S15 Multi-Band Coadd Processing Prototype

Motivation

One of the DRP's tasks in S15 is to backport the new multi-band processing scheme we've been using on the HSC side. This is a preliminary approach to deblending and measuring objects consistently across all bands. We believe it represents a clear improvement over our current way of doing things, and while it's not the way we plan to do this sort of processing in operations, it's also closer to what we envision than what we have at present in terms of of the structure of the processing. This will represent a major change to tasks we use to handle coadds, and we're *hoping* to bring it over largely as-is at least at first, with improvements delayed until the HSC fork is synchronized with the mainline LSST stack.

(Refer to Deep Processing for the larger context for this proposal.)

Overview

At present, we detect, deblend, and measure sources on coadds in each band separately. We can then do forced photometry from one reference band to any other band, either on individual sensors or coadds from other bands. None of the forced photometry will be deblended, however, as we don't have a way to transform the HeavyFootprints from the reference WCS/PSF to the measurement WCS/PSF.

With the new approach on the HSC side, we introduce a new cross-band "footprint merge" stage immediately after detection, and a new cross-band "reference selection" stage after measurement. The overall processing flow looks something like this:

1. Detect separately in each band, generating parent footprints.
2. Merge parent detection footprints and peaks from all bands, tracking which band(s) a peak originated in.
3. Deblend and measure separately in each band, guaranteeing that each peak will generate a child source (and thus guaranteeing that all bands will have the exact same sources).
4. Merge the catalogs from all bands, choosing one band for each object to serve as the reference band for that object, by searching for the first band in which the object was detected from a priority-sorted list of bands.
5. Run forced photometry on other coadds using the reference measurements to provide centroids and shapes, but deblending via the HeavyFootprints from the measurement band's previous deblending.
6. Run forced photometry on individual sensors as before, using the reference measurements (but still no deblending).

Already, this procedure gives us the following improvements:

- consistent deblending in all bands
- deblended forced photometry on coadds (and hence reasonable colors for blended objects)
- measurements of drop-out objects that are not detected in the highest-priority band.

In the future, we expect the pipeline to continue to follow this overall procedure, though individual steps may change considerably:

- The multiple detection coadds won't (just) be those from each band; we may have chi^2 coadds, coadds optimized for different SEDs, coadds optimized for seeing vs. depth, etc.
- The merge stage will get considerably more sophisticated, in terms of being able to identify corresponding peaks and discard invalid ones via properties saved in them during detection.
- The deblend and merge stages may need to be iterated.
- The deblend stage may be run on all bands available simultaneously, and/or on data units smaller than a patch.
- We may come up with better ways to do deblending on coadd forced photometry.

New Tasks

The HSC versions of all of these can be found here.

DetectCoaddSourcesTask

Detect sources (generate Footprints for parent sources) and model background for a single band.

Inputs:

- `deepCoadd`{tract,patch,filter}: ExposureF

Outputs:

- `deepCoadd_det`{tract,patch,filter}: `SourceCatalog` (only parent `Footprints`)
- `deepCoadd_calexp_det`{tract,patch,filter}: `ExposureF`
- `deepCoadd_calexp_detBackground`{tract,patch,filter}: `BackgroundList`

Data Unit: tract, patch, filter

Implementation: Should be able to delegate almost everything to the existing `SourceDetectionTask`.

The only reason we're writing a new coadd image here is to update the background and the detection mask plane; it'd be more efficient to just run detection and background modeling at the end of `AssembleCoaddTask` and just write one coadd dataset. But we'll defer trying to string those together (as we do on the HSC side) until we have more powerful parallelization primitives on the LSST side.

MergeDetectionsTask

Merge `Footprints` and `Peaks` from all detection images into a single, consistent set of Footprints and Peaks

Inputs:

- `deepCoadd_det`{tract,patch,filter}: `SourceCatalog` (only parent `Footprints`)

Outputs:

- `deepCoadd_mergeDet`{tract,patch}: `SourceCatalog` (only parent `Footprints`)

Data Unit: tract, patch

Most of the work here will be delegated to C++ code in afw (already transferred over, but due for its own refactoring). The output source catalog will contain only parent sources with IDs, `Footprints`, and flags indicating which bands contributed to the source. The `Peaks` in the `Footprints` will have additional flags indicating their origin.

MeasureMergedCoaddSourcesTask

Deblend and measure on per-band coadds, starting from consistent `Footprints` and `Peaks` for parent objects.

Inputs:

- `deepCoadd_mergeDet`{tract,patch}: `SourceCatalog`
- `deepCoadd_calexp_det`{tract,patch,filter}: `ExposureF`

Outputs:

- `deepCoadd_meas`{tract,patch,filter}: `SourceCatalog`

Data Unit: tract, patch, filter

We should be able to delegate nearly everything to existing the `SourceDeblendTask` and `SourceMeasurementTask`. We need to update the deblender to ensure it preserves all peaks and deblends as PSFs any `Peaks` that don't seem to appear in the band its working on; that's already in progress, but it does lead to some occasional bad behavior that we'll eventually have to fix rather than work around (as the deblender was written to be able to sometimes give up on `Peaks`, and that's no longer an option).

MergeMeasurementsTask

Combine separate measurements from different bands into a catalog suitable for driving forced photometry. Essentially, it must have a centroid, shape, and CModel fit for all objects, even for objects that were not detected on the canonical band. Will assume that all input catalogs already have consistent object lists.

Inputs:

- `deepCoadd_meas`{tract,patch,filter}: SourceCatalog

Outputs:

- `deepCoadd_ref`{tract,patch}: SourceCatalog

Data Unit: tract, patch

Migration

Adding these new tasks and the datasets they'll utilize doesn't mean we'll have to remove processCoadd.py immediately, but it does mean the new tasks probably shouldn't be run in the same output data repository as processCoadd.py (while they no longer share the `deepCoadd_calexp` dataset as we have changed the naming of this output here to deepCoadd_calexp_det, it is possible we may want to save others if, for example, we wanted to appropriate `deepCoadd_src` for one of the new datasets proposed here). I think we will want to remove processCoadd.py eventually.

Because the forced photometry tasks have changed in a different way on the LSST side (with the introduction of meas_base), the changes to do deblended forced photometry on coadds will not be transferred over directly; they'll have to be at least partially reimplemented. For everything else, I expect we'll just be able to cherry-pick and then make minor modifications afterwards (depending on how many changes are requested in the design review stage).

On the HSC side, we have two driver tasks for coaddition and multiband processing that use parallel processing primitives that aren't yet available on the LSST side. These combine `DetectCoaddSourcesTask` with `MakeCoaddTempExpTask` and `AssembleCoaddTask` (so we only have to write one coadd image, as noted above), and then combine the rest of these into a single driver. I'd eventually like to get to a similar position on the LSST side, but I think the transfer or reimplementation of those parallel processing primitives is a separate issue, so for now I'm assuming we'll just transfer all of these tasks individually. They'll still have the same functionality, but they'll rely more on I/O for communication and require more human intervention.