LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)

Proposal for Deblender Outputs as Level 2 Data Products

James Bosch

LDM-nnn

Latest Revision: 2017-03-31

revision: TBD
status: draft

Abstract

Most catalog measurements produced in a LSST Data Release Production will be derived from pixel values that have been *deblended* – we have apportioned the flux in each pixel to each of the Objects that overlap that pixel. These deblended pixels values are thus a critical link in our provenance chain, and should be provided to users as a first-class data product. This document provides further justification for the proposal to add this data product and provides a preliminary estimate of the impact on data storage requirements.

Change Record

Version	Date	Description	Owner name
1	2017-03-31	Initial release.	James Bosch

Contents

1 Introduction

Essentially all measurements in LSST's *Object* table [**?**] are directly or indirectly derived from running the "deblender" algorithm on coadd images. The deblender apportions the flux in each pixel between the Objects that overlap that pixel (with Object extent defined by a surface-brightness threshold). Subsequent measurements on the coadds themselves use a combination of the original pixels and the deblended pixels as their inputs. We will also use the deblendd pixels indirectly for Forced Source measurements and the Bulge-Disk Model and Moving Point Source Models that are fit to individual per-epoch images, as these measurements will be initialized from coadd processing and may utilize per-epoch deblended pixels that are computed from the deblended coadd pixels.

These deblended pixel values are thus a critical link in the provenance change between the coadd images and the measurements in the *Object* and *Forced Source* tables. DM developers and science users (especially those interested in rare objects) will need to access deblended pixel values to understand any measurement outliers that may be due to deblending problems.

Deblended pixel values are also an important input to Level 3 algorithms that perform new measurements on individual Objects.

The deblender itself is a compuationally expensive algorithm that will run simultaneously on multiple coadds. It will likely use some combination of *deep* and *best-seeing* coadds, but it may utilize *short-period* or *PSF-matched* coadds as well.

We also run a variant of the deblender on visit-level images prior to running the measurements that populate the *Source* table. This variant requires only the CCD-level processed visit image as input.

2 Proposal and Implementation Options

We propose adding deblended *deep* coadd pixel values (in each band) for each object as a first-class Level 2 data product. Our recommendation for the data structure for this information is one `HeavyFootprint` for each object. A `HeavyFootprint` combines a run-length encoding of the region covered by an object (which we call a `Footprint`) with a 1-d array of 32-bit floating point values containing just the pixels in that region. `Footprints` themselves are small: three 32-bit integers for each pixel *row* in the region. Objects that were not deblended (because they had no neighbors) need only have a `Footprint`; their `HeavyFootprint` can be generated on-the-fly efficiently from the `Footprint` and the coadd images.

Despite the fact that they are tied to objects, the access patterns for `HeavyFootprints` should be assumed to be more coupled to the access patterns for coadd images, but will be even less frequent: users who request `HeavyFootprints` will almost always want coadd images as well (though the converse is not true). We anticipate that in most cases, these requests will be for single objects (and cut-outs of coadd images near them), not all objects in a field.

We do not believe regenerating `HeavyFootprints` by re-running the deblender on-demand is a viable way to implement this proposal; the deblender's computational performance and large inputs would make that approach both inefficient and high-latency in access patterns that involve rare objects. The deblender algorithm *may* permit a highly compressed output (e.g. an analytic model) that allows the true `HeavyFootprints` to be regenerated from the coadd data (which would be an acceptable implementation), but this cannot be guaranteed today.

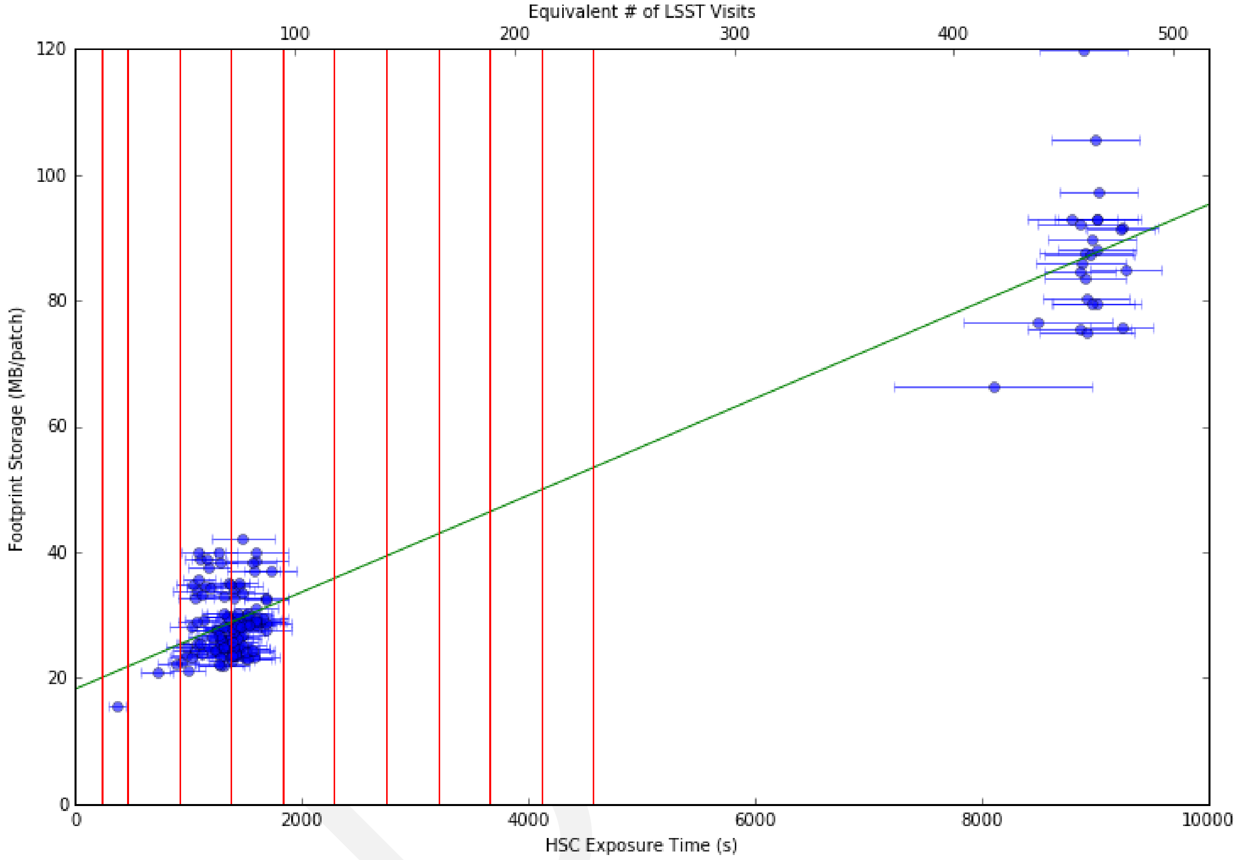
We do not propose storing `HeavyFootprints` for *best-seeing* or other coadd flavors besides *deep*. It is likely these could be derived efficiently from those coadd images and the *deep* coadd `HeavyFootprints`, but since most measurements will be done on the *deep* coadds we anticipate the need for deblender outputs for other flavors to be much less.

We also do not propose storing `HeavyFootprints` for *Source*. We expect `Source` measurements to be much more lightly used than *Object* or *Forced Source* measurements, and *Source* `HeavyFootprints` can be more efficiently regenerated because are produced by a simpler, faster form of the deblender.

3 Impact on Storage

Because the pixels included in the set of `HeavyFootprints` are defined by a surface-brightness threshold, the storage costs for `HeavyFootprints` are a function of depth. We have attempted to estimate this scaling by using LSST DM Stack processing of Hyper-Suprime Cam Subar Strategic Program Survey (HSC SSP) data. The HSC SSP data includes both a Wide layer ($1200\,s$

in i) and an UltraDeep layer ($9000\,s$ in i to date).



As shown in Figure 3, we fit a linear regression to the total storage cost of the `Footprints` and `HeavyFootprints` 117 patches (each $0.03484\,\mathrm{deg}^2$) of Wide data and 25 patches of UltraDeep data. While the scaling is likely nonlinear, we do not currently have processed data available from the intermediate SSP Deep fields that could constrain a higher-order fit. In the current DM stack, the size of the `HeavyFootprints` in all bands are mostly determined by the depth of the deepest band. This should remain at least approximately true in future versions of the pipeline.

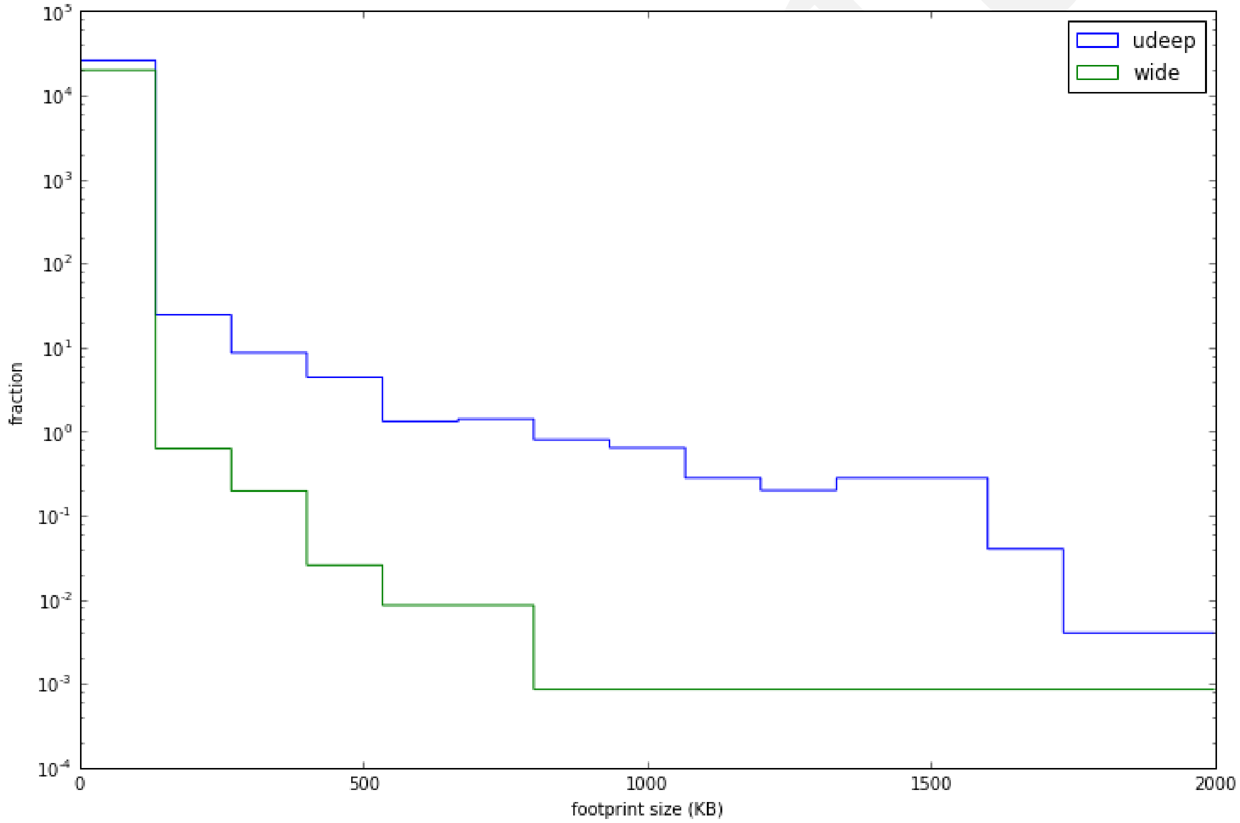
The regression yields the following formula:

$$\frac{\mathrm{storage}}{(\mathrm{MB})(N_{\mathrm{patches}})(N_{\mathrm{bands}})} = N_{\mathrm{visits},i} \times 0.149 + 18.26$$

The conversion from HSC exposure time to LSST visits assumes 30s for each LSST visit and simply accounts for the difference in effective primary mirror size.

The patch area of $0.03484\,\mathrm{deg}^2$ represents only the inner, non-overlapping area of the patch, but the footprint storage estimate comes from the full area of $0.03841\,\mathrm{deg}^2$. The inner area should be used when estimating the number of patches from the area of a survey, which will automatically account (in part; tracts also overlap) for the current overlap fraction.

The effect of the difference in seeing between HSC and LSST is not clear; LSST's larger PSF will both increase the area covered by bright objects and prevent objects at the edge of HSC's detection limit from being detected at all. HSC has smaller ($0.168\,\mathrm{arcsec}$) pixels as well, however, which should make this an overestimate (and hence a conservative estimate) for LSST storage costs.



It is also worth noting that the dynamic range of per-object `HeavyFootprint` storage sizes spans several orders of magnitude, as shown in Figure 3. Storage cannot be assumed to be even approximately constant across objects.