

Test report: RunCam Night Eagle Astro  
Bob Anderson  
1 May 2017 (update of the April 20 report)

Summary:

The RunCam Astro version of the Night Eagle line is a small, low cost, black and white camera with a light sensitivity very close to that of the best cameras in common use today for occultation timing.

The camera can operate in either 640x480 NTSC format or 720x576 PAL format with format selection controlled by an OSD (on-screen display) menu item.

It offers three levels of light integration: 2 fields are integrated during standard frame rate observations; for increased sensitivity at reduced temporal resolution, 4 field and 8 field integration may be selected.

The camera response curve is gamma-compressed, which cannot be turned off, but in an appendix to this report it is shown that post-processing of captured images to linearize the response was very successful, should that be needed.

The image sensor is large, utilizing CMOS technology coupled with a chip interface designed to make it appear like a CCD sensor. As a result of its CMOS underpinnings, the camera exhibits what is commonly referred to as a 'rolling shutter/readout'. This characteristic is not an impediment to millisecond precision timing of occultation events, but does require that some different procedures be followed. Needed is: a change to light curve generation procedures by processing in field mode rather than frame mode; a record of the vertical (y) position of the target star on the image at occultation D and R events be kept; the application of a simple linear equation to calculate the required camera delay adjustment to edge times utilizing the y image position of the target star at event time.

No hot pixels were present in the camera tested for this report or in four other RunCam Night Eagle cameras with the same image sensor. No dead (non-responsive) pixels were observed in the tested camera. The image sensor appears to be resistant to hot-pixel generation.

The prototype supplied for testing exhibited light leakage around the back panel. RunCam has indicated that production versions of this model would be light-sealed.

## Introduction:

The tests reported in this document were carried out using a significantly modified version of the ArtStar apparatus. The original (ArtStar 1) device was described in detail in an earlier document, but briefly, it consists of a four inch diameter mailing tube with an optics assembly at one end (shown in the photo below) and two microprocessor controlled LED 'targets' at the other end. The microprocessor has access to field synch pulses extracted from the composite video output of the camera and the 1 pps (pulse per second) signal from a GPS chip. This, and a lot of software, comprises a very flexible testing system for characterizing camera performance.

### About the modifications:

In ArtStar 1, the target illumination was controlled by lighting the target LEDs for a fraction of a field or frame time, starting at the beginning of a field or frame as indicated by the synch pulses extracted from the composite video. This approach works well with a CCD camera because all of the internal operations such as light accumulation and readout are synchronized to field/frame boundaries. I would describe ArtStar 1 illumination scheme as a field-rate pwm (pulse width modulation) system which runs at approximately 50 to 60 hZ.

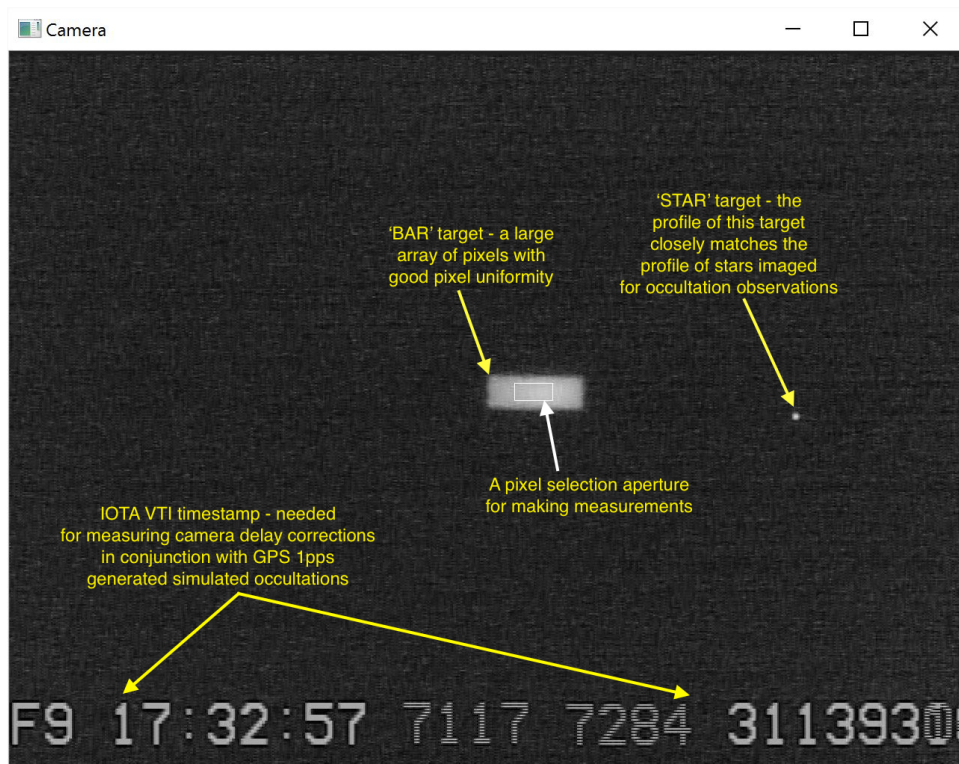
But with the 'rolling shutter/readout' characteristic of CMOS based image sensors, the assumption of internal operations being synchronized to field/frame boundaries is not necessarily valid. To avoid depending on such an assumption, the LED illumination control was changed to a 4.8 kHz pwm system; synchronization to the composite video is no longer utilized.

The change was not without its challenges because, at that frequency, low levels of light correspond to fractions of a microsecond on/off times for the LED. At this timescale, it becomes important to take into account the fact that an LED takes a finite amount of time to produce light after current flow is initiated through the device (there is a depletion zone capacitance that must be filled by the incoming charges before carrier recombination can start emitting light). This problem is worse at low LED currents such as are used in ArtStar --- the LED current is usually in the microamp range rather than the normal milliamp range.



A special driver circuit was designed that utilizes a high speed (gold doped) switching transistor and a bias circuit to keep the LED 'almost on' at all times. That scheme works well down to about 150 nanosecond on/off times for the target LEDs --- no detectable non-linearity above that point.

Below is an image of the LED targets as captured by the RunCam Astro during a test run:



To complete this introduction, the camera-under-test was supplied to me by RunCam as a prototype. It is identified as follows:

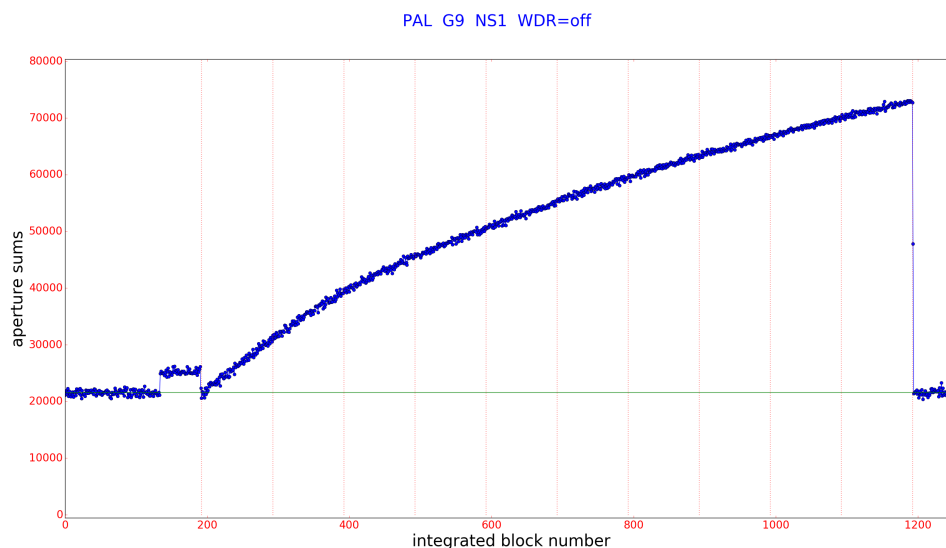
from the box label:      SKU: NIGHT-EAGLE-AW-N  
from OSD:                software version: 2017031952

## Report Organization:

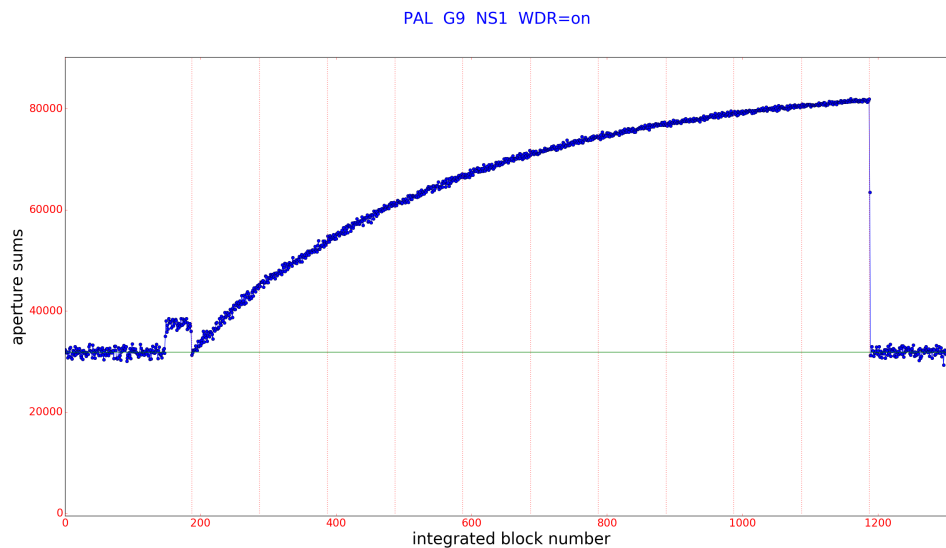
This turned out to be a rather long document. To minimize the burden on a reader, it is organized in 'sections' and 'appendices'. The 'sections' are designed to present measurement results without much discussion of methodology. Some 'appendices' focus on methodology (how the measurements were obtained). Others document specialized tests.

## Section 1: Camera response

A basic test of camera operation is how it responds to illumination that is increased linearly from zero to a maximum value (which is usually adjusted to avoid saturation, as it was for the plots that follow). The following curve shows the gamma-compression employed by the Astro. The 'bar' target is used for this test and the WDR (wide dynamic range) option is turned off. The notation on the plot is to be read as: PAL format in use, Gain (described in the next section) is set at its maximum value of 9, the Night Shutter (which controls level of integration) is set at the minimum value of 1 (= 2 fields). The little 'pedestal' at the beginning of the ramp is just to show clearly where the zero illumination level begins and is preceded by a section where the 'bar' LED is off. The plot shows that the Astro response starts immediately from zero illumination --- not all cameras are this well behaved in this region.

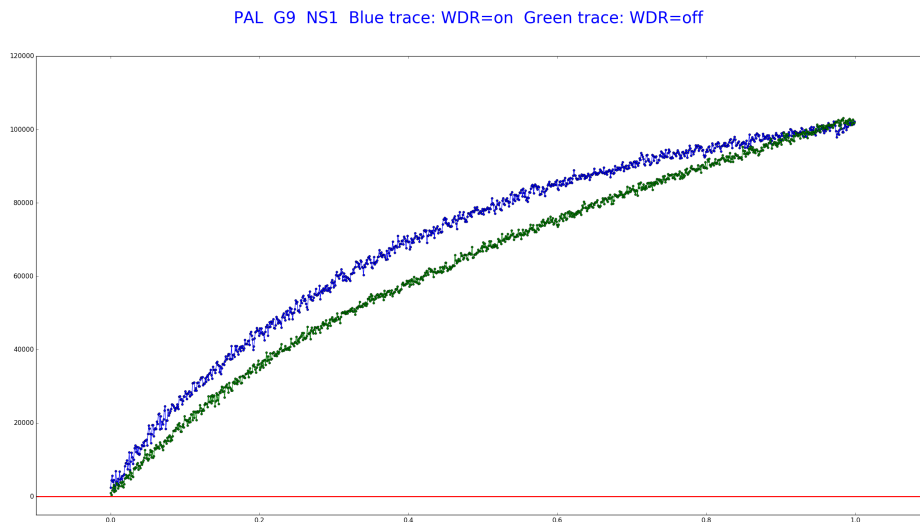


If the 'wide dynamic range' (WDR) option is turned on, the response curve becomes:





To be able to clearly compare the response both with and without the WDR option, the above curves were plotted together, after a needed baseline adjustment (the baseline level increases when WDR is turned on) and the comparison plot looks like:



Note: The gamma response of the Astro cannot be turned off --- that is, there is no gamma = 1.0 option. However, the response of the camera is very well behaved and lends itself to post-processing linearization. This process is described in the appendix titled '*Linearization of camera response*'. I was able to define a single linearization function that was effective for two different cameras, operating in both PAL and NTSC format, at all three levels of integration, with the top three Gain settings (9,8, and 7) and two different StarTech frame grabber models (SVID2USB2 and SVID2USB23). Limovie can also be effective when its gamma reverse is used with a setting of 0.45 (for WDR=off).

Without linearization of the camera response curve (at the pixel level), astrometry will be compromised and sub-frame timing should be avoided.

## Section 2: Camera gain control

As will be explained later, this camera has an 'always-on' AGC (automatic gain control) system. But unless the overall image brightness passes some threshold, the AGC system will operate the camera at its maximum possible gain. It is sometimes the case that this max gain is too high for a particular observation situation. The Astro provides for an eight step gain reduction from max gain, controlled by the OSD. Gain 9 (max) and Gain 8 (about 50% of Gain 9) are well behaved. At Gain 7, a clipping phenomenon begins to emerge, becoming more severe at Gains 6 down to 1. It is likely that, for occultation observations, the camera will be operated at max gain (9) or one step down (8), but below that, one should be aware of the response distortion that will occur. For that reason, the following response curves were recorded to provide a visual of what is going on.

Test conditions:

The 'bar' target is used for data collection

WDR is off.

NS (Night Shutter) 1

All curves (except Figure 5) plotted at same vertical and horizontal scale

Figure 5 has a compressed x axis for ease of viewing ramp.

LED current was set so that at max gain, pixel saturation was reached and then held constant

For the following curve, the peak LED current has been set to cause pixel saturation to occur, mimicking the type of observation situation where a gain reduction might be desired.

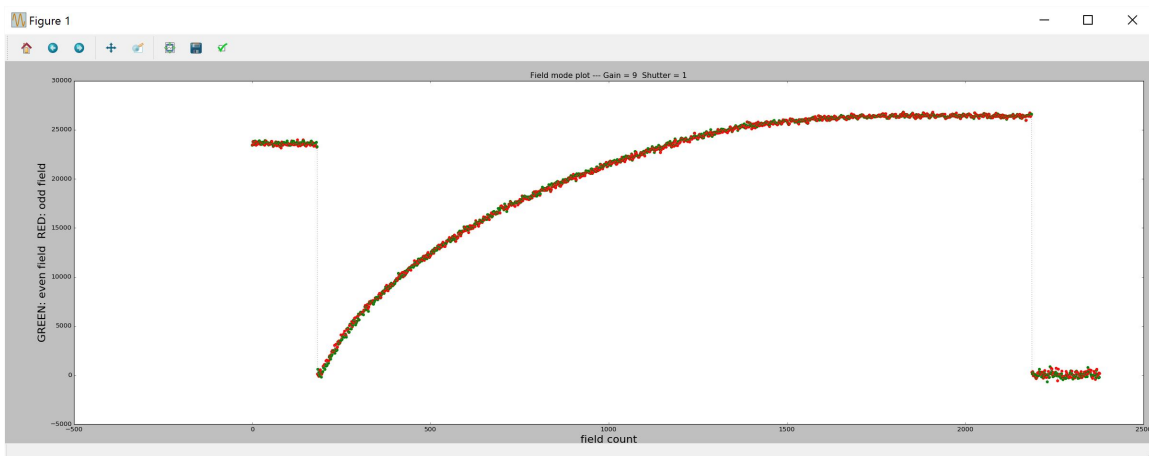


Figure 1 was run at Gain 9 (OSD setting). The photon gain is  $G = 3.45 \text{ e}^-/\text{ADU}$  (see appendix)

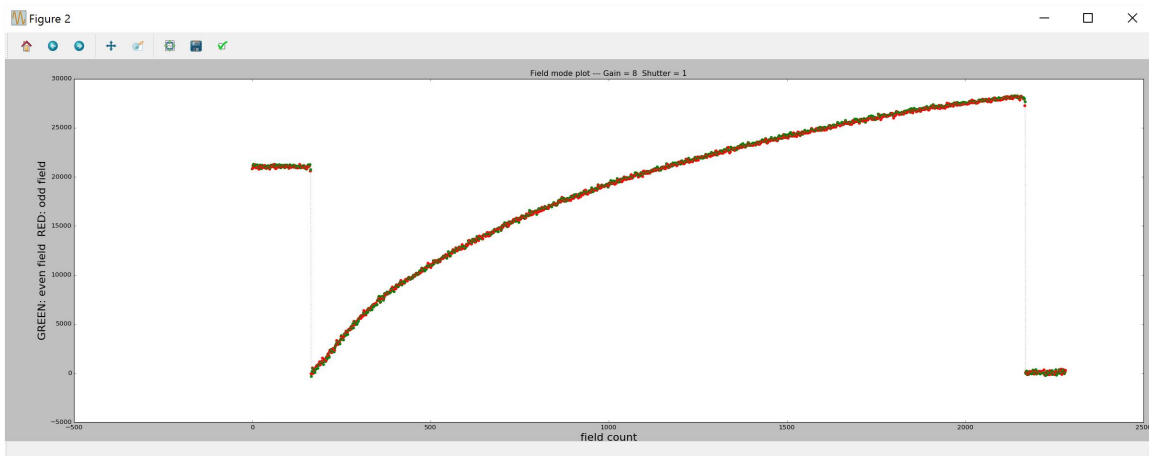


Figure 2 was run at Gain 8. The photon gain is  $G = 6.94 \text{ e}^-/\text{ADU}$

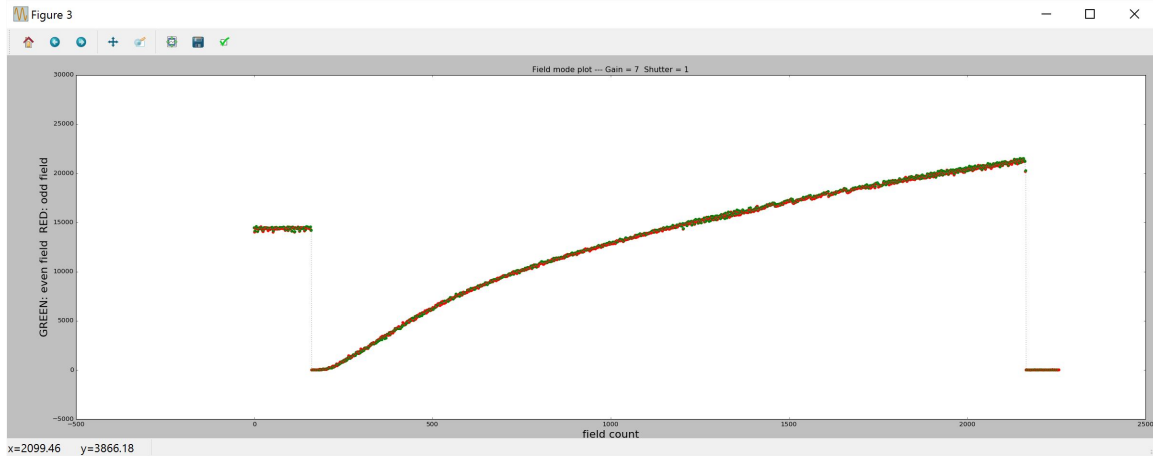


Figure 3 was run at Gain 7. At this setting, the onset of 'clipping' at low light levels begins to appear. This amount of 'clipping' is probably tolerable, making this a usable gain level. The photon gain is  $G = 10.38 \text{ e}^-/\text{ADU}$

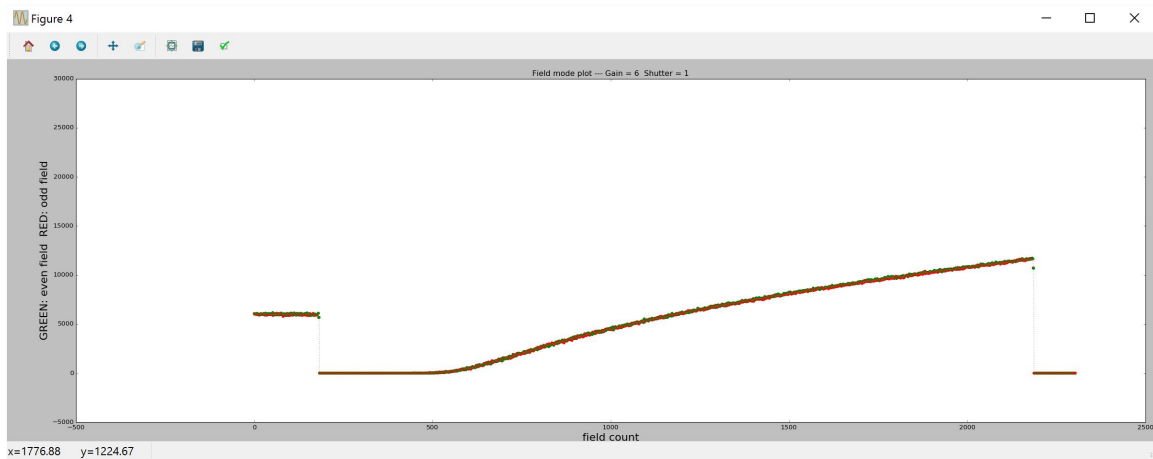


Figure 4 was run at Gain 6 and shows significant clipping. The photon gain is  $G = 18.71 \text{ e}^-/\text{ADU}$ .

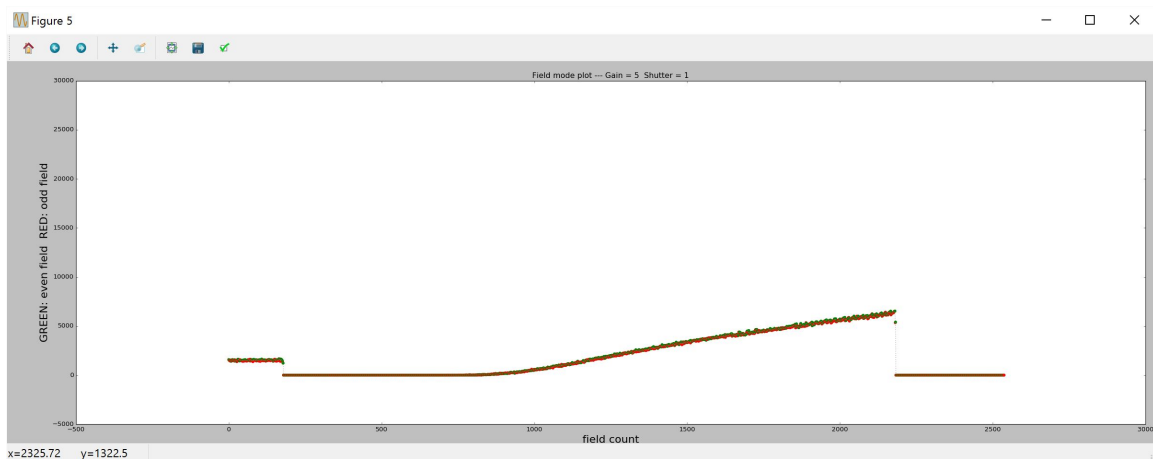
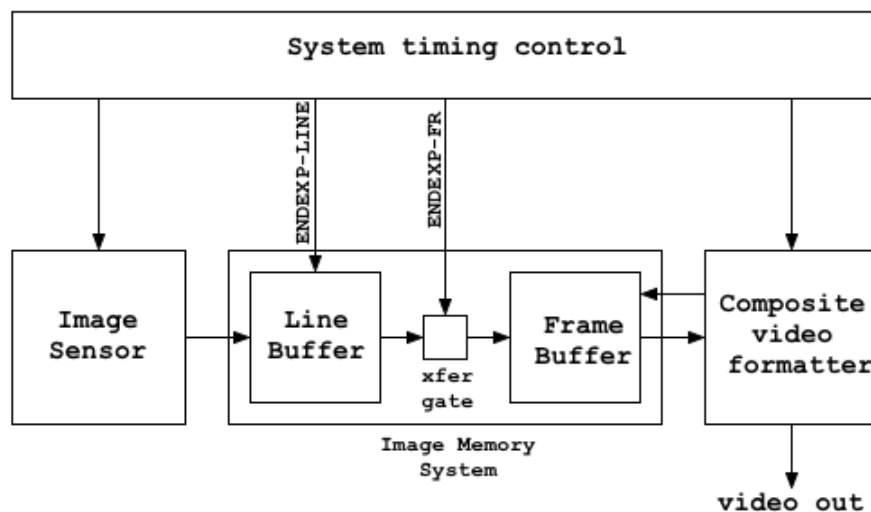


Figure 5 was run at Gain 5 and exhibits even more clipping. A G value was not measured for this Gain setting. The trend of increased 'clipping' continues for lower Gain settings but were not recorded for this report.

### Section 3: Rolling shutter/readout discussion

I was unable to find a detailed description of the internal timing of this camera --- there is no published block diagram or theory of operation that I know of; unfortunately, this is the normal practice for manufacturers; they have a need to protect their proprietary designs. However, by making a couple of special measurements with an oscilloscope and perusing the descriptions of how CMOS image sensors usually work, I have cobbled together a hypothetical model of how this particular camera works that fits/explains all the measurements undertaken so far. The proposed model is educated guesswork and was conceived without input from the manufacturer. In this section I describe this model verbally and show the 'evidence' used to support the model.

The hypothesized block diagram is shown below:



The presence of an Image Memory System is deduced from the integration capability of the camera. As we have become accustomed to with integrating cameras, when some number of field times are set as an integration interval, the field containing the accumulated light is repeatedly output in the composite video stream while a new integration interval is underway. That mechanism requires a memory element, such as the Image Memory System shown in the block diagram, from which the pixel values can be repeatedly read and formatted into the composite video output. In Section 4, we will see what such an integrated output looks like.

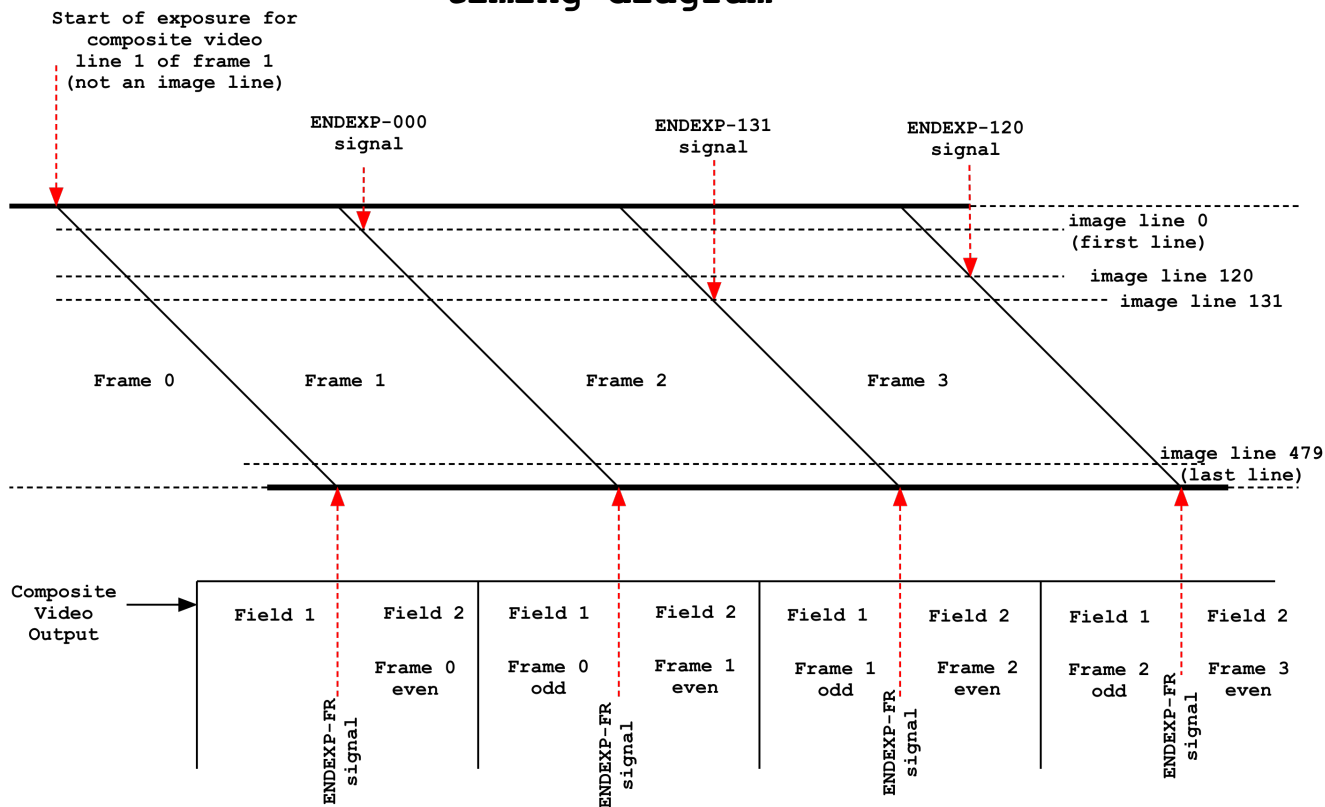
The readout sequence of the sensor is very simple: each pixel is read in sequence, starting at the upper left corner, proceeding horizontally along a line, and then moving to the left side of the next line down. When a pixel is read, its accumulated charges are cleared from the photo-diode and so begins a new accumulation interval for that pixel. As a result, each pixel (and each line) has a different start time for light accumulation. **It is important to understand that each line is read in sequence, not all odd lines, then all even lines. So the accumulation start time for line n+1 is only about 65 microseconds later than line n.**

The timing diagram for this activity, keyed to the above block diagram is shown below. To simplify the model, it is assumed that all of the pixels from a given line are recorded at the same time, triggered by the signal labelled ENDEXP-LINE where LINE is a line number, which occurs at the end of each line scan. The notation used here has been chosen to align with that used by Steve Preston in a separate, in-depth discussion of Astro timing that he will be publishing shortly. Each time a line accumulation interval is complete, it is written into the 'Line Buffer' and cleared, thus starting the subsequent light accumulation interval. When the last line has been recorded, an ENDEXP-FR signal occurs, marking the end to the accumulation interval for all lines in that frame and writing them all en masse to the 'Frame Buffer' where the 'Composite video formatter' has access to them and can then generate the interlaced composite video signal. Note that there are leading and trailing lines in a composite video signal that do not carry image information. For NTSC, there are 525 lines in the composite video signal,



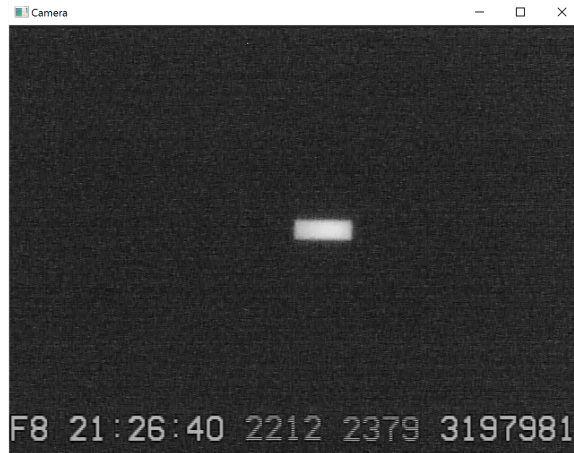
but only 480 of them carry image data. For PAL, it's 576 image lines out of 625 total lines.

## RunCam Astro exposure timing diagram



The CCD sensors that we are more familiar with work quite differently: all odd line pixel charges are simultaneously transferred to a covered charge storage/readout region (thus halting light accumulation) and a new accumulation interval is started that has the same start time for all odd field pixels. Then, a field time later, all even line pixel charges are simultaneously transferred to the charge storage/readout region and a new accumulation interval is started that has the same start time for all even field pixels.

To compound the potential pitfalls of this different timing system, recall that a video time inserter, such as the IOTA VTI, overlays a visual timestamp on each field that shows a GPS accurate timestamp (with 0.1 millisecond precision) that specifies the start time of the composite video output for that field. That implies that there is a field time difference between the accumulation interval for odd fields versus even fields. **That's correct for CCD sensors, but not for CMOS sensors with 'rolling readout'.** Great care must be taken in applying timestamp values as part of a calculation to determine the actual accumulation start time. The camera delay correction equations that have been derived for this camera (presented in Section 5) take this into account (if the user processes the light curve in field mode), so the user doesn't have to worry about this interpretation difficulty, but if manual timing by visual inspection of the video is attempted, this can/will be a source of systematic error. The evidence that I offer to support the proposed operation model for the Astro comes from interpreting two oscilloscope pictures of the composite video waveform response to a field duration light pulse and then a two field duration light pulse. The picture below shows the image used for this test.

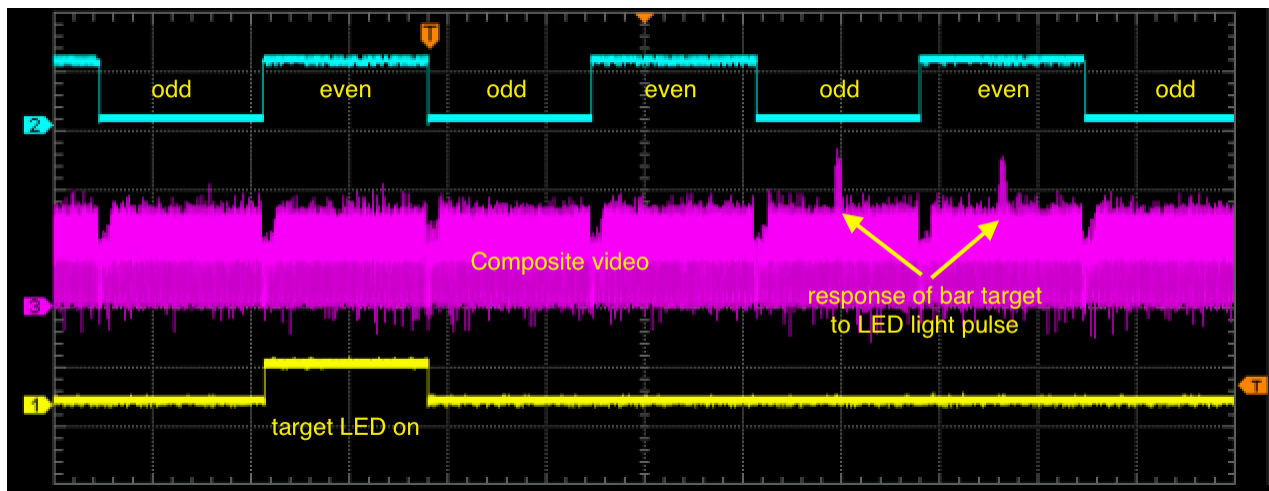


In order to proceed with this study, it was first necessary to verify that the convention for labelling a field odd versus even was well defined. In ArtStar, the convention is well defined/specified by using as a reference an industry standard composite video synch pulse extractor chip, the LMH1980. A synch pulse extractor chip is a device used to analyze the composite video output according to published standards. Of importance here is that there is an output pin on that chip that indicates field parity. To specify a convention, one must assign a meaning to that pin when it is in a high state (+5 volts) or a low state (0 volts). **The convention used in ArtStar is that when this parity pin is high, the field is even; when low, the field is odd.** Field 1 and Field 2 notation is also commonly used to identify fields and is particularly useful in describing the 'on-the-wire' sequence of fields. With the ArtStar/IOTA-VTI semantics, the even field corresponds to Field 1, while the odd field corresponds to Field 2. It is (usually) the case that frame grabbers assemble Field 1 and then Field 2 into a 'frame'.

A careful set of tests was performed and discussions had with the designers of the IOTA VTI to ensure that the selected ArtStar parity convention was the same as the IOTA VTI parity convention.

I also confirmed that Limovie is consistent with this convention as well. So, if ArtStar identifies a field as 'odd', all other elements in the chain will 'agree'.

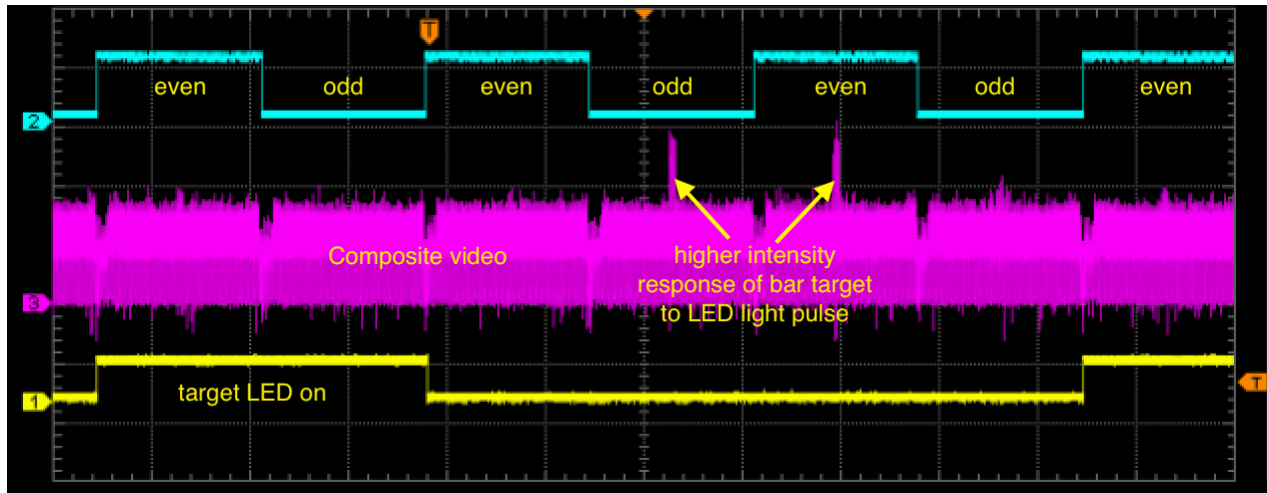
In the picture below, the blue trace at the top shows the voltage at the field parity pin (OEOUT) of the synch pulse extractor chip (LMH1980). The middle trace shows the composite video output. The response to the 'target' being pulsed 'on' appears near the middle of the third and fourth fields after the pulse. The spikes are near the middle of each field because the 'bar target' is physically located near the center of the image.



If this had been a CCD sensor, only even lines would have been 'lit' and there would have been no odd field

composite video 'spike'. But with the CMOS sensor, because the target is only about 12 lines in height, lines  $n$  through  $n+12$  'see' the same light and so appear in the video out as two spikes. This is consistent with the proposed operation model.

The next picture is designed to show light accumulation in action. The difference from the previous picture is that the target LED is on for two consecutive fields, rather than one. It takes close inspection, but the composite video output spikes are now larger than the previous picture. They don't appear twice as large (as one would expect with twice the accumulated light) because of the gamma-compression response curve shown earlier. This demonstrates that each line accumulates light for a full frame time. (The camera was set to Night Shutter 1 which integrates two fields.)



The 'model' explains the above picture as follows:

- Light is accumulated by all lines in the sensor while the yellow trace is high
- The lines are read out sequentially ( $n$ ,  $n+1$ ,  $n+2$ ,  $n+3$ , etc) and 'stored' in the line buffer until the ENDEXP-FR signal occurs
- The composite video formatter extracts the even lines (or maybe the odd lines --- we can't tell for certain and the difference is unimportant because of the small 65 microsecond difference between lines) from the frame buffer and outputs them during the Field 2 of the composite video signal ...
- then the composite video formatter extracts the odd lines (or even) from the image buffer and outputs them during the following Field, Field 1, of the composite video signal.

Additional levels of integration are achieved by accumulating light for 4 field times or 8 field times and loading the line buffer only at the end of an accumulation interval. Similarly, the ENDEXP-FR signal is emitted only at the end of a completed integration interval. The composite video formatter simply keeps extracting and outputting odd/even line groups at the normal time --- it need not 'know' about integration setting/activity.

## Section 4: Integration block framing

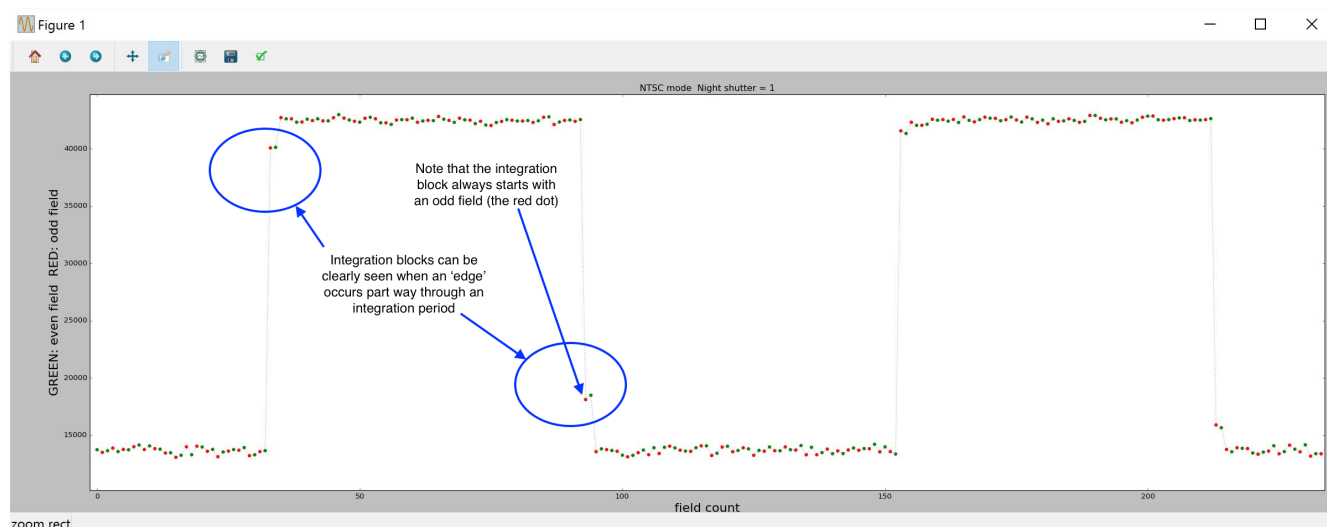
In Section 3, I discussed the convention used by ArtStar to label fields as either odd or even. That convention was selected to be consistent with the convention used by the IOTA VTI (which is an important part of the test system) and Limovie (which is used when timestamped light curves are needed --- such light curves are the critical data input used to measure camera delay corrections). Without such a well defined and consistently applied convention, one could not depend on plots like the ones shown in this section.

The light curve shown below was produced by summing the pixel values in a measurement aperture positioned within the bar target, which was then toggled on and off. Pixel sums from odd lines and pixel sums from even lines were computed separately. In the plot below, odd field sums are plotted using red dots; even field sums are plotted using green dots.

The intermediate values circled in the plot are caused by an illumination change part way through an integration period.

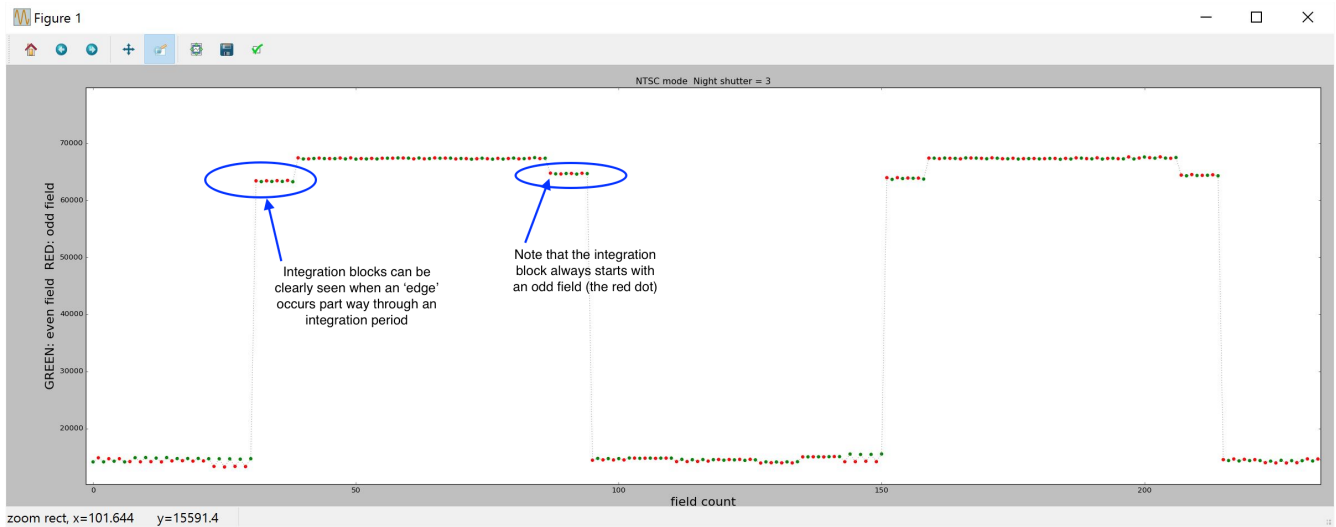
There are three points to be made from this plot:

- We can see the operation of a 2 field integration (this is the Night Shutter=1 setting). It shows up as a pair of repeated values in the output, most obvious in the circled points.
- The 'phasing' is that an integration group (the repeated pattern most clearly seen with the intermediate values produced at illumination transitions) always starts with 'odd'. Here we have an odd-then-even sequencing of associated field values
- Light curve processing software (like Limovie) displays fields in an even-then-odd sequence when run in frame mode (this is actually up to the frame grabber, but even-then-odd is all that I have seen in practice). If that were done with the plot below, the result would be the creation of two intermediate values from the actual single intermediate point. This is why it is necessary to operate Limovie in 'field mode' when producing light curves for analysis by R-OTE or AOTA. Those programs then can reassemble the fields in the proper sequence and so avoid producing incorrect values at occultation D and R events.



The plot below shows what the field by field plot looks like when 8 field integration (Night Shutter = 3) is in use:



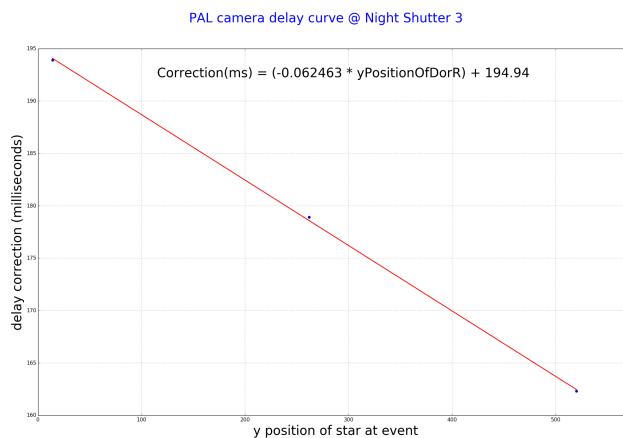
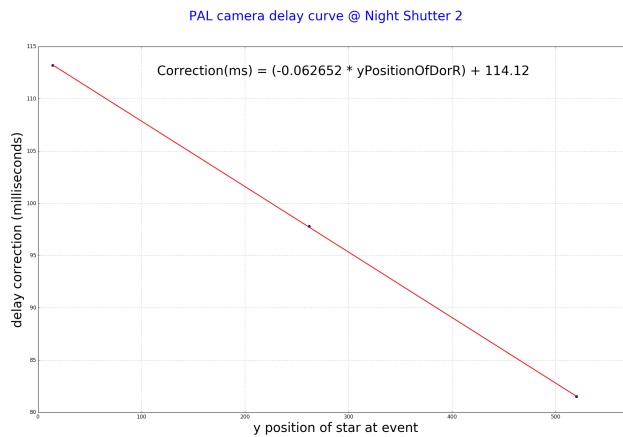
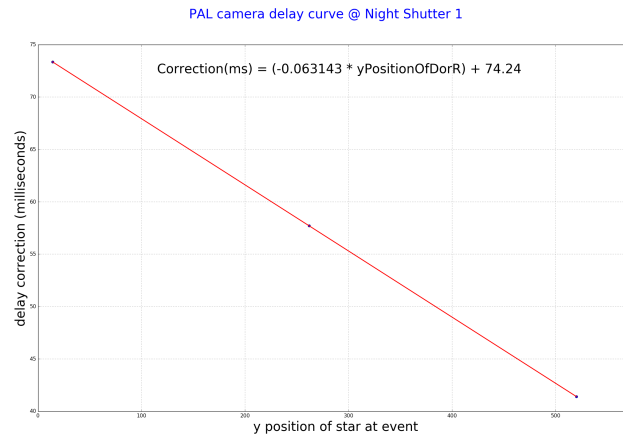


The plot for 4 field integration (Night Shutter = 2) would look similar except that the 'integration group' would have four points. An odd field remains the starting field.

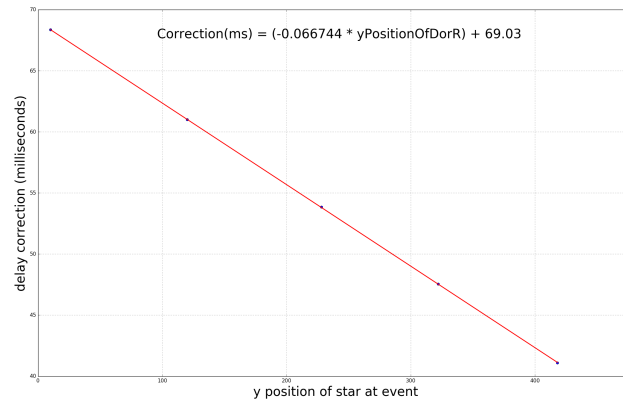
This pattern persists when the camera is operated in PAL format. In short, PAL integration and 'phasing' is identical to NTSC integration and phasing.

## Section 5: Camera delay correction formulas

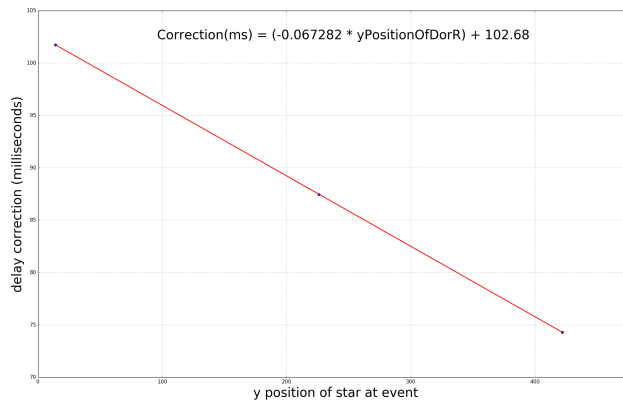
The methodology for computing the camera delay correction equations is complex and is explained in its own appendix. Here we simply present the equations and emphasize that in order to apply these equations, not only is it necessary to note/record the y position of the image at the D or R event, but the light curve has to be processed in field mode and subsequently block integrated by R-OTE. This is because the equations take into account and correct for the way timestamps are registered for odd and even fields.



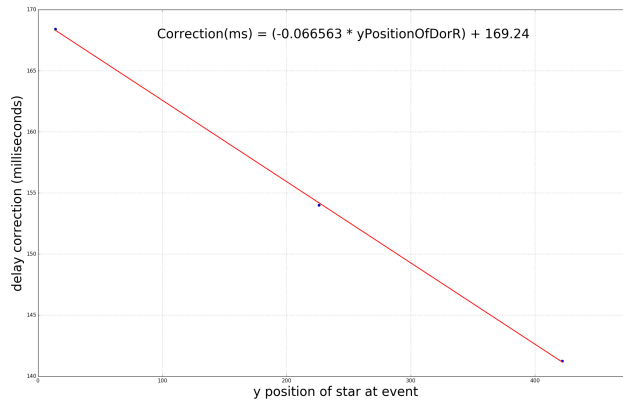
Astro 2 NTSC Gain=9 NS=1 camera delay corrections



Astro 2 NTSC Night Shutter 2 camera delay corrections



Astro 2 NTSC Night Shutter 3 camera delay corrections



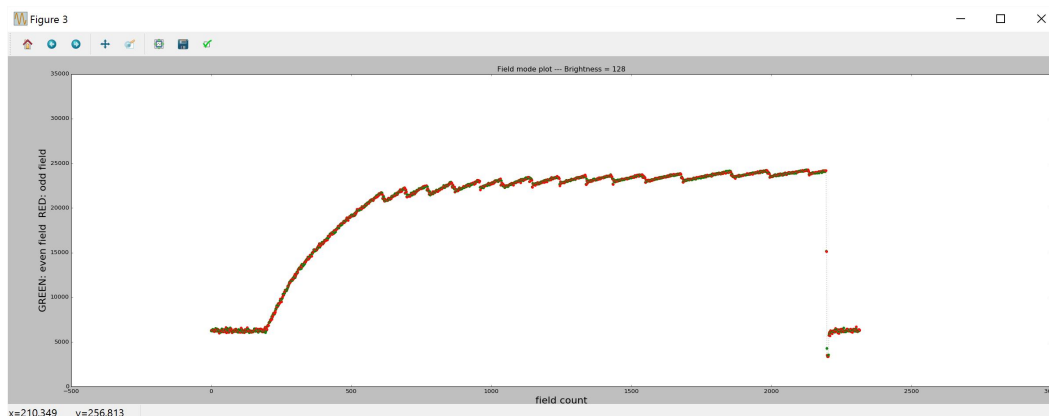
## Section 6: Automatic Gain Control (AGC)

The prototype Astro has an automatic gain control (AGC) that cannot be disabled. RunCam has been asked to provide a way to disable the AGC activity, but it is not known at this time whether this can be done and, if it can be done, when it will be done.

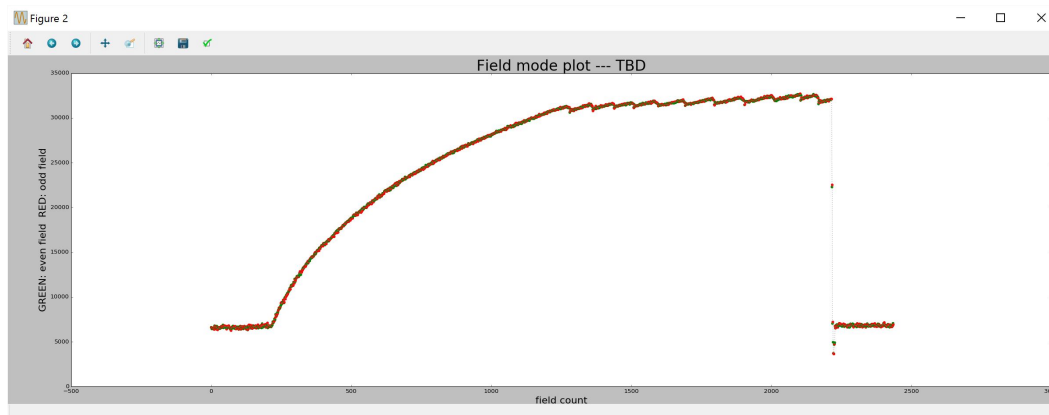
Other cameras used for occultation observing have had a similar feature, so this is something that has been seen and dealt with before. It is likely that for an observation that is basically dark sky, the AGC will consider the scene 'too dark' and set the gain to maximum to produce the brightest possible image. But if a bright object, like the moon, is part of the image, this will certainly cause the AGC system to reduce the camera gain and thus potentially interfere with occultation observations.

Below are some plots showing the AGC taking control and are presented to show that setting the Brightness control to its maximum possible value will beneficially change the point at which the AGC begins to affect the gain downward from maximum.

For this test, ArtStar Lite was used. This device presents the camera with uniform intensity 'flats' so that all pixels are uniformly lit. The illumination is linearly ramped from zero to a maximum value known to saturate pixel response when only a few pixels are exposed to this level of illumination. But when all pixels are illuminated in this way, the AGC system will become active and decrease the camera gain in steps. The plot below shows that when the illumination ramp reaches a certain level, the AGC begins to reduce the gain in discrete steps to avoid a totally saturated image. For this plot, a Brightness of 128 was set using the OSD. (The default value of Brightness is 34.)



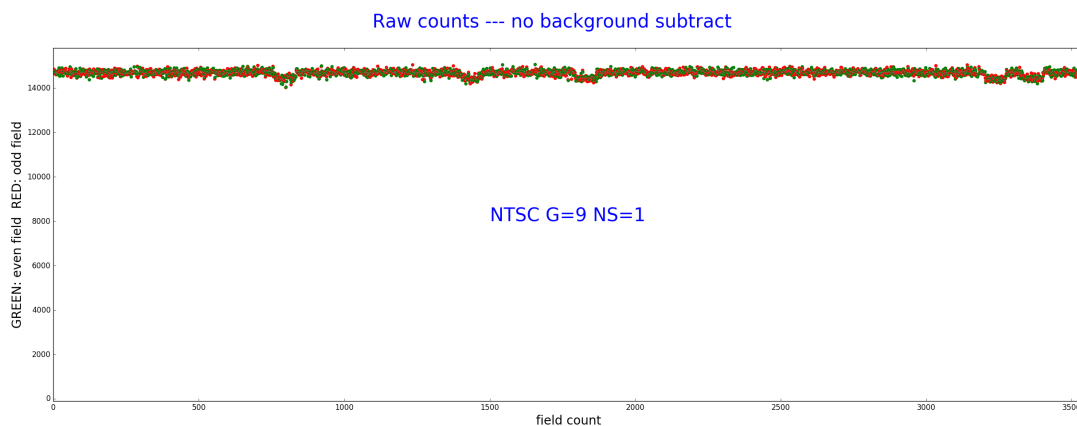
In the above plot, the AGC 'kicks in' fairly early. But if the Brightness is at its max value of 248, we get:





So, for that reason, in the Recommended practices section, it is suggested that Brightness be set to 248 and a SAVE & EXIT performed to make this the default setting at power up.

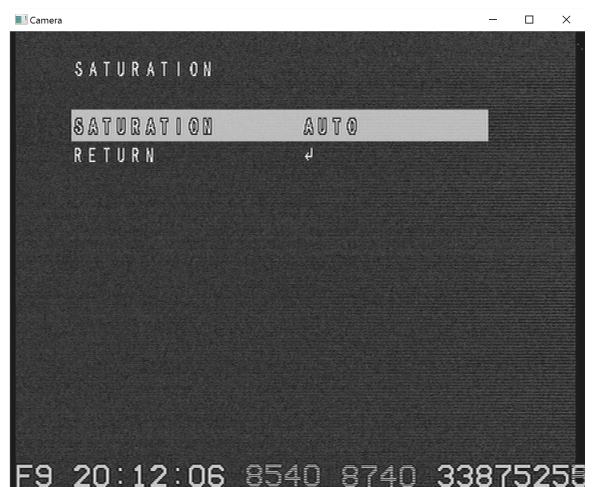
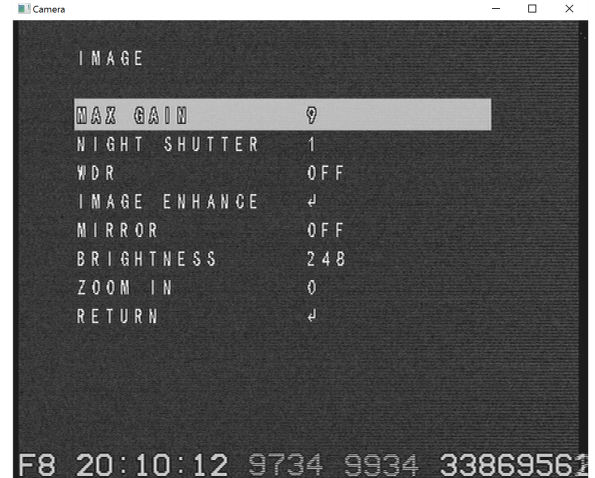
The following artifact, possibly associated with AGC activity, was noted when the camera was operating in NTSC format:



The small 'dips' in what should have been a constant intensity measurement may be associated with AGC activity. These 'dips', while visible when viewed in a very high signal to noise desk environment (like ArtStar with bar target), would be unnoticeable in the presence of typical scintillation noise in real sky observations.

## Section 7: Recommended practices

The OSD settings used in this report are shown below. In typical use, after the video standard has been set, only MAX GAIN and NIGHT SHUTTER settings are likely to be changed. After each such change, perform a SAVE&EXIT, then power cycle the camera. This ensures that field phasing will always be the same.



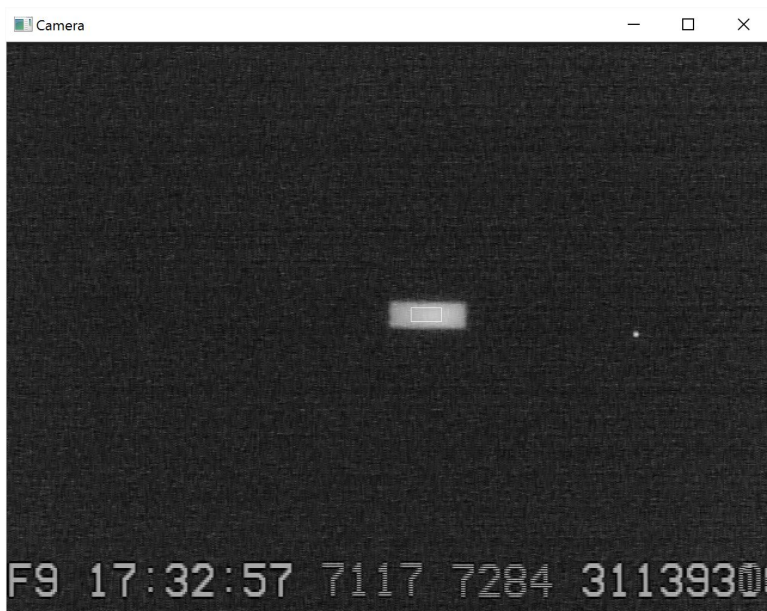
### Recommended practices:

- Process videos in field mode
- Record the y position of the occulted star at D and R events (needed for camera delay correction)
- Do a SAVE&EXIT and subsequent power cycle for all OSD setting changes

## Appendix: Linearization of camera response

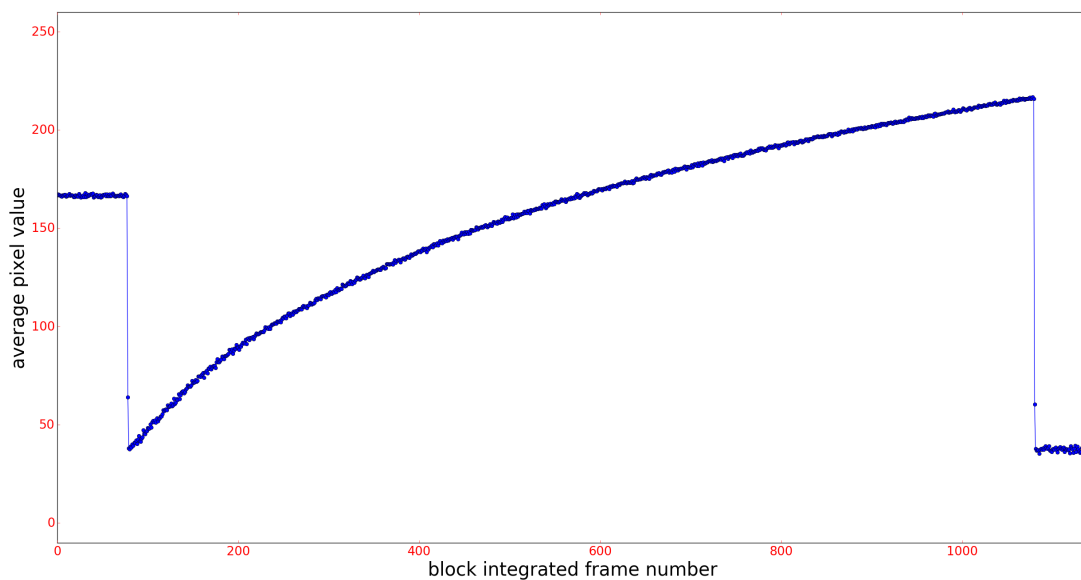
One measurement that I like to make on a camera is its photon gain; that is, the number of absorbed photons in a pixel needed to produce a one unit change in the pixel ADU (analog to digital unit) value, usually expressed as  $e^-/\text{ADU}$ . But to make this measurement, the camera response must be linear. For that reason, I undertook the task of linearizing the camera response --- that is, un-doing the built-in gamma-compression.

The following bar target was used to obtain a low noise response curve.

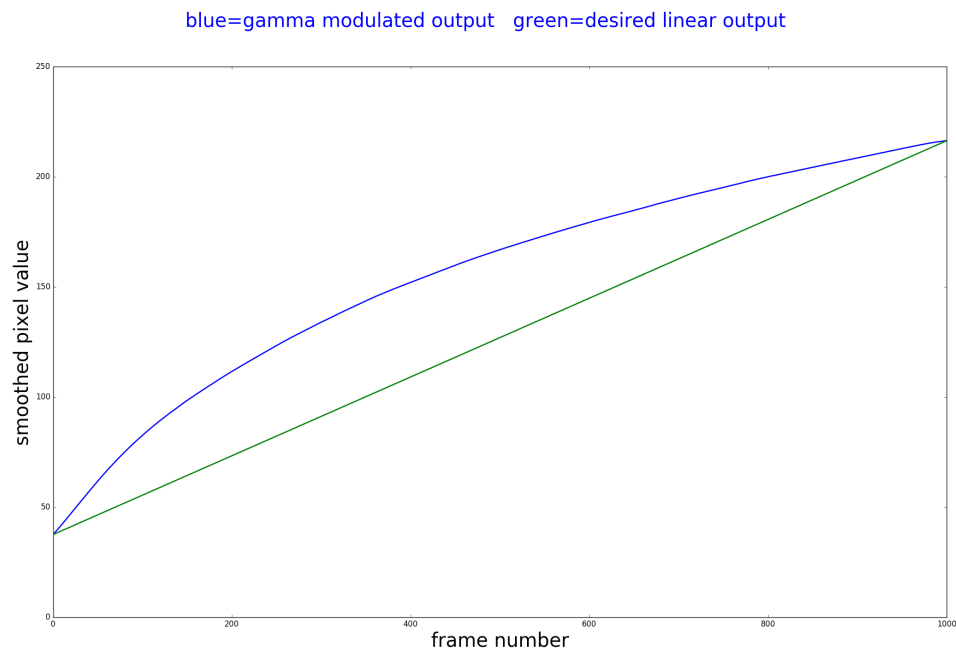


This is the camera response (PAL format, NS 1, Gain 9) to a linearly increasing illumination ramp.

Calibrating ramp --- pixel value versus linear illumination ramp



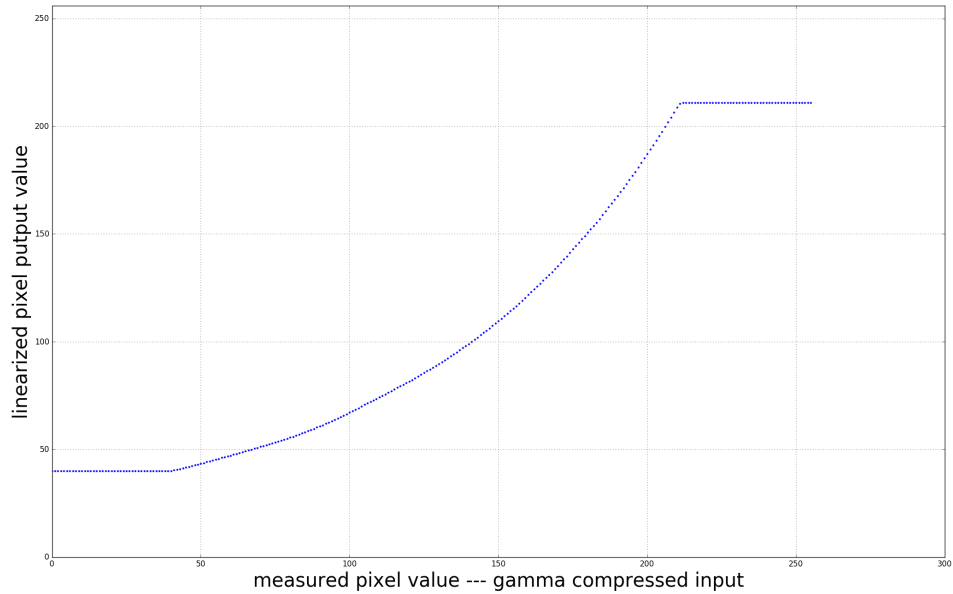
The response curve shown above was smoothed with Savitsky-Golay filter using 201 points and a 2<sup>nd</sup> degree function. The 'filter' operates by fitting (least squares) a parabola to 201 points at a time. The value at the center of the fitted parabola is taken as the smoothed value. The 'fit' position is slid from one extreme of the data to the other. End positions are dealt with by using the last parabola that involved 201 points. The blue curve in the plot below shows the smoothed response curve. Also plotted is the desired linear output.



Next, at each integer value of the smoothed gamma response curve, we compute the corresponding value (now a fractional number) from the linear (green) curve. That creates a transfer function in table form that can be applied to each pixel as a post-processing step. The plot below shows what this transfer function looks like.

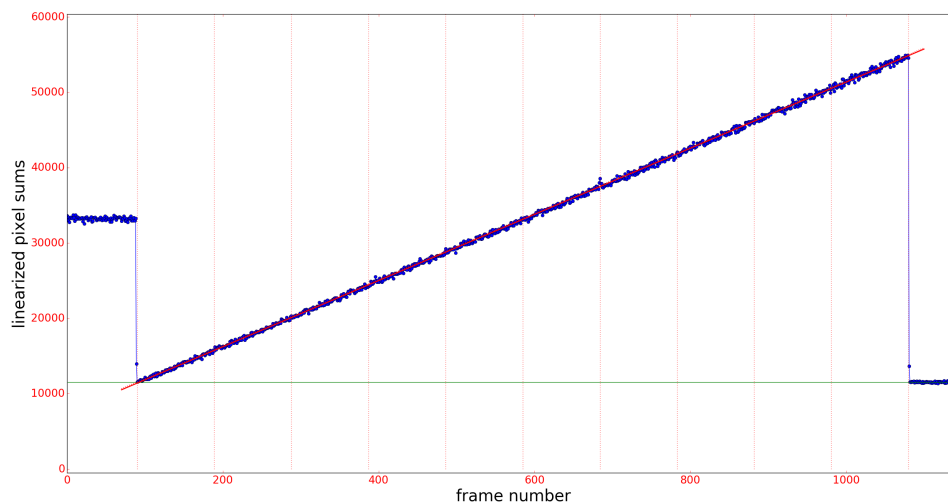


RunCam Astro linearization transfer curve --- Gain=9

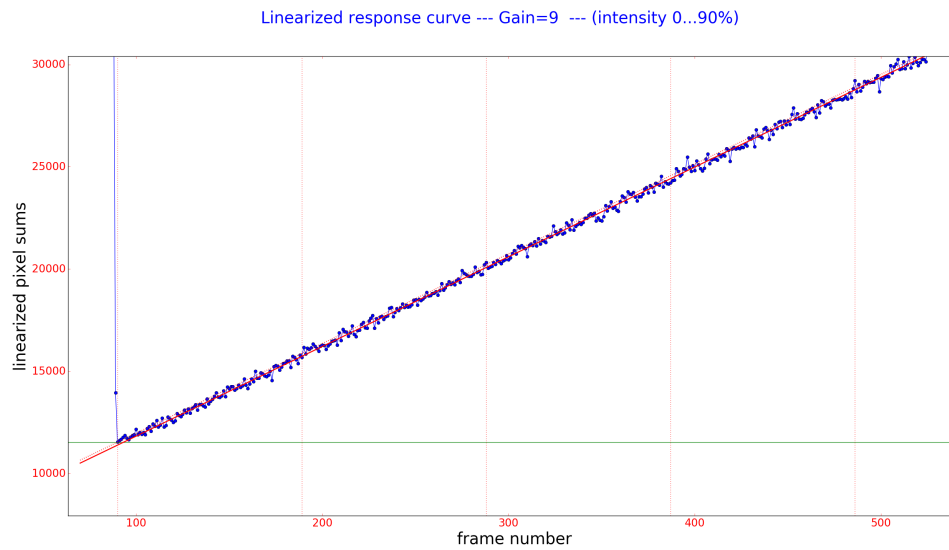


When that transfer function is applied to a subsequent response curve, the resulting response is now quite linear. In the plot below, the peak current was adjusted so that only the first 90% of the camera response is explored. There is a slight bending in the final 10% of the response range that would have detracted from seeing the excellent fit obtained over the majority of the operating range --- that was the only purpose in showing a reduced operation range.

Linearized response curve --- Gain=9 --- (intensity 0...90%)



Below is an expanded view of linearized response curve (so that behavior at zero can be better seen).

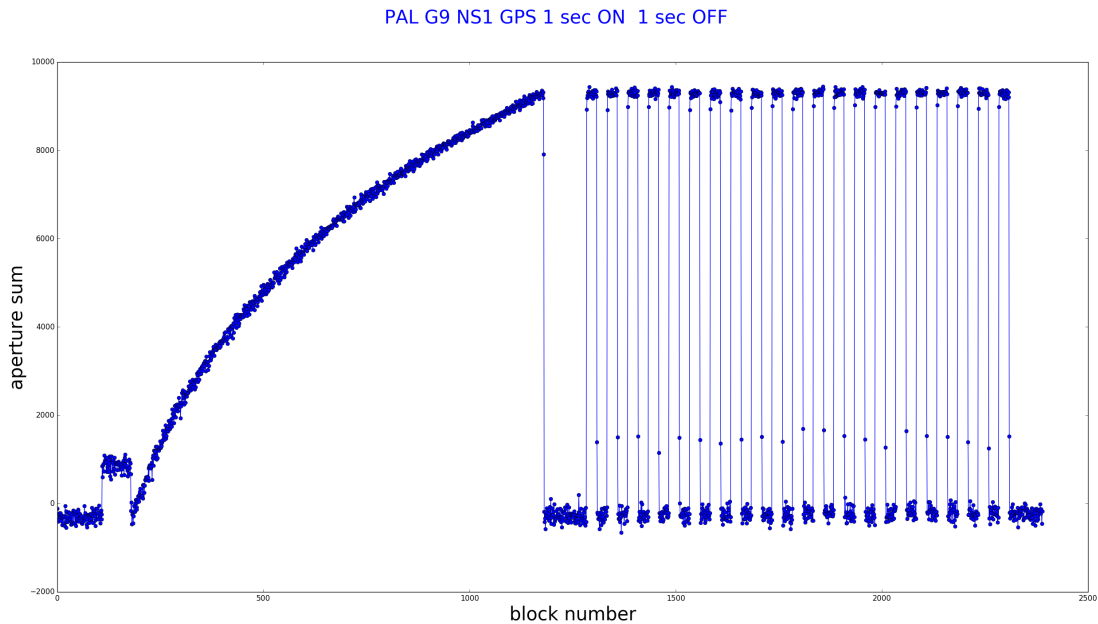


Later testing showed that only the single transfer function shown above was needed to linearize the output from two different cameras, each operating in either PAL or NTSC format, at any of the three integration levels, and Gains of 8 or 9 and even 7 was well handled.

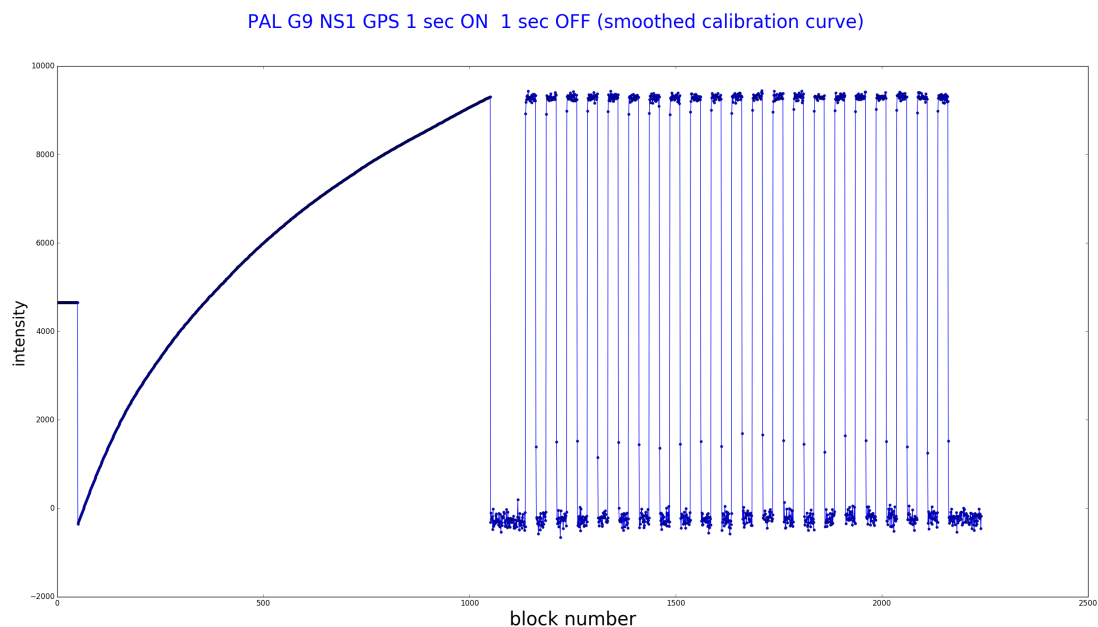
Unfortunately, the frame grabber characteristics affect the transfer function, so it is likely that a different transfer function would be needed for each frame grabber, although it's possible that only different models of frame grabbers would need to be considered.

## Appendix: Methodology for camera delay measurement

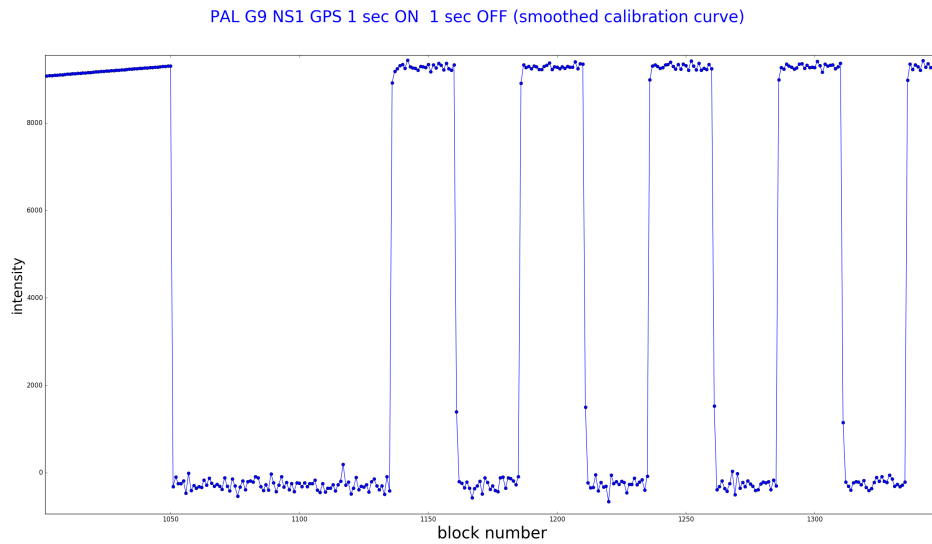
The basic idea for calculating camera delay corrections is to record a video with GPS accurate timestamps overlaid. The video begins with a response curve (linear illumination ramp) that will later be used to enable correct sub-frame timing (in spite of the non-linear response curve) on the intermediate points that occur when a series of 1 second on, 1 second off pulses are recorded that have their edges occurring within a few microseconds of the 1pps output of a GPS chip. The recording below was produced by field processing the video (using Limovie), then block integrated by the test software to get the plot shown below ...



The linear illumination response curve is smoothed with a Savitsky-Golay filter (201 points, degree 2), so the above recording is transformed to this ...



Here is an expanded view of the above plot --- the intermediate intensity points on the GPS timed pulses are easier to see here ...



Next, with some operator guidance, the intermediate points that show up because the GPS 1pps edges usually occur part way through a light accumulation period are extracted and analyzed. A result table like that shown on the next page results.

Limovie file: C:/Users/Bob/Dropbox/NE Astro report/PAL delay/pal-g9-ns1-x338-y248.csv

Comments from Limovie file:

"FileName : C:\Users\Bob\Dropbox\NE Astro report\PAL delay\pal-g9-ns1-x338-y248.avi"  
"Video System : PAL , FrameRate=25.00"

D and R transition points analyzed with PAL timing. Frame time = 0.0400 seconds

R @ 85 =	8921.0	frame time: 31.0546	edge time: 31.0578	edge delay: 0.0578	f: 0.92
D @ 110 =	1389.0	frame time: 32.0546	edge time: 32.0576	edge delay: 0.0576	f: 0.07
R @ 135 =	8909.5	frame time: 33.0546	edge time: 33.0579	edge delay: 0.0579	f: 0.92
D @ 160 =	1501.0	frame time: 34.0546	edge time: 34.0578	edge delay: 0.0578	f: 0.08
R @ 185 =	8986.5	frame time: 35.0546	edge time: 35.0573	edge delay: 0.0573	f: 0.93
D @ 210 =	1523.5	frame time: 36.0546	edge time: 36.0578	edge delay: 0.0578	f: 0.08
R @ 235 =	8987.0	frame time: 37.0546	edge time: 37.0573	edge delay: 0.0573	f: 0.93
D @ 260 =	1148.5	frame time: 38.0546	edge time: 38.0571	edge delay: 0.0571	f: 0.06
R @ 285 =	8976.5	frame time: 39.0546	edge time: 39.0574	edge delay: 0.0574	f: 0.93
D @ 310 =	1486.0	frame time: 40.0546	edge time: 40.0578	edge delay: 0.0578	f: 0.08
R @ 335 =	8916.0	frame time: 41.0547	edge time: 41.0580	edge delay: 0.0580	f: 0.92
D @ 360 =	1445.0	frame time: 42.0547	edge time: 42.0578	edge delay: 0.0578	f: 0.08
R @ 385 =	8927.5	frame time: 43.0547	edge time: 43.0579	edge delay: 0.0579	f: 0.92
D @ 410 =	1362.5	frame time: 44.0547	edge time: 44.0576	edge delay: 0.0576	f: 0.07
R @ 435 =	8898.0	frame time: 45.0547	edge time: 45.0581	edge delay: 0.0581	f: 0.92
D @ 460 =	1455.5	frame time: 46.0547	edge time: 46.0578	edge delay: 0.0578	f: 0.08
R @ 485 =	8959.5	frame time: 47.0547	edge time: 47.0576	edge delay: 0.0576	f: 0.93
D @ 510 =	1508.5	frame time: 48.0547	edge time: 48.0579	edge delay: 0.0579	f: 0.08
R @ 560 =	1401.0	frame time: 50.0547	edge time: 50.0577	edge delay: 0.0577	f: 0.07
D @ 585 =	8933.5	frame time: 51.0547	edge time: 51.0578	edge delay: 0.0578	f: 0.92
D @ 610 =	1696.0	frame time: 52.0547	edge time: 52.0583	edge delay: 0.0583	f: 0.09
R @ 635 =	8999.5	frame time: 53.0547	edge time: 53.0573	edge delay: 0.0573	f: 0.94
D @ 660 =	1657.0	frame time: 54.0547	edge time: 54.0582	edge delay: 0.0582	f: 0.09
R @ 685 =	8960.5	frame time: 55.0547	edge time: 55.0576	edge delay: 0.0576	f: 0.93
D @ 710 =	1532.0	frame time: 56.0547	edge time: 56.0579	edge delay: 0.0579	f: 0.08
D @ 760 =	1449.0	frame time: 58.0547	edge time: 58.0578	edge delay: 0.0578	f: 0.08
R @ 785 =	8984.5	frame time: 59.0547	edge time: 59.0574	edge delay: 0.0574	f: 0.93
D @ 810 =	1269.0	frame time: 0.0548	edge time: 0.0576	edge delay: 0.0576	f: 0.07
R @ 835 =	8988.0	frame time: 1.0548	edge time: 1.0575	edge delay: 0.0575	f: 0.93
D @ 860 =	1638.0	frame time: 2.0548	edge time: 2.0583	edge delay: 0.0583	f: 0.09
R @ 885 =	8971.5	frame time: 3.0548	edge time: 3.0576	edge delay: 0.0576	f: 0.93
D @ 910 =	1535.0	frame time: 4.0548	edge time: 4.0581	edge delay: 0.0581	f: 0.08
D @ 960 =	1508.0	frame time: 6.0548	edge time: 6.0580	edge delay: 0.0580	f: 0.08
D @ 1010 =	1394.0	frame time: 8.0548	edge time: 8.0578	edge delay: 0.0578	f: 0.07
R @ 1035 =	8943.5	frame time: 9.0548	edge time: 9.0578	edge delay: 0.0578	f: 0.92
D @ 1060 =	1250.5	frame time: 10.0548	edge time: 10.0575	edge delay: 0.0575	f: 0.07
R @ 1085 =	8979.0	frame time: 11.0548	edge time: 11.0576	edge delay: 0.0576	f: 0.93
D @ 1110 =	1517.0	frame time: 12.0548	edge time: 12.0580	edge delay: 0.0580	f: 0.08

D delay: 0.0578 (+/- 0.0003) seconds

R delay: 0.0576 (+/- 0.0003) seconds

==== Timestamp OCR results: largest time step=0.0201 smallest = 0.0199

The numbers of interest are D delay (0.0578) and R delay (0.0576). These are the corrections that must be applied to (subtracted from) the time values reported by R-OTE and AOTA for an event that occurs at the image position used for this particular measurement.

The next step is to repeat this measurement, changing only the y position of the target. At a minimum, y positions from close to the top, near the middle, and close to the bottom were measured. Fortunately, the relationship of time correction to y position is linear, so only the measurements with y at top and bottom are really necessary, but measuring the middle value provides a good check that nothing has gone wrong. One of the NTSC equations was plotted with five y positions as a check that no unexpected behavior was being overlooked.

These values are then plotted with an overlaid least squares fit straight line to get the delay correction equation plots shown in Section 5.

I also measured the time correction needed if the image was near the lefthand edge or the righthand edge. No measurable difference could be determined, which is not surprising as the operation model described in Section 3 predicts only about a 65 microsecond difference --- this small difference is below the noise threshold of the

measurement technique.

## Appendix: Inter-frame interaction test results

Sometimes manufacturers include special processing circuitry to enhance the imaging properties of their cameras. Sharpening of images and edge enhancement are examples of the kind of processing that might be present in the camera electronics.

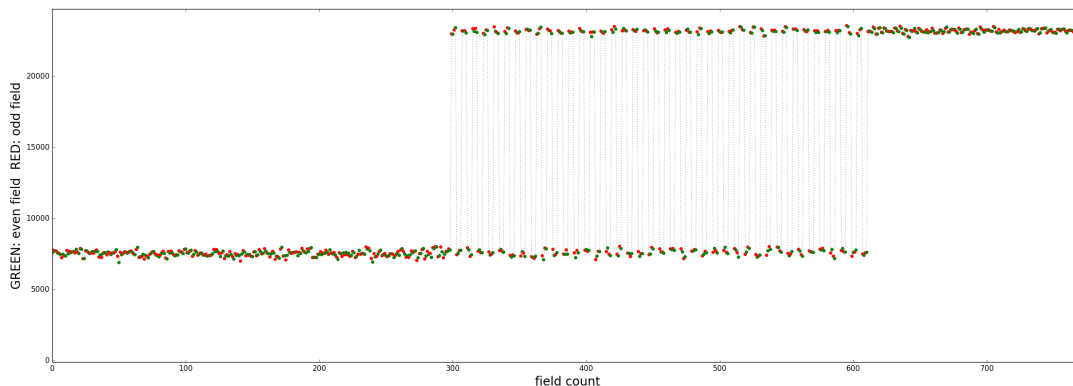
For occultation timing, it is important to know that such processing is absent, can be disabled, or is inactive in the default settings.

The biggest concern arises when the imaging processing involves some algorithm that incorporates a mathematical procedure on a sequence of frames. This has the potential of producing a memory effect where the image in frame  $n$  is modified by the contents of the image in frame  $n-1$  or earlier and sometimes frame  $n+1$ , etc. This has the potential for affecting occultation edge timing.

The RunCam Astro specifications do not indicate explicitly that any such processing is built into the camera, but there is an OSD menu item that refers to 'Image Enhance'. For this reason, it was felt that a test for inter-frame interactions should be carried out.

The test is to turn the target 'off' for a while, then rapidly switch the target illumination between full 'off' and full 'on', then finish with the target full 'on'. If there is no difference between the extreme levels attained in the rapid switching region and the steady 'off' and 'on' regions, and no 'edge' effects --- overshoot; undershoot; rounding --- then there are no memory effects of concern.

The following test of the Astro was run with the default Image Enhance settings for Sharpness|Detail=Auto, Sharpness|Edge=Auto, and Saturation=Auto. It is clear that from the results below that the default settings create no detectable inter-frame memory effects.





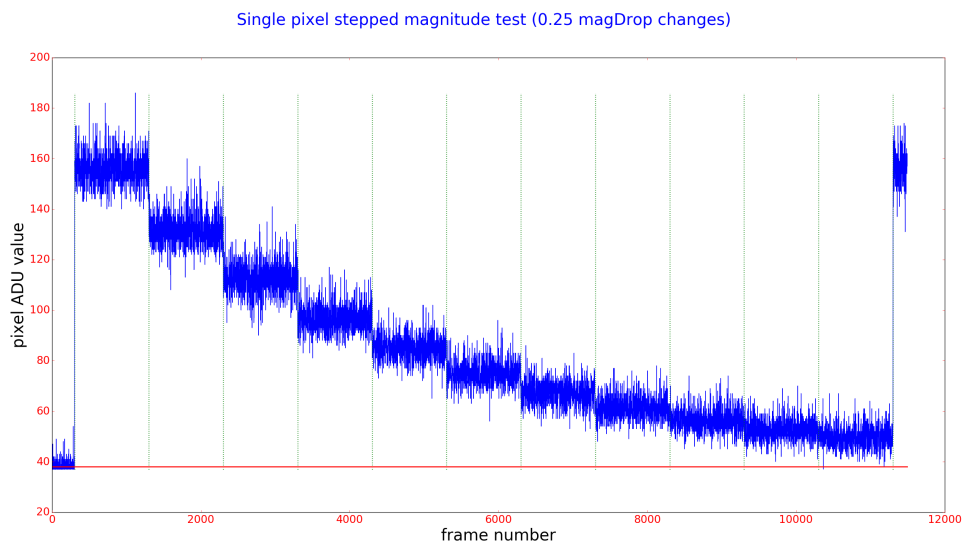
## Appendix: Methodology for determination of camera gain ( $e^-/\text{ADU}$ )

The best way to measure the photon gain of a camera is to make a series of measurements on a single pixel, recording average intensity and variance at multiple intensities. The slope of pixel intensity versus variance then gives us the desired number.

A single pixel measurement is the easiest and most reliable way to measure this parameter because the variance calculation is well determined, which is not the case when multiple pixels are included in the aperture measurements. If multiple pixels are included in the measurement and summed to produce intensity values, for the variance value we will actually get the 'variance-of-the-sums' --- but what we need is the 'sum-of-the-variances'. These two quantities are **only** equal if there is no correlation between the pixels. But this camera, and all others that I have tested, has a strong correlation between neighboring pixels along a line, and also between lines, surprisingly enough. I believe it is likely true that at the photo-diode level, pixel values are uncorrelated. But the readout electronics changes this. It is easy to see how pixel value correlations occur along a line: it's a consequence of slew rate limitations in readout amplifiers; this produces a memory effect which causes pixel to pixel values to depend on each other --- i.e., be correlated --- the amplifier has to slew from pixel  $n$  value to pixel  $n+1$  value --- as this cannot happen instantaneously, the value at pixel  $n+1$  will share a little bit of the value at pixel  $n$ .

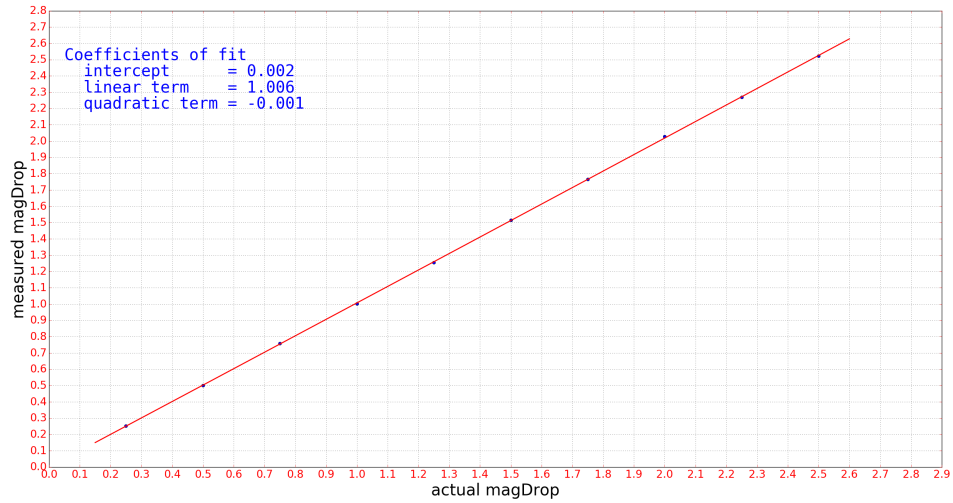
In any case, ArtStar 2 can measure a single pixel and, critically, a previously determined transfer function is available to linearize its response curve.

Here is the plot of a linearized, single pixel, stepped magnitude test. This test conveniently provides a set of 11 intensity and variance measurements as well as showing the potential astrometric performance.



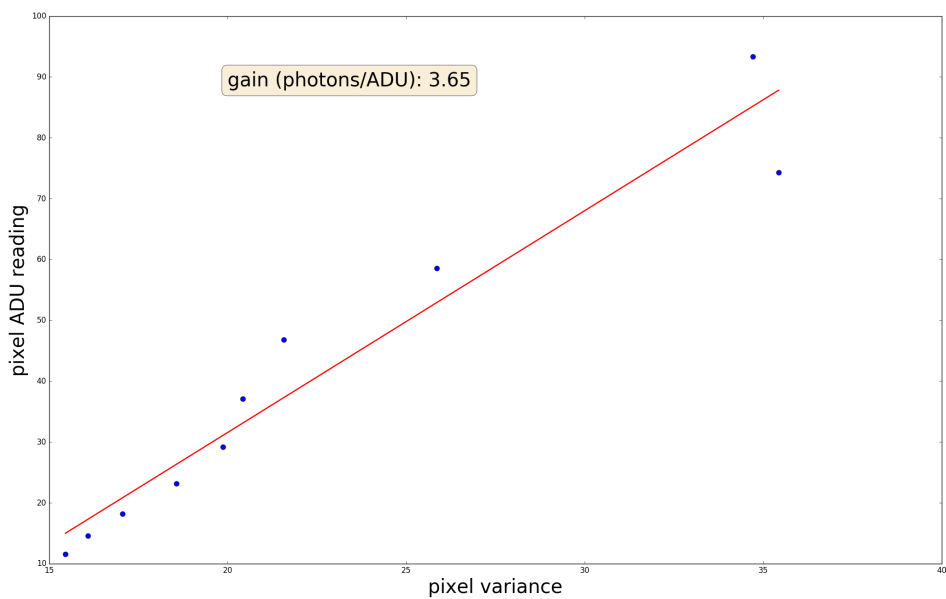
Along the way toward the photon gain plot, it's worth noting how good the astrometric performance of the camera can be, even for a single pixel, so I added that plot as well ...

Single pixel magDrop performance curve



Finally, the plot of pixel intensity versus variance with a least squares straight line fit is ...

Single pixel estimate of camera gain (electron charges/ADU)



and the slope of that line gives the desired photon gain number.

## Appendix: RunCam Astro FOV (field of view)

The picture below (provided by Ted Blank) shows the RunCam sensor (on the left) compared to the PC164CEX-2 sensor (on the right). The obviously different aspect ratio of the two sensors naturally raises the question, “what is the field-of-view of the Astro and is it a function of whether PAL or NTSC is selected?”.



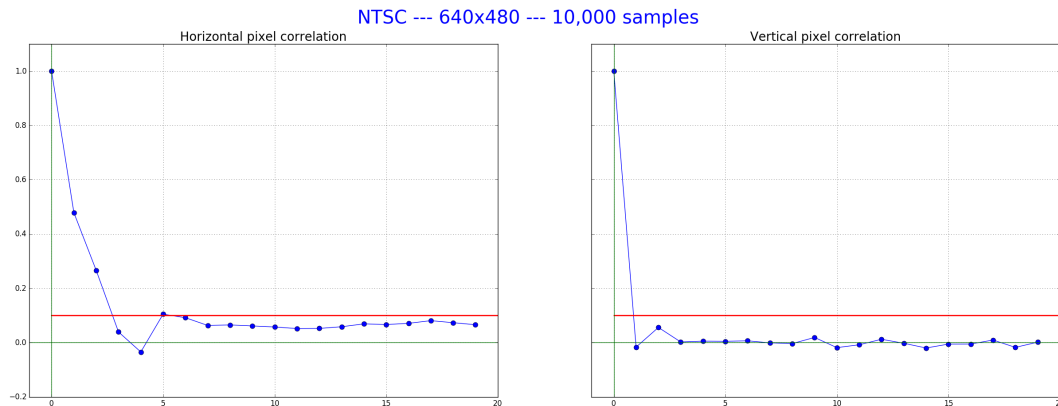
Tony George asked RunCam about this and they responded that the sensor is a 16:9 aspect chip that they convert to 4:3 aspect ratio by removing columns on the left and right.

The pictures below show that the ROI (region of interest) is exactly the same for PAL and NTSC operation modes. RunCam is apparently resampling (down-sampling?) the ROI to produce either a 640x480 or a 720x576 composite video output and so the FOV is equal for PAL/NTSC

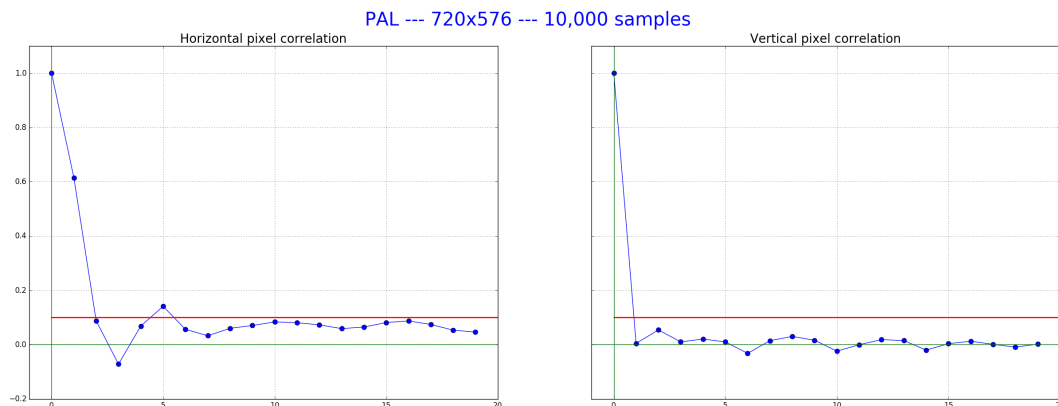


Any time a pixel array of size  $m \times n$  is resampled into size  $j \times k$ , there is the possibility of side-effects from that process. The most likely side-effect is the introduction of adjacent pixel correlations in the output that did not exist in the original pixels.

The ArtStar software has a test that measures pixel to pixel correlations in the horizontal and vertical directions. A set of 20 pixels in each direction, taken from near the center of the image (background noise only) were recorded (10,000 samples) and the auto-correlation function computed. A StarTech SVID2USB2 frame grabber was used. The results are shown below for the NTSC mode. Lag (pixel offset) is plotted on the x-axis; Correlation Coefficient is plotted on the y-axis.



And below are the results for the PAL 720x576 mode...

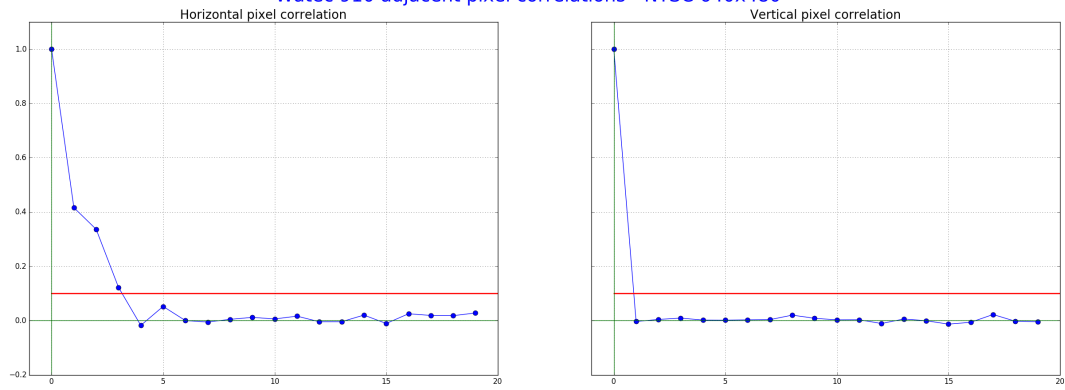


The red line in the above plots is an estimate of statistical significance --- only correlation coefficients above this line are considered 'significant'.

The vertical correlation numbers are flawed in that my software does not take into account the Astro 'framing issue' described earlier that causes the odd and even lines to appear in different frames --- the current software assumes that odd and even lines exposed at the same time appear in the same frame, but they don't in the Astro. However, the correlation coefficients for lag 0, 2, 4, etc are correct. But not for lag 1, 3, 5, etc.

For comparison purposes, the picture below shows the same test carried out using a Watec 910 camera operating in 640x480 NTSC format and the same StarTech SVID2USB2 frame grabber. For this camera, it is likely that there is no pixel resampling going on. Yet the pixel correlation pattern is very similar to that from the Astro.

### Waterc 910 adjacent pixel correlations - NTSC 640x480



These results, coupled with the negative findings in the appendix titled: “Inter-frame interaction test results”, indicate that whatever sampling/resampling/downsampling/image processing may be present in the RunCam Astro, it's not going to affect its usage for occultation timing.