

# WFC3/IR Cycle 19 Bad Pixel Table Update

---

B. Hilbert  
June 08, 2012

---

## ABSTRACT

*Using data from Cycles 17, 18, and 19, we have updated the IR channel bad pixel table for WFC3. The bad pixel table contains flags that mark the position of pixels that are dead, unstable, have a bad zeroth read value, or are affected by “blobs”. In all, 28,500 of the science pixels (2.77%) are flagged as bad. Observers are encouraged to dither their observations as a means of lessening the effects of these bad pixels. The new bad pixel table is in the calibration database system (CDBS) as w681807ii\_bpx.fits.*

---

## Introduction

We report here on an updated bad pixel table for the IR channel detector. As expected, extended exposure to the harsh environment of low Earth orbit has led to changes over time in the behavior of some of the pixels in the detector. The bad pixel table, which is a reference file in the Calibration Database System (CDBS) and contains a list of the pixels on the detector which are considered permanently bad, is applied by the *calwf3* calibration pipeline to IR channel data. By using *calwf3* with the most up-to-date bad pixel table, observers should be able to avoid using bad pixels in their analyses through the use of the data quality (DQ) arrays of the *ima* and *flt* versions of their observations.

## Data

This update was performed using calibration data collected during observing cycles 17, 18, and 19. A full list of the data files used is included in Appendix 1. Two types of observations were used for this bad pixel search: 56 internal flat field files and 117 dark current files. The flat field observations were all made through the F140W filter. All but three were taken with the RAPID sample sequence and had an exposure time of 38.1 seconds each. The other three were taken with the SPARS10 sample sequence and had an exposure time of 32.9 seconds each. The dark current files were all collected using the SPARS100 and SPARS200 sample sequences, which have exposure times of 1,403 and 2,803 seconds, respectively. These files represented the set of data which were found, through a manual search, not to have persistence sources. Other files, which were found to contain persistence, were not used in order to avoid having the persistence be interpreted by our scripts as instability in the signal.

All files were run through the *calwf3* data reduction pipeline prior to searching for bad pixels. In order to obtain a set of consistently calibrated *flt* files, we ran *calwf3* with the following data reduction steps turned on: DQICORR, ZSIGCORR, ZOFFCORR, BLEVCORR, UNITCORR and CRCORR. The DARKCORR step was also turned on for the processing of the internal flat field observations.

In order to be sure that effects from cosmic ray hits did not cause erroneous identification of bad pixels, our plan was to ignore all pixels flagged as impacted by cosmic rays in a given ramp. We also wanted to be sure that *calwf3*'s cosmic ray search was as effective as possible at identifying cosmic rays. Therefore, we used a modified version of the cosmic ray rejection table (CRREJTAB header keyword), where we changed the threshold used by *calwf3* in the cosmic ray search. By default, when looking at a pixel's signal values up the ramp, *calwf3* uses a  $4\sigma$  threshold (where  $\sigma$  is calculated from the noise properties of the pixel's signals) to identify cosmic rays. After experimenting with various values of the threshold, we decided to use a value of  $2.5\sigma$ . With this more aggressive value, *calwf3* was better able to identify and flag small-magnitude (i.e. tens of counts) cosmic ray hits in the data while still keeping to a minimum the number of falsely identified cosmic rays. With the large data set being used, we decided it was better to be aggressive and throw out a few false cosmic rays than to include unflagged small-magnitude cosmic rays in the analysis and potentially misinterpret them as unstable or bad pixels.

In addition, we used a soon-to-be released version of *calwf3* which has the added ability to identify and correct negative cosmic ray hits. (An example of a negative cosmic ray hit during a ramp is shown in Figure 1 of Hilbert and Bushouse, 2010.) This new ability prevented pixels which were impacted by negative cosmic ray hits from having anomalously low signal rates in the final output *flt* file due to the large jump downward in measured signal attributed to the cosmic ray. A large enough negative jump and subsequent low signal rate could potentially cause a pixel to be erroneously

flagged as an unstable pixel. With *calwf3*'s added ability to identify and fix these negative jumps, we believe that the resulting map of unstable pixels is more accurate than the previous version.

Changes to the default behavior of *calwf3* since the previous bad pixel table update meant that no pixels in the calibrated *flt* images had a value of zero due to bad pixel flags. Previously, many of the pixels which were flagged in the data quality array were automatically given a value of zero in the *flt* file. In the previous bad table update, these pixels were given non-zero values by manually line-fitting their measured signals up the ramp. For this bad pixel table update we did not have to perform any manual line fitting. This implies a more consistent set of pixel values were used in this bad pixel table update.

## Analysis

For this update we followed the same procedure used during the last update, outlined in Hilbert and Bushouse, 2010. The four types of bad pixels contained in the table include: bad in zeroth read, dead, unstable, and pixels within a blob.

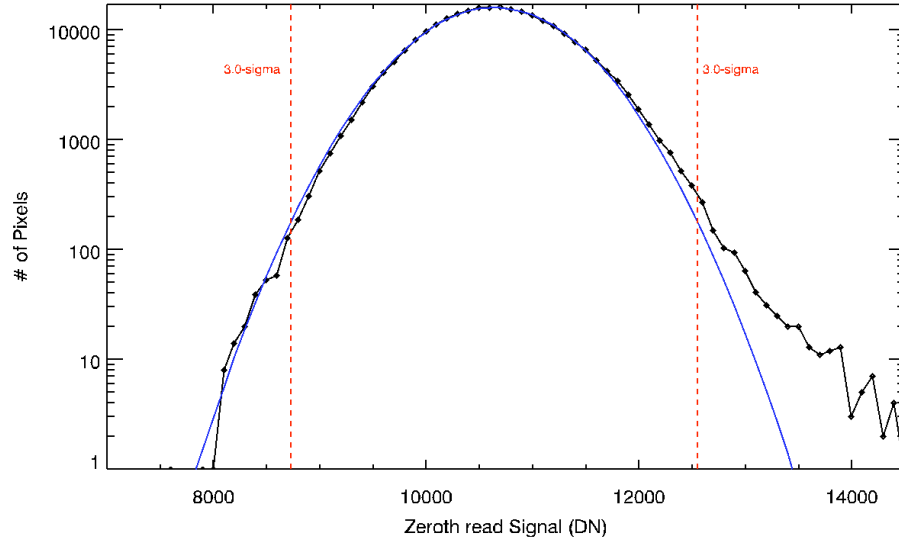
### *Bad in Zeroth Read Pixels*

These are pixels which exhibit anomalous signals in the zeroth read of a ramp, usually due to being shorted or unbonded (Hilbert, 2003). To identify these pixels, we began with the zeroth reads of the raw (untouched by *calwf3*) dark current ramps listed in Appendix 1. Using the zeroth reads from all of the input ramps, we created a sigma-clipped mean zeroth read image. We then created a separate histogram of the values in this mean zeroth read for each quadrant of the detector. Since raw ramps have had neither the mean bias level nor the zeroth read subtracted from them, and with each quadrant of the detector read out through a different amplifier, it was important to treat each quadrant separately. A Gaussian was fit to the histogram from each quadrant, and pixels falling more than  $3\sigma$  from the mean value were flagged as bad. Figure 1 shows an example of a histogram and Gaussian fit.

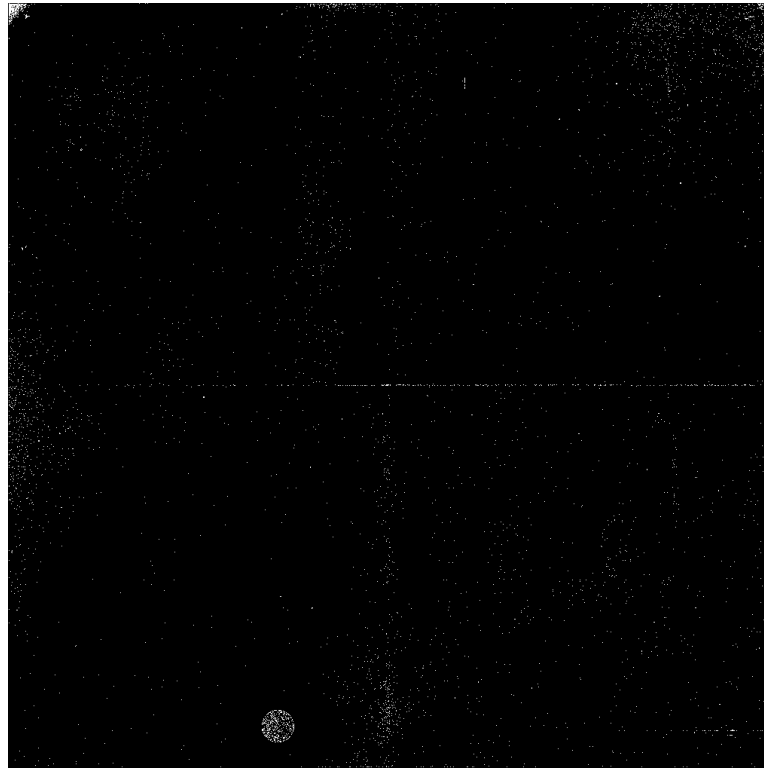
We also repeated this process on each input ramp individually, in order to catch any pixels which may have been occasionally beyond the  $3\sigma$  threshold. With this two-step method, we identified 4,923 pixels (0.479% of the detector's light-sensitive pixels) which are bad in the zeroth read. This is a net decrease of 67 pixels over the population in the current bad pixel table. However, there were not 67 pixels that simply moved within the  $3\sigma$  limits of the histogram. Rather, there were 294 pixels which were previously flagged as bad that moved within the  $3\sigma$  limits and are now considered good. These were mostly offset by 227 pixels which were previously good but now fall outside the  $3\sigma$  limits. With an order of magnitude more dark current files going into the creation of the mean zeroth

read compared to the previous bad pixel table update, we attribute these small changes in the number and locations of bad pixels to better statistics.

Figure 2 shows a map of the updated bad zeroth read pixels, which are flagged with a value of 8 in the bad pixel table.



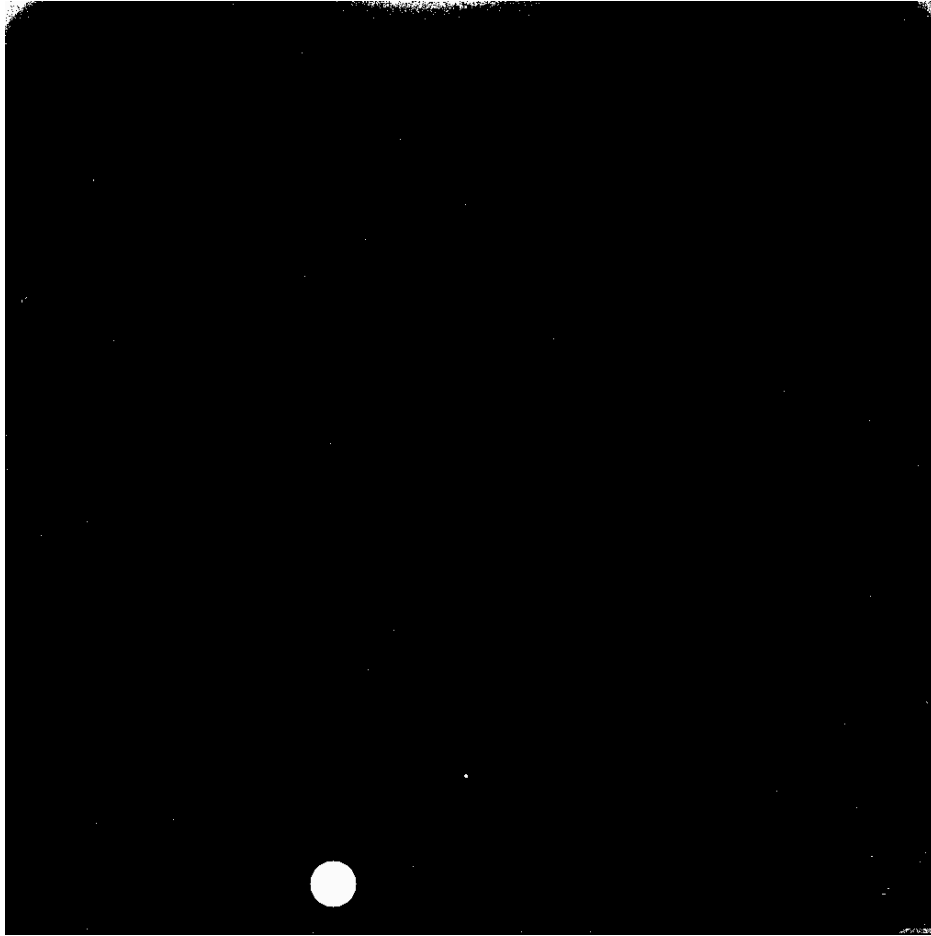
**Figure 1:** Histogram of pixel signals in quadrant 1 of the mean zeroth read image constructed from the raw dark current files.



**Figure 2:** Updated map of pixels that are bad in the zeroth read.

### ***Dead Pixels***

The next population of bad pixels to be updated was the dead pixels. These pixels display an abnormally low quantum efficiency (QE) and therefore measure very little or no signal when illuminated. As such, we used *calwf3*-processed *flt* images of the internal flat field observations listed in Appendix 1 for this search. Unlike in the previous search for dead pixels (Hilbert and Bushouse, 2010) we decided that the large amount of data acquired in recent observing cycles allowed us to use flat field observations taken through a single filter, F140W. During the previous update, flat fields from multiple filters were combined into a mean flat field. For the current update, we simply combined the 56 individual F140W flat field images into a mean flat field image. This was done by calculating the sigma-clipped mean signal on a pixel-by-pixel basis. As in the previous update, two different searches were made to find dead pixels. The first, which was for any pixels in the mean flat field with zero or negative signal, netted 2,399 pixels (0.23% of the science pixels). For the second search we stepped pixel-by-pixel across the mean flat field image. For each pixel, we calculated the sigma-clipped mean signal in a 50x50 pixel box centered on the pixel. If the pixel had a signal that was less than 30% of the mean signal level of the surrounding box, it was flagged as dead. In this way, we were able to identify pixels with abnormally low QE-values, while not erroneously flagging pixels located on areas of the detector which received only vignetted illumination. This search identified 1,397 dead pixels. Finally, as was done previously, the pixels within the “death star” were manually added to the list of dead pixels. In all, this update resulted in the identification of 3,358 dead pixels (0.327%). This is 552 fewer than the 3,910 found by Hilbert and Bushouse (2010). Similar to the case for the bad zeroth read pixels, this net difference of 552 pixels comes from 80 pixels which were previously not flagged as dead but were found to be in this search, and 632 which were previously marked as dead but not identified as dead in this search. These variations can be attributed to pixels with signal rates very close to the 30% cutoff threshold mentioned previously slipping above or below the threshold in the expanded set of data used in this search. Figure 3 shows a map of the dead pixels found in the current search. These pixels are flagged with a value of 4 in the final bad pixel table.



**Figure 3:** *Updated dead pixel mask. The majority of the dead pixels are confined to the death star and the corners and edges of the detector.*

### ***Unstable Pixels***

Once again, the flavor of pixel which makes up the majority of the bad pixel table is the unstable pixel. These pixels, flagged with a value of 32, exhibit non-repeatable behavior across nominally identical observations. This behavior manifests itself in many different ways, making “unstable pixels” a broad definition. Given a set of 100 nominally-identical flat field ramps, some unstable pixels will behave repeatably for 99 ramps, but display a nicely linear ramp with a fraction of the nominal signal rate in the 100<sup>th</sup> ramp. Others will jump back and forth between two well-behaved states, but with signal rates that are very different. Still others will never exhibit the same signal rate twice, or show a variability in their linearity response or saturation level from ramp to ramp. See the gallery in Appendix 2 of Hilbert and Bushouse (2010) for examples.

We used an identical method with identical thresholds for the current unstable pixel search as were used in the previous update. Two separate searches were each performed

on the set of flat field ramps, and then on the set of dark current ramps, as previous analysis showed that some unstable pixels were only unstable in one type of observation.

### *Method 1: Temporal Search*

The goal of the first search was to find pixels which displayed only a small number of deviant ramps amongst a large number of nominally identical observations. For this we used a series of *flt* images and compared a pixel's value in a sigma-clipped mean flat field (or dark current) image to its values in the individual flat fields (or dark current files) used to create the mean. Since we are examining each pixel's signal values through time, we refer to this as the "temporal search" for unstable pixels. If the value in any of the individual *flts* was more than  $5\sigma$  from that in the mean *flt*, then the pixel was flagged as unstable. Using the flat field images listed in Appendix 1, this method found 1,493 unstable pixels (compared to 2,223 in the previous bad pixel table). Repeating the process on the dark current images yielded 12,667 flagged pixels (compared to 6,001 previously). Combining the two, 13,254 pixels were flagged as unstable in the temporal search. Figure 4 shows maps of these unstable pixels found in both the flat field and dark current data.

As can be seen in Table 1, this is a large increase in the number of dark current file-identified unstable pixels, while using an identical search method and threshold value used in the previous bad pixel table update. The mean dark current image for this search was made with 117 files, compared to only 13 files in the previous bad pixel table update. This implies a significantly higher signal-to-noise mean dark current image, allowing unstable pixels to be more easily identified. Also, with 117 images taken over 637 days, this dataset provided a much longer timescale and sample size within which unstable pixels could reveal themselves, compared to the 13 images over 33 days used in the previous search.

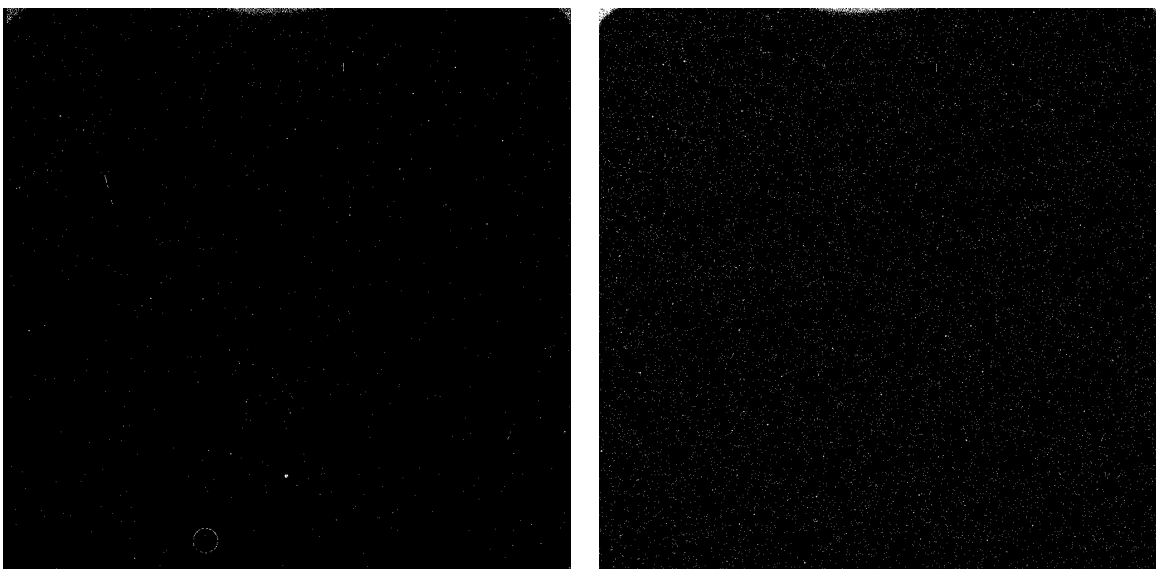
We believe that false positives are not a significant contributor to the unstable population identified in this search. For comparison, in an ideal detector where each pixel's signal rates follow a Gaussian distribution centered about a mean value, we would expect to see a value outside of the  $5\sigma$  threshold only 0.00006% of the time. For an input data set of 117 dark current images, each with  $1014 \times 1014$  pixels, this implies that in an ideal detector, we would expect to see only 72 pixels outside the  $5\sigma$  range from the Gaussian variability in signal rate. This is roughly 0.5% of the 12,667 found in this search.

We also took steps to be sure that persistent signal from observations made prior to each of the dark current ramps was not causing pixels to be accidentally identified as unstable. As we stepped through the dark current input images and compared each to the mean dark current image, we created a map of the locations of pixels identified as unstable in that particular input image. By examining each map, we were able to confirm

that the unstable pixels identified did not fall into point source-like patterns, suggesting that the pixels were truly unstable.

	Flat Field Images	Dark Current Images	Combined
Updated Search	1,493	12,667	13,254
Previous Bad Pixel Table	2,223	6,001	7,504

**Table 1:** Number of unstable pixels found using the temporal search on the flat field and dark current data, compared to the number found in the previous bad pixel table update.

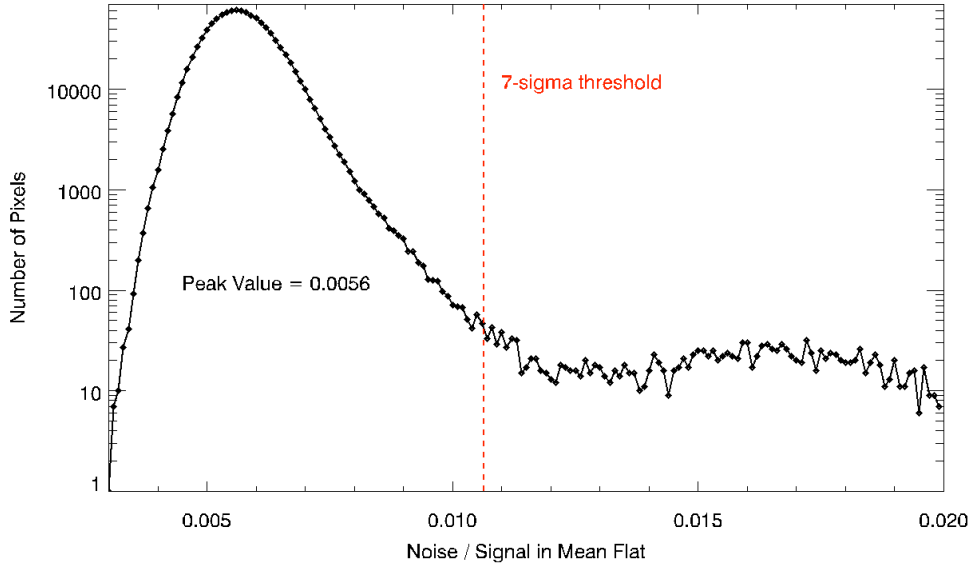


**Figure 4:** Maps of unstable pixels found using the temporal search on flat field images (left), and dark current images (right).

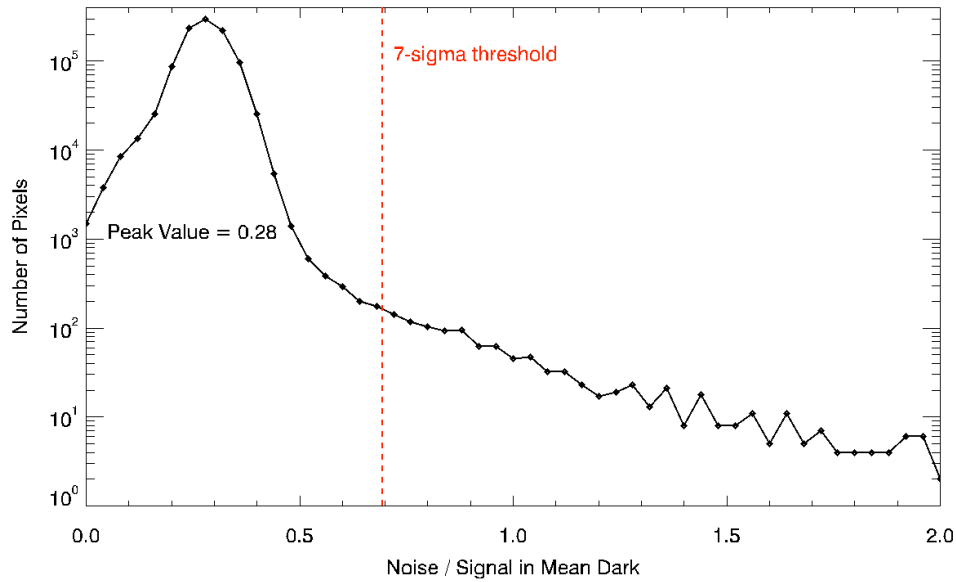
### *Method 2: Spatial Search*

The second type of unstable pixel search was designed to catch pixels which are wildly unstable and variable from ramp-to-ramp. These pixels have a very large spread of signal rates across the input data, meaning that a simple sigma threshold, such as that used in the temporal search above, would not necessarily catch these pixels. Therefore, we used a spatial comparison of pixels across the detector for this search. We first calculated the sigma-clipped standard deviation of each pixel's values through the set of input images. This created a map of standard deviations across the detector. Histograms of these maps for the mean flat field and dark current images are shown in Figures 5 and 6. We then calculated the sigma-clipped mean and standard deviation of the standard deviation map, as a way of defining the “mean” behavior across the detector. Finally, any pixel which had a value in the standard deviation map that was more than  $7\sigma$  away from the mean of the standard deviation map was flagged as unstable. Another way to

explain this filter is to say that a pixel caught by this search method is considered unstable due to the fact that its measured signal rates vary from ramp-to-ramp much more than the average pixel's signal rates do. As with the temporal search above, this search was performed first on the set of flat field *flt* files, and then separately on the set of dark current *flt* files. A more detailed explanation of both the temporal and spatial searches for unstable pixels are given in Hilbert and Bushouse (2010). Figure 7 shows maps of the unstable pixels from the flat field and dark current data identified using this method.



**Figure 5:** Histogram of the noise/signal ratio image created from the mean flat field and used in the spatial unstable pixel search.

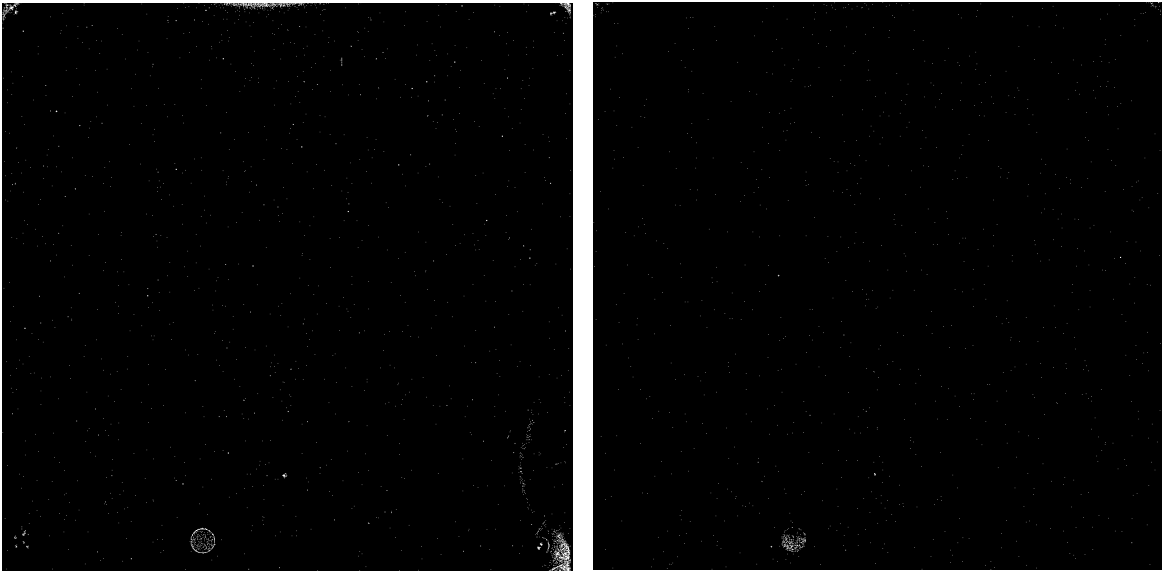


**Figure 6:** Histogram of the noise/signal ratio image created from the mean dark current image and used in the spatial unstable pixel search.

Table 2 shows the results of this search on the flat field and dark current data. As expected, with a similar number of individual flat field images used in this search compared to the previous bad pixel table update (56 files, compared to 54 previously), we identified a similar number of unstable pixels in the mean flat field. And with an order of magnitude more dark current images going into the present search compared to the previous update, the increased signal-to-noise in the mean dark current image meant that fewer pixels fell outside the  $7\sigma$  threshold in the current search.

	Flat Field Images	Dark Current Images	Combined
Updated Search	3,908	1,332	5,041
Previous Bad Pixel Table	3,795	3,318	6,633

**Table 2:** Number of unstable pixels found using the spatial search on the flat field and dark current data, compared to the number found in the previous bad pixel table update.

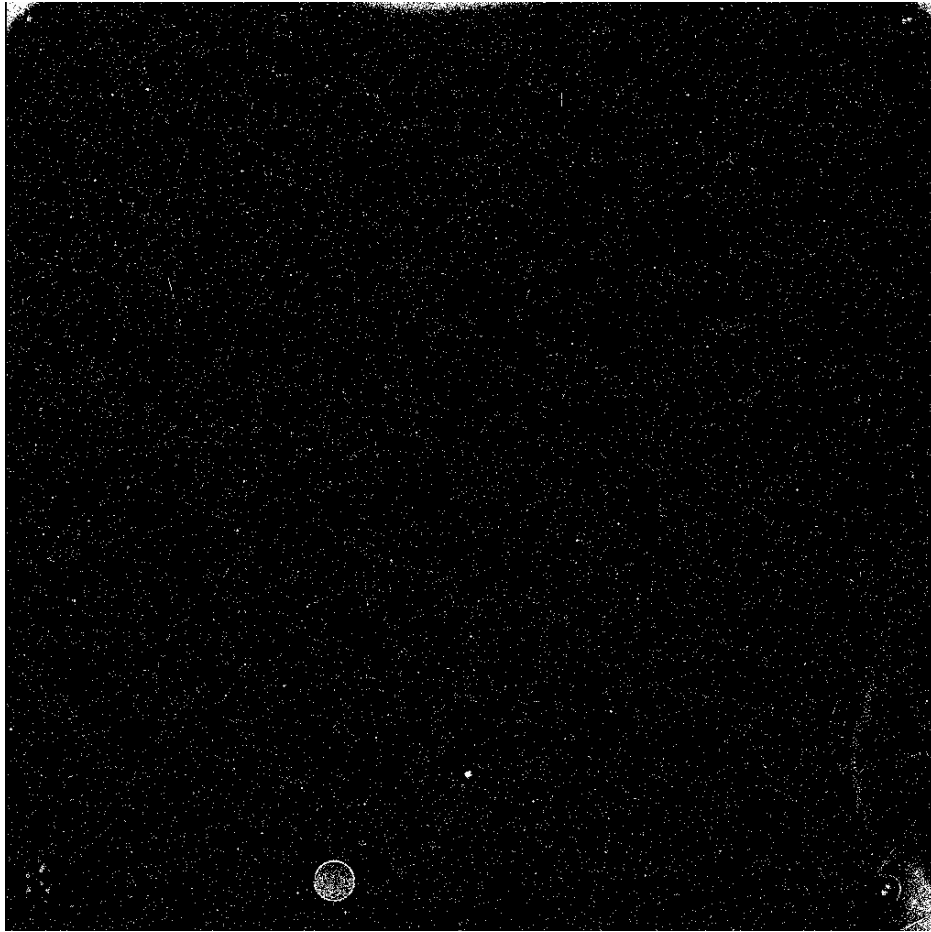


**Figure 7:** Map of unstable pixels netted by search method 2 on the flat field (left) and dark current (right) images.

### *Total*

Combining the results of the two searches on the flat fields and dark current images, we arrive at the updated unstable pixel map, shown in Figure 8. This map has a total of 16,500 pixels flagged as unstable, compared to 10,885 in the previous bad pixel table.

As mentioned previously this number has increased primarily due to the much larger data set of dark current files used for this search, yielding more unstable pixels in the temporal search.



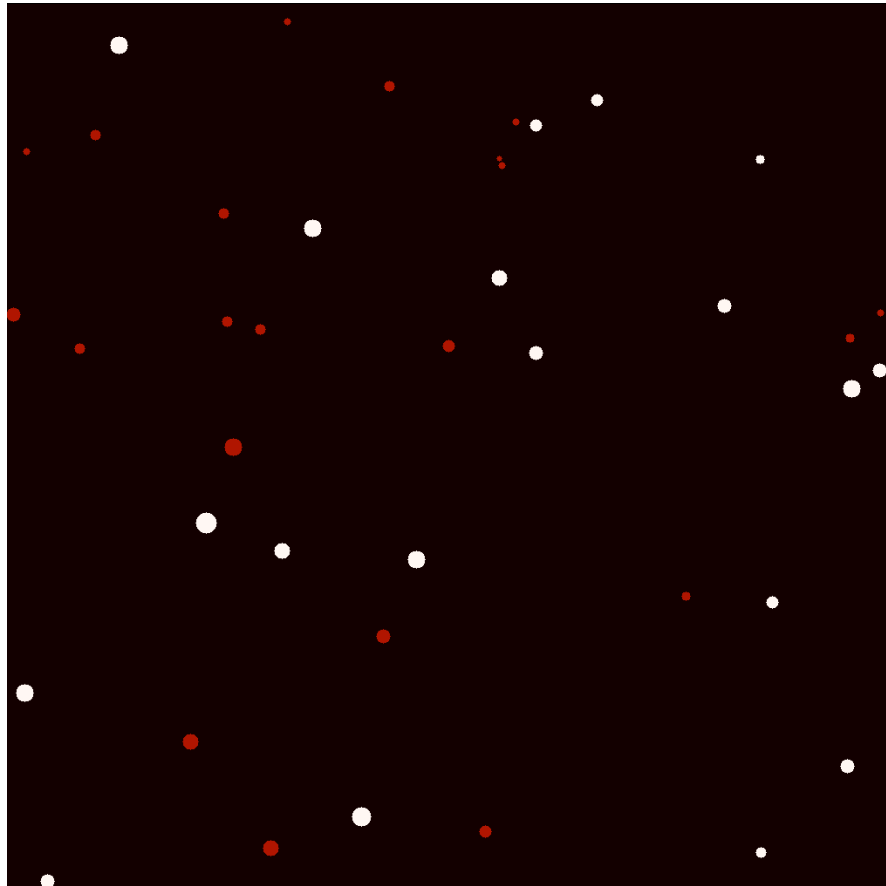
*Figure 8: Map of all unstable pixels identified as part of this bad pixel search.*

### *IR “Blobs”*

This update to the bad pixel table also afforded the opportunity to update the list of currently known “blobs”. These blobs are thought to be the result of particulates resting on the Channel Select Mechanism (CSM), resulting in out-of-focus blobs with decreased throughput at the detector. Details are given in Pirzkal et. al, 2010. Pirzkal’s ongoing monitoring program identifies new blobs as they appear in IR images. Pixels affected by

these blobs are then marked in the bad pixel table with the value 512. This value is typically used for pixels which are found to be “bad in flat”.

As a part of this update, the number of pixels flagged as affected by blobs has risen from 4,534 to 7,162, due to an increase in the number of blobs from 19 to 40. Figure 9 shows a map of the updated blob positions and diameters. Newly identified blobs (since the previous bad pixel table update) are shown in red.



**Figure 9:** New blob mask created for this bad pixel table update. White blobs were present in the previous version of the bad pixel table. Red blobs have been newly added.

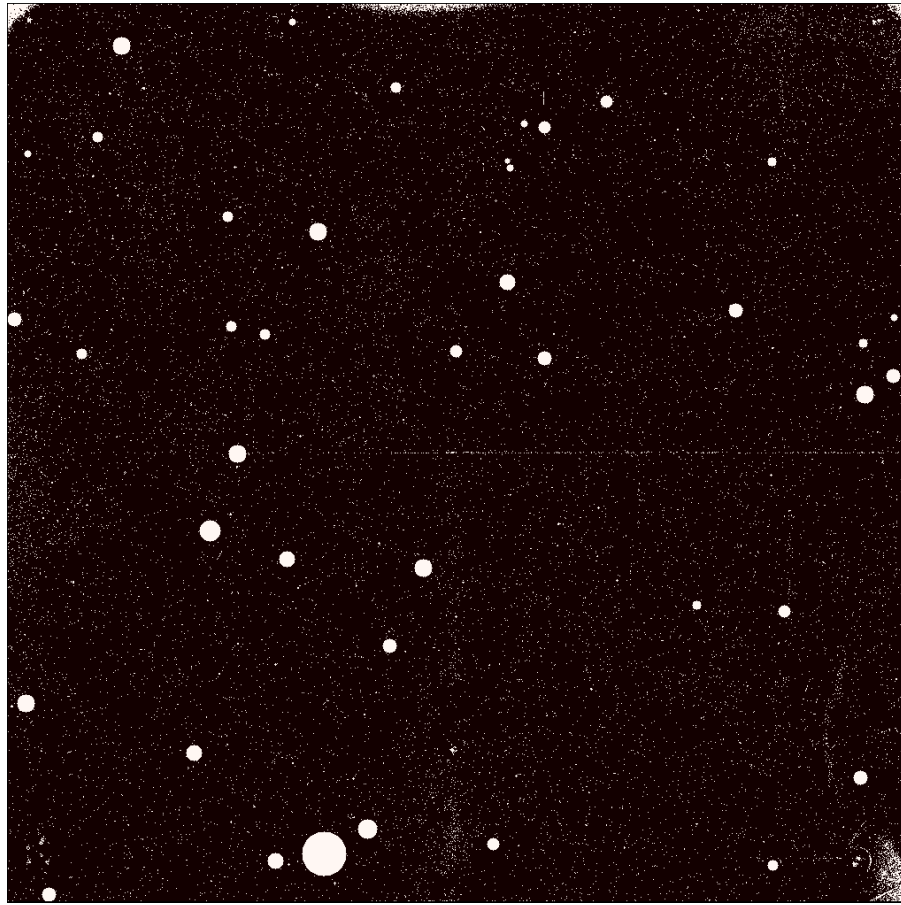
## Conclusions

Using bit-wise addition, the maps of the four types of bad pixels are combined into a final bad pixel map, shown in Figure 10 below. This map has a total of 28,500 pixels (2.77% of the light-sensitive pixels) flagged as bad. Added to this list of bad pixels are the outermost row and column of the reference pixels, which are flagged with a value of 128 in the bad pixel table, indicating “bad reference pixels”. Note that with the reference pixels stripped away, these flags are not present in the *flt* files output from *calwf3*. This final set of bad pixels is placed in a fits table for use by *calwf3*, where the flags are propagated into the data quality arrays of individual observations. Table 3 summarizes

the results of this analysis, listing the various flavors of bad pixels, the flag value associated with each, as well as the size of the population.

The current default behavior for *calfw3* is to process bad pixels in exactly the same way that good pixels are processed. Previous versions of *calwf3* forced bad pixels to have a value of zero in the output *flt* file, effectively forcing bad pixels to be unusable. This behavior has been changed, so that users must now decide which types of bad pixels they wish to ignore and which they wish to use. This can be done by reading in and examining the data quality array (in the third extension) of the output *flt* files, and ignoring pixels with the desired bad pixel flags.

With bad pixels spread widely across all areas of the detector, observers are encouraged to dither their observations, which will allow the effects of these bad pixels to be minimized.



**Figure 10:** Map of the location of all bad pixels found in this bad pixel table update. 28,500 pixels (2.77%) have been flagged.

Bad Pixel Type	Flag Value in Bad Pixel Table	Number of Pixels	Percentage of Light-Sensitive Pixels
Good Pixels	0	999696	97.23%
Dead	4	3358	0.33%
Bad in 0 <sup>th</sup> Read	8	4923	0.48%
Unstable	32	16500	1.60%
Bad Reference Pixels	128	4092	NA
Blobs (Bad in Flat)	512	7162	0.70%

**Table 3:** Results from this bad pixel search. 97.23% of the light-sensitive pixels are unflagged and considered good.

## Recommendations

WFC3/IR users are encouraged to dither their observations in order to minimize the impact of bad pixels. Also, during data analysis, careful attention should be paid to the data quality arrays in the data, so that inappropriate pixels are not used in analysis.

## References

Hilbert, B., and H. Bushouse, 2010. WFC3 ISR 2010-13. “*WFC3/IR Bad Pixel Table: Update Using Cycle 17 Data*”.

Hilbert, B., S. Baggett and M. Robberto, 2003. WFC3 ISR 2003-06. “*Masking Technique on WFC3 IR Images*.”

Pirzkal, N., A. Viana, and A. Rajan, 2010. WFC3 ISR 2010-06. “*The WFC3 IR ‘Blobs’*”.

Rajan, A. et al. 2010, “WFC3 Data Handbook”, Version 2.0, (Baltimore:STScI).

## Acknowledgements

Many thanks to Kai Noeske for helpful comments and Nor Pirzkal for supplying an updated list of blob positions and sizes.

## Appendix

List of ramps used in this analysis.

Proposal ID	Sample Sequence	Exposure Time (sec)	Filenames
<i>Dark Current Observations</i>			
11929	SPARS200	2803	ibcu5ydaq_raw.fits, ibcu6lq3q_raw.fits, ibcu6ydsq_raw.fits ibcu8ldzq_raw.fits, ibcub1tyq_raw.fits, ibcubecnq_raw.fits
	SPARS100	1403	ibcu6gc5q_raw.fits, ibcu6nd7q_raw.fits, ibcu6oz7q_raw.fits ibcu7ad0q_raw.fits, ibcu7byqq_raw.fits, ibcu7neuq_raw.fits ibcu7owmq_raw.fits, ibcu7tavq_raw.fits, ibcu8ne7q_raw.fits ibcu9bvqk_raw.fits, ibcu9a6a6q_raw.fits, ibcu9acyq_raw.fits ibcu9aeyq_raw.fits, ibcu9ajfq_raw.fits, ibcu9aqgfq_raw.fits ibcub9axq_raw.fits
12349	SPARS200	2803	iblt15c6q_raw.fits, iblt1bdcq_raw.fits, iblt2hfxq_raw.fits iblt2xfq_q_raw.fits, iblt31bkq_raw.fits, iblt3nbyq_raw.fits iblt4tetq_raw.fits, iblt5jf5q_raw.fits, iblt79bgq_raw.fits iblt8ldzq_raw.fits, iblt95btq_raw.fits, iblt9bf9q_raw.fits iblt9rbkq_raw.fits
	SPARS100	1403	iblt0akrq_raw.fits, iblt0idmq_raw.fits, iblt0pd1q_raw.fits iblt0qcrq_raw.fits, iblt19clq_raw.fits, iblt1fdeq_raw.fits iblt1gcqz_raw.fits, iblt1ngqq_raw.fits, iblt1og5q_raw.fits iblt1wsgq_raw.fits, iblt20cqq_raw.fits, iblt27crq_raw.fits iblt28f8q_raw.fits, iblt2dg8q_raw.fits, iblt2ehlq_raw.fits iblt2ldeq_raw.fits, iblt2mcqq_raw.fits, iblt35mbq_raw.fits iblt36jbq_raw.fits, iblt3jcaq_raw.fits, iblt3kclq_raw.fits iblt43jfq_raw.fits, iblt44ihq_raw.fits, iblt4hl2q_raw.fits iblt4ij4q_raw.fits, iblt4pfjq_raw.fits, iblt51j9q_raw.fits iblt59q6q_raw.fits, iblt5nldq_raw.fits, iblt5olnq_raw.fits iblt5wb6q_raw.fits, iblt60mlq_raw.fits, iblt67n9q_raw.fits iblt68kiq_raw.fits, iblt6dnfq_raw.fits, iblt6enhq_raw.fits iblt75dhq_raw.fits, iblt76d9q_raw.fits, iblt7bjtq_raw.fits iblt7cjoq_raw.fits, iblt7jj6q_raw.fits, iblt7kj2q_raw.fits iblt7rdnq_raw.fits, iblt8hk0q_raw.fits, iblt8iizq_raw.fits iblt8pb4q_raw.fits, iblt8qndq_raw.fits, iblt8xhwq_raw.fits iblt8yhqq_raw.fits, iblt91cj_q_raw.fits, iblt92c8q_raw.fits iblt99lpq_raw.fits, iblt9fe0q_raw.fits, iblt9ncdq_raw.fits iblt9obyq_raw.fits, iblt9vj5q_raw.fits, iblt9wiyq_raw.fits iblt94iyq_raw.fits, iblt9bf5q_raw.fits, iblt9achs_q_raw.fits iblt9aknmq_raw.fits
12695	SPARS200	2803	ibvi15feq_raw.fits, ibvi16d1q_raw.fits, ibvi25vdq_raw.fits ibvi34tmq_raw.fits, ibvi35trq_raw.fits, ibvi53cfq_raw.fits ibvi54ciq_raw.fits, ibvi72a3q_raw.fits, ibvi91caq_raw.fits ibvi92deq_raw.fits
	SPARS100	1403	ibvi08exq_raw.fits, ibvi18iiq_raw.fits, ibvi27m9q_raw.fits ibvi28m7q_raw.fits, ibvi37z5q_raw.fits, ibvi46f8q_raw.fits ibvi47dkq_raw.fits, ibvi66m2q_raw.fits, ibvi75iqq_raw.fits ibvi84f4q_raw.fits, ibvi85e0q_raw.fits

<i>F140W Internal Flat Field Observations</i>			
11915	RAPID	38.1	ib9m0anoq_flt.fits, ib9m0erwq_flt.fits, ib9m0fhoq_flt.fits ib9m85d3q_flt.fits, ib9m89t9q_flt.fits, ib9m90tgq_flt.fits ib9m95arq_flt.fits
12338	RAPID	38.1	ibm901gqq_flt.fits, ibm902x6q_flt.fits, ibm903nxq_flt.fits ibm904drq_flt.fits, ibm905nnq_flt.fits, ibm906lvq_flt.fits ibm907jrj_flt.fits, ibm908nfq_flt.fits, ibm909n9q_flt.fits ibm910heq_flt.fits, ibm912v7q_flt.fits, ibm913fzq_flt.fits ibm919i1q_flt.fits, ibm920xfq_flt.fits, ibm921leq_flt.fits ibm922exq_flt.fits, ibm923nyq_flt.fits, ibm924m6q_flt.fits ibm925c5q_flt.fits, ibm926mzq_flt.fits, ibm927mdq_flt.fits ibm928gwq_flt.fits, ibm930z6q_flt.fits, ibm931fhq_flt.fits ibm937irq_flt.fits, ibm938y5q_flt.fits, ibm939lmq_flt.fits ibm940k0q_flt.fits, ibm941nfq_flt.fits, ibm942lnq_flt.fits ibm943j2q_flt.fits, ibm944k0q_flt.fits, ibm945mtq_flt.fits ibm946h4q_flt.fits, ibm948zuq_flt.fits, ibm949fnq_flt.fits
12712	RAPID	38.1	ibvf01zpq_flt.fits, ibvf02s6q_flt.fits, ibvf03uiq_flt.fits ibvf07eeq_flt.fits, ibvf13zxq_flt.fits, ibvf15voq_flt.fits ibvf19erq_flt.fits, ibvf25zcq_flt.fits, ibvf26ssq_flt.fits ibvf27teq_flt.fits
	SPARS10	32.9	ibvf31cpq_flt.fits, ibvf32g2q_flt.fits, ibvf33gvq_flt.fits