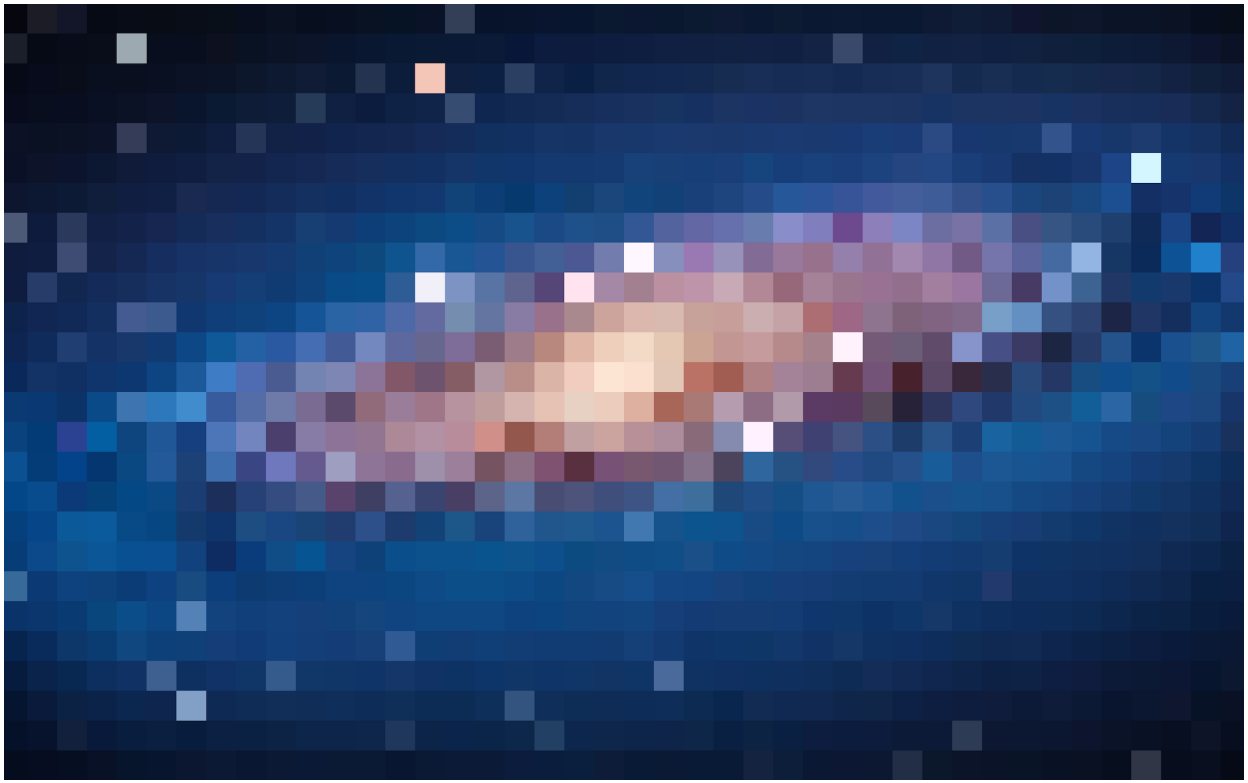
# Sensor model and validations



**Chris Walter**
**SLAC GalSim meeting**
**05/19/16**

# Needed for sensor model

- A whole set of silicon effects
  - Currently only BF implemented (more detail later)
- Electronics readout and camera geometry
  - LSST WCS and camera geometry exist but are not yet merged (Scott D.) Some of it is in GalSim interface, some in a outstanding GalSim PR.
  - There is no real electronics readout module but Jim C. etc are willing to help implement.

# Current implementation of these features in PhoSim
## (at various levels of validations)

- A/R Coating
- Charge Diffusion
- Fringing
- Debris on surfaces (-> non-uniform QE)
- Field-free non-uniform layer (-> short wavelength QE variations)
- Brighter-Fatter effect
- Tree Rings
- Edge effects
- Pixel boundary errors
- Dead Layer

# Not in PhoSim (but we need it)

- Cross talk (infrastructure exists, but not used?)
- Midline Stop Blooming
- Pixel size variation?
- More?

# For realistic output need to match real CSS/raft electronics:

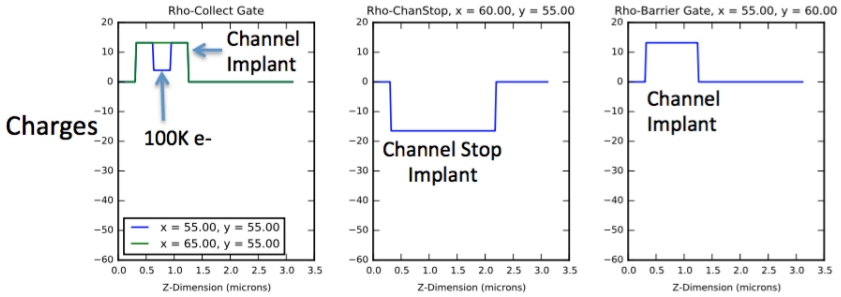
- With the real CCS and raft electronics we need:

  Bias, gain, segmentation, pre/over scans, non-linearity, cross-talk, CTE, hot pixels/columns, ADC errors, dark current, read noise, cosmic-rays etc.)

# Craig L. has made a physics based electrostatic solver.

## What are the Free Parameters?

- Diffusion Model:
  - None - Well established Silicon parameters.
- Potentials at Boundaries:
  - None - Applied voltages and geometries are known.
- Charges in Silicon Bulk:
  - Total Charge, Depth, and Profile in Channel region
  - Total Charge, Depth, and Profile in Channel Stop region
  - Is Channel Stop region depleted or are there free holes?
  - We will attempt to determine them with CCD measurements.



# But… currently too slow to plug in directly. So we need an interpolation scheme.

## Review of Strategy for Integrating Poisson Simulator into PhoSim

- Use the Poisson simulator to pre-calculate pixel vertex displacements as a function of charge and Z-height.
  - This will only need to be re-calculated when the CCD model or CCD parameters (voltages, temperatures, etc.) change.
- At PhoSim start-up read in pixel vertex displacements.
- For each photo-electron, build displaced pixel boundaries based on charge built up in surrounding pixels.
  - Currently going to +/- 3 pixels.
- Use distorted pixel boundaries to decide where photoelectron ends up.

# How can we use this model with GalSim?

- In order to deal with dynamic effects we need to build up the image.
  - Use the 'photon shooting method'. Photos are sampled from a probability distribution based on the object profile and projected onto pixels.
  - Doing this one-object-a-time won't work. We need to first build a "truth profile" of the whole and image and then sample from that.

# At DESC meeting they made a first pass:

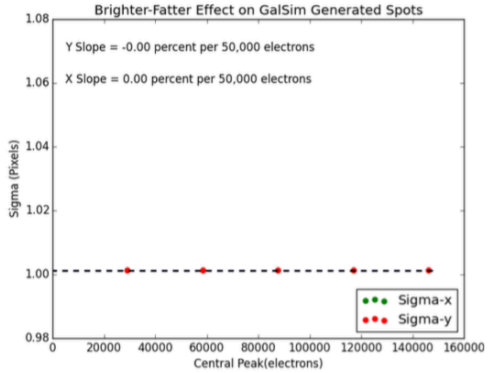
## Summary

- At the SLAC DESC meeting Hack Day, we successfully integrated distorted pixels in the GalSim photon shooting routine to model the brighter-fatter effect.
- Initial implementation resulted in 10X slowdown of the photon shooting, from $\approx 1,000,000$ photons/second to $\approx 100,000$ photons/second.
  - This first implementation recalculates the pixel shape for every photon.
  - Going forward, we will only re-calculate the photon shape for every N photons added to a pixel, with $\mathrm{N} \approx 1000$.
  - We have mapped out how to do this, but it still needs to be implemented.

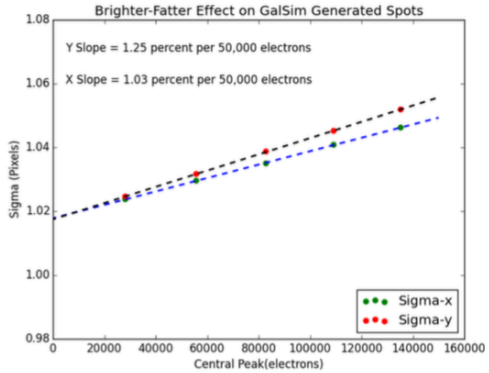
# Working to 1st order.

## GalSim Comparisons

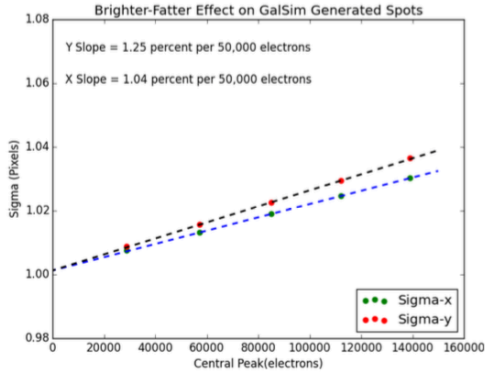
### Existing GalSim-1.3 Code



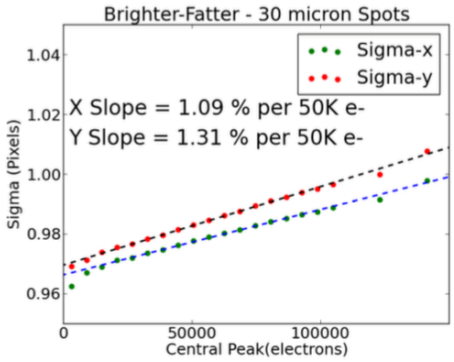

### With B-F, With Diffusion




### With B-F, No Diffusion




### Measurements



# Main issues

- Sensor effects are wavelength dependent! We need a way to keep this information.
  - Could interface into chromatic objects by binning truth in frequency and drawing from that.
  - Should we consider a real raytracing mode?
    - I have a student who will do some tests with chromatic photon shooting starting in about a week, for a month.
  - Need validated diffusion model.
- We need the API to be quite general.
  - It shouldn't be tailored to (e.g.) the interpolated model.

# Ray tracing questions

- If we used raytracing for the atmosphere (for speed), could we turn this back into wavefronts for the optics (by calculating the OPDs)? Then, of course we would need to go back to photon shooting.
- Does this make sense?

# DESC SRM DC Needs

| Q13 Which group are you representing for this survey? | Q9_u | Q9_g | Q9_r | Q9_i | Q9_z | Q9_y | Q8_#u | Q8_#g | Q8_#r | Q8_#i | Q8_#z | Q8_#y | Q11 Prefered Cadence | Q12 SSOs? | Q10 If you care: Max # back-to-back exposures with coherence in atmospheric PSF. | Q5 Variation of the PSF across the focal plan should be: | Q6 Sky model should be based on: | Q7_1_1 Clouds? | Q7_2_1 Vingnetting? | Q9_1 In Sky Model: Gradients across image? | Q9_2 In Sky Model: Twilight? | Q10_1 Sensor: Fringing | Q10_2 Sensor: AR Coatings | Q10_3 Sensor: Brighter Fatter | Q10_4 Sensor: Tree Rings | Q10_5 Sensor: Edge Effects | Q10_6 Sensor: Saturation | Q10_7 Sensor: Blooming | Q10_8 Sensor: Simulated electronics readout |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Photoz** | 1 | 1 | 1 | 1 | 1 | | | | | | | | WFD | | N/A | | OpSim | Yes | Yes | No | No | | Yes | | | | | | |
| **SL CX2 Twinkles** | 1 | 1 | 1 | 1 | 1 | | 56 | 80 | 184 | 184 | 160 | 160 | WFD | No | N/A | LSST Like | OpSim | No | No | Yes | Yes | No | | No | No | No | Yes | Yes | No |
| **Supernova** | | | 1 | | | | 0 | 0 | 100 | 0 | 0 | 0 | Twinkles DDF | No | N/A | | OpSim | No | No | No | No | No | No | No | No | No | Yes | Yes | No |
| **LSS (Updated)** | | | 1 | | | | | | 50 | | | | WFD | No | N/A. | 8m optics | OpSim | No | Yes | | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **WL** | | | 1 | 1 | | | | | 10 | | | | DDF | No | | 1 LSST Like | OpSim | No | Yes | No | No | No | No | Yes | No | No | No | No | No |
| **Clusters (Updated)** | | | 1 | 1 | | | | | 50 | 50 | | | | No | N/A | | Fixed bright and dark with and without moon. | No | No | | | No | No | No | No | No | Yes | Yes | No |
| **Notes / Consensus** | Photo-Z probably not necessary in DC1<br>R&I will fulfill everyone except Twinkles.<br>150 visits in both?<br>WL will use GalSim<br>Twinkles will run themselves / Done?<br>Make LSS, Clusters in R&I | | | | | | | | | | | | | No<br>(BLANK = Don't Care) | Only WL cares | LSST Like | OpSim | No | Mixed | Only Twinkles for both | | LSS WANTS EVERYYTHING<br>(BLANK = Don't Care) | | | | Except WL | Except WL | |

For DC1 only saturation and blooming needed immediately.

For DC2 will likely also need BF etc

# Sensor Model Validation

**Sensor Anomalies WG Key Projects:**



We need so model and validate sensor effects by using test stand data from BNL, SLAC, and Davis.

Sensors also taking data on the sky

# Validation efforts happening in SAWG

- Brighter-Fatter ← Main focus now
- Tree Rings
- Edge effects

Work is happening with many techniques:

- Test stands, flats, pin holes, optical simulators
- PhoSim simulations, Electrostatic Solvers, commercial chip programs, ..

# More validation thoughts

- Adding in all of the other effects is a fair amount of work.
  - Person power? How do these get added to Craig's model? I want to insist that we build a "physics based model".
- But, I feel we are not really "behind" when it comes to the validation. That all really needs to be done (again?) anyway for our chosen sensors.
- If we can build a simulation package, validating it is something we can naturally slot into the existing plan.
- We do need a plan and people so we can work together.

# Conclusions / Comments

- We should write a document capturing this (broader) discussion, proposing what we need to do so we can see the big picture and check off on projects as they are completed.
- Calibration systems need to be simulated (lab + telescope).
- To make a quick start I think we also need a phosim instance catalog parser + LSST chip output package. Scott has code that does part of this in the CatSim interface. Having this would allow rapid progress on all other fronts.
  - This would allow us to use the infrastructure we have already built for validation.