

Improving the Pixel-Based CTE-correction Model for ACS/WFC

Jay Anderson and Jenna Ryon
August 29, 2018

ABSTRACT

The pixel-based CTE correction was last constructed for ACS/WFC in 2010 and for WFC3/UVIS in 2013. Each of these instruments has now been in orbit for about twice as long as when the model was last constructed. Since the strength of CTE generally increases linearly with time, similar to the population of warm pixels, it makes sense to revisit the correction now that the effect is twice as strong and there exist twice as many pixels to measure it with. This ISR will demonstrate how we constructed a new model for ACS/WFC and will evaluate the model with on-sky data.

1. Introduction

HST's vantage point above the atmosphere allows us to see crystal-clear images of the sky. But at the same time, being above the atmosphere makes the telescope's detectors much more vulnerable to cosmic-ray damage than detectors on the ground. Hot pixels or warm pixels (WPs) are one by-product of this degradation¹. Charge-transfer-efficiency losses (CTE losses) are another.

Imperfect CTE means that the electrons originally generated in the pixels of the detector are not all faithfully transferred to the readout electronics as a single charge packet. During the transfer process, traps in the silicon lattice can delay the arrival of individual electrons at the readout amplifier and thus cause them to be associated with pixels that are upstream from their original

¹ Many studies make a distinction between hot pixels and warm pixels. We will make no distinction here, but will refer to them all as "warm pixels" since the median pixel with extra dark current does not have a lot.

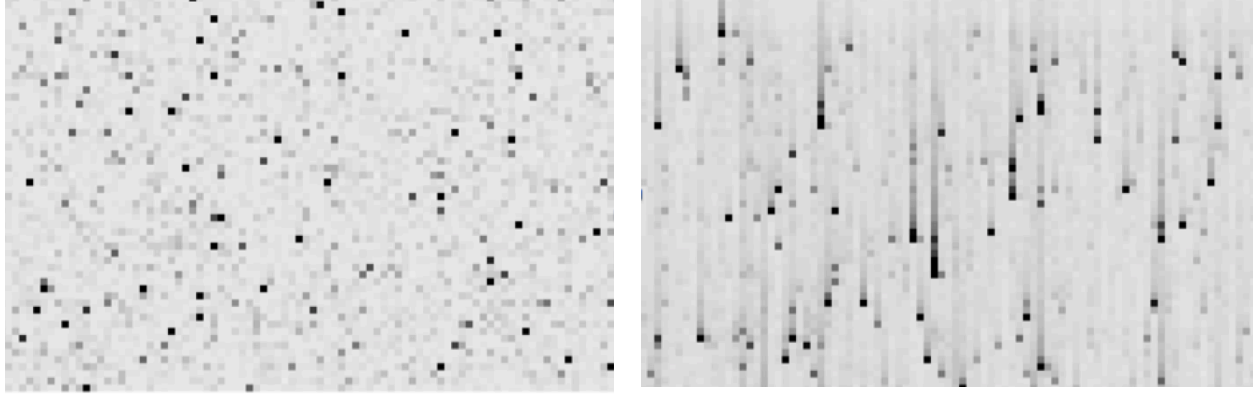


Figure 1: Left: A cutout from an unflashed long dark exposure very close to the readout amplifier where very little CTE losses are present. Right: the same image, but far from the amplifier (the top few pixels shown are overscan). Readout is in the downward direction.

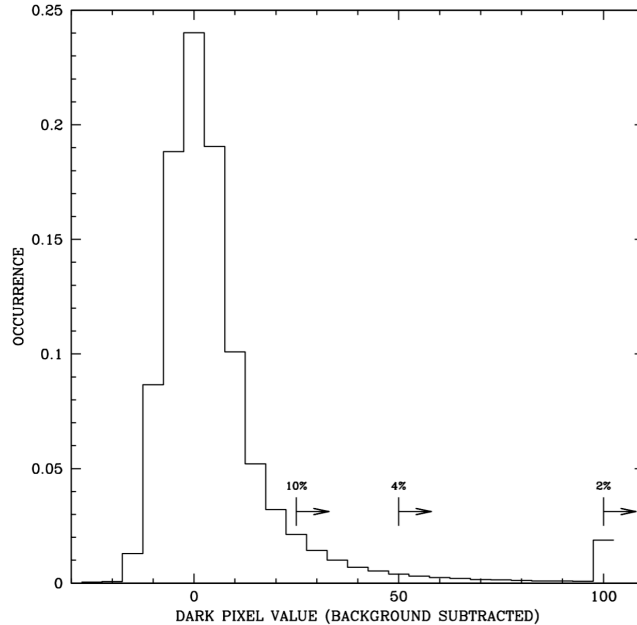


Figure 2: Distribution of pixel values in a 1001-s post-flashed dark near the readout register (left panel of Figure 1), scaled to represent the fraction of pixels at each intensity level.

location. The number of traps — and hence, the severity of CTE losses — increases monotonically with time in the high-radiation environment of space.

Figure 1 shows the current impact of CTE on ACS images. The left panel shows what a long dark looks like at the bottom of the detector, where the small number of transfers means that transfer losses are minimal. The right panel shows the same dark exposure at the top of the detector, where pixels suffer about 2000 imperfect transfers.

The right panel *should* look just like the left panel qualitatively, since all the pixels have been exposed equally to radiation and should have the same distribution of hot pixels. Of course, the pixels on the detector do not know how far they are from the readout register. But the readout causes significant blurring: the bright pixels all have significant trails and many of the fainter WPs have been trailed away into obscurity.

Figure 2 shows a histogram of the left, CTE-free panel of **Figure 1**. About 10% of the pixels in this 1001-s exposure have increased dark current of 25 electrons or more (above the average). This is about 6 times the readnoise. About 2% of the pixels have WPs with more than 100 electrons. This may seem like a small fraction of pixels, but even this small fraction makes it difficult to study the CTE trails in isolation, in a 10-pixel trail behind a bright WP we would expect at least one 25-electron WP to interfere with our measurement of the trail.

In 2010, a model was developed by Anderson & Bedin (2010; AB10) to describe the charge-transfer process for WFC/ACS. It was a generalization of a specialized model constructed by Massey et al (2010) to reduce the GOODS field data. The success of AB10's general model led to a quick implementation in the pipeline. WFC/ACS had been in orbit for 8 years by that time, so CTE was starting to have a significant impact on science.

When WFC3/UVIS was launched in May 2009, it was hoped that it would have several largely CTE-free years. However, thanks to its combination of low dark current, high sensitivity in the UV (where the background is low), and complement of narrow-band filters, WFC3/UVIS images tended to have very low backgrounds and ended up suffering from CTE much more quickly than ACS images. For this reason, we developed a model for WFC3/UVIS in 2013, and that model was quickly incorporated into the pipeline. The WFC3/UVIS model differed a bit from the original ACS model. Since for WFC3/UVIS, the model needed to work at low backgrounds, special attention was taken to constrain the model at the small-packet end as well as to avoid readnoise amplification when the background was low.

Over the last few years, the pixel-based CTE correction has become a very important pipeline product. The correction has been working quite well, as evidenced by yearly external tests (Chiaberge et al. 2012, and numerous CAL programs, such as CAL-14949 and CAL-11510). But even with continued good performance, it is worth revisiting the model to see whether it can be improved. Since ACS has been in orbit for twice as long as it had in 2010, we now have 4 times as much data to constrain the model (the CTE-loss signal is $2\times$ larger and the population of WPs is $2\times$ larger). In general, this allows better constraints, but the sheer number of WPs currently present in the CCDs also introduces complications.

In this document, we will go through the steps we took to re-derive the parameters of the ACS model. We will also perform some on-sky demonstrations so that users can know when to trust the correction products and how much.

This ISR is organized as follows. In **Section 2** describes the basic CTE model parameters and algorithm. In **Section 3** describe how to remove bias-related artifacts so that the algorithm can operate on real electrons. **Section 4** describes the data sets we used to pin-down the model parameters. **Section 5** evaluates the current corrections and shows aspects where it could be improved.

The new model is constructed in stages. **Section 6** re-pins the model for the largest charge packets, and **Section 7** focuses on the intermediate-sized packets. **Section 8** treats the smallest packets, where the background can have a significant impact on which traps are filled and which are left open to impact sources. **Section 9** explores the trail shape in detail. Finally, **Section 10** provides some on-sky evaluation of the model, and **Section 11** describes the pipeline implementation.

2. The Basic CTE Model

The basic CTE model is pretty simple. The “forward” model takes an image as it would be recorded perfectly on the detector and shows what it would look like after getting pushed through the CTE-blurring readout process. The “reverse” model does the opposite: it takes an observed image that has been pushed through the readout process and tries to determine the most likely input image that could have generated it. We use an iterative forward-modeling procedure to arrive at the reverse model. This is described in detail in AB10.

The basic aspect of the model is the bucket-brigade transfer of flux down a column. If a pixel contains a trap, and a charge packet that is large enough to be affected by the trap gets shuffled into that pixel, then the trap grabs one of the electrons and the packet gets shuffled down into the next pixel without that electron. That electron is eventually released in an upstream pixel according to the trap-release profile. So the traps in the model serve as “sinks” and “sources”.

2.1 How to model the traps?

It is very hard to know where the traps are. There does exist an engineering procedure called “pocket-pumping” that can locate traps (see Mostek et al 2010 for a detailed description). Briefly, the detector is exposed to a small amount of charge, and this charge is shuffled around in x and y in a deliberate way so that it ends up getting stuck in the traps. When the detector gets read out, the resulting image reveals which pixels contain traps.

The ACS/WFC electronics are not set up to allow this sophisticated experimentation, but even if we had access to such a procedure, knowing which pixels contained how many traps would not really help us model things much better, since not only do we need to know which pixels contain traps, but we would also need to know which traps in each pixel affect which electrons in charge packets.

The reason for this is that not only are traps localized within specific pixels, but they are also located at different places within each pixel. Depending on where they are within the pixel, traps can affect charge packets of different sizes. A trap near the center of a pixel might affect all charge packets, but one near the edge of the pixel might affect only the largest electron packets, which occupy a larger volume within the pixel. Furthermore, new traps appear regularly, so we would need to specify for each trap which pixel it is in, where it is within the pixel (in terms of what size of charge packets it will affect), and when it first appeared. In other words, for each trap we would need to specify four quantities $[i, j, q, t]$.

It is clearly impossible to come up with a CTE model that deals with a detector in all of this complexity. Thankfully, this is not necessary. It appears that even though traps may be clustered in specific pixels (as the pocket-pumping papers suggest), they are distributed among a very large number of pixels. A study by Ogaz et al (2013) examined the overscan pixels in flashed images and showed that the variation in the number of traps in a given column is typically less than 10%. This means that even though traps may not be distributed perfectly evenly among pixels, they are not clustered to the degree that their specific distribution has a strong impact on the total amount of CTE suffered by a source trailed down in a specific column. For this reason, it makes sense to adopt a CTE model that treats all pixels as having the same distribution of traps.

Now, on average, each pixel does not have a large number of traps, so our “even distribution” of traps will not deal with whole traps in each pixel, rather we will model each pixel as having an array of fractional traps. Each pixel will contain, say, a trap that affects the 100th electron in a packet, and that trap will grab 0.001 electron each time a packet with 100 or more electrons is shuffled through. Each pixel will also contain many more traps that affect a variety of other marginal electrons. In this way, we deal with the trapping and release in a purely homogeneous way.

2.2 Parameterizing the model: $\phi(q)$ and $\tau(\Delta j)$

The model has two basic sets of parameters to deal with the trapping and the release of charge. The first set of parameters tells us how many traps affect electron packets of various sizes. We parameterize this with a simple one-dimensional cumulative function $\phi(q)$ that gives the total number of traps that affect packet sizes q or greater (where q is the number of electrons in the pixel packet) within a 2048-pixel column.

The function $\phi(q)$ is specified by its value at 16 fiducial points: $q = \{1, 2, 3, 5, 7, 12, 20, 30, 50, 70, 100, 300, 1000, 3000, 10000, 30000, 99999\}$. (Although full-well for ACS is about $\sim 80,000$, our model goes up to 99,999 to preserve the logarithmic spacing.) The model uses log interpolation to turn this cumulative function into a list of traps, each of which affects a particular electron in a pixel packet (i.e., the 3rd electron or the 31st electron). This list of traps corresponds to the total number of traps within a 2048-pixel column, so each individual pixel has traps, and each one grabs about 0.000488 electrons as packets are transferred into and out of that pixel.

The other aspect of the model is the release of charge. The trapped charge that is retained in one pixel is eventually released into an upstream pixel. The model contains an additional set of parameters that regulate the probability that an electron that is trapped will be released after 1, 2, 3, etc., shifts. The function $\tau(\Delta j)$ specifies the fractional release probability as a function of pixel shifts. In practice we specify $\tau(\Delta j)$ at a number of Δj values and interpolate it linearly in between.

The function τ can also be a function of q , $\tau = \tau(\Delta j; q)$. This would make sense if the probability of trap release is not just a simple time constant but rather is some function of the local intensity of the oscillating electric current involved in shifting the charge down the detector. It could be that the traps closer to the center of the potential well of the pixel (where the smaller electron packets spend their time) could experience stronger (or weaker) forcing.

In summary, the parameters of the model are $\phi(q)$ and $\tau(\Delta j; q)$, where $\phi(q)$ corresponds to the number of traps affecting the q^{th} charge and $\tau(\Delta j; q)$ describes the release profile of the trap that grabs the q^{th} charge. To simulate the detector readout, we turn $\phi(q)$ into a discrete list of traps, q_1 through q_N , where N is the total number of traps that affect a full packet. Each of these traps then has a release profile, $\tau(\Delta j; q)$.

2.3 The actual algorithm

The readout algorithm is described in detail in AB10, but we will summarize it here. The algorithm deals with one trap at a time. Each trap affects a particular marginal electron in the

packet (say the 44th). The algorithm shuffles the charge down the column through the pixel with the trap. When the trap-containing pixel encounters a packet that has at least 44 electrons, it removes an electron from the packet and resets the trap to full ($\Delta j = 0$). It then shuffles the charge down to the next pixel in the column. If the next incoming charge packet has fewer than 44 electrons, the trap releases a fraction of the electron into that charge packet— $\tau(1; 44)$ — then shuffles it out and shuffles in another charge packet. If that pixel also has fewer than 44 electrons, it releases $\tau(2; 44)$, about 10%, and so on. If it encounters a pixel that has 44 electrons or more, then it determines what fraction of the trap is still empty and fills it, then resets the trap to $\Delta j = 0$.

As described in AB10, in practice, it would take a very large number of floating-point operations to model thousands of traps affecting each of the 2048×1024 pixel-to-pixel transfers that take place². This is more than 2 billion calculations for each column. And that does not even include modeling of the charge release, which could take 100× more calculations!

For this reason we make use of the fact that all the pixels are the same. Shuffling the charge through 100 pixels is the same thing as shuffling it through one pixel with traps that grab 100× more charge. As such, a pixel at $j=1000$ experiences traps with 1000× the strength of those at $j=1$. Since the model parameters correspond to the amount of charge in the entire column (i.e., $j=2048$), a pixel in row j experiences traps that are $j/2048$ times the nominal intensity.

A slight complication is that the distribution of charge in the column can change somewhat as the charge is shuffled down the detector, so a packet with 100 electrons at the top of the detector may have only 75 at the bottom, and hence would be affected by fewer traps during a portion of its journey. For this reason, we perform the shuffle in 7 stages, shuffling a seventh of the way each stage. (We found through experimentation that 7 shuffles were adequate to account for the impact of the change of the pixel distribution during the charge-transfer process.)

We deal with time-scaling in a similar way. The new model will be pinned to describe CTE losses as of November 12, 2016, the date the calibration data were taken. For dates before or after this time, the model should be scaled down or up, and the easiest way to do this is to just change the “size” of each trap. This scaling factor is simply $f = (T_{\text{OBS}} - T_{\text{INSTALL}})/(T_{\text{PIN}} - T_{\text{INSTALL}})$.

The code that was used to determine the optimal parameters was developed in FORTRAN. This code was then translated into C and is implemented in CALACS. The algorithm described above has not changed significantly from the original implementation in AB10, however some of the loops have been reorganized to speed-up performance and to conform to good coding practices. For all these reasons, rather than tweak the old model parameters, we will re-derive the parameterization completely.

2.4 Setting the Model Parameters

As mentioned above, there are a large number of traps in each column, and it would be impossible to specify each of them precisely, so the model parameterizes this. We specify the cumulative function $\Phi(q)$ at 16 representative values of q and use log-interpolation to turn this

² The bottom pixel goes through 1 transfer to get to the serial readout register, the top pixel goes through 2048 transfers, so the average pixel goes through 1024 transfers, hence 2048×1024 total pixel-to-pixel transfers.

into a list of traps, each of which affects a particular electron from $q=1$ to $q=99999$. (Saturation kicks in at around 80,000 e^- , so traps above this do not come into play).

Similarly, the release profile is not specified for every Δj shift and q level, but we specify it at $\Delta j = 1, 2, 3, 5, 8, 12, 16, 20, 25, 30, 40, 50, 60, 70, 80$, and 99, and we do this for $q=1, q=10, q=100$ and $q=1000$, to allow the profile to change with packet size. To get a value of $\tau(\Delta j; q)$ for every value of Δj for trap level q , we interpolate this two dimensional parameter space, linearly along Δj and in log along q . The reference files simply contain a list of N charge levels where the individual traps are, and a 2-D image that records the probability that the N^{th} trap will release its charge after Δj shifts.

The PCTETAB reference file contains the complete interpolated model for the pixel-based correction. This is what CALACS uses to perform the pixel-based CTE correction. The first extension of the PCTETAB file provides $\Phi(q)$, which is a table that lists up to 9999 traps. For each trap it specifies which marginal electron (q) that the trap affects, along with the number of electrons removed by the trap (for the ACS model, this is always 1.0). The second extension allows for a column-by-column scaling of the correction, which is currently set to 1.0 for all columns (Ogaz & Anderson 2013 showed that the column-to-column variation is not very significant.). The third extension of the PCTETAB file contains a 9999×100 element array, each column of which specifies the release probabilities for the N^{th} trap after Δj shifts. Finally the fourth extension is similar to the third, but it contains the fraction of the trap that is still full after Δj shifts.

3. Separating the electrons from the electronics

Now that we are ready to study the observed electrons in images, it is important to know which counts in the recorded image correspond to real electrons that went through the charge-shuffling process and which counts may have other origins. There are two issues to deal with here. The first is the bias (and possible bias gradients) and the second is the readnoise.

3.1 Bias Gradient

Since the A-to-D converter can record only positive data numbers (DNs), there is a moderate bias applied to the output of the amplifier to ensure that the recorded voltage is always positive. This typically translates into 4000 or so electrons for ACS.

In addition to the baseline voltage, there are other electronic signatures that impact the images. These are often caused by a feedback-type hysteresis in the readout electronics, and they manifest themselves as horizontal gradients in the images (see Golimowski 2012). Both ACS/WFC and WFC3/UVIS have sharp gradients within the pre-scan pixels, which turn into gentle gradients across the detector. [Figure 3](#) shows the horizontal bias gradients for ACS/WFC.

ACS BIAS COLUMN AVERAGES 2009–2016

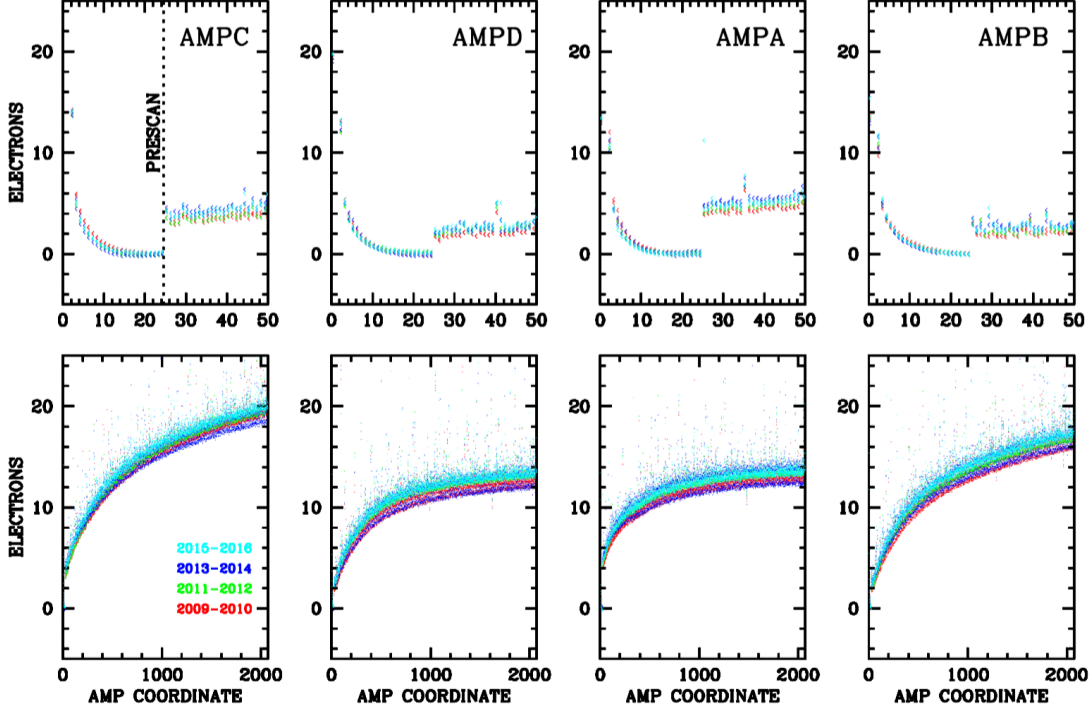


Figure 3: The horizontal bias gradients present in post-SM4 ACS from 2009 to 2016. The different colors refer to different years (see legend in bottom left panel). The baseline voltage for each amplifier has been subtracted. Top row: a close up around the pre-scan pixels for each amp. Bottom: the full amp trend with serial-register coordinate (x). Before SM4, the gradient had a different shape.

The bottom panels show the entire gradient across the detector for the four amplifiers, and the top panels show a close-up around the pre-scan pixels. The jump in level between the virtual prescan and the real image pixels at $j=25$ in the top panels of **Figure 3** represents real electrons. Note that these jumps are slightly larger for later epochs. The prescan pixels are not shifted down the detector into the readout register: they simply represent the null, unfilled value of the readout register. The image pixels, on the other hand, *are* shifted down the detector and in this process do pick up dark current.

Ryon et al. (2017) report on this “readout” dark current, which amounts to about 4 electrons for a pixel shifted from the top of the detector to the readout-register at the bottom. Treating this as simply part of the bias works for bias subtraction (since all images that are read-out have it), but this dark current does technically represent real electrons that are present during at least part of the shift time, so to the extent that a few electrons can make a difference in the CTE impact, we should include them in our analysis.

Reality is actually even a bit more complicated than that. These electrons are not present when the charge first starts shifting. They get added slowly as the charge moves down the detector. So, the CTE algorithm should technically start out with these electrons subtracted, then it should slowly add them in during the simulated journey. We decided that since ACS images typically have decent backgrounds (much higher than this), and since it turns out that the ACS CTE model

is not pathologically sensitive to one or two electrons, we wouldn't concern ourselves with these issues here. We will simply subtract the super-bias and run the algorithm on the residual electrons.

An additional issue is that the new post-SM4 electronics of ACS ended up adding some $1/f$ noise into the baseline voltage, and a correction for this (involving a row-based adjustment to the baseline bias) largely removes the “bias striping” present in post-SM4 ACS/WFC images. Grogin et al (2011) describe this effect and present ways to combine the prescan pixels of all the amps (which all experience the same noise) to come up with a correction for the effect to better than 0.5 electron (see Section 3.4.1 of the ACS Data Handbook for more details).

3.2 Avoiding Readnoise amplification

In addition to ensuring that bias-related issues are not involved in the CTE reconstruction, we also need to ensure that our correction is robust against readnoise. The “electrons” that come from readnoise are not real electrons at all. They are simply voltage fluctuations that impact the DN measurement when the pixel value is read out. These “electrons” never went through the charge-transfer process and as such, the algorithm should not operate on them.

This introduces a complication. Of course we cannot remove the readnoise—manufacturers have been trying to do that for decades! But we should do our best to minimize its impact on our correction. Given that readnoise is incoherent from pixel to pixel, it can cause particularly sharp artifacts if it is present during the de-blurring process. All we can do is minimize the damage it can do to the pixel-based reconstruction.

Our goal, therefore, will be to attribute as much of the pixel-to-pixel variation in the image as possible to readnoise, so that the impact of readnoise on the reconstruction will be minimized. This involves finding the smoothest possible image that is consistent with being the observation, after a pure readnoise image is added.

We will decompose the 4096×4096 observed image \mathbf{I} into two images, $\mathbf{I} = \mathbf{S} + \mathbf{R}$, where image \mathbf{R} is consistent with being pure readnoise and image \mathbf{S} is as smooth as possible. We know that for ACS/WFC image \mathbf{R} should have an RMS of about 4.25 with a Gaussian distribution, and it should also be as close as possible to random, not having any spatial structure in it.

We note that the algorithm used to exclude the readnoise from the CTE-correction algorithm that was used initially for ACS (see Anderson & Bedin 2010) is an early version of what we present here. When that algorithm was applied to WFC3/UVIS data, the correction turned out much noisier than expected. Although the readnoise in WFC3/UVIS is lower than that in ACS ($\sim 3 \text{ e}^-$ as opposed to 4.25 e^-), thanks to its lower dark current, UV sensitivity and complement of narrow-band filters, WFC3/UVIS often takes science exposures with extremely low backgrounds, which are much more frequently read-noise limited. We developed an improved version of the correction for WFC3/UVIS and now use this improved version on ACS. In [Appendix B](#), we describe the procedure whereby we decompose image \mathbf{I} into its \mathbf{S} and \mathbf{R} components.

4. The dataset and the images we worked with

The original AB10 model was constructed by modeling the trails behind the WPs in standard dark exposures, which at the time were all 1040s and unflashed. This approach allowed an exquisite solution for the CTE losses for charge packets where the losses were small relative to the charge in the WP. But when losses were more than just a perturbation, it was hard to determine the original WP from the observable trail. This occurred for charge packets of about 100 electrons.

The fact that we could not assess CTE losses for small charge packets introduced two complications. The first is that faint stars could not be as accurately corrected, since the model was not accurate at the small-packet end. The second is that without a model for how many traps affect various small packets, it was impossible to correctly infer how much a certain background sky level would mitigate CTE losses. (When there is a background in an image, that background keeps the lowest-level traps filled, so that electrons above the background do not have to deal with the plentiful low-level traps.)

When we began to pin the CTE model for WFC3/UVIS in 2011, it was clear that we needed more information at the small-packet end than the long darks could provide. For WFC3/UVIS, we took a series of short dark exposures with various amounts of post-flash added³. We could scale down the long darks to infer how many counts each WP in the short darks should have started with, then we could examine in the short darks how much of this charge survived as a function of the background level. This is what enabled us to show that WFC3/UVIS has a mini channel that provides extra CTE loss protection when the background is above about 12 electrons.

We will use some of these same tricks for ACS. Whereas for AB10, we had only one flavor of long darks to work with, we now have several sets of data we can work with. The list of images used for this study are given in [Appendix A](#).

Long darks. The regular dark monitor programs probably still contain the most useful information for exploring charge-transfer issues. Several darks are taken every few days with the ACS CCD Daily Monitor calibration program. These darks are post-flashed in an effort to minimize losses for the faint WPs, so that the dark current can be subtracted effectively from science images (which typically have moderate backgrounds). There are also some regular monitoring observations that take darks with no post-flash. These unflashed darks are useful to help us understand the total losses as a function of WP packet size.

Post-flashed short darks. In addition to these standard deep calibration products, we also took some internal observations for the express purpose of helping us pin-down the model. ACS/CAL-14865 took a series of short dark exposures (25 s exposure time) with a variety of backgrounds so that we could evaluate WP losses as a function of background level.

The average date of the ACS/CAL program 14865 for the short darks was Nov 15, 2016, corresponding to 2016.923 in fractional years or 57707 in MJD. This will correspond to the effective model re-pinning date.

³ Post-flash was in fact a capability that was enabled for both ACS/WFC and WFC3/UVIS once we realized how critical it was to have a minimum background.

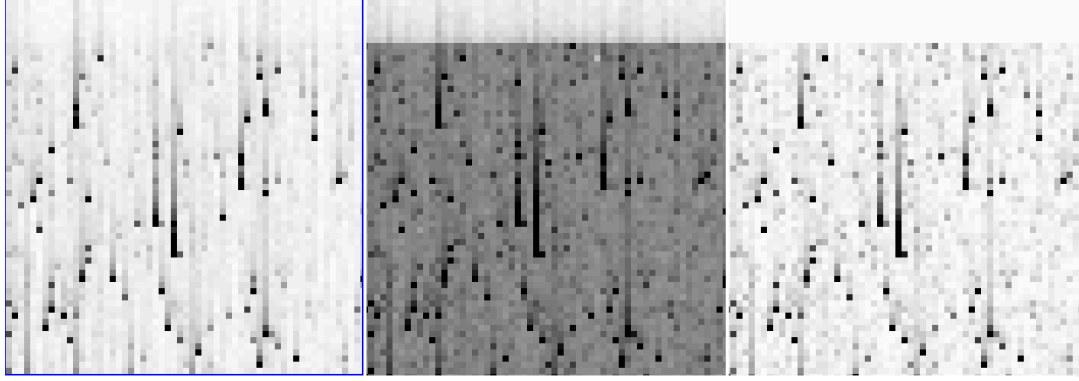


Figure 4: Stacks of long darks. Left: no postflash. Middle: with postflash (same image scaling). Right: with postflash, but background-subtracted.

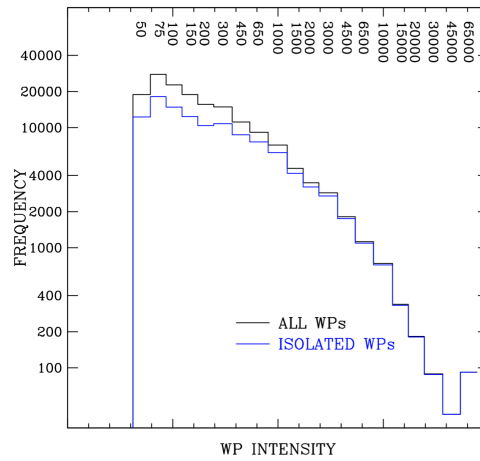


Figure 5: Distribution of WP intensities. The “isolated” WP list is what we took as our official list. It contains just over 100,000 WPs.

We collected the flashed and unflashed dark exposures taken close in time to this date (and all within the same anneal period) and averaged them to remove the impact of CRs. There were 12 flashed darks with exposure times of 1001s and 16 unflashed darks with exposure times of 1040s. **Figure 4** shows a portion of the unflashed and flashed long-dark at the top of the detector, where CTE blurring is greatest. By comparing the left and right panels, it is clear that more structure in the image is preserved when the background is higher.

To get our master list of WPs, we analyzed the set of post-flashed 1001-s darks. We identified a WP to be any pixel that was a local maximum (brighter than its 8 neighboring pixels) in more than 10 out of 12 exposures. This gave us a total of just over 100,000 WPs to work with, with 1000-s intensities ranging from 50 electrons to over 50,000 electrons. We further purged this list by removing any WP that had another nearby WP (± 1 in x , ± 5 in y) with more than 25% of its intensity, so that we could focus on the relatively isolated WPs. **Figure 5** shows the distribution of WPs.

External data. In order to provide an independent assessment of the quality of our correction, we took some external data of a field in the bulge to assess how well the reconstruction algorithm works in various real-world environments. This will be presented in [Section 10](#).

5. Evaluating the Current Pipeline

The ACS CTE model has been running in the pipeline for almost 8 years now. It is worth examining how well it does before trying to improve it. To evaluate the old correction, we simply ran the current pipeline on a few observations.

5.1 The current trails

AB10 showed that the easiest way to directly measure CTE losses is to identify WPs in the darks and examine the trails behind them. In principle, the WPs are just single high pixels, but because of imperfections in charge transfer, they get trailed up the detector during the readout process. The amount of charge in the trail corresponds to the amount of charge lost to the WP. The initial AB10 model was simply an effort to find the readout model that reproduced the observed trails. It was based solely on the existing unflashed long darks that had been taken as a part of the regular calibration programs.

Figure 6 on the next page shows the efficacy of the current correction for the flashed and unflashed long darks in the ACS/WFC calibration program. This plot shows the average trails behind various WPs far from the readout register ($1500 < j < 1900$) separated into six intensity bins (100 e^- , 300 e^- , 1000 e^- , 3000 e^- , 10000 e^- , and 30000 e^-). For each panel, we identified all qualifying WPs. There were 1818 in the first panel, over 10,000 in the second panel, etc. We then determined a robust average of the two upstream pixels (shown at $\Delta j = -2$ and $\Delta j = -1$ on the x-axis) and the downstream pixels out to $\Delta j = 20$. The black points show the values of the trails for the uncorrected image. The green shows the profile for the corrected image with no background and the blue for a corrected image with a background of 70 electrons.

Overall, the existing correction appears to be pretty good, but there does appear to be an over-correction of between 5 and 15 percent in most panels. This can be seen as the area between the blue and green curves and the baseline. The two pixels to the left of trail node 0 give us a sense of what the “background” before the WP looks like. The current pipeline correction is *worse* when there is background, indicating that the model is not well specified at the background flux levels.

Note that even though **Figure 5** shows that there should be many more WPs in the 100-electron bin than in the other bins, here we find only about 20% as many in the 100- e^- bin as in the 300- e^- bin. This is because far from the readout, many of these WPs have been blurred so much that there is no longer a head-tail structure to study. The residuals of observed 100 e^- WP trails are also hard to interpret, since it looks like these events can only be seen when the background is higher than usual (note the level of the two preceding pixels). We are clearly not looking at a representative set of trails in this bin. This is why we must take short exposures in order to better study the fainter WPs (and hence the CTE losses for small pixel packets).

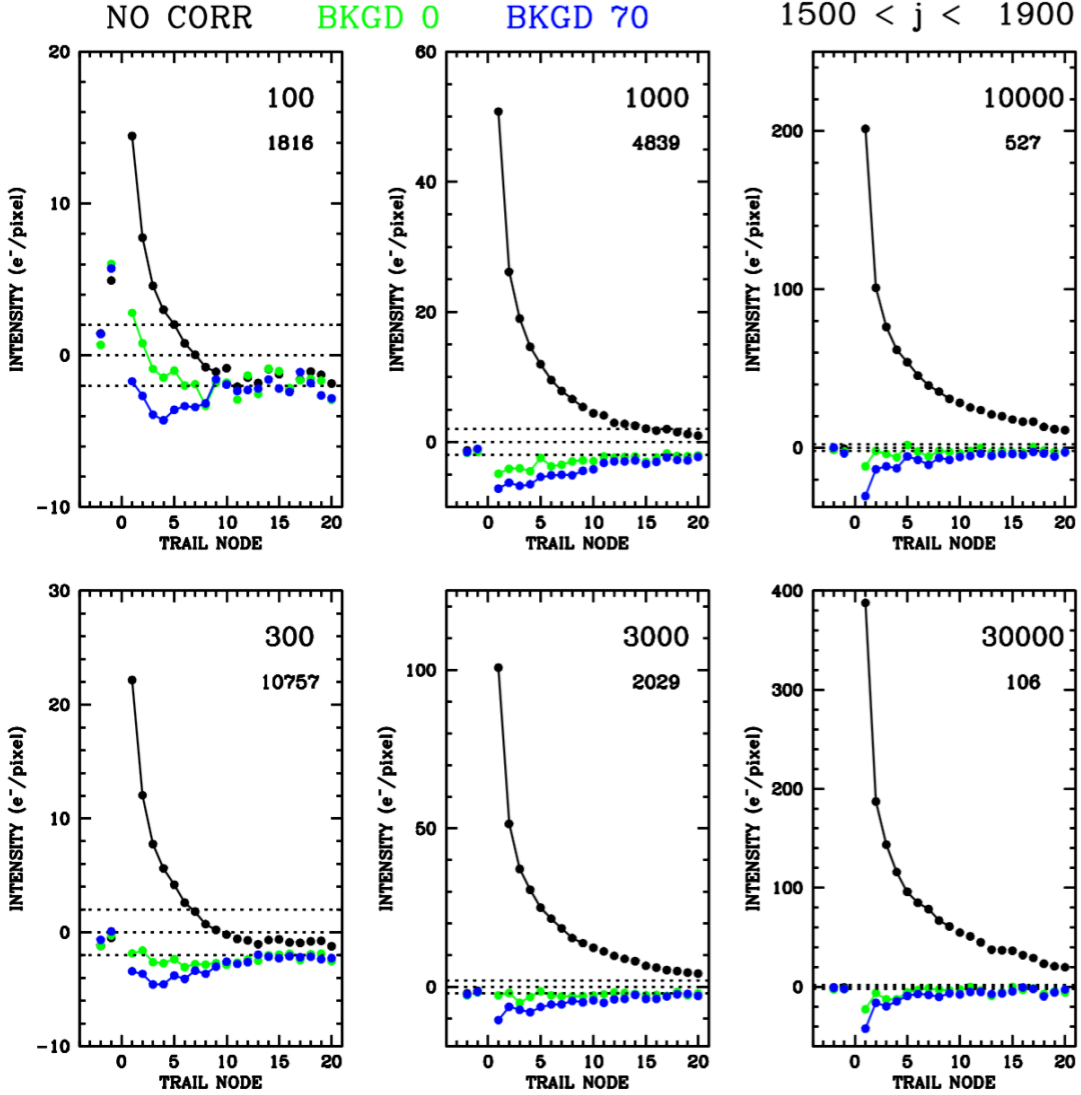


Figure 6: Each panel shows a WP for a different range of intensity levels, labeled in the upper right in large numbers. The small numbers show how many trails were averaged in this WP intensity bin between $j=1500$ and $j=1900$, far from the readout register. The black curve shows the uncorrected long-dark images. The green shows the AB10-corrected *unflashed* long darks, and the blue shows the AB10-corrected *flashed* long darks.

Note that dotted horizontal lines are plotted at a trail residual of 0 and $\pm 2 e^-$. It is worth noting that the baseline detector level in these curves is closer to -2 than it is to zero. To understand this, we must look in detail at how the trails are computed.

5.2 Complications measuring the trails

In the original AB10 effort, we computed trails by determining a baseline (essentially, a sky value) for each pixel in the trail by determining a robust average of the 5 pixels to the left and 5 pixels to the right of each one (see Figure 4 in AB10). We began by using the same technique for the current study, but we quickly found that the trail had a systematic bias of -5 electrons or

so relative to the “sky” measurement. We were able to trace this down to the asymmetric contribution of dark current to the background. [Figure 2](#) shows that there are more pixels affected by the dark current in a positive way than in a negative way, since WPs are positive excursions but not negative (there are not as many sink pixels as WPs, see Ryon & Grogan 2017). This introduces a *positive* bias to the sky average. This is all the more the case at the top of the detector where WPs affect many more pixels, thanks to their long trails. The net effect of this is to give any average of the background a bias: the average value is larger than the most likely value, even after harsh sigma clipping.

To alleviate this, we tried to compute the baseline value from the 10 or so upstream pixels in the same column and subtracting this from the downstream pixels. This worked well for un-flashed darks, but for flashed darks, the intensity of the flash in each pixel increases up the trail (due to the overall gradient in the flash), thus introducing a spurious systematic trend in the trail, and making it hard to know the real trail contribution. We clearly needed to find another way to measure the trail intensity.

To this end, we returned to the horizontal neighbors, but instead of using the 10 pixels on either side, we just used the ± 1 pixel on either side. This minimized the amount of systematic bias, but the result is that the ACS trails are still systematically low by about 2 electrons. Note that this lower baseline is also present in the pixels upstream of the WP as well, which should not be affected by the WP at all, so the bias is clearly a property of the background subtraction.

5.3 Trailing of Trails

One of the challenges of measuring accurate release profiles for the traps is that the trails we see in the images are not a direct reflection of the release profile. This is because the electrons that are released in the trail often get trapped again by other traps as they themselves get shuffled downstream. The result of this is that even if the release profile is the same for small and large packets, the observed profile *can* be elongated by CTE. This is of course worse at the top of the detector far from the readout register than it is for the bottom of the detector. But the trail is naturally easier to measure when more shifts are involved and the trail amplitude is larger.

Note the difference on the right of [Figure 6](#) between the profiles for WP=10,000 and WP=30,000. There is a sharp spike in the trail residual for the first upstream pixel. This is likely not a change in the trap-release profile, but rather an indication that our model is not properly accounting for the subsequent re-trailing of the first pixel in the trail (which has about 200 electrons for these WPs).

Note also that the AB10 model appears to do a better job with zero background (green curve) than it does with a background of 70 electrons (blue curve). This is clearly an indication that the AB10 model does not properly account for “shielding”. That is, it does not account for how much the background of 70 e^- keeps the more plentiful low traps filled, so that the WPs (which are transported on top of this background) see fewer traps. This “background shielding” effect is why it is helpful to have a reasonable amount of background in ACS images. All of these issues make pinning down the CTE model a complicated and highly iterative process. We begin in the next section.

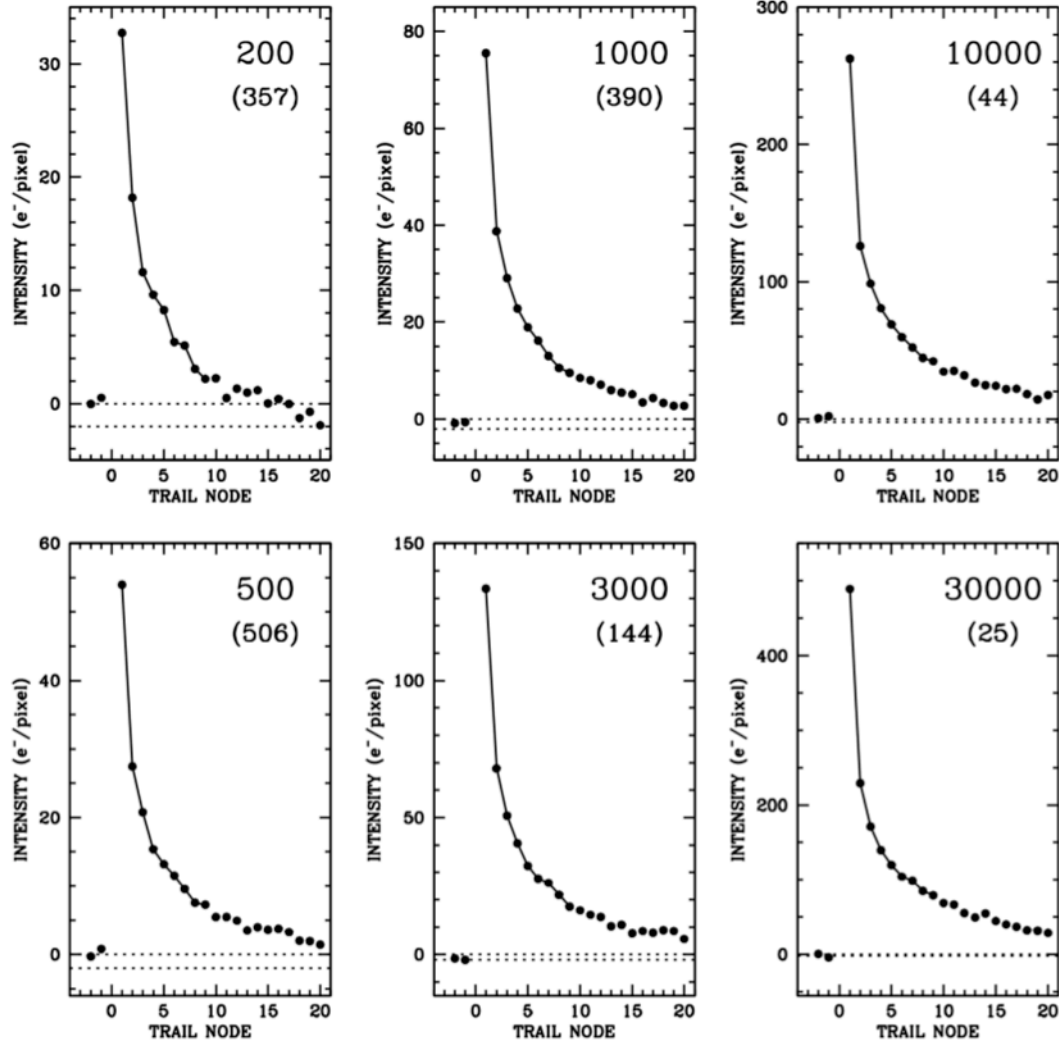


Figure 7: The raw trails in the unflashed dark exposures.

6. Step#1: Re-pinning the bright end

Since the new ACS pixel-based CTE-correction code is based on a reformulation of the WFC3/UVIS code that has been in the pipeline for over 6 years, we decided to do the model-fitting from scratch rather than attempt to perturb the old AB10 model. Thus, this exercise will serve as an illustration of how one might constrain a model to fit any other instrument.

To begin, we construct [Figure 7](#), which shows the uncorrected trails at the top of the detector ($j \sim 1700$) for the long darks with no postflash. The different panels show the trails behind WPs of increasing intensity. The trails appear to have the same shape, but it is hard to measure the trails beyond 15 pixels for the fainter WPs. Note that we decided to change the WP level for the leftmost two panels, since the 100-electron WP suffered from too much CTE losses to include it in the bright-end analysis. Also, we have fewer WPs all around, since we are focusing more on the very top of the detector.

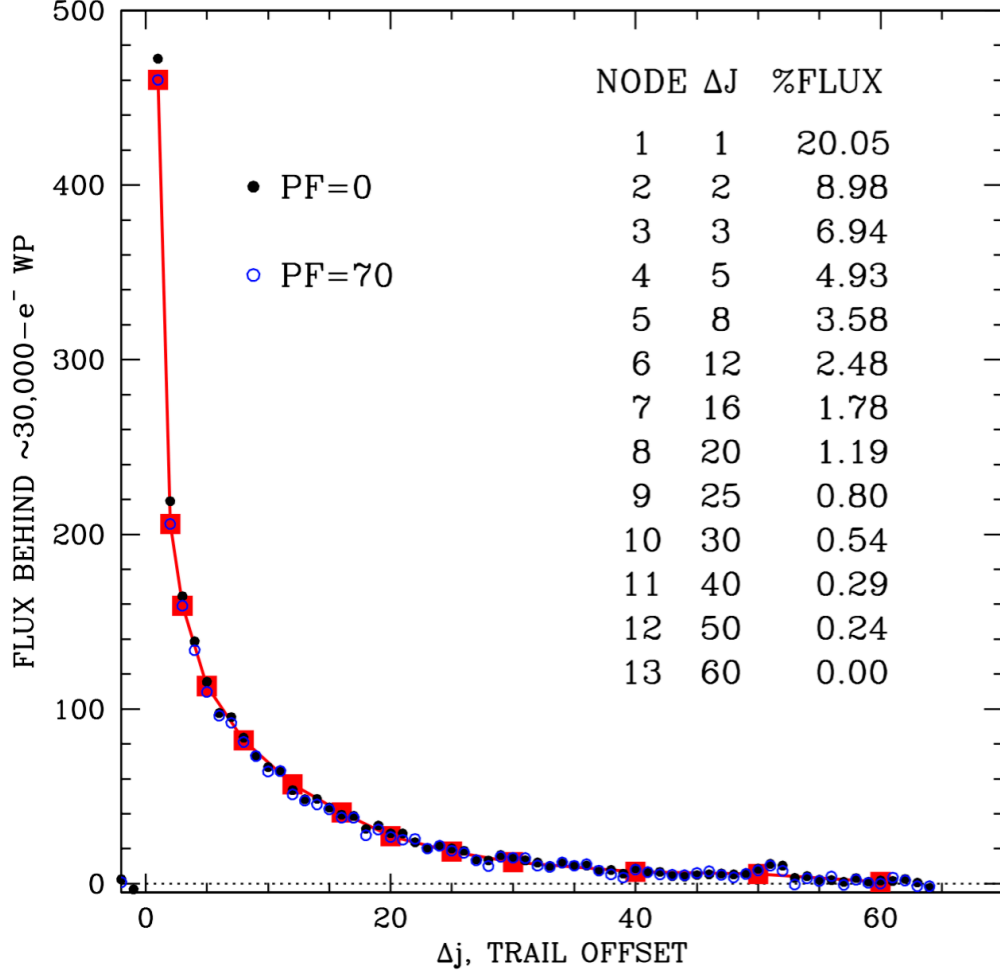


Figure 8: The profile behind the 30,000 e^- WP for the unflashed (black) and 70 e^- flashed (open blue) long dark exposures. The red squares correspond to the locations where the model is pinned down. The model is linearly interpolated between these locations.

We measure the trail shape from the trail behind the 30,000 e^- WP. **Figure 8** shows this curve, for the unflashed long darks and for the flashed long darks. Flashing clearly makes little difference in the shape of the curve for these bright WPs.

The trail is parameterized in the CTE model as $\tau(\Delta j)$, shown as the red squares above. It is essentially the fraction of light that is released at various fiducial distances upstream of the WP and is linearly interpolated between these nodes to get the release fraction at intermediate Δj offsets. About 20 percent of the time a trap releases its electron in the first pixel, about 9 percent of the time, it does so in the second pixel upstream, etc.

We took this curve as the release profile and used it as the starting profile for our model. The curve does not necessarily correspond to the accurate internal profile, since this observed profile could be affected by re-trailing, but it should be a good starting point. We will revisit the trail shape later.

Table 1: The WP_{OBS} and $\phi \times \tau_1$ (the flux in the first trail pixel) were read off from Figure 7 and were used to construct initial estimates for ϕ and WP_{ORIG} . The initial model is simply $\phi(WP_{\text{ORIG}})$.

WP_{OBS}	$\phi \times \tau_1$	ϕ	WP_{ORIG}
200	33	165	365
500	54	270	770
1000	76	380	1380
3000	135	675	3675
10000	232	1160	11160
30000	490	2450	32460

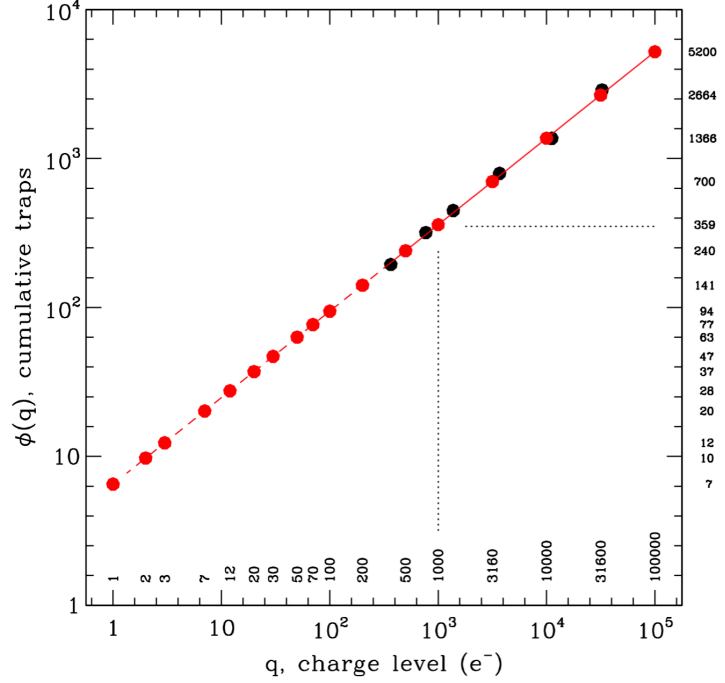


Figure 9: The initial model for $\phi(q)$.

In order to construct our initial model, we need to construct $\phi(q)$, the number of traps a charge packet of intensity q will have to deal with on a trip from the top of a column to the register at the bottom. To do this, we read off from each panel of Figure 7 the amount of charge in the first pixel of the trail (τ_1), then multiply this by 5 (the reciprocal of $\tau_1 = 20\%$, the fraction of the total trail in the first pixel). This gives us $\phi(q)$, the estimate of the total amount of flux in the trail for a WP with q electrons.

Table 1 summarizes this information. Note that the original WP is different from the observed WP that is read out — the trail we are measuring is for the original WP:

$$WP_{\text{ORIG}} = WP_{\text{OBS}} + \phi(WP_{\text{ORIG}}).$$

Note also that for the lowest-intensity WP in the table, the original WP is almost a factor of two brighter than the observed WP, so the trail is not simply a perturbation on the WP, but the trailing changes the WP significantly.

Figure 9 shows this initial model for $\phi(q)$ in a log-log plot. The black points correspond to the fourth column in the table above, and the red points correspond to the fiducial q locations where $\phi(q)$ is specified. Note that we have fit a line to the black points and have extrapolated this downward to lower charge levels below $q = 200$, where we cannot measure the trail intensity in this direct way.

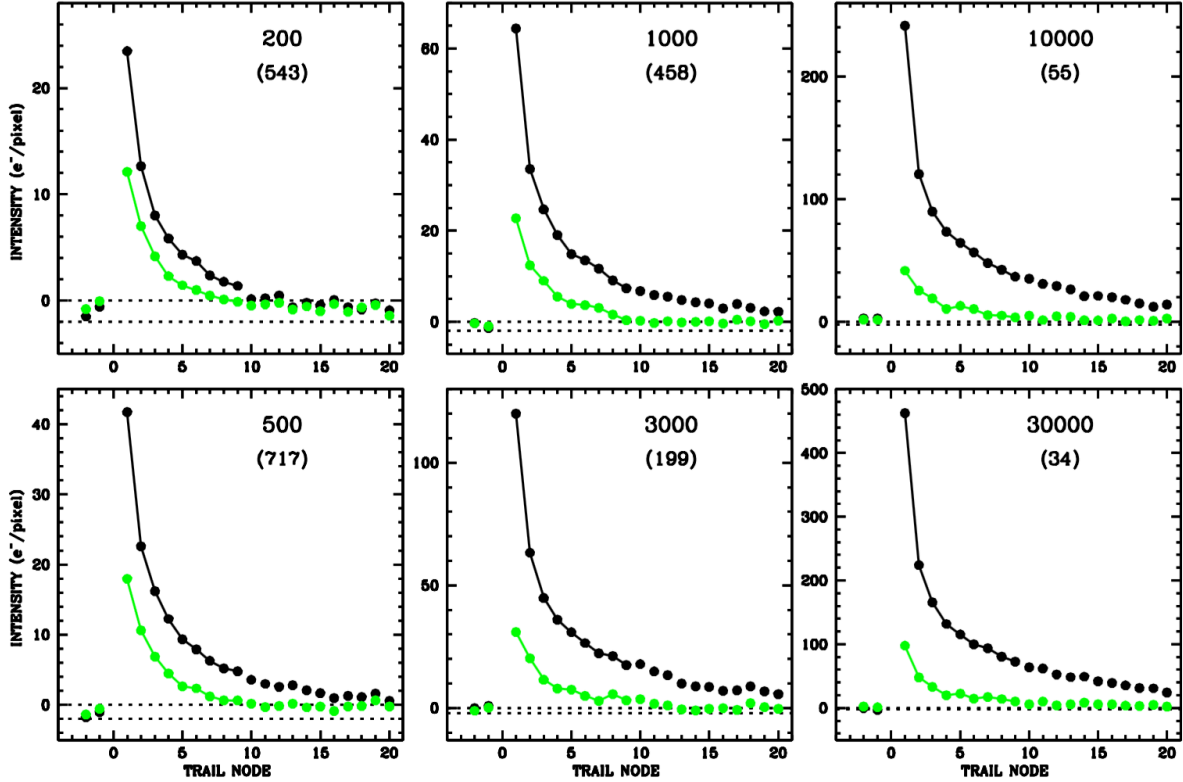


Figure 10: This shows the trail residuals behind the WPs for the initial model based on Table 1.

So, how well does this initial model work? We ran the newly-pinned CTE-correction code on the stack of the unflashed darks and examined the trails behind the same WPs. Figure 10 above shows that it improves the trails, but leaves a lot to be desired, particularly at the faint end. Upon reflection, this shouldn't be so surprising. As the WP is shuffled down the detector, it may start with 1,000 electrons, but by the time it reaches the readout, it only has just over 600, so the number of traps we estimated for (say) 1,000 e^- WPs really apply to fainter average WPs. In other words, $\phi(q)$ is under-estimated in the first iteration.

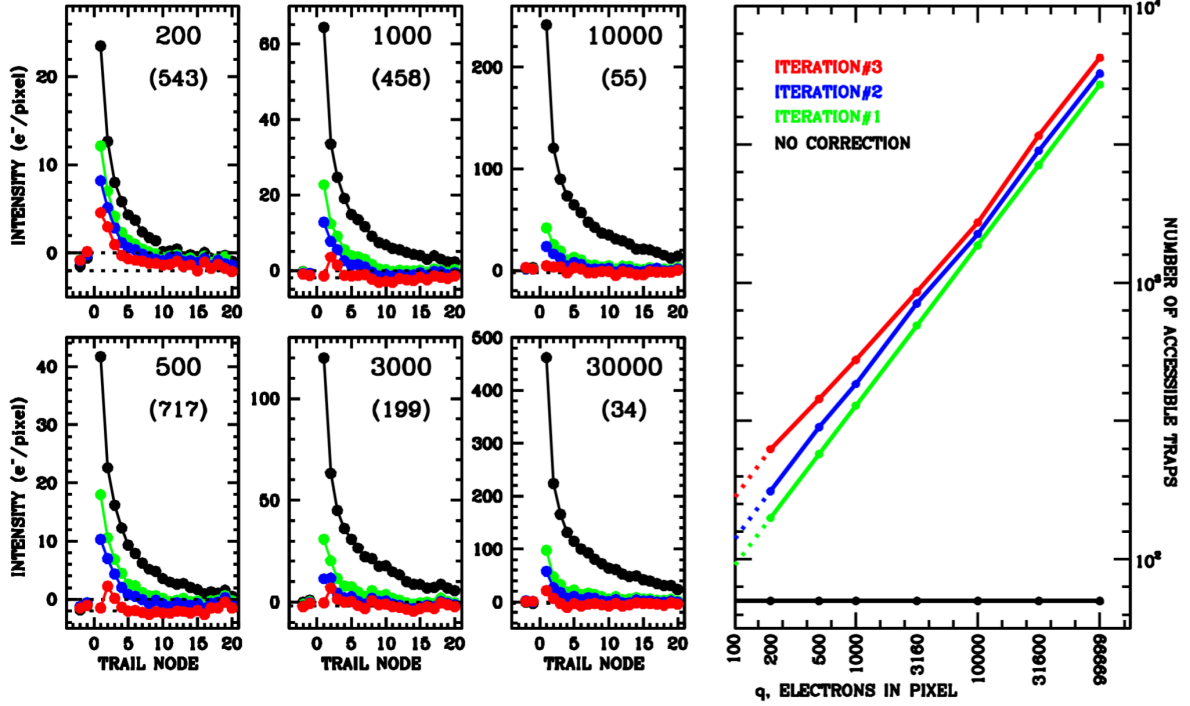


Figure 11: This shows the residuals behind the trails at the top of the detector (~ 1700 transfers) for no correction (black), the first iteration (green), the second iteration (blue), and the third iteration (red). The $\phi(q)$ model itself is shown on the right. It is clear that as we increase the number of accessible traps for a given packet size, the trail residuals go down.

We can of course improve the model by tweaking $\phi(q)$ in such a way as to make the subtracted trails flatter. **Figure 11** shows the results of two additional iterations of tweaking. The tweaks were made by hand in an attempt to make the residual trails as flat as possible. The plots in the right panel show the $\phi(q)$ model for the three iterations. During this process, the trail-release profile was left unchanged.

The red profiles in **Figure 11** look good enough to move on to the next step. We choose not to adjust the profile shape — $\tau(\Delta j)$ — yet, since it will be significantly affected by the faint end of the model, and we need to adjust that first⁴.

A final consideration about the bright end is that when we consider the trails in post-flashed exposures, some of the losses appropriate for a given packet size are masked by the traps kept filled by the background. Thus, instead of measuring $\phi(1000)$ for a WP with 1000 electrons, we actually measure $\phi(1000) - \phi(100)$ when there is a background of 100 electrons. This can make it much more complicated to tie down the model, since we must deal with two points in $\phi(q)$ at a time rather than one. For this reason, when doing the initial pinning of the model, we always start with unflashed darks, since these report the total losses rather than relative losses. In the next section, we examine the faint WPs in the unflashed darks.

⁴ It is complicated to adjust both of these at the same time since there is considerable interplay. For instance, if the model over-subtracts the trail in the first pixel, it is unclear whether we should adjust the profile shape, the total area underneath the trail, or increase $\phi(q)$ at low q , in order to have more trailing of the trail. Therefore, we start by fixing the trail shape (τ) and focus on the trail intensity, $\phi(q)$.

7. Step#2: Intermediate WPs

In the first step, we examined trails in the long darks behind WPs that had between 200 and 30,000 electrons in them. We identified a single trail profile $\tau(\Delta j)$ and a profile for $\phi(q)$ that did a reasonable job modeling the trails seen behind relatively bright WPs in the 1040-s unflashed dark stacks.

The model looked reasonably good for $q \geq 500$, but not very good at the $q=200$ level, where the losses were more than a perturbation and the WP saw many more traps at the top of the detector than it did at the bottom. Below $q = 200 e^-$, we simply extrapolated the trend seen above. Getting down to $q = 500 e^-$ does not quite bring us to a place where we can examine the faint end, where we are more concerned about dealing with the background than restoring sources. Typically, science images have backgrounds between 40 and 80 electrons (see Sokol et al. 2012), so the faint end would extend to about $q = 50 e^-$.

Since the WPs below $q = 200 e^-$ are massively impacted by CTE losses when they are transferred all the way down from the top of the detector, we turn to the *bottom* of the detector where CTE losses can still be handled as a perturbation. In addition, to study these intermediate-brightness WPs, we turned to the short darks, so that these intermediate-intensity WPs will suffer less interference from brighter WPs. To examine the WPs in these shorter darks we used the same list of 100,000 WPs we have been using, which we extracted from the unflashed long darks. This allows us to have a reliable list of WPs even when the WP itself is a low signal-to-noise event in the short exposures. Both the long and short darks are contemporaneous.

Figure 12 on the next page shows the trails behind WPs with between 49 and 350 e^- , for WPs at $j \sim 500$, near the bottom of the detector for unflashed short-dark exposure `jdd811kbq`. As before, the black curve shows the no-correction losses and the red curve shows the post-correction losses for the current iteration of the model (i.e., the red curve in **Figure 12**). Note that with this shorter dark, the WP background is much flatter, and the background bias is no longer at $-2 e^-$; it is much closer to $0 e^-$.

It looks like the trail behind the 350 e^- WP is well modeled, but the other trails are all moderately under-corrected. To improve the model, as before, we adjusted the $\phi(q)$ at the appropriate locations and re-evaluated the model. In particular we adjusted $\phi(200)$ up by 5%, $\phi(100)$ up by 15%, $\phi(70)$ by 20% and everything below $\phi(50)$ up by 25%.

The green points in **Figure 12** show that the corrected trails look much flatter now. This iteration gets us close enough to examine the $\phi(q)$ model at the very lowest levels of q (i.e., for the very smallest charge packets). We do this by examining the impact of background on WPs.

BACKGROUND $\sim 1.5 \text{ e}^-$

$250 < j < 750$

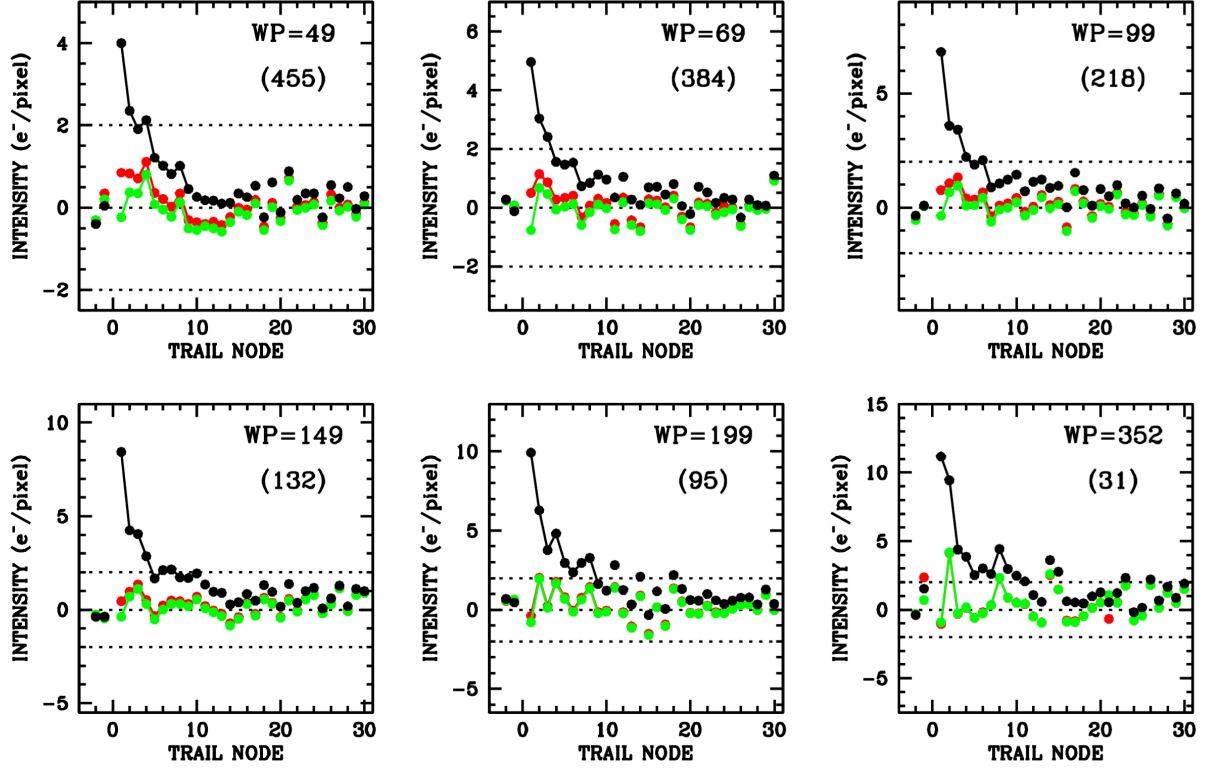


Figure 12: The residuals for fainter WPs in the unflashed short-dark image jdd811kbq at $j \sim 500$, where the CTE losses are about a factor of four lower than at the top of the detector. This enables us to examine the trails behind faint WPs while they are still perturbations. The shorter dark time (25s as opposed to 1040s) allows these faint WPs to be more dominant. As in the previous diagrams, the trails represent a sigma-clipped average of between 455 (upper left) and 31 (lower right) trails. The bin width is about $\pm 10\%$. The red points correspond to the iteration#3 model and the green points to an additional iteration.

8. Step#3: The Faintest WPs

In the first step, we examined trails in the unflashed long darks behind WPs that had between 500 and 30,000 counts. Even after 1700 shifts from the readout register, these WPs had a very clear head-tail structure, making it easy to infer how much flux is in the trail and therefore what the original WP looked like.

In the second step, we examined the trails behind WPs with between 50 and 500 electrons. At the top of the detector, these WPs suffered so much CTE loss that the tails had more flux than the head, making it hard to put everything back. For these intermediate-brightness WPs, therefore we examined WPs closer to the readout (at $j \sim 500$) in the short unflashed darks, so that we could better measure the background and also so that the CTE losses would remain at the perturbation level. This allowed us to constrain the $\phi(q)$ function all the way down to $q = 50 e^-$.

The final step is to pin the $\phi(q)$ function all the way down to $0 e^-$. Recall that we supplied initial values for the lower part of $\phi(q)$ by simply extrapolating the trend seen at the bright end in [Figure 9](#); as we adjusted $\phi(q)$ at lower and lower q , we simply shifted the faint end of $\phi(q)$ to keep up with the intermediate part. Now we need to adjust this lower part independently.

To examine CTE losses for very small packets, it is hard to study the head-tail structure directly, since the trails are often very faint and get lost in the background very quickly. The easiest way to examine CTE losses for small packets is to examine the “shielding” impact of various levels of background on brighter WPs.

[Figure 13](#) on the next page shows the trails behind $\sim 100 e^-$ WPs on a variety of backgrounds. (These WPs were identified on the long darks and scaled down to predict the flux in the short darks.) The nine panels show the trails behind these WPs as seen on postflash backgrounds of $1 e^-$, $2 e^-$, $3 e^-$, $8 e^-$, $13 e^-$, $33 e^-$, $56 e^-$, $70 e^-$, up to about $100 e^-$. As usual, the black points show the uncorrected trail. Within each panel, the labeled “OBS” value in the upper-right corner reports the intensity of the WP in the uncorrected image.

In the upper left panel, the sky background is only $1 e^-$. We see that only 60 of the original 100 electrons survive to the readout register after 500 transfers. This is a 40% fractional loss. If such a WP were transferred all the way down the detector, the transfer losses would whittle the 100-electron WP down to close to the readnoise level.

As we look at panels with higher and higher postflash backgrounds, we see that there is less and less flux in the trail, and more and more flux remains in the WP. At a sky background of $\sim 35 e^-$, the losses are half what they are with zero background. With a sky background of $\sim 70 e^-$, they are a third. It is worth noting that WPs will suffer 2-3 times *more* CTE loss than a star, since a star is not a single pixel and will benefit from a considerable amount of self-shielding. Of course we pin down the model with WPs, not stars, but the pixel-based model should take into account the shielding that stars benefit from. [Section 10](#) will evaluate this.

The green curve shows the reconstructed current Step#2 model. The current model actually does a pretty good job in most of the panels removing the flux from the trail and restoring it to the WP. This means that the power-law extrapolation we used in [Figure 9](#) was a pretty good approximation to the overall behavior of $\phi(q)$. A few of them (such as SKY=33) show a slight over-subtraction. We can tweak $\phi(q)$ to improve this.

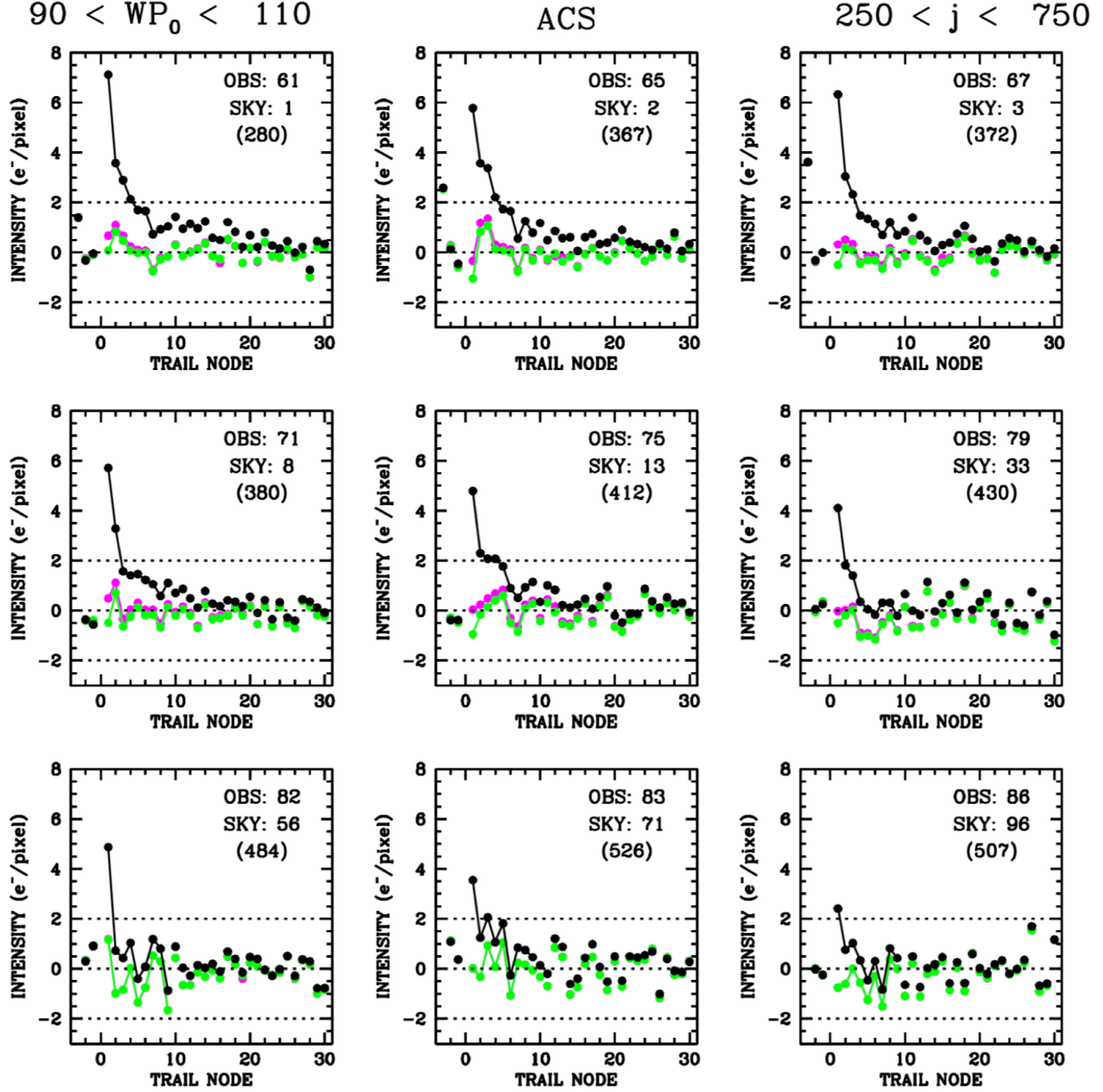


Figure 13: The trails behind $100 e^-$ WPs at $j \sim 500$ for various levels of postflash background: from $1 e^-$ in the upper-left panel to $100 e^-$ in the lower-right panel. The black points show the uncorrected trails, the green the correction after Step#2, and finally the magenta shows the Step#3 correction.

It is worth noting that when we are looking at the trails on top of a sky background, when we *increase* $\Phi(q)$ at $q=\text{SKY}$, then we actually *decrease* the amount of flux in the trail, since we take more of the traps off the table for the WP at that background. This is somewhat counter-intuitive, but this is why the background mitigates CTE losses.

The magenta points in the first few pixels of the trail are on average at about the same level as the points where the trail has leveled off (beyond 10 or 20 pixels). Some are a little high, some are a little low, but by and large we have done a good job modeling the trails behind these WPs.

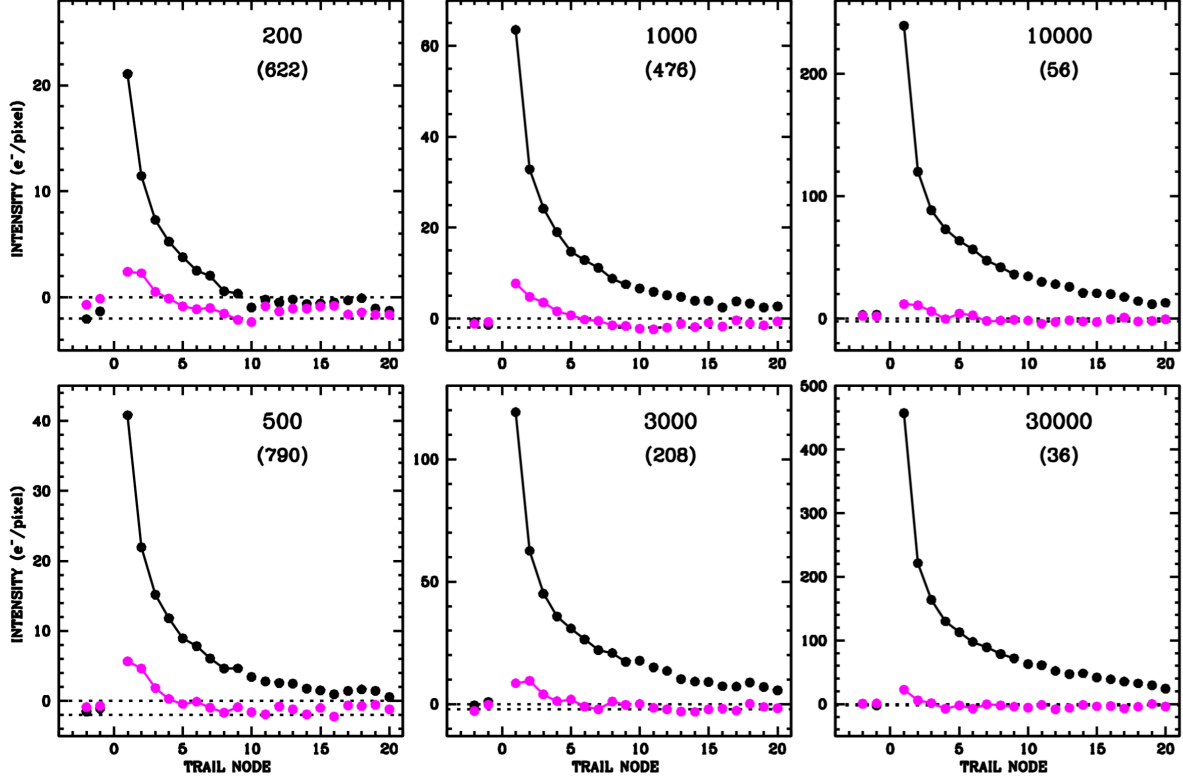


Figure 14: Trail profiles behind intermediate and bright WPs in the unflashed long-dark stack of exposures at $j \sim 1700$. The black points show no correction and the magenta points show the Step#4 correction.

9. Step#4: Revisiting the Trail Profile

When we started pinning the model, we took note of the challenge of solving for $\phi(q)$ and $\tau(\Delta j)$ simultaneously, since CTE-trailing of the profile itself can cause an interplay between the trail profile and $\phi(q)$ for q in the trail. For this reason, we examined the bright WP $\sim 30,000 e^-$ trails and extracted a trail profile from them, and have used this unmodified profile in our analysis up to this point.

It was not clear that the trail should be considered uniform for all WPs — indeed AB10 claimed that the trail was shallower for faint WPs — but it seemed a good assumption to start with. We note that in our study of the trails behind faint WPs in the previous section, we saw no indication that the trail was steeper than the $\tau(\Delta j)$ we obtained from bright WPs, so it looks like the trail profile actually is uniform, but it may be in need of slight improvement. Perhaps the improved model of charge transfer for faint packets now allows us to disentangle the intrinsic trailing from the CTE-blurred trailing.

Figure 14 above shows the trails behind the intermediate and bright WPs in the unflashed long darks far from the readout register. It is clear that the trail profile shows similar residuals in all panels and could benefit from improvement.

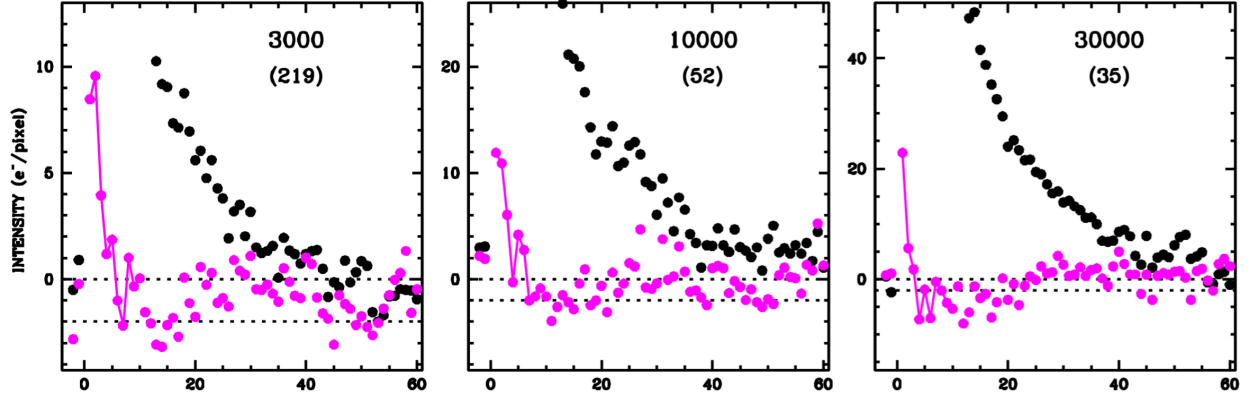


Figure 15: The uncorrected (black) and step#4-corrected (green) trails for the bright WPs in the unflashed long-dark stack, shown out to 60 pixels. We see the same residual trend in all panels, indicating that $\tau(\Delta j)$ needs some slight tweaking.

Figure 15 above shows the full trail profile and residuals out to 60 pixels and zooms in ($\times 10$) on the residuals. It is clear that the trail needs somewhat *less* flux between 5 and 20 pixels than we had initially modeled, and it needs somewhat *more* flux between 20 and 40. We made these adjustments and arrived at Figure 16 below. The cyan points show the new subtracted trails.

The subtracted trails look flatter from $\Delta j = 5$ and beyond, but they show a bit too much flux subtracted close to the WP. The residual has the same shape as the unsubtracted trail (but negative), which tells us we can simply adjust $\phi(q)$ upward a bit at the q level corresponding to the WP. This exercise shows the complicated interplay between the trail profile and the trap distribution.

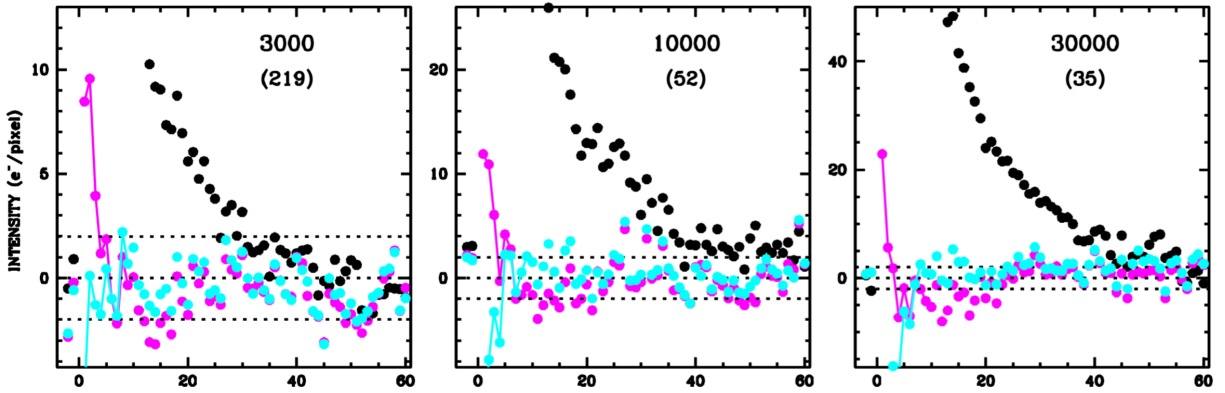


Figure 16: Same as Figure 15, except that we have adjusted the trail profile as described in the text. Cyan shows the adjusted profile.

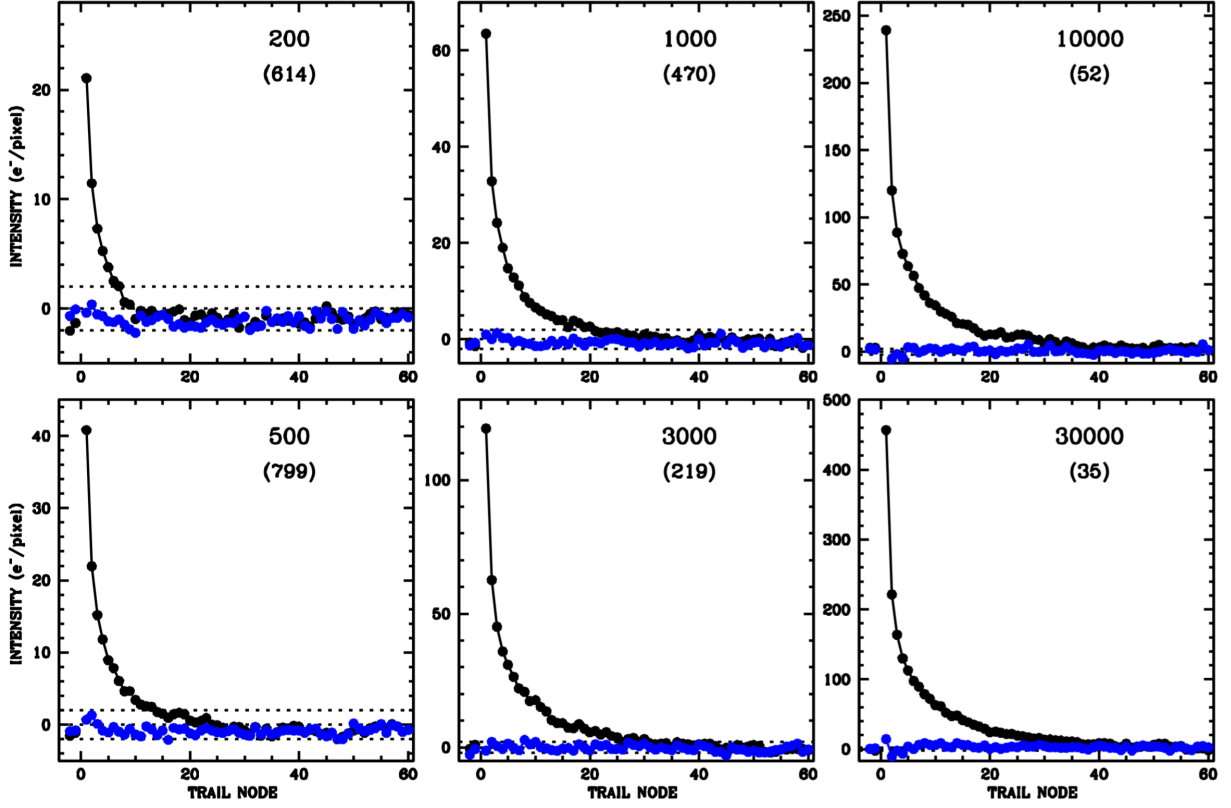


Figure 17: Same as Figure 14, except that we show the full trail all the way out to 60 pixels. This is for the final model.

After the slight adjustment of $\phi(q)$ at the high end to remove the slight residuals seen in Figure 16, we arrive at Figure 17, which represents the final model. Tables 2 and 3 provide the detailed prescription of the final model, and Figure 18 shows the graphical progression to the final model.

Table 2: Final model for the number of traps that impact a q -sized pixel packet.

N	q	$\phi(q)$	N	q	$\phi(q)$
1	1	13	10	100	190
2	2	25	11	200	260
3	3	37	12	500	380
4	7	60	13	1000	520
5	12	80	14	3160	900
6	20	100	15	10000	1625
7	30	115	16	31600	3100
8	50	140	17	99999	6200
9	70	160			

Table 3: Final-model probability that a trapped electron will be released in a particular downstream pixel. This profile applies to all traps.

Node	Δj	$\tau(\Delta j)$	Node	Δj	$\tau(\Delta j)$
1	1	0.1975	9	25	0.0065
2	2	0.0925	10	30	0.0042
3	3	0.0680	11	40	0.0025
4	5	0.0450	12	50	0.0012
5	8	0.0300	13	60	0.0000
6	12	0.0190	14	70	0.0000
7	16	0.0135	15	80	0.0000
8	20	0.0095	16	90	0.0000

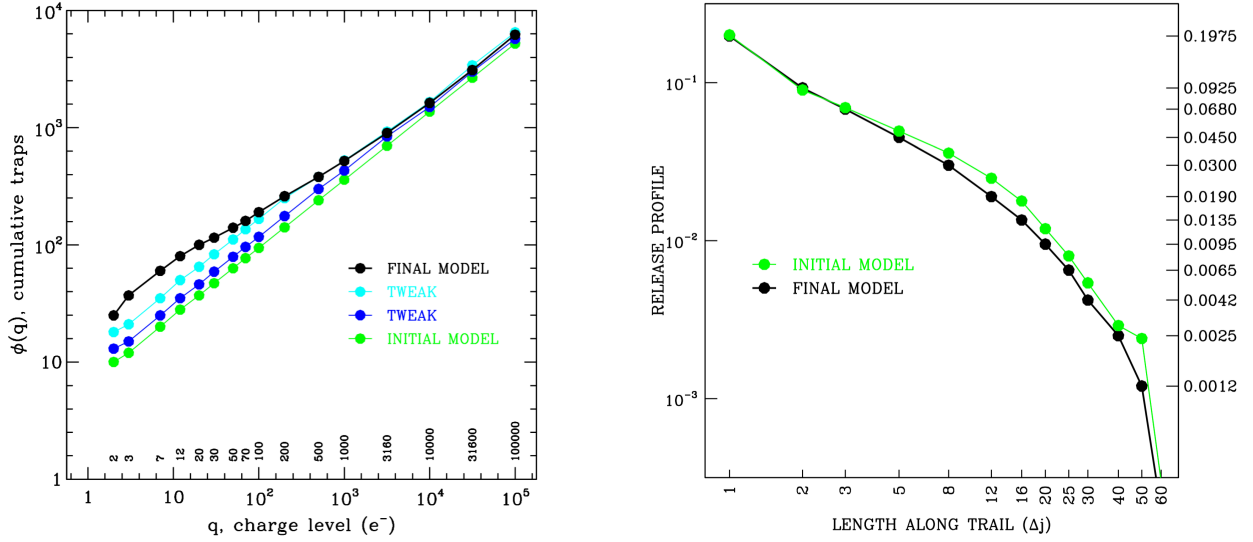


Figure 18: The change of the model parameters, from the initial model to the final model. The left panel shows the number of traps accessible to pixel packets of various sizes, and the right panel shows the evolution of the release profile.

It is worth noting that although AB10 found that the trail profile was steeper and shorter for the smaller packets, the more comprehensive study we have done here shows that the profile is actually the same for all studied traps. The error in AB10 was likely related to the degeneracy between trail shape and trail length, particularly at the low levels where the trail gets lost in the noise after a few pixels. We have used the flashed short exposures with various levels of background to do a more comprehensive study than was possible before. The similarity of the trail profiles and the flatness of the residuals in [Figure 13](#) show that we can in fact use the same trail profile for all trap levels.

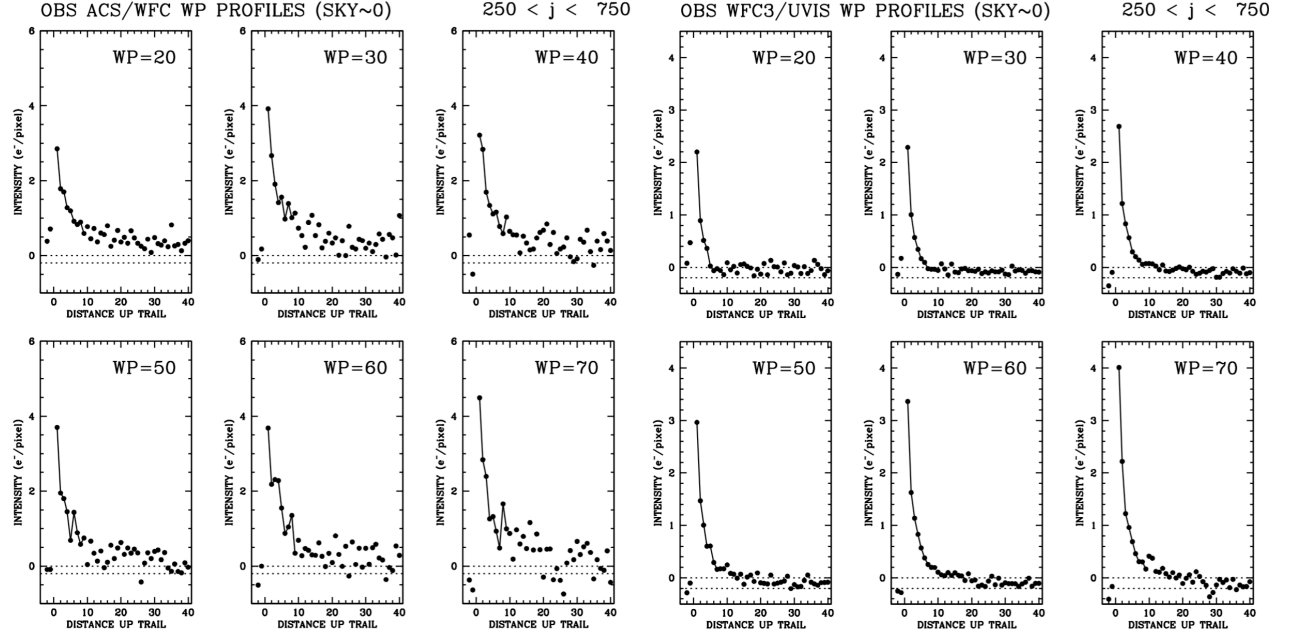


Figure 19: Trail profiles for faint WPs near the readout for ACS (left) and UVIS (right).

As an additional demonstration of this, [Figure 19](#) shows uncorrected trails for WPs with 20, 30, 40, 50, 60, and 70 e^- on zero background near the readout register (where the trails should be pure and not lengthened by secondary CTE). The left set of panels shows the results for ACS/WFC where the profiles are extremely similar in all panels. By contrast, the panels on the right show the same selection, but for the WFC3/UVIS detector. WPs with less than 40 electrons exhibit an extremely steep release profile: in the first two panels (for WP = 20 e^- and WP = 30 e^-) there is *no* discernable flux beyond about 6 pixels, even though it would clearly be observable if it were there. (See the upcoming WFC3/ISR by Anderson.)

10. Empirical comparisons

The next step in the CTE-pinning process is to evaluate the model. We will use on-sky data to do this. In December 2016, Program CAL-14881 took two orbits of observations of a bulge field in parallel with UVIS for just this purpose. We chose a location in the bulge where there are a good number of stars, but not so many that crowding would be a major issue. We took some long exposures and some short ones, all at the same pointing so that the reconstruction of the shorts could be evaluated by comparing to the “truth” of the long exposures.

For ACS/WFC, we used the AMP-B subarray (WFC1B-2K), since it contained plenty of stars and the buffer dump allowed more observations in an orbit. Without changing the pointing we took 4×700 -s deep exposures and four short 35-s exposures with various levels of postflash, achieving sky backgrounds of 26 e^- , 55 e^- , 83 e^- , and 112 e^- , which span the range of backgrounds typical of ACS observations (see Sokol et al. 2012). The eight exposures are listed in [Appendix A](#). Although the pipeline does not perform the de-stripping correction on such subarrays, since it does not have the pre-scan pixels from all four amplifiers, we did our own custom de-stripping using the 24 pixels of pre-scan for each row.

We generated a stack of the four deep exposures (after CTE correction) then scaled this down by a factor of 20 to generate a “truth” image for the 35-s exposures. [Figure 20](#) below shows the full field and a close-up of the deep stack and of one of the uncorrected short exposures. It is easy to see by eye the impact of CTE on stars. It is worth noting that the losses we observe in WPs is much worse than what we see for stars, since there is a considerable amount of self-shielding that goes on with stars, which helps them preserve more of their flux.

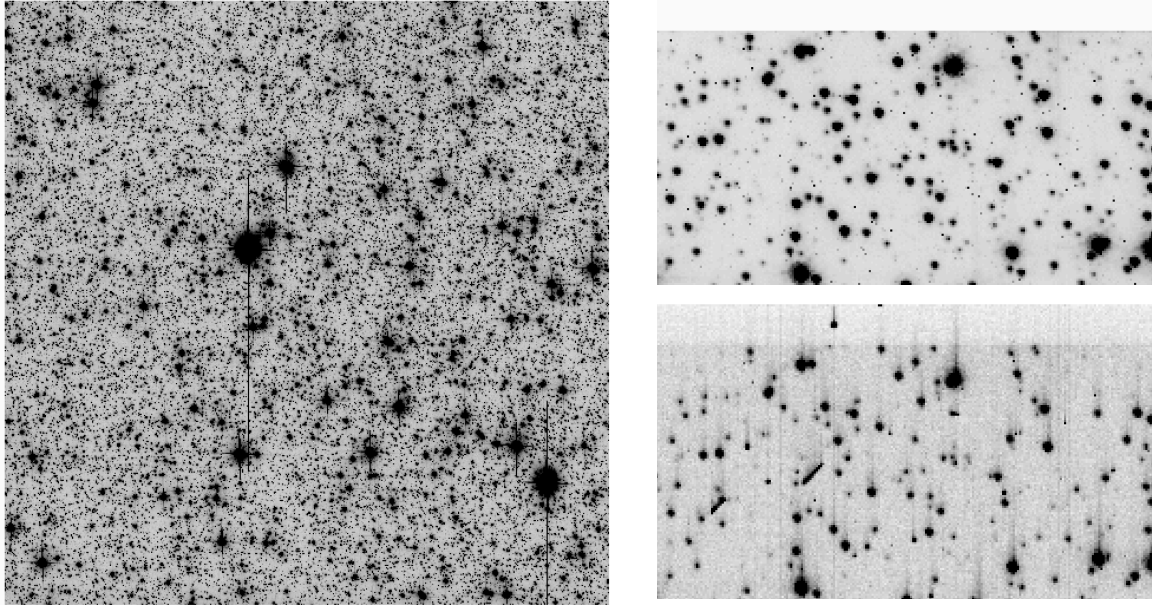


Figure 20: (Left) Stack of 2048×2048 -frame 700s-image of the “Stanek” bulge field that was observed in GO-11664 (PI-Brown). (Top right) Close-up of a 300×150 -pixel region in the upper-central portion of the full field. (Bottom right) Same region as above, but for the unflashed short 35-s exposure that has a background of 26 e^- . The CTE blurring is clear. The top right panel shows what this would look like without CTE blurring (from the scaled-down deep exposure).

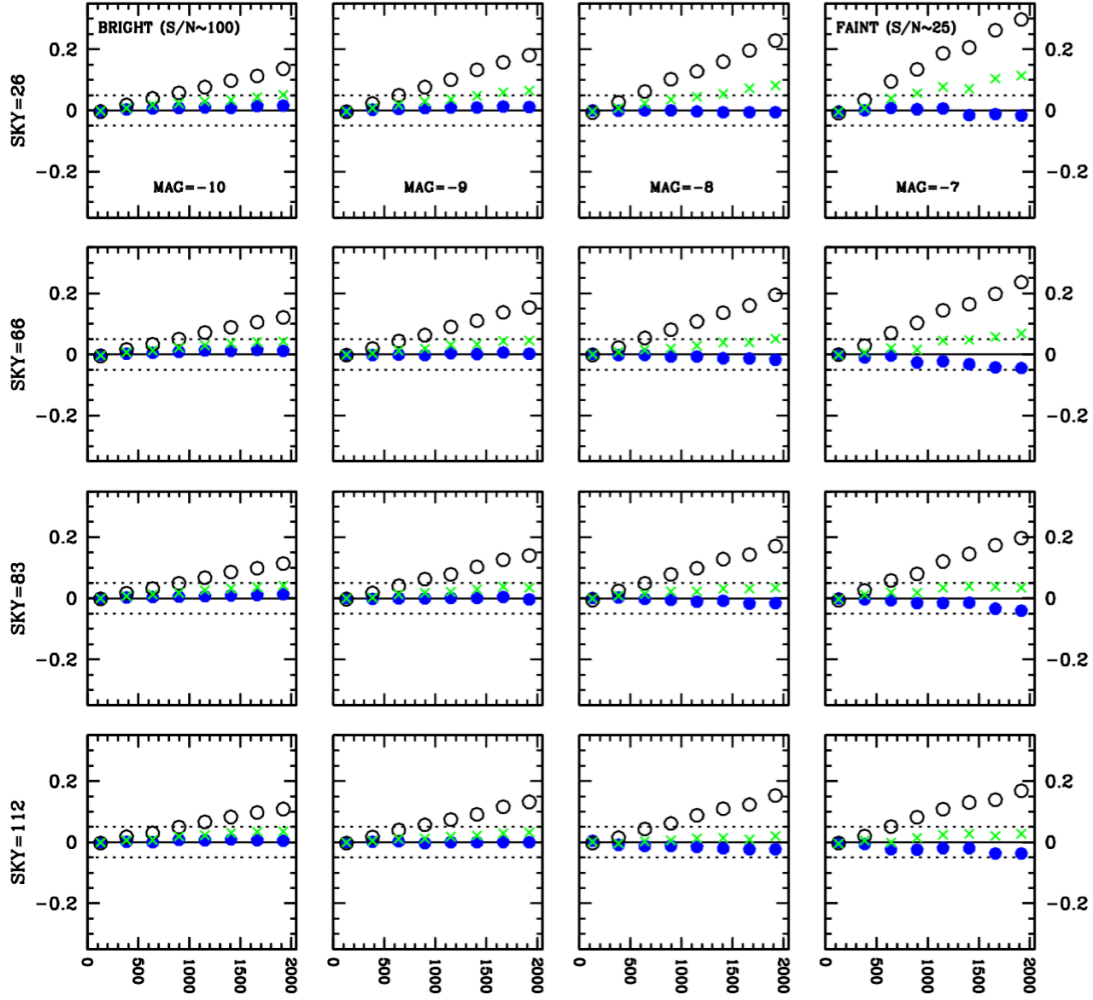


Figure 21: Each panel reports the photometric recovery statistics for a particular brightness of star at a particular sky level as a function of the number of parallel transfers. The four rows of panels show stars on a particular level of sky as labeled at left, and the four columns of panels show stars of the same brightness, instrumental magnitudes as labeled in the top panel of each column. The vertical axis shows the photometric residuals in magnitudes between the CTE-corrected deep stack and the various short exposures. The black open circles correspond to the uncorrected short exposures, while the green crosses correspond to the old model and the blue filled circles correspond to the new pixel-based correction. The upper panels show low sky, while the lower panels show higher sky. The left panels show high S/N stars, while the panels on the right show lower S/N stars. The dotted lines are drawn at $\pm 5\%$.

We performed PSF-fitting photometry on the deep stack and on the individual short exposures (both the corrected and uncorrected images). Figure 21 shows the photometric results as a function of star brightness and sky level. The linear trends with position (the x axis) simply reflect the number of vertical transfers from a star's initial location to the readout register.

It is clear that fractional CTE losses go up when either the flux goes down or the background goes down. For relatively faint stars on low background, the losses can be more than 0.3 magnitude. Even bright stars on relatively high backgrounds suffer about 10% losses in the

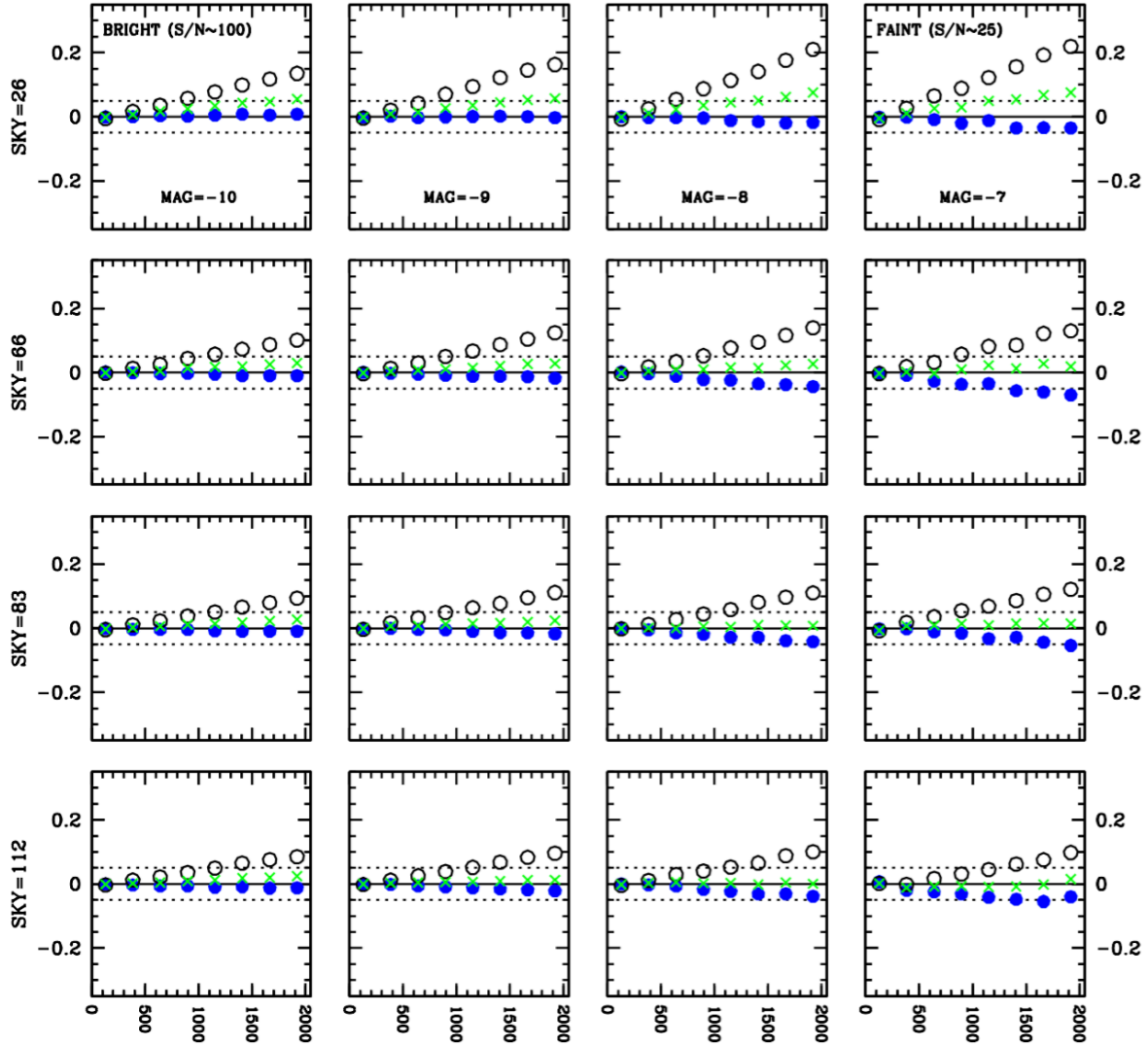


Figure 22: Same as Figure 20, except for the astrometric residuals. The y-axis is in pixels and reports the difference between where the star was in the deep exposure and where it was measured to be in the various short exposures.

uncorrected images. The blue points show that the correction does a pretty good job restoring the photometry, at least for these moderately bright stars.

The astrometric residuals are shown in [Figure 22](#). The moderately faint ($S/N \sim 25$) stars in the low-sky exposures suffer over 0.2 pixel shift away from the amplifier (+y direction, in the system where the pixels are shifted downward) in the uncorrected images. The correction does a good job removing this for the brighter stars, but the fainter stars show a slight over-correction of up to 0.05 pixel.

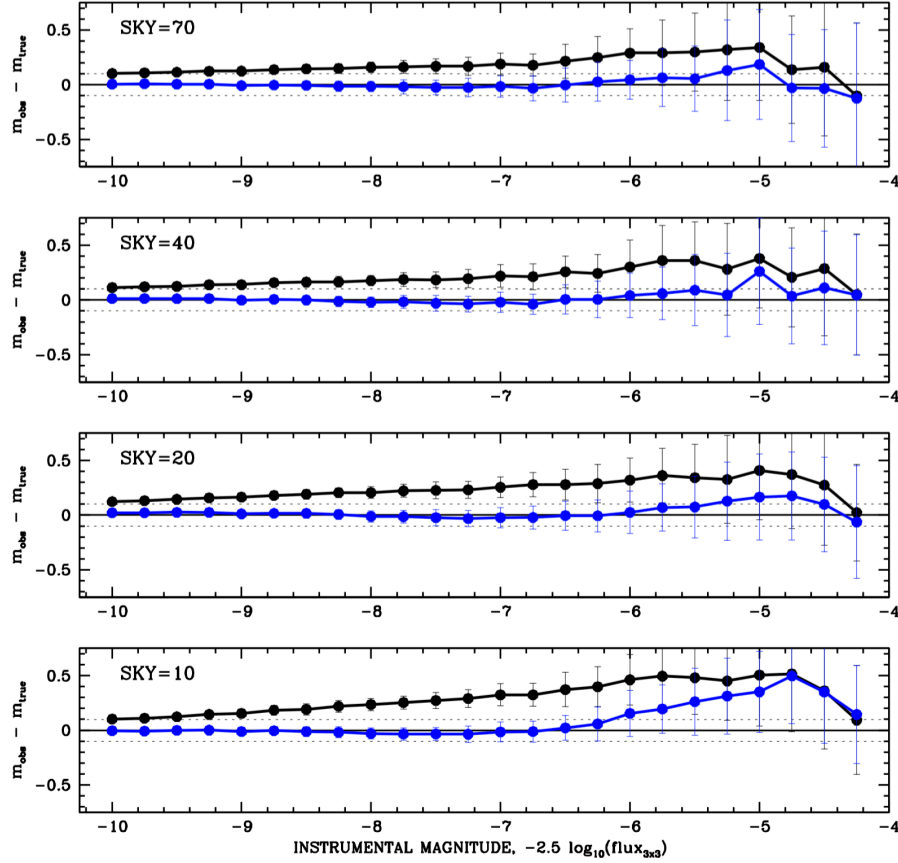


Figure 23: Uncorrected (black) and corrected (blue) 3×3-pixel aperture photometry for stars that experience at least 1500 shifts. Stars at an instrumental magnitude of -10 (which should have a rough S/N of ~ 100) and the faintest stars shown should have S/N less than 10. The four panels show the results for four different background levels.

Figure 23 shows the same results in a more conventional plot for a larger range of star brightnesses. Note that even for stars with S/N of ~ 100 (instrumental magnitude -10), the uncorrected CTE losses are about 5%. We cannot explore the losses brighter than this here, since the stars brighter than this are saturated in the deep exposures and therefore we cannot compare the observations in the short exposures against “truth”.

All of the curves in **Figure 23** take an interesting turn downward at the faint end below an instrumental magnitude of -5 (100 total electrons, corresponding to about $20 e^-$ in the central pixel). It is unclear whether this represents true preservation of flux or is simply an artifact of the difficulty of measuring the background when the pixel distribution is asymmetric. (See above for the trails.)

It makes sense that when a source’s flux is small relative to Poisson-related fluctuations of the background, we might expect the background fluctuations to pre-fill many of the CTE traps so that a good fraction of the faint source’s electrons might make it to the readout register. Such hidden-in-the-background sources would have low S/N (< 1) in a single exposure and as such couldn’t be measured accurately in individual exposures, but perhaps this is telling us that faint sources in stacks might not suffer the pathological CTE losses that we might otherwise expect. This cannot be explored with this data set, but it might be worth exploring this further in future

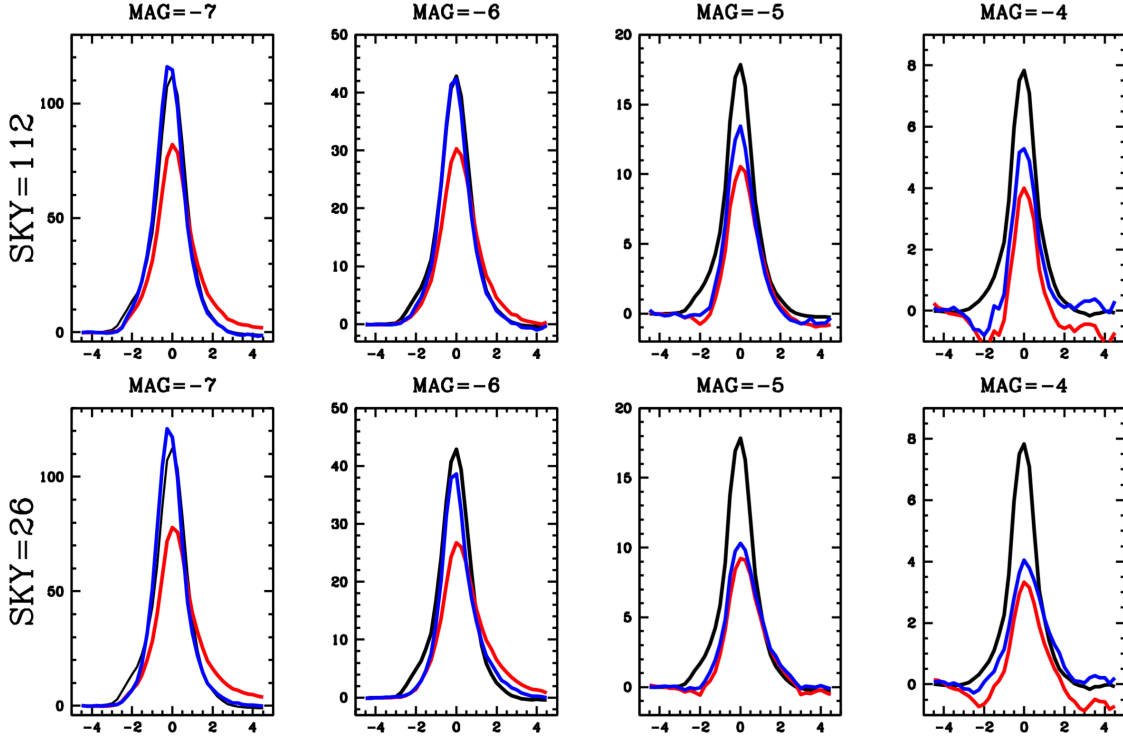


Figure 24: Vertical profiles of stars at $j \sim 1500$ as a function of brightness and star background. The lower panels have sky backgrounds of 26 e^- and the upper panels a background of 112 e^- .

documents, either by studying photometry on deep stacks or by doing some forward-modeling-type experimentation.

Figure 24 shows the CTE losses in a somewhat more direct way. From the deep images we have measured positions and fluxes for stars at $j \sim 1500$, where the CTE losses are large. Using the known positions, we use the deep exposures to estimate how much light should land at various Δy offsets relative to the star's center and scale this flux down to correspond to the exposure time of the short exposures. The black lines show this estimate of the true vertical distribution. In the brightest flux bins (on the left), we would expect 120 e^- in the central pixel and in the faintest bins (on the right), we would expect 8 e^- . The readnoise is 4.25 e^- in all images and the Poisson noise in the background in the top panels is about 10 e^- .

The red curve shows the uncorrected observed profiles. Note that there is more flux lost on the left side of the star and there is often flux gained on the right side. This naturally induces an astrometric shift. The blue curve shows the profiles corrected by the new pixel-based-model corrected profiles. The correction does a pretty good job down to $m \sim -6$ (about 200 total counts), but below this the reconstruction has a very hard time. The reason for this is that it cannot rebuild the profile when the losses are pathological. On the left side of faint stars, the losses are so great that there is nothing to scale-up for reconstruction.

Pretty much anytime the pixel values are close to the readnoise, the model has a very difficult time with reconstruction. This makes sense. We had to be careful not to amplify readnoise, and in the process, we ended up smoothing out some of the faint signal from the sources. This was an unavoidable trade-off.

It is clear that the reconstruction model is not ideal in these cases. When the losses are so great that there's nothing left to reconstruct, then we really should find other ways to make measurements. For these sources, perhaps it would be better to do no pixel-based correction, but to later implement a formula-based correction (see Chiaberge et al. 2012).

11. Implementation in pipeline

The WFC/ACS correction was implemented in CALACS Version 9.2.0 released in July 2017. Products available from MAST include `flc` images, which represent the pixel based correction for single exposures and `drc` images, which represent drizzle-stacked versions of the `flc` products. A routine to perform a forward CTE modeling on simulated images, so that the impact of CTE can be estimated in particular situations, will soon be made available to users.

References

- Anderson, J. & Bedin, L. R. 2010 PASP 122 1035: *An Empirical Pixel-Based Correction for HST's ACS*
- Chiaberge, Marco et al. ACS/ISR 2012-05: *A New Accurate CTE Photometric Correction for ACS/WFC*
- Coe, D. & Grogin, N. ACS/ISR 2014-02: *Readout Dark: Dark Current Accumulation During ACS/WFC Readout*
- Golimowski, D. et al. ACS/ISR 2012-01: *Pixel-Based Correction of the ACS/WFC Signal-Dependent Bias Shift*
- Grogin, N. A. ACS/ISR 2011-05: *Post-SM4 Bias Striping: Characterization and Mitigation*
- Massey, R. et al. 2010. MNRAS 401 371: *Pixel-Based Correction for Charge-Transfer Inefficiency in HST's ACS*
- Mostek, N. et al. (2010). Proceedings of SPIE - The International Society for Optical Engineering. 7742. 10.1117/12.855936: *Charge trap identification for proton-irradiated p+ channel CCDs.*
- Ogaz, S. et al. ACS/ISR 2013-02: *Column Dependency in CTE Correction*
- Ryon, J. & Grogin, N. A. ACS/ISR 2017-03: *Sink Pixels in ACS/WFC*
- Ryon, J. et al. ACS/ISR 2017-13: *Accounting for Readout Dark in ACS/WFC Superbiases*
- Sokol, J. et al., ACS/ISR 2012-04: *Assessing ACS/WFC Sky Backgrounds*

APPENDIX A

Three sets of images were used to for the model-pinning part of this study. [Tables A1](#) and [A2](#) detail the exposures that went into this study for the unflashed long darks and variably flashed short darks. [Table A3](#) lists the exposures used in the on-sky test.

Table A1: List of the 16 individual long dark exposures used to make the stack that was analyzed.

Filename	PID	DATE-OBS	EXPTIME	Filename	PID	DATE-OBS	EXPTIME
jczxmapdq	14400	Oct 19, 2016	1040s	jd50abxgq	14509	Nov 16, 2016	1040s
jczxmbphq	14400	Oct 19, 2016	1040s	jd50acxxs	14509	Nov 16, 2016	1040s
jczxmcpcq	14000	Oct 19, 2016	1040s	jd50ady1q	14509	Nov 16, 2016	1040s
jczxmdquq	14400	Oct 19, 2016	1040s	jd50age8q	14509	Nov 17, 2016	1040s
jczxmgsq	14400	Oct 19, 2016	1040s	jd50aheeq	14509	Nov 17, 2016	1040s
jczxmht5q	14400	Oct 20, 2016	1040s	jd50aiejq	14509	Nov 17, 2016	1040s
jczxmitaq	14400	Oct 20, 2016	1040s	jd50ajeuq	14509	Nov 17, 2016	1040s
jczxmjtpq	14400	Oct 20, 2016	1040s	unflashed_long	combo	2016.89	16x1040s
jc50aax6q	14509	Nov 16, 2016	1040s				

Table A2: List of the 28 post-flashed short dark (25-second) exposures

Rootname	Date	Postflash (s)	Bkgd (e ⁻)	Rootname	Date	Postflash (s)	Bkgd (e ⁻)
jdd811kbq	16.85446	0.0	2.58	jdd808ilq	16.85349	2.9	37.33
jdd801fjq	16.85297	0.1	3.98	jdd809iwq	16.85353	3.2	41.06
jdd802glq	16.85314	0.2	5.12	jdd809iyq	16.85355	3.6	45.93
jdd803gsq	16.85318	0.3	6.32	jdd810j8q	16.85367	4.3	54.13
jdd812m7q	16.85528	0.4	7.61	jdd810jaq	16.85368	5.0	62.69
jdd804h9q	16.85324	0.5	8.72	jdd803guq	16.85319	5.7	70.68
jdd804hbq	16.85326	0.6	9.95	jdd801fhq	16.85296	6.4	79.25
jdd805hxq	16.85331	0.7	11.03	jdd811kdq	16.85448	7.1	88.96
jdd805hzq	16.85333	0.9	13.42	jdd802gjg	16.85312	8.6	105.54
jdd806i4q	16.85337	1.1	15.90	jdd812m9q	16.85530	10.0	124.75
jdd806i6q	16.85339	1.4	19.41	jdd803gwq	16.85321	12.1	146.95
jdd807iaq	16.85342	1.8	24.31	jdd806i8q	16.85340	14.3	173.72
jdd807icq	16.85344	2.1	27.87	jdd807ieq	16.85346	16.1	195.78
jdd808ijq	16.85348	2.5	32.57	jdd810jcq	16.85370	17.9	218.14

Table A3: List of the on-sky exposures of the Bulge from GO-14881.

Image stem	Date-OBS	PostFlash	Sky Bkgd
jddk01m6q	02/12/2017	0.0 s	350 e ⁻
jddk01meq	02/12/2017	0.0 s	350 e ⁻
jddk01mmq	02/12/2017	0.0 s	350 e ⁻
jddk01mtq	02/12/2017	0.0 s	350 e ⁻
jddk01m4q	02/12/2017	0.0 s	26 e ⁻
jddk01mcq	02/12/2017	2.4 s	55 e ⁻
jddk01mkq	02/12/2017	4.7 s	83 e ⁻
jddk01mrq	02/12/2017	7.1 s	112 e ⁻

APPENDIX B

This appendix provides background on the procedure we use to separate an observed image \mathbf{I} into a smooth component (\mathbf{S}) and a component (\mathbf{R}) that is consistent with being pure readnoise, which is random pixel-to-pixel and with an RMS of σ_{RN} . We construct image \mathbf{R} by iteration. We start with a null image \mathbf{R} . We then examine each pixel (i, j) and determine how \mathbf{R} can change to make $\mathbf{S} = \mathbf{I} - \mathbf{R}$ smoother and make \mathbf{R} look more like readnoise. We note that the smoothness that is most important to minimize the CTE amplification is the smoothness along columns, so that is the aspect of smoothness that we will focus on.

We determine how to adjust \mathbf{R} for pixel $[i, j]$ — $R[i, j]$ — according to four “votes”. The first two votes come from the difference between $S[i, j]$ and $S[i, j-1]$ and $S[i, j]$ and $S[i, j+1]$, and aim to make $S[i, j]$ as close as possible to its neighbors. (This is where the preference for vertical smoothing enters in.) The third vote comes from the sum of \mathbf{R} over the local 3×3 box: this should have a distribution centered on zero with a $3 \sigma_{\text{RN}}$. If our effort to make \mathbf{S} smooth ends up putting too much “readnoise” in the same place, then that is not physical. Therefore, if this sum is too large (either on the positive or negative side), the third vote will pull $R[i, j]$ down (or up). Finally, the fourth vote comes from the value of \mathbf{R} itself at this location, which should not be much larger than the readnoise. Each of these four votes can increase or decrease R by at most $0.025 \sigma_{\text{RN}}$ in the relevant direction. The first two votes are down-weighted when the differences between neighboring pixels are much larger than can be explained by σ_{RN} , and the latter two votes are down-weighted when they are less than what we expect from σ_{RN} . In this way, we take a conservative approach to attributing image variations to readnoise, making the image smooth, but at the same time ensuring that \mathbf{R} has the properties of readnoise.

With each iteration, we adjust each pixel in \mathbf{R} according to these four votes. This adjustment is at most $0.1 \sigma_{\text{RN}}$ for a single iteration. After each iteration, the neighboring pixels change and we re-compute how each pixel in \mathbf{R} should adjust to make image $\mathbf{S} = \mathbf{I} - \mathbf{R}$ smoother. We iterate this process until the RMS of \mathbf{R} is equal to the readnoise or until we reach 100 iterations, whichever comes first. Generally, things converge to readnoise-level adjustments after 70 or so iterations. In this way, we slowly arrive at an image \mathbf{R} that removes as much of the pixel-to-pixel variation as possible without attributing *too* much variation to readnoise.

The result of this is that we have an image S that should be generally representative of the structure in the image (modulo readnoise). **Figure B1** shows this procedure for a bias-only image, and **Figure B2** shows the procedure for a starfield far from the readout amplifier.

Once we have separated I into S and R , we then determine S' , the CTE-corrected version of S . The difference $\Delta S = S' - S$ then tells us the likely shift of flux that resulted from the readout process. We can then adjust the original image by this amount: $I' = I + (S' - S)$. In this way, the readnoise is never actually removed from the image. It is simply removed from the image that the CTE correction is determined from.

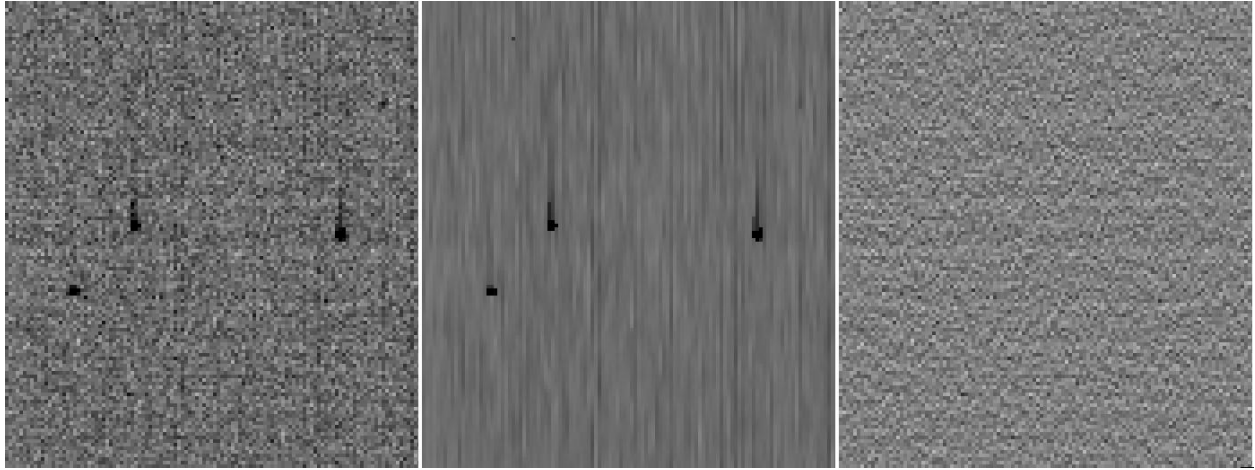


Figure B1: These images show the readnoise-separation procedure for a bias image, which is all noise with a few readout-CRs. Left: the original raw image (image I). Middle: The smooth image (image S), and Right: The pure-noise image (image R). $I = S + R$.

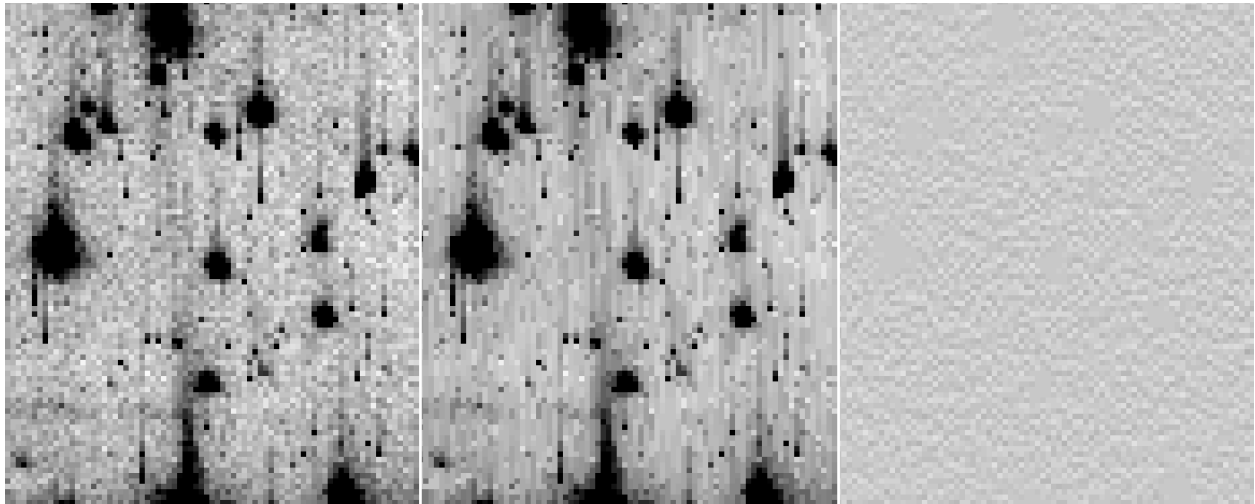


Figure B2: Same as in Figure B1, except that it is an external image with stars in it. This portion of the image is far from the readout register, hence CTE blurring is significant.