Latent-Image Flagging

Frank Masci

August 24, 2001

Preliminaries

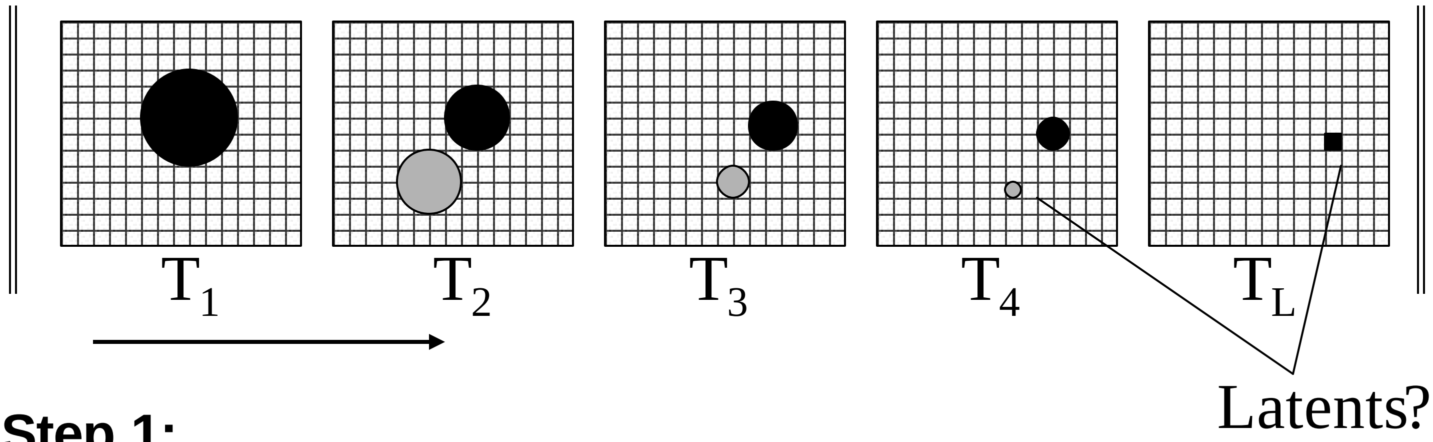
- **AIM**: To self-consistently flag pixels containing a latent by predicting forward in time which pixel intensities in an ensemble of images are likely to persist as latents in subsequent images of the ensemble.

- Will be performed at the post BCD level and comprise a BQD product.

- Main products:
  - Each BCD will have an accompanying 8-bit FITS image (called an L-mask) which specifies latent pixels with the value “1” and “0” if not.
  - For storage limitations, L-mask is only produced if latents are found.
  - A table which reports latent-pixel locations.

- Will involve ensemble processing of BCDs within a **single AOR.** There will be no crossing of AOR boundaries.

Proposed Algorithm



Step 1:

- From a latent decay model, compute the pixel threshold intensity (or total count) $DNthres_i$ in each image *i* of the AOR ensemble which will produce a latent above some noise level in all subsequent images.
- The predicted latent intensity has following functional dependence:

$$DN_{pred}(L)_i = f(\mathrm{T}_L, DN_i, \Delta T_{Li}),$$

  - T_L = Total time between resets in latent-reporting image (i.e. “frame time”). This determines the number of (latent) charge traps released.

Algorithm continued..

- DN_i = Total count within exposure time of the initial illumination frame *i* (or "fluence").
- ΔT_{Li} = Time elapsed since the start of the latent reporting image *L* and the end of illumination period in frame *i* - i.e. the latent decay time.

- From above model, we want to determine the threshold DN_i ($DNthres_i$) above which a latent will persist above some factor of the noise xs_L in image *L*:

$$xs_L = f(T_L, DN\,\mathrm{thres}_i, \Delta T_{Li}),$$

- In the above example, <u>each</u> initial illumination frame will have a list of thresholds corresponding to each subsequent "latent-reporting" image:

<u>Img 1</u>: $DNthres_1(T_2, xs_2, \Delta T_{2-1})$, $DNthres_1(T_3, xs_3, \Delta T_{3-1})$, $DNthres_1(T_4, xs_4, \Delta T_{4-1})$. . .

<u>Img 2</u>: $DNthres_2(T_3, xs_3, \Delta T_{3-2})$, $DNthres_2(T_4, xs_4, \Delta T_{4-2})$, $DNthres_2(T_5, xs_5, \Delta T_{5-2})$. . .

⋮

Algorithm continued..

Step 2:

- Flag all “suspected” latent pixels in each image of the ensemble by flagging those pixels in the preceding illumination images that have a total count (fluence) above the corresponding predicted thresholds.

- In the above example, suppose we desire a latent image report for image number 4 in the ensemble. This will be accomplished by flagging all pixels in images 1 -- 3 which have a total count:

$$\mathrm{DN} > \mathrm{DNthres}_1(T_4,\ xs_4,\ \Delta \mathrm{T}_{4\text{-}1}).$$

&

$$\mathrm{DN} > \mathrm{DNthres}_2(T_4,\ xs_4,\ \Delta \mathrm{T}_{4\text{-}2}).$$

&

$$\mathrm{DN} > \mathrm{DNthres}_3(T_4,\ xs_4,\ \Delta \mathrm{T}_{4\text{-}3}).$$

Require from Instrument Team

- Require a model in terms of a look-up table which shows the dependence of latent fluence (in electrons) on:

  - The initial source intensity (fluence) at t = 0 for a fixed exposure time (T_EXP). This can be later re-scaled for arbitrary T_EXP.
  - Time since the illumination was turned off.
  - Latent image frame time. (Duration in which the resulting latent fluence was measured).

- Pixel dependent noise model in the form of a look-up table, otherwise a single noise value will be computed from the distribution of background pixel counts.