DM Plans for Blended Objects

Jim Bosch, Princeton University

LSST2016 PROJECT AND COMMUNITY WORKSHOP

AUGUST 15-19, 2016 | TUCSON, ARIZONA

What Makes Blending Hard

- Unknown per-object models.
- Computational scaling for large blends ("percolation").
- Lack of isolated stars in single-visit images.
- False/junk detections.
- Small scale "background" features.

Kinds of Blends

	Crowded Stellar Fields	Deep High-Latitude Fields
Unknown Models	none	significant
Percolation	significant	moderate
Lack of Isolated Stars	significant	none
False Detections	moderate	moderate
Background Features	irregular, can actually be foreground (e.g. dust lanes)	smooth, extended (e.g. ICL)
State of the Art	mostly solved (outside LSST)	mostly unsolved (anywhere)

DM Plans for Crowded Stellar Fields

Requirements vs. Expectations

- Processing crowded fields is formally a *best effort* task for DM.
- We're nevertheless very serious about doing it well; the community should expect us to do no worse than what they'd expect to get from running state-of-the-art third-party codes.

Why not D[A]OPHOT?

We're *probably not* going to use an existing third-party code.

- We need something we can integrate closely with the rest of our Python/C++ codebase.
- We need something that transitions smoothly to a high-latitude deblender.
- We may need something faster and/or less memory-intensive.

DM Plans for Crowded Fields

- We plan to do both photometry and astrometry in crowded fields, almost certainly via some kind of simultaneous fitting to individual epochs.
- Even when direct simultaneous fitting fails, we expect to be able to do transient/variable science and at least some astrometry via difference imaging.

DM Plans for Crowded Fields

- None of this exists yet, and we don't yet know when we will be implement it (a big DM replan is underway).
- I'm most worried about:
  - getting good PSF models when there aren't (m)any isolated stars
  - irregular nebulosity and dust
  - computational performance

Transitions

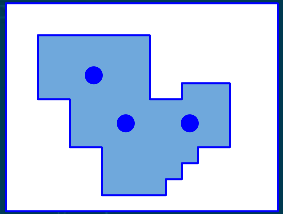
- A key trick for any high-latitude deblender is to identify stars early instead of fitting them with complex galaxy models.
- As we increase the prior probability of identifying an object as a star, any high-latitude deblender transitions to a crowded stellar field algorithm.

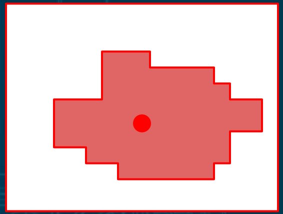
What We're Doing Now

It's a lot like what SDSS did, and it worked well there. It doesn't work nearly as well at LSST depths.

Single-Pass Processing

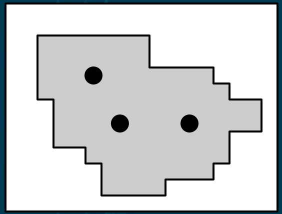
1. Detect: find above-threshold footprints and peaks in all images.
2. Associate: merge footprints and peaks into a single consistent set.
3. Deblend: apportion flux to each peak, generating children.
4. Measure: run algorithms on child pixels.





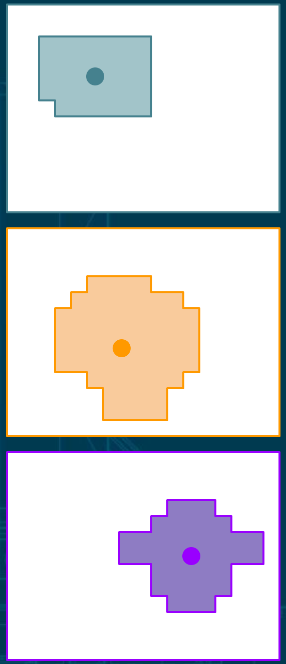
Single-Pass Processing

1. Detect: find above-threshold footprints and peaks in all images.
2. Associate: merge footprints and peaks into a single consistent set.
3. Deblend: apportion flux to each peak, generating children.
4. Measure: run algorithms on child pixels.



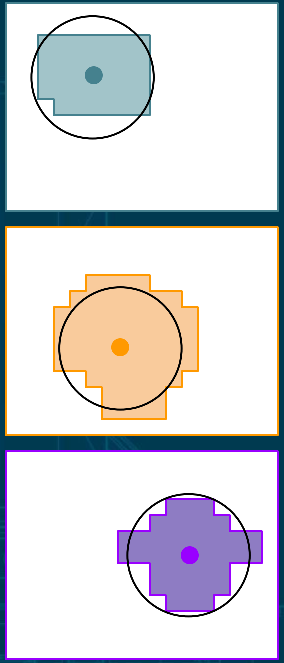
Single-Pass Processing

1. Detect: find above-threshold footprints and peaks in all images.
2. Associate: merge footprints and peaks into a single consistent set.
3. Deblend: apportion flux to each peak, generating children.
4. Measure: run algorithms on child pixels.



Single-Pass Processing

1. Detect: find above-threshold footprints and peaks in all images.
2. Associate: merge footprints and peaks into a single consistent set.
3. Deblend: apportion flux to each peak, generating children.
4. Measure: run algorithms on child pixels.



Single-Pass Processing

- Doesn't work in crowded stellar fields: need to subtract bright stars before trying to detect faint stars.
- Doesn't work around bright galaxies: wings push noise peaks above detection threshold.
- Measurement errors don't account for uncertainty in deblending.

Simultaneous Fitting

- Instead of (or in addition to) reapportioning flux, "just" fit models simultaneously to all objects in a blend.
  - What model for galaxies?
  - Fit bands jointly together?
  - How do we represent covariance?
  - We can also use these models to do the flux reapportioning.
- Computationally expensive.
- Requires good models and probably good priors.

DM Plans for Model Fitting

- Sample from the full, multi-epoch posterior of a galaxy model in at least griz for each object.
  - *maybe* jointly across blended objects.
  - *maybe* jointly across bands.
- Fit the flux, position, proper motion, and parallax of each object assuming point source morphology.
  - *maybe* jointly across blended objects.
  - *maybe* jointly across bands.
  - *maybe* sample here too.

Here There Be Dragons

Beyond this point, it's all conjecture. The slides in this section do not represent official DM plans. They're just things people in DM want to try.

Hybrid Models

- If we don't want to assume which objects are galaxies and which are stars, how do we fit them simultaneously?
  - Define a model that can transition between a galaxy (as radius -> 0) and moving point source (as proper motion -> 0);
  - sample jointly with this hybrid model: posterior chooses which model gets more attention.
  - In the limit of perfect galaxy models, this is an optimal star/galaxy classifier.
  - In stellar fields, just increase P(star).

Using Colors

Galaxies don't have the same morphology in every band...

...but they don't change much either.

If we fit jointly across bands, we need a prior for how morphologies differ across bands.

It might not have to be a very good prior to work.

Fixing Detection

- Stellar fields: "just" subtract detected stars and iterate.
  - Requires good PSF models, including the wings.
- Near bright galaxies:
  - Intentionally oversubtract the background before detection, then add it back in? (code already exists in DM stack)
  - Fit simple models or elliptical isophotes to brightest galaxies and subtract them?
  - Throw away junk detections later, when we have models for bright objects?

Multi-Scale Deblending

The SDSS approach worked for SDSS. If we degraded our data to SDSS quality, it would work for us too.

- This isn't crazy: we could do it *in addition* to deblending the full-quality data.
- Puts a bound on how badly we'll handle the biggest, brightest galaxies.
- The difference between the degraded processing and the full-quality processing could be an important source of diagnostic.
- The degraded processing could provide a good starting point for the full-quality processing...

Multi-Scale Deblending

Maybe we don't have to actually degrade our data:

1. Bin images by a factor of N (after low-pass filter to preserve sampling). *This doesn't change SNR much for objects larger than the bin size.*
2. Detect with a high threshold.
3. Deblend and model detections.
4. Subtract models from the *original* image.
5. Repeat with $N_{new} < N$ and a lower detection threshold, until N = 1 and detection is full-depth.

Multi-Scale Deblending

The SDSS approach worked for SDSS. If we degraded our data to SDSS quality, it would work for us too. But maybe we don't have to actually degrade it:

1. Bin images them by a factor of N (after low-pass filter to preserve sampling).
2. Detect with a high threshold to detect the biggest, brightest objects.
3. Deblend and model detections.
4. Subtract models from the *original* image.
5. Repeat with N -> N / M.

Include and update models from previous scales to correct biases?

How To Help

Don't write a deblender

...unless you're very, very ambitious.

And if you do, please contact us about your plans and work closely with DM developers (you don't have to work for DM to work on our codebase).

Do investigate detection

The DM stack's current detection algorithms aren't too far off from what we expect to do in the future, and this stage is critical for getting a decent starting "hypothesis" for the deblender (how many peaks and where).

- Run our detection code on datasets your care about.
- Test out the background-oversubtraction-reinsertion options, tweak the parameters.
- Try to cook up other easy and robust ways to identify junk detections.

Don't forecast prematurely

We know people want to study how LSST processing will interact with their science cases, but the deblending algorithms in the DM stack are simply not ready for this.

Think of the current deblender as more of a placeholder than a prototype, and if it doesn't do what you think it should, it's a Known Issue.

Do think about test cases

While you can't use the current DM stack to forecast how it will process the blends you care about now, you can make it easier in the future.

- Identify test datasets (simulations or real data). Include PSF models (etc).
- Define metrics that test how well the dataset has been deblended.
  - can use fields from catalogs generated by DM code (e.g. specific fluxes, centroids, or shapes)
  - can use images of reapportioned pixels
- Submit your tests to DM: if they're easy for us to use, they'll be what we try to optimize when we work on these algorithms.

Time for Questions?