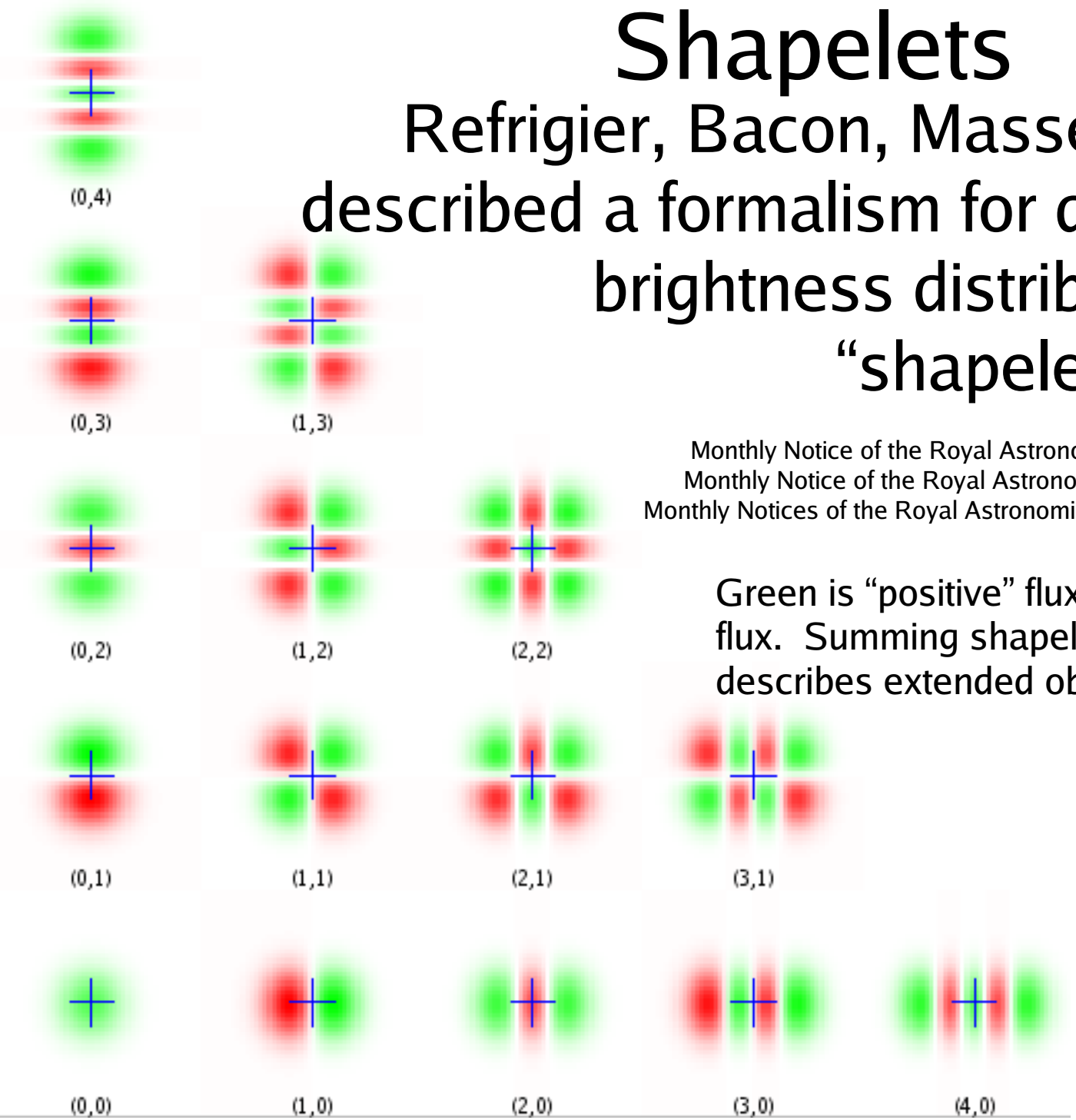
# Pixel Image Simulations

Update  December 7, 2006

# Shapelets

Refrigier, Bacon, Massey, and Conselice described a formalism for describing surface brightness distributions in terms of "shapelet" eigenfunctions.




Monthly Notice of the Royal Astronomical Society, Volume 338, Issue 1, pp. 35-47
Monthly Notice of the Royal Astronomical Society, Volume 338, Issue 1, pp. 48-56.
Monthly Notices of the Royal Astronomical Society, Volume 348, Issue 1, pp. 214-226.

Green is "positive" flux and red is "negative flux. Summing shapelet Eigenvalue*Eigenfunction describes extended objects well.

Algorithms for convolution with PSF (atmospheric + optical); rotation, and distortion due to gravitational shear.

# gov.* Packages

- gov.fnal.eag.matrix: complex matrix algebra used in shapelets
- gov.fnal.eag.shapelets: render and transform surface brightness
- gov.fnal.eag.sim.pixel:
    - PixeledImage is the central class
    - PixeledImageFactory orchestrates rendering catalogs to images
    - ImageRepository deals with FITS i/o; PixeledImagePlot makes .png with a ColorTable and ScaleTable.
    - interface DetectorObject represents one thing to be rendered: position, shape, and scale in pixels; number of photons. *Monochromatic.*
    - interface BackgroundLight gives photons/pixel in pixel coordinates
    - interface PSF deals with convolution
    - interface PixelValueTransformation implemented for noise and electronics effects, including persistence.

# Drizzle co-add

The class gov.fnal.eag.pixel.Drizzle has a main method that takes n images, registers them based on the WCS information in the FITS headers, and uses the drizzle algorithm for coaddition.

gov.fnal.eag.pixel.test.DrizzleTest makes four images in /tmp/ImageRepository.*username* and combines them.

# gov.* Packages

- gov.fnal.eag.sim.catalog:
  - AstroObject is the central class: position, redshift, SED spectral energy distribution (normalized to an absolute magnitude), Morphology
  - AstroObjectGenerator makes a catalog, using CoordinateGenerator, SimRandomRedshift, SEDGenerator, MorphologyGenerator, and SimRandomLuminosity
  - Zucca extends AstroCatalogGenerator for galaxies following Zucca et al. astro-ph/0506393 "The VIMOS VLT Deep Survey"
  - Filter has 51 transmission functions from http://www2.iap.fr/users/fioc/PEGASE.html

# Detector+Electronics Artifacts

- read noise, dark current, bias, gain
- poisson noise
- saturated/dead columns
- bleed trails
- cosmic rays (from data)
- readout ghosting
- charge transfer inefficiency (CCDs)
- persistence (IR devices)

The detector model is easily extensible.

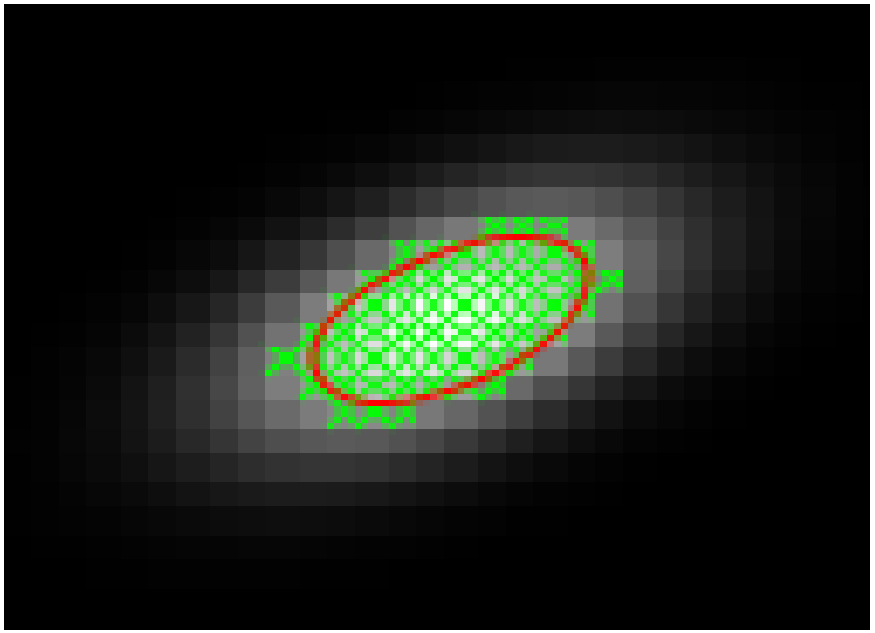
# gov.* Packages

- gov.fnal.eag.sim.extractor
  - Extractor is the central class.  It is similar to Sextractor.
  - It generates a collection of SourceObject which have first and second moments and total counts in

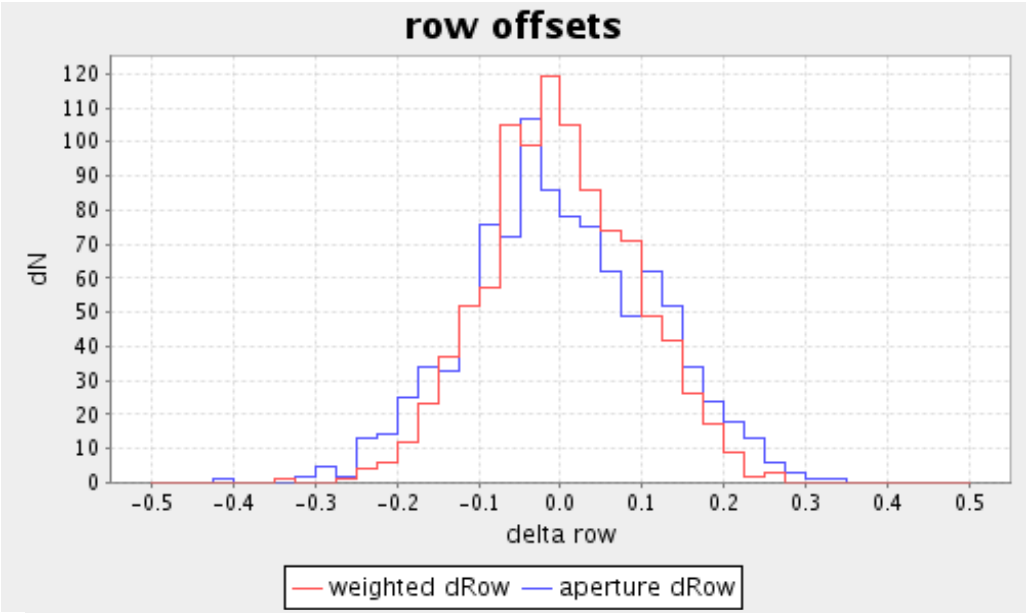
# gov.fnal.eag.sim.extractor

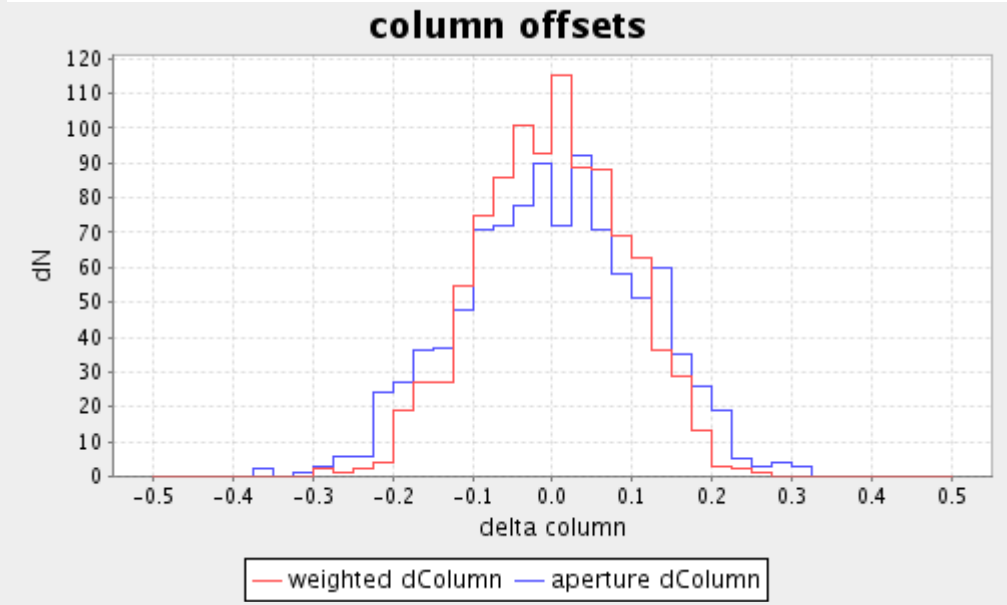
public class SourceObject – list of pixel+values; measure moments with various window functions

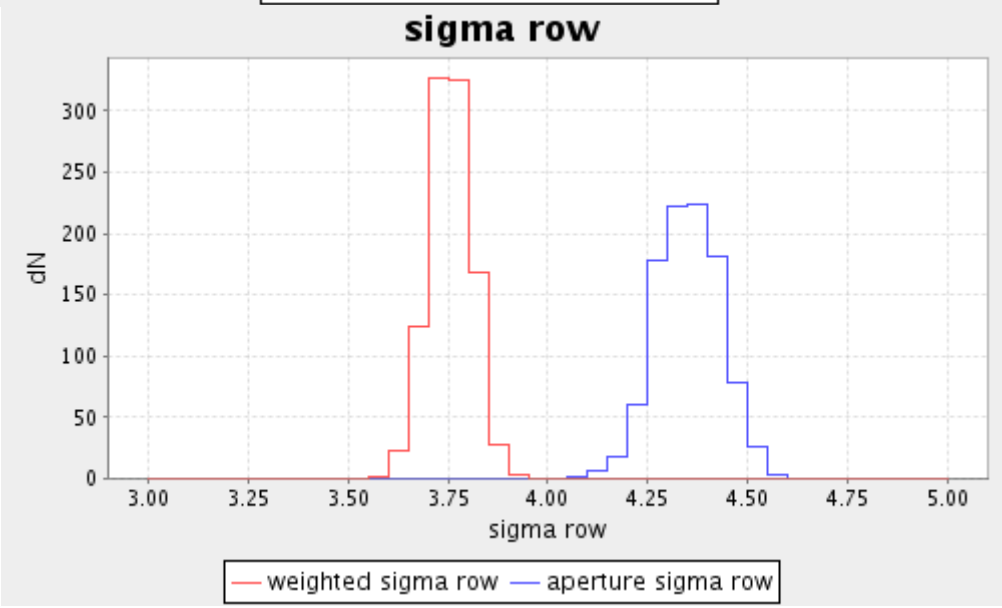
public class Extractor – find and measure SourceObjects in a PixeledImage

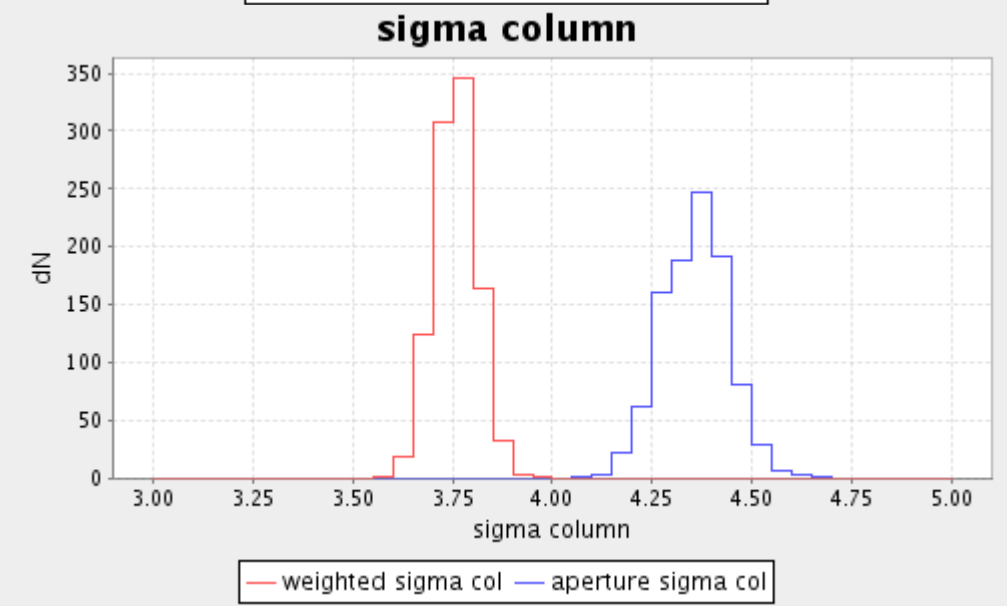


# Simple demo: 1000 gaussian stars

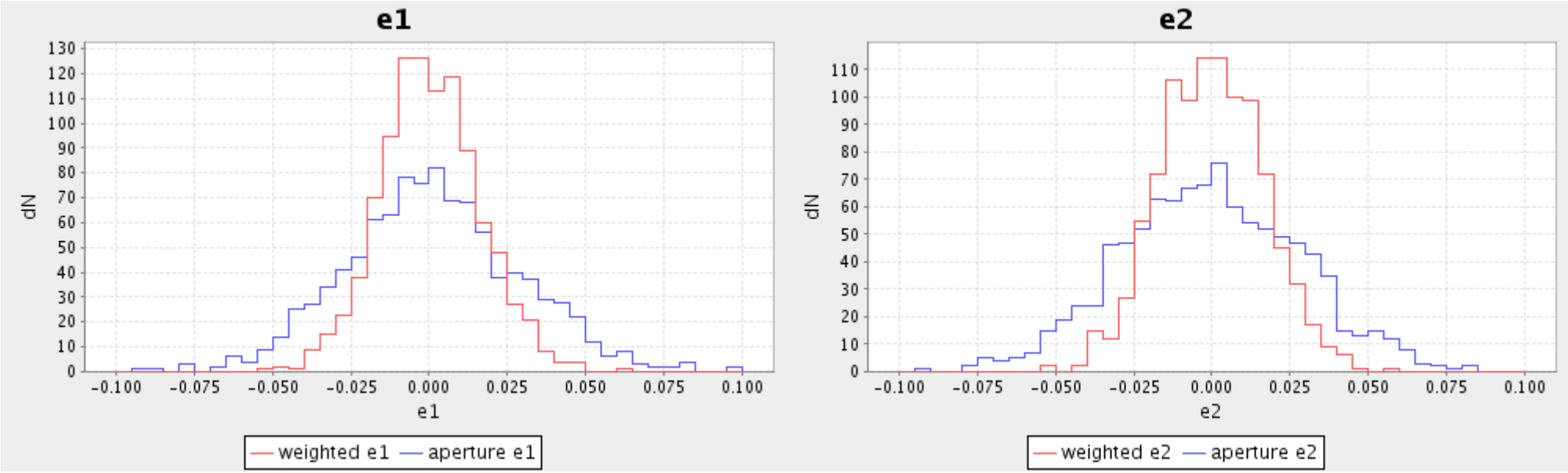


# Simple demo: 1000 gaussian stars




Generated by
gov.fnal.eag.sim.pixel.extractor.test.ExtractorDemo.testPositionAndApertureWeighted()

# Performance for DES

We are producing images for the Dark Energy Survey data challenge 2.

This is for 3340 pointings, where each pointing simulates data from 62 2048x4096 CCDs.

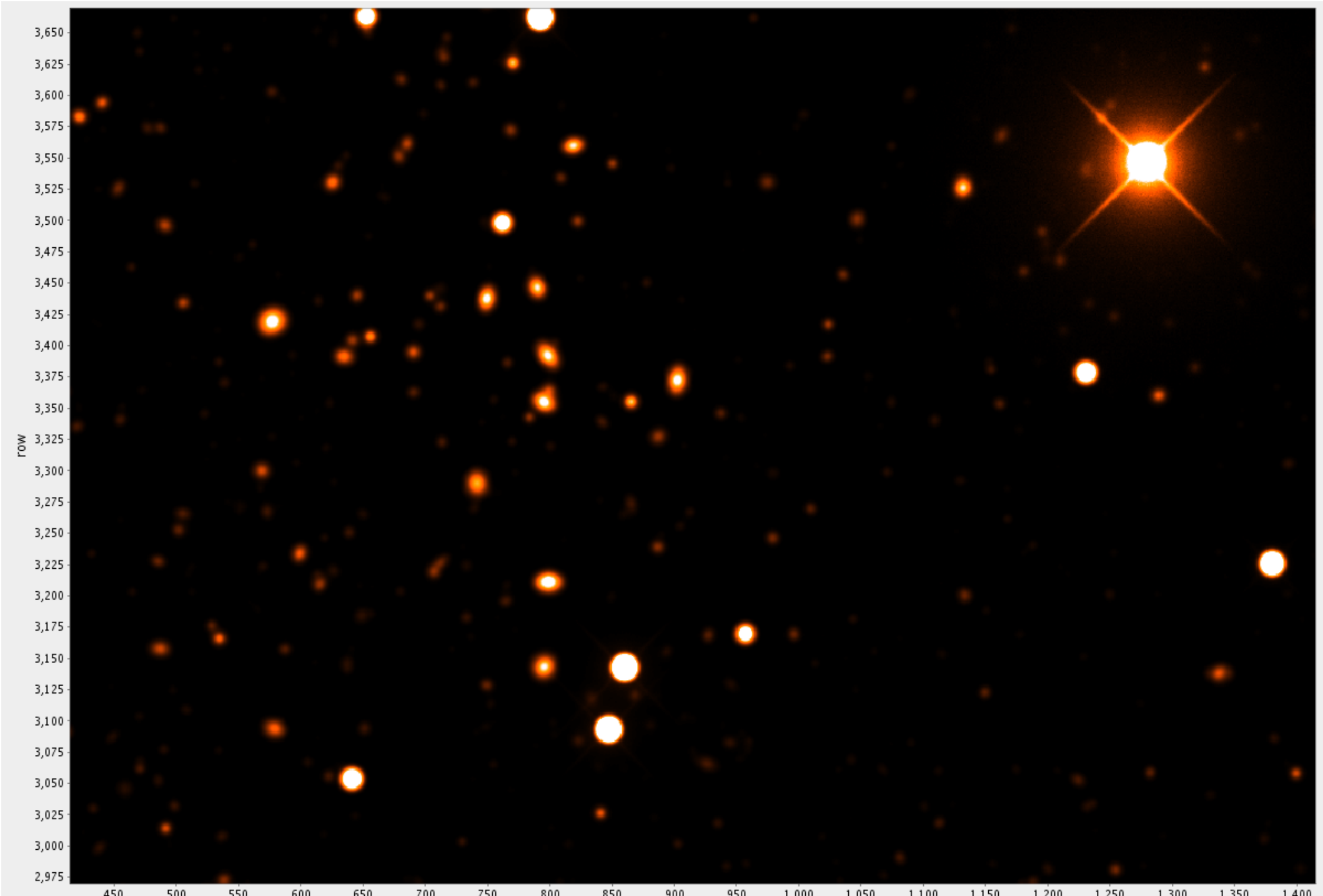
We save one file with 62 image HDUs (16-bit unsigned integers) to be fed to data processing.

We also save a “true” noiseless image and truth table of objects for each CCD.

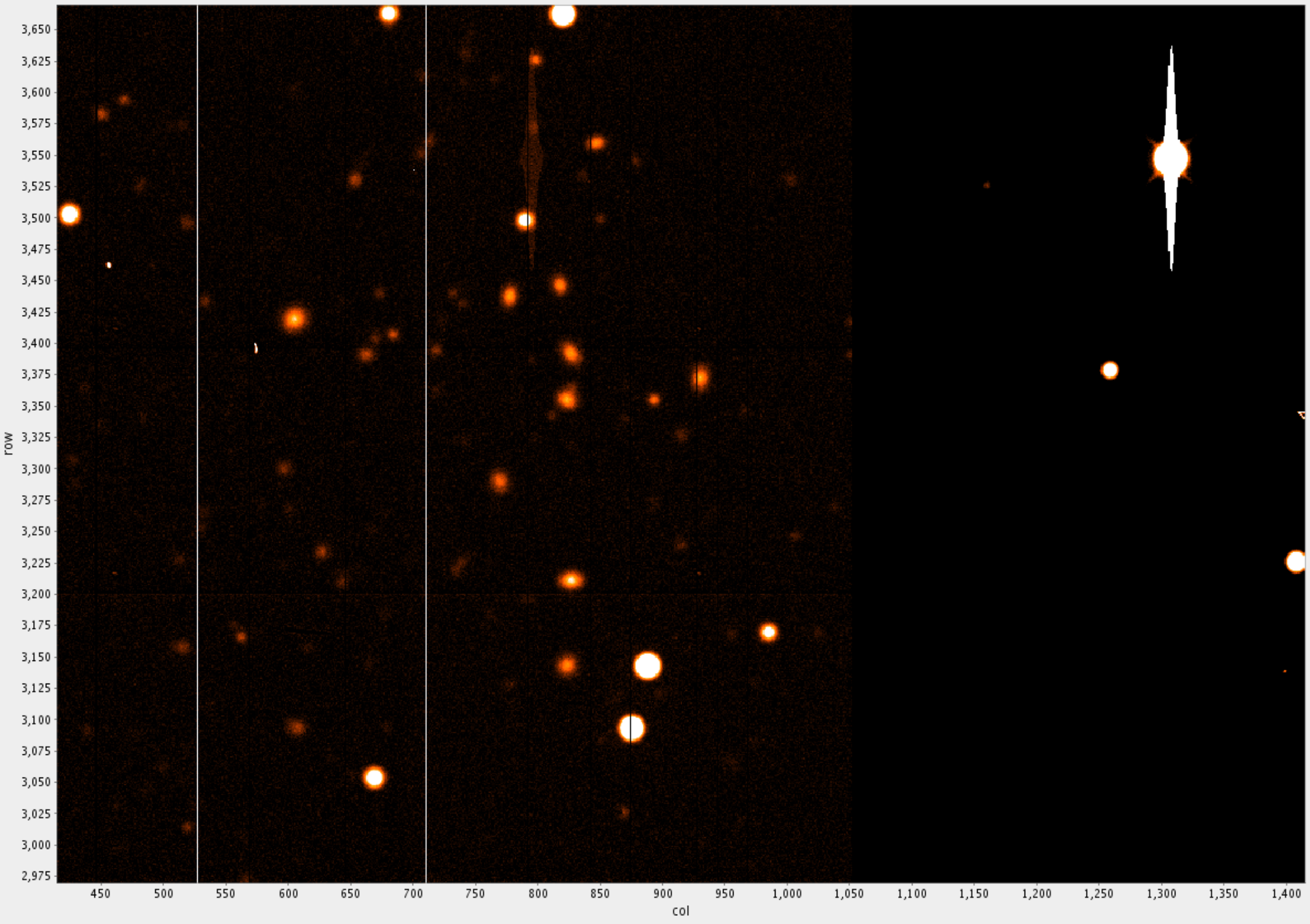
One pointing takes 2 hours on a 3 Ghz machine and uses 500 MB of memory.

“Raw” images will be 2 Tbyte; truth image+tables 8 TBytes

# Noiseless Image



# Noise and defects added



# CTI

Charge Transfer Inefficiency: (simple model; better model ~ total charge)

For a charge independent CTI, Janesick Eq. 5.2a (after error fix).

$$Q_{N_p+n} = \frac{Q_0 N_p!}{(N_P - n)!n!}(1-CTI)^{N_p-n} CTI^n$$

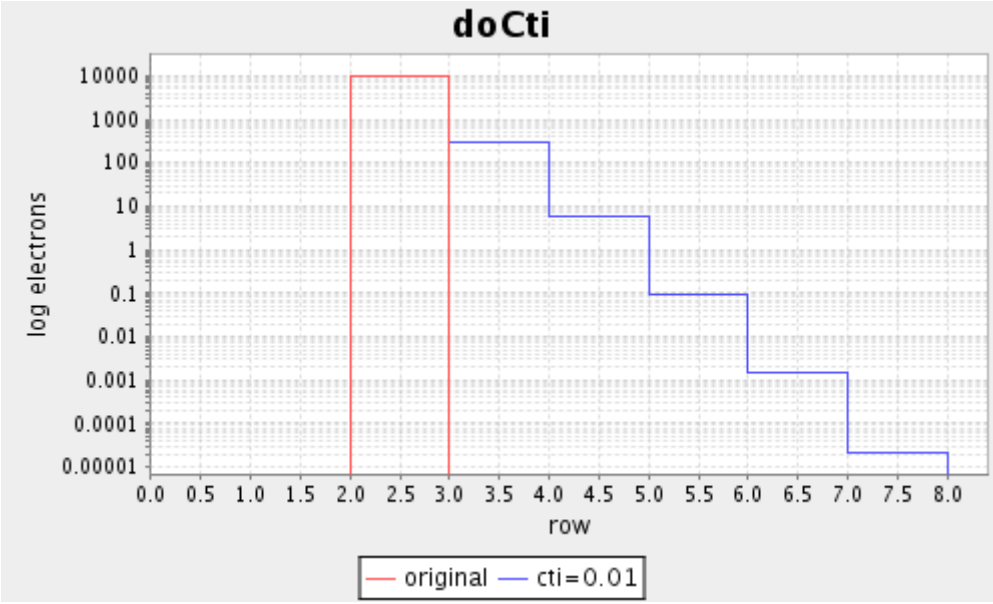
$Q_0$ is initial charge in target pixel.

$N_p$ is number of parallel shifts.

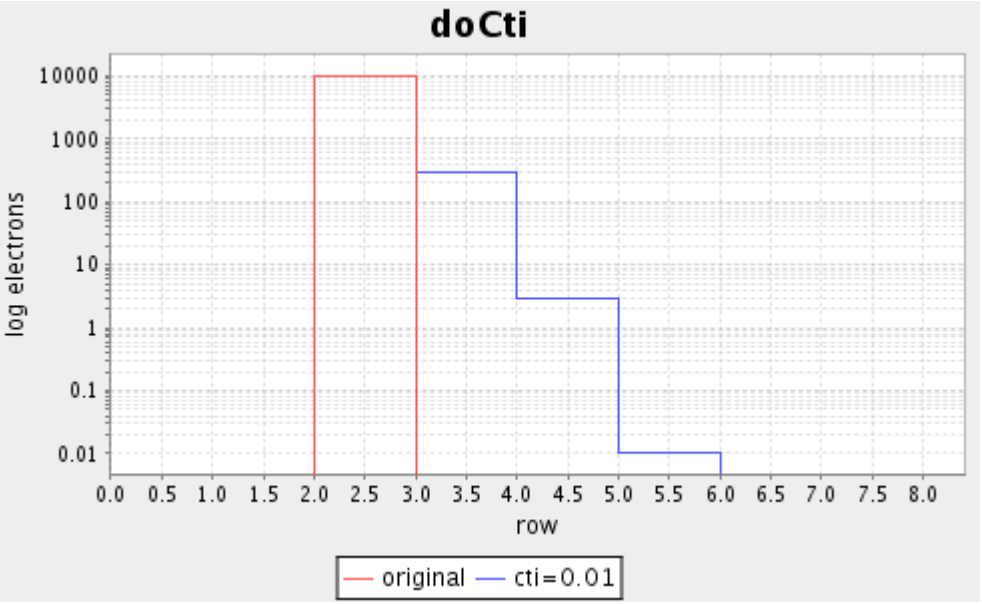
$n$ is the pixel (n=0 is target, n=1 is first trailing pixel, ….)

The “slow” implementation moves charge down a column. Subtract (1-CTI)*Q row N and add it to row N-1; for row=1,nRow.

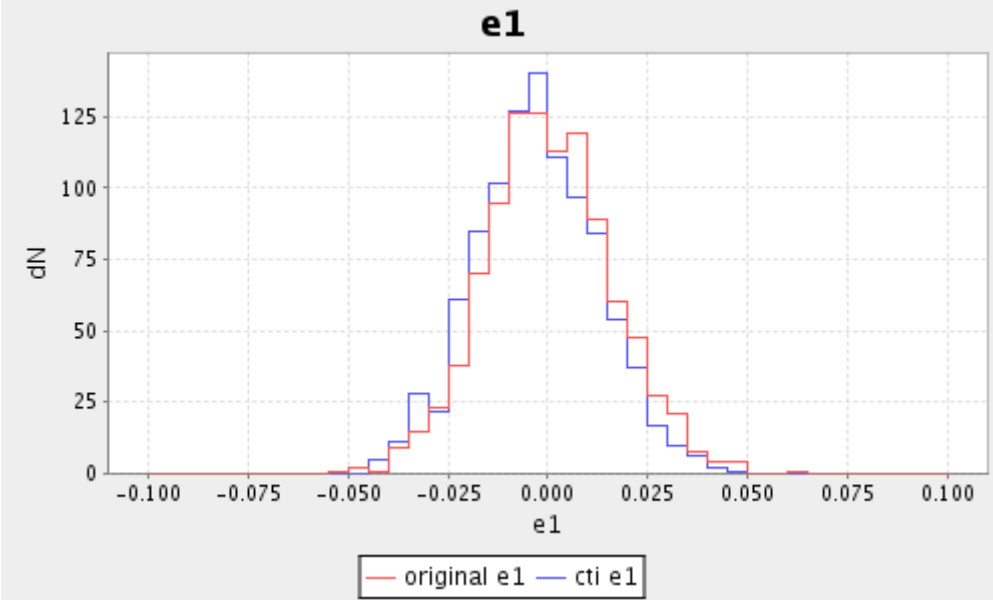
# CTI for 10000 electrons in row 2

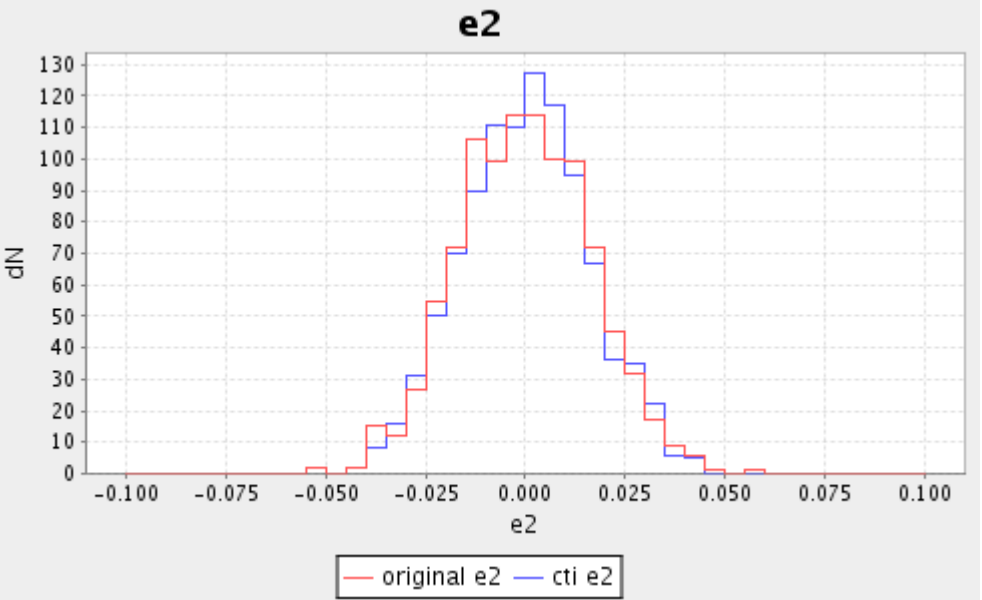



"explicit"



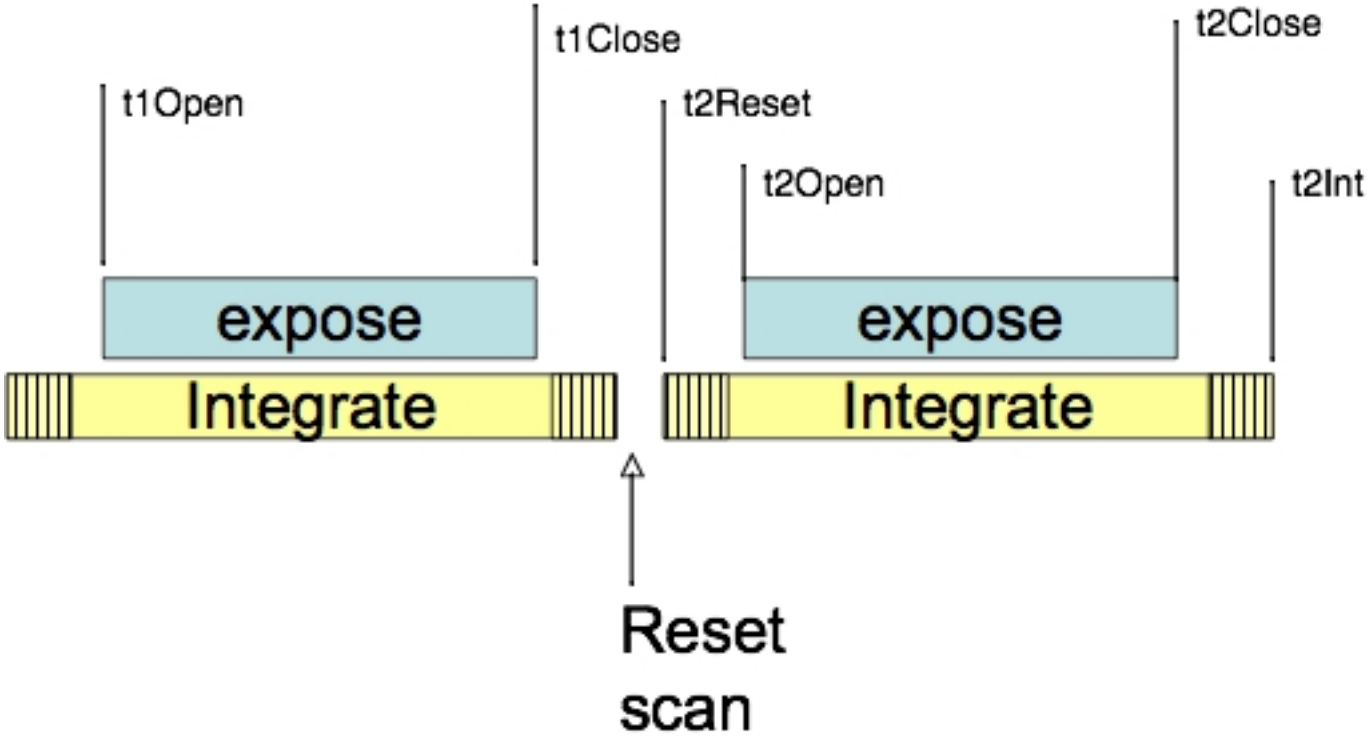

"fast"



# Persistence

Pixels in near IR devices have a persistence charge, which decays as the sum of two exponentials. How does dithering on the scale of a pixel or an extended object bias the shape measurements? How do previous “random” fields bias shape measurements?




The class gov.fnal.eag.sim.pixel.PixelValueTransformationPersistence does this, but we're working with Roger and Max to get the correct parameters.

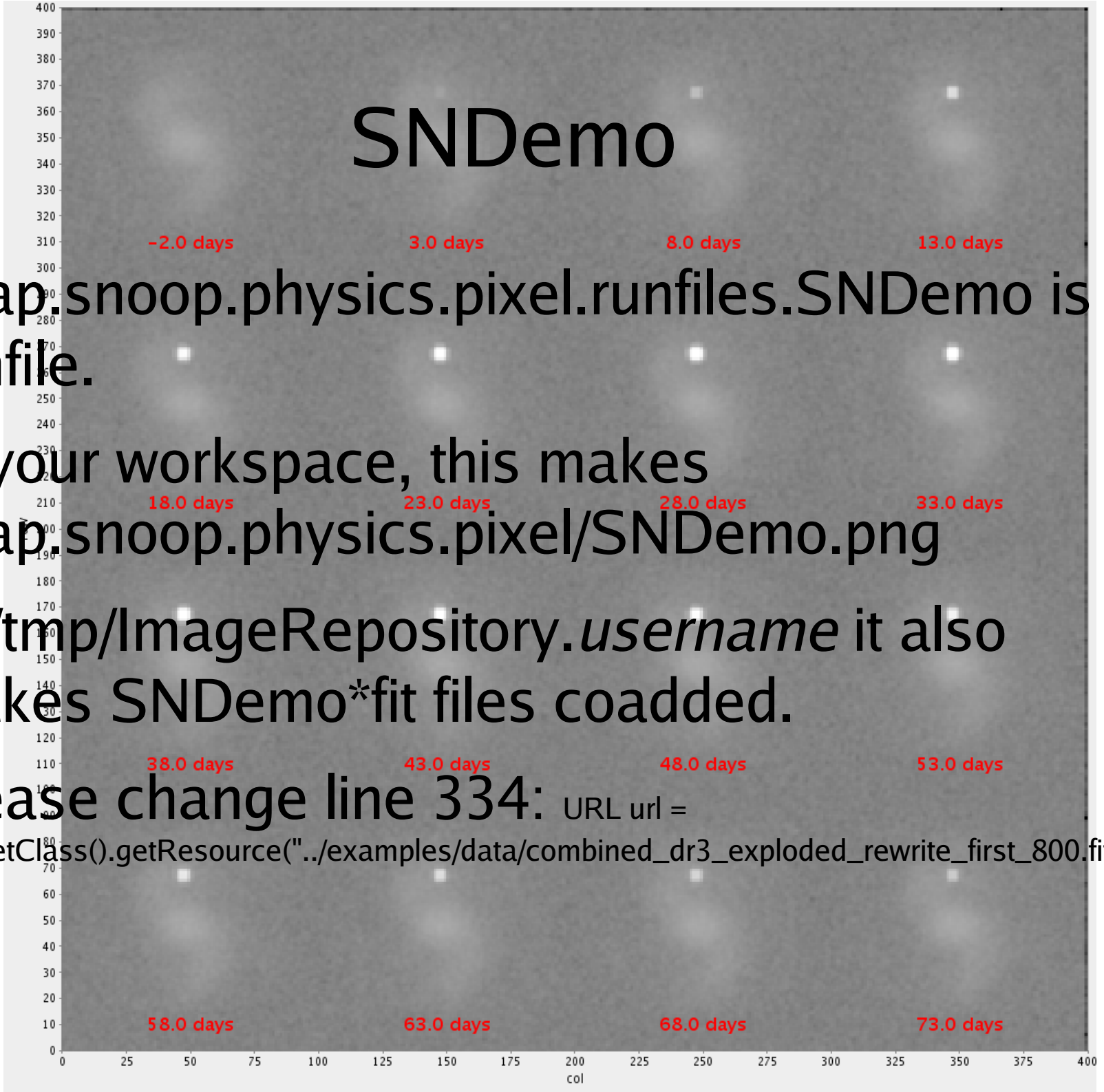
# snap.snoop.physics.pixel

- Integration of gov.fnal.eag.* occurs in this SNAP package.
- It depends on snap.snoop.physics.pixel.util, mission, observatory, etc.
- Some SNAP classes need to use snap.snoop, so they are extended or implemented in classes following a parallel structure:
  - snap.snoop.physics.pixel.mission.ObservationPixel extends snap.snoop.physics.mission.Observation

# extended Classes in snap.snoop.physics.pixel.

- .util.sb2d.SB2DShapelet and .util.sb2d.SB2DProductShapelet to implement sb2d in shapelet space
- .observatory.obs.TelescopePixel to deal with wavelength-dependent Airy function
- .observatory.obs.RealDetectorPixel to return SB2DShapelet for the point spread function
- .observatory.obs.ConfigurationFactoryPixel to use these
- .universe.sources.ConstantSED to define an AstronomicalObject with an (arbitrary) spectrum expressed as an SED

# SNDemo



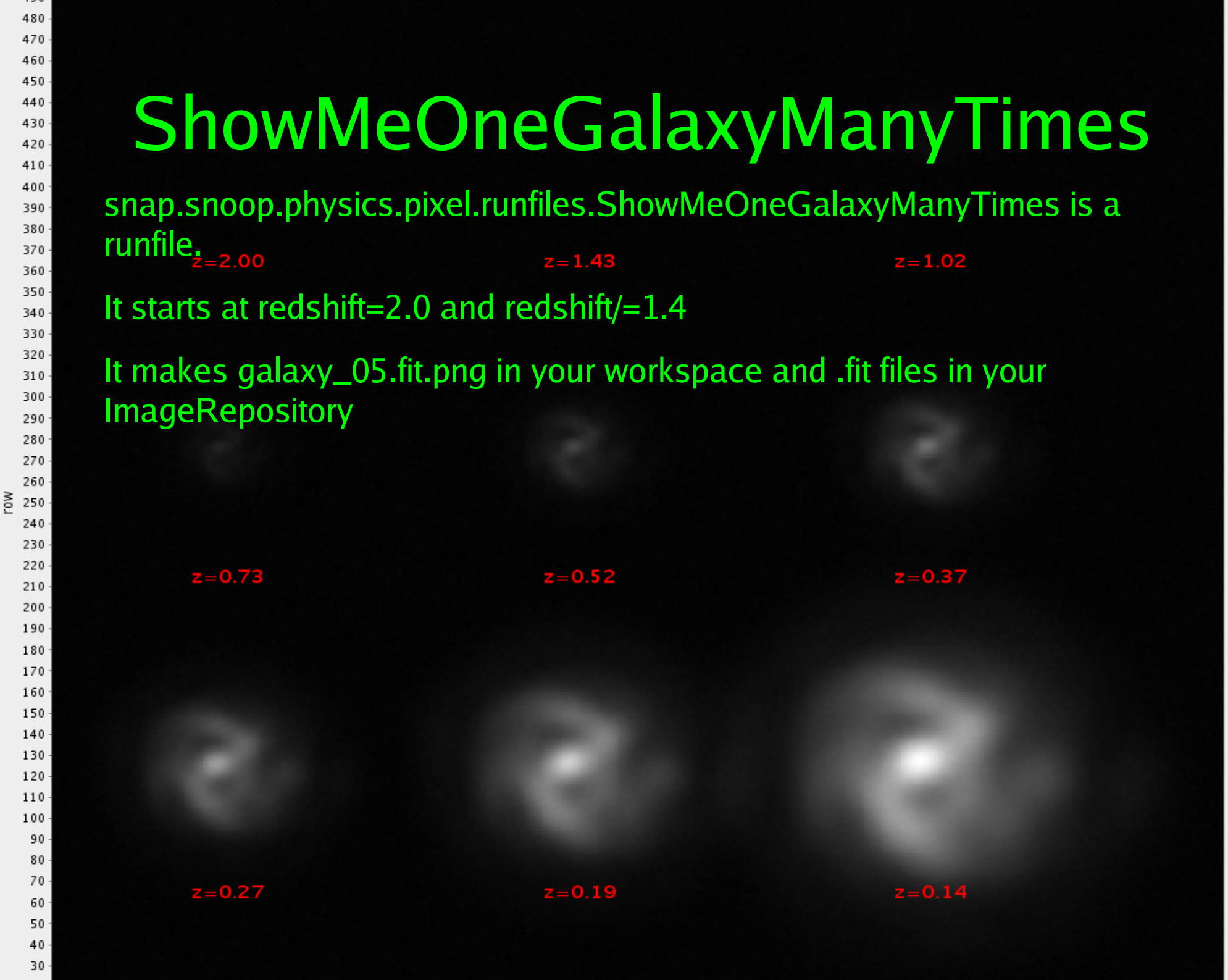

- snap.snoop.physics.pixel.runfiles.SNDemo is a runfile.
- In your workspace, this makes snap.snoop.physics.pixel/SNDemo.png
- In /tmp/ImageRepository.*username* it also makes SNDemo*fit files coadded.
- Please change line 334: URL url = this.getClass().getResource("../examples/data/combined_dr3_exploded_rewrite_first_800.fit");

# ShowMeOneGalaxyManyTimes

snap.snoop.physics.pixel.runfiles.ShowMeOneGalaxyManyTimes is a runfile.

It starts at redshift=2.0 and redshift/=1.4

It makes galaxy_05.fit.png in your workspace and .fit files in your ImageRepository



# Galaxy Field

- Uses the Zucca class from gov.fnal.eag.sim that implements Zucca et al. astro-ph/0506393 "The VIMOS VLT Deep Survey."
- This is a first hack at generating a catalog of galaxies in the SNAPSim framework. I hope that Alex et. al can help me get this right.
- I want to confirm that the fluxes are correct.
- It makes .png and .fit files in a similar way

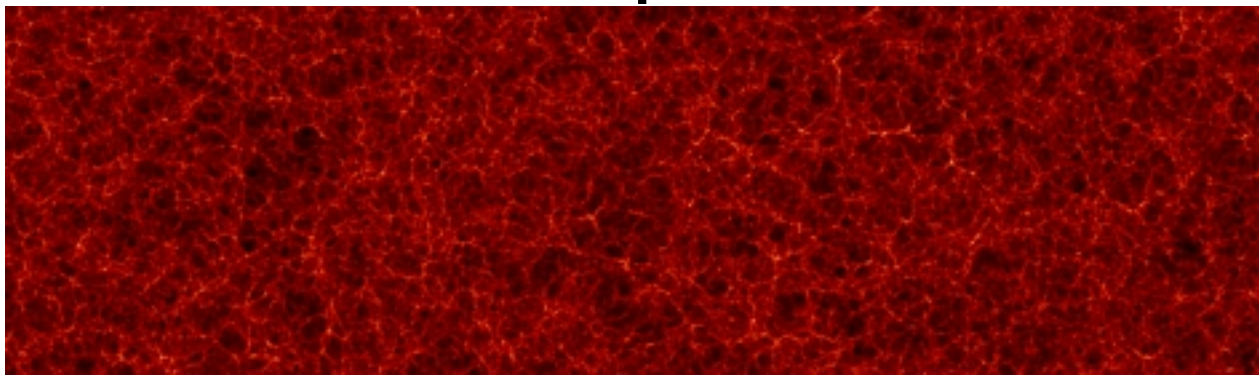
# two more examples

- snap.snoop.physics.pixel.examples has classes with a main() method
  - ShapeletGalleryDemo makes gallery_0.fit and gallery_1.fit in your workspace, each with 400 galaxies.
  - CosmicRayDemo makes CosmicRayDemo.fit in your /tmp/ImageRepository.*username* by tracking the “correct” number of minimum-ionizing particles through the detector. Need to check with reality! (This one is painfully slow.)

# DES Catalog Preparation

For the Dark Energy Survey, truth tables are generated by:

1. Results of Hubble volume ($10^9$ particles) N-body simulation (Virgo Consortium)



2. Assign galaxies to particles
3. Assign magnitude, color, and morphology based on environment
4. Add stars from USNO catalog, SDSS catalog, and Besancon simulation
5. Pick an extinction for each pointing
6. Convert to apparent magnitudes and scale

# to do

- snap.snoop.physics.pixel.runfile.ShowMeVega has trouble getting position from the astronomical source.  Plus, I'd like to confirm that the flux is correct!
- ConstantSEDTest.java does not integrate the ProductSpectralFunction.  Need to confirm fluxes.
- Discussion of catalog use and preparation