The ZTF Science Data System (ZSDS)

Definitions: Software, Deliverables & Services

Version 1.3, February 26, 2016

Frank Masci, IPAC – Caltech

ZWICKY TRANSIENT FACILITY

CIN #XXX <not yet assigned>

Revision History

Date	Version	Author	Description
February 3, 2016	1.0	Frank Masci	Initial version
February 4, 2016	1.1	Frank Masci	Updated public-release schedule details
February 16, 2016	1.2	Frank Masci	Included internal feedback from Project Scientist and other local stakeholders
February 26, 2016	1.3	Frank Masci	Updated pipeline processing-flows according to feedback from collaboration

Table of Contents

1 PURPOSE

This document defines the high-level baseline design to accommodate the core deliverables, capabilities and services of the ZTF Science Data System (ZSDS) in accord with the Science Requirements (v1; Bellm 2015) and the Mid-Scale Innovations Program (MSIP) proposal (Kulkarni 2014).

The ZSDS shall consist of the data processing pipelines, data archives, infrastructure for long-term curation, and the services for data retrieval. The ZSDS will be housed at the Infrared Processing and Analysis Center (IPAC), Caltech.

In particular, the goals of this document are to:

- Advertise to the ZTF collaboration the specifics of all deliverables and services.
- Outline the high-level development effort needed to achieve the products and services. This includes anticipated algorithmic development at the pipeline level.
- Establish and confirm in advance what can realistically be achieved and delivered given the resources and schedule.

These will allow IPAC to architect a processing and archival system tailored to the precise deliverables. This will also enable better informed decisions on hardware purchases over the course of the project. It is important to note that taking the existing iPTF Data System as a blueprint for the ZSDS is only a presumption. There is no practical justification that the iPTF design will scale in some manageable and resourceful manner to sustain ZTF's data volumes and rates. A definition of all core deliverables and user-services, with some prioritization thereof, is therefore crucial.

An overview of the baseline system and functions is summarized in Section 3. This meshes closely with the planned ZTF development effort in Section 4 and deliverables in Section 5.

2 POINTS OF CONTACT

Frank Masci
(ZSDS Task Lead: fmasci@caltech.edu)

Steven Groom
(PTF/ZTF Archive Lead and IRSA Task Lead: sgroom@ipac.caltech.edu)

Lisa Storrie-Lombardi
(Assistant Director for Ground-Based Astronomy, IPAC, lisa@ipac.caltech.edu)

3 BASELINE SYSTEM AND DEPENDENCIES

Below is a summary of the ZSDS, broken down into subsystems with dependencies in broad terms. Most of these were inherited from the existing iPTF Data System. Note: some of the details are subject to revision and/or refinement according to priorities and performance during the course of development. **Figure 1** depicts the overall data flow. Further details on the core pipelines are given in Section 7.

1. **Camera image and metadata packaging at P48 and transfer to IPAC:** includes data compression strategy, transmission size, and specification of all required image metadata. (*Depends on camera software and data-link/relay network*)

2. **Ingest pipeline**: camera-image splitting, raw-image preparation, collation of telescope and camera performance metrics, simple data QA, accountability and checksum statistics, and archival.

3. **Calibration-derivation pipelines**: all calibration-derivation software across system that generate products for use by other pipelines, either dynamically (near realtime) or on a need-by basis: flat-fielding; bias corrections (including floating-bias using over-scan); optical artifact/ghost masking; bad-pixel masks; other TBD instrumental calibrations in response to detector characterization.
   (*Depends on special instrumental image data, science image data and parameters*)

4. **Instrumental and photometric calibration pipeline:** generates astrometrically and photometrically calibrated epochal images and source catalogs for archival; includes both aperture and PSF-fit photometry. TBD: execute *only* in realtime so can immediately feed products to [5].
   (*Depends on raw image data [2], instrumental calibrations and parameters [3], external catalogs to support astrometric and photometric calibration, processing parameters*)

5. **Realtime (transient discovery) pipeline**: uses the outputs from [4] to perform image subtraction and extraction of transient candidates using an improved version of PTFIDE; includes machine-learned vetting. Also optionally detects streaks (fast moving objects) with metadata and quality scores from machine-learned classification.
   (*Depends on raw image data [2], calibrated image and catalog products from [4], reference images and catalogs [7], ML-training model, processing parameters and thresholds*)

6. **Lightcurve pipeline:** performs source matching of epochal catalogs from [4] with reference image catalogs from [7] to generate initial lightcurves. Then computes relative photometric corrections across all available epochs to refine photometric zero-points. Includes the computation of metrics on a lightcurve basis.
   (*Depends on calibrated epochal images and catalogs from [4], reference image catalogs from [7], processing parameters*)

7. **Reference image pipeline:** involves co-addition of processed epochal images from [4], selected according to certain quality criteria (TBD). Generates reference images and accompanying catalogs for archival: both aperture and PSF-fit photometry. (*Depends on calibrated epochal images from [4], processing parameters and thresholds*)

8. **Moving Object Pipeline System (MOPS):** links pre-filtered and vetted transient candidates from the image-subtraction pipeline [5] to generate moving object tracklets with metadata to support human-vetting. Tracklets are periodically delivered to the MPC. (*Depends on ML-vetted transient candidates from [5] and processing parameters*)

9. **Quality assurance:** collection and storage of metrics across all subsystems for both short-term and long-term monitoring. (*Primarily collected from four pipelines: [2], [3 – flat-fielding only], [4], [5]*)

10. **Configuration management and pipeline executive:** job scheduling and control; (virtual) pipeline operations. (*Depends on configuration parameters and pipeline control/prioritization rules*)

11. **System engineering:** hardware maintenance (fileservers, databases, storage and backups).

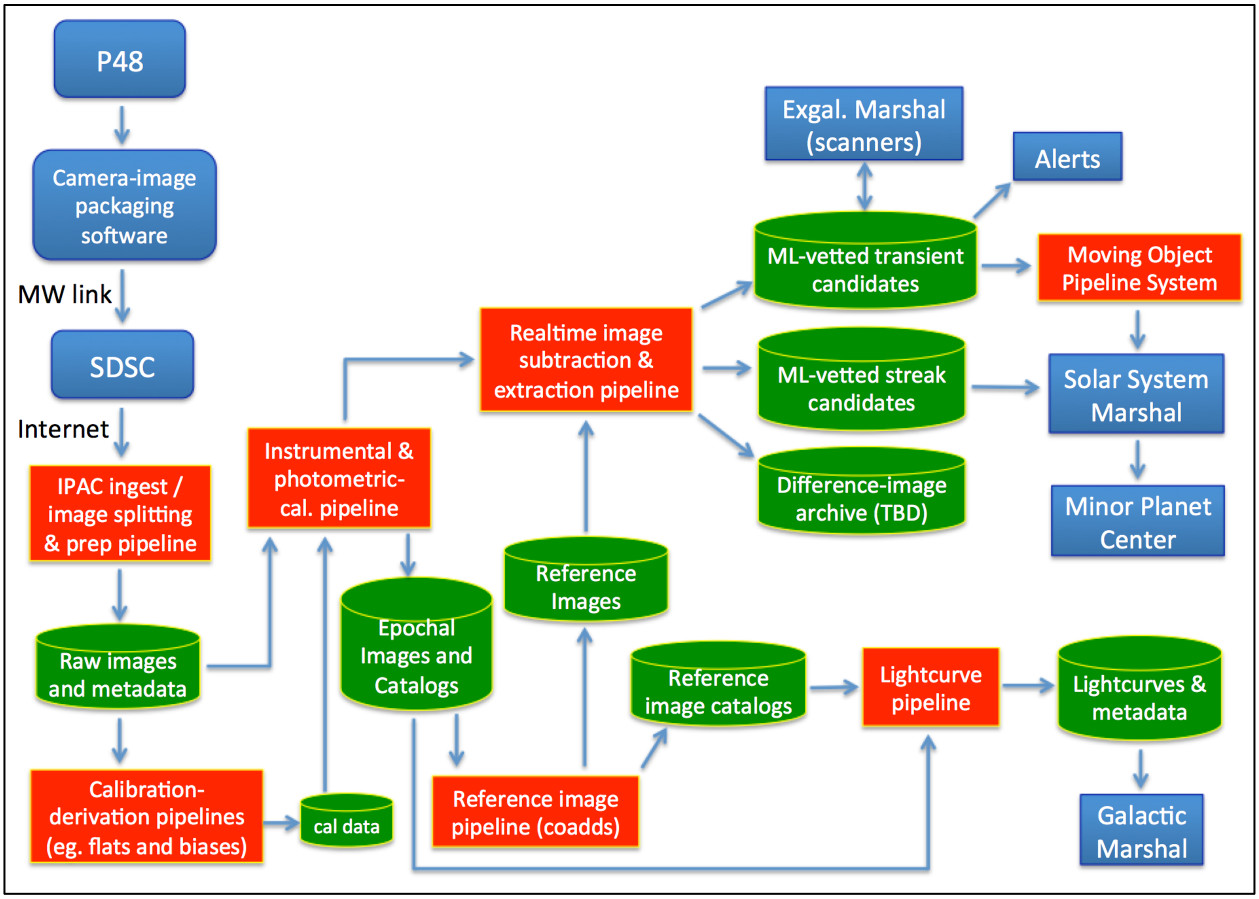

Figure 1: Data flow in the ZTF Science Data System. Green components indicate storage in the form of an archive and/or database residing at IPAC to serve either the public, science marshals and/or anyone in the collaboration. Red components represent the core pipelines. Further details on these core pipelines are given in Section 7.

4 OVERVIEW OF DEVELOPMENT PLAN

The following is a list of the development activities identified at this time. This includes the implementation of new algorithms to be refactored into the existing code base. Some of these will require assistance from the collaboration.

1. **Improve astrometric calibration methodology**: robustness of existing software everywhere on sky, in particular in the galactic plane. If existing software/methodology does not allow, explore alternatives: e.g., improve distortion representation via a *prior* multi-chip solution or a model that separates optical system from *dynamic* atmospheric effects (TBD); consider use of Gaia as astrometric-reference catalog.

2. **Flatfield generation pipeline**: reoptimize existing software or rewrite to process illuminated dome-screen images to generate relative pixel-to-pixel responsivity maps.

3. **Low-spatial frequency (star) flats:** will interplay with (2); purpose is to catch throughput variations in full optical train; methodology and software needed. TBD: may be caught by the zero-point variations map (ZPVM) on a per-image basis from photometric calibration. This ZPVM will need to be interpolated or smoothed before application.

4. **Absolute photometric calibration:** derivation of epochal image *magnitude* zero-points using the Pan-STARRS1 (PS1) data release in the instrumental calibration pipeline. Includes ZPVM generation (see item 3 above). All absolute photometric calibrations will be based on PSF-fit photometry.

5. **Tuning/optimization of source-extraction/photometry software:** primarily *SExtractor* for concentric aperture photometry and *DAOPhot* for PSF-fitting. Ensure detection uses appropriate PSF matched-filter.

6. **Tuning/optimization of realtime pipeline:** includes re-optimizing all pre-calibration and image-preparation steps, and existing image-differencing/extraction software (PTFIDE); consider adding branch to conditionally execute ZOGY algorithm; iterative training of machine-learned (RealBogus) classifier and integration (for both point source transients and streaking asteroids); redesign of *transients* database to enable more efficient loading and scanner-queries; management of potentially new "*N*-day roll-off" scheme in transients database to support near-realtime discovery versus archival analyses.

7. **Tuning/optimization of reference image (co-add) pipeline:** primarily filtering of input epochal images to support requirements of realtime pipeline (6 above); relative image-to-image astrometric refinement needs validating; image-combination method needs re-optimizing.

8. **Lightcurve pipeline redesign:** entails matching PSF-fit extracted sources from epochal images with reference image catalogs, then subsequent epochal zero-point refinement using relative corrections. Any new lightcurve metrics need to be identified. Pieces exist; development effort is related to pipeline configuration and I/O to streamline archive/database loading. TBD: resolve lightcurve duplication issue for same sources falling on different fields and/or chips.

9. **Quality Assurance:** identify, refine or implement new QA metrics for products from the ingest and realtime pipelines; periodically push to staging area for external monitoring.

10. **Summit-to-IPAC data transfer protocols:** define all metadata to be written by camera-image software, including scheduling/program details to enable tagging of data products.

11. **Develop user-access interfaces to pipeline products:** in general, includes tasks to review and rebuild new functionality to support all the services outlined in Section 5: primarily lightcurves (from 8 above) and metadata for transients (from 6) to support realtime scanning marshal. The existing *ptfdepot* access portal may not be optimal.

12. **Design and build infrastructure to support transient-alert streaming:** service planned for *survey start + 12 months* (See Section 6.2 for definition of *survey start*). Identify metadata (e.g., image cutouts, metrics) for streaming. This is tied to realtime pipeline outputs and interface to scanning marshal.

13. **Pipeline executive and configuration management:** job scheduling software and prioritization; failure recovery plans and tractability; virtual pipeline operator setup.

14. **File and database server needs and optimization:** includes optimizing connectivity and throughput across compute cluster. Since a key goal is to lower end-to-end latency, continuous overall system resource utilization monitoring will be necessary (e.g., CPU, memory, networks, disk and database performance).

15. **Data archive management, organization:** includes development of backend query software to support user-interfaces and functionality to manage access control to all deliverables and services.

5 DELIVERABLES AND SERVICES FOR ZTF

The list below builds upon the existing PTF/iPTF functionality and heritage. Using the numbered list below [**1…6**], be aware that:

- Items **1**, **2** were the public deliverables for PTF. Equivalent products from iPTF will be publically released in September 2016.
- Item **3** products are currently being constructed from PTF/iPTF data and will also be publically released in September 2016. **Note:** the lightcurve retrieval interface for ZTF is likely to be a more evolved version.
- Item **4** will be a completely new deliverable (not experimented with for iPTF).
- Items **5** and **6** are currently in production for iPTF.

1. Instrumentally calibrated epochal image data with metadata (includes global photometric zero-points); accompanying source catalogs with calibrated *PSF-fit* photometry and metadata; separate catalogs containing concentric aperture photometry; **note:** all absolute calibration will be tied to PSF-fit photometry; raw image data will also be available.
   ACCESS METHOD: query interface to IRSA database/archive.
   CAPABILITIES: queries based on known object name, spatial location and extent, field/chip identifiers, Solar System Object name and time range.

2. Reference images (co-adds of epochal image data) with ancillary images (depth-of-coverage and uncertainty maps); accompanying source catalogs with calibrated *PSF-fit* photometry and metadata; separate catalogs containing concentric aperture photometry.
   ACCESS METHOD: query interface to IRSA database/archive; same portal as (1).
   CAPABILITIES: queries based on known object name, spatial location and extent, field/chip identifiers.

3. Lightcurves constructed from PSF-fit photometry of matched sources extracted from the epochal images. Photometry is refined using relative photometric corrections.
   ACCESS METHOD: query interface to IRSA archive/database. Will be subject to data-access control protocols (Section 6).
   CAPABILITIES: use cases TBD; at least simple queries based on spatial location and extent; ability to assign upper limits for non-detections across all covered epochs.

4. Transient alert stream from real-time image differencing to commence at *survey start + 12 months*. See Section 6.2 for definition of *survey start.*
   DELIVERY METHOD: TBD; e.g., VOEvents with metadata and image cutouts; includes historical candidates at same positions over last *N*-days. Products will be subject to data-access control protocols (Section 6).

5. External queries to Transients Database on operations system to retrieve transient candidate records, metadata, and image cutouts to support *discovery in near-realtime*.
   ACCESS METHOD: marshal-driven scanning interface(s) to realtime pipeline products; primarily from the *Extragalactic* [including ToO via program ID] and *Galactic* marshals.
   CONSIDERATIONS (TBR):
   - Have a *N*-day rolling window for persistence of *realtime* pipeline products to support near-realtime discovery only. Older data will be periodically pushed to a growing archive to facilitate historical queries and analyses.
   - Goal: ability to check if an event was detected at same position in *any* past epoch. A simple Boolean flag will suffice.
   - Have marshal(s) moderate (or automate) caching of products from "standard" queries to avoid bottlenecks; e.g., resubmit query every *X* minutes and allow scanners to access a central web-server where scanning page can be refreshed.
   - Database optimization possibilities for more efficient queries: don't store candidates below a specific *RealBogus* score and/or signal-to-noise ratio. Will require trade study with size of *N*-day rolling window.

6. Solar System/NEO support: moving-object tracklets from linking transients extracted from realtime image differencing; accompanied by "fast" object detections from single image-streak extractions. Delivered to the IAU's Minor Planet Center following human-vetting according to MPC protocols.
   ACCESS METHOD: marshal-driven query interface; perhaps also interfacing with services offered by the MPC.
   CAPABILITIES: metadata and image cutouts on streaks from realtime pipeline; MOPS-generated tracklets with metadata and image cutouts.

6 DATA ACCESS CONTROL AND PUBLIC RELEASES

As a reminder, ZTF *observing time* will be divided among the following three entities:

- Partners: 40%
- Public surveys funded by the Mid-Scale Innovations Program (MSIP): 40%
- Caltech Time Allocation Committee: 20%

6.1 Clarification on data product access

- The ZTF collaboration will have access to 80% of all data taken (partner and public surveys), the only detail is that products from the public surveys will be released (to the *public*) sooner. See schedule below.
- Partners will have access to all the products and services listed in Section 5, as soon as they are available (80% of ZTF observing time).
- Observers granted Caltech-time will have access to all the products and services listed in Section 5, as soon as they are available (20% of ZTF observing time).

6.2 *Public* release schedule and products

The deliverables below are abbreviated as follows (see Section 5 for details on each):

EP – epochal images and catalogs
RF – reference images and catalogs
LC – lightcurve database
TA – transient alerts

"*survey start*" is defined as the official start of the routine science survey following the commissioning period (TBD).

End of year 1 (*survey start* + 12 months): initial release of **EP, RF, LC,** start of **TA**

Middle of year 2 (*survey start* + 18 months): another release of **EP, RF, LC**
End of year 2 (*survey start* + 24 months): another release of **EP, RF, LC**

Middle of year 3 (*survey start* + 30 months): another release of **EP, RF, LC**
End of year 3 (*survey start* + 36 months): final release of **EP, RF, LC,** end of **TA**

6.3 Overview of data-tagging

In brief, scheduling software will associate each observing program (or P.I./observer) with an identifier and the camera image-generation software will propagate this as metadata to accompany the raw images transferred to IPAC. These identifiers will then propagate into the respective data products during processing. Data products will be tagged accordingly in the ZSDS archive to enable or restrict access through user-interfaces in consultation with an internal user-access control list.

7 PROCESSING FLOW IN CORE PIPELINES

This section expands on some of the high-level processing (red components) shown in **Figure 1**. The details of each processing step are still preliminary, and are presented here in modular form to guide development and discussion. The plan is to recycle and refactor some of the components from the existing PTF codebase. There is also a plan to accept software modules from the collaboration (details to follow).

The pipeline steps below omit some of the operational details, for example, interactions with databases, sandboxes, network, archive, user-interfaces, web-portals for realtime data access, job management and prioritization, and cluster parallelization strategies.

Five key design goals for the overall processing system are to:

- Minimize end-to-end latencies from telescope to archive (or relevant data access portal). Processing runtime will need to be tracked during the course of development and testing.
- Take a minimalist approach: only execute or perform what is absolutely needed to get the job done, according to the requirements of each deliverable.
- Allow modularity and stand-alone execution of pipeline software components with clearly defined I/O interfaces and dependencies.
- Make all steps of the processing chain tractable, for example, the ability to recover from pipeline or hardware failures with an accountability of all data affected or lost.
- Allow reprocessing of any corrupt or missing data products (subsystem dependent).

7.1 Summary of Processing Flow

Below is an overview of the approximate execution order and frequency of the nine pipelines listed in Sections 7.2 – 7.10, including the origin of their input data.

A pipeline can be either:
single-image-based: operates on individual readout-channel images or camera-image files.
ensemble-based: operates on a collection of preprocessed readout-channel images at once.

(1) **Ingest & preparation** (Section 7.2): *realtime and continuous*; raw data from SDSC, either calibration or science-survey image data with metadata; **single-image-based.**

(2) **Super-bias derivation** (Section 7.3): *late afternoon/early evening*; uses outputs from (1); **ensemble-based.**

(3) **High-ν flat or pixel-to-pixel responsivity map** (Section 7.4): *late afternoon/early evening*; uses outputs from (1) and (2); **ensemble-based.**

(4) **Low-ν flat from either dithered-star observations or long-term ZPVM** (Section 7.5): *weekly, monthly or longer* (TBD); uses instrumentally calibrated outputs from (5) and products from a former run of (4); **ensemble-based.**

(5) **Instrumental and photometric calibration** (Section 7.6): *realtime and continuous*; uses outputs from (1), (2), (3), and (4); **single-image-based.**

(6) **Reference image generation/co-addition** (Section 7.7): *weekly, biweekly or longer, contingent on survey plan and data availability*; uses outputs from (5); **ensemble-based.**

(7) **Realtime transient discovery** (Section 7.8): *realtime and continuous*; uses outputs from (5) and (6); **single-image-based.**

(8) **Source-matching & lightcurve generation** (Section 7.9): *weekly, biweekly or longer* (TBD); uses outputs from (5) and (6); **ensemble-based.**

(9) **Moving object pipeline system** (MOPS; Section 7.10): *every three or four hours throughout night and into mid morning* (TBD); uses outputs from (7); **ensemble-based.**

7.2 Ingest and Raw-image preparation Pipeline

Receive Rice-compressed camera-image files (in MEF format?) with one file per CCD + over-scan regions + metadata. Also receive manifest summary files containing sizes & checksums. A cron-job checks for file existence. Need to be able to distinguish between images intended for calibration (e.g., bias frames, flats, special experiments) and science-survey.

↓

Check file for corruption and intactness

↓

Verify checksums

↓

Copy to raw-image archive and store image metadata, including telemetry on observing system/telescope and observer/program information in database

↓

Decompress and execute image-splitting pipeline on either full camera-image or CCD image, depending on packaged input. Outputs are <u>four readout channel image files per CCD</u>, with over-scan regions retained.

↓

Compute simple raw-image QA metrics and store in database

↓

TBD: depending on accuracy of telescope control system [as promised by engineers], compute initial astrometric solutions for each readout channel image using commanded WCS as input, e.g., via *astrometry.net* and/or SCAMP. This facilitates quick-look image overlays. Also verifies if *actual* image pointing is consistent with commanded. Is all this really needed? If commanded pointing turns out to be accurate enough for quicklook/verification purposes, simply redefine the CRPIX1,2 keyword values attached to each readout channel image.

↓

JPEG preview images -- **TBD**: generate on individual readout channel or CCD-image basis

7.3 Super-bias derivation Pipeline (pixel-based bias map)

Inputs are bias frames acquired in the afternoon or at beginning of night, already ingested and split into readout channel images with accompanying over-scan regions (see 7.2). Static bad-pixel mask containing *prior*-tagged bad pixels.

↓

Compute floating-bias per image using over-scan on a per-row basis and subtract

↓

Mask (NaN) bad pixels using CCD-specific or readout-channel specific bad-pixel mask

↓

Compute super-bias image by collapsing *N* preprocessed images using trimmed average per pixel. **TBD:** compute stacked standard-deviation image to enable tagging of more (transient-like) bad pixels (see next step).

↓

TBD: use stacked standard-deviation image to find and tag more bad (noisy) pixels and augment input static mask to propagate along.

↓

Compute simple image metrics on product (details TBD). Should include at least number of inputs used, global median level, and spatial noise.

↓

Super-bias correction image, stack standard-deviation image, new/augmented pixel mask (TBD), and metadata are stored in calibration archive and database for later retrieval

7.4 High-ν flat-field derivation Pipeline (pixel-to-pixel responsivity map)

Inputs are images of flat-field dome screen acquired in the afternoon or at beginning of night, already ingested and split into readout channel images with accompanying overscan regions (see 7.2); also super-bias correction image and accompanying augmented pixel mask from 7.3.

↓

Compute floating-bias per image using over-scan on a per-row basis and subtract

↓

Subtract super-bias correction image output from 7.3

↓

Mask (NaN) bad pixels using readout-channel specific bad-pixel mask

↓

Normalize or regularize image inputs to have either a spatially trimmed-average of unity, or first divide by low-order surface fit to remove lamp pattern (TBD), then renormalize to trimmed-average or median of unity

↓

Compute responsivity (flat-field correction) map by collapsing *N* preprocessed images using trimmed average per pixel. TBD**:** compute stacked standard-deviation image/$\sqrt{N}$ as proxy for flat-field uncertainty. A robust version of the StdDev will be better, e.g., using percentiles.

↓

Compute simple image metrics on product (TBD). Should include at least number of inputs used, and robust second and third order moments on pixel distribution to capture 2D spatial variation.

↓

Flat-field product (with uncertainty image) and metadata are stored in calibration archive and database for later retrieval

7.5 Low-ν responsivity map derivation Pipeline ("dithered-star flats" or ZPVM)

Inputs are instrumentally (and astrometrically) calibrated readout-channel science images from 7.6, processed over a *week*, *month,* or *longer* (TBD according to some S/N criterion for the final product). For the dithering option, one may also need the positional mappings of each readout-channel image on the full focal plane (original camera-image mosaic). Also required will be the precise low-ν flat-field calibration products originally used on these science images using 7.6.

↓

Back out (multiply each science image by) their original low-ν flat-field calibration product. This leaves images that are only bias-subtracted and flattened using their high-ν flat-fields (pixel-to-pixel responsivity maps). These are now ready to derive a **new** low-ν flat-field.

↓

Extract sources and compute photometry using favorite source extractor

↓

Positionally match sources against each other if deriving dithered-star flats, otherwise, match against external calibrator catalog (PS1) if deriving ZPVM-based responsivity map.

↓

Compute photometric residuals; spatially-bin the residuals according to number of matches available (to satisfy some accuracy criterion); smooth or interpolate binned-map to account for bin boundaries; normalize image product to have trimmed-average or median of unity. Details TBD and are left to the owner of this pipeline.

↓

Low-ν flat-field product and metadata are stored in calibration archive and database for later retrieval

7.6 Instrumental & Photometric Calibration Pipeline

Inputs are raw science-survey images acquired throughout night, already ingested and split into readout-channel images with accompanying over-scan regions (see 7.2). Other inputs comprise calibration products from above.

↓

Detect and mask satellite and aircraft tracks; detect and mask saturated pixels; combine with *prior* readout-channel specific bad-pixel mask to initialize overall image mask to propagate and update below

↓

Compute floating-bias per image using over-scan on a per-row basis and subtract

↓

Subtract super-bias correction image output from 7.3

↓

Apply high-ν flat-field (pixel-to-pixel responsivity map) from 7.4

↓

Apply low-ν flat-field (optical-throughput responsivity map from dithered-star flat or ZPVM) from 7.5. **Note:** this does not need to be combined with the high-ν flat to create an effective superflat. In fact, doing this will complicate the low-ν flat generation process in 7.5.

↓

Execute *SExtractor* (or other) for sole purpose of obtaining source positions to support astrometric calibration below. Not all output columns needed. Filter and retain "clean" point sources above a specific S/N that also adequately sample entire image (e.g., within partitions)

↓

Astrometric calibration: includes local distortion calibration to capture effects from optical system and atmosphere; external catalog TBD (Gaia?). It should be sufficient to seed the input WCS using the commanded pointing [promised by engineers].

↓

Verification and QA of astrometric solution using external catalog (TBD); sanity check distortion polynomial using image-binned residuals; can also compute vector map and compare to expectations (priors) for each CCD readout-channel

↓

TBD: detect ghosts, halos, and other optical artifacts (to be characterized) and update masks;

↓

Estimate spatially-varying PSF model (already automated using *DAOPhot*) with QA metrics

↓

Execute PSF-fit photometry (already automated using *DAOPhot*) to detect sources via matched-filter and generate *instrumental* magnitudes with metrics. Source positions therein will have been implicitly astrometrically calibrated.

↓

Photometric calibration using PSF-fit extractions and PS1 to compute global-image *magnitude* zero-point (ZP). Compute QA metrics associated with calibration, e.g., number and magnitude range of PS1 matches, RMS, uncertainty etc.

↓

Apply ZP to instrumental photometry of each source in PSF-fit catalog only

↓

Execute *SExtractor* (or other) to generate concentric aperture photometry catalog with shape metrics. Will contain astrometrically calibrated source positions and *instrumental* aperture photometry

↓

Update headers of (i) image; (ii) PSF-fit photometry catalog; (iii) aperture photometry catalog with global-image ZP, RMS, and uncertainty. **Note:** this ZP will only apply to the biggest aperture in the aperture catalog (i.e., that converges with the total *instrumental* PSF-fit flux). Instrumental photometry in the aperture catalog <u>will not</u> be explicitly calibrated.

↓

Compute simple QA metrics on image and catalog products

↓

Store calibrated science image, accompanying mask, two catalog files (aperture and PSF-fitting), and QA metrics in archive and database. These products will also feed directly into the transient discovery pipeline (7.8).

7.7 Reference Image (co-addition) Pipeline

Inputs are astrometrically and photometrically calibrated epochal images for a given filter from pipeline 7.6. These images are associated with a specific readout channel of a CCD and are partitioned on the sky according to a pre-defined sky grid (survey fields).

↓

Select best quality images to co-add in terms of seeing, astrometric & photometric quality

↓

Mask bad pixels in each image using the accompanying image masks

↓

Gain-match pixel values in each image to a fixed common zeropoint value using their photometric ZPs (already assigned in pipeline 7.6).

↓

Resample and undistort each image using its astrometric solution onto a common co-add footprint with boundaries defined by the sky grid. The absolute astrometric solution of each input image should be sufficient for registration purposes. Relative astrometric refinement of the input image solutions prior to reprojection is TBD (FM votes against this).

↓

Combine resampled images (or all good/unmasked pixels in stack) using a trimmed average to create co-add product. TBD: inverse-variance weighting using prior background-noise estimates.

↓

Compute ancillary co-add products: uncertainty and depth-of-coverage maps

↓

Generate PSF-fit photometry catalog (already automated using *DAOPhot*); note: this catalog will automatically be photometrically calibrated to a fixed ZP following the image gain-matching step above. All sources therein will have the new ZP applied to their instrumental photometry.

↓

Generate *SExtractor* catalog containing concentric aperture photometry; do not apply ZP to photometry therein, only write global ZP to its header. As a reminder, this ZP would have been derived using only PSF-fit photometry in pipeline 7.6. Therefore, this ZP will only apply to the biggest aperture in the aperture catalog (i.e., that converges with the total *instrumental* PSF-fit flux). Instrumental photometry in the co-add aperture catalog will not be explicitly calibrated.

↓

Compute simple QA metrics on products: calibrated co-add image and catalog contents

↓

Store co-add products (primary image, depth-of-coverage map, uncertainty image, PSF-file), two catalog files, and QA metrics in archive and database

7.8 Realtime Transient Discovery Pipeline

Astrometrically and photometrically calibrated images from 7.6. In fact, the transient-discovery pipeline will be a continuation of the instrumental and photometric calibration pipeline in 7.6, which will also run in realtime. We also require reference-image products from 7.7.

↓

Next 10 steps will be performed by a streamlined version of PTFIDE; where pertinent, the ZOGY image-subtraction algorithm will also be explored

Relative photometric-gain matching between science and reference image pixels

↓

TBD: relative astrometric refinement of science image with respect to reference

↓

Reprojection/resampling of reference image onto "distorted" frame of science image

↓

Match smoothly varying (spatially-dependent) backgrounds in science and reference images

↓

Estimate spatially varying PSF in science and reference images for use in deriving PSF-matching convolution kernel

↓

Derive PSF-matching convolution kernel with QA metrics

↓

Apply PSF-matching kernel to match PSFs in science and reference image

↓

Image differencing (both *positive* and *negative* difference images); also generate mask, uncertainty products, image metrics and features to support machine-learned vetting below

↓

Estimate spatially varying PSF from *best* image following PSF-matching; extract transient candidates with PSF-fit and aperture photometry; also compute source metrics and features to support machine-learned vetting below

↓

Apply simple/loose filtering on candidate list to remove *obvious* false positives (e.g., cosmic rays, bright-source residuals)

↓

Execute machined-learned classifier (*RealBogus*) on transient/variable candidates extracted from *positive* and *negative* difference images

↓

Execute streak detector to find fast moving objects (NEO candidates) in *positive* diff. image; includes generation of metadata, metrics, and features for detected streaks to support machine-learned vetting below

↓

Execute machined-learned classifier on detected streaking NEO candidates

↓

Predict and record positions of known asteroids in difference image footprint using ephemerides of known objects from MPC-archives. A static copy of these ephemerides will reside in a local database for fast retrieval and will be periodically updated.

↓

- Archive *positive* difference image, uncertainty, log file, and PSF (all compressed);
- Load following into the Transients Database: difference image metadata and QA metrics; all transient/variable candidates extracted from *positive* and *negative* difference images above some *RealBogus* and possibly S/N cut (TBD). These transient-source records will include positions, photometry, features, and *RealBogus* scores;
- Store metadata and metrics for detected streaks in database
- Match transient candidate records in database with known asteroids, star and galaxy catalogs, and other contextual information, and update transient-record flags.

↓

Copy select products to public web-server to support retrieval of image cutouts on queried transient and streak candidates in near realtime to support discovery.

↓

Roll-off products from Transients Database older than some TBD date to long-term storage.

7.9 Source-matching and Lightcurve-generation Pipeline

Inputs are astrometrically and photometrically calibrated epochal PSF-fit catalogs for a given filter from pipeline 7.6. These are associated with a specific readout channel of a CCD and are partitioned on the sky according to a pre-defined sky grid (survey fields). We also require the reference-image PSF-fit photometry catalog from 7.7.

↓

Match source positions from the reference-image catalog with those from each epochal catalog. Each source position in the reference image catalog (with photometry brighter than some magnitude limit) therefore gets labeled as a new lightcurve. **TBD:** should we also match sources across the epochal catalogs regardless of detection in the reference image (i.e., transient-like sources)? All source matches are stored in a table file.

↓

For each epochal catalog, isolate non-variable (or the least-variant) sources across all available epochs to use as anchors for the relative photometry step below

↓

Compute corrections to the epochal photometric ZPs (delta ZP per epoch) using a global minimization method based on the relative photometry of anchors selected above. Note: matrices will be of the sparse type and it has been suggested that the conjugate gradient method will work best. These ZP corrections are stored in a table file.

↓

Generate three database-load files containing (i) object (lightcurve) identifiers with metadata and metrics; (ii) table of association indices between each object ID and source IDs from epochal catalogs; (iii) epochs-based table storing photometric ZP corrections and flux upper-limit information (one per epoch). These flux upper-limits in particular will be needed for assigning lightcurve measurements to epochs where no detections from the epochal photometry exist.

↓

Periodically load contents of above files into lightcurve database to facilitate queries through user-interface

7.10 Moving Object Pipeline System (MOPS)

Machine-learned vetted transient candidates from 7.8, read from Transients Database

↓

Further optional filtering of single-epoch candidates to avoid bright superposed sources from reference image catalogs

↓

Generate moving object-tracklet candidates by linking transients using velocity matching across pairs of epochs, then merging of those pairs to create object-track candidates. The MODE software will be used (Moving Object Discovery Engine)

↓

Compute QA metrics and features for moving object candidates

↓

Filter moving object candidates according to available metrics or use a pre-trained machine-learned classifier to support vetting process downstream (TBD)

↓

Human vetting of filtered moving object candidates from above with generation of report for Minor Planet Center. This report will contain matches to known (archival) objects as well as potentially new objects for dissemination by MPC

↓

Submit report to MPC; store provisionally designated (potentially new) objects assigned by MPC in a local database (TBD)

8 ACRONYMS

CCD	Charge Coupled Device
CPU	Central Processing Unit
DAOPhot	Dominion Astrophysical Observatory Photometry tool
DB	Database
FITS	Flexible Image Transport System
I/O	Input / Output
IAU	International Astronomical Union
IPAC	Infrared Processing and Analysis Center
iPTF	*intermediate* Palomar Transient Facility
IRSA	InfraRed Science Archive
MEF	Multi-Extension FITS file format
ML	Machine Learning (or Machine-Learned)
MW	Micro-Wave data link
MODE	Moving Object Discovery Engine
MOPS	Moving Object Pipeline System
MPC	Minor Planet Center
MSIP	Mid-Scale Innovations Program
NEO	Near-Earth Object
P48	Palomar 48 inch

PS1	Pan-STARRS 1 Data Release products
PSF	Point Spread Function
PTF	Palomar Transient Facility
PTFIDE	PTF Image Differencing & Extraction pipeline
QA	Quality Assurance
RB	RealBogus (moniker for machine-learned vetting)
RMS	Root-Mean-Square deviation
S/N	Signal-to-Noise ratio
SDSC	San Diego Supercomputing Center
SExtractor	Source Extraction tool
StdDev	Standard Deviation
TBD	To Be Determined
TBR	To Be Resolved
ToO	Target of Opportunity (e.g., in context of a LIGO event or Fermi burst)
VO	Virtual Observatory
WCS	World Coordinate System
ZOGY	Zackay, Ofek, & Gal-Yam image-subtraction algorithm
ZP	Zero-Point (for photometric calibration in terms of magnitudes)
ZPVM	Zero-Point Variations Map generated from absolute calibration
ZSDS	ZTF Science Data System
ZTF	Zwicky Transient Facility
ZTFIDE	ZTF Image Differencing & Extraction pipeline