Weak Lensing Algorithms and Lessons Learned from Precursor Experiments

Erin Sheldon

Brookhaven National Laboratory

Match 19, 2020

The charge from the organizing committee:

"...to speak on algorithms and pipelines you have developed or used in other surveys, lessons learnt, particularly any unexpected challenges you have encountered, and to highlight what you would do differently or additionally with LSST in order to maximize scientific return."

Outline

- The challenge
- Lessons from precursor surveys
- What we in DESC are doing for weak lensing shear measurement.

The Challenge

- We have set the bar extremely high. We want to make full use of the rich data we will take at Rubin Observatory.
- For example, for the weak lensing probe we need to recover the signal with an accuracy of about 0.1%. At the time of writing, the most accurate measurements in the literature are at the 1-2% level on real data.
- Other probes also have very tight requirements.
- In order to meet these goals we must put forth a huge effort, embrace novel ideas and work with great discipline.

Precursor Survey Work

- SDSS shear pipeline development and lensing analysis
- BOSS target selection framework
- DES shear pipeline development and lensing analysis
- What did we learn from all this, and how are we applying this knowledge to our work for LSST?

Lesson: Identify the Critical Problems that are Unsolved

- There are unsolved algorithmic problems which *must be solved* if we are to succeed.
- Dedicate as many resources as possible to solving these algorithmic problems
  - This will require some *convincing*
- Spend as little time as possible on solved problems (more on this later)

Example from My Experience

- DES was proposed in about 2003 with weak lensing (WL) as a primary probe. We thought we would soon have a WL method that would meet our needs.
- Weak lensing measurement is still not a fully solved problem in 2020. Only recently have candidate algorithms been introduced that can work in principle.
- Over the years there have been at most a few persons full time sustained effort (FTE) spent on this problem within DES, at a given time.
- With hindsight we should have dedicated more effort to it.

Effort on Unsolved Problems in LSST

- The situation is similar in LSST right now, with about 1.3 FTE to make existing WL algorithms work for LSST (zero officially for research, PS are not funded to do research).
- We understand a lot more now. We have good reason to *think* we have excellent candidates (BFD, METACALIBRATION etc) but....
- If we fail we must fall back to *calibration from simulations*. More on this later.

Lesson: Use Existing Solutions if they are Good Enough

- These solutions might not fit perfectly, we may have to bend them out of shape or adapt to their idiosyncrasies
- In most cases it will save so much time that it is worth it
- There is often more than one solution available, especially in cases where the software industry has already solved the problem.
- There are unsolved problems which *must be solved* if we are to succeed. We must put our effort towards those problems.

Lesson: Aggressively test algorithms with validation simulations

- Build up simulations in such a way that we can toggle all the relevant features, independently if possible
- Turn on features one at a time until we a) find a problem which we will try to fix or b) we reach full planned simulation complexity and the algorithm provides acceptable accuracy.
- This requires dedication and effort that can equal or exceed the work on the algorithm itself.

Lesson: Aggressively test algorithms with validation simulations (cont.)

- Each test must be much more precise than the expected precision of the measurements we plan to make on real data. For example, we want to be confident the bias is less than our requirements with at least 99.7% confidence. This is necessary but can be resource intensive.
- Don't waste your time thinking about 1 sigma error bars: always quote at least 99.7% confidence regions. We will run thousands of tests and 2 sigma fluctuations will be seen regularly. I've seen results move my more than 3 sigma.
  - This applies to real data analysis too

Lesson: Use best software practices, e.g. Unit Tests

- Validation tests are in addition to unit tests, which may also be partly simulation based
- Use extensive unit tests and continuous integration (e.g. circleci on github)

Lesson: Confirmation bias is even more relevant for validation simulations than for real data analysis

- The process is by definition iterative, we look for problems, fix them and rerun. It's easy to subconsciously create a situation within the simulation or analysis that artificially produces a zero bias result.
- We tend to stop when we measure no bias and move on to the next test.
- I'm not sure how to combat this except through diligence.
  - We could try to have two completely independent simulation packages, with one not influenced by the testing results. But this does not seem practical.

There May be Some Remaining Bias in the Validation Sims

- There may be some remaining biases in our measurements which we either have good reason to think we can't fix, or we don't have time; think of a year 1 cosmology deadline.
- For lensing we have no absolute calibration sources, so we may need to use simulations to make the correction.
- How should we approach these corrections?

Lesson: Calibration simulations are harder than good data analysis

- Often with data analysis, we don't need to understand everything that we measured, or how it is measured.
  - We don't do absolute photometric calibration by simulating the instrument. We calibrate to reference sources.
- If we instead need to use a sim for calibration, and the correction is large, we must *really* know what we are doing.
  - We need to understand what we put into the sim and it must match the real world very well.
- Therefore we should put as much effort as possible into developing algorithms that require small corrections.
  - If the correction is 0.5% we will be less sensitive to the inevitable errors in the simulations than if the correction is 10%.

What are we doing in DESC for weak lensing shear?

I'll first give the big picture of what we are doing and then give details and results toward the end

What are we doing in DESC for weak lensing shear?

- LSST DM needs the community to deliver state of the art algorithms. Shear algorithms are still an area of research.
- Pipeline scientists (PS) Sheldon, Becker and Armstrong are working to implement existing algorithms at about 1.3 FTE
- (DESC PS are technically not funded to do research, only to implement existing algorithms. So the research happens outside of DESC or "in our spare time").

What are we doing in DESC for weak lensing shear?

- We in DESC plan to deliver at least two algorithms.
  - BFD (DESC PS Armstrong)
  - Metacalibration (DESC PS Sheldon & Becker)
- I'm not going to go into details of our algorithms because they are not of general interest.
  - Both methods work well enough for isolated sources.
  - Metacalibration includes a detection phase to deal with blending (Sheldon et al. 2019).
- What are we doing to make these work for Rubin Obs. data?

Coadding

- We coadd in small regions of sky
- For lensing we need a *continuous PSF*.
- We must only coadd images that do not have an edge in the region of sky.
- We waste less if we use smaller regions
  - Technical points
  - Too small is difficult for METACALIBRATION, as we also need to rerun detection on sheared versions of the images. About an arcminute is what we currently use, but we have not optimized this.
  - We may be able to simply redefine the patch size in the standard DM code. This is TBD, currently we are defining the regions arbitrarily.

PSF Coadding

- We coadd the PSF exactly the same way we coadd the images, at least statistically.
- This is different than the stack PSF coadding, more on that later.

Noise propagation

- We propagate a noise image through all of the relevant processing stages.
  - Image resampling for coaddition produces correlated noise.
  - Interpolation of bad regions produces biases and correlated noise
  - Algorithms such as BFD and METACALIBRATION can propagate these effects using the noise image in order to produce unbiased results, all else being equal.

Noise propagation

- For BFD the noise field should include the noise from undetected sources. However, the extra variance can be included separately by measuring the effect of undetected sources in the data (Eckert, Bernstein et al. 2020, in prep).
- For METACALIBRATION the noise field should match the background noise (Sheldon & Huff 2017). Undetected sources are part of the signal.

Detection

- We detect on a combined multi-band "straight" coadd, nominally $r + i + z$.
- We process objects in multiple bands simultaneously
  - Note after coadding BFD and METACALIBRATION work separately

Details of how are we in DESC doing the work

- Are we applying the lessons learned?
- How is the work going?

Details of how are we in DESC doing the work

- All code for working with the stack, METACALIBRATION, and the sims is open source on github, heavily unit tested with continuous integration.

Using Existing Solutions

- We use the stack for everything we can
  - Take the calibrated exposures as input
  - Use the stack coadding code
  - Use the stack PSF
  - Use the stack detection code
  - Use the stack WCS code

Feed back our solutions when we can't

- We had to write code to include the sub-pixel shifts in the PSF coaddition. Important because the resampling causes a small amount of blurring and we need this to be fully reflected in the PSF.
- Not currently possible to propagate noise images through the image stack CR interpolation, for technical reasons, so we are doing our own noise propagation and interpolation. We will work with DM to make this a stack feature.
- We will need to mask bright stars. It may be this can happen down stream. We hope we can pull code from HSC to do this, but we may need something custom.

Aggressive testing with validation simulations

- We are developing image simulations in parallel with the analysis code (a good fraction of the 1.3 FTE goes towards the simulations)
- We can toggle each feature independently.

Current Simulation Features

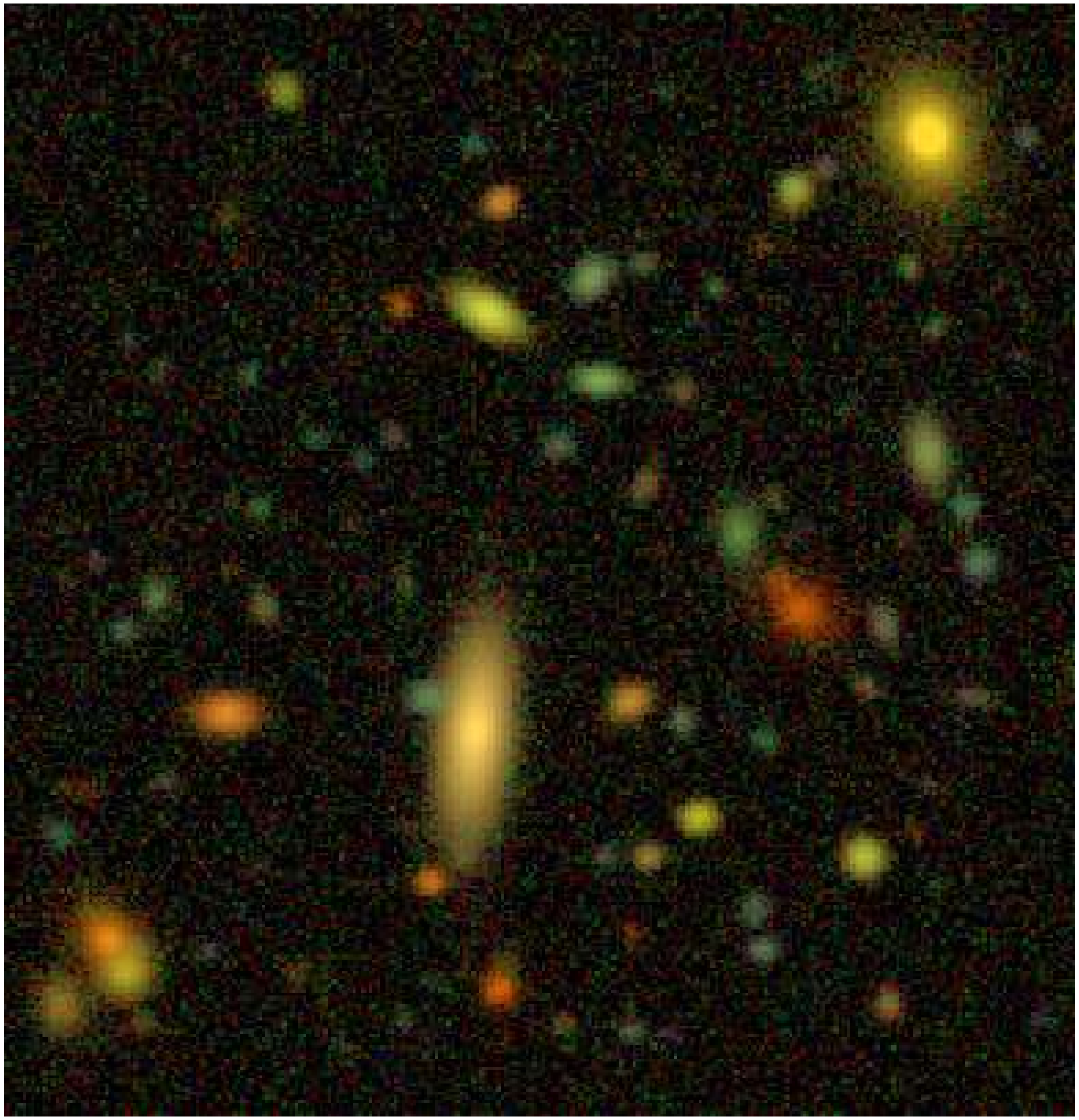
- field dithers, rotations
- TAN WCS with variations in pixel scale and wcs shear (need more realism)
- cosmic rays, bad columns
- Realistic galaxy size and mag distributions and bulge+disk+AGN model (WeakLensingDeblending)
  - More morphological complexity will be added, but for modern algorithms this is not critical for most testing.

Current Simulation Features cont..

- Realistic multi-band star mags and galactic stellar density variations following DC2 (thanks to J. Sanchez)
- Primitive Star saturation and fake bleed trails
    - TODO realistic astrometric distortions, more realistic bright star effects (Jim Chiang helping to use DC2 examples)

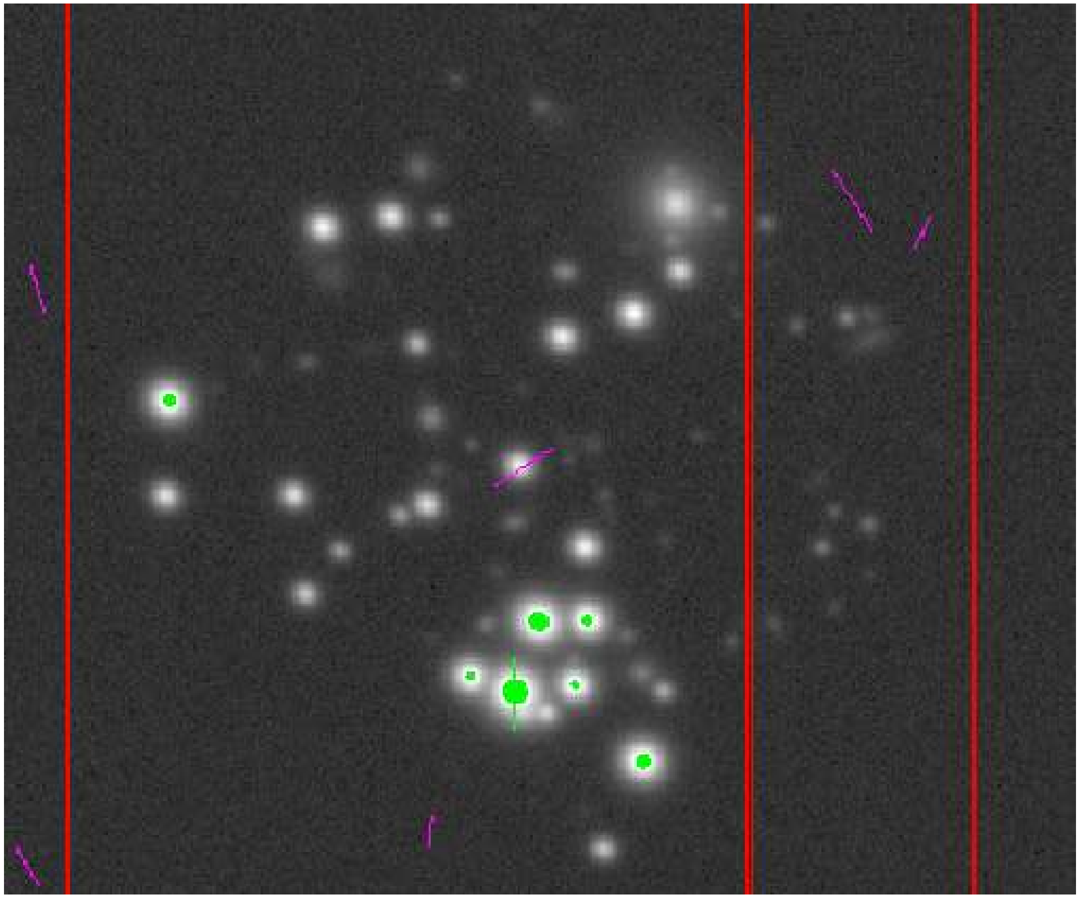
Example Galaxies

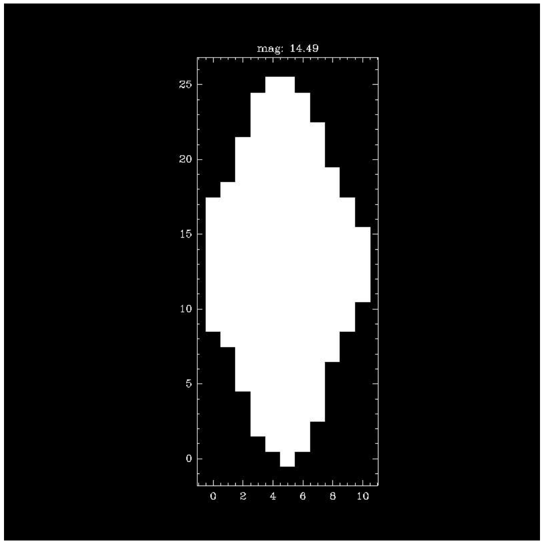
Familiar to those who have used the WeakLensingDeblending package



Example Sim with Artifacts

High stellar density field (80/sq arcmin)





Current Results

- Turn on features one by one
- Use wide confidence regions (99.7%)
- Results for METACALIBRATION shown at the right

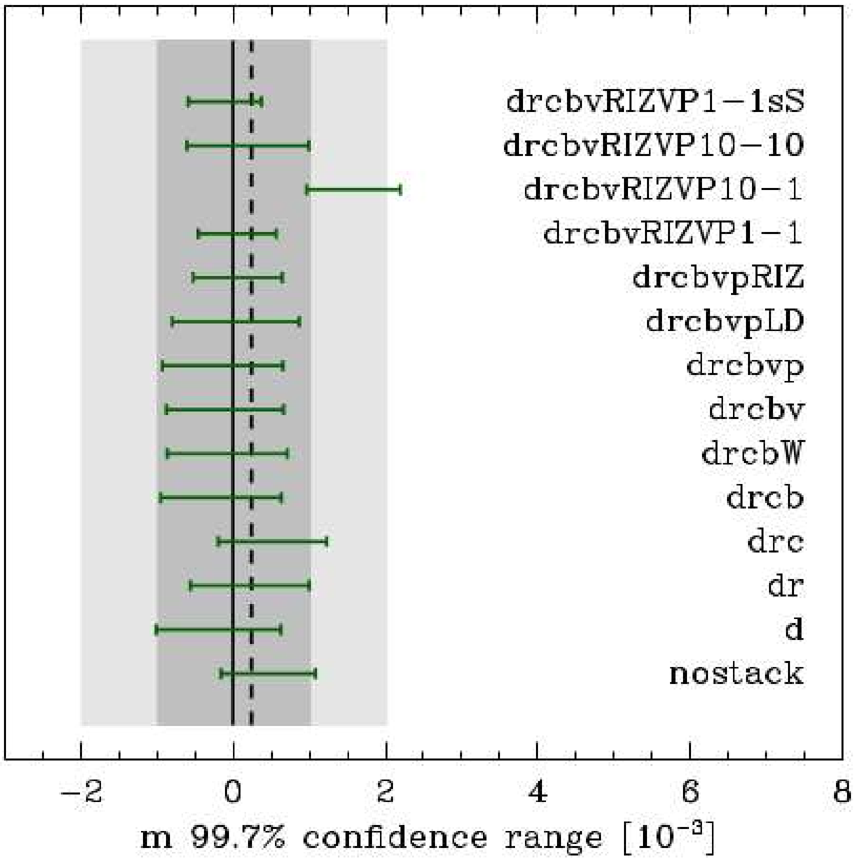


Figure 1. 99.7% confidence range for the multiplicative bias m in various simulations. The light gray region represents the total error budget for LSST, the dark gray region represents our target. The dashed line is the expected bias due to second order shear effects. Each point represents the m value measured in a particular simulation. To the right of each point is a code representing the simulation features used, which are nostack: the LSST DM stack was not used; d: dithers; r: rotations; c: cosmic rays; b: bad columns; W: realistic galaxy properties and noise using the WeakLensingDeblending package; v: variable pixel scale and WCS shear; p: psf $g_2 = 0.02$; RIZ: r, i and z bands used; LD: large dithers; VPX-Y: spatially variable moffat PSF with X times the expected variation for LSST, and Y epochs per band; s: stars included at 2/sq arcminute, S: star masks and bleeds included.

Example of something that needs exploration

- High stellar density is causing a bias
- Not clear yet what the cause is: for METACALIBRATION we would expect stars to produce a negative bias at much lower amplitude.

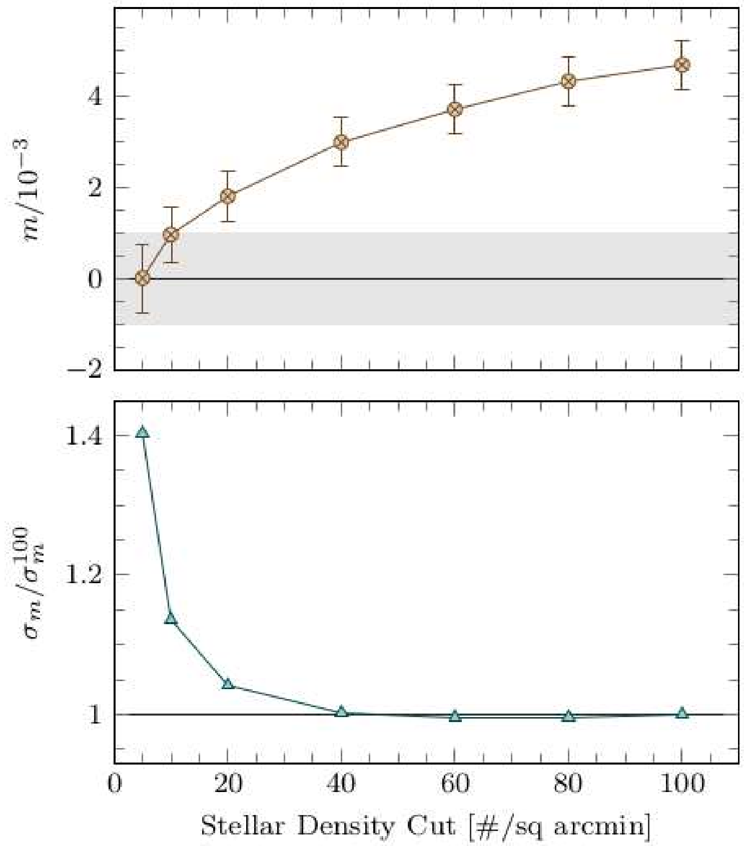


Figure 4. Metadetection performance as a function of maximum stellar density. Simulations were run with the expected stellar density distribution for the full LSST 18,000 square degree survey, with a maximum allowed value of 100/sq. arcminute. Stars bright enough to saturate were not included. Top panel: the multiplicative bias m as a function of the stellar density cut. Bottom panel: increase in the uncertainty on the mean shear, relative to a cut at 100/sq. arcminute, as a function of the stellar density cut. Note that many lensing probes are dominated by cosmic variance, which is not included in these uncertainties.

Summary

- It will be extremely challenging to utilize the full statistical power of the LSST data set.
- We are doing our best to apply the lessons learned from precursor surveys to LSST data processing.
- The weak lensing work is proceeding quickly and the results so far look very promising.