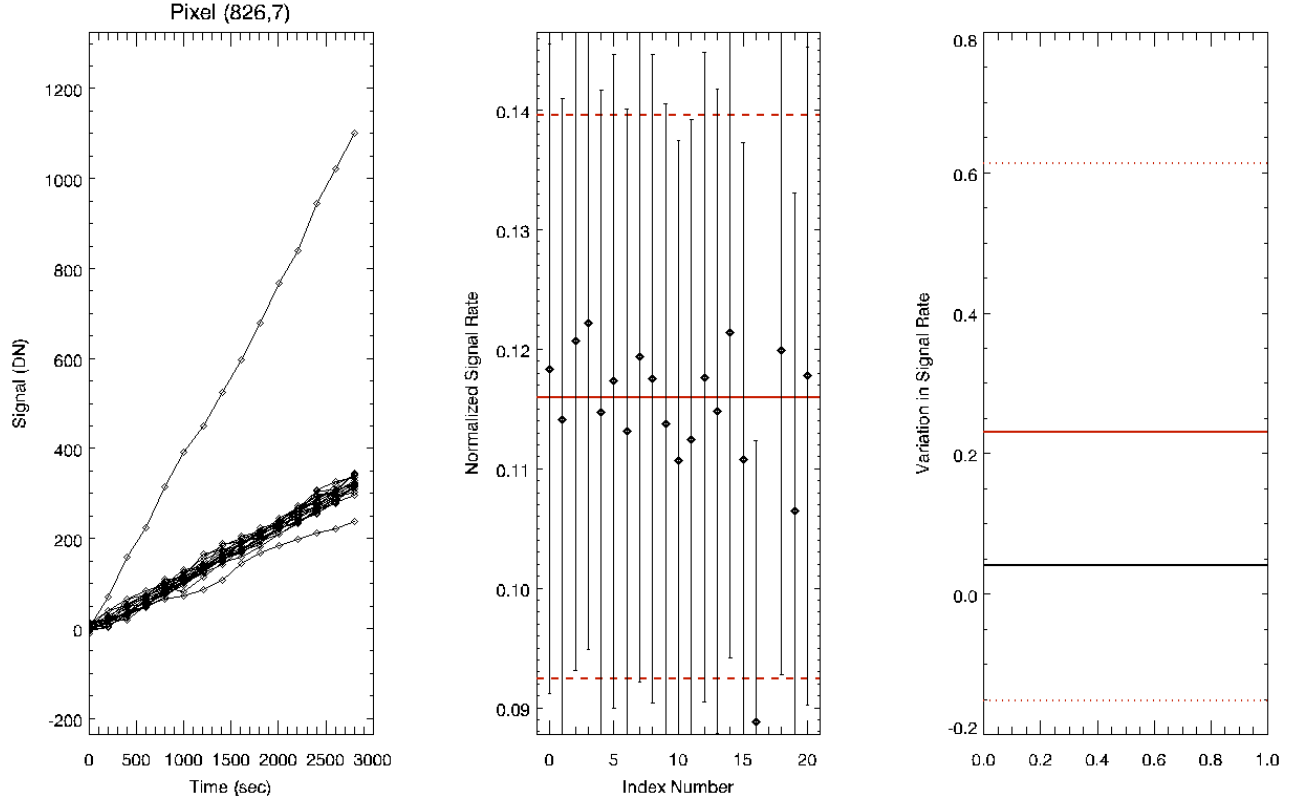


Figure 6: An example of an unstable pixel caught using Equation 1. The panel on the left shows the measured signal up the ramp for each of 21 ramps. Most ramps exhibit similar signals, except for one where the signal rate is much higher than normal, and one where the signal rate is slightly lower than normal. The center panel shows the graphical equivalent of Equation 1. The measured signal rate for each ramp is displayed as a black diamond, with 5σ error bars. The solid red line indicates the sigma-clipped mean signal rate calculated from all ramps, and the dashed red lines indicate the 5σ uncertainty around the mean value. In this case, we see that ramp #16, while it measured a low signal rate, still has error bars which fall within 5σ of the mean. However, the measured signal rate for ramp #17 is high enough that its diamond and error bars are outside of the 5σ area, and are in fact above the range of the plot. This results in this pixel being flagged as unstable. The right panel gives a graphical representation of search method #2, described below.

Search Method 2

Unlike the mostly-stable pixels netted in the first search above, the second search for instability looked for pixels which were more extreme in their instability, and had wildly varying signals across most or all ramps of a nominally identical set. This behavior results in a very large value of σ_{ij} , meaning that these pixels can slip through the simple filter described by Equation 1. In this search, we looked at the distribution of values in the input flat fields ($f^1_{ij}, f^2_{ij}, f^3_{ij}$, etc) versus the distribution of values in the “average” pixel.

First, we calculated the sigma-clipped standard deviation of the input values for each pixel using all of the normalized flat field images, as shown in Equation 2.

$$dev_{ij} = stdev(f^1_{ij}, f^2_{ij}, f^3_{ij} \dots) \quad 2)$$

We then calculated the sigma-clipped mean of these standard deviation values across all pixels on the detector, using Equation 3.

$$mean_dev = \langle dev \rangle \quad 3)$$

This mean gave the variation in measured signal for the “average” pixel. For the input set of flat field images, we found $mean_dev$ to be 0.51. This shows that the mean variation in signal level, normalized to the mean signal level, is 0.51%. We also measured the standard deviation of the collection of standard deviations, in order to determine to what extent the “average” pixel was representative of all pixels across the detector.

$$dev_dev = stdev(dev) \quad 4)$$

This resulted in an uncertainty on the variation in the “average” pixel of 0.06%. For each pixel, we then compare the standard deviation of its input flat field values (dev_{ij}) to that of the “average” pixel (dev_dev). Any pixel satisfying Equation 5 was flagged as unstable.

$$dev_{ij} > (M \times dev_dev + mean_dev) \quad 5)$$

An example of a pixel which was identified as unstable using this search method is shown in Figure 7.

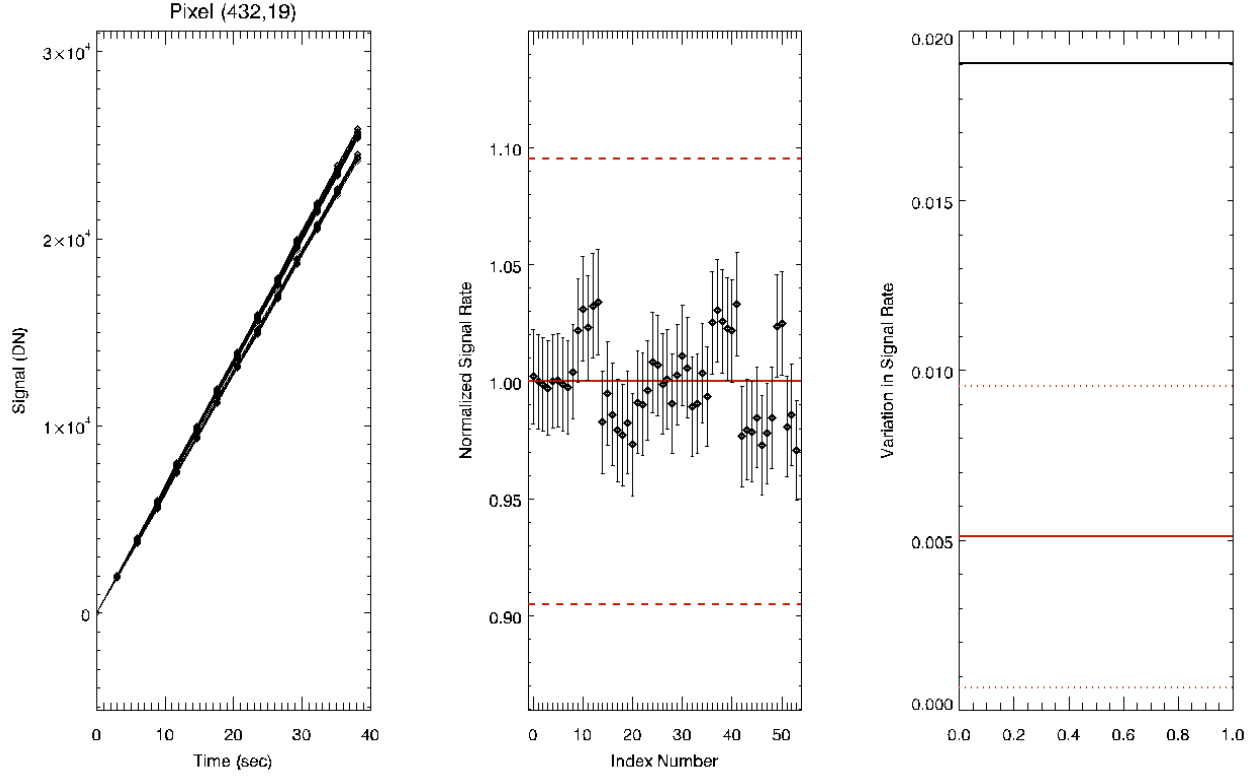


Figure 7: An example of a pixel found to be unstable using Equation 5. As with Figure 6, the left panel shows the measured signal up the ramp for 54 input ramps. Note the bimodal distribution of measured rates. The center panel shows the calwf3-derived normalized signal rate value for each ramp, along with 5σ error bars (dashed red lines) where σ is the standard deviation of the measured rates. Note that for this pixel, all ramps have measured signal rates within 5σ of the mean (solid red line at 1.0). This pixel was not flagged as unstable in the first search method because the large number of measured signal rates at each value in the bimodal distribution artificially increases the value of $\sigma_{432,19}$ used in Equation 1. The right panel gives a graphical representation of the second search method. Here, the red solid line shows the variation observed in the “average” pixel (i.e 0.51%). The dotted red lines show the upper and lower limits associated with a 7σ uncertainty around this mean. The solid black line shows the variation measured in this particular pixel. Here we see that the measured variation in signal rate for this pixel is well above of the 7σ threshold around the “average” pixel, causing this pixel to be flagged as unstable.

Threshold Values

After repeating these searches many times and studying the populations of unstable pixels resulting from various threshold values of N and M in the two search methods described above, we settled upon values in the final analysis of $N = 5$ and $M = 7$.

Flat Field Data

Using a value of 5 for N in Equation 1 resulted in 2,223 pixels found to be unstable in the flat field data, most of which are located in the two upper corners as well as the lower right corner of the detector, as shown in Figure 8.

A value of 7 for M translates into a 0.93% signal variation when normalized to the mean signal (i.e. from Equation 5: $0.51\% + 7 \cdot 0.06\% = 0.93\%$). In other words, given a set of nominally identical flat field ramps, if the signal rate measured in a given pixel varies by more than 0.93% of the mean rate, that pixel was considered unstable. We chose the 0.93% threshold level to keep in line with the ultimate goal of 1% photometry for the IR channel. Using this threshold, this method identified 3,795 unstable pixels, which were mainly clustered in three corners of the detector, along with an area halfway across the top edge (see Figure 9). Approximately 1,450 of the pixels flagged as unstable in the flat field data were identified using both the first and second search methods. Roughly 775 unstable pixels were found only through the first search method, with Equation 1, and 2,350 pixels were only found through the second search method, with Equation 5.

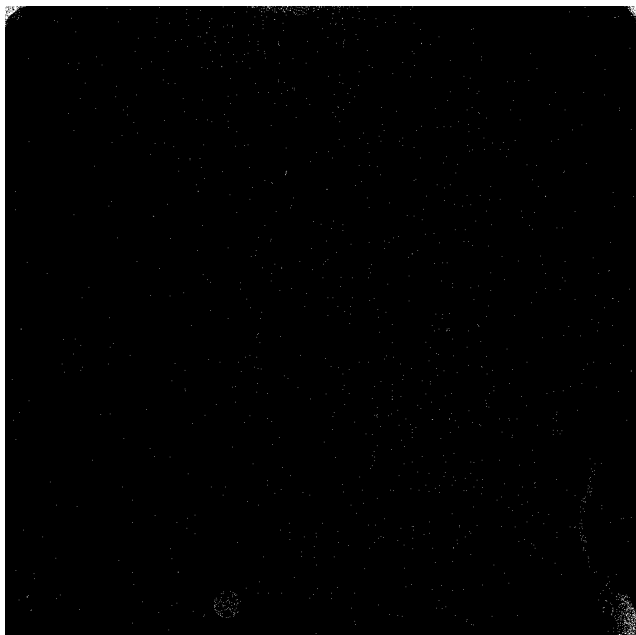


Figure 8: Unstable pixels in the flat field data from Equation 1.

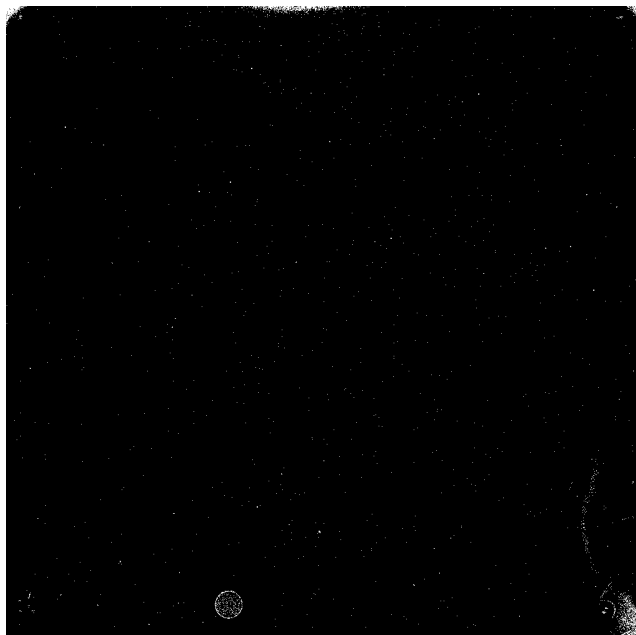


Figure 9: Unstable pixels in the flat field data from Equation 5.

Dark Current Data

A value of 5 for N in the dark current data resulted in 6,001 pixels being flagged as unstable. These pixels are also concentrated in the upper corners, as well as in the center of the top edge of the detector as shown in Figure 10. Of these, only 720 pixels were also found using Equation 1 on the flat field data set (i.e., pixels flagged as unstable in the dark are not necessarily unstable in the flat field).

Using the threshold value of $M = 7$ for the much lower signal-to-noise dark current input files, we arrived at a cutoff a 61.5% signal variation. Again, using Equations 3 and 4, we found a value of 23.0% for *mean_dev*, the standard deviation of signal in the “average” pixel, and 5.5% for *dev_dev*. This gives a final cutoff value of $23.0\% + 7. * 5.5\% = 61.5\%$ for the maximum level of normalized signal variation in a pixel before flagging the pixel as unstable. This level of signal variation is necessarily much larger than that of the flat field analysis due to the much larger effect of readnoise on the low-signal dark current data. Using this threshold value, we found 3,318 unstable pixels in the dark current data, spread evenly across the detector, as shown in Figure 11. Only 880 of these pixels were also contained in the 6,001 pixels flagged using the first search technique. In addition, only 480 of these flagged pixels were common to the 3,795 unstable pixels found using this technique on the flat field data.

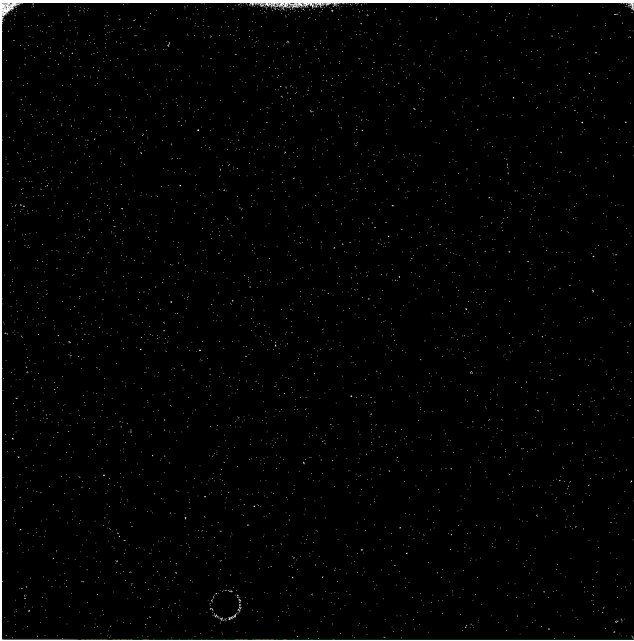


Figure 10: Unstable pixels in the dark current data from Equation 1.



Figure 11: Unstable pixels in the dark current data from Equation 5.

Combined

In all, the search methods described above found 4,571 unstable pixels in the flat field dataset, and 8,432 unstable pixels in the dark current dataset. Combining these two masks, we find a total of 10,885 unstable pixels (1.06% of all science pixels) on the IR detector. Figure 12 shows a map of all unstable pixels identified in this analysis.

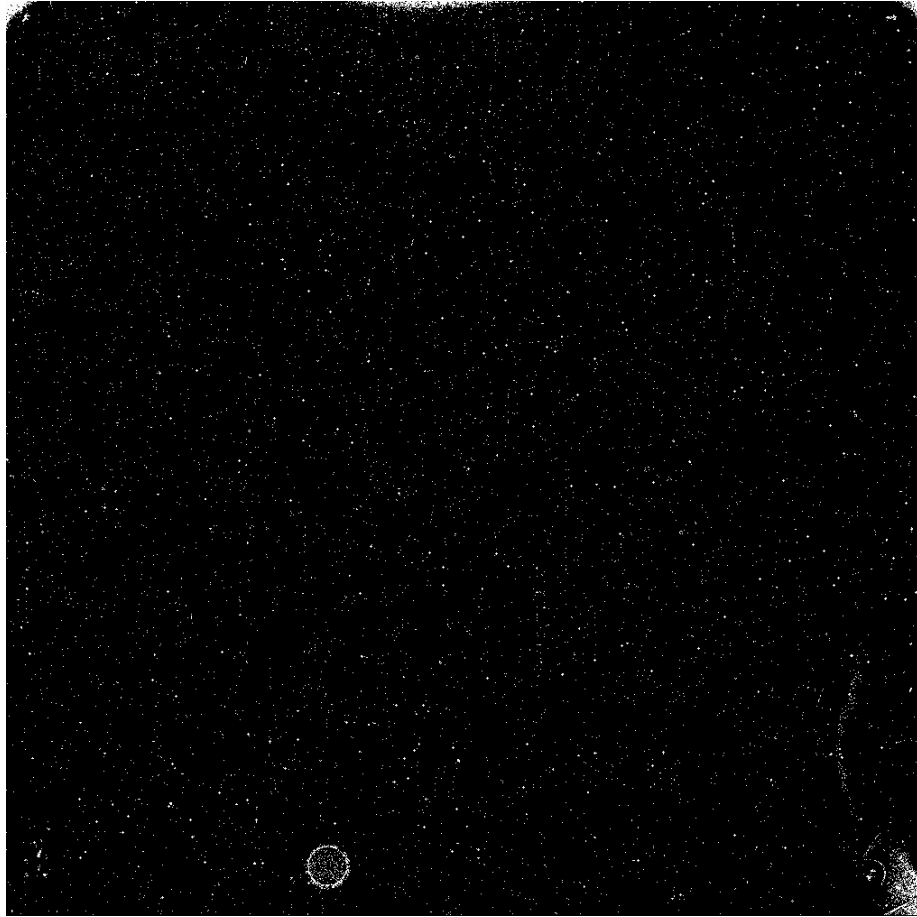


Figure 12: All unstable pix found using both searches on all dark and flat field data.

Conclusions

Once we had in hand maps containing all four types of bad pixels on the IR detector, we combined these maps into a final bad pixel map. As with all bad pixel maps and data quality arrays associated with Hubble data, bit-wise addition was used to combine the individual bad pixel masks. Table 1 shows the flag values associated with the bad pixels contained in the bad pixel table. Any pixel which was found to be bad in more than one way has a value in the final bad pixel table equal to the sum of the flags corresponding to the ways in which it is bad. Figure 13 shows the locations of all pixels flagged as bad in any way in this work. On the light-sensitive portion of the detector, 20,700 pixels (2.0% of all pixels) were found to be bad. The default behavior of *calwf3* at the time this analysis was performed was for pixels flagged as dead, unstable, or with a bad zeroth read to be ignored. This was specified using the cosmic ray rejection parameter table, *tag1826ki_crr.fits*. This means that for a given observation, the processed *flt* file contained a value of zero in all pixels flagged as dead, unstable, or with a bad zeroth read. (Since this analysis was completed, a new cosmic ray rejection parameter table, *u6a1748ri_crr.fits*, has been created and delivered to the calibration database. Using this file, which is now the default, causes *calwf3* to attempt line fitting on all pixels regardless of whether they are flagged as bad or not. This forces the observer to manually decide which flavors of bad pixels to ignore in their analysis.)

Currently, *calwf3* does not set the blob affected pixels to zero, but instead just sets a data quality flag of 512 in the DQ array of the *flt* file. This reduces the number of pixels which will be zeroed-out due to the bad pixel table to 16,210. In either case, the distribution of bad pixels across the entire detector, as seen in Figure 13, highlights the importance of dithering when designing a WFC3/IR observing proposal. By shifting the position of the detector on the sky between individual observations and then combining the observations into a final image, it is possible to fill in holes caused by the bad pixels in the IR detector.

In addition to the four types of bad science pixels, the bad pixel table also contains a population of pixels flagged with a value of 128. This flag designates bad reference pixels on the detector. Currently, the only bad reference pixels on the detector are the outermost column and row of reference pixels, described as “outboard” reference pixels by Robberto et al. (2002). Adding these 4092 bad pixels to the 20,700 bad science pixels (2% of the science pixels), we arrive at a total of 24,792 bad pixels, representing 2.4% of the total detector.

Upon delivery to the calibration database system on 12 April 2010, this new bad pixel table was named *u4c1709ri_bpx.fits*.

Type of Bad Pixel	Bad Pixel Table Flag Value	Number of Pixels Affected
Dead Pixel	4	3,910
Bad Zeroth Read Pixel	8	4,990
Unstable Pixel	32	10,885
Bad Reference Pixel	128	4,092
Blob Affected Pixel	512	4,534

Table 1: Flag values used to designate the various types of bad pixels.

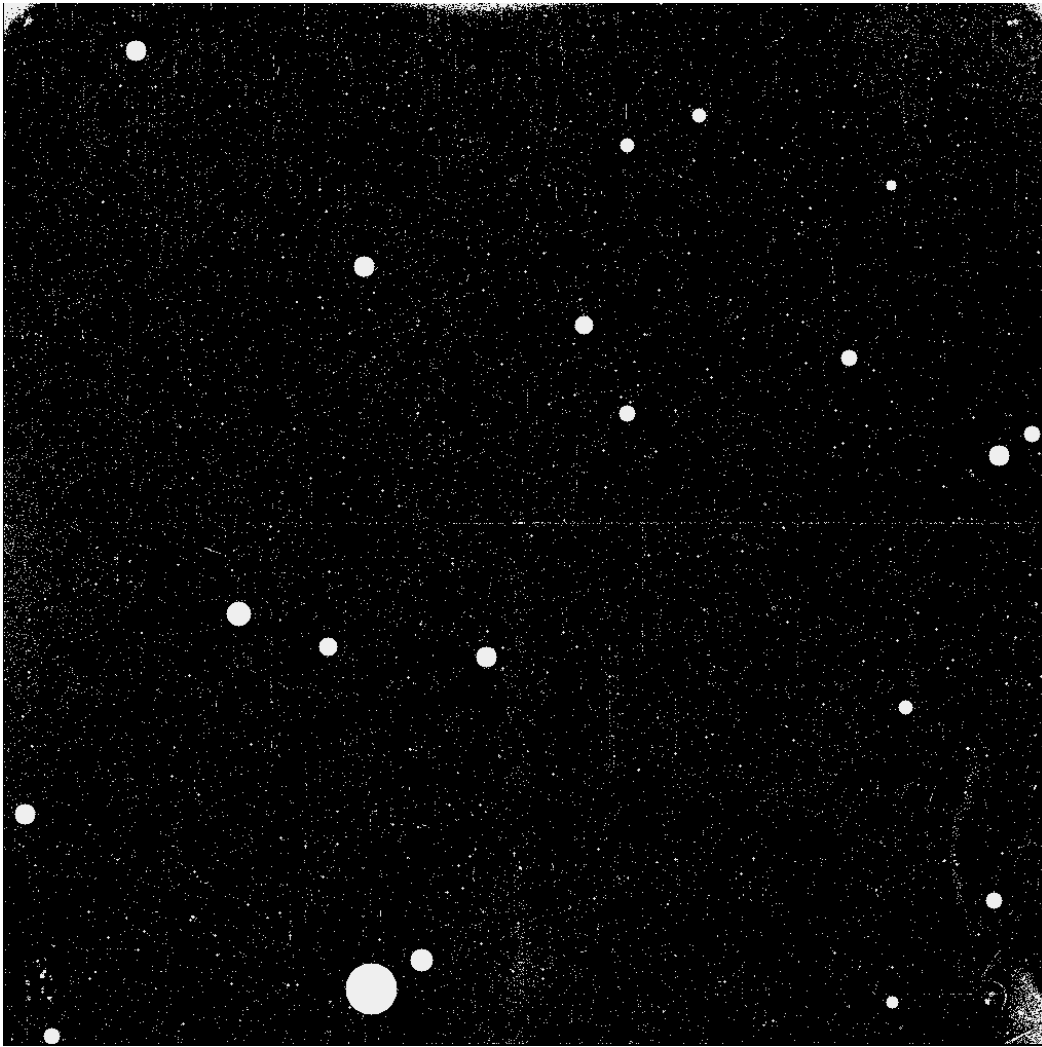


Figure 13: Map of all bad pixels on the active portion of IR detector. Reference pixels are not shown.

Recommendations

We recommend that observers dither their IR channel observations, in order to allow software such as Multidrizzle to fill in holes in individual images due to bad pixels.

References

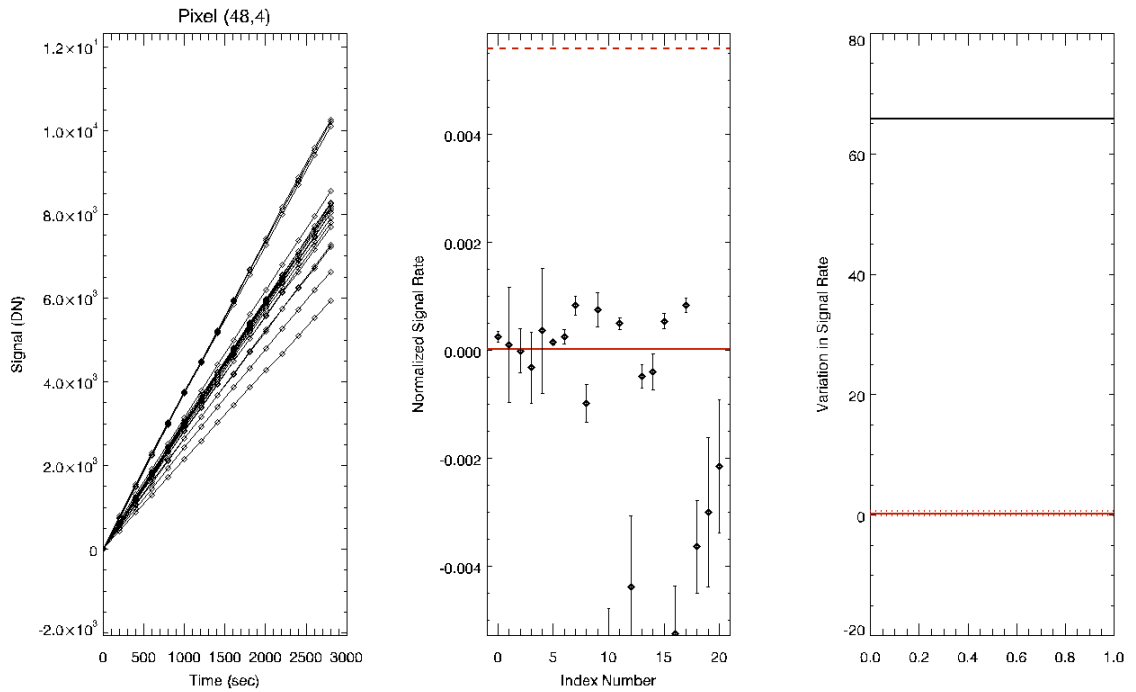
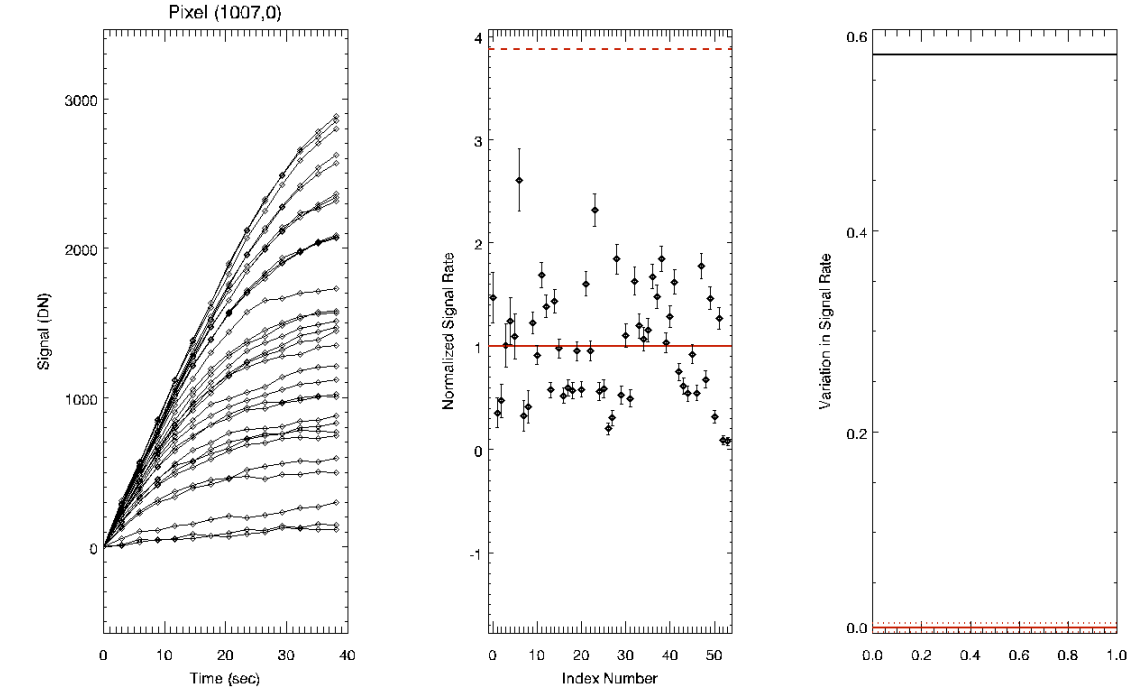
- Baggett, S. *WFC3 SMOV Proposals 11423/ 11543: IR FSM and Lamp Checks*. WFC3 ISR 2009-28. <http://www.stsci.edu/hst/wfc3/documents/ISRs/WFC3-2009-28.pdf> November 2009
- Hilbert, B., S. Baggett and M. Robberto. *Masking Technique on WFC3 IR Images*. WFC3 ISR 2003-06. <http://www.stsci.edu/hst/wfc3/documents/ISRs/WFC3-2003-06.pdf> July 2003.
- Hilbert, B. *WFC3 TV2 Testing: IR Channel Non-linearity Correction and Unstable Pixel Masking*. WFC3 ISR 2007-25. <http://www.stsci.edu/hst/wfc3/documents/ISRs/WFC3-2007-25.pdf> November 2007.
- Hilbert, B. *WFC3 SMOV Program 11433: IR Internal Flat Field Observations*. WFC3 ISR 2009-42. <http://www.stsci.edu/hst/wfc3/documents/ISRs/WFC3-2009-42.pdf> October 2009.
- Hilbert, B. and P. McCullough. *WFC3 SMOV Program 11420: IR Channel Functional Tests*. WFC3 ISR 2009-23. <http://www.stsci.edu/hst/wfc3/documents/ISRs/WFC3-2009-23.pdf> June 2009.
- Pirzkal, N., A. Viana, and A. Rajan, *The WFC3 IR "Blobs"*. WFC3 ISR 2010-06. <http://www.stsci.edu/hst/wfc3/documents/ISRs/WFC3-2010-06.pdf> April 2010.
- Robberto, M., C. Hanley, and I. Dashevsky. *The reference pixels on the WFC3 IR detectors*. WFC3 ISR 2002-06. <http://www.stsci.edu/hst/wfc3/documents/ISRs/WFC3-2002-06.pdf> June 2002.

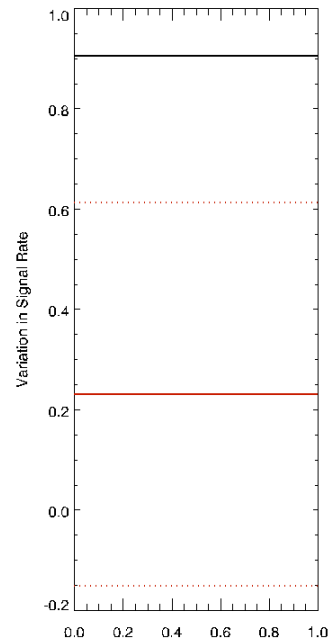
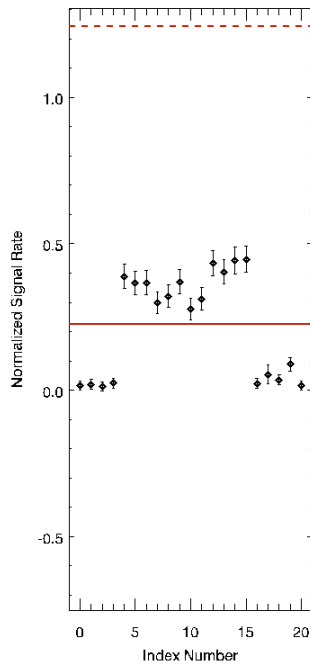
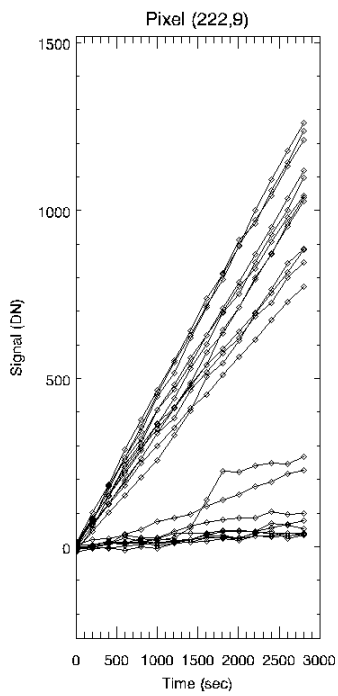
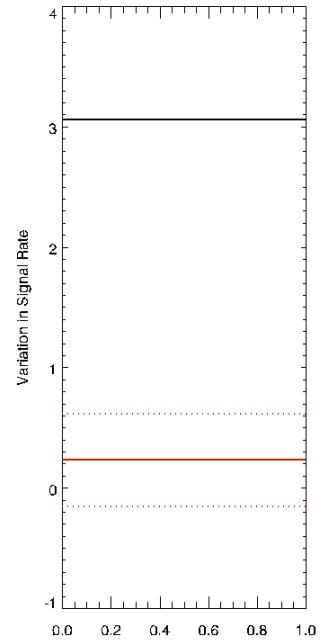
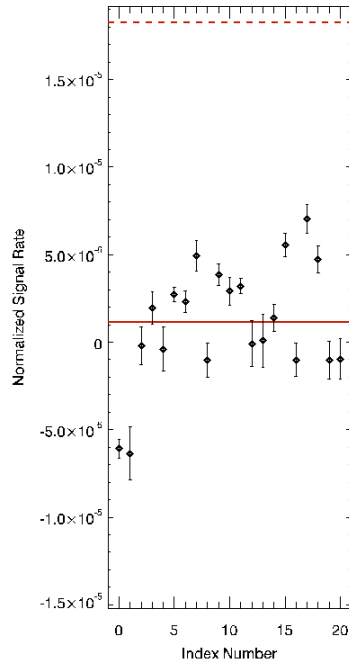
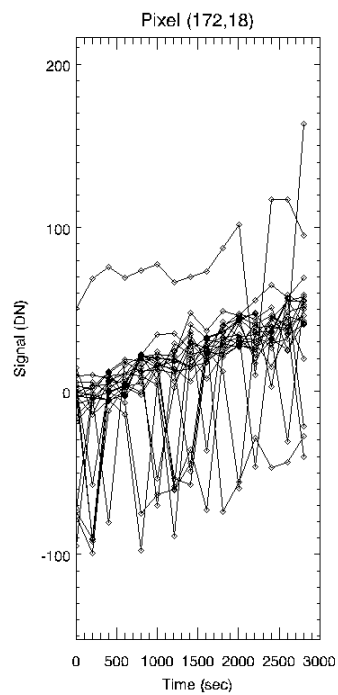
Appendix 1

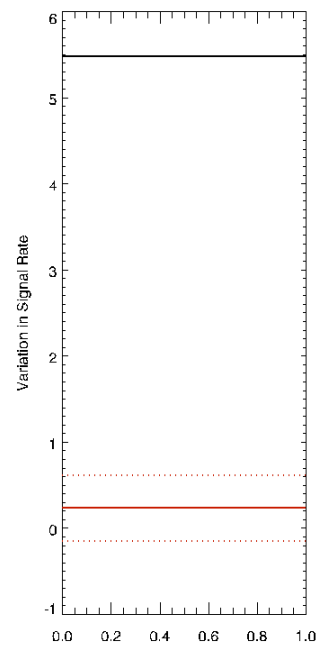
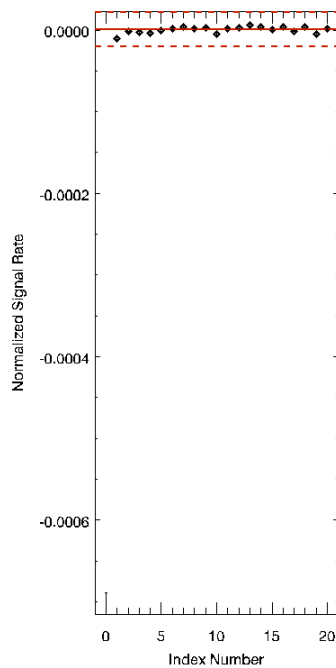
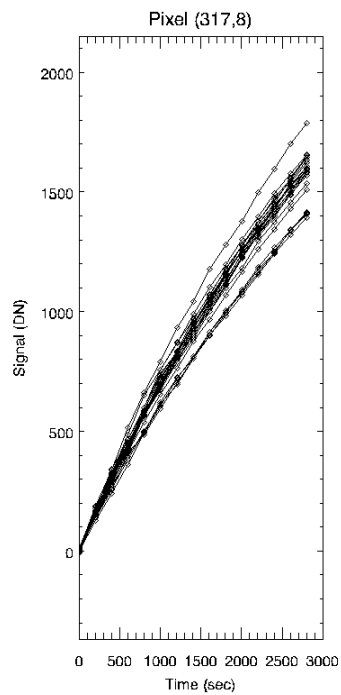
Filename	Proposal ID	Obs. Type	Filter	Exptime (sec)	Samp Seq / # Reads
iabz05xqq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz06geq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz07axq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz08ojq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz09edq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz10ghq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz11acq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz12d4q_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz13lfq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz14ljq_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz15h7q_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz16h4q_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iabz17q3q_raw.fits	11447	Dark	blank	2802.94	SPARS200 / 16
iaak01akq_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
iaak05irq_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
iaak08jfq_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
iaak022eiq_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
iaak22emq_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
iaak26fgq_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
iaak29oaq_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
iaak33p6q_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
iaak48w0q_ima.fits	11443	Tungsten Flat	F110W	43.98	RAPID / 16
ib9m12okq_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m17shq_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m25f9q_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m32zaq_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m36frq_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m47e6q_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m49keq_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m54lcq_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m59v1q_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m62vgq_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14
ib9m64vtq_ima.fits	11915	Tungsten Flat	F110W	38.12	RAPID / 14

Appendix 2

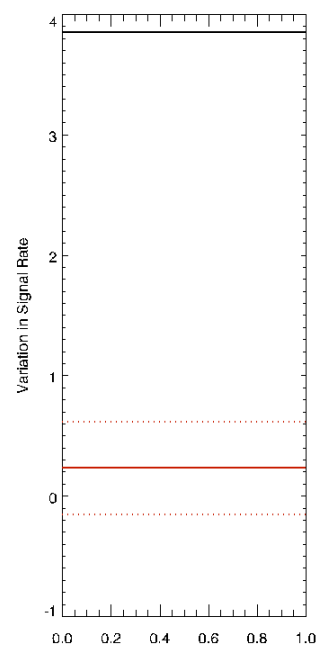
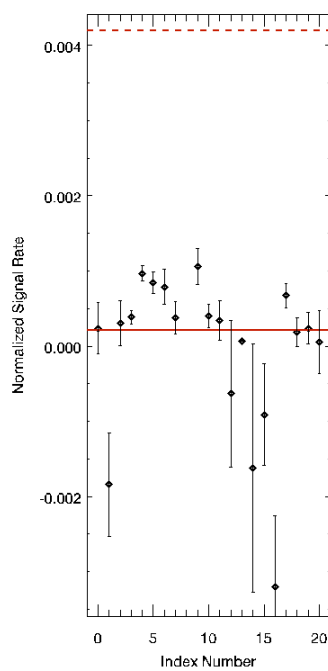
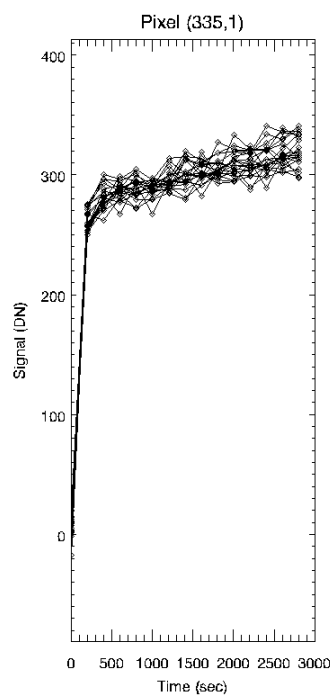
Plots of various types of unstable pixels, showing the variability of unstable pixels' behavior.



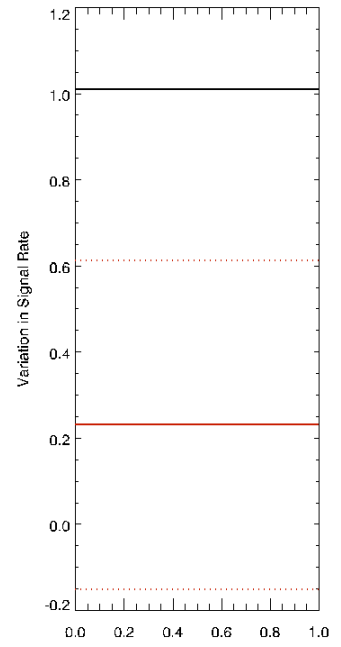
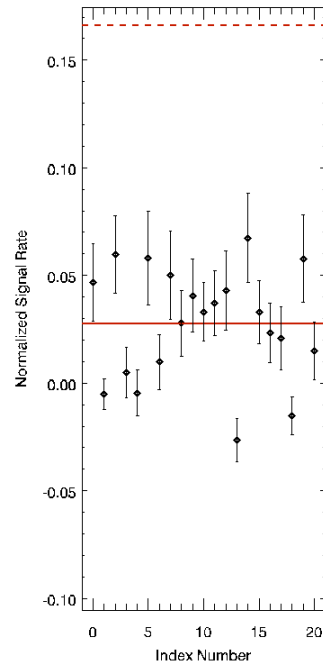
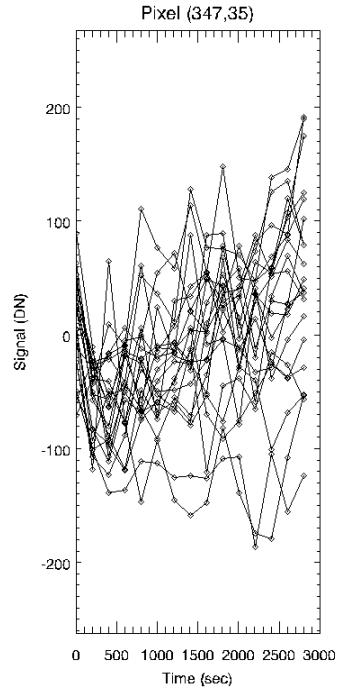




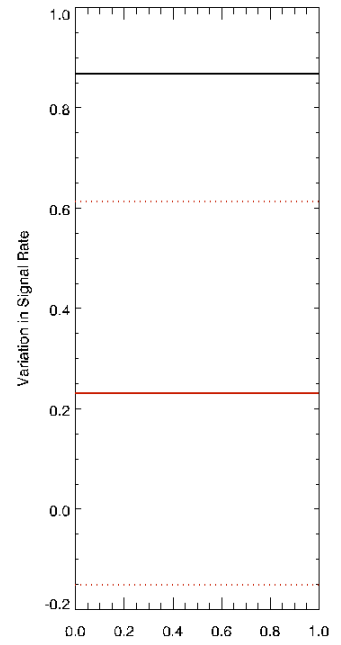
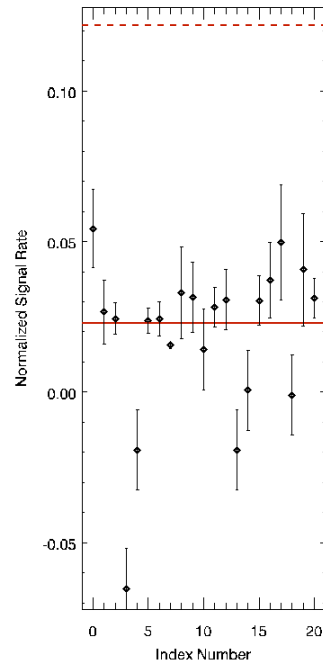
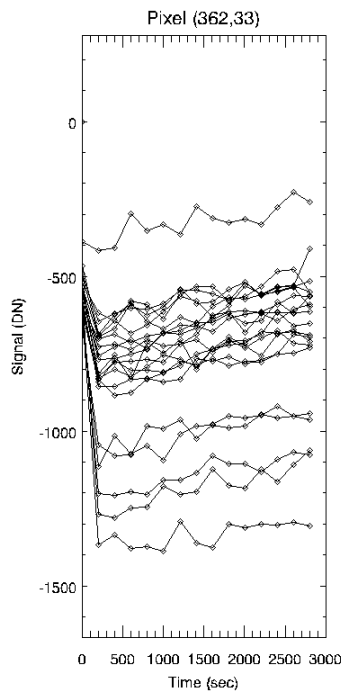
■



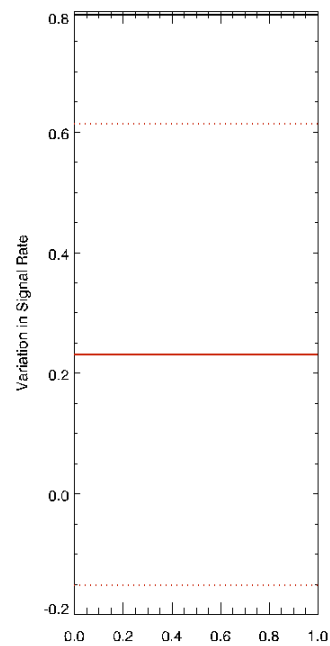
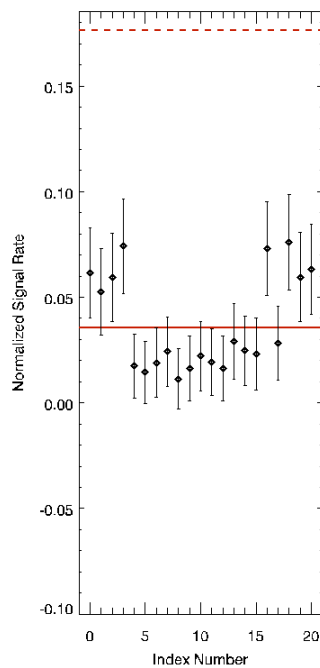
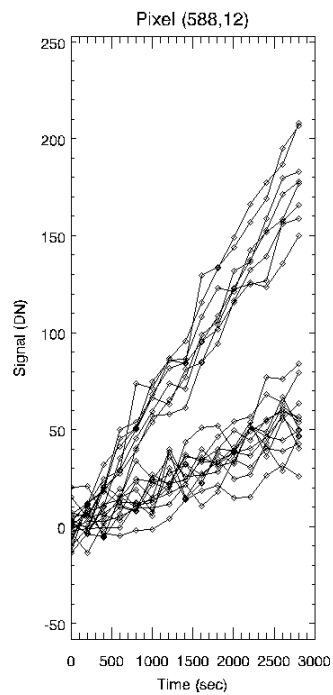
■



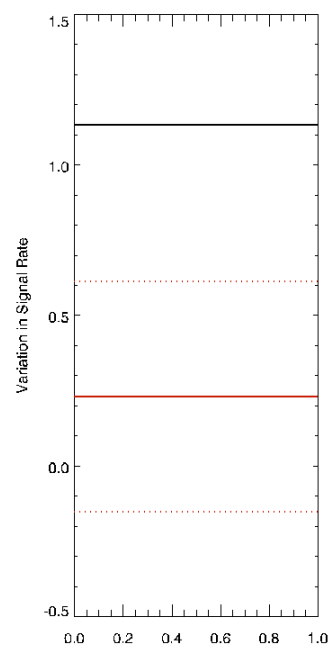
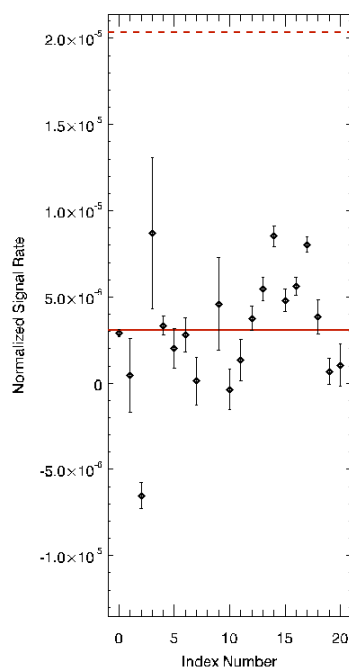
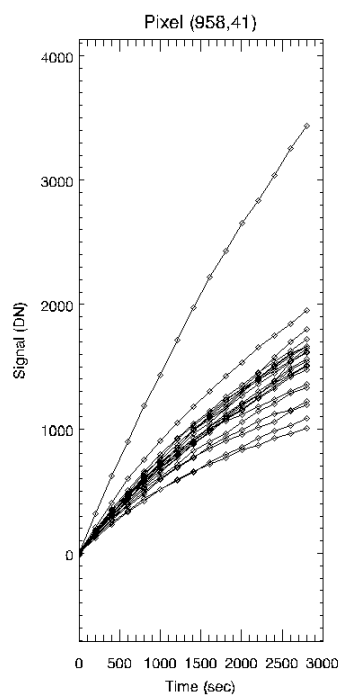
■



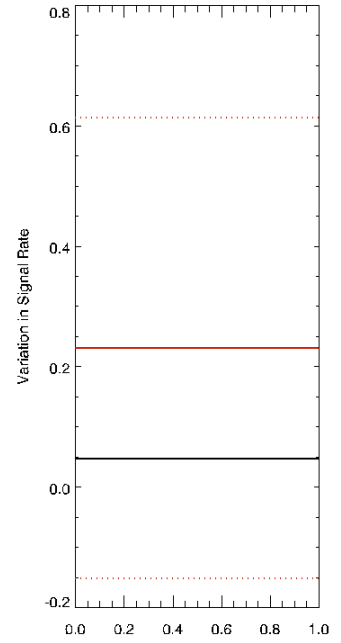
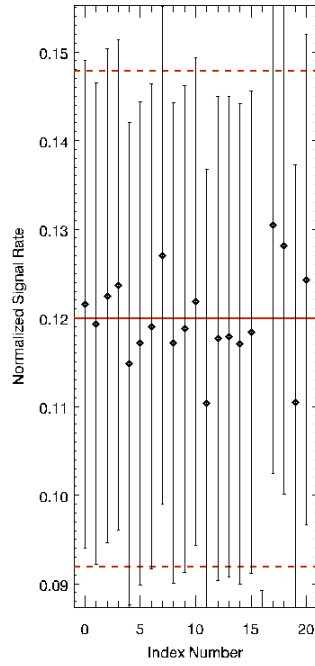
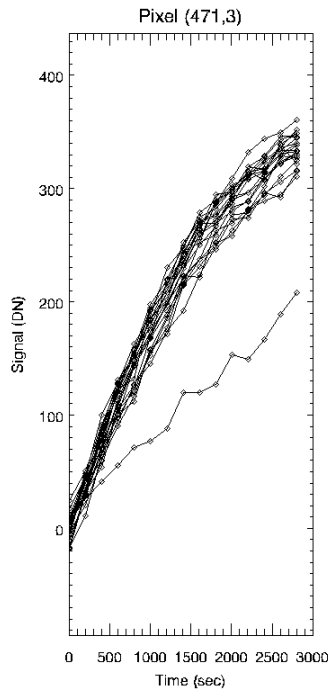
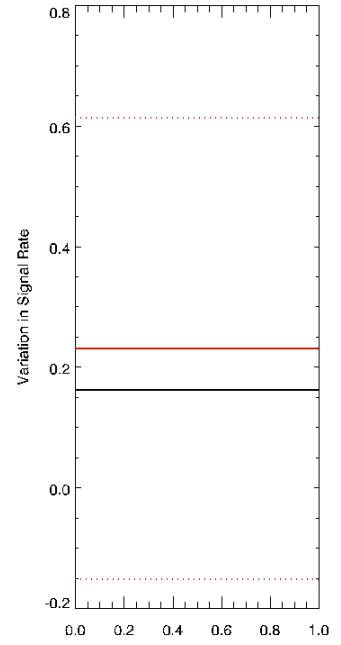
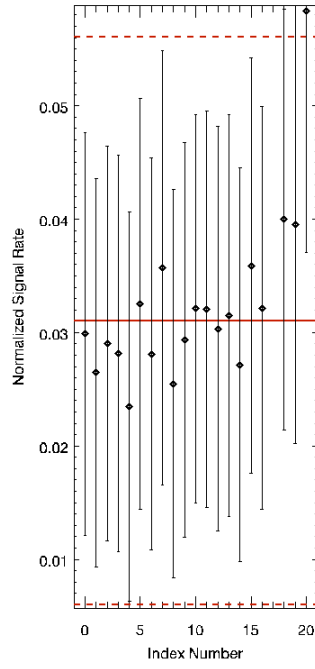
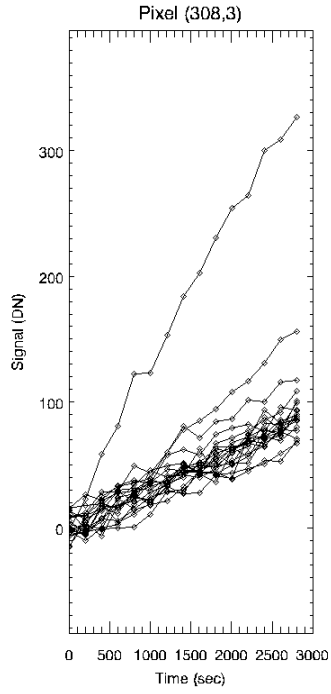
■

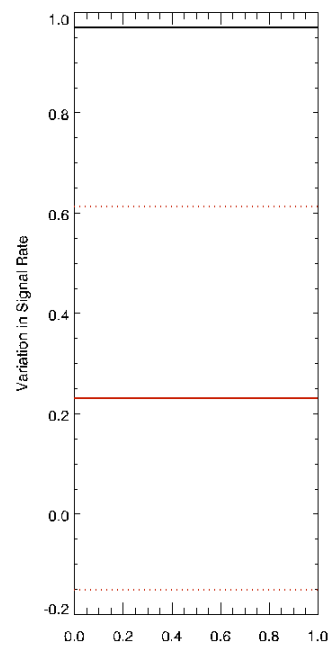
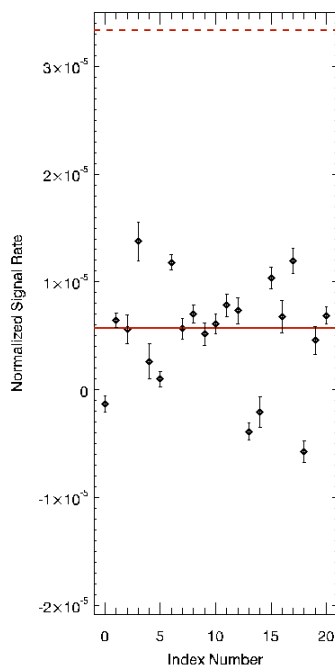
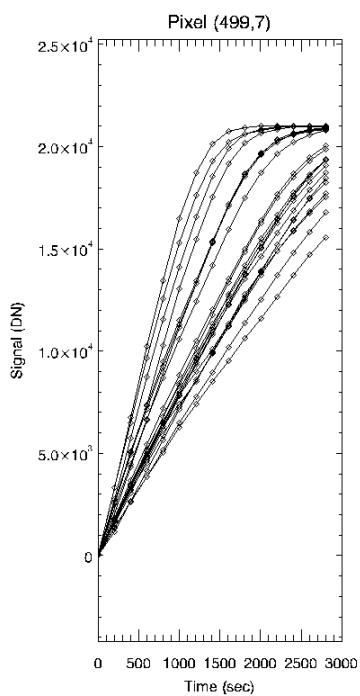
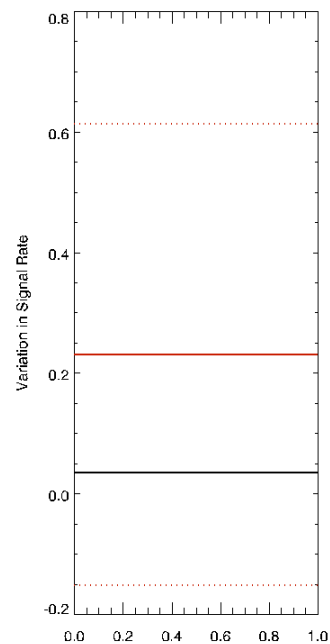
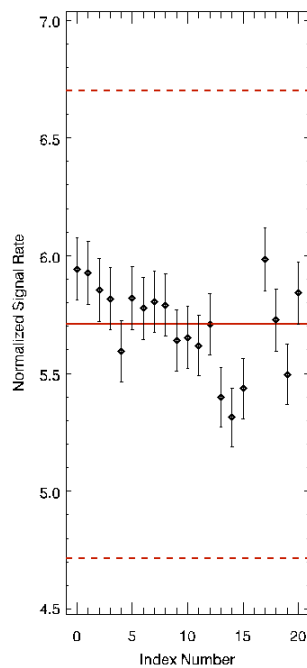
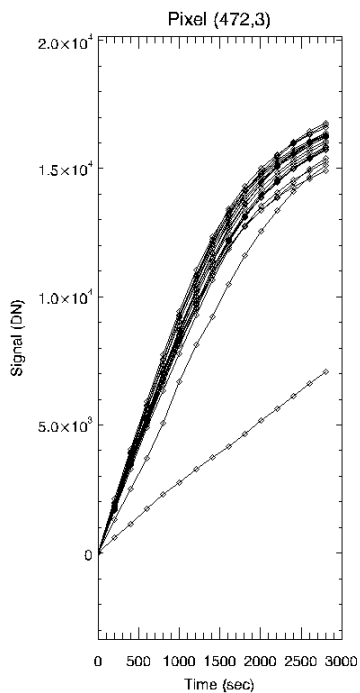


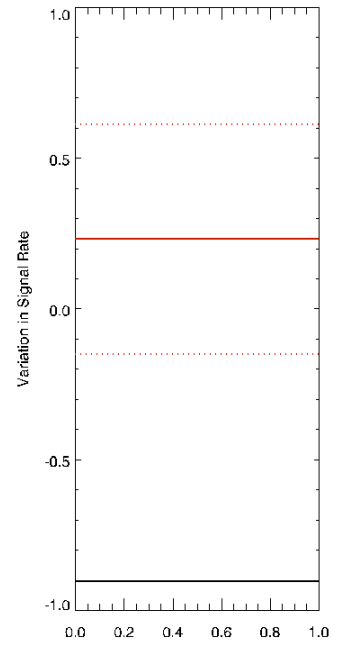
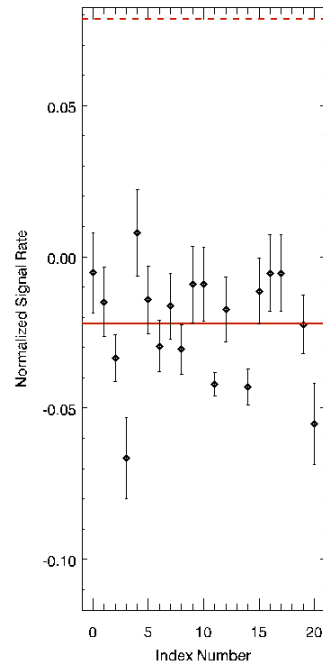
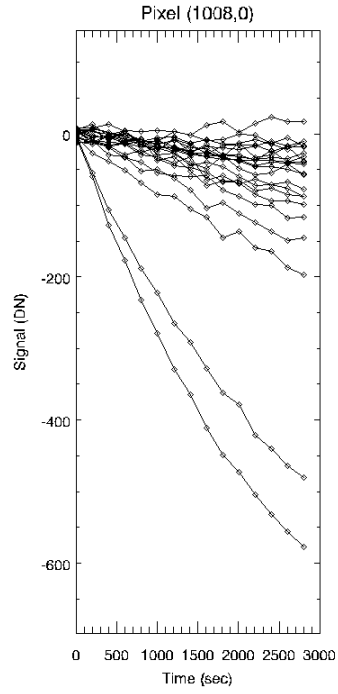
■



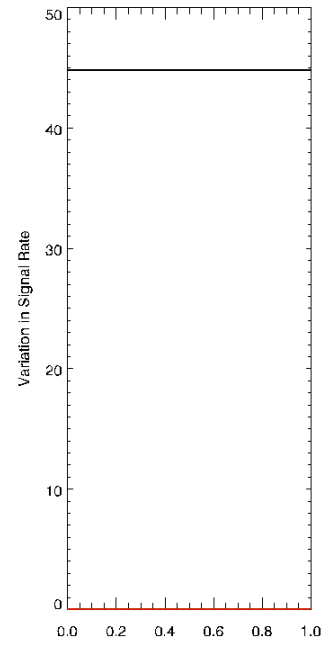
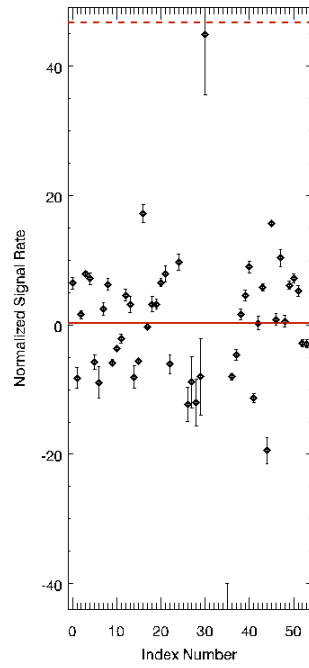
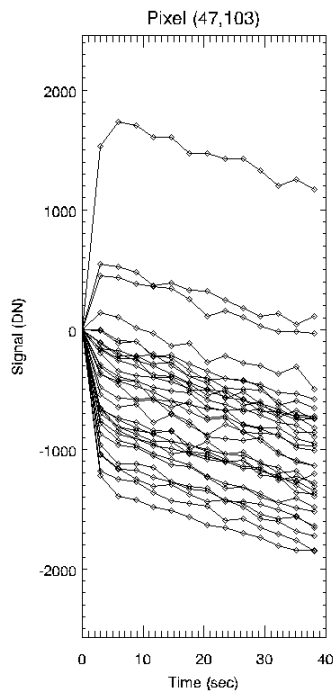
■



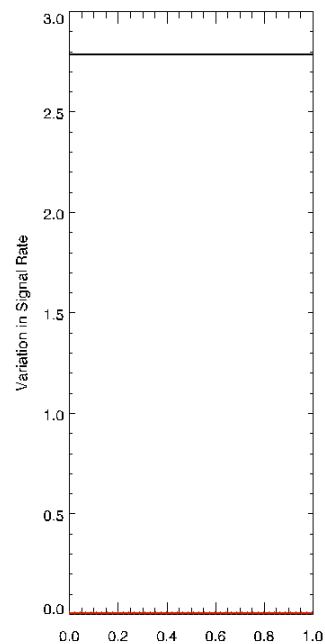
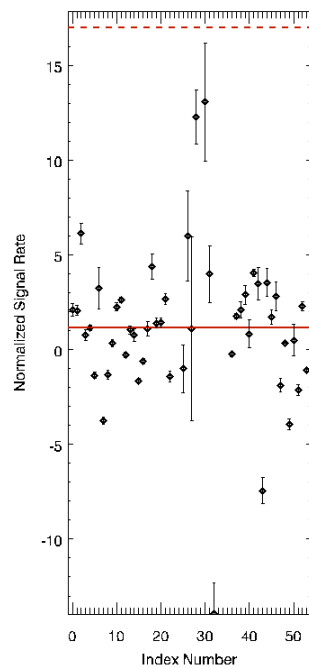
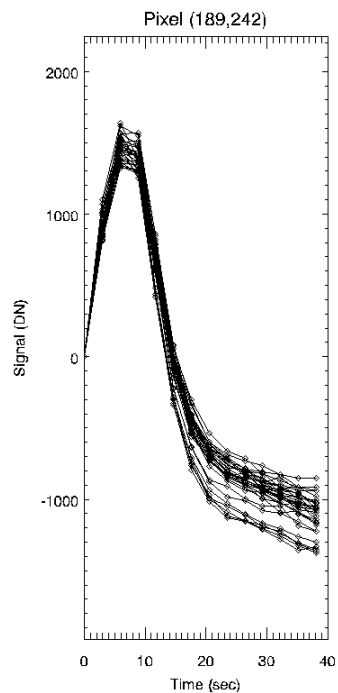




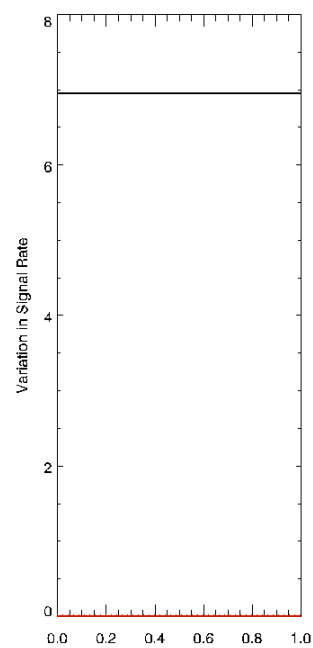
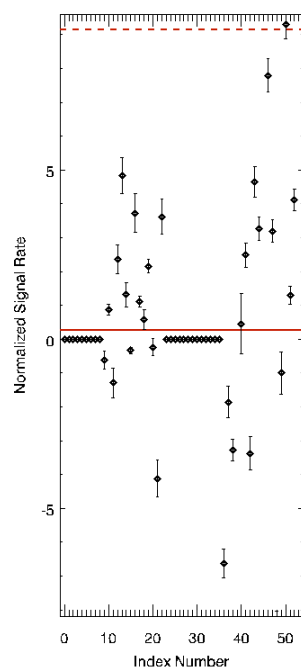
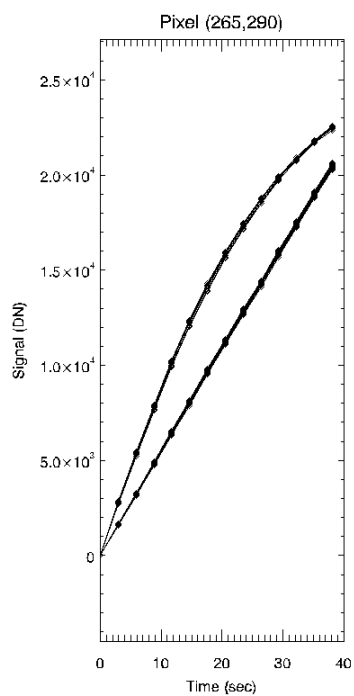
■



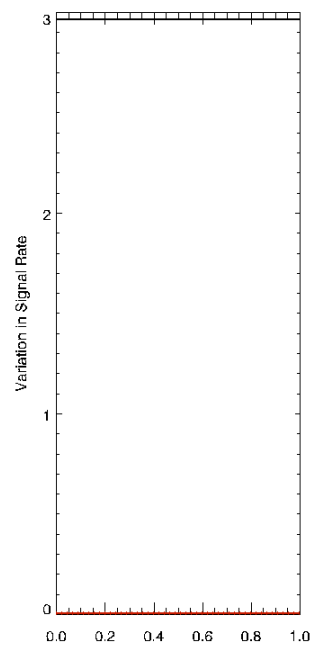
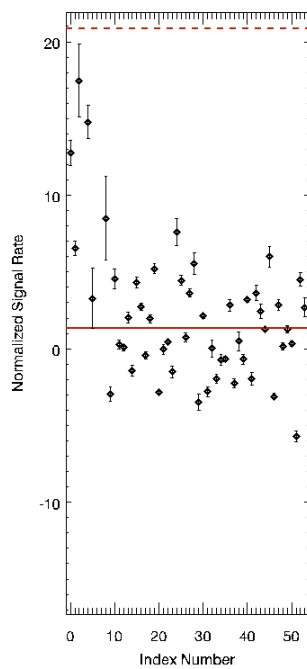
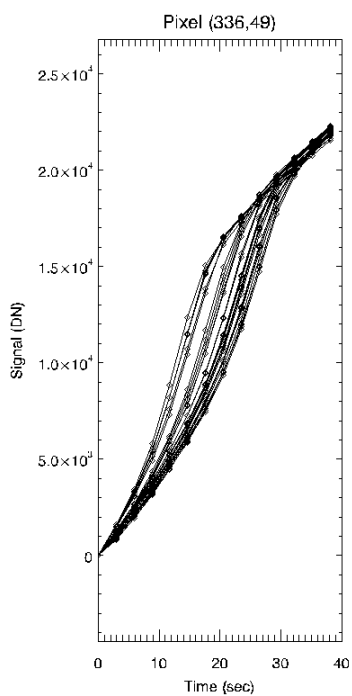
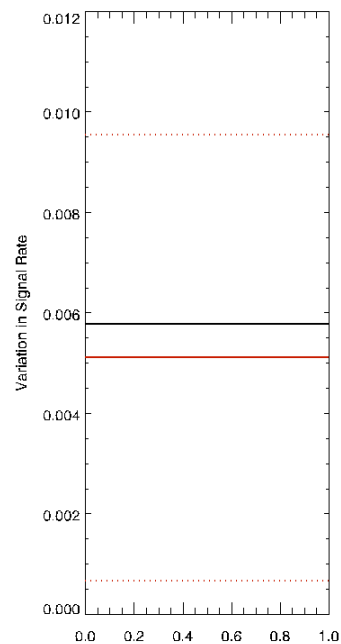
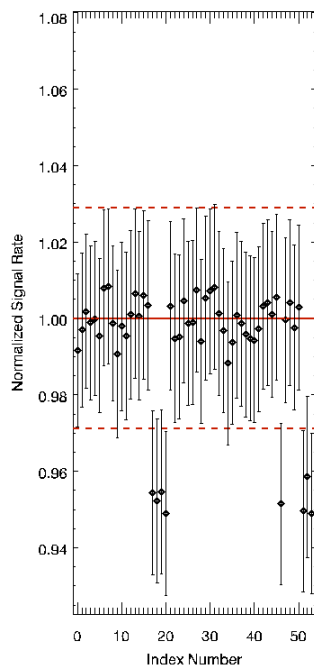
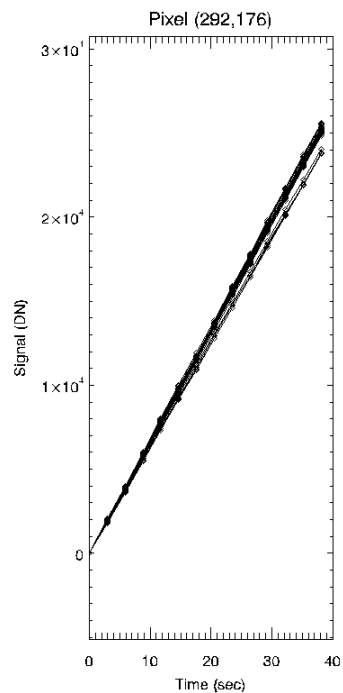
■

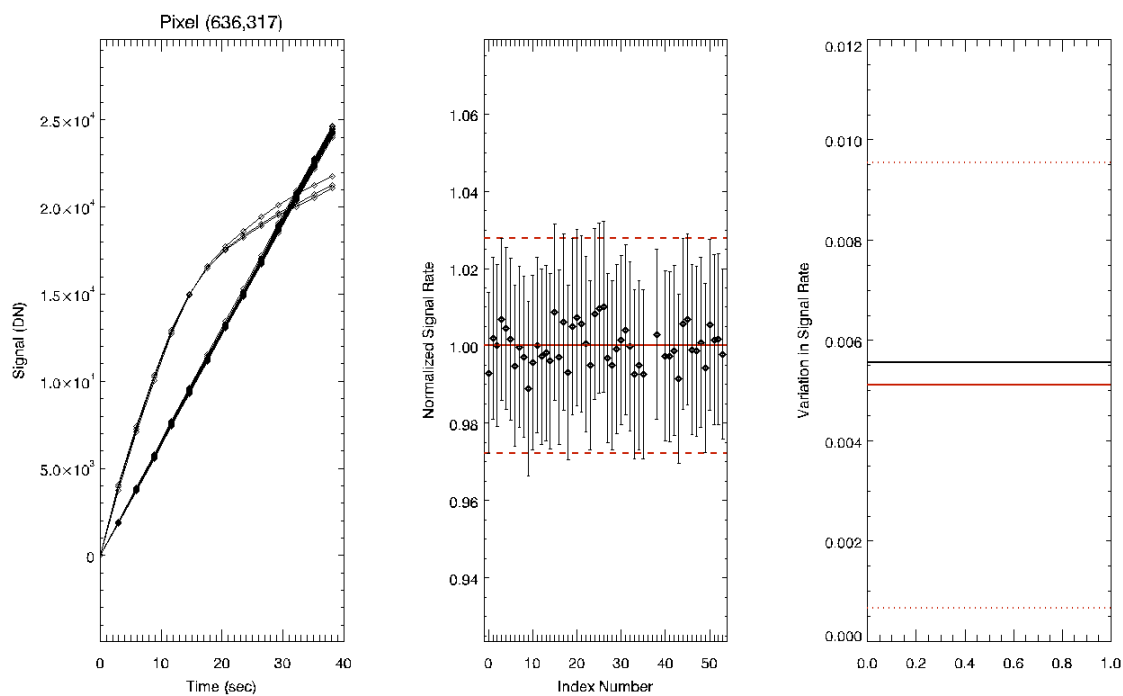
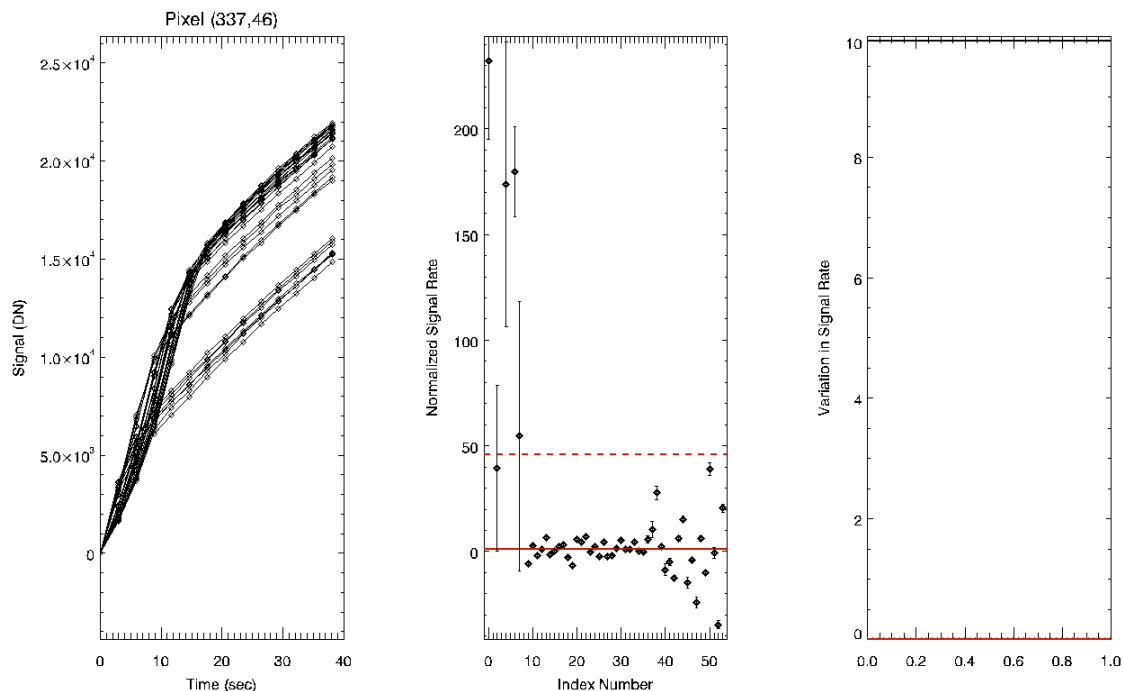


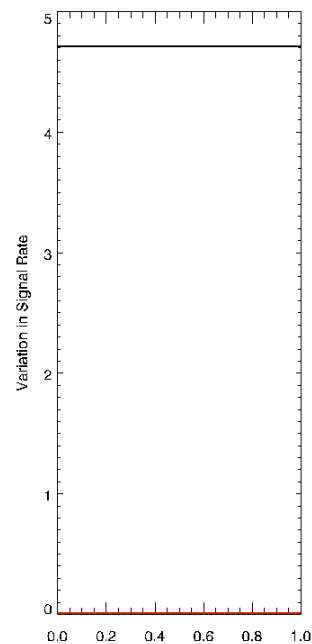
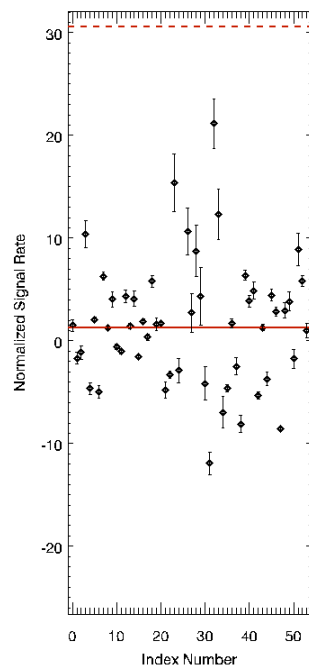
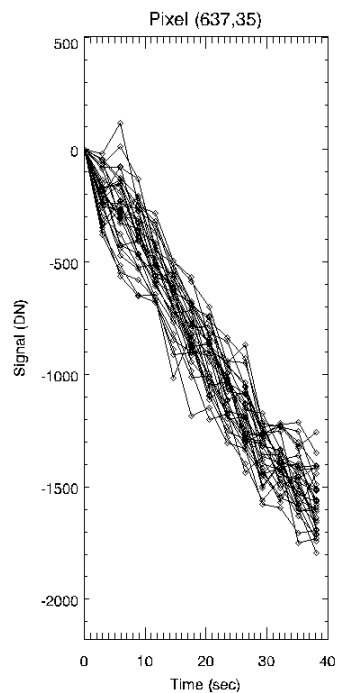
■



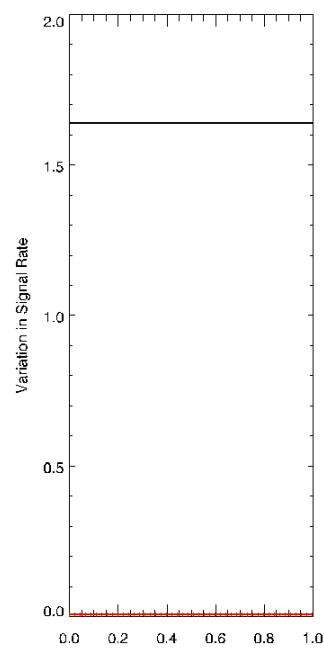
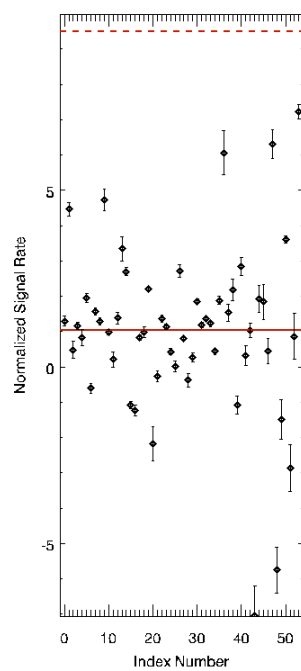
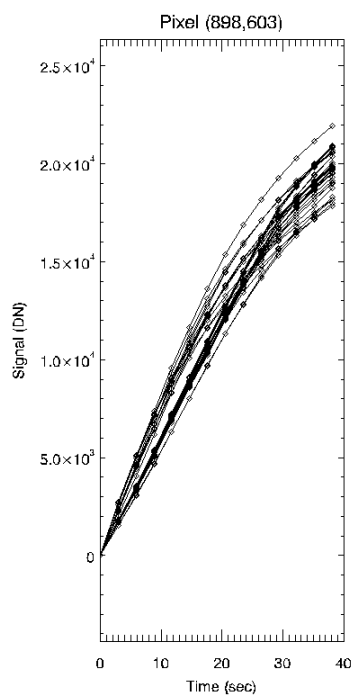
■







■



■

