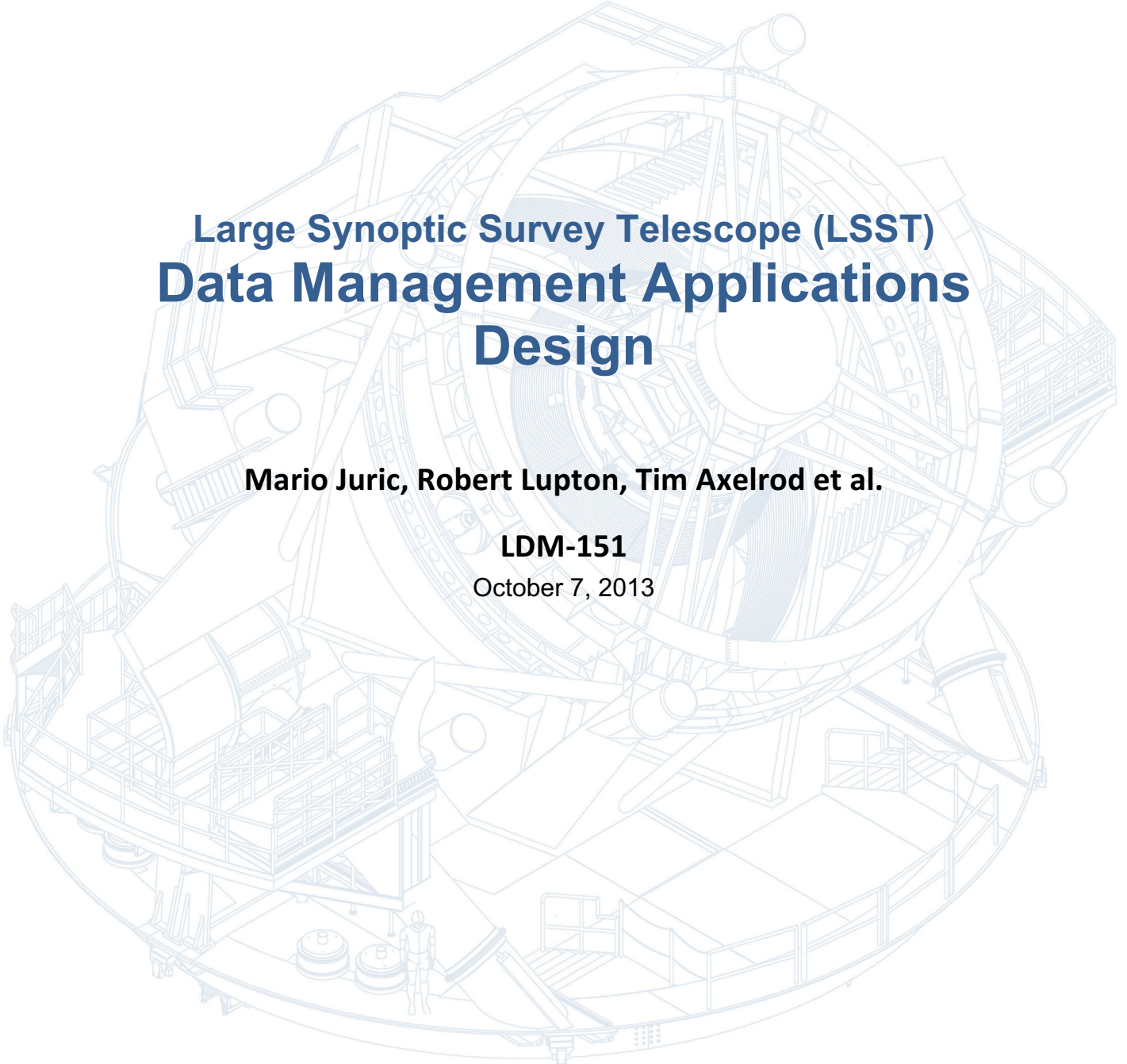




---

LARGE SYNOPTIC SURVEY TELESCOPE

---

A detailed white line-art cutaway diagram of the LSST dome, showing the internal structure, including the telescope's primary mirror, secondary mirror, and various support structures and platforms. A small human figure is shown at the bottom for scale.

# 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<sup>th</sup> May, 2016, 14:52hrs

---

\*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 guidelines, 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

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	LSST Data Management System . . . . .	7
1.2	Data Products . . . . .	10
1.3	Science Pipelines Overview . . . . .	11
1.3.1	Level 1 Pipelines Overview . . . . .	13
1.3.2	Level 2 Pipelines Overview . . . . .	16
1.3.3	Enabling Level 3 Pipelines . . . . .	19
1.3.4	Science Data Quality Analysis Pipeline and Toolkit . . . . .	19
<b>2</b>	<b>Level 1 Pipelines</b>	<b>22</b>
2.1	Single Frame Processing Pipeline (WBS 02C.03.01) . . . . .	22
2.1.1	Key Requirements . . . . .	22
2.1.2	Baseline Design . . . . .	23
2.1.2.1	Instrumental Signature Removal: . . . . .	23
2.1.2.2	PSF determination: . . . . .	24
2.1.2.3	Sky Background Determination: . . . . .	25
2.1.2.4	WCS determination and image registration . . . . .	26
2.1.3	Constituent Use Cases and Diagrams . . . . .	26
2.1.4	Prototype Implementation . . . . .	26
2.2	Image Differencing Pipeline (WBS 02C.03.04) . . . . .	27
2.2.1	Key Requirements . . . . .	27
2.2.2	Baseline Design . . . . .	27
2.2.2.1	Dipole model fit . . . . .	28
2.2.3	Constituent Use Cases and Diagrams . . . . .	28
2.2.4	Prototype Implementation . . . . .	28
2.3	Association Pipeline (WBS 02C.03.02) . . . . .	30
2.3.1	Key Requirements . . . . .	30
2.3.2	Baseline Design . . . . .	30
2.3.3	Constituent Use Cases and Diagrams . . . . .	30
2.3.4	Prototype Implementation . . . . .	30
2.4	Alert Generation Pipeline (WBS 02C.03.03) . . . . .	31
2.4.1	Key Requirements . . . . .	31
2.4.2	Baseline Design . . . . .	31
2.4.3	Constituent Use Cases and Diagrams . . . . .	31
2.4.4	Prototype Implementation . . . . .	31
2.5	Moving Object Pipeline (WBS 02C.03.06) . . . . .	32

2.5.1	Key Requirements . . . . .	32
2.5.2	Baseline Design . . . . .	32
2.5.3	Constituent Use Cases and Diagrams . . . . .	32
2.5.4	Prototype Implementation . . . . .	33
<b>3</b>	<b>Calibration Pipelines</b>	<b>34</b>
3.1	Calibration Products Pipeline (WBS 02C.04.02) . . . . .	34
3.1.1	Key Requirements . . . . .	34
3.1.2	Baseline Design . . . . .	34
3.1.2.1	Instrumental sensitivity . . . . .	34
3.1.2.2	Atmospheric transmissivity . . . . .	35
3.1.2.3	Detector effects . . . . .	35
3.1.2.4	Ghost catalog . . . . .	37
3.1.3	Constituent Use Cases and Diagrams . . . . .	37
3.1.4	Prototype Implementation . . . . .	38
3.2	Photometric Calibration Pipeline (WBS 02C.03.07) . . . . .	39
3.2.1	Key Requirements . . . . .	39
3.2.2	Baseline Design . . . . .	39
3.2.3	Constituent Use Cases and Diagrams . . . . .	39
3.2.4	Prototype Implementation . . . . .	39
3.3	Astrometric Calibration Pipeline (WBS 02C.03.08) . . . . .	40
3.3.1	Key Requirements . . . . .	40
3.3.2	Baseline Design . . . . .	40
3.3.3	Constituent Use Cases and Diagrams . . . . .	40
3.3.4	Prototype Implementation . . . . .	40
<b>4</b>	<b>Level 2 Pipelines</b>	<b>41</b>
4.1	PSF Estimation Pipeline (WBS 02C.04.03) . . . . .	41
4.1.1	Key Requirements . . . . .	41
4.1.2	Baseline Design . . . . .	41
4.1.3	Constituent Use Cases and Diagrams . . . . .	42
4.1.4	Prototype Implementation . . . . .	42
4.2	Image Coaddition Pipeline (WBS 02C.04.04) . . . . .	43
4.2.1	Key Requirements . . . . .	43
4.2.2	Baseline Design . . . . .	44
4.2.2.1	Coaddition . . . . .	44
4.2.2.2	Warping . . . . .	45
4.2.2.3	Background Matching . . . . .	46

	4.2.2.4	Sky Tessellation and Coadd Projections . . . .	46
	4.2.2.5	CoaddPsf . . . . .	47
	4.2.2.6	Image artifact rejection . . . . .	47
	4.2.2.7	Differential Chromatic Refraction . . . . .	48
	4.2.3	Constituent Use Cases and Diagrams . . . . .	48
	4.2.4	Prototype Implementation . . . . .	48
4.3		Object Detection and Deblending (WBS 02C.04.05) . . . . .	49
	4.3.1	Key Requirements . . . . .	49
	4.3.2	Baseline Design . . . . .	49
	4.3.2.1	Object detection . . . . .	49
	4.3.2.2	Deblender . . . . .	50
	4.3.2.3	Sky coverage . . . . .	51
	4.3.3	Constituent Use Cases and Diagrams . . . . .	51
	4.3.4	Prototype Implementation . . . . .	51
4.4		Object Characterization Pipeline (WBS 02C.04.06) . . . . .	53
	4.4.1	Key Requirements . . . . .	53
	4.4.2	Baseline Design . . . . .	55
	4.4.2.1	Single-epoch Characterization . . . . .	55
	4.4.2.2	MultiFit . . . . .	55
	4.4.2.3	Point source model fit . . . . .	56
	4.4.2.4	Bulge-disk model fit . . . . .	57
	4.4.2.5	Trailed model fit . . . . .	57
	4.4.2.6	Dipole model fit . . . . .	57
	4.4.2.7	Hybrid model fit . . . . .	57
	4.4.2.8	Centroids . . . . .	58
	4.4.2.9	Adaptive moments . . . . .	58
	4.4.2.10	Petrosian and Kron fluxes . . . . .	58
	4.4.2.11	Aperture surface brightness . . . . .	59
	4.4.2.12	Resolved/Unresolved object separation . . . . .	59
	4.4.2.13	Star-Galaxy classification . . . . .	59
	4.4.2.14	Colors . . . . .	59
	4.4.2.15	Photometric Redshifts . . . . .	60
	4.4.2.16	Variability Characterization . . . . .	60
	4.4.3	Constituent Use Cases and Diagrams . . . . .	60
	4.4.4	Prototype Implementation . . . . .	60

<b>5</b>	<b>Science Data Quality Analysis Pipeline</b>	<b>62</b>
5.1	Key Requirements . . . . .	62
5.2	Key Tasks for Each Level of QA . . . . .	62
5.2.1	QA0 . . . . .	63
5.2.2	QA1 . . . . .	64
5.2.3	QA2 . . . . .	65
5.2.4	QA3 . . . . .	66
5.2.5	Prototype Implementation of PipeQA . . . . .	66
<b>6</b>	<b>Science User Interface and Toolkit</b>	<b>67</b>
6.1	Science Pipeline Toolkit (WBS 02C.01.02.03) . . . . .	67
6.1.1	Key Requirements . . . . .	67
6.1.2	Baseline Design . . . . .	67
6.1.3	Constituent Use Cases and Diagrams . . . . .	67
6.1.4	Prototype Implementation . . . . .	67
<b>7</b>	<b>Algorithmic Components: AP</b>	<b>68</b>
<b>8</b>	<b>Algorithmic Components: DRP</b>	<b>69</b>
<b>9</b>	<b>Software Primitives</b>	<b>70</b>
9.1	Applications Framework (WBS 02C.03.05, 02C.04.01) . . . . .	70
9.1.1	Key Requirements . . . . .	70
9.1.2	Baseline Design . . . . .	70
9.1.3	Prototype Implementation . . . . .	71
<b>10</b>	<b>Glossary</b>	<b>72</b>



# 1 Introduction

## 1.1 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](#)).

02C.05 Science User Interface and Analysis Tools	02C.01.02.02 - 03 SDQA and Science Pipeline Toolkits
02C.06.01 Science Data Archive (Images, Alerts, Catalogs)	02C.01.02.01, 02C.02.01.04, 02C.03, 02C.04 Alert, SDQA, Calibration, Data Release Productions/Pipelines
02C.03.05, 02C.04.01 Application Framework	
02C.06.02 Data Access Services	
02C.07.01, 02C.06.03 Processing Middleware	
02C.07.02 Infrastructure Services (System Administration, Operations, Security)	
02C.07.04.01 Archive Site	02C.08.03 Long-Haul Communications
02C.07.04.02 Base Site	
Physical Plant (included in above)	

Data Management System Design LDM-148

**Application Layer (LDM-151)**

- Scientific Layer
- Pipelines constructed from reusable, standard “parts”, i.e. Application Framework
- Data Products representations standardized
- Metadata extendable without schema change
- Object-oriented, python, C++ Custom Software

**Middleware Layer (LDM-152)**

- Portability to clusters, grid, other
- Provide standard services so applications behave consistently (e.g. provenance)
- Preserve performance (<1% overhead)
- Custom Software on top of Open Source, Off-the-shelf Software

**Infrastructure Layer (LDM-129)**

- Distributed Platform
- Different sites specialized for real-time alerting vs peta-scale data access
- Off-the-shelf, Commercial Hardware & Software, Custom Integration

Figure 1: Architecture of the Data Management System

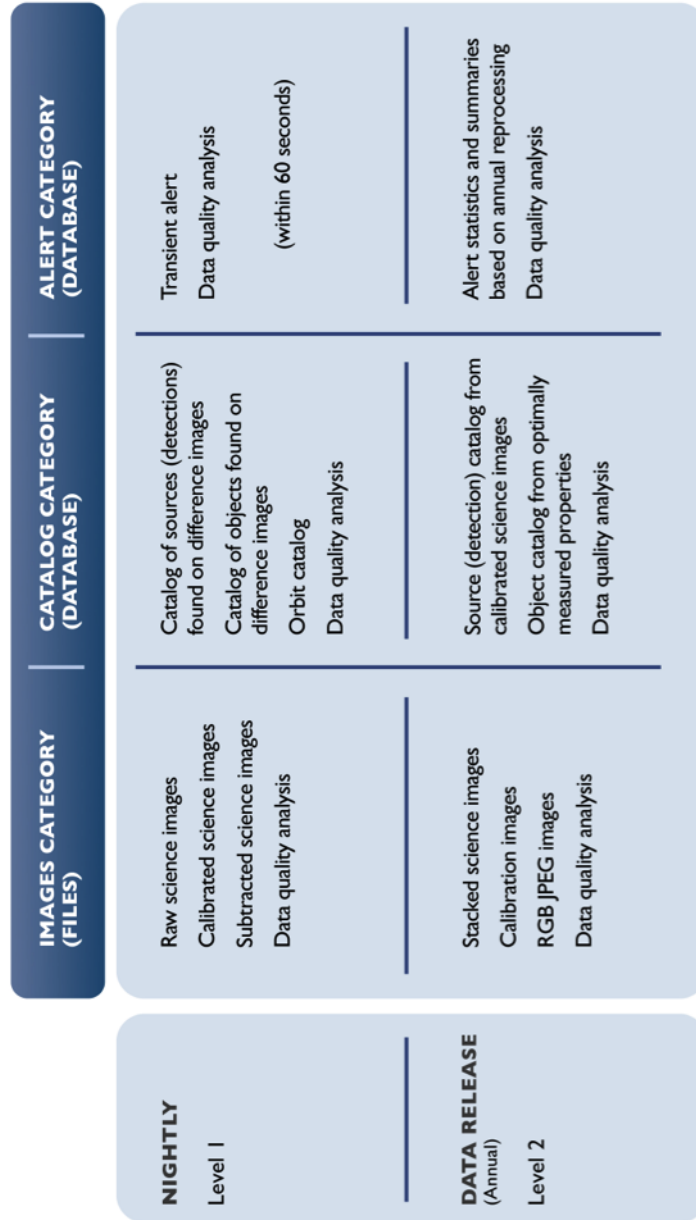


Figure 2: Organization of LSST Data Products

## 1.2 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”<sup>1</sup> 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.

---

<sup>1</sup>Science Data Quality Analysis

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.

### 1.3 Science Pipelines Overview

We recognize four major groups of science pipelines residing in the Applications layer:

- **Level 1 Pipelines**, grouped under the **Alert Production** element of the WBS, are designed to generate Level 1 data products. These include:
  - *Single Frame Processing (“SFM”) Pipeline* (WBS 02C.03.01), to reduce acquired visits and detect and characterize astrophysical sources present in these visits.
  - *Image Differencing Pipeline* (WBS 02C.03.04), to create difference images, and detect and characterize sources in them.
  - *Association Pipeline* (WBS 02C.03.02), to associate sources detected in the difference images with known objects.
  - *Alert Generation Pipeline* (WBS 02C.03.03), to generate and transmit alerts to time-domain events (e.g., transients) to the astronomical community, and
  - *Moving Object Pipeline* (WBS 02C.03.06), to identify, link and compute orbits for Solar System objects detected in difference images.

Level 1 pipelines run as the data are being acquired. They primarily focus on image differencing, and the reduction of information extracted from difference images. The algorithms they employ are designed and chosen to complete processing on minute (alert production) to day (*DayMOPS*) time scales. They are also rerun as a part of Data Release Production (DRP), potentially in somewhat different configurations to achieve greater precision at the expense of increased runtime.

- **Level 2 Pipelines** run annually or semi-annually (for the first year of data), and are designed to generate deep co-adds and catalogs stemming from analysis of direct image data. These include:
  - *PSF Estimation Pipeline* (WBS 02C.04.03), to estimate the PSF properties and variation across the focal plane for each visit, to the degree of precision required by the [SRD](#). Note that the work of this pipeline goes beyond the typical single-CCD PSF estimation present in the SFM pipeline.
  - *Image Coaddition Pipeline* (WBS 02C.04.04), to generate and characterize coadded images of the sky, as well as create templates for image differencing.
  - *Object Detection and Deblending* (WBS 02C.04.05), to detect sources in images of the sky and decompose them into individual astronomical objects.
  - *Object Characterization Pipeline* (WBS 02C.04.06), to characterize (perform measurements of) astrophysical objects detected in LSST imaging (both in single frames and coadds).
- **Calibration Pipelines** process the collected calibration data and perform calibration of LSST instruments and data products. These include:
  - *Calibration Products Pipeline* (WBS 02C.04.02), that generates the necessary calibration data products (e.g., master flats, biases, atmospheric models, etc.). It is run periodically as new calibration data are acquired.
  - *Photometric Calibration Pipeline* (WBS 02C.03.07), that performs global photometric self-calibration of the Level 2 dataset.
  - *Astrometric Calibration Pipeline* (WBS 02C.03.08), that performs global astrometric self-calibration of the Level 2 dataset.

The calibration products pipeline is also rerun as a part of Data Release Processing. Global self-calibration steps run in DRP only.

- **Science Data Quality Assessment (SDQA) pipelines and toolkits**, to enable collection, computation, visualization, monitoring and analysis of data quality metrics across all pipelines. These are divided into:

- *Science Data Quality Assessment Pipeline* (WBS 02C.01.02.02), that provides low-level data collection functionality for SDQA and
- *Science Data Quality Analyst Toolkit* (WBS 02C.01.02.02), that provides the visualization, analysis and monitoring capabilities for SDQA.

In addition to these four, we recognize two other, cross-cutting, elements of DMS functionality:

- *Common Image and Catalog Processing Framework* (WBS 02C.03.05, 02C.04.01), known as the **Application Framework (afw)**, that collects base classes and algorithms used by the DM Applications layer. The framework is split in two WBS elements, to reflect the multi-institutional nature of the work, but is functionally viewed as a single, integrated, component (class library).
- The *Science Pipeline Toolkit* (WBS 02C.01.02.03), a collection of software components (and design principles) designed to enable construction of Level 3 pipelines relying on reusable lower-level components produced in support of other LSST DM software.

### 1.3.1 Level 1 Pipelines Overview

The production of Level 1 products is generally performed nightly, directly fed by the output data stream from the Camera SDS<sup>2</sup> during observing. This data stream contains both unprocessed (raw) camera images, and images that have been corrected for crosstalk by the SDS on the Summit. The normal observing pattern is to take two 15 second exposures of the same field in immediate succession. These two exposures together form a *visit*, which is the typical image data unit processed by the rest of the DM system.

The logical flow of Level 1 processing is shown in the Use Case diagram presented in Figure 3. For every observation, the following sequence of events will unfold:

1. A visit is acquired (**Prepare for Observing**) and reduced (**Process Raw Exposures to Calibrated Exposure**) to a single *visit image*. This

---

<sup>2</sup>Science Array Data Acquisition (DAQ) Subsystem

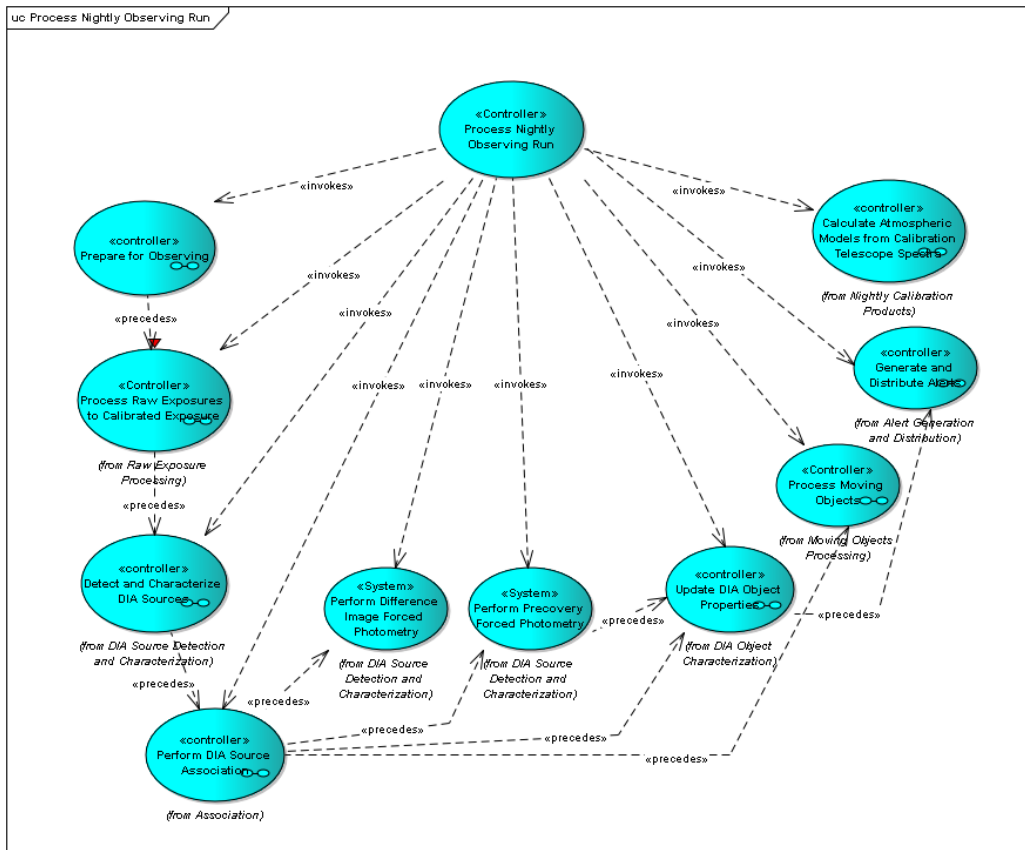


Figure 3: Level 1 Processing Flow Diagram



includes instrumental signature removal<sup>3</sup>, combining of snaps, etc.

2. The visit image is differenced against the appropriate template and `DIASources` are detected (`Detect and Characterize DIA Sources`). If necessary, deblending is performed at this stage.

The flux and shape of the `DIASource` are measured on the difference image. PSF photometry is performed on the visit image at the position of the `DIASource` to obtain a measure of the absolute flux.

3. The Level 1 database is searched for a `DIAObject` or an `SSObject` with which to positionally associate the newly discovered `DIASource`. If no match is found, a new `DIAObject` is created and the observed `DIASource` is associated to it.

If the `DIASource` has been associated to an `SSObject` (a known Solar System object), it will be flagged as such and an alert will be issued. Further processing will occur in daytime (`Process Moving Objects`).

4. Otherwise, the associated `DIAObject` measurements will be updated with new data (`Update DIA Object Properties`). All affected columns will be recomputed, including proper motions, centroids, light curves, nearest Level 2 `Objects`, etc.
5. An alert is issued (`Generate and Distribute Alerts`) that includes all required components, as described in the [DPDD](#).
6. For all `DIAObjects` overlapping the field of view to which a `DIASource` from this visit has *not* been associated, forced photometry will be performed (`Perform Difference Image Forced Photometry`). No alerts will be issued for these measurements.

Within 24 hours of discovery, LSST DM system will perform *precovery* PSF forced photometry on any prior difference image overlapping the predicted position of new `DIAObjects` taken within the past 30 days (`Perform Precovery Forced Photometry`).

Similarly, in daytime after the nightly observing run, atmospheric models from the calibration telescope spectra will be calculated (`Calculate`

---

<sup>3</sup>E.g., subtraction of bias and dark frames, flat fielding, bad pixel/column interpolation, etc.

Atmospheric Models from Calibration Telescope Spectra) and made available to the users.

In addition to these, the Moving Object Pipeline (MOPS; WBS 02C.03.06; `Process Moving Objects`) will also be run in daytime. It is described in its own section of this document, with a detailed design in a separate Moving Object Pipeline Design Document ([MOPSD](#)).

### 1.3.2 Level 2 Pipelines Overview

Figure 4 presents a high-level overview of the Level 2 data processing workflow. Logically<sup>4</sup>, the processing begins with single-frame (visit) image reduction and source measurement, followed by global astrometric and photometric calibration, coadd creation, detection on coadds, association and deblending, object characterization, and forced photometry measurements. The UML Use Case model ([LDM-134](#)) captures these activities in the `Produce a Data Release` diagram.

The following is a high-level description of steps which occur during regular Level 2 data processing:

1. *Single Frame Processing*: Raw exposures are reduced to *calibrated visit exposures*, and `Sources` are independently detected, deblended, and measured on all visits. Their measurements (instrumental fluxes and shapes) are stored in the `Source` table. This step is performed by the *Single Frame Processing Pipeline* (WBS 02C.03.01).
2. *Relative calibration*: The survey is internally calibrated, both photometrically and astrometrically using the *Astrometric* (WBS 02C.03.08) and *Photometric Calibration Pipelines* (WBS 02C.03.07). Relative zero points over the focal plane and astrometric corrections are computed for every visit.
3. *Coadd creation*: Deep, seeing optimized, and short-period per-band coadds are created in *ugrizy* bands, as well as deeper, multi-color, coadds. This task is performed by the *Image Coaddition Pipeline*

---

<sup>4</sup>The actual implementation may parallelize these steps to the extent possible; see LDM-230, the Automated DM Operations Document ([DM OpsCon](#)).

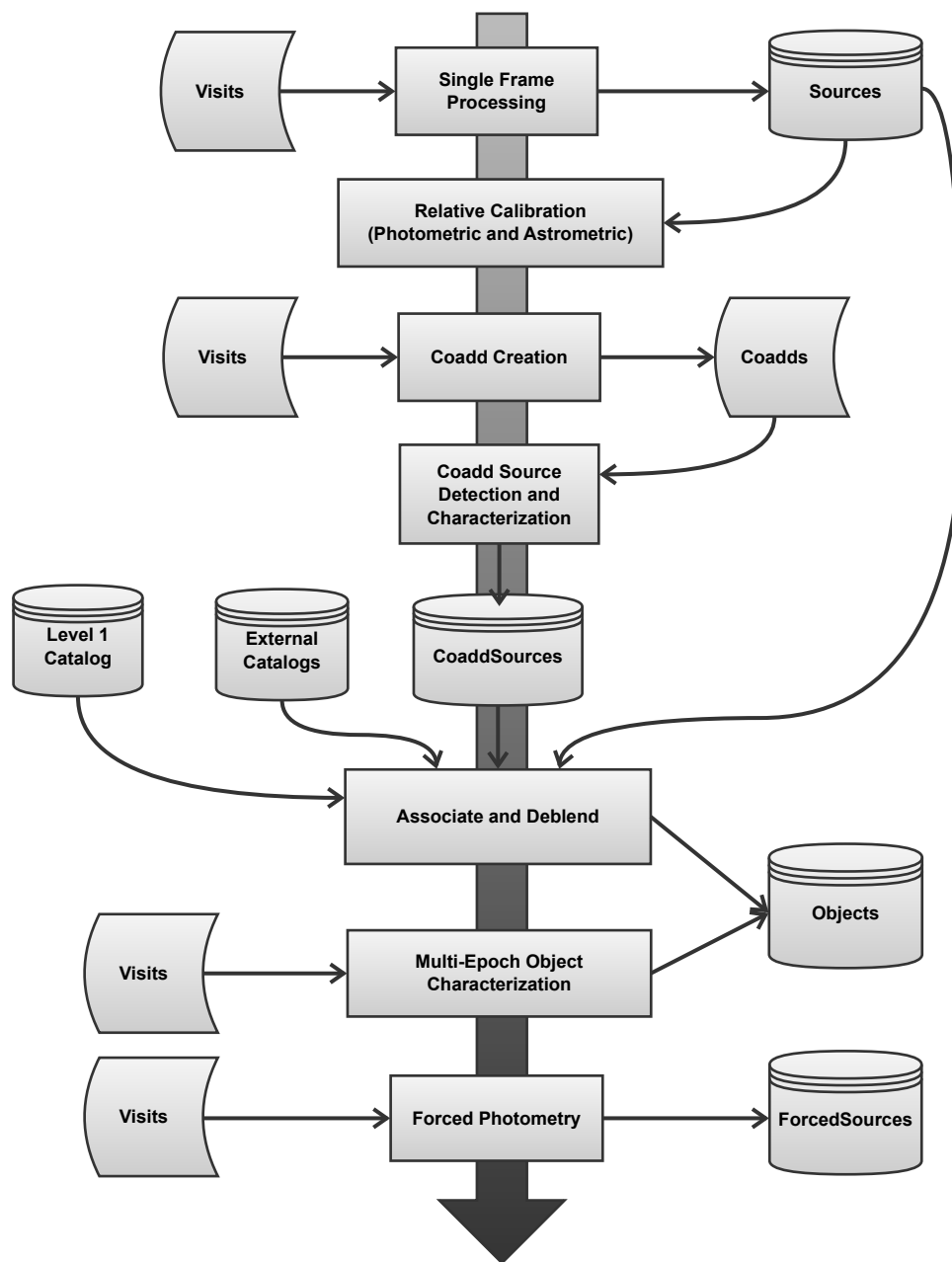


Figure 4: Level 2 Processing Overview

(WBS 02C.04.04). Transient sources (including Solar System objects, explosive transients, etc), will be rejected from the coadds.

4. *Coadd source detection.* Sources will be detected on all coadds generated in the previous step. The source detection algorithm will detect regions of connected pixels, known as *footprints*, above the nominal  $S/N$  threshold in the *PSF-likelihood image* of the visit. Each footprint may have one or more *peaks*, and the collection of these peaks (and their membership in the footprints) are the output of this stage. This information will be stored in a catalog of `CoaddSources`. The detection is performed by the ***Object Detection and Deblending*** system (WBS 02C.04.05).
5. *Coadd source deblending and characterization.* The next stage in the pipeline will decompose the `CoaddSources` into a set of individual astronomical sources which is consistent across all bands, a process known as *deblending*. The deblender may make use of the catalogs of `Sources` and `CoaddSources`, catalogs of `DIASources`, `DIAObjects` and `SSObjects` detected on difference images, and objects from external catalogs. The deblended objects will then be characterized by measuring their positions, shapes and fluxes on the coadded images and by fitting galaxy models. This functionality is contained within the ***Object Detection and Deblending*** system (WBS 02C.04.05) and the ***Object Characterization Pipeline*** (WBS 02C.04.06).
6. *Multi-epoch object characterization.* A set of measurements (including predefined classes of model fits) will be performed on each of the `Objects` identified in the previous step, taking all available multi-epoch data into account. Model fits will be performed using *MultiFit*-type algorithms. Rather than coadding a set of images and measuring object characteristics on the coadd, MultiFit simultaneously fits PSF-convolved models to the objects multiple observations. This reduces systematic errors, improves the overall  $S/N$ , and allows for fitting of time-dependent quantities degenerate with shape on the coadds (for example, the proper motion). The models we plan to fit will *not* allow for flux variability. Object characterization is a part of the ***Object Characterization Pipeline*** (WBS 02C.04.06).
7. *Forced Photometry.* Source fluxes will be measured on every visit, with

the position, motion, structural parameters, and deblending characterized in the previous step kept fixed. This process of *forced photometry*, will result in the characterization of the light-curve for each object in the survey. Forced photometry is functionally a part of the ***Object Characterization Pipeline*** (WBS 02C.04.06).

### 1.3.3 Enabling Level 3 Pipelines

Level 3 capabilities are envisioned to enable science cases requiring further custom user-specified processing, especially the kind that would greatly benefit from co-location within the LSST Data Access Center. The high-level requirement for Level 3 is established in §3.5 of the LSST SRD.

To enable Level 3 use cases, LSST Data Management pipelines will be designed in a modular fashion to maximize the potential for reusability and synergy between Level 3 and Levels 1 and 2.

For example, a typical Level 3 use case will be to perform a different kind of measurement on objects detected in the course of Level 2 processing. A user will be able to do this by reusing the desired components of Level 2 processing, plugging in (via Python `import` directives in the appropriate configuration file) the modules for their custom measurement, and executing the pipeline. The ***Science Pipeline Toolkit*** (WBS 02C.01.02.03) will provide the necessary components to support user-driven construction and execution of custom pipelines.

### 1.3.4 Science Data Quality Analysis Pipeline and Toolkit

Science Data Quality Analysis requirements are described in the Data Quality Assurance Plan ([LSE-63](#)) document. They will be implemented by the ***SDQA Pipeline*** (WBS 02C.01.02.02; the data collection backend) and the ***SDQA Toolkit*** (WBS 02C.01.02.02; the data analysis front-end).

LSST QA will include four main components, which to some extent reflect the Level 1-3 structure of LSST data products. Level 0 QA is software development related, Level 1 QA relates to nightly operations, Level 2 QA relates to data releases, and Level 3 QA is science based.

- **Level 0 QA** includes the extensive and thorough testing of the DM subsystem during the pre-commissioning phase, as well as the tests

of software improvements during the commissioning and operations phases (regression tests based on pipeline outputs and input truth). A common feature of Level 0 QA is the use of LSST simulations products, or any other dataset where the truth is sufficiently well known (e.g., the use of high-resolution observations from space telescopes to test resolved/unresolved object separation algorithms). The main goal of Level 0 QA is to quantify the software performance against these known expected outputs (e.g., to measure the completeness and false positive rate for an object finder; to measure the impact of blended sources on pipeline outputs; to measure the performance of calibration pipelines and MOPS), and to test for algorithm implementation problems (a.k.a. “coding bugs”).

- **Level 1 QA** assesses the system status and data quality in real time during commissioning and operations. Its main difference from other observatory, telescope, and camera status reporting tools will be heavy reliance on the massive science imaging data stream (in addition to various telemetry and metadata generated by the subsystems). This level is tasked with nightly reporting of the overall data quality, including the nightly data products (difference images and transient source event stream) and calibration products. Real-time information about observing conditions, such as sky brightness, transparency, seeing, and about the system performance, such as the achieved faint limit, will be delivered by Level 1 QA. Because the actual science data stream will be analyzed, Level 1 QA tools will be in a good position to discover and characterize subtle deterioration in system performance that might not be easily caught by tools employed by the telescope and the camera subsystems for self-reporting purposes.
- **Level 2 QA** assesses the quality of data products scheduled for the Data Releases, and provides quantitative details about data quality for each release (including the co-added image data products, and the properties of astrometrically and photometrically variable objects). This level also performs quality assessment for astrometric and photometric calibration, as well as for derived products, such as photometric redshifts for galaxies and various photometric estimators for stars. Subtle problems with the image processing pipelines and systematic problems with the instrument will be discovered with Level 2 QA.

- **Level 3 QA** quality assessment will be based on science analysis performed by the LSST user community. LSST will not develop Level 3 QA tools, but Level 0-2 visualization and data exploration tools will be made available to the community to form a basis on which Level 3 tools can be built. Common features expected for tools at this level are sensitivity to subtle systematic issues not recognized by Level 2 QA, and feedback about data quality to the project by external teams. It is envisioned that especially useful Level 3 QA tools would be migrated to Level 2 QA.

## 2 Level 1 Pipelines

### 2.1 Single Frame Processing Pipeline (WBS 02C.03.01)

#### 2.1.1 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.



### 2.1.2 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 functionality is assumed to be the same as that used by SDSS *Photo* pipeline [15].

**2.1.2.1 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 patterns 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.

**2.1.2.2 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).

**2.1.2.3 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).

**2.1.2.4 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.

### 2.1.3 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;

### 2.1.4 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/ip_isr), [https://github.com/lsst/meas\\_algorithms](https://github.com/lsst/meas_algorithms), [https://github.com/lsst/meas\\_astrom](https://github.com/lsst/meas_astrom), [https://github.com/lsst-dm/legacy-meas\\_mosaic](https://github.com/lsst-dm/legacy-meas_mosaic), [https://github.com/lsst/pipe\\_tasks](https://github.com/lsst/pipe_tasks).

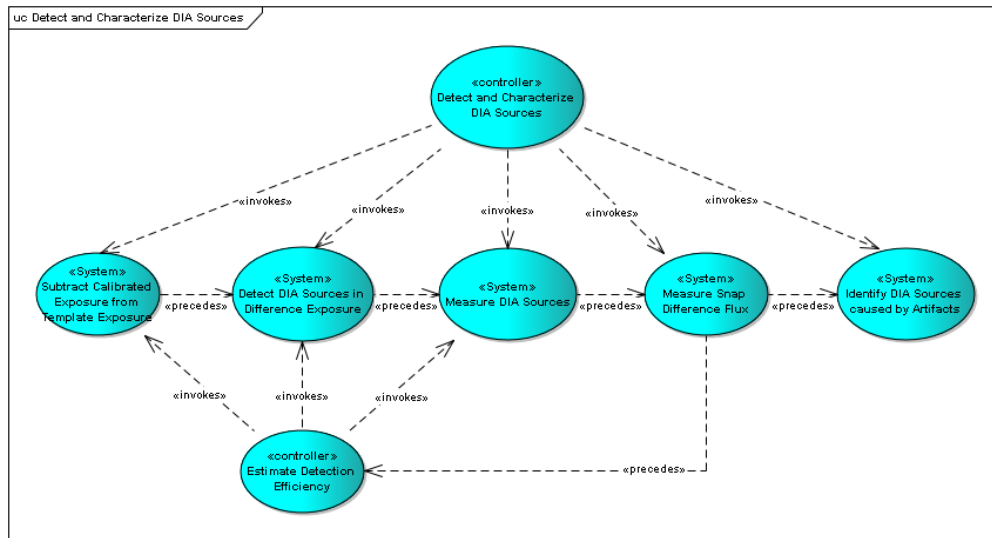


Figure 5: Image Differencing Pipeline Use Case Diagram

## 2.2 Image Differencing Pipeline (WBS 02C.03.04)

### 2.2.1 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.

### 2.2.2 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 §2.2.2.1).

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 5.

**2.2.2.1 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 (§4.4.2.4), but with the model being a mixture of positive and negative point sources, instead of Sérsic profiles.

### 2.2.3 Constituent Use Cases and Diagrams

Subtract Calibrated Exposure from Template Exposure; Identify DIA Sources caused by Artifacts; Perform Preccovery 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;

### 2.2.4 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](https://github.com/lsst/ip_diffim). The current prototype, while functional, will require a partial redesign to be transferred to construction to address performance and extensibility concerns.

## 2.3 Association Pipeline (WBS 02C.03.02)

### 2.3.1 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.

### 2.3.2 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.

### 2.3.3 Constituent Use Cases and Diagrams

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

### 2.3.4 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.



## 2.4 Alert Generation Pipeline (WBS 02C.03.03)

### 2.4.1 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.

### 2.4.2 Baseline Design

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

### 2.4.3 Constituent Use Cases and Diagrams

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

### 2.4.4 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.

## 2.5 Moving Object Pipeline (WBS 02C.03.06)

### 2.5.1 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<sup>5</sup> 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*.

### 2.5.2 Baseline Design

The baseline NightMOPS design is to adopt and adapt an existing ephemeris computation pipeline such as OrbFit<sup>6</sup> or OpenOrb<sup>7</sup>. 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*).

### 2.5.3 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;

---

<sup>5</sup>Also sometimes referred to as ‘Moving Object’

<sup>6</sup><http://adams.dm.unipi.it/orbfit/>

<sup>7</sup><https://github.com/oorb/oorb>

#### 2.5.4 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](https://github.com/lsst/mops_daymops) and [https://github.com/lsst/mops\\_nightmops](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.

## 3 Calibration Pipelines

### 3.1 Calibration Products Pipeline (WBS 02C.04.02)

#### 3.1.1 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]<sup>8</sup>. 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.

#### 3.1.2 Baseline Design

**3.1.2.1 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;

---

<sup>8</sup>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*).

**3.1.2.2 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<sup>9</sup> 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.

**3.1.2.3 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 possible 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,

---

<sup>9</sup>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 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 this 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<sup>10</sup> and model both as a function of position (and possibly time);

---

<sup>10</sup>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.

**3.1.2.4 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.

### 3.1.3 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;

### 3.1.4 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.



## 3.2 Photometric Calibration Pipeline (WBS 02C.03.07)

### 3.2.1 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.

### 3.2.2 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](#)).

### 3.2.3 Constituent Use Cases and Diagrams

Perform Global Photometric Calibration;

### 3.2.4 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.

### **3.3 Astrometric Calibration Pipeline (WBS 02C.03.08)**

#### **3.3.1 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.

#### **3.3.2 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.

#### **3.3.3 Constituent Use Cases and Diagrams**

Perform Global Astrometric Calibration;

#### **3.3.4 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.

## 4 Level 2 Pipelines

### 4.1 PSF Estimation Pipeline (WBS 02C.04.03)

#### 4.1.1 Key Requirements

PSF Estimation pipeline must enable the estimation of the zero-intensity point spread function at any point on the focal plane with residual ellipticity correlations at the levels required by LSR-REQ-0097.

#### 4.1.2 Baseline Design

LSST's PSF models will be implemented as a plugin. Simple PSF estimation algorithms (e.g., `pcaPsf`) will be implemented by the Single Frame Processing pipeline (WBS 02C.03.01). The `CoaddPsf` algorithm will be implemented in the Image Coaddition Pipeline (WBS 02C.04.04).

Current state-of-the-art PSF estimation methods typically use some basis functions (e.g.,  $\delta$ -functions, PCA) and a spatial model (e.g., polynomials) to estimate the PSF. This is not expected to be sufficient to reach LSST requirements. We have therefore adopted the following baseline:

- Decompose the PSF into a component due to the atmosphere and a component due to the telescope and camera system.
- Estimate the telescope+camera component using the wavefront sensor information (the reconstructed wavefront) and camera metrology (the laboratory or on-sky measurement of  $z$  offsets of individual sensors)
- Estimate the atmospheric contribution by modelling the PSF from bright, isolated, stars, and interpolating using Kriging.

This estimation will be performed on the full focal plane. It will make use of wavefront sensor information and camera metrology provided by the Telescope & Site subsystem, and use the per-CCD estimates of the PSF and cutouts of bright stars (PSF estimation candidates) to determine the atmospheric contribution. The latter may be augmented by including a physical model of the atmosphere.

In thick deep depletion devices such as LSST, the electric field from photoelectrons distorts the pixel grid, giving bright objects a broader effective PSF than faint objects. We expect most and perhaps all correction for this

effect to happen in Instrument Signature Removal (part of WBS 02C.03.01), but we may need to mitigate residuals from that correction in PSF estimation, likely by iteratively forward-modeling the estimated zero-intensity PSF until convergence is achieved. The same forward-modelling algorithm will allow us to estimate the PSF in undersampled imaging, as is expected for roughly the best quartile of the seeing distribution; the resulting estimate should not be of lower quality than that of an equivalent well-sampled PSF assuming that enough stars are included in the model.

It is necessary to take account of chromatic effects from both the atmosphere and the telescope optics when creating the PSF model. It is well understood how the optical and atmospheric components of the PSF should depend on wavelength, so any model that separates these components effectively should automatically include the correct chromatic dependency, allowing it to be evaluated at any wavelength or (assuming color-independent morphology) the SED of a particular source. This will allow us to validate and constrain the chromatic dependence of the model using the known or inferred SEDs of stars.

### 4.1.3 Constituent Use Cases and Diagrams

Perform Full Focal Plane PSF Estimation.

### 4.1.4 Prototype Implementation

Prototype code for wavefront reconstruction has been developed by the LSST Telescope and Site group. We expect this code will be rewritten in Construction to follow LSST Data Management standards and be straightforward to incorporate into the PSF Estimation pipeline.

The remaining components of the PSF Estimation pipeline have not been prototyped at this time.

## 4.2 Image Coaddition Pipeline (WBS 02C.04.04)

### 4.2.1 Key Requirements

The Image Coaddition Pipeline will produce coadds given a stack of input calibrated exposures. Generated coadds must be characterized, at a pixel level, for validity and variance (i.e., have a Mask plane and a Variance plane). It must be capable of supporting these coaddition schemes within a single band:

- Direct;
- PSF-matched;
- Likelihood (“Kaiser”).

The implementation of likelihood coaddition used to generate coadds suitable for the Detection and Deblending system (WBS 02C.04.05) may utilize extensive approximations that limit the applicability of the coadds in the Object Characterization Pipeline (WBS 02C.04.06). A prototype version of likelihood coaddition algorithm suitable for object characterization must also be developed within this WBS; whether that more advanced algorithm progresses beyond prototype stage will be determined within WBS 02C.04.06.

The baseline implementation of likelihood coaddition may utilize extensive approximations that limits the applicability of the coadds in the Object Characterization Pipeline (WBS 02C.04.06). A higher quality implementation need only be implemented in response to emergent object characterization requirements during construction.

The Image Coaddition Pipeline must be capable of combining coadds from multiple bands into a single, multi-band, coadd. When doing so, it must support both the “ $\chi^2$ ” scheme described by [24] and a scheme in which inputs are weighted by the expected SED.

When non-PSF-matched coadds are produced, the effective PSF and its variation across the full coadd must be determined and retained.

Outliers appearing in a small subset of input images (e.g. transients or moving objects) must not appear in the final co-adds.

The Image Coaddition Pipeline must make it possible to generate the coadds from background-subtracted input images or perform partial background subtraction via matching during coaddition (4.2.2.3).

The pipeline must be capable of generating coadds in varying geometries and tessellations of the sky, plugged in at runtime via Python modules and configuration directives. To reduce its memory footprint, it should be capable of generating the coadds in small patches set by the operator at runtime via configuration directives.

The pipeline should be capable of co-adding undersampled images without aliasing.

For performance reasons, the pipeline should be able to simultaneously generate multiple coadds from the same input stack, avoiding unnecessary rewarping of inputs to generate each coadd.

## 4.2.2 Baseline Design

**4.2.2.1 Coaddition** The Coaddition Pipeline will apply the results of the relative astrometric solution (developed as a part of Single Frame Pipeline, WBS 02C.03.01) to the input images, warp (§4.2.2.2) them to a common coordinate system (“sky map”; §4.2.2.4) and coadd the pixels. Multiple coadds may be created from a given set of input images by combining subsets of the input defined explicitly by the operator or algorithmically by quality cuts.

To accurately measure the colors of galaxies (e.g., for photometric redshifts) in the presence of intra-object color gradients, seeing changes and imperfect galaxy models, the baseline implementation will be capable of producing “PSF-matched coadds” using well known PSF-matching algorithms. The Object Characterization Pipeline (WBS 02C.04.06) will measure object properties on both direct (centroids and shapes) and PSF-matched (fluxes) coadds, as well as fitting galaxy models to the direct coadds.

Likelihood coaddition is required for detection in both the Image Difference Pipeline (WBS 02C.03.04) and the Detection and Deblending system (WBS 02C.04.05). It also presents a possible alternative to direct and PSF-matched coaddition for object characterization. Assessing the suitability of likelihood coadds for use in LSST data releases is contained within the Object Characterization Pipeline (WBS 02C.04.06); the Image Coaddition Pipeline will deliver likelihood coadds as required to satisfy that requirement.

The Detection and Deblending system (WBS 02C.04.05) will make use of multi-band coadds. These may be generated either using the  $\chi^2$  algorithm [24] or based on SED-weighted combinations of per-band coadds. The Image Coaddition Pipeline will make it possible to generate either of these on

demand.

If the seeing is not constant it becomes impossible to produce a coadd with a PSF varying in a continuous fashion over the field, unless the data is deliberately degraded by convolving the inputs to a common PSF (known as “PSF matching”). In order to enable dealing with the discontinuous PSF, the Coaddition Pipeline will construct a “CoaddPsf” (§4.2.2.5) PSF model, which is a sum of the PSFs of the input images at each point of interest (as proposed as part of ‘StackFit’ [12]).

Determining the background from individual visits separately is problematic (because different choices can be made in each, especially at the edge of a CCD; and because extended, faint astrophysical flux is misinterpreted as background), commonly manifesting as dark rings around very bright stars and the suppression of extended flux around galaxies. Therefore, the pipeline will include the capability to perform “background matching” (§4.2.2.3) to produce a coadd with a high signal-to-noise realization of the background in a single reference exposure. This background will be measured over large scales and be possible to subtract with a high degree of accuracy. Background matching will be the default mode of operation. Nevertheless, it will also be possible to produce coadds by subtracting the background first using Sky Background determination and subtraction components developed in the Single Frame Processing Pipeline (WBS 02C.03.01).

**4.2.2.2 Warping** To warp an image from the detector frame to the sky frame, a resampling kernel will be used. The kernel is set according to the sub-pixel position on the input image of the centre of the corresponding output pixel. This must support Lanczos (of configurable order), bilinear or nearest-neighbour kernels, with the default being a 3rd-order Lanczos (with  $10^6$  cache realizations), as a compromise between the infinite *sinc* function and the need for speed. Other schemes (e.g. polynomial interpolation) may also be available.

The LSST focal plane will be undersampled at seeing less than  $0.4''$ . It must be possible to warp and coadd undersampled images without introducing aliasing. The approach taken should provide the best possible image quality while maintaining adequate computational throughput; a slight loss of image quality is acceptable.

**4.2.2.3 Background Matching** Background matching will be implemented to enable reaching the maximum depth in the coadds and preserve the astrophysical backgrounds. We adopt as our baseline the algorithm of Huff et al. [9], extended to two-dimensional data.

The common practice of subtracting the background from each input individually removes features on moderate scales (such as the outer haloes of galaxies, and Galactic cirrus) and can be unstable (especially when the features appear at the edge of a frame) causing increased noise.

Instead, the following algorithm (following Huff et al.) will be implemented: the pipeline will choose one or more reference exposures, and match the backgrounds of each of the other exposures to the reference. This will be done by subtracting the reference from each of the other exposures (which mostly removes astrophysical sources, especially the extended sources which normally contaminate background measurement) and fitting a background model to the difference between the backgrounds<sup>11</sup>. These models will then be subtracted from the other inputs, so that all exposures have the same large-scale background as the reference exposure.

The coadd produced with the above algorithm will retain the extended astrophysical features at high signal-to-noise, and the background can be carefully removed over multiple patches at once. The subtracted image also provides an opportunity to identify sharp features such as optical ghosts, glints and other contaminants that can be masked.

**4.2.2.4 Sky Tessellation and Coadd Projections** The “skymap” is a tessellation of the sky, providing suitable pre-defined coordinate systems for operations on the sky such as coaddition. The sky map divides the sky into “tracts”. For convenience and parallelism, each tract is sub-divided into “patches”. Tracts and patches may overlap, so that sources are not lost in the gaps. Tessellations will be pluggable Python modules.

The baseline tessellation is one using a stereographic dodecahedron. The sky will be subdivided into 12 overlapping<sup>12</sup> *tracts*, spanning approximately  $75 \times 72$  degrees. The sky will be stereographically projected onto the tracts<sup>13</sup>,

---

<sup>11</sup>It is anticipated this code will be shared with the Sky Background determination modules from the Single Frame Processing pipeline.

<sup>12</sup>We’re planning for 3.5 degrees of overlap, roughly accommodating a full LSST focal plane.

<sup>13</sup>See <https://dev.lsstcorp.org/trac/wiki/DM/SAT/SkyMap> for details.



and pixelized into (logical) images 2.0 x 1.9 megapixels in size (3.8 terapixels in all). Physically, these large images will be subdivided into smaller, approximately  $2k \times 2k$  pixel, non-overlapping, *patches*, though that subdivision is to be transparent to clients. Clients will be able to request arbitrarily chosen regions in each tract<sup>14</sup>, and receive them back as afw **Exposure** objects.

**4.2.2.5 CoaddPsf** One of the main challenges in producing quality measurements from non-PSF-matched coadds is the complexity of the effective point-spread function on the coadd. Because the PSF has discontinuities at the location of chip boundaries, modeling approaches based on interpolating with smooth functions cannot be used. Instead, when creating a coadd image, we also combine the PSF models of all the input exposures, using an approach similar to that devised by Jee and Tyson [11]. This combination is lazy, in that we simply store the PSF models, their bounding boxes, and the coordinate transforms that relate them to the coadd pixel grid. Then, when measuring an object on the coadd, we obtain the coadd PSF model at that location by coadding the PSF models of all exposures that contributed to the relevant part of the coadd, after warping them by the appropriate coordinate transform. We approximate the PSF as spatially constant in the region of an individual object.

#### 4.2.2.6 Image artifact rejection

TODO:  
 This is a placeholder by JDS — a fuller description of the approach is required.

Artifacts which may be seen on images include:

- Optical ghosts and glints;
- Satellite and asteroid trails;
- Transient and variable sources.

These must not be included in coadds. Furthermore, the **CoaddPsf** must be updated to take account of their removal.

<sup>14</sup>Up to some reasonable upper limit, defined by available memory

**4.2.2.7 Differential Chromatic Refraction** The baseline design for difference imaging (WBS 02C.03.04) calls for creation of coadds in bins of different airmass to account for differential chromatic refraction. However, initial work suggests better results may be achieved by modelling and compensating for DCR when templates are being constructed [4]. A similar approach will also be investigated when building coadds for deep detection (WBS 02C.04.05). A prototype DCR compensation algorithm should be developed in this WBS based on an assumed SED; further development will be driven by future requirements from other WBS elements.

### 4.2.3 Constituent Use Cases and Diagrams

Create Deep Coadd Exposures; Create Short Period Coadd Exposures; Coadd Calibrated Exposures; Create Best Seeing Coadd Exposures; Create PSF-matched Coadd Exposures; Create Template Exposures;

### 4.2.4 Prototype Implementation

A prototype implementation of all major components of the Coaddition Pipeline baseline design has been completed in LSST Final Design Phase. We expect it will be possible to transfer a significant fraction of the existing code into Construction, for continued improvement to meet LSST performance and accuracy requirements.

The existing prototype has been extensively tested with image simulation inputs, as well as real data (SDSS Stripe 82). Using Stripe 82 data, it demonstrated the benefits of background matching and CoaddPsf approaches.

The design and performance of the current prototype pipeline is described in the Summer 2012 Data Challenge (<http://ls.st/vho>) and Winter 2013 Data Challenge Report (<http://ls.st/ofk>). The prototype codes are available in the `coadd_*` git repositories browsable through the LSST git repository browser at <https://github.com/lsst>.

## 4.3 Object Detection and Deblending (WBS 02C.04.05)

### 4.3.1 Key Requirements

The Object Detection and Deblending system is responsible for the detection of sources on coadds produced by the Coaddition Pipeline and for disentangling collections of superimposed objects into their constituent parts.

**TODO:**  
 There are no requirements specified for the coverage maps. For now,  
 I am adding a vague statement here. We should make this more  
 concrete, perhaps by updating the DMSR.

It must also construct maps describing completeness and selection effects in the LSST survey.

### 4.3.2 Baseline Design

The baseline is based on the SDSS detection and deblending algorithms [15, 16]. See also the notes on Deep Processing available on Confluence<sup>15</sup>.

**4.3.2.1 Object detection** Object detection on a single image is performed by correlating the image with the CoaddPsf PSF and searching for peaks above the preset threshold. Multiple adjacent peaks will be detected and merged to reduce spuriously detected objects due to noise or object substructure.

To enable detection of extended objects, the Pipeline will perform detection on recursively binned coadds.

To construct the object catalog, detection may be performed on multiple different types of coadd, as generated by the Image Coaddition Pipeline (WBS 02C.04.04). These may include coadds from multiple separate bands, multi-band coadds, and coadds made with different algorithmic approaches (direct, likelihood, etc). Detection is performed on each coadd independently, then a *peak association* procedure combines the detection outputs from different coadds, merges footprints that cover the same patch of sky and associates peaks that (we believe) correspond to the same object. This association procedure may also incorporate other sources, e.g. transient and moving

<sup>15</sup><https://confluence.lsstcorp.org/display/DM/Deep+Processing>

sources detected in image differencing (WBS 02C.03.04) and known objects from other surveys. The result is a set of merged, non-overlapping footprints, each containing one or more peaks.

Per §4.2.2.4, the sky is tessellated into overlapping tracts and patches. In general, these may be processed independently, but care is required to ensure that objects which appear in the overlapping regions appropriately merged rather than being duplicated.

[ **TODO:**  
Define how this merge will take place. ]

The **DPDD** requires that the resultant collection of peaks and their membership in the footprints be provided to end users as a **CoaddSource**

**4.3.2.2 Deblender** At the depths probed by LSST images, many of the sources are superimposed on each other on the sky (“blended”), which makes detecting and measuring such sources difficult. Often this blending is not severe, but the footprints of reasonably well separated objects can overlap (and therefore merge together) slightly. In other cases, distinct objects will be superimposed directly on more extended objects (e.g., a star on a resolved galaxy).

In order to disentangle the multiple objects, we adopt a baseline for the “deblender” based on the algorithms used in SDSS.

The baseline deblender assumes that discrete sources generally have twofold rotational symmetry. When one side of a source is blended, we can recover its appearance by examining the symmetric side. The deblender begins by building these symmetric templates for each source. Next, the flux in each pixel is split among the blended source in proportion to their templates. The deblender produces cutouts of each source in a blended group, so that the measurement algorithms (fluxes, galaxy shapes, etc) need not know that the source was blended. This deblending algorithm works well in practice for moderately crowded fields (as demonstrated by SDSS).

This SDSS-style deblender will be run on one or more single-band coadds. When deblending across multiple bands, the deblender must ensure that consistent results are achieved across all bands.

After deblending, a set of PSF-convolved models will be fit to the deblended children. The initial templates will then be replaced by these models,

convolved with the appropriate PSF, and flux assigned as before. The use of physically-motivated templates will help with identification and reduction of non-physical deblends, generalization to multi-epoch data, and data with very different image quality. Fitting these model templates will also allow us to improve centroids of objects whose positions were affected by their neighbors.

In areas of significant stellar crowding (i.e., Galactic plane, star clusters), this approach lends itself to imposition of appropriate template priors (i.e., the correct template being that of a point source convolved with the PSF). This, effectively, makes the deblender into a crowded field code, allowing this baseline to satisfy the requirements for crowded field photometry.

[ **Note:**  
The “divide and conquer” technique needs to be properly fleshed out. ]

The deblender must be capable of handling extremely large blended sources using a so-called “divide-and-conquer” approach to split them up and handle them in manageable pieces.

**4.3.2.3 Sky coverage** This is an area of active and ongoing research in DES and HSC. We intend to learn from their experience before baselining a design. Relevant prior work includes STOMP<sup>16</sup>, Mangle<sup>17</sup> and Venice.

### 4.3.3 Constituent Use Cases and Diagrams

[ **Note:**  
The UML models are comprehensively outdated. ]

Detect and Characterize AstroObjects; Detect Sources on Coadds;

### 4.3.4 Prototype Implementation

A functional detection and deblending system was completed in the LSST Final Design Phase and has and extensively tested with image simulation inputs as well as real data (SDSS Stripe 82). A significant fraction of this code has been transferred into Consturction.

<sup>16</sup><https://code.google.com/p/astro-stomp/>

<sup>17</sup><http://space.mit.edu/~molly/mangle/>

The multi-band processing architecture has been prototyped on HSC and is now available in the LSST codebase.

The detection functionality is a part of the `afw` package browsable through the LSST git repository browser at <https://github.com/lsst/afw>; deblending functionality is in the `meas_deblender` package at [https://github.com/lsst/meas\\_deblender](https://github.com/lsst/meas_deblender); the multi-band processing architecture in `pipe_tasks` at [https://github.com/lsst/pipe\\_tasks](https://github.com/lsst/pipe_tasks).

## 4.4 Object Characterization Pipeline (WBS 02C.04.06)

### 4.4.1 Key Requirements

**Note:**

Here we account to itemize all the pieces of functionality that the Obj Char Pipeline needs to meet the DPDD (or some Bosch Blended Measurement [8] modified version thereof). Some of these are arguably design decisions rather than requirements on the output, but the contents of the DPDD are explicitly driven by the algorithmic and design decisions – we commit to providing the results of certain measurements, which makes the measurements requirements.

Given one or more cutouts of a detected object, observed in one or more epochs, the Object Characterization Pipeline must perform all measurements required to generate the Level 2 Catalogs as described by the [DPDD](#), within the computational budget allotted by the LSST sizing model.

Functionally, the components of this pipeline must be available for incorporation into or use by other pipelines. In particular, both the Single Frame Processing Pipeline (WBS 02C.03.01) and the Imaging Differencing Pipeline (WBS 02C.03.04) will delegate source characterization to the algorithms defined herein.

The Object Characterization Pipeline must be capable of:<sup>18</sup>

- Measurement of surface brightness in an elliptical aperture;
- Measurement of flux using a linear least squares fit with the model PSF (“PSF flux”);
- Computing flux as the dot product of an elliptical Gaussian weight function with the image (“Gaussian flux”);
- Measurement of Petrosian flux and radius;
- Measurement of Kron flux and radius.

The Object Characterization Pipeline must be capable of fitting the following source models:

- Moving point source (required by Single Frame Processing);
- Trailed point source (required by the Image Differencing Pipeline);

---

<sup>18</sup>The [DPDD](#) specifies detailed algorithmic requirements for each of these measurements

- Hybrid galaxy-moving point source (required for Level 2 Catalogs).

[ **Note:**  
The **DPDD** specifies separate point source and galaxy models;  
here we follow Bosch. ]

It must be possible to fit a point source model to an image while holding the position, motion, shape and deblending parameters (if applicable) fixed (to perform “forced photometry”).

It must be possible to perform *simultaneous fitting*, whereby we fit the models for multiple objects simultaneously.

It must be possible to fit objects using either numerical optimization or by Monte Carlo sampling.

It must be possible to characterize sources in multiple epochs by fitting a parameterized model to each epoch (rather than co-adding the data and fitting to the result). This approach, which we term “MultiFit”, is explored further in §4.4.2.2.

When measuring the same object in coadds covering multiple bands, it must be possible to ensure the results are consistent (contain the same set of objects) in all filters. This is achieved by basing the measurement routines on the output of the peak association procedure defined in §4.3.2.1 (Object Detection and Deblending; WBS 02C.04.05).

It must be possible to derive the following quantities based on the measurements made using “best practice” algorithms at the time of commissioning:

- Color of the object measured in the “standard seeing”;
- Extendedness of the object;
- Photometric redshift likelihood samples;

It must be possible to characterize a time series of forced photometry measurements of the same object for aperiodic and periodic variability<sup>19</sup>.

When constructing Level 2 Catalogs, a probabilistic star-galaxy classification should be available for each object.

---

<sup>19</sup>The **DPDD** suggests specific metrics, but cautions they should be revised during construction.



**Note:**

The DPDD only specifically calls out DCR in the context of MultiFit for a point source model. That is presumably incomplete; our requirements for DCR in object characterization require fleshing out.

Per §2.2.2, the baseline design requires that measurement of point sources on likelihood coadds (also known as Kaiser coadds or detection maps) must be possible. It is possible that much of the other object characterization techniques described in this baseline may be reconsidered and replaced with measurements made on likelihood coadds [8, §8.2.1]; see also the discussion of the Image Coaddition Pipeline, WBS 02C.04.04. To facilitate the best possible science outputs, this baseline allocates time to explore the scientific potential of likelihood coaddition in terms of both theoretical work the mathematical formalism and the development of proof-of-concept measurements, with the possibility of rebaselining during construction if necessary.

**Note:**

Can we define specific success criteria for experimenting with likelihood coaddition?

#### 4.4.2 Baseline Design

**4.4.2.1 Single-epoch Characterization** Single-epoch (including coadd) object characterization will primarily rely on forward-modeling. Models will be convolved with the independently estimated PSF and compared with the pixel data until  $\chi^2$  is minimized.

**4.4.2.2 MultiFit** For multi-epoch characterization and some science measurements — particularly shape measurement for weak lensing — measurement on a coadd image may not provide the necessary precision and control of systematics. To perform these measurements, we baseline the “MultiFit” approach [6, 7], in which a parametrized model for each astronomical object is fit simultaneously to all of the data in which that object appears. Rather than transform and combine the data, we instead transform the model to the coordinate system of each exposure, convolve it by the appropriate PSF, and compare it to the data from that exposure. Measurements from the coadd

will be used as a starting point, so the MultiFit processing will proceed by iterating over the catalog generated from the coadd, loading postage stamps from the original images, and fitting to these data. Note that the output measurements will generally have the same form as the coadd-based measurements - one set of measurements per astronomical object. There is a single set of the parameters that describe each object, not a different set of parameters for each exposure in which it appears.

Because it does not involve transforming noisy data (which is usually lossy in some sense), and instead transforms analytic models, a MultiFit approach is theoretically optimal for measurements that can be framed as the results of model-based fits to the data (note that not all measurements can be framed in such a way, but those that cannot usually do not properly account for the PSF). While it may be possible to construct an optimal coadd that would also be theoretically optimal, we expect that the difficulty in creating such a coadd (i.e., perfectly tracking the covariances between pixels introduced by resampling to a common pixel grid) would be prohibitively complex. On the other hand, for many measurements a non-optimal coadd may be practically sufficient, in the sense that the improvement produced by a MultiFit-based measurement of the same would be negligible. In these cases, the coadd measurement is likely to be far more efficient computationally. Whenever possible, then, we will do as much work as possible on the coadd first, and only use a MultiFit approach to “tune up” the final result. And when this final tuning is determined to be unnecessary, it can be skipped entirely.

To further make the MultiFit approach computationally tractable, we baseline use the multi-shapelet MultiFit algorithm implementation, as described in [7].

**4.4.2.3 Point source model fit** To satisfy the point-source model fit requirements, we model each observed object as a point source with finite proper motion and parallax and with constant flux (allowed to be different in each band). This model is a good description for stars and other unresolved sources. Its 11 parameters will be simultaneously constrained using information from all available observations in all bands. Per §5.2.1 of the DPDD, the fitting procedure will account for differential chromatic refraction. The fitting procedure will account for differential chromatic refraction. MultiFit will be used to perform the fit.

#### 4.4.2.4 Bulge-disk model fit

[ **Note:**  
Deprecated by Bosch. ]

To satisfy the bulge-disk model fit requirements, the pipeline will be capable of modelling objects as a sum of a de Vaucouleurs (Sérsic  $n = 4$ ) and an exponential (Sérsic  $n = 1$ ) component. This model is a reasonable description of galaxies. The object is assumed not to move (i.e., have zero proper motion). The components share the same ellipticity and center. The model is independently fit to each band. There are a total of 8 free parameters, which will be simultaneously constrained using information from all available epochs for each band. Where there's insufficient data to constrain the likelihood (eg., small, poorly resolved, galaxies, or very few epochs), the pipeline will have the capability to take into account priors that limit the range of its sampling.

In addition to the maximum likelihood values of fitted parameters and their covariance matrix, the pipeline shall be capable of sampling independent samples from the likelihood function (or posterior). MultiFit will be used to perform the fit.

**4.4.2.5 Trailed model fit** The pipeline shall be capable of fitting a trailed 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 line segment instead of a mixture of Sérsic profiles.

**4.4.2.6 Dipole model fit** Dipole model fitting is provided directly within the Image Differencing Pipeline (WBS 02C.03.04; §2.2.2.1); this section is maintained to avoid renumbering breaking references to this document.

#### 4.4.2.7 Hybrid model fit

[ **Note:**  
Required by Bosch. Need specifications as to the form the model will take, particularly as relates to requirements for galaxy modelling in DLP; we baseline it as being “advanced” relative to CModel, but specify nothing else at present. ]

The pipeline shall be capable of fitting a hybrid model that transitions between a moving point source and a galaxy model. Because a static point source is a limit shared by both models, the transition is continuous. Fitting will be performed by both numerical optimization and Monte Carlo sampling; where there is insufficient data to constrain the likelihood (eg., small, poorly resolved, galaxies, or very few epochs), the pipeline will have the capability to take into account priors that limit the range of sampling. The model is independently fit to each band.

**4.4.2.8 Centroids** Centroids will be computed independently for each band using an algorithm similar to that employed by SDSS [15]. Information from all epochs will be used to derive the estimate. These centroids will be used for adaptive moment, Petrosian, Kron, standard color, and aperture measurements.

**4.4.2.9 Adaptive moments** Adaptive moments will be computed using information from all epochs, independently for each band, using the algorithm of Bernstein & Jarvis (2002).

**4.4.2.10 Petrosian and Kron fluxes** Petrosian and Kron radii and fluxes will be measured in standard seeing using self-similar elliptical apertures computed from adaptive moments. The apertures will be PSF-corrected and *homogenized*, convolved to a canonical circular PSF. This is in order to derive a definition of elliptical apertures that does not depend on seeing. For example, for a large galaxy, the correction to standard seeing will introduce little change to measured ellipticity. Corrected apertures for small galaxies will tend to be circular (due to smearing by the PSF). In the intermediate regime, this method results in derived apertures that are relatively seeing-independent. Note that this is only the case for *apertures*; the measured flux will still be seeing dependent and it is up to the user to take this into account.

The radii will be computed independently for each band. Fluxes will be computed in each band, by integrating the light within some multiple of *the radius measured in the canonical band*. The shape of the aperture in all bands will be set by the profile of the galaxy in the canonical band alone. This procedure ensures that the color measured by comparing the flux in different bands is measured through a consistent aperture. See <http://www.sdss>.

[org/dr7/algorithms/photometry.html](http://org/dr7/algorithms/photometry.html) for details. The pipeline shall be capable of computing radii enclosing 50% and 90% of light.

The baseline for both Petrosian and Kron flux implementation is to derive these as an a mathematical transformation of aperture surface brightness measurements (see below).

**4.4.2.11 Aperture surface brightness** Aperture surface brightness will be computed in a variable number, depending on the size of the source, of concentric, logarithmically spaced, PSF-homogenized, elliptical apertures, convolved to standard seeing.

**4.4.2.12 Resolved/Unresolved object separation** We baseline the resolved/unresolved object separation algorithm based on the ratio of PSF and model fluxes. We use the extendedness criterion as defined by the HSC [10]:

$$\text{extendedness} = (0.95 \times \text{flux.gaussian} < \text{flux.psf}) ? 0.0 : 1.0$$

**4.4.2.13 Star-Galaxy classification**

[	<p><b>Note:</b> Not explicitly required in any of the requirements documents, but common consensus is that this will be necessary. Should expand the below.</p>	]
---	---	---

We expect to adopt the technique used by HSC: a sigmoid based on the difference between the model and Gaussian flux, trained using Support Vector Machines.

**4.4.2.14 Colors** The [DPDD](#) requires that the pipeline provide the color for each object using some (unspecified) algorithm which is guaranteed to be independent of seeing and suitable for determination of photometric redshift (§4.4.2.15).

Our default assumption is that all model fitting on coadds is performed to direct coadds, which will produce the best estimates of structural parameters and a more appropriate starting point for Monte Carlo sampling. However, measuring galaxy colours to the required level of fidelity may require fitting to PSF-matched coadds [8, §8.2.2]. We therefore plan to prototype color measurement on both direct and PSF-matched coadds, preferring the former

for reasons of efficiency but being ready to take the latter to operations if required.

#### 4.4.2.15 Photometric Redshifts

[ **TODO:**  
Define properly. ]

DMS-REQ-0046 requires that a photometric redshift be calculated for all Objects. The [DPDD](#) requires that we provide 100 pairs of  $(z, \log L)$  likelihood samples for each object using an algorithm which is “widely accepted at the time of LSST commissioning”.

**4.4.2.16 Variability Characterization** Two groups of parameters are required to be provided (see `lcPeriodic` and `lcNonPeriodic` in the [DPDD](#)), designed to characterize periodic and aperiodic variability features. We baseline the metrics and algorithms described in [21] for production of these data products.

#### 4.4.3 Constituent Use Cases and Diagrams

Measure AstroObjects; Exposure Stack Measurements; Create Sky Coverage Maps; Perform Deblending and Association; Perform Forced Photometry; Characterize AstroObject Flux Variability;

#### 4.4.4 Prototype Implementation

A prototype implementation of all major components of the Object Characterization Pipeline baseline design has been completed in LSST Final Design Phase. The existing prototype has been extensively tested with image simulation inputs, as well as real data (SDSS Stripe 82). We expect it will be possible to transfer a significant fraction of the existing code into Construction, for continued improvement to meet LSST performance and accuracy requirements.

Missing from the current prototypes are the moving point source fit (implemented algorithms assume the source does not move), the trailed source fit, and the aperture, Kron and Petrosian magnitudes using elliptical apertures

(implemented algorithms assume the apertures are circular).

The design and performance of the current prototype pipeline is described in the Summer 2012 Data Challenge (<http://ls.st/vho>), Winter 2013 Data Challenge Report (<http://ls.st/ofk>), and Summer 2013 Data Challenge Report (<http://ls.st/grd>). The codes are available in the `meas_*` git repositories browsable through the LSST git repository browser at <https://github.com/lsst>.

## 5 Science Data Quality Analysis Pipeline

### 5.1 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).

### 5.2 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.

### 5.2.1 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

### 5.2.2 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

### 5.2.3 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

#### 5.2.4 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.

#### 5.2.5 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](https://github.com/lsst/testing_displayQA) git repository.

## 6 Science User Interface and Toolkit

### 6.1 Science Pipeline Toolkit (WBS 02C.01.02.03)

#### 6.1.1 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.

#### 6.1.2 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<sup>20</sup>. 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).

#### 6.1.3 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;

#### 6.1.4 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).

---

<sup>20</sup>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.

## **7 Algorithmic Components: AP**

Put algorithmic components for the alert production system here.

## 8 Algorithmic Components: DRP

Put algorithmic components for the data release production system here.

## 9 Software Primitives

### 9.1 Applications Framework (WBS 02C.03.05, 02C.04.01)

#### 9.1.1 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;<sup>21</sup>
- 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.

#### 9.1.2 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.

---

<sup>21</sup>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.

### 9.1.3 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>.

## 10 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](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](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  $\Omega_M$  and  $\sigma_8$  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).