

EXCAM exposure time calculators for spatial resolution element (resel), analog and photon counting

Kevin Ludwick

January 6, 2023

For a given observational sequence and target, the method `calc_flux_rate()` in the class `CGIETC` provides the outputs `flux_rate` (ϕ_{tot} , the integrated photon flux) and `flux_rate_peak_pix` (ϕ_b , the photon flux for the peak pixel). These two are related in this way:

$$\phi_b = \phi_{tot}x, \quad (1)$$

where x is the ratio of the flux from the peak pixel to the integrated flux, which can be found in `sequences.yaml`.

The functions `calc_exp_time_resel()` and `calc_pc_exp_time_resel()` optimize considering the signal-to-noise ratio (SNR) per resel, in the analog and photon-counting cases respectively. We can get ϕ_{resel} , the flux in a resel, in the following way:

$$\phi_{resel} = \phi_{tot}f, \quad (2)$$

where f is the fraction of the integrated flux that is in the resel.

We can utilize the calculator functions in `excam.tools.py`, which can consider the SNR on a per-pixel or multi-pixel basis, by using the average flux per resel pixel, ϕ_{resel}/n , where n is the number of pixels in the resel. The functions `calc_exp_time_resel()` and `calc_pc_exp_time_resel()` use ϕ_{resel}/n as the input flux, and they use the peak pixel's flux ϕ_b as usual for saturation constraints. The functions then output the SNR per resel.

For reference, the SNR per average pixel for the analog case without any correction for cosmic rays is

$$SNR(g, t_{fr}, N) = \frac{\phi_{ij}t_{fr}}{\sqrt{\frac{1}{N\eta_{ij}^2} \left(\frac{\sigma_r^2}{g^2} + F(g)^2 [\phi_{ij}\eta_{ij}t_{fr} + i_d t_{fr} + C] \right)}} \equiv \frac{S}{\sqrt{no^2}}. \quad (3)$$

The numerator is the average signal per pixel S , and the denominator is the average noise per pixel no . ϕ_{ij} is the photon flux per pixel, η_{ij} is the quantum efficiency per pixel (in e^-/photon), t_{fr} is the exposure time per frame, N is the number of frames, g is the EM gain, and $F(g)$ is known as the extra noise factor. (See `etc_snr_v3b.pdf` in the `doc` folder of `EETC` for more details.) The read noise σ_r , dark current i_d , and clock-induced charge C are the values corresponding to a single pixel. These noise sources are not correlated among pixels, so we can write the SNR for n pixels (SNR_n) in terms of the SNR per pixel by summing the total signal and dividing by the total noise:

$$SNR_n(g, t_{fr}, N) = \frac{\sum_i S_i}{\sqrt{\sum_i no_i^2}} = \frac{nS}{\sqrt{n}\sqrt{no^2}} = \sqrt{n} SNR(g, t_{fr}, N). \quad (4)$$

This logic is valid for both analog and photon-counting observations.

The correction factor for cosmic rays (CRs) is applied to the number of frames taken, N . It accounts for the fact that a pixel that is invalidated by a CR hit needs more frames taken to compensate. That correction factor acts as such:

$$N \rightarrow N e^{-Xat_{fr}\ell_{ij}}. \quad (5)$$

X is the rate of cosmic ray hits per unit area, a is the area per pixel, and ℓ_{ij} is the number of pixels per row that must not be hit in order for the pixel in question not to be invalidated due to spillover of electrons from cosmic ray hits. (See `etc_snr_v3b.pdf` in the `doc` folder of `EETC` for more details.) Let us model a resel of n pixels as a circle of pixels, which would have a diameter of $\sqrt{4n/\pi}$. The number of pixels that must be protected ℓ_{ij} is typically fairly large, and so each row of \sqrt{n} pixels in the resel would likely all be protected by ℓ_{ij} . So the correction for one pixel in the above equation is, on average, valid for each row in the resel, with the longest row of resel pixels being of length equal to the diameter. Therefore, to correct for CRs for the circular resel, we need to correct for $\sqrt{4n/\pi}$ rows, so we need

$$N_n = N e^{-X \sqrt{4n/\pi} a t_{fr} \ell_{ij}}. \quad (6)$$

Therefore, for both analog and photon-counting modes, the SNR per resel is

$$SNR_n(g, t_{fr}, N_n) = \sqrt{n} SNR(g, t_{fr}, N). \quad (7)$$

We perform the same optimization process we do for SNR per pixel for SNR per resel by using the above expression for SNR.