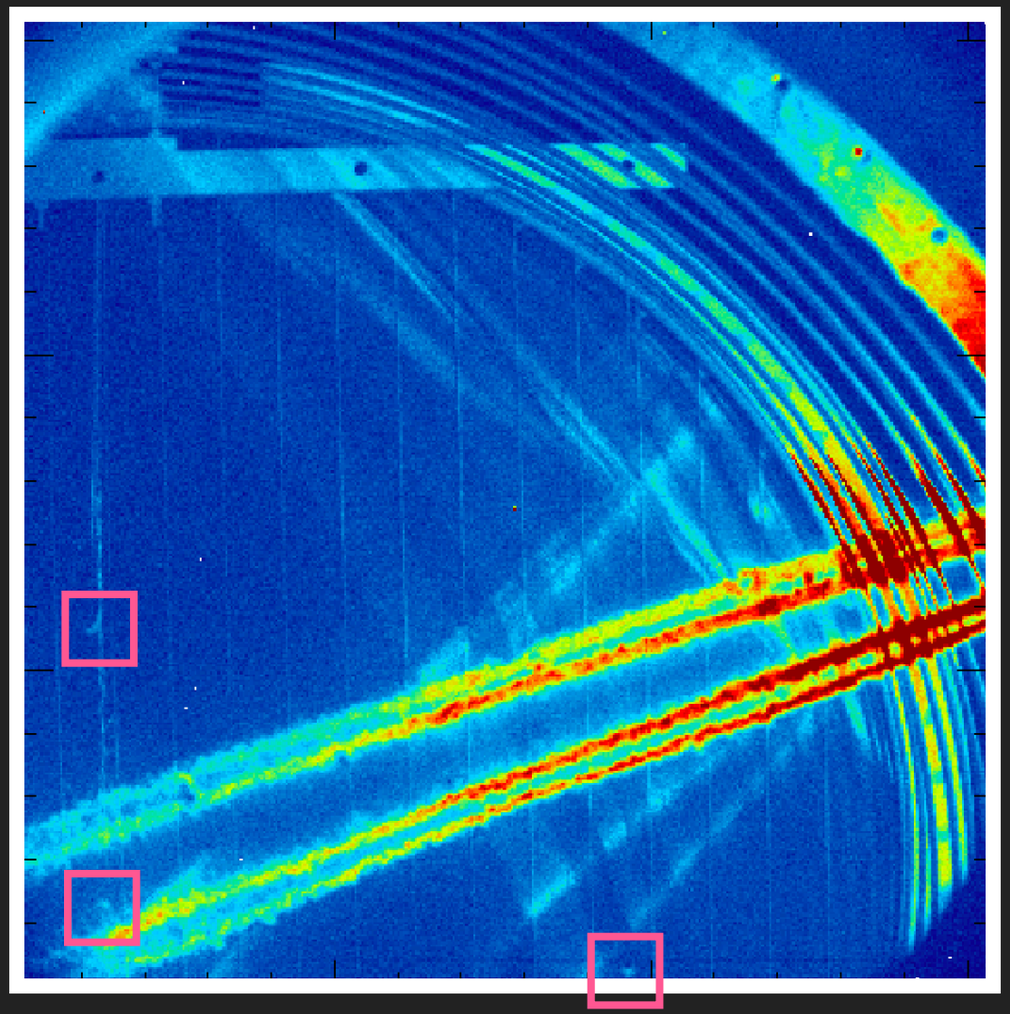
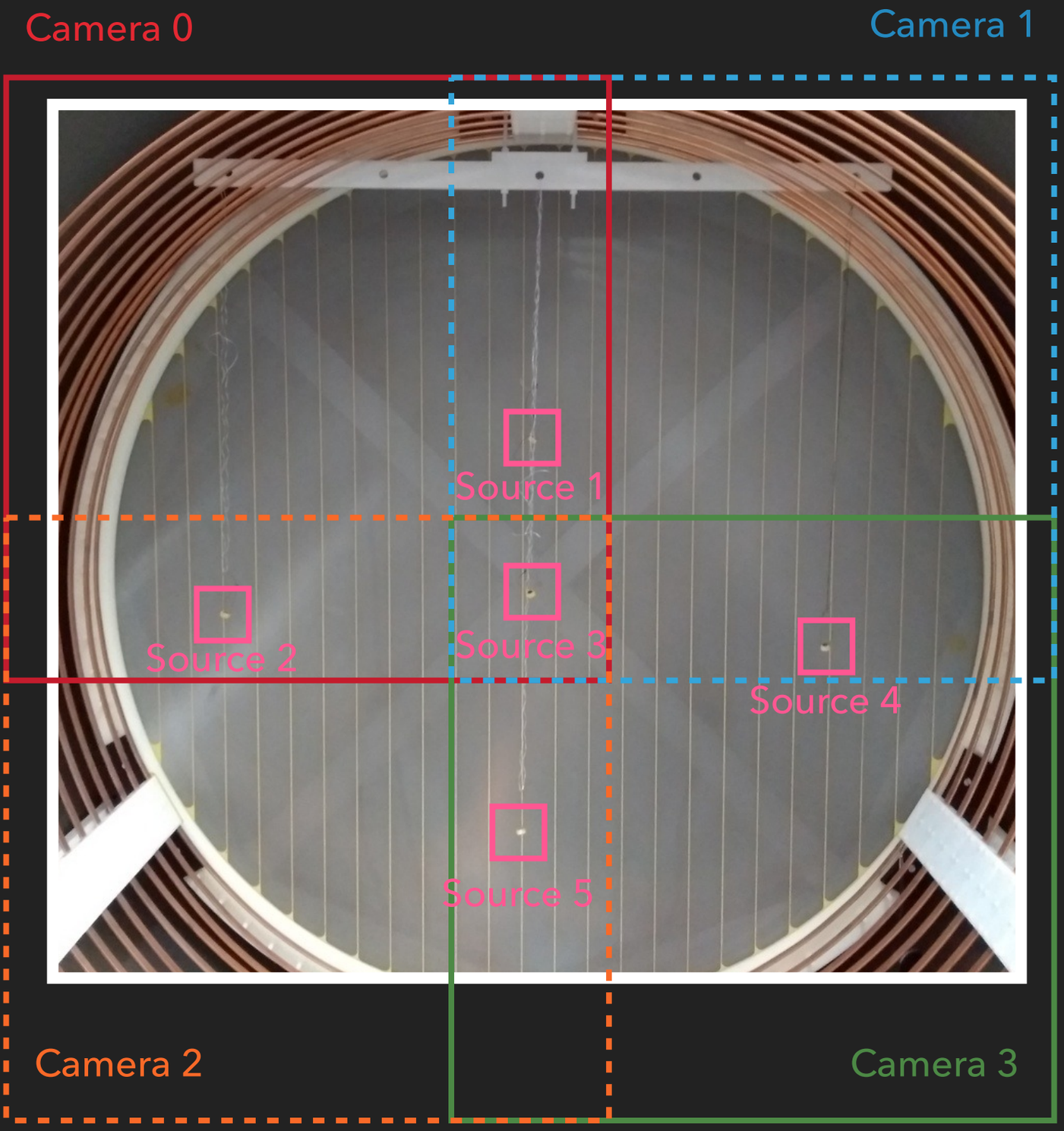


ZACHARY CHEN-WISHART 01/02/2019

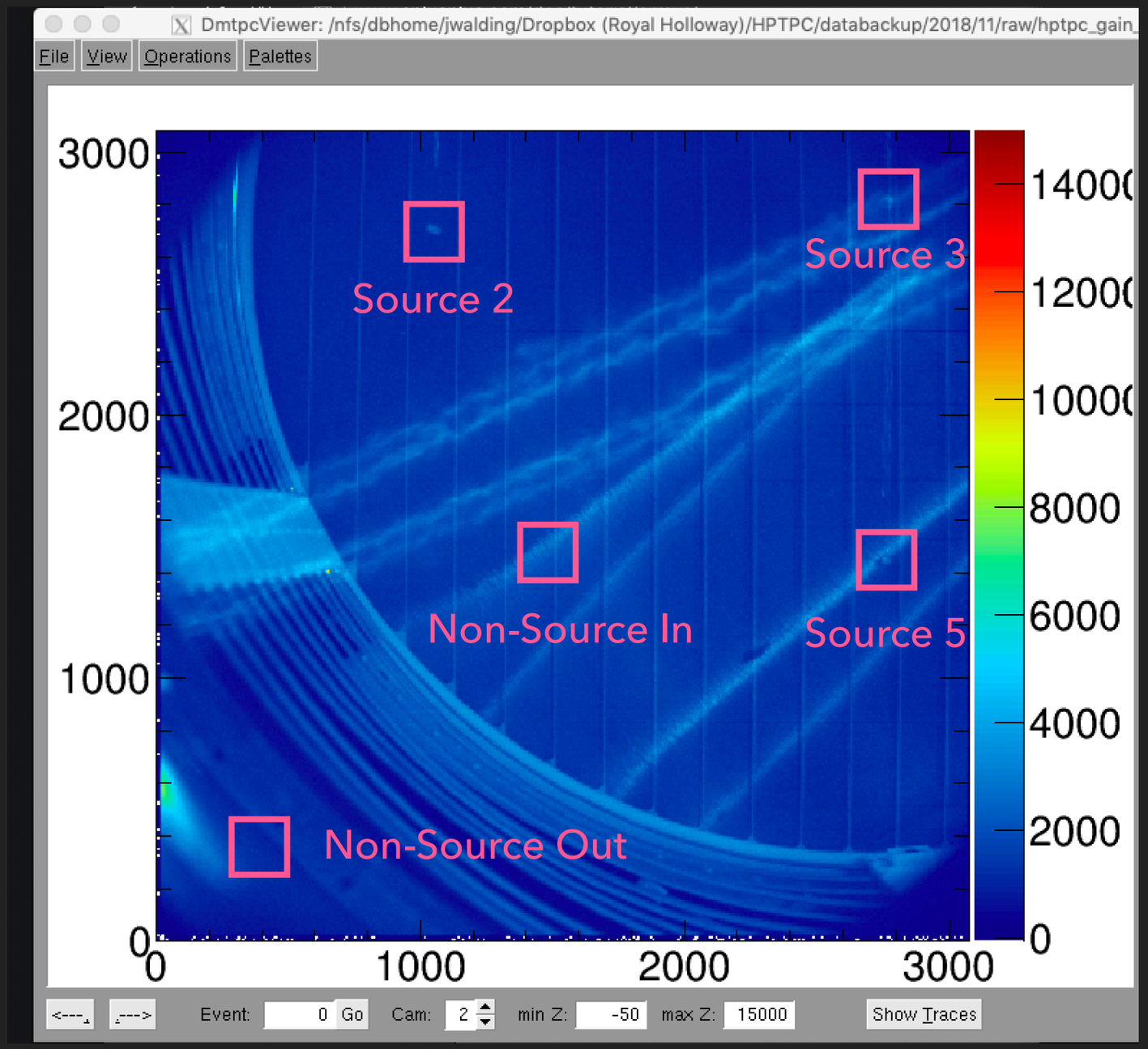
LIGHT SUM SQUARE

LIGHT SUM SQUARE – OVERVIEW: SOURCE LOCATIONS



- The code will be used for light gain calibration
- Aim: measure light at CCD from each caesium sources
- <- Source locations at instillation.
- Each camera can see three sources:
  - All cameras can see Source 3 and two cameras can each each of the other sources

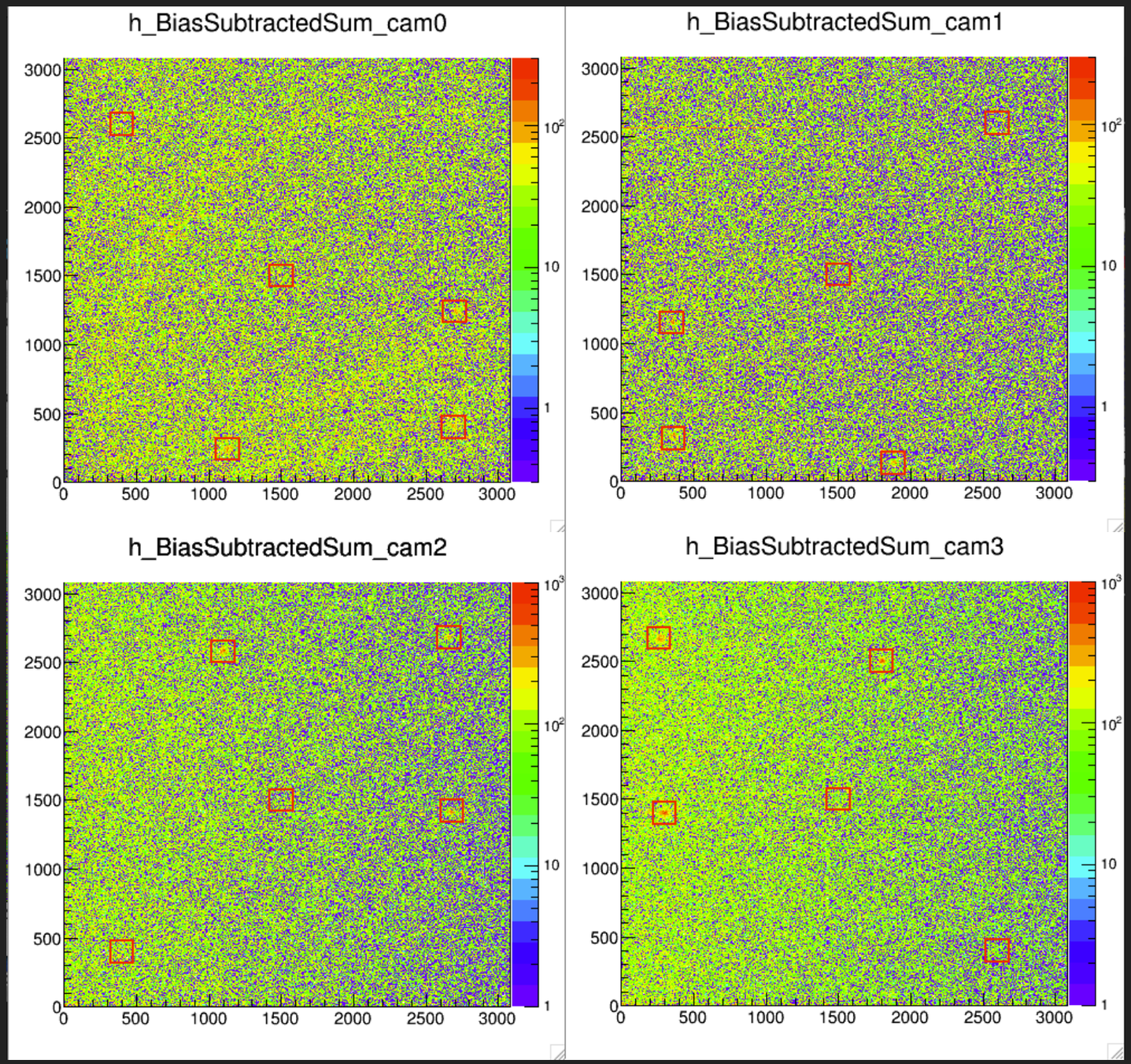
LIGHT SUM SQUARE – OVERVIEW: LOCATIONS IN CAMERAS



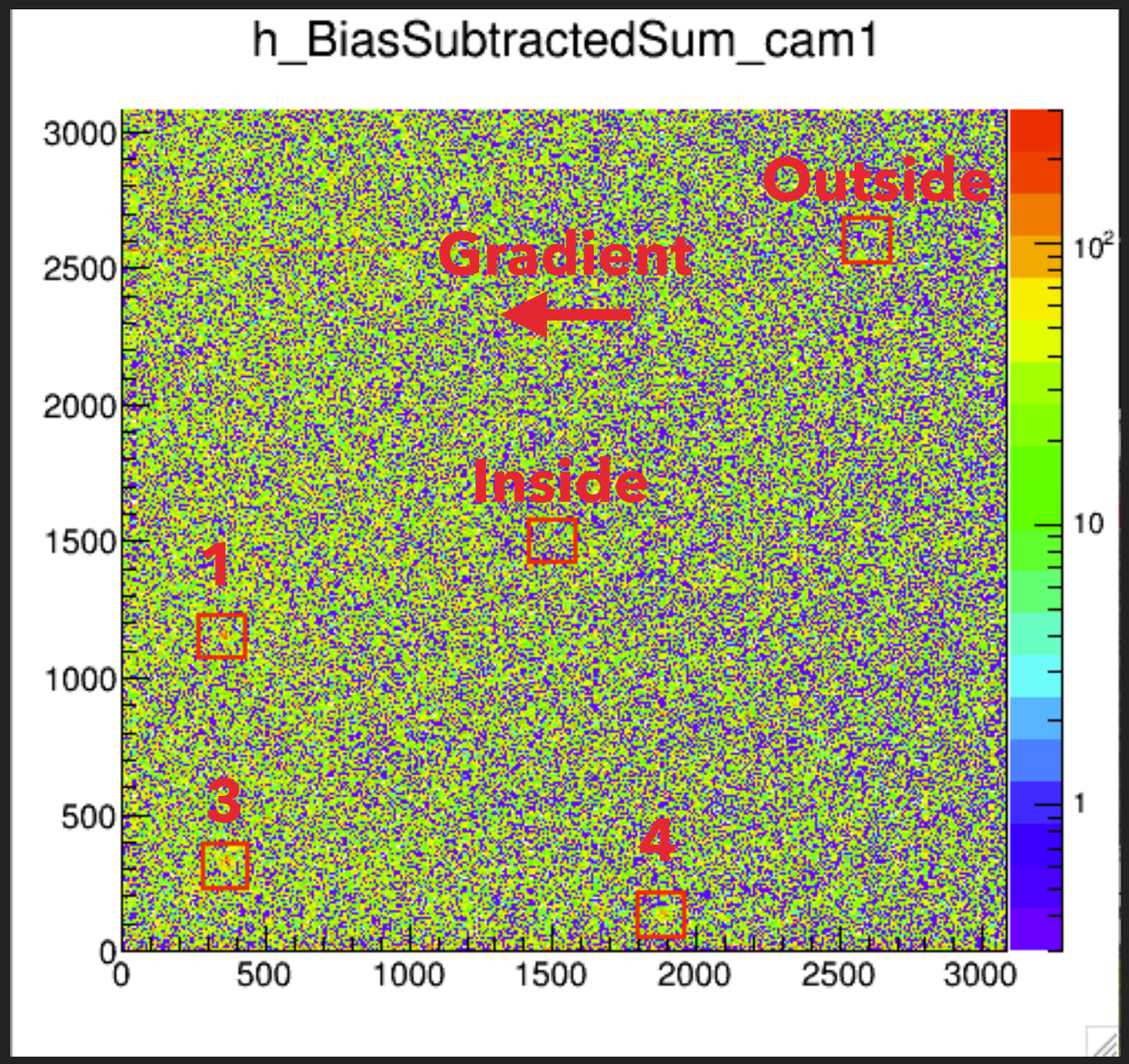
- A list of the following locations of interest for each camera was produced from spark events and summed runs:
  - 3x source locations; a
  - non-source location inside; and
  - non-source location outside of the amplification region
- <- Spark event R1323015 Event 0

LIGHT SUM SQUARE – OVERVIEW: SOURCE LOCATION FEEDBACK

- lightsumsquare currently accepts one image for each camera stored in a root file
- This output produced right is produced purely to visually check we have correct locations
- We will eventually use a light frames to "nail down" exact locations



LIGHT SUM SQUARE – OVERVIEW: OUTPUT



- lightsumsquare sums pixels within some defined region surrounding the stated locations - outputting summed pixel value and region size seen below for a summed run:
  - Source sums are approx twice that of non-source sums
  - Note there is a slight gradient from right to left which accounts for:
    - SUM (Inside > outside); and
    - SUM (Source 1 & 3 > Source 4)

```
<<< Camera 1 >>>
Source 1:     Sum of square: 11307      with average: 25.6395      pixels: 441
Source 3:     Sum of square: 12134      with average: 27.5147      pixels: 441
Source 4:     Sum of square: 8109       with average: 18.3878      pixels: 441
Inside:      Sum of square: 6144       with average: 13.932       pixels: 441
Outside:     Sum of square: 5218       with average: 11.8322      pixels: 441
```

LIGHT SUM SQUARE: NEXT STEP – INTEGRATING WITH RAPTORR

- Next step - to create a lightsumsquare function within raptorr to be able to:
  - run over a bias subtracted event level images
  - run over a number of bias subtracted event level images within a run
    - produce histograms showing the variation in summed regions over the course of a run and also between runs (note this may be able to aid in spark finding)
- As I am aware raptorr does not currently have the functionality to look at bias subtracted event level images yet - However this is currently being worked on by Toby
- My current plan, as discussed with Toby, Dom and Patrick is to wait till a bias subtracted event level loop is developed -> I can then implement my code