Deepspace Interplanetary Navigation Operations Colorado Research EXplorer (DINO C-REx)

DINO C-REx Technical Memorandum

Document ID: DINO_C-REx-Image Generation

HOT DARK PIXELS

Prepared by	Ishaan Patel

Status: Initial Version
Scope/Contents
An overview of hot and dark pixels and the implementation of this random process in the DINO C-REx camera object.

Rev:	**Change Description**	**By**
1.0	Initial Release	Ishaan Patel

Contents

1 Problem

All camera charged-coupled devices (CCDs) exhibit anomalies that in turn, affect the produced image. One class of anomalies are hot and dark pixels. Over time, especially in the presence of a high electric field, a pixel's dark current may escalate. The pixel will then store charge more easily and become even more sensitive to light. On the opposite end, cold or dark pixels do not store charge as easily and thus, are less sensitive to light. This document seeks to describe the implementation of hot and dark pixels in the DINO C-REx camera module. As there is no true formula for such random phenomenon in CCDs, the model used to do so is based on a combination of various sources that provide possible numbers of this occurrence in a camera detector array.

2 Assumptions

To better understand the scale between hot and dark pixels, let 125 be the assigned pixel value that defines proper light sensitivity for an image. Over the time of flight, a current may build up on certain camera pixels – causing them randomly become warm. If so, it may remain stagnant in with its warm value, or it may continue to get warmer. Once it reaches a certain threshold (in this case, a pixel value of 225), the pixel shall be classified as hot.

Pixel Type	Threshold Value
Average	125
Hot	225
Warm	200
Cold	50
Dark	25

Table 1: Allocated Pixel Values

On the opposite spectrum, a pixel may randomly start to go cold if not enough current is passed through. Once it reaches a threshold pixel value of 25, then the pixel is classified as dark. With regards to terminology, an impure pixel is referred to as any pixel that has a value other than the average 125. One important assumption to note is that once a pixel becomes either a hot pixel or a dark pixel, that pixel cannot revert back to a respective colder or warmer value. However, warm and cold pixels have the opportunity to fluctuate between its current value and one closer or farther from the average, depending on which spectrum of the scale it inhabits.

3 Method

3.1 Creating an Impure Image

The method camera.calculate_hot_dark() is a function of the resolution height and the resolution width of the camera CCD array. First, a completely impure array of same dimensions as the image detector array is initialized. The initialized values are normally distributed values between 1 and 255. Note that these values shall be altered in a later portion of the hot-dark pixel implementation. An example of this completely impure image with a varying range of hot and dark pixels is shown below in Fig. 1. The color-map legend on the right of the image shows the colors depicting a range of hot (red) and dark (blue) pixels. Average pixels are represented by a pure white hue as noted by the median of the scale.

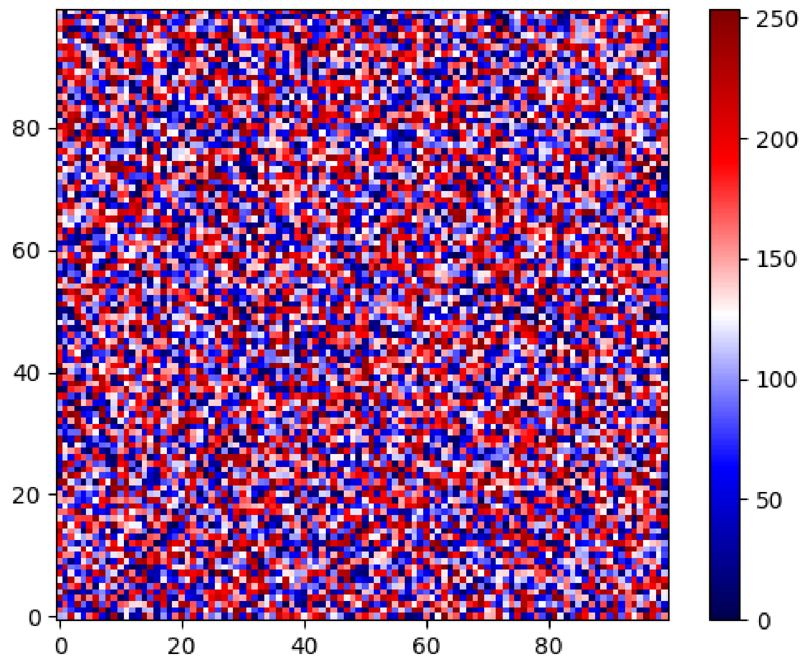

Fig. 1: An Exaggerated Impure Image

3.2 The Gaussian Knob

To choose how many pixels will be impure out of all the pixels that make up a camera image detector array, a Gaussian "knob" is implemented. This is a portion of the camera module that the user is encouraged to change to better reflect the user desired number of impure pixels per image. The values for the mean (μ) and standard deviation (σ) shall dictate the amount of impure pixels to start the hot-dark randomness in the camera image. The Recommendations section of this document describes other methods for this implementation according to the researched sources.

$$Knob = N \sim (\mu, \sigma) \tag{1}$$

$$ImpurePixels = Knob * (TotalPixels) \tag{2}$$

$$NormalPixels = (1 - Knob) * (TotalPixels) \tag{3}$$

As per the equations above, the knob itself is simply a randomly (Gaussian) chosen value with a user defined mean and standard deviation to dictate the percentage of total impure pixels in the detector array per integration time step.

3.3 Integration

Based on the number of pure and impure pixels allocated from the Gaussian knob, a Boolean mask with the size of the detector array is created. Mask values of the Boolean [1] portray impure pixels (Note: these do not have a specific hot or darkness to them yet), while a mask value of [0] portrays a perfect pixel (of value 125). The initial Boolean masks simply creates an array with zeros and ones, consecutively. However, to distribute these values throughout the image, the mask is shuffled multiple times in order spread the impure pixels randomly among the normal pixels. Using this mask, the perfectly pure image (of average values 125) is filtered by replacing Boolean array indices of 1 with random impure values ranging from 0 to 255 from the completely impure image as described in the first subsection. This creates the new initialized detector array with randomness dependent on the Gaussian knob, before a time-step integration is performed.

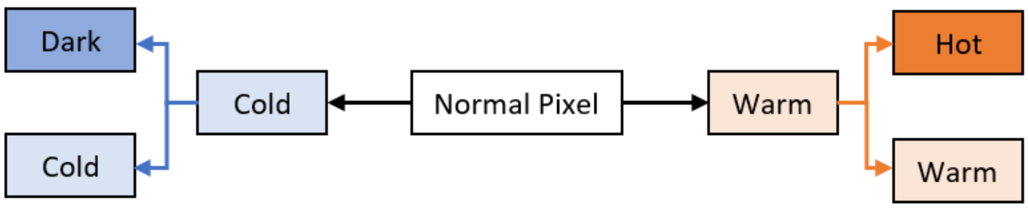

Fig. 2: Block Diagram Showing the Behaviour of Pixels Upon Losing Normality

During each integration time step, which the user may also alter according to preference or specification, impure pixels may retain the same magnitude of impurity or become even more impure. Figure 2 shows a flow chart of how pixels will behave once they lose their average threshold value. To summarize this aspect of the phenomenon, the following logic is put in place. Any pixel that is between the warm threshold and hot threshold has the possibility of its pixel value randomly increasing with each time step. Similarly, a pixel may also have the tendency to randomly lower its pixel value if it is between the cold and dark thresholds. However, once a pixel becomes impure, it can no longer become pure again. For example, if a pixel becomes warm, it does not have the possibility of returning to the average value of 125. This general logic flow is re-iterated at each time step.

3.4 Normalization of Array

Once the integration loop is finished, the hot-dark array pixels of random amount will have magnitudes ranging from 0 to 255. This entire array is then divided by the average value of 125 in order to create a "normalized" hot-dark array. Once the camera.image class is called upon to take an image, the normalized hot-dark array will be multiplied by the original detector array of the image. This will then either amplify, weaken, or keep the same magnitude of each pixel in the detector array. Figure 3 below depicts a 100x100 blank image with the pixel values randomly allotted and distributed via the calculate_hot_dark() method in the camera module.

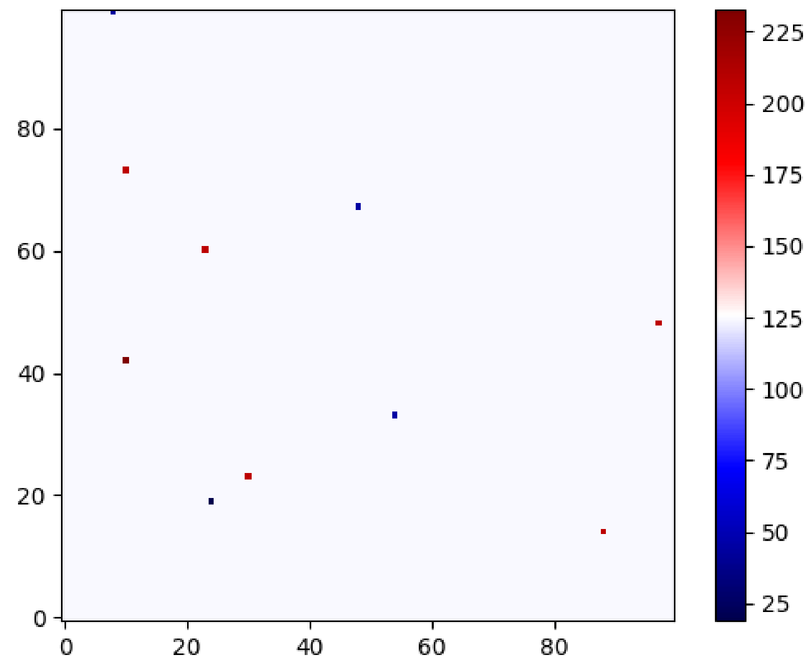

Fig. 3: Hot-Dark Pixels Added to a Blank Image

4 Recommendations

One of the key aspects of the hot-dark pixel modeling is determining how many of the total pixels are faulty. The current implementation in the camera module utilizes a Gaussian knob to initially determine this value. However, a high-performing scientific camera manufacturer, Andor Technology, grades their cameras with the following tolerances of allowable hot or dark pixels. The manufacturer dictates approximately 80 dark pixels per million total pixels, and around 60 hot pixels per million total pixels. The following are suggested improvements to the hot-dark pixel calculation for the camera class.

1. Implement hot-dark columns. This is a scenario where an entire pixel column of the image randomly becomes over or under sensitive to light. Andor dictates 1 hot or dark column each per thousand detector array columns.

2. Another possible method of implementation is directly dependent on the dark current passing through the CCD. The dark current value (with units of e-/pixel/time) would dictate the number of impure pixels on a logarithmic scale. One may reference source [2] for more information.

3. Saturation and blooming is another anomaly that could be added to the hot-dark calculation. This is the idea where if a pixel becomes too hot, the current passing through that pixel may overflow and spill into the surrounding pixels. Thus, these surrounding pixels will gradually become hotter as more current spreads - resulting in a patch of hot pixels in an image. This anomaly would pair well with the aforementioned dark current input affecting the impurity of pixels in an image.

5 References

1. Andor, CCD Blemishes and Non Uniformities, *Andor* Available: http://www.andor.com/learning-academy/ccd-blemishes-and-non-uniformities-black-pixels-and-hot-pixels-on-a-ccd-sensor.

2. *ACS Data Handbook* Available: http://www.stsci.edu/hst/acs/documents/handbooks/currentDHB/acs_Ch44.htm

3. Widenhorn, R., Rest, A., Blouke, M. M., Berry, R. L., and Bodegom, E., Computation of dark frames in digital imagers, *Sensors, Cameras, and Systems for Scientific/Industrial Applications VIII*, 2007.

4. Sanabria, J. J. C. A., "XParallax viu a free software for automated astrometry," *Hot, cold, dead pixels and cosmic rays* Available: http://www.xparallax.com/blog/hot-cold-dead-pixels-and-cosmic-rays.

5. Dunlap, Justin Charles, "Characterization and Modeling of Nonlinear Dark Current in Digital Imagers" (2014). *Dissertations and Theses*. Paper 2073.