LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)

Data Management Applications Design

Mario Juric, Robert Lupton, Tim Axelrod et al.

LDM-151

October 7, 2013

Change Record

Version	Date	Description	Owner name
1	3/26/2009	Initial version as Document-7396	Tim Axelrod, et. al
1.2	3/27/2009	Minor edits	Tim Axelrod
1.3	4/17/2009	General edits and updates	Tim Axelrod
1.4	5/8/2009	Explicit reference to multifit added to Section 6.1	Tim Axelrod
1.5	2/11/2010	General edits and updates; generated from SysML model	Jeff Kantor
2	8/4/2011	Elevated to LDM handle; general updates and edits	Tim Axelrod
3	10/7/2013	Updates for consistency with FDR baseline	Mario Juric
WIP	2015	Work in progress as part of 2015 replanning exercise	Lim, Bosch, Economou, Krughoff, Swinbank, et al.

Large Synoptic Survey Telescope
Data Management Applications Design

Mario Jurić*, R.H. Lupton, T. Axelrod, J.F. Bosch, P.A. Price,
G.P. Dubois-Felsmann, Ž. Ivezić, A.C. Becker, J. Becla,
A.J. Connolly, J. Kantor, K-T Lim, D. Shaw,
for the LSST Data Management

Friday 27$^{\rm th}$ May, 2016, 18:47hrs

*Please direct comments to <mjuric@lsst.org>.

Abstract

The LSST Science Requirements Document (the LSST SRD) specifies a set of data product guidelines, designed to support science goals envisioned to be enabled by the LSST observing program. Following these guidlines, the details of these data products have been described in the LSST Data Products Definition Document (DPDD), and captured in a formal flow-down from the SRD via the LSST System Requirements (LSR), Observatory System Specifications (OSS), to the Data Management System Requirements (DMSR). The LSST Data Management subsystem's responsibilities include the design, implementation, deployment and execution of software pipelines necessary to generate these data products. This document, in conjunction with the UML Use Case model (LDM-134), describes the design of the scientific aspects of those pipelines.

Contents

1 Preface

The purpose of this document is to describe the design of pipelines belonging to the Applications Layer of the Large Synoptic Survey Telescope (LSST) Data Management system. These include most of the core astronomical data processing software that LSST employs.

The intended audience of this document are LSST software architects and developers. It presents the baseline architecture and algorithmic selections for core DM pipelines. The document assumes the reader/developer has the required knowledge of astronomical image processing algorithms and solid understanding of the state of the art of the field, understanding of the LSST Project goals and concepts, and has read the LSST Science Requirements (SRD) as well as the LSST Data Products Definition Document (DPDD).

This document should be read in conjunction with the LSST DM Applications Use Case Model (LDM-134). They are intended to be complementary, with the Use Case model capturing the detailed (inter)connections between individual pipeline components, and this document capturing the overall goals, pipeline architecture, and algorithmic choices.

Though under strict change control[1], this is a ***living document***. Firstly, as a consequence of the "rolling wave" LSST software development model, the designs presented in this document will be refined and made more detailed as particular pipeline functionality is about to be implemented. Secondly, the LSST will undergo a period of construction and commissioning lasting no less than seven years, followed by a decade of survey operations. To ensure their continued scientific adequacy, the overall designs and plans for LSST data processing pipelines will be periodically reviewed and updated.

[1]LSST Docushare handle for this document is `LDM-151`.

2 Introduction

2.1 LSST Data Management System

To carry out this mission the Data Management System (DMS) performs the following major functions:

- Processes the incoming stream of images generated by the camera system during observing to produce transient alerts and to archive the raw images.

- Roughly once per year, creates and archives a Data Release ("DR"), which is a static self-consistent collection of data products generated from all survey data taken from the date of survey initiation to the cutoff date for the Data Release. The data products (described in detail in the DPDD), include measurements of the properties (shapes, positions, fluxes, motions, etc.) of all detected objects, including those below the single visit sensitivity limit, astrometric and photometric calibration of the full survey object catalog, and limited classification of objects based on both their static properties and time-dependent behavior. Deep coadded images of the full survey area are produced as well.

- Periodically creates new calibration data products, such as bias frames and flat fields, that will be used by the other processing functions, as necessary to enable the creation of the data products above.

- Makes all LSST data available through interfaces that utilize, to the maximum possible extent, community-based standards such as those being developed by the Virtual Observatory ("VO"), and facilitates user data analysis and the production of user-defined data products at Data Access Centers ("DAC") and at external sites.

The overall architecture of the DMS is discussed in more detail in the Data Management System Design (DMSD) document. The overall architecture of the DMS is shown in Figure 1.

This document discusses the role of the Applications layer in the first three functions listed above (the functions involving *science pipelines*). The fourth is discussed separately in the SUI Conceptual Design Document (SUID).

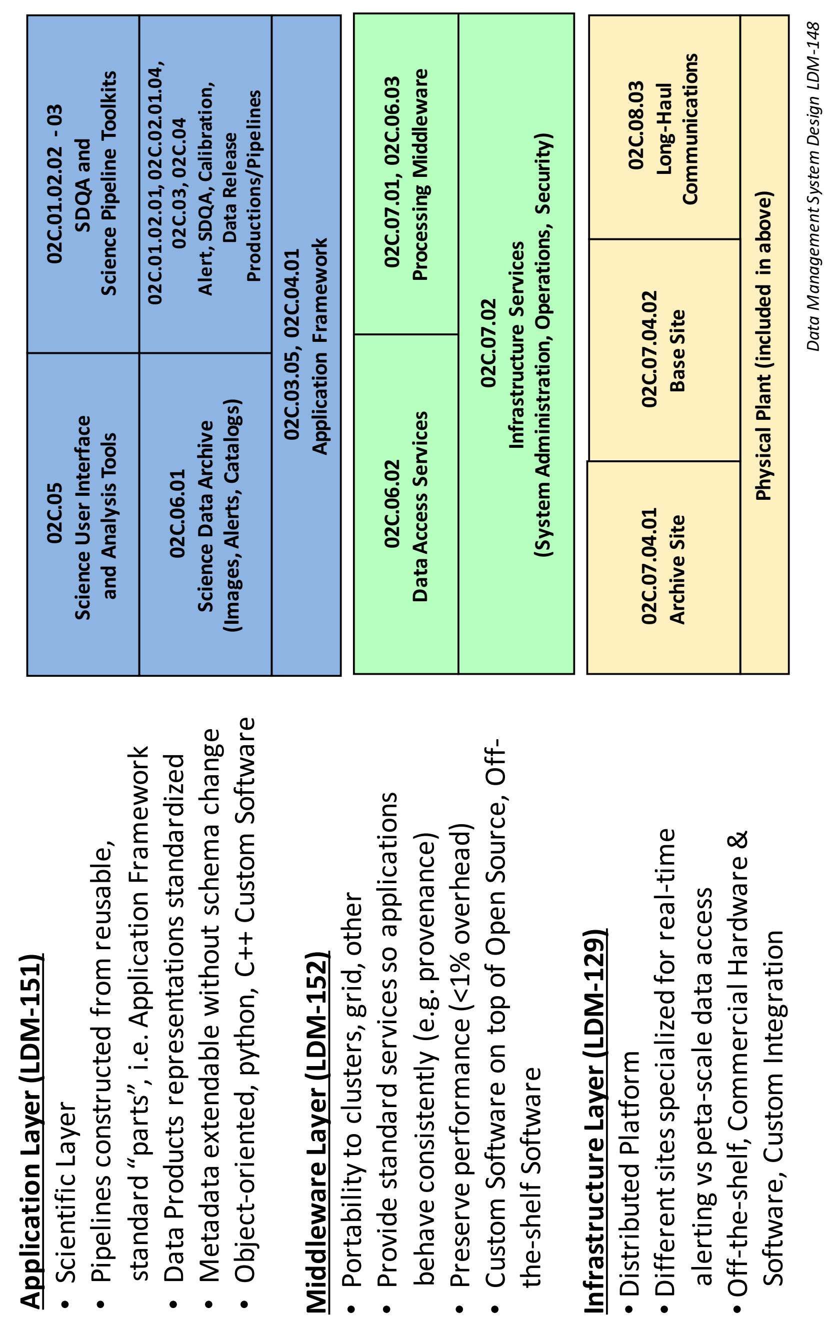

Figure 1: Architecture of the Data Management System

	IMAGES CATEGORY (FILES)	CATALOG CATEGORY (DATABASE)	ALERT CATEGORY (DATABASE)
NIGHTLY Level 1	Raw science images Calibrated science images Subtracted science images Data quality analysis	Catalog of sources (detections) found on difference images Catalog of objects found on difference images Orbit catalog Data quality analysis	Transient alert Data quality analysis (within 60 seconds)
DATA RELEASE (Annual) Level 2	Stacked science images Calibration images RGB JPEG images Data quality analysis	Source (detection) catalog from calibrated science images Object catalog from optimally measured properties Data quality analysis	Alert statistics and summaries based on annual reprocessing Data quality analysis

Figure 2: Organization of LSST Data Products

2.2 Data Products

The LSST data products are organized into three groups, based on their intended use and/or origin. The full description is provided in the Data Products Definition Document (DPDD); we summarize the key properties here to provide the necessary context for the discussion to follow.

- **Level 1** products are intended to support timely detection and follow-up of time-domain events (variable and transient sources). They are generated by near-real-time processing the stream of data from the camera system during normal observing. Level 1 products are therefore continuously generated and / or updated every observing night. This process is of necessity highly automated, and must proceed with absolutely minimal human interaction. In addition to science data products, a number of related Level 1 "SDQA"[2] data products are generated to assess quality and to provide feedback to the Observatory Control System (OCS).

- **Level 2** products are generated as part of a Data Release, generally performed yearly, with an additional data release for the first 6 months of survey data. Level 2 includes data products for which extensive computation is required, often because they combine information from many exposures. Although the steps that generate Level 2 products will be automated, significant human interaction may be required at key points to ensure the quality of the data.

- **Level 3** products are generated on any computing resources anywhere and then stored in an LSST Data Access Center. Often, but not necessarily, they will be generated by users of LSST using LSST software and/or hardware. LSST DM is required to facilitate the creation of Level 3 data products by providing suitable APIs, software components, and computing infrastructure, but will not by itself create any Level 3 data products. Once created, Level 3 data products may be associated with Level 1 and Level 2 data products through database federation. Where appropriate, the LSST Project, with the agreement of the Level 3 creators, may incorporate user-contributed Level 3 data product pipelines into the DMS production flow, thereby promoting them to Level 1 or 2.

[2] Science Data Quality Analysis

[**Note:**
JFB comment: Should we mention intermediate data products as a category here, too? Those will be an important part of the Pipeline descriptions that happen later in the document.]

The organization of LSST Data Products is shown in Figure 2.

Level 1 and Level 2 data products that have passed quality control tests will be accessible to the public without restriction. Additionally, the source code used to generate them will be made available, and LSST will provide support for builds on selected platforms.

2.2.1 Data Units

[**TODO:**
This section still in outline form.]

- Source vs. Object
- Visit, CCD, Snap
- Tract, Patch
- Filters and object SEDs

2.3 Science Pipelines Organization

[**TODO:**
This section still in outline form. Expanded form should be short; I'm proposing we move the longer overviews that used to be in the introductions into the beginning of the sections for each Production, and shorten them to little more than a description of that Production's overview diagram.]

- In sections (bla) we describe Alert Production, Calibration Product Production, and Data Release Production by decomposing them Pipelines. Pipelines are not reusable; they occupy a specific place in a Production.
- In section (bla) we describe the SDQA system, a set of smaller Productions as well as independent Pipelines within AP and DRP that verify the quality of the algoritms and data.

- In section (bla) we say something with scope TBD about SUIT and enabling Level 3.
- Pipelines are composed of reusable Algorithmic Components, which are described in section (bla).
- Algorithmic Components rely on shared software primitives, described in section (bla).

3 Level 1 Pipelines

3.1 Single Frame Processing Pipeline (WBS 02C.03.01)

3.1.1 Key Requirements

Single Frame Processing (SFM) Pipeline is responsible for reducing raw image data to *calibrated exposures*, and detection and measurement of `Sources` (using the components functionally a part of the Object Characterization Pipeline).

SFM pipeline functions include:

- Assembly of per-amplifier images to an image of the entire CCD;
- Instrumental Signature Removal;
- Cosmic ray rejection and snap combining;
- Per-CCD determination of zeropoint and aperture corrections;
- Per-CCD PSF determination;
- Per-CCD WCS determination and astrometric registration of images;
- Per-CCD sky background determination;
- Source detection.

Calibrated exposure produced by the SFM pipeline must possess all information necessary for measurement of source properties by single-epoch Object Characterization algorithms.

It shall be possible to run this pipeline in two modes: a “fast” mode needed in nightly operations for Level 1 data reductions where no source characterization is done beyond what’s required for zero-point, PSF, sky, and WCS determination (image reduction); and a “full” mode that will be run for Level 2 data reductions.

3.1.2 Baseline Design

Single Frame Processing pipeline will be implemented as a flexible framework where different data can be easily treated differently, and new processing steps can be added without modifying the stack code.

It will consist of three primary components:

- A library of useful methods that wrap a small number of atomic operations (e.g., `interpolateFromMask`, `overscanCorrection`, `biasCorrection`, etc.)
- A set of classes (`Tasks`) that perform higher level jobs (e.g., `AssembleCcdTask`, or `FringeTask`), and a top level class to apply corrections to the input data in the proper order. This top level class can be overridden in the instrument specific `obs_*` packages, making the core SFM pipeline camera agnostic.
- A top-level Task to run the SFM pipeline.

In the paragraphs to follow, we describe the adopted baseline for key SFM algorithms. If not discussed explicitly, the algorithmic baseline for all other functionallity is assumed to be the same as that used by SDSS *Photo* pipeline [15].

3.1.2.1 Instrumental Signature Removal: The adopted pipeline design allows for straightforward addition of correction for instrumental effects that will be discovered in the as-built Camera. The effects currently baselined to be corrected are:

- Bias: A master bias frame, created from a set of overscan corrected zero length exposures, is subtracted from the image to correct for 2D structure in the bias level. For each exposure, overscan columns will be averaged and fit with a 1D function and subtracted row by row to account for time variable bias level.
- Assembly: CCDs will be assembled by trimming the prescan and/or overscan rows and columns from amplifier images and storing them into a Image object.

- Dark current: A master dark frame created from a set of bias corrected exposures taken with the shutter closed is scaled to the science image exposure time and subtracted to correct for 2D structure in the dark current.

- Cross-talk: Cross talk is generated by interaction of fields produced by the current in physically proximate electronic components. This results in bright features from one amp showing up in other. Correction is to subtract each aggressor amp (possibly flipped) modulated by a measured coefficient from the victim amp. The implementation may assume the cross-talk is small enough to be correctable by first order correction only.

- Non-linearity: CCDs do not have perfectly linear response. At both almost empty and almost full well the response can become non-linear. Given a measurement of the linearity of the CCD response, along with any temperature dependence, the data values will be corrected to linear response by simple mapping and interpolation.

- Flat-field: The correction is a division by the normalized master flat. The master flat will be generated assuming a nominal flat spectrum for all sources. Photometric corrections will be applied downstream on a source by source basis given an SED for each source.

- Fringing: Fringe parerns are an interference effect that result from the sharp emission lines in the night sky spectrum. This effect is the strongest in redder bands. The best fit modeled fringe pattern, constructed from monochromatic flats assuming a low-dimensional parametrization of the night sky emission spectrum, will be subtracted from the image.

- Cosmic ray rejection and snap combining: Exposures will be taken in pairs separated by the readout time. The two images and the expected statistics on those images are used to reject pixels that are significant outliers. Once cosmic rays are flagged the two snaps will be added to produce an image with a longer effective exposure.

3.1.2.2 PSF determination: We will run separate algorithms to select candidate stars and determine the point-spread function (PSF, the light distribution for a point source, a critical ingredient to understanding the data and measuring accurate fluxes and shapes). Both the star selector and PSF determiner algorithms will be pluggable Python modules, so that different algorithms can be run as desired for different analysis needs.

Three selectors will be implemented. The "second-moment" star selector will build a histogram of the X and Y second moments of flux, search for a peak, and select sources in the peak as point source candidates. The "catalog" star selector, in contrast, will make use of an external catalog of point sources and use astrometric matching to select point source candidates. The "objectSize" star selector will identify point source candidates from the cluster of sources with similar sizes regardless of magnitude. When selecting point source candidates by size (i.e., for the "second-moment" and "objectSize" algorithms), the sizes will be corrected by the known optical distortion of the camera.

Given the irregularly sampled grid of PSFs represented by selected stars, the variation of the PSF across the CCD will be determined. The baselined "principal-components" PSF (pcaPsf) determiner performs a singular value decomposition (also known as a principal components analysis, or PCA) on the point-source candidates in pixel space to produce a set of eigen-images. Using the dominant eigen-images, it constructs polynomial interpolants for their relative weights. This produces a spatially-varying PSF model that captures the most important changes in the PSF over the CCD.

These algorithms are intended to be sufficient to enable Level 1 processing. More advanced PSF determination algorithms will be developed in the PSF Estimation Pipeline (WBS 02C.04.03).

3.1.2.3 Sky Background Determination: We will estimate the smooth sky background by measuring the background level in cells (typically 256 or 512 pixels square) using (by default) a clipped mean, and ignoring pixels that are part of detected sources. An interpolating spline (an Akima spline, by default) will be constructed to estimate the background level in each pixel. Backgrounds will be possible to estimate simultaneously over multiple sensors, including the full focal plane.

Background models will be saved, for later subtraction or restoration (e.g., in background matching, as implemented by the Coaddition Pipeline, WBS

02C.04.04).

3.1.2.4 WCS determination and image registration The absolute World Coordinate System (WCS) will be determined using an *astrometry.net* type algorithm [14], seeded with the approximate position of the boresight.

This module will also include the capability to perform relative registration of a set of images to enable coaddition and image differencing, using the *meas_mosaic* registration algorithm developed by Furusawa et al. [10] as the baseline.

3.1.3 Constituent Use Cases and Diagrams

Assemble CCD; Determine Aperture Correction; Determine PSF; Remove Instrument Signature; Detect Sources; Determine Photometric Zeropoint; Measure Single Visit Sources; Determine WCS; Sum Exposures, Combine Raw Exposures, Remove Exposure Artifacts; Determine Sky Background Model; Calibrate Exposure; Process Raw Exposures to Calibrated Exposure; Perform Single Visit Processing;

3.1.4 Prototype Implementation

A prototype implementation of all major components of SFM baseline design has been completed in LSST Final Design Phase. The achieved accuracy is comparable to state-of-the art codes today (e.g., SDSS, SExtractor). We expect it will be possible to transfer a significant fraction of the existing code into Construction, with continued improvement to meet LSST accuracy requirements.

WCS determination and image registration modules are an exception, and will require extensive redesign and rewrite. The sky determination module will have to be enhanced to support multi-CCD fitting capability.

The prototype codes are available in the following repositories: `https://github.com/lsst/ip_isr`, `https://github.com/lsst/meas_algorithms`, `https://github.com/lsst/meas_astrom`, `https://github.com/lsst-dm/legacy-meas_mosaic`, `https://github.com/lsst/pipe_tasks`.

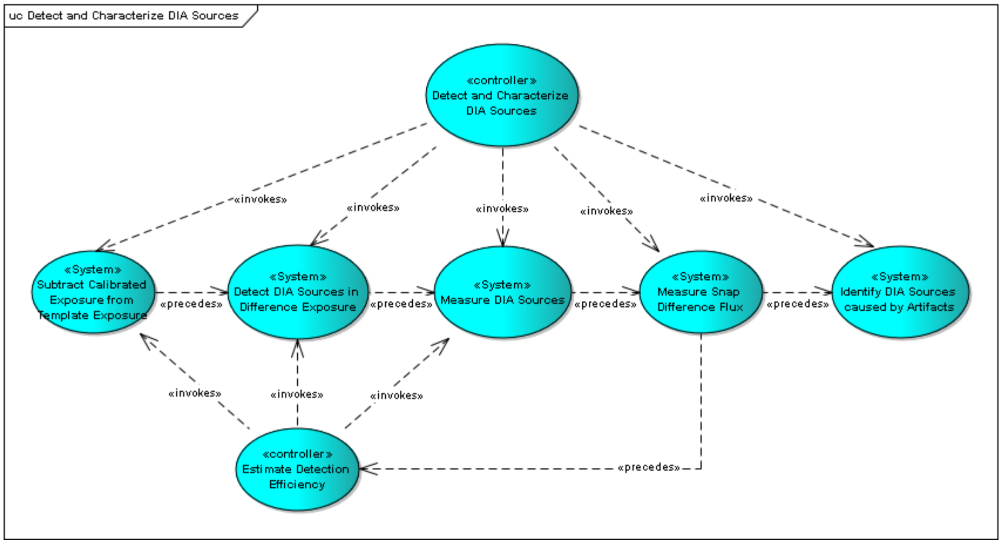

Figure 3: Image Differencing Pipeline Use Case Diagram

3.2 Image Differencing Pipeline (WBS 02C.03.04)

3.2.1 Key Requirements

The image differencing pipeline shall difference a visit image against a deeper template, and detect and characterize sources in the difference image in the time required to achieve the 60 second design goal for Level 1 alert processing (current timing allocation: 24 seconds). The algorithms employed by the pipeline shall result in purity and completeness of the sample as required by the DMSR . Image differencing shall perform as well in crowded as in uncrowded fields.

3.2.2 Baseline Design

The Image Differencing pipeline will difference, detect, and deblend objects in the resulting image using the “preconvolution” algorithm as described in Becker et al. [3]. Differencing will be performed against a deeper template, and differential chromatic refraction (DCR) will be handled by having templates in several bins of airmass.

All `DIASource` measurements described in the DPDD, including post-processing such as variability characterization, will be performed for all sources detected in this manner. The measurements will be performed on the pre-convolved likelihood image. This involves invoking some algorithms defined in the Object Characterization Pipeline (WBS 02C.04.06) as well as algorithms specific to difference images, which are defined in this WBS (see §3.2.2.1).

If necessary a *spuriousness metric* using machine-learning techniques (e.g., Bloom et al. [5]) will be developed to help in the discrimination of real sources from those caused by artifacts.

Details of this baseline design have been captured in the `Detect and Characterize DIA Sources` and related diagrams, presented in Figure 3.

3.2.2.1 Dipole model fit The pipeline shall be capable of fitting a dipole object model, as described in the DPDD. The baseline algorithm is analogous to that employed by the bulge-disk model fit (§**??**), but with the model being a mixture of positive and negative point sources, instead of Sérsic profiles.

3.2.3 Constituent Use Cases and Diagrams

Subtract Calibrated Exposure from Template Exposure; Identify DIA Sources caused by Artifacts; Perform Precovery Forced Photometry; Measure DIA Sources; Detect DIA Sources in Difference Exposure; Measure Snap Difference Flux; Perform Difference Image Forced Photometry; Calculate DIA Object Flux Variability Metrics; Fit DIA Object Position and Motion;

3.2.4 Prototype Implementation

A prototype implementation partially implementing the baseline design has been completed in the LSST Final Design Phase. It includes detection, centroiding, aperture and PSF photometry, and adaptive shape measurement. This implementation was used to benchmark the speed of the image differencing code and examine the expected levels of false positives. Deblending on difference images, fits to trailed sources, and dipole fits were not prototyped. The final report on prototype design and performance can be found in Becker et al. (`http://ls.st/x9f`).

The prototype code is available at `https://github.com/lsst/ip_diffim`. The current prototype, while functional, will require a partial redesign to be transfered to construction to address performance and extensibility concerns.

3.3 Association Pipeline (WBS 02C.03.02)

3.3.1 Key Requirements

The Association Pipeline has two key responsibilities: i) it must be able to associate newly discovered `DIASources` with previously known `DIAObjects` and `SSObjects`, and ii) it must be able to associate `DIAObjects` with known `Objects` from the Level 2 catalogs.

3.3.2 Baseline Design

The baseline design for `DIASources` to `DIAObject` association and `DIAObject` to `Object` association is to use simple nearest-neighbor search while taking proper motions and positional errors ellipses into account.

For matches to `SSObjects`, the `SSObject`'s ephemeris are to be computed by NightMOPS (functionally a part of the Moving Object Pipeline, WBS 02C.03.06). Matching to the computed ephemeris is to be performed as if they were `DIAObjects`.

When Level 1 data is reprocessed, a more sophisticated clustering algorithm [1] will be employed.

3.3.3 Constituent Use Cases and Diagrams

Create Instance Catalog for Visit; Associate with Instance Catalog; Perform DIA Object Association; Perform DIA Source Association;

3.3.4 Prototype Implementation

Prototype implementation of the baseline design has been completed in LSST Final Design Phase. The nearest-neighbor matching has been implemented as a part of the Application Framework, while clustering using OPTICS resides in the database-related ingest modules.

The prototype code is available at `https://github.com/lsst-dm/legacy-ap`. The current prototype, while functional, will require a partial redesign in Construction to address scalability and performance.

3.4 Alert Generation Pipeline (WBS 02C.03.03)

3.4.1 Key Requirements

Alert Generation Pipeline shall take the newly discovered `DIASources` and all associated metadata as described in the DPDD, and generate alert packets in `VOEvent` format. It will transmit these packets to VO Event Brokers, using standard IVOA protocols (eg., VOEvent Transport Protocol; VTP. End-users will primarily use these brokers to classify and filter events for subsets fitting their science goals.

To directly serve the end-users, the Alert Generation Pipeline shall provide a basic, limited capacity, alert filtering service. This service will run at the LSST U.S. Archive Center (at NCSA). It will let astronomers create simple filters that limit what alerts are ultimately forwarded to them. These *user defined filters* will be possible to specify using an SQL-like declarative language, or short snippets of (likely Python) code.

3.4.2 Baseline Design

The baseline design is to adopt and upgrade for performance and functionallity the Skyalert package (`http://lib.skyalert.org/skyalert/`).

3.4.3 Constituent Use Cases and Diagrams

Distribute to Subscribed Brokers; Distribute to Subscribed Users; Generate Alerts; Generate and Distribute Alerts;

3.4.4 Prototype Implementation

No prototype implementation has been developed by LSST, as the Skyalert package (`http://lib.skyalert.org/skyalert/`) was found to be mature enough to baseline the architecture and estimate costs.

3.5 Moving Object Pipeline (WBS 02C.03.06)

3.5.1 Key Requirements

The Moving Object Pipeline System (MOPS) has two responsibilities within LSST Data Management:

- First, it is responsible for generating and managing the Solar System[3] data products. These are Solar System objects with associated Keplerian orbits, errors, and detected `DIASources`. Quantitatively, it shall be capable of detecting 95% of all Solar System objects that meet the findability criteria as defined in the OSS . The software components implementing this function are known as ***DayMOPS***.

- The second responsibility of the MOPS is to predict future locations of moving objects in incoming images so that their sources may be associated with known objects; this will reduce the number of spurious transient detections and appropriately flag alerts to detections of known Solar System objects. The software components implementing this function are known as ***NightMOPS***.

3.5.2 Baseline Design

The baseline NightMOPS design is to adopt and adapt an existing ephemeris computation pipeline such as OrbFit[4] or OpenOrb[5]. The baseline DayMOPS design uses Kubica et al. [13] algorithms to identify and link Solar System object candidates.

The design of these components are explained in detail in the MOPS Design Document (MOPSD).

3.5.3 Constituent Use Cases and Diagrams

Process Moving Objects; Fit Orbit; Prune Moving Object Catalog; Perform Precovery; Recalculate Solar System Object Properties; Link Tracklets into Tracks; Find Tracklets;

---

[3] Also sometimes referred to as 'Moving Object'

[4] `http://adams.dm.unipi.it/orbfit/`

[5] `https://github.com/oorb/oorb`

3.5.4 Prototype Implementation

A prototype implementation implementing the key components of DayMOPS baseline design has been completed in LSST Final Design Phase. NightMOPS has not been extensively prototyped, as it is understood not to be an area of significant uncertainty and risk. An extensive report on MOPS prototyping and performance is available as a part of the MOPS Design Document (MOPSD).

Prototype MOPS codes are available at `https://github.com/lsst/mops_daymops` and `https://github.com/lsst/mops_nightmops`. We expect it will be possible to transfer a significant fraction of the existing code into Construction. Current DayMOPS prototype already performs within the computational envelope envisioned for LSST Operations, though it does not yet reach the required completeness requirement.

4 Calibration Pipelines

4.1 Calibration Products Pipeline (WBS 02C.04.02)

4.1.1 Key Requirements

The work performed in this WBS serves two complementary roles:

- It will enable the production of calibration data products as required by the Level 2 Photometric Calibration Plan (LSE-180) and other planning documents [17][6]. This includes both characterization of the sensitivity of the LSST system (optics, filters and detector) and the transmissivity of the atmosphere.
- It will characterize of detector anomalies in such a way that they can be corrected either by the instrument signature removal routines in the Single Frame Processing Pipeline (WBS 02C.03.01) or, if appropriate, elsewhere in the system;
- It will manage and provide a catalog of optical ghosts and glints to other parts of the system upon demand.

4.1.2 Baseline Design

4.1.2.1 Instrumental sensitivity We expect laboratory measurements of the filter profiles. We further baseline the development of a procedure for measuring the filter response at 1 nm resolution using the approach described in [17].

We baseline the following procedure for creating flat fields:

1. Record bias/dark frames;
2. Use "monochromatic" (1 nm) flat field screen flats with no filter in the beam to measure the per-pixel sensitivity;
3. Use a collimated beam projector (CBP) to measure the quantum efficiency (QE) at a set of points in the focal plane, dithering those points to tie them together;

[6] Resolving contradictions between these documents is out of scope here.

4. Combine the screen and CBP data to determine the broad band (10–100 nm) QE of all pixels;
5. Fold in the filter response to determine the 1 nm resolution effective QE of all pixels.

This WBS is responsible for the development of the data analysis algorithms and software required and the ultimate delivery of the flat fields. Development and commissioning of the CBP itself, together with any other infrastructure required to perform the above procedure, lies outwith Data Management (see 04C.08 *Calibration System*).

4.1.2.2 Atmospheric transmissivity Measurements from the auxiliary instrumentation—to include the 1.2 m "Calypso" telescope, a bore-sight mounted radiometer and satellite-based measurement of atmospheric parameters such as pressure and ozone—will be used to determine the atmospheric absorption along the line of sight to standard stars. The atmospheric transmission will be decomposed into a set of basis functions and interpolated in space in time to any position in the LSST focal plane.

This WBS will develop a pipeline for accurate spectrophotometric measurement of stars with the auxiliary telescope. We expect to repurpose and build upon publicly available code e.g. from the PFS[7] project for this purpose.

This WBS will construct the atmospheric model, which may be based either on MODTRAN (as per LSE-180) or a PCA-like decomposition of the data (suggested by [17]).

This WBS will define and develop the routine for fitting the atmospheric model to each exposure from the calibration telescope and providing estimates of the atmospheric transmission at any point in the focal plane upon request.

4.1.2.3 Detector effects An initial cross-talk correction matrix will be determined by laboratory measurements on the Camera Calibration Optical Bench (CCOB). However, to account for possibile instabilities, this WBS will develop an on-telescope method. We baseline this as being based on measurement with the CBP, but we note the alternative approach based on cosmic rays adopted by HSC [10].

Multiple reflections between the layers of the CCD give rise to spatial variability with fine scale structure in images which may vary with time [17,

[7] Subaru's Prime Focus Spectrograph; `http://sumire.ipmu.jp/pfs/`.

§2.5.1]. These can be characterized by white light flat-fields. Preliminary analysis indicates that these effects may be insignificant in LSST [20]; however, the baseline calls for a a routine developed in this WBS to analyse the flat field data and generate fringe frames on demand. This requirement may be relaxed if further analysis (outside the scope of thie WBS) demonstrates it to be unnecessary.

This WBS will develop algorithms to characterize and mitigate anomalies due to the nature of the camera's CCDs.

> **Note:**
> There's a complex inter-WBS situation here: the actual mitigation of CCD anomalies will generally be performed in SFM (WBS 02C.03.01), based on products provided by this WBS which, in turn, may rely on laboratory based research which is broadly outside the scope of DM. We baseline the work required to develop the corrective algorithms here. We consider moving it to WBS 02C.03.01 in future.

The effects we anticipate include:

- QE variation between pixels;
- Static non-uniform pixel sizes (e.g. "tree rings" [23]);
- Dynamic electric fields (e.g. "brighter-fatter" [2]);
- Time dependent effects in the camera (e.g. hot pixels, changing cross-talk coefficients);
- Charge transfer (in)efficiency (CTE).

Laboratory work required to understand these effects is outwith the scope of this WBS. In some cases, this work may establish that the impact of the effect may be neglected in LSST. The baseline plan addresses these issues through the following steps:

- Separate QE from pixel size variations[8] and model both as a function of position (and possibly time);

[8] Refer to work by Rudman.

- Learn how to account for pixel size variation over the scale of objects (e.g. by redistributing charge);
- Develop a correction for the brighter-fatter effect and develop models for any features which cannot be removed;
- Handle edge/bloom using masking or charge redistribution;
- Track defects (hot pixels);
- Handle CTE, including when interpolating over bleed trails.

4.1.2.4 Ghost catalog The Calibration Products Pipeline must provide a catalog of optical ghosts and glints which is available for use in other parts of the system. Detailed characterization of ghosts in the LSST system will only be possible when the system is operational. Our baseline design therefore calls for this system to be prototyped using data from precursor instrumentation; we note that ghosts in e.g. HSC are well known and more significant than are expected in LSST.

> **Note:**
> It is not currently clear where the responsibility for characterizing ghosts and glints in the system lies. We assume it is outwith this WBS.

4.1.3 Constituent Use Cases and Diagrams

Produce Master Fringe Exposures; Produce Master Bias Exposure; Produce Master Dark Exposure; Calculate System Bandpasses; Calculate Telescope Bandpasses; Construct Defect Map; Produce Crosstalk Correction Matrix; Produce Optical Ghost Catalog; Produce Master Pupil Ghost Exposure; Determine CCOB-derived Illumination Correction; Determine Optical Model-derived Illumination Correction; Create Master Flat-Spectrum Flat; Determine Star Raster Photometry-derived Illumination Correction; Create Master Illumination Correction; Determine Self-calibration Correction-Derived Illumination Correction; Correct Monochromatic Flats; Reduce Spectrum Exposure; Prepare Nightly Flat Exposures;

4.1.4 Prototype Implementation

While parts of the Calibration Products Pipeline have been prototyped by the LSST Calibration Group (see the LSE-180 for discussion), these have not been written using LSST Data Management software framework or coding standards. We therefore expect to transfer the know-how, and rewrite the implementation.

4.2 Photometric Calibration Pipeline (WBS 02C.03.07)

4.2.1 Key Requirements

The Photometric Calibration Pipeline is required to internally calibrate the relative photometric zero-points of every observation, enabling the Level 2 catalogs to reach the required SRD precision.

4.2.2 Baseline Design

The adopted baseline algorithm is a variant of “ubercal” [19, 22]. This baseline is described in detail in the Photometric Self Calibration Design and Prototype Document (UCAL).

4.2.3 Constituent Use Cases and Diagrams

Perform Global Photometric Calibration;

4.2.4 Prototype Implementation

Photometric Calibration Pipeline has been fully prototyped by the LSST Calibration Group to the required level of accuracy and performance (see the UCAL document for discussion).

As the prototype has not been written using LSST Data Management software framework or coding standards, we assume a non-negligible refactoring and coding effort will be needed to convert it to production code in LSST Construction.

4.3 Astrometric Calibration Pipeline (WBS 02C.03.08)

4.3.1 Key Requirements

The Astrometric Calibration Pipeline is required to calibrate the relative and absolute astrometry of the LSST survey, enabling the Level 2 catalogs to reach the required SRD precision.

4.3.2 Baseline Design

Algorithms developed for the Photometric Calibration Pipeline (WBS 02C.03.07) will be repurposed for astrometric calibration by changing the relevant functions to minimize. This pipeline will further be aided by WCS and local astrometric registration modules developed as a component of the Single Frame Processing pipeline (WBS 02C.03.01).

Gaia standard stars will be used to fix the global astrometric system. It is likely that the existence of Gaia catalogs may make a separate Astrometric Calibration Pipeline unnecessary.

4.3.3 Constituent Use Cases and Diagrams

Perform Global Astrometric Calibration;

4.3.4 Prototype Implementation

The Astrometric Calibration Pipeline has been partially prototyped by the LSST Calibration Group, but outside of LSST Data Management software framework. We expect to transfer the know-how, and rewrite the implementation.

5 Data Release Production

[**Overview Diagram:**
Collapse and summarize "DRP Top-Level Overview" on confluence along the lines of this outline, mostly by dropping the "Task/Process" boxes and merging data products with the same data units.]

[**Data Products Table:**
Table of data products with brief descriptions (including intermediates, so not just DPDD), expanding the groups defined in Overview Diagram]

Brief summary of all Pipelines:

- Start with iteration between single-visit processing and joint calibration; iteration necessary to get consistent WCS/PSF, but also helpful to identify stars for PSF modeling.
- Iteratively build coadds and detect differences between images. Improve backgrounds, find artifacts, and define "static sky", then use this to detect transient/variable/moving astrophysical sources.
- Detect on coadds, associate detections (inc. DIASources), and deblend on coadds.
- Measure objects on coadds and individual epochs.
- Measure selection functions, compute classifications, absorb level-3 contributions we depend on. Highlight uncertainty.

5.1 Single Visit Processing and Joint Calibration

[**ImChar/JointCal Diagram:**
Extract ImChar/JointCal pipelines from "DRP Top-Level Overview" on confluence and expand detail to show data flow and ordering of "Task/Process" boxes.]

5.1.1 BootstrapImChar

5.1.2 BootstrapJointCal

5.1.3 RefineImChar

5.1.4 RefineJointCal

5.1.5 FinalImChar

5.1.6 FinalJointCal

5.2 Coaddition and Difference Image Processing

[**Coaddition, DiffIm Diagram:**
Extract Coaddition and DiffIm pipelines from "DRP Top-Level Overview" on confluence and expand detail to show data flow and ordering of "Task/Process" boxes.]

5.2.1 WarpAndPsfMatch

5.2.2 BackgroundMatchAndReject

5.2.3 WarpAndPsfCorrelate

5.2.4 CoaddTemplate

5.2.5 DiffIm

5.3 Detection, Association, and Deblending

[**Detection/Association/Deblending Diagram:**
Extract process_coadds pipeline from "DRP Top-Level Overview" on confluence and expand detail to show data flow and ordering of "Task/Process" boxes.]

5.3.1 DecorrelateCoadds

5.3.2 ProcessCoadds (Part 3)

5.4 Object Characterization

[**Object Characterization Diagram:**
Extract multifit/forced_photometry pipelines from "DRP Top-Level Overview" on confluence and expand detail to show data flow and ordering of "Task/Process" boxes.]

5.4.1 ProcessCoadds (Part 2)

5.4.2 MultiFit

5.4.3 ForcedPhotometry

5.5 Postprocessing

[**Postprocessing Diagram:**
Extract Afterburner pipelines from “DRP Top-Level Overview” on confluence and expand detail to show data flow and ordering of “Task/Process” boxes.]

5.5.1 MOPS

5.5.2 Classification

5.5.3 MakeSelectionMaps

5.5.4 GatherContributed

6 Science Data Quality Analysis Pipeline

6.1 Key Requirements

- SDQA Pipeline shall provide low-level data collection functionality for science data quality analysis of Level 1, 2, and Calibration Processing pipelines.
- In addition, SDQA Pipeline shall provide low-level data collection functionality to support software development in Construction and Operations.
- SDQA Pipeline shall provide the visualization, analysis and monitoring capabilities for science quality data analysis. Its inputs will be provided by the SDQA Pipeline.
- The toolkit capabilities shall be made flexible, to provide the analyst with the ability to easily construct custom tests and analyses, and "drill down" into various aspects of the data being analyzed.
- The toolkit will enable automation of tests and monitoring, and issuance of warnings when alerting thresholds are met.
- SDQA Pipeline implementation will monitor and harvest the outputs and logs of execution of other science pipelines, computing user-defined metrics.
- The outputs of SDQA Pipeline runs will be stored into a SDQA repository (RDBMS or filesystem based).

6.2 Key Tasks for Each Level of QA

The SDQA system will be an integrated framework that is capable of providing useful information at four different levels of the data system.

- QA Level 0 - Testing and Validation of DM sub-system in pre-commissioning
- QA Level 1 - Real-time data quality and system assesment during commissioning + operations
- QA Level 2 - Quality assessment of Data Releases

- QA Level 3 - Tools for the community to evaluate the data quality of their own analyses. These will be made available as well-documented and packaged versions of QA Levels 0–2.

6.2.1 QA0

Test the DM software during pre-commissioning as well as test software improvements during commissioning and operations, quantifying the software performance against known expected outputs. Validating the software and its performance on (selected) data.

("Make me a three-color diagram, compute the width of point sources in the blue part of the locus")

("I have 20 visits all over the sky, I want to match up the results")

The main components:

1. CI system that compiles code
2. Test execution harness – that runs test up to "weekly" scale?
3. Library of validation metrics codes – some has to come from Science Pipelines, but KPMs are delivered by SQuaRE
4. Instrumentation capability for computational performance metrics
5. Library of "instrumentations"
6. Interface to data products and QA metrics (including visualization)
    (a) Tabular query result interface
    (b) Visualizer for images
    (c) Plotter
    (d) SuperTask execution on selected data
7. Curated datasets to use in tests
8. Toolkit for analysis of QA outputs (drilldown into existing tests, ad hoc tests, ad hoc afterburners) – some has to come from Science Pipelines or SUIT but SQuaRE provides examples [move to Shared Software Components section]

   (a) Tools that perform computations
   (b) Tools that perform visualization (using Butler if astronomical, maybe direct database if not)

9. Connection from analysis toolkit to validation metrics (attach common interactive plots to validation metrics)

10. QA database including ingestion

11. Notification system for threshold crossings

Prototypes for all of these exist except “Toolkit for analysis of QA outputs” and “Connection from analysis toolkit to validation metrics”

“Toolkit for analysis of QA outputs” will take more resources than the others listed above, but some may be already scheduled in other teams

6.2.2 QA1

Data quality in real time during commissioning and operations. Analyzes the data stream in real time, information about observing conditions (sky brightness, transparency, seeing, magnitude limit) as well as characterize subtle deterioration in system performance.

Validating the operational system.

Main components from above:

1. Library of validation metrics codes

2. Instrumentation capability for computational performance metrics

3. Library of “instrumentations”

4. Interface to results (including visualization)

5. Curated datasets to use in tests

6. Toolkit for analysis of failures (drilldown into existing tests, ad hoc tests, ad hoc afterburners) – some has to come from Science Pipelines or SUIT but SQuaRE provides examples

7. Connection from analysis toolkit to validation metrics

8. QA database including ingestion

New main components:

1. Harness for analyzing alert contents (and perhaps format)
2. Faster metrics codes to meet overall 60 second performance requirement for alert publication (but not necessarily for all QA processing, which must meet only throughput requirements)
3. Additional metrics/instrumentation codes (that must not disturb production system, including its performance, when dynamically inserted)
4. Output interface to "comfort" display (aggregation, trending, etc.)
5. Output interface to automated systems (drop alerts, reschedule field, etc.)
6. Correlator between telemetry streams and metrics
7. Input interface from sources of data not already present in Prompt Processing system
8. Fake source injection and analysis
9. Metrics codes specific for calibration/engineering/special- purpose images

6.2.3 QA2

Assess the quality of data releases (including the co-added image data products) performing quality assessment for astrometric and photometric calibration and derived products, looking for problems with the image processing pipelines and systematic problems with the instrument. Validating the Data Release products. All components from QA0 New main components:

1. DRP-specific dataset
2. Release data product editing tools (including provenance tracking)
3. Output interface to workflow system based on QA results and provenance
4. Provenance analysis tools
5. Output interface to Science Pipelines, including from QA database

6. Comparison tools for overlap areas due to satellite processing
7. Metrics/products for science users to understand quality of science data products (depth mask/selection function, etc.)
8. Characterization report for Data Release

6.2.4 QA3

Data quality based on science analysis performed by the LSST Science Collaborations and the community. Level 0-2 visualization and data exploration tools will be made available to the community. Make all results from the above available. Make all of the above components available to some part of the community (could be just affiliated data centers or could be individual scientists) as a supported product. Ingest external science analysis data as Level 3 data products; ingest useful external science analysis tools.

6.2.5 Prototype Implementation of PipeQA

The pipeQA prototype is a useful reference for exploring ideas and we mention it here to capture this prototype work.

A prototype implementation of the SDQA was implemented in LSST Final Design Phase. The existing prototype was tested with image simulation inputs, as well as real data (SDSS Stripe 82).

The prototype used a set of statically and dynamically generated pages (written in php) to display the results of data production runs. While proving invaluable for data analysis, the prototype design was found it to be difficult to extend with new analyst-developed tests.

The prototype code is available in the `https://github.com/lsst/testing_displayQA` git repository.

7 Science User Interface and Toolkit

7.1 Science Pipeline Toolkit (WBS 02C.01.02.03)

7.1.1 Key Requirements

The Science Pipeline Toolkit shall provide the software components, services, and documentation required to construct Level 3 science pipelines out of components built for Level 1 and 2 pipelines. These pipelines shall be executable on LSST computing resources or elsewhere.

7.1.2 Baseline Design

The baseline design assumes that Level 3 pipelines will use the same `Tasks` infrastructure (see the Data Management Middleware Design document; DMMD) as Level 1 and 2 pipelines[9]. Therefore, Level 3 pipelines will largely be automatically constructible as a byproduct of the overall design.

The additional features unique to Level 3 involve the services to upload/download data to/from the LSST Data Access Center. The baseline for these is to build them on community standards (VOSpace).

7.1.3 Constituent Use Cases and Diagrams

Configure Pipeline Execution; Execute Pipeline; Incorporate User Code into Pipeline; Monitor Pipeline Execution; Science Pipeline Toolkit; Select Data to be Processed; Select Data to be Stored;

7.1.4 Prototype Implementation

While no explicit prototype implementation exists at this time, the majority of LSST pipeline prototypes have successfully been designed in modular and portable fashion. This has allowed a diverse set of users to customize and run the pipelines on platforms ranging from OS X laptops, to 10,000+ core clusters (e.g., BlueWaters), and to implement plugin algorithms (e.g., Kron photometry).

[9] Another way of looking at this is that, functionally, there will be no fundamental difference between Level 2 and 3 pipelines, except for the level of privileges and access to software or hardware resources.

8 Algorithmic Components: AP

Put algorithmic components for the alert production system here.

9 Algorithmic Components: DRP

9.1 Instrument Signature Removal

- All the usual stuff (bias, dark, linearity, crosstalk, fringing, ...?)
- Includes B-F correction.
- Does not include frozen-in coordinate distortions.
- Flat-field to sky SED.

9.2 Artifact Detection

9.2.1 Single-Exposure Morphology

- Find CRs via morphology.
- Find satellites via Hough transform.
- Find some optical ghosts (etc?) from bright star catalog and optics predictions.

9.2.2 Snap Subtraction

- All of the above, but improve by looking at both snaps.

9.2.3 Warped Image Comparison

- Find more optical artifacts by looking at differences between warped images (this is run during background matching).
- Find transient astronomical sources we don't want to include in coadds.

9.3 Artifact Masking/Interpolation

- Set mask planes for all artifacts.
- Eliminate small artifacts by interpolating them.

9.4 Source Detection

- Detect above-threshold regions and peaks in direct or difference images.

9.5 Deblending

9.5.1 Single Visit Deblending

- Generate HeavyFootprint deblends using only a single image.

9.5.2 Multi-Coadd Deblending

- Generate consistent HeavyFootprint deblends from coadds over multiple bands and possibly epoch ranges.

9.6 Measurement

[**NOTE:**
Each bullet here is really a subsubsubsection, but Latex doesn't like that. Could we bump up Sections to Chapters or Parts to get another level, or do something equivalent?]

[**Measurement Algorithms Table:**
Matrix of measurement algorithms and the contexts in which they're run, indicating which combinations are supported]

9.6.1 Variants

- Single Visit Measurement
- Multi-Coadd Measurement
- Forced Measurement
- Difference Image Measurement
- Multi-Epoch Measurement

9.6.2 Algorithms

- Centroids
- Second-Moment Shapes
- Aperture Photometry
- Kron Apertures
- Petrosian Apertures
- Galaxy Models
- Moving Star Models
- Trailed Point Source Models
- Dipole Models

9.6.3 Blended Measurement

- Deblend Template Projection
- Neighbor Noise Replacement
- Simultaneous Fitting
- Hybrid Models

9.7 Background Estimation

- Fit or interpolate large-scale variations while masking out detections.
- Needs to work in crowded fields.
- Needs to work on both difference images and direct images.
- Need to be able to compose backgrounds measured in different coordinate systems on different scales.

9.8 Build Background Reference

- Given multiple overlapping visit images (already warped to a common coordinate system), synthesize a continuous single-epoch image that can be used as a reference for background matching.

9.9 PSF Estimation

9.9.1 Single CCD PSF Estimation

- Fit simple empirical PSF model to stars from a single exposure.
- No chromaticity.
- May use external star catalog, but doesn't rely on one.
- Used in Alert Production and BootstrapImChar in DRP.

9.9.2 Full Visit PSF Estimation

- Decompose PSF into optical + atmosphere.
- Constrain model with stars, telemetry, and wavefront data.
- Wavelength-dependent.
- Used in RefineImChar in DRP.
- Must include some approach to dealing with wings of bright stars.

9.10 Aperture Correction

- Measure curves of growth from bright stars.
- Correct various flux measurements to infinite.
- Propagate uncertainty in aperture correction to corrected fluxes; covariance is tricky.

9.11 Astrometric Calibration

9.11.1 Single Visit

- Fit multi-component WCS to all CCDs in a single visit simultaneously after matching to reference catalog.

9.11.2 Joint Multi-Visit

- Fit multi-component WCS to all CCDs from multiple visits simultaneously after matching to reference catalog.

9.12 Photometric Calibration

9.12.1 Single Visit

- Fit zeropoint (and some small spatial variation?) to all CCDs simultaneously after matching to reference catalog.
- Need for chromatic dependence unclear; probably driven by AP.

9.12.2 Joint Multi-Visit

- Derive SEDs for calibration stars from colors and reference catalog classifications.
- Utilize additional information from wavelenth dependent photometric calibration built by calibration products production.
- Fit zeropoint and possibly perturbations to all CCDs on multiple visits simultaneously after matching to reference catalog..

9.13 PSF Matching

9.13.1 Image Subtraction

- Match template image to science image, as in Alert Production and DRP Difference Image processing.

9.13.2 PSF Homogeniziation for Coaddition

- Match science image to predetermined analytic PSF, as in PSF-matched coaddition.

9.14 Image Warping

9.14.1 Oversampled Images

- Just use Lanczos.

9.14.2 Undersampled Images

- Can use PSF model as interpolant if we also want to convolve with PSF (as in likelihood coadds). Otherwise impossible?

9.14.3 Irregularly-Sampled Images

- Approximate procedure for fixing small-scale distortions in pixel grid.

9.15 Image Coaddition

- Can do outlier rejection (but usually doesn't).
- Needs to propagate full uncertainty somehow.
- May need to propagate larger-scale per-exposure masks to get right PSF model or other coadded quantities.
- Should be capable of combining coadds from different bands and/or epoch ranges ranges as well as combining individual exposures.

9.16 DCR-Corrected Template Generation

- Somwewhat like coaddition, but may need to add dimensions for wavelength or airmass, and may involve solving an inverse problem instead of just compute means.

9.17 Image Decorrelation

9.17.1 Difference Image Decorrelation

- Fourier-space (?) deconvolution of preconvolved difference images before measurement - ZOGY as reinterpreted by Lupton.
- Need to test with small-scale research before committing to this approach.

9.17.2 Coadd Decorrelation

- Fourier-space/iterative deconvolution of likelihood coadds, as in DMTN-15.
- Need to test with small-scale research before committing to this approach.

9.18 Star/Galaxy Classification

9.18.1 Single Visit S/G, Pre-PSF

- Select stars to be used in PSF estimation (mostly from moments).

9.18.2 Single Visit S/G, Post-PSF

- Extendedness or trace radius difference that classifies sources based on single frame measurements that can utilize the PSF model. Used to select single-frame calibration stars, and probably aperture correction stars.

9.18.3 Multi-Source S/G

- Aggregate of single-visit S/G post-PSF numbers in jointcal.

9.18.4 Object Classification

- Best classification derived from multifit and possibly variability.

9.19 Variability Classification

- Periodograms derived from forced photometry.

9.20 Association and Matching

9.20.1 Single Visit to Reference Catalog, Semi-Blind

- Run prior to single-visit WCS fitting, with only telescope's best guess as a starting WCS.

9.20.2 Multiple Visits to Reference Catalog

- Match sources from multiple visits to a single reference catalog, assuming good WCSs.

9.20.3 DIAObject Generation

- Match new DIASources to existing DIAObjects and generate new DIAObjects. Definitely run in AP, maybe run in DRP.

9.20.4 Object Generation

- Match coadd detections from different bands/SEDs/epoch-ranges, merging Footprints and associating peaks.

- Also merge in DIASources or (if already self-associated) DIAObjects.

9.20.5 Cross-Patch Merging

- Resolve duplicates in patch overlap regions by flagging "primary" objects. Difficult due to blending.

9.20.6 Cross-Tract Merging

- Resolve duplicates in tract overlap regions by flagging "primary" objects. Difficult due to blending.

9.21 MOPS

- Link trailed (and other?) DIASources to form SSObjects, fit for orbits.

10 Software Primitives

10.1 Applications Framework (WBS 02C.03.05, 02C.04.01)

10.1.1 Key Requirements

The ***LSST Applications Framework*** (`afw`) is to provide the basic functionality needed by an image processing system. In particular, it will provide:

- Classes to represent and manipulate mappings between device and astronomical coordinates;
- Classes to represent and manipulate images and exposures;[10]
- Classes to represent and estimate backgrounds on images;
- Classes to represent the geometry of the camera;
- Base classes to represent and manipulate the point spread function (PSF);
- Routines to perform detection of sources on images, and classes to represent these detections (*"footprints"*);
- Classes to represent astronomical objects;
- Classes to represent and manipulate tables of astronomical objects;
- Other low-level operations as needed by LSST science pipelines.

10.1.2 Baseline Design

This library will form the basis for all image processing pipelines and algorithms used for LSST, so special attention will be paid to performance. For that reason, this baseline design calls for a library of C++ classes and functions. Throughout construction these low level routines will be continually upgraded and refined to meet the performance and algorithmic fidelity requirements driven by the algorithmic requirements in other WBSs. If it proves impossible to meet performance goals based on pure C++ code, GPU support for some functions has been prototyped.

[10] images with associated metadata.

We expect that LSST stack developers, Level 3 pipeline developers, and a substantial group of end users will need to interact directly with the `afw` functionality. For that reason, it is exposed to Python callers through a Python module named `lsst.afw`. Throughout the construction period, we expect to devote effort to refining this interface layer to provide an idiomatic system which adheres to community norms and expectations.

10.1.3 Prototype Implementation

A prototype version of the required classes was described in the UML Domain Model (LDM-133) and implemented in LSST Final Design Phase, including prototype GPU (CUDA) support for major image processing functions (e.g., warping). A significant fraction of this code will be transferred into construction.

Work-in-progress code is available at `https://github.com/lsst/afw/`. The documentation for this code is located at `http://ls.st/w3o` and `http://ls.st/6i0`.

11 Glossary

API Applications Programming Interface

CBP Collimated Beam Projector

CCOB Camera Calibration Optical Bench

CTE Charge Transfer Efficiency

DAC Data Access Center

DAQ Data Acquisition

DMS Data Management System

DR Data Release.

EPO Education and Public Outreach

Footprint The set of pixels that contains flux from an object. Footprints of multiple objects may have pixels in common.

FRS Functional Requirements Specification

MOPS Moving Object Pipeline System

OCS Observatory Control System

Production A coordinated set of pipelines

PFS Prime Focus Spectrograph. An instrument under development for the Subaru Telescope.

PSF Point Spread Function

QE Quantum Efficiency

RGB Red-Green-Blue image, suitable for color display.

SDS Science Array DAQ Subsystem. The system on the mountain which reads out the data from the camera, buffers it as necessary, and supplies it to data clients, including the DMS.

SDQA Science Data Quality Assessment.

SNR Signal-to-Noise Ratio

SQL Structured Query Language, the common language for querying relational databases.

TBD To Be Determined

Visit A pair of exposures of the same area of the sky taken in immediate succession. A Visit for LSST consists of a 15 second exposure, a 2 second readout time, and a second 15 second exposure.

VO Virtual Observatory

VOEvent A VO standard for disseminating information about transient events.

WCS World Coordinate System. A bidirectional mapping between pixel- and sky-coordinates.

References

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel and J. Sander, **OPTICS: Ordering Points To Identify the Clustering Structure**, Proc ACM SIGMOD (1999).

[2] P. Antilogus, P. Astier, P .Doherty, A. Guyonnet and N. Regnault **The brighter-fatter effect and pixel correlations in CCD sensors** Journal of Instrumentation, Volume 9, Issue 3, article id. C03048 (2014).

[3] A. Becker et al, **Report on Late Winter 2013 Production: Image Differencing** `http://ls.st/x9f`.

[4] A. Becker, **Report on Summer 2014 Production: Analysis of DCR**, `https://github.com/lsst-dm/S14DCR/blob/master/report/S14report_V0-00.pdf`.

[5] J. S. Bloom et al, **Automating discovery and classification of transients and variable stars in the synoptic survey era**, PASP 124, 1175–1196 (2012).

[6] J. F. Bosch, **Modeling Techniques for Measuring Galaxy Properties in Multi-Epoch Surveys**, PhD Thesis, University of California, Davis (2011). `http://adsabs.harvard.edu/abs/2011PhDT.......226B`

[7] J. Bosch, P. Gee, R. Owen, M. Juric and the LSST DM team, **LSST DM S13 Report: Shape measurement plans and prototypes**, `https://docushare.lsstcorp.org/docushare/dsweb/ImageStoreViewer/Document-15298`

[8] J. Bosch, **Measurement of Blended Objects in LSST**.

[9] E. M. Huff et al, **Seeing in the dark – I. Multi-epoch alchemy**, `http://arxiv.org/abs/1111.6958`.

[10] H. Furusawa et al, **Hyper Suprime-Cam Survey Pipeline Description**, `http://hsca.ipmu.jp/pipeline_outputs.pdf`.

[11] M. J. Jee and J. A. Tyson, **Toward Precision LSST Weak-Lensing Measurement. I. Impacts of Atmospheric Turbulence and Optical Aberration**, PASP 123, 596(2011).

[12] M. J. Jee, J. A. Tyson, M. D. Schneider, D. Wittman, S. Schmidt and S. Hilbert, **Cosmic shear results from the Deep Lens Survey. I. Joint constraints on Ω_M and σ_8 with a two-dimensional analysis**, ApJ 765 74 (2013).

[13] J. Kubica et al, **Efficiently Tracking Moving Sources in the LSST**, Bulletin of the American Astronomical Society, 37, 1207 (2005).

[14] D. Lang, D. Hogg, S. Jester and H.-W. Rix, **Measuring the undetectable: Proper motions and parallaxes of very faint sources**, AJ 137 4400–4111 (2009).

[15] R. H. Lupton et al, **SDSS Image Processing II: The *Photo* Pipelines**. `http://www.astro.princeton.edu/~rhl/photo-lite.pdf`

[16] R. Lupton and Ž. Ivezić, **Experience with SDSS: the Promise and Perils of Large Surveys**, Astrometry in the Age of the Next Generation of Large Telescopes, ASP Conferences Series, Vol 338 (2005). `http://adsabs.harvard.edu/abs/2005ASPC..338..151L`

[17] R. Lupton, M. Jurić and C. Stubbs, **LSST's Plans for Calibration Photometry**, July 2015.

[18] L. Denneau, J. Kubica and R. Jedicke, **The Pan-STARRS Moving Object Pipeline**, Astronomical Data Analysis Software and Systems XVI ASP Conference Series, Vol. 376, proceedings of the conference held 15-18 October 2006 in Tucson, Arizona, USA. Edited by Richard A. Shaw, Frank Hill and David J. Bell., p.257.

[19] N. Padmanabhan et al, **An Improved Photometric Calibration of the Sloan Digital Sky Survey Imaging Data**, ApJ 674 1217–1233 (2008).

[20] A. Rasmussen, **Sensor Modeling for the LSST Camera Focal Plane: Current Status of SLAC Originated Code** July 2015. `https://docushare.lsstcorp.org/docushare/dsweb/Get/Document-8590`.

[21] J. Richards et al, **On Machine-learned Classification of Variable Stars with Sparse and Noisy Time-series Data**, ApJ 733 10 (2011).

[22] E. F. Schlafly et al, **Photometric Calibration of the First 1.5 Years of the Pan-STARRS1 Survey**, ApJ 756 158 (2012).

[23] C. W. Stubbs, **Precision Astronomy with Imperfect Fully Depleted CCDs – An Introduction and a Suggested Lexicon**, Journal of Instrumentation, Volume 9, Issue 3, article id. C03032 (2014).

[24] A. S. Szalay, A. J. Connolly and G. P. Szokoly, **Simultaneous Multicolor Detection of Faint Galaxies in the Hubble Deep Field**, AJ 117 68–74 (1999).