UNIVERSITY OF HAWAII AT MĀNOA
Institute for Astrononmy

Pan-STARRS Project Management System

DVO Software Design Description

The Desktop Virtual Observatory: Astronomical Object Databasing in the IPP

Grant Award No.	**: F29601-02-1-0268**
Prepared For	**: IPP**
Prepared By	**: Eugene Magnier**

Document No.	**: PSDC-430-016**
Document Date	**: January 22, 2006**
Revision	**: DR**

DISTRIBUTION STATEMENT
Approved for Public Release – Distribution is Unlimited

Submitted By:

____ ____________
[Insert Signature Block of Authorized Developer Representative] Date

Approved By:

____ ____________
[Insert Signature Block of Customer Developer Representative] Date

Contents

1 Overview

DVO, the Desktop Virtual Observatory, is a software system which stores data related to astronomical objects derived from various sources, and provides mechanisms to related multiple detections together as astronomical objects. DVO deals with two related concepts: *objects* and *detections*. The *objects* are descriptions of astronomical objects while the *detections* are the specific measurements of those objects, typically measured from astronomical images. A collection of *detections* may be used to derive average quantities which describe a particular *object*. A third class of measurement to be considered are those supplied by external references. Such measurements may be treated as *detections*, with the caveat that access to the raw measurements and metadata are usually unavailable: the reported measurements and errors must be accepted as they are reported.

DVO stores the collections of detections which were derived from specific images. It provides a mechanism to determine the image from which a specific detection was derived, and in conjunction with the Image Server locate the corresponding data file. DVO also makes it possible to extract all detections derived from a specific image and to determine quantities such as the pixel coordinates of the detection on the image.

DVO also has the capability to associate multiple detections of a specific object. Several major classes of objects will be present, each of which must be handled correctly. DVO distinguished the following types of objects.

Stars, compact galaxies, and QSOs will have nearly fixed locations relative to other distant stars, with only small deviations for individual measurements. The association between multiple detections of such objects is made on the basis of their coincident positions. DVO determines the average position of the object and the deviations of the individual detections from that average on the basis of the ensemble of individual detection.

Solar System Objects do not have a fixed location. Detections of such objects are linked by their orbits, and depend on both the position and the time of the image. DVO does not attempt to make this link; this is the role of the MOPS system. However, it has the ability to accept identifications made externally with specified detections and to return the identifier of the moving object associated with the specific detections. These associations also include descriptive information such as the offset of the detection from the predicted location of the detection based on the orbit. This functionality is required to allow DVO to ignore known moving object detections from other types of queries.

High-proper-motion objects in the general vicinity of the solar system fall in between these first two classes of objects. Their proper motion and parallax response is significant enough (> 0.2 arcsec in 1 year) that they are not well-described by an average location and a collection of offsets. These objects are better described by a distance and a proper motion vector. DVO provides the association between the specific detections and an average object which includes finite parallax and proper motion.

Orphaned detections are not associated with a specific astronomical object of any of the above classes. Most of these will be spurious (not representing real objects), some will be from solar system objects for which orbits are not yet determined, some will be from faint stars near the detection limits, and some will be from short-term transients which have only been detected once. DVO maintains these detections until they have been associated with one of the objects above. DVO provides mechanisms by which individual detections may be migrated back and forth between the orphan state and association with an astronomical object.

DVO stores the information about the detection, the related objects, and the images which provided the measurements. For every detection, DVO provides the mechanisms to link the detection back to the image which supplied it. DVO also provides the capability to determine the images containing a specific location but for which no detection was made. The minimum set of information which must be carried for these non-detections is the image and the associated object or orphan.

DVO also stores the relationships between various photometric systems and the evolution of that relationship. It provides mechanisms to convert between the measured instrumental magnitude of a detection with a specific filter, detector, and telescope, and at a particular time and the implied magnitude in the average Pan-STARRS photometry system, given a determined set of calibrations. It also provides the capability to convert magnitudes in one system to the magnitudes in another system; an example of such a conversion is between the average Pan-STARRS filter systems and the various reference systems appropriate for those filters.

2 Photometric systems and the DVO Photcodes

One of the major roles of DVO is to relate different photometric measurements made with different instruments and detectors together. We may have observations made with the same basic filters, but using a number of different detectors. We may have observations from different telescopes in similar filters. We may have reference data related to some filter, but obtained and published by other observers. We would like to related these measurements together in optimal ways, making use of whatever information we have available. DVO provides several mechanisms to enable these relationships.

We identify three distinct types of photometry measurements within DVO:

- **reference photometry** These measurements are provided by external observers. For reference photometry, we do not have access to very must information used to determine the magnitudes of the objects of interest. We have the reference magnitudes corresponding to a type of filter, and presumably some information of the error on the measurement. We might possibly know the epoch of the observations, but not necessarily.
- **detection photometry** This is our primary measurement of interest: the photometry of objects measured from images which we have processed. More specifically, the detection photometry is an instantaneous measurement from a specific image with well-known properties, such as exposure time, airmass, instrument source, etc.
- **internal photometry** With the application of an appropriate zero point and other calibration terms, any detection photometry can be calibrated to represent a measurement in a well-known photometric system. The internal photometry measurements are calibrated to be on a photometric system which represents a consistent system for a particular telescope or collection of data, minimizing the calibration transformations necsessary.

Defining the relationships between the different types of measurements is part of the process of photometric calibration. DVO uses the concept of the 'photcode' to identify the source of the photometry, and to define the relationships between different photometry sources. A photcode identifies a photometric system: for the detection photometry measurments, each combination of telescope, camera, filter, and detector is associated with a unique photcode; there are also unique photcodes for the internal photometry systems and any distinct external reference source.

As a concrete example, consider the Pan-STARRS PS-1 system. There will be three different cameras in use at different times: GPC-1, TC-3, and the SkyProbe camera. There are at least 6 filter systems: *grizy* and *w*. The SkyProbe camera has a single CCD, TC-3 has 16 different detectors, and GPC-1 has up to 64 different devices. Each of these combinations is potentially a different photometric system, so a different photcode is defined for each combination. These photcodes would have names such as: GPC1.02.r (r filter with the GPC1 camera and OTA 02) or SP1.00.g (SkyProbe 1, g filter). These ($64 \times 6 + 16 \times 6 + 5 = 485$) photcodes are all identified as 'detection' photcodes, specifying that detection photometry is associated with them

There are also 6 different internal photometric systems of interest, namely those associated with the 6 named filters, *grizy* and *w*. Each of these 6 systems is identified with an internal photcode. The internal photcodes are further distinguished as 'primary' or 'secondary', which specifies how the DVO system stores average quantities related to these types of photcodes (see the discussion of the tables below).

Finally, there may be multiple external photometric systems of interest, some of which are related to the major internal photometry systems, some of which are not. For example, the Pan-STARRS project may refer to photometry from the SDSS secondary standards, the SDSS data releases, Johnson photometry from Landolt (1992), observations from 2MASS in JHK, USNO-B observations, and so forth. Each of these photometric systems is assoiciated with a different photcode; only some of these are relevant to the detection or internal photometry system.

Within DVO, the detection and internal photcodes each define a relationships as well as a specific photometric system. Associated with each of these photcodes are the parameters of the photometry transformation from the photometric system of the photcode to another photometric system. For the detection photcodes, the parameters define the transformation to the equivalent internal photcode system. The currently-defined transformation parameters consist of the following photometry equation:

$$M_i = M_r + C_r + K_r(\text{airmass} - 1) + \sum_{i=1}^{i<N} A_{r,i}(\text{color} - \text{color}_r)^i$$

where C_r represents the zero-point of the transformation, K_r represents the slope of the airmass trend, airmass is the airmass for a given measurement, color is the color of the source of interest (as identified below), color_r is the reference color for sources in this photometry system, and $A_{r,i}$ is the coefficient of the i power of the color difference. Up to fourth order color terms are currently allowed. For any photcode, the color is defined as the difference of the measurements in two other photcodes, usually two 'internal' photcodes. The photcode information also specified the equivalent photcode to which the transformation corresponds.

For the detection photcodes, the target of the transformation must be an internal photcode. For the internal photcodes, the target of the transformation is an external reference photcode system. This restriction implies that the internal photometry may only be transformed (and thus compared with) a single external reference. This is in fact the best practice as far as photometric calibration is concerned: the 'standard' observations from different references should always be treated as different photometric systems. To allow for the relationship of the internal photometry to multiple sources of reference photometry, an additional set of photcodes are defined which identify 'alternative' transformations for the internal photcodes.

It is important to note that not all of the photometry transformation parameters identified above are relevant for each of the three major types of photcode. The detection photcodes will in general make use of all of these elements, though the order of the color transformation will hopefully be limited if the different devices are sufficiently similar. For the transformation from the internal photcodes, which are derivative in some way of the detection photcodes, the airmass component is invalid: for a single measurement, the detection-to-internal transformation has already removed the airmass trend; for an averaged internal photometric measurement, no single airmass corresponds to the observations. Finally, no transformation parameters are defined for the reference photcodes at this time.

DVO provides methods by which these photometry transforamtions are automatically applied. The specific measurements (detection photometry) are stored in the database tables as instrumental magnitudes, and any operation which examines these measurements must make use of the APIs to convert to an appropriate common system. A further complication to note is that the photcodes defined above are static; they do not include any information about changes to the system sensitivity. This information is carried externally to the photcode calibration information; the transformations defined by the photcodes must be considered the *starting point* for any photometric analysis. An additional adjusment can be applied.

The detections from a specific image may all have a 'calibration' offset applied which bring the measured photometry into a common relative system. This calibration offset is associated with the image and may be a function of position on the detector. The tables which carry the individual measurements also include the calibration magnitude appropriate for each measurement to speed up the application of this offset. In a well-calibrated collection of photometry, all of the detection measurements will have a measured calibration magnitude, yielding a collection of internal photometry measurements which are all consistent. An additional piece of information is the zero-point history, which tracks the system-wide variations in the average sensitivity. The zero-point history can be used to predict the calibration magnitudes for any observation which is not tied directly via relative photometry to the rest of the photometric observations.

Putting all of these pieces together, the photometry APIs in DVO can be used to return any of the following types of photometric measurements:

- raw instrumental magnitudes for any detection
- ’catalog’ magnitudes, applying only the airmass and static zero-point calibrations to a detection magnitude; this is useful to test the detector-color transformation.
- ’system’ measurements, applying the complete static transformation for a detection magnitude to the internal photometry system; for photometric weather and no zero-point variations, this would be a measurement in the internal photometry system.
- ’relative’ magnitudes, applying the measured calibration offset to the calibrated detection magnitude determined above; in a well-calibrated system, this represents a consistent internal photometry measurement.
- ’calibrated’ magnitudes, correcting the measure detection photometry by applying the transformation from the internal magnitude system to the external reference magntiude system.
- ’average’ magntiudes, the raw internal photometry magnitudes (note the distinction between the ’average’ quantities, which are derived from a collection of detections an the ’relative’ quantities which represent an instantenous measurement in the same system).
- ’reference’ magnitudes, in which the ’average’ internal photometry values are transformed to the refernce magnitude system.

The complexity of these transformations is necessary to allow the examination of the trends of actual measurements with external parameters.

3 DVO Database Tables

Figure 1 illustrates the data managed by DVO, and Table 1 provides a complete listing. The contents of these tables are outlined in Appendix A. Below, the use of these tables by DVO software is discussed below. Several of the tables are not just simple tables in the database but are instead table groups divided into many subtables, each of which represents a portion of the sky (a `region`). These subtables may also be distributed across different computers to distribute the processing load.

3.1 Sky Regions Table

The `Regions` table is used to subdivide the tables of images, objects, and detections, etc, as discussed above. DVO divides the sky into a hierarchy of regions (portions of the sky) each of which is in turn subdivided into smaller portions. Since nearly all interactions with DVO performed by the IPP are limited in spatial coverage, subdividing the tables allows a specific interaction to search only a small subset of the data. The table of images is the smallest of the three; the table of detections is likely to be the largest. As a result, the `Images` table group will be subdivided at a shallow hierarchical level, while the `Objects` and `Detections` are subdivided on deeper (more finely sampled) levels. The `Regions` table defines the boundaries of the sky regions and specifies if the region corresponds to an `Images` table, an `Objects`

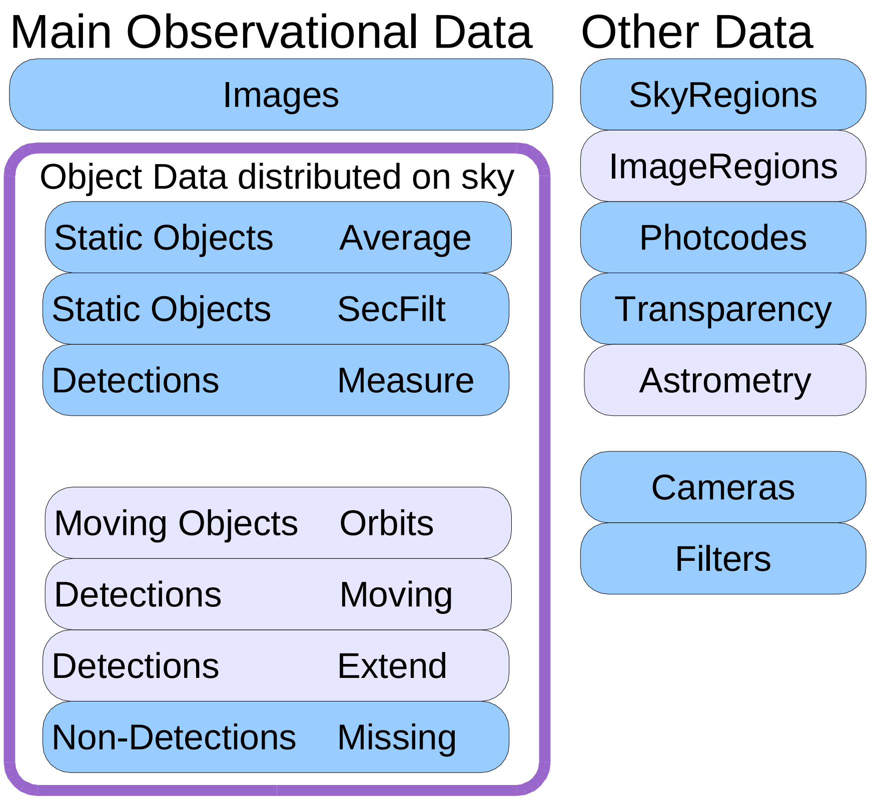

Figure 1: Data types managed by DVO

table, and/or a `Detections` table. It also specifies which regions in the next level of the hierarchy are contained by the region, and which parent region it belongs to. In addition to improving the spatial access to the image, object, and detection data, the `Regions` table allows for multiple computers to serve the database tables. The region file specifies the machine which stores the specific table. Figure **??** illustrates schematically the subdivision of the sky and the association between different levels of the hierarchy with different subtables.

3.2 Images Table Group

The `Images` table group lists all of the images which provided the data in DVO. These tables are subdivided by region on the sky. In general, the images listed in this table correspond to the Chips. This group of tables includes sufficient astrometric parameters to represent the coordinates of the detections to a sufficient accuracy. Parallel to the Images table is the Mosaic table. This table is very similar to the Images table, but defines the Mosaic which corresponds to a group of Images. The parameters include the astrometric information needed to define the camera distortion.

3.3 Image Overlaps Table Group

The specific subtable of `Images` which contains a given image is the one which contains the center pixel of that image. An additional table group, `Image Overlaps` (with the same subtable organization as the `Images` subtables), lists images which overlap that specific subtable. Thus, given a particular coordinate, in order to find that images which overlap that coordinate, it is necessary to search the images in the `Images` subtable which includes that coordinate, and all images in the `ImageOverlaps` subtable for that coordinate.

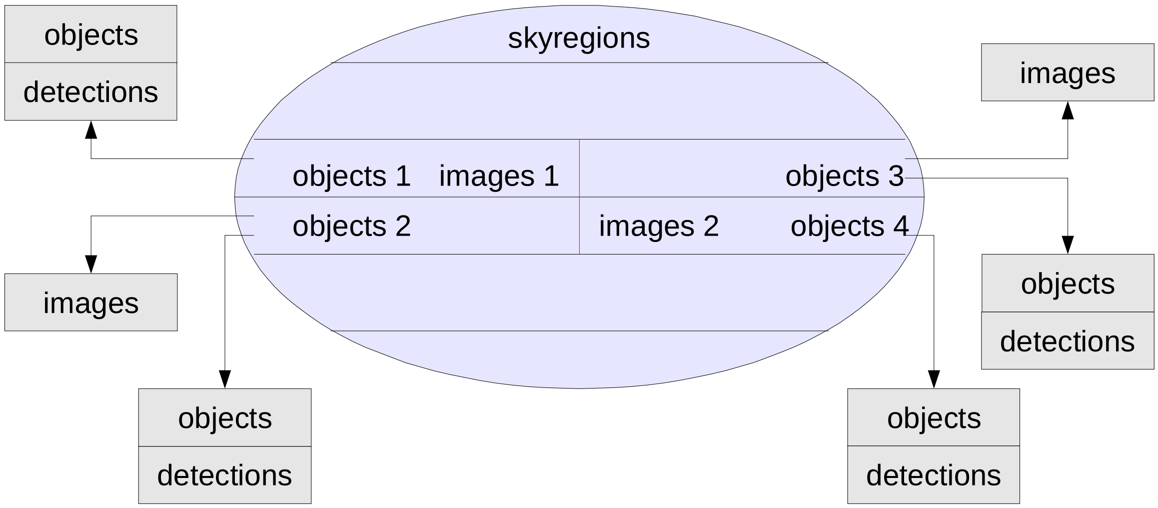

Figure 2: DVO Regions and Image / Object tables

3.4 Objects Table Group

The `Objects` table group (also divided by region) stores the average parameters for each astronomical object. Certain details of this table have not yet been specified. In particular, objects with significant parallax and/or proper motion may potentially be stored in a distinct table. Solar system object identifications, to the extent average properties are maintained in DVO, will certainly be stored in a separate table.

3.5 Average Magnitudes Table Group

A related table, also divided into the same regions, is the `Average Magnitudes` table. In this table, there are multiple rows per object, one for each of the primary filters of interest for which photometric averaging is performed. This organization makes the number of primary (averaged) filters a configurable value.

3.6 Matched Detections Table Group

The `Matched Detections` table stores all of the measurements of astronomical objects on specific images. This table includes all detections associated with the average `Objects`. As discussed below, bright objects (above a configuration-specified signal-to-noise level) are defined object even if only one detection has been found at that position. Faint orphaned objects are not added to this list or the list of objects. The different types of detections (P2, P4Δ, P4Σ) are distinguished by their photometry codes. (This is only valid if DVO does not store different quantities for these types of detections.)

3.7 Orphaned Detections Table Group

The `Orphaned Detections` table stores the detections which have not been correlated with an existing object. This table is only populated for objects below a configuration-specified signal-to-noise limit (e.g., 5σ). Bright orphaned detections are assigned an object and added to the `Matched Detections` table.

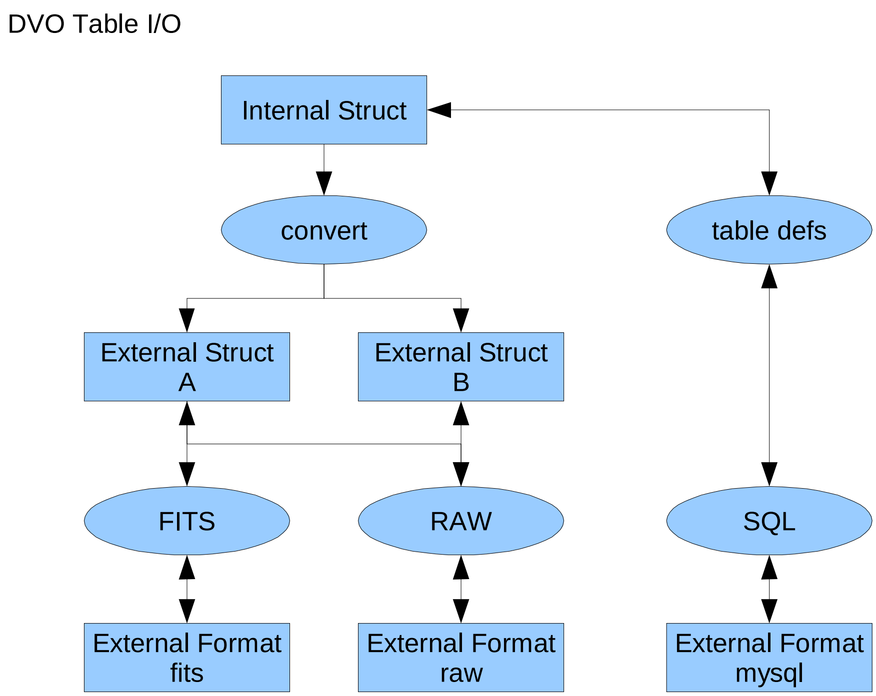

Figure 3: DVO Table I/O

3.8 Non-detections Table Group

The `Non-detections` table stores information about detection failures for each object. If an image is added to the database which overlaps an object but the object is not detected, an entry is made in this table. In practice, this table may store only the most recent non-detection and the total number, or a similar reduced set of non-detection statistics.

3.9 Other Reference Tables

The `Filters` table identifies all of the physical filters (specific pieces of glass) known to the system. A related table, `Photcodes`, defines relationships between photometry systems. A photometry system may consist of a detector, telescope, and specific filter, or it may be a derived photometry system. The `Database Machines` table identifies all of the computers available to DVO.

4 Database Table I/O

DVO allows for a flexible representation of its data on disk. Data may be written to disk in one four possible mode: RAW, FITS MEF, FITS SPLIT, and MYSQL. These modes define the overall organization of the data on disk. In the RAW mode, the data is written to disk in a pseudo-FITS table format which consists of a simple FITS header describing the layout followed by the binary data in a block. This storage mode is maintained for historical reasons. There are also two types of FITS modes in which the data tables are written as valid FITS Binary Tables. In the SPLIT format, every data table is written as a separate file, while in the MEF format, the object and detection tables are bundled together into a single FITS file with multiple table extensions. The MEF format has the advantage of minimize the proliferation of files, while

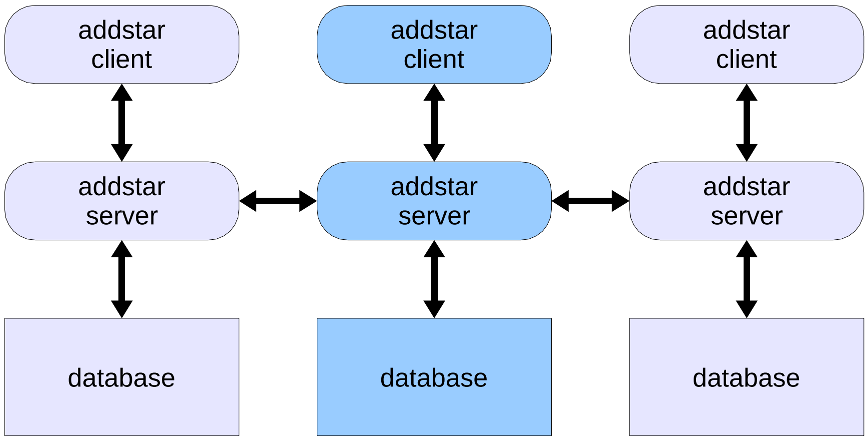

Figure 4: a figure

the SPLIT format is required to make use of the fastest read/write capabilities of DVO. DVO makes use of these raw data formats as a throughput risk mitigation strategy. As discussed below, this strategy has proven very successful.

There are also multiple formats in which the data may be stored. The different formats define which specific database table columns are stored and with what numerical format and precision. Figure **??** illustrates the conversion process which DVO performs when loading in the data. When DVO loads data from a file-based table (FITS or RAW), it first loads from the disk file into a data structure representing the external format in use. The external structure is then converted into the internal format. The internal structure is always specified to be the superset of all external data formats. This capability allows DVO to maintain backwards compatibility with data tables written with early versions. As DVO is extended and new elements are added to the tables, it is only necessary to define the methods to convert the new internal table into the external table. In addition, DVO makes use of autocoded table manipulation and I/O APIs which are generated for each data structure based on a descriptive table. This makes it easy to add new data types and input/output methods without significant re-coding.

DVO mysql table storage is not yet implemented (TBD)

5 addstar : Insert Image & Detection Set

One of the most basic operations needed by DVO is to insert a collection of detections derived from a specific image, and add the definition of that image to the database. This operation is critical in terms of the processing throughput. After the detections have been assigned to the appropriate regions, they are matched against all objects in the `Objects` table. Matches are performed only on the basis of positional coincidence, using a matching radius which may depend on the image astrometry errors, or may be a fixed distance. Any matched detections are added to the `Matched Detections` table. Any unmatched detections brighter than the Faint Detection cut-off are specified as a new `Object` and also added to the `Matched Detections` table. Any faint unmatched detections are added to the `Orphaned Detections` table. This division is important because it allows the automatic association of new detections with existing bright objects while limiting the I/O volume required to make the detections. In general, there will be many fewer `Objects` than `Detections`, and there will be fewer bright orphans than faint orphans.

A wide range of options are available to addstar. These can be used to modify the object matching rules, to reduce the number of tables which are updated, to specify the output data format, and so forth. A few options modify the behavoir in substantial ways, as discussed in the two sections below.

flesh out discussion of the options (TBD)

5.1 Insert Reference Objects

`addstar -ref (filename)`

This mode of addstar reads a text file and adds the listed objects to the database as a reference photcode type. A collection of reference objects are added to the database as a collection of detections. The reference photometry should in general be given its own photometry code. The reference data is different from the image detection set because the associated image information is not included. Thus, no corresponding images are added to the database.

5.2 Insert Catalog Objects

`addstar -cat (name) -region ra ra dec dec`

In this mode, any of several all-sky or large-scale reference catalogs are used for the input sources. The catalog objects are added to the database as reference objects. The valid catalogs consist of 2MASS, USNO, GSC. Tycho and USNO-B will be added shortly. Specific photcode names are defined for each of these catalogs, and must be appropriately requested and defined in the photcode table. The optional region restriction limits the insert to a subset of the sky. The user does not always want to add 50GB of 2MASS detections to any DVO database...

5.3 Addstar Client/Server Interactions

DVO currently uses stand-alone programs which are run from the command line (like addstar, or the programs listed below), or it works with the interactive DVO shell, which allows the user to query portions of the database. These programs all interact with the database tables directly, making use of file locking to prevent conflicts.

Unlike the other DVO programs (currently), it is possible to run addstar as a client/server system. In this configuration, the program `addstard` is launched to run in the background as a server. It monitors a socket waiting for clients to contact it. The client program, `addstarc` appears to the user identical to the stand-alone addstar. However, rather than directly insert data into the database, `addstarc` contacts the addstar server and sends it the detections and associated image data (along with the information about the user options). The daemons accepts the incoming data and then loads this data into the database, just as the stand-alone addstar does.

The purpose of the addstar client/server design is three-fold. First, the client can be used by processes to send data to the DVO database and then immediately exit. The addstar loading process is one of the more time-critical functions within the IPP. However, unlike the other portions of the IPP, the addstar processes must operate in serial, at least when they are updating the same portion of the sky (or the image table). If the IPP analysis routines all needed to run the stand-alone addstar program, they would eventually block waiting for each addstar to complete, preventing other processing from continuing. The addstar client / server model allows the processing node to invoke the addstar client, sending the data to the addstar server. The addstar server will then be the entity that manages the serialization of the incoming data stream. The addstar server has two threads which run in parallel. One thread monitors the socket and accepts new data sets from addstar clients, adding the data to an internal queue. The other thread pulls data off of the queue and updates the database with the data.

A second advantage of the client/server interaction is that only the new detections need to be sent across the network. To update the database, addstar must load the average objects for the region from the database tables. In the stand-alone mode, the addstar program loads this data via NFS across the network from whatever device stores the addstar tables. In the client/server model, the addstar server always runs locally on the machine which holds the database tables. Thus, for the server, all database access is local disk access.

The final advantage of the client/server model is that it enables the parallel database model, which is not yet implemented as of Jan 2006. In this model, there are multiple addstar servers. Each one has a fraction of the sky in the local tables. The identification of which table is managed by this host/addstar server is stored in the SkyRegion table. The addstar server simply accepts incoming detections from the addstar clients. Any detections which it receives which fall within the boundaries of tables that it manages are updated as normal. The server then identifies the other addstar servers which are responsible for the other detections. It then sends these detections to those servers using the same socket communication used by the addstar clients. The addstar server must also be ready accept detections from other addstar servers. This relationship is completely parallel, and any addstar client may send its data to any addstar server, letting the servers hash out who owns what. The only difficulty with this model is in handling sources near the boundaries of the tables. Note that this issues exists whether those tables are distributed across multiple machines or not.

Addstar uses the following strategy to handle detections on the table boundaries. Detections are first added to each table completely ignoring the neighboring tables. A detection which is close to the boundary may either be associated with an average object contained within the table, or not. If it is, the detection is associated with that average object. If not, a new average object is created at the location of the detection. So far, this process is identical to the behavoir in the middle of the table. One a longer time-scale, a process is run which mediates the table boundaries. In this analysis, the two neighboring tables are simultaneously examined. The border region, in a strip wider than the correlation radius, is examined in detail. If two objects within the border region fall within 2x the correlation radius of each other, their individual detections are re-examined. These detections are re-added to a temporary table which encompases the overlap. the resulting objects will in general have detections from either side of the boundary. The average objects are kept within the table as normal, but the detections are allowed to migrate between the tables to stay with their object. **this boundary cleanup process is not implemented to date (TBD)** .

6 Relphot : Relative Photometry Analysis

This operation uses the overlaps of images and multiple observations of the same objects to determine the relative photometry zero-points for a collection of images. This is a task that is run much more infrequently than the object insertion tasks.

The relphot analysis is currently performed with a single Sky region as the starting point. All images (or all chips from all mosaic iamges) which overlap the sky region are identified in the image table. This set of images are considered set A. Next, all skyregions which are overlapped by all of these images are selected. Finally, all additional images which overlapped the new regions only are selected. These are considered as image set B. The image selections are also restricted to images of a single, user-selected photcode.

All of the objects and detections which are contributed by the images in sample A are extracted from the average and measure tables. Only a subset of the detections for which the S/N is greater than a user-selected limit are kept. Other restrictions, such as time range or instrumental magnitude ranges may also be specified. The collection of average objects, their detections, and the images from which they were derived now define a system of photometry equations. In this system, every image has a calibration offset magnitude (M_{cal}), every object has an average magnitude in a relative system (M_{rel}), and every detection of that object has a magnitude defined by the equation $M = M_{rel} + M_{cal}$. The goal is to solve for the values of M_{ref} and M_{cal}.

There are two points to note about this operation. First, the system of equations is generally much too large to solve directly; we must use an iterative technique to converge on a solution. Second, it is important in the analysis to use robust averaging and identify detections, stars, or images which are deviant in some way. These should be marked and given set weight in the solution. These cases may represent poorly measured objects (perhaps detections on or near a bad column), variable stars, and images obtained in poor weather conditions.

Relphot can also be used to determine the mosaic grid used to generate photometrically corrected flats (-grid option).

7 Uniphot : Zero Point Analysis

This operation uses the time history of relative photometry zero points for images and the spatial overlap information to determine a best set of image zero points which have a specific time scale for the atmospheric stability.

8 global astrometry analysis

This operation uses the reference and image detections to determine an optical distortion model for the camera and static astrometry model components. The astrometry model includes: (1) field distortion introduced by the telescope optics, which is a smoothly-varying function of the field position relative to the center of the telescope boresite coordinates. (2) focal plane geometry, which includes the chip positions and rotations in the focal relative to the boresite, along with chip-dependent plate-scale modifications needed to represent tilts or warps of the individual detectors relative to the ideal flat focal plane. .

9 DVO shell

9.1 User Commands

```
gcat                        -- get catalog at location
gimages                     -- get images at location
gstar                       -- get star statistics
extract                     -- extract average vectors from catalogs
mextract                    -- extract measurement vectors from catalogs
imstats                    -- plot image statistics
imextract                   -- extract image vectors from database
lcat                        -- list catalogs in display region
cmatch                      -- match two catalogs
```

There are a variety of other commands which directly refer to the photometry database. Some of these functions extract data of various types from the database, others perform more complex plotting operations. The commands listed above are those which simply extract data from the database. The first three list information relevant to a specific RA, DEC location on the sky: gcat (RA) (DEC) lists the catalog at the specified location and places the name in the variable $CATNAME, gimages (RA) (DEC) lists all images which overlap the specified location, gstars (RA) (DEC) (RADIUS) lists data about the stars within a specified radius of the specified location (all numbers above are given in decimal degrees). Similarly, lcat lists the catalogs in the region. Imstats lists statistics about each image

The next three commands extract a specific piece of information from the photometry database and places it in a vector. First, extract will extract average values for each star and place it in a vector. Next, mextract will extract measurement values for each star and place it in a vector: as a result a single star may have multiple entries in the measurement vectors. Finally, imextract will extract image statistics into vectors (not yet implemented).

DVO provides several ways to access the photometry information stored in the database. Several simple commands allow the user to extract 1 dimensional information directly from one of the primary database tables. The commands are:

```
 * imextract
 * avextract
 * mextract
 * imsearch
 * detsearch
```

imextract

This command allows the user to extract one of the columns from the image table, applying filtering as desired:

```
dvo: imextract
USAGE: imextract (value) [-region] [-time start range] [-photcode photcode]
dvo: imextract help
value may be one of the following:
 ra dec airmass Mcal dMcal Xm photcode time fwhm exptime nstar ncal sky flag
```

The extracted data is saved in a vector with the same name used to select the column. The vector name will have the same case as the choice given, but the column selection is case-insensitive (since there are no ambiguities in the database columns names by case).

avextract

This command allows the user to extract data from one of the Average table columns:

```
dvo: avextract
USAGE: avextract (from) (value) [options]
  from: cpt name or 'all'
  value: average.parameter or photcode
dvo: avextract all help
value may be one of the following:
 ra dec dmag Nmeas Nmiss Xm Xp Nphot Ncode flag type
```

This command takes as the first argument the name of one of the database regions. Alternatively, all regions currently displayed may be selection with the word 'all'. The second option specifies which column to select from the Average table. In addition to the basic data columns (ra, dec, etc), the magnitude-related average values (mag, dmag, Xm, Nphot, Ncode) are coupled to a photcode, which is thus required for these selections. The value 'mag' may also be subsituted with a primary or secondary photcode. Eg:

```
avextract all ra : select ra for all objects in displayed region
avextract all g  : select g magnitudes
avextract all mag -photcode r : select r magnitudes
avextract all Xm -photcode r  : select chisq values for r average mags
```

mextract

This command allows the user to extract data from one of the Measure table columns:

```
dvo: mextract
USAGE: mextract (from) (value) [options]
  from: cpt name or 'all'
  value: measure.parameter or photcode
dvo: mextract all help
value may be one of the following:
 ra dR dec dD mag dmag Mrel Mcal photcode time fwhm dophot xccd yccd xmosaic ymosaic flags
```

This command takes as the first argument the name of one of the database regions. Alternatively, all regions currently displayed may be selection with the word 'all'. The second option specifies which column to select from the Measure table. In addition to the basic data columns (ra, dec, etc), the magnitude-related average values (mag, dmag) are coupled to a photcode, which is thus required for these selections. The value 'mag' may also be subsituted with a primary or secondary photcode. Eg:

```
mextract all ra : select ra for all objects in displayed region
mextract all g  : select g magnitudes
mextract all mag -photcode r : select r magnitudes
mextract all Xm -photcode r  : select chisq values for r average mags
```

option filtering

The following extraction options allow the user to restrict the selections:

-time (start) (range)

select data for images within the given time range. The start date is given in the format YYYY/MM/DD,hh:mm:ss (any of these element may be dropped, in which case they default to 00 [for hh,mm,ss] or 01 [for MM,DD]). The date may also be written as a number of days followed by j (JD) or J (MJD). The two special date names "now" and "today" are also valid. The range is written as a number followed by a unit, with valid units of s (seconds), m (minutes), h (hours), d (days). Times are in UT. For example:

```
-time 2001/1/1 30d : select the range starting at midnight on 2001/1/1
 and ending 30 days (86400*30 seconds) later.

-time now -3h : select the time range starting three hours ago and
 ending now.
```

-region

restrict the selection to the currently display portion of the sky. This filter expects a portion of the sky to be plotted, and will only select data for images in the part of the sky. The algorithm for selecting the displayed region may not be perfect, so images near the boundaries may be unexpected included or excluded (this depends on the exact overlap details). Multiple queries with the same region will result in the same subset of images selected.

-photcode (photcode)

This restricts the selection to the given photcode. Images may only have photcodes of 'Dependent' type (eg, CFH12K.R.00). This filter allows the selection of images by exact match (selected photcode is Dependent type) or by equivalence match (selected photcode is Primary or Secondary type). Thus, selecting images based on CFH12K.R.00 will ignore images with photcode CFH12K.R.10, while selecting images based on photcode 'r' will return all images with CFH12K.R.*, since all are equivalent to 'r'. (NOTE: these details depend on the layout of the photcode table). fill this out with all other restrictions provided by photometry.c

```
catalog                    -- plot catalog stars
cgrid                      -- plot sky coordinate grid
cplot                      -- plot vectors in sky coordinates
czplot                     -- plot scaled vectors in sky coordinates
images                     -- plot image boxes
imdense                    -- image density plot
lcurve                     -- plot lightcurve for a star
pcat                       -- plot catalog boundaries
region                     -- define sky region for plot
```

Figure 5: Map of the entire sky, and images added to database.

Figure 6: Map of the sky in polar project, and images added to database.

```
resid                          -- plot residuals
simage                         -- plot stars in an image
```

There are two types of database plotting functions: those that display or refer to the spatial charateristics of the data and those that refer to other types of charatersitics, such as the time domain. The graphics window 0 is reserved for all plots of objects on the sky. The command region defines the current sky coordinates for plots in graphic window 0. The command pcat plots the outline of all photometry database files which are within the currently defined region (and by default, only those with data). images plots the outline of the images in the image database, while imdense shows the number of images at a location by randomly spacing dots within the boundary of the images. The command cgrid draws a grid in celestial coordinates on the for the current region.

The most complex, but also one of the most useful command is catalog, which plots the positions of stars in the photometry database (and others) on the sky. There are many options to this command. One set allows the user to plot stars from the photometry database (the default), from the HST GSC, or from an ASCII text file with RA, DEC, and Mag in specified columns. If the ASCII file has a fixed number of bytes per line, the data can be more quickly loaded. The size of the points may be scaled by the star magnitude, by the number of observations of the star, or by the number of missing datapoints for the star. In addition, points may be plotted only if they land in specified magnitude ranges, or with specified numbers of measurements, or missed measurements. Also, objects may be plotted only if they have a specified Average.code, so that only asteroids or only perfect stars may be plotted. The plotted vectors may be saved, if desired, and the source catalog epoch may be specified as different from J2000 (only valid for ASCII data).

Several other commands relate to non-spatial charateristics of images and stars. lcurve will plot a light curve for all stars within some radius of a point. resid plots the photometry residuals for a particular region file. Some Examples

Fig. 5 shows a map of the entire sky, and the location of the images currently in the database. This picture was made with the following commands: (output is not shown)

```
dvo: region 0 0 90 gls
dvo: cgrid
dvo: style -lw 2 -c red
dvo: images
dvo: ps
```

In this example, on the graphics window, the image boxes are shown in red. The user now has the possiblitiy of using the cursor command to narrow in on a specific region, and so forth.

Fig. 5 shows a map of the entire sky, and the location of the images currently in the database from a polar project. This picture was made with the following commands: (output is not shown)

```
dvo: region 0 0 90 zea
dvo: cgrid
dvo: style -lw 2 -c red
dvo: images
dvo: ps
```

Figure 7: Comparison between HST GSC and photometry database astrometry.

In this example, on the graphics window, the image boxes are shown in red. The user now has the possiblitiy of using the cursor command to narrow in on a specific region, and so forth.

Fig. 7 shows an example comparison of the photometry database star positions and the HST Guide Star Catalog star positions. The crosses are all objects in the photometry database, while the boxes are only the stars identified as USNO stars. The circles are the stars from the HST GSC. The size of both points is a function of brightness. This plot was made with the following commands (starting from the previous image):

```
dvo: cursor  (typed 1 on region of interest)
1 137.097858 22.698305
q 137.097858 22.698305
dvo: region $R1 $D1 0.2 TAN
dvo: cgrid
dvo: box
dvo: style -pt 0
dvo: gcat $R1 $D1
  0 n2230/1951.cpt *
dvo: style -pt 2; cat -all -m 12 18
dvo: style -pt 1; cat -all -m 12 18 -ID $USNO
dvo: style -pt 7; cat -all -m 12 18 -g
```

10 other user tools

10.1 delstar

10.2 getstar

10.3 imphotset

imphotset allows you to set certain phot.image table entries. here are the options:

imphotset [-photcode code] [-name foo] [-trange (start) (stop)] -flag and value

Here is a complete list of relphot configuration variable names, a quick description, and reasonable values to start with:

```
--- configuration variables used by relphot ---
MAG_LIM : float
  ignore measurements fainter than this absolute magnitude

SIGMA_LIM : float
  ignore measurements with magnitude error larger than this value

STAR_SCATTER : float
  mark stars as bad if their scatter is larger than this value

IMAGE_SCATTER : float
  mark images as bad if their scatter is larger than this value

IMAGE_OFFSET : float
  mark images as bad if the absolute value of thie zero point offset
  is larger than this number

STAR_CHISQ : float
```

```
  mark stars as variable if their reduced chisq are larger than this value

STAR_TOOFEW : int
  mark stars as bad if the have fewer than this number of valid measurements

IMAGE_TOOFEW : int
  mark images as bad if the have fewer than this number of valid measurements

IMAGE_GOOD_FRACTION : float
  mark images as bad if the have fewer than this fraction of valid measurements

IMAGE_CATALOG : string
  name of the image catalog file

IMAGE_CATALOG_TEMPLATE : string
  name of the template file to create the image catalog file

CATALOG_TEMPLATE : string
  name of the template file to create the catalog file

GSCFILE : string
  name of the GSC region table

CATDIR : string
  directory where the database is stored

PHOTCODE_FILE : string
  file containing photometry code information

ZERO_PT : float
  default zero point for random data

RELPHOT_GRID_X : int
  scale of mosaic correction grid

RELPHOT_GRID_Y : int
  scale of mosaic correction grid

RELPHOT_GRID_BINNING : int
  deprecated

CAMERA_CONFIG : string
  name of the file containing descriptive information about the camera

--- sample ConfigFile entries with typical values ---

MAG_LIM                 24.0
SIGMA_LIM                0.05
STAR_SCATTER             0.05
IMAGE_SCATTER            0.05
IMAGE_OFFSET             0.2
STAR_CHISQ              10.0
STAR_TOOFEW              3
IMAGE_TOOFEW            10
IMAGE_GOOD_FRACTION
IMAGE_CATALOG           $CATDIR/Images.dat
IMAGE_CATALOG_TEMPLATE  $REFSDIR/elixir/template.cat
CATALOG_TEMPLATE        $REFSDIR/elixir/template.cat
GSCFILE                 $REFSDIR/gsc/GSCregions.tbl
CATDIR                  $CATDIR
PHOTCODE_FILE           $CONFDIR/camera/$CAMERA.photcode
ZERO_PT                 25.0
RELPHOT_GRID_X          4
RELPHOT_GRID_Y          8
RELPHOT_GRID_BINNING    512
CAMERA_CONFIG           $CONFDIR/camera/$CAMERA.config
```

11 Performance

DVO design partly driven by the need to make the detection-object associations quickly and to processes the incoming detections at a sufficiently high rate to meet the throughput requirements. For each upload of the object detections from a complete FPA, DVO must match roughly 1.4×10^6 detections from an FPA with roughly 6.4×10^6 objects, including orphaned bright detections. This corresponds to roughly 640 MB, if each object uses 100 bytes for its descriptive informations (more than is currently specified in the Object table). With a throughput of 100 MB/s for reads from a RAID, DVO can perform the data read in a fraction of a second if the data is distributed across 10 computers.

A DVO Tables

Table 1: DVO Database Tables

Table Name	**Description**
Images	The images that have objects in the DB.
Image Overlaps	Image regions which are touched by specific images.
Objects	The objects — average properties of multiple detections of the same object.
Average Magnitudes	Average photometry in multiple filters
Solar System Objects	Identification of solar system objects
Matched Detections	Detections of sources in an image identified with an Object.
Orphaned Detections	Detections of sources in an image not identified with an Object.
Non-detections	Non-detections of objects in an image.
SkyRegions	spatial distribution of tables
Filters	Filters understood by the system.
Photcodes	Transformations between different photometric systems
Zero Points	History of Zero-point & Airmass terms
Distortion Models	History of Optical Distortion terms
Database Hosts	computers used to store the tables

Table 2: DBO Detection Classes & Object Parameters

Object Parameter	P2	P4S	P4D	SS
PSF x,y, covar, α, δ	+	+	+	+
PSF mag, σ_{mag}	+	+	+	+
star/gal sep	+	+	+	+
σ_x, σ_y, θ	+	+	+	+
local sky data	+	+	+	+
Petrosian R, M, R_{50}, R_{90}	-	+	-	+
Sérsic R, M, AB, ϕ, ν	-	+	-	+
W.L. γ_1, γ_2, pol. terms	-	-	-	+
exp. spaced aps., Poisson noise, variance	-	-	-	+

Table 3: Images

Column Name	**Datatype**	**Description**
NAME	char[32]	name of original image
TZERO	e_time	readout time (row 0)
COORDS	Coords	astrometry
NSTAR	unsigned int	number of stars on image
SECZ	float	airmass, mag
NX	short	image width
NY	short	image height
APMIFIT	float	aperture correction
DAPMIFIT	float	apmifit error
MCAL	float	calibration mag
DMCAL	float	error on Mcal
XM	short	image chisq
SOURCE	short	photcode
EXPTIME	float	exposure time (seconds)
ST	float	sidereal time of exposure
LAT	float	observatory latitude (degrees)
DETECTION_LIMIT	unsigned char	detection limit (10*mag)
SATURATION_LIMIT	unsigned char	saturation limit (10*mag)
CERROR	unsigned char	astrometric error (50*arcsec)
FWHM_X	unsigned char	PSF x width, (25*arcsec)
FWHM_Y	unsigned char	PSF y width, (25*arcsec)
TRATE	unsigned char	scan rate, (100 usec/pixel)
CODE	char	image quality flag
CCDNUM	unsigned char	CCD ID number
ORDER	short	Mrel polynomial order
MREL_POLY	short[14]	Mrel polynomial
DUMMY	char[18]	expansion

Table 4: Objects

Column Name	Datatype	Description
RA	double	RA,
DEC	double	DEC,
MAG	float	primary mag,
MAG_ERR	float	error on primary mag,
V_RAf	float	proper-motion (arcsec/year),
V_DEC float	proper-motion (arcsec/year)	
PAR float	parallax (arcseconds)	
D_V_RA float	proper-motion error (arcsec/year)	
D_V_DEC float	proper-motion error (arcsec/year)	
D_PAR float	parallax error (arcseconds)	
SIGMA_POS	short	position scatter,
CHISQ_MAG	short	chisq for primary mag,
CHISQ_GAL	short	chisq for galaxy mags,
NMEAS	unsigned short	number of measures
NMISS	unsigned short	number of missings
CODE	unsigned short	ID code (star, ghost, etc)

Table 5: SecFilt Magnitudes - NOTE: corresponding photcodes defined externally for the table sequence, Average Object association defined by sequence

Column Name	Datatype	Description
MAG	float	other mags, mags
MAG_ERR	float	scatter on mag mags
MAG_CHI	short	chisq on mag [100*log(value)]

Table 6: Matched Detections

Column Name	Datatype	Description
D_RA	float	RA offset, arcsec
D_DEC	float	DEC offset, arcsec
MAG	float	catalog mag, mag
MCAL	float	image cal mag, mag
MGAL	float	galaxy mag, mag
DM	float	mag error, mag
AIRMASS	float	(airmass - 1), airmass
DT	float	exposure time, 2.5*log(exptime)
FWX	short	object fwhm major axis, 1/100 of arcsec
FWY	short	object fwhm minor axis, 1/100 of arcsec
THETA	unsigned char	angle wrt ccd X dir, (0xff/360) deg
DOPHOT	char	dophot type
SOURCE	unsigned short	photcode
FLAGS	unsigned short	flags for various uses
T	unsigned int	time in seconds (UNIX)
AVEREF	unsigned int	reference to average entry
OBJECT ID		
SKY		
D_SKY		

Table 7: Orphaned Detections

Column Name	Datatype	Description
α		
δ		
σ_α		
σ_δ		
$M_{\rm inst}$		
$M_{\rm cal}$		
$\sigma_{\rm mag}$		
photcode		
type		
flags		
time/date		
airmass		
σ_x		
σ_y		
θ		
exptime		
sky		
$\sigma_{\rm sky}$		
etc		

Table 8: Non-detections

Column Name	Datatype	Description
object ID		
$N_{\mathrm{non-det}}$		
last time/date		
last mag		
faintest time/date		
faintest mag		

Table 9: Regions

Column Name	Datatype	Description
`R_MIN`	`float`	
`R_MAX`	`float`	
`D_MIN`	`float`	
`D_MAX`	`float`	
`CHILD_S`	`int`	sequence number in full table of first child
`CHILD_E`	`int`	sequence number in full table of last child + 1
`PARENT`	`int`	sequence number in full table of parent
`INDEX`	`int`	sequence number in full table of this entry
`DEPTH`	`char`	depth of this entry
`CHILD`	`char`	does this entry have children?
`TABLE`	`char`	does this entry have a table?
`NAME`	`char[21]`	name / filename

Table 10: Image Overlaps

Column Name	Datatype	Description
Image ID		
Region Table		

Table 11: Filters

Column Name	Datatype	Description
Filter ID		
Filter name		
Photcode		
λ_0		
δ_λ		
ϵ		
transmission curve		
time/date		

Table 12: Photcodes

Column Name	Datatype	Description
CODE	unsigned short	code number (stored in Measure.source)
NAME	char[32]	name for filter combination
TYPE	char	PRI/SEC/DEP/REF
C_LAM	short	primary phot calibration terms (millimags)
C_LAM_ERR	short	primary phot calibration terms (millimags)
X_ERR	short	primary phot calibration terms (millimags)
K	float	secondary phot calibration terms (millimags)
C1	int	color is average.M[c1] - average.M[c2]
C2	int	color is average.M[c1] - average.M[c2]
EQUIV	int	this dependent filter is equivalent to equiv PRI/SEC
NC	int	number of color terms
X	float[4]	color terms X[0]*mc + X[1]*mcˆ2 + X[2]*mcˆ3
Telescope		
Camera		
Detector		
Filter		

Table 13: Zero Point History

Column Name	Datatype	Description
ZP_OBS	float	measured zero point, mag
ZP_REF	float	nominal zero point, mag
ZP_ERR	float	error on zero point, mag
C_AIRMASS	float	airmass coeff, mag per airmass
C_COLOR	float	color coeff, mag per mag
START_TIME	e_time	start time of measurement, seconds since 1 Jan 1970 UT
STOP_TIME	e_time	stop time of measurement, seconds since 1 Jan 1970 UT
C1_CODE	short	code 1 for color, photcode
C2_CODE	short	code 2 for color, photcode
PHOTCODE	short	photcode, photcode
LABEL	char[64]	data label
REFCODE	rawshort	photcode, photcode
N_TIME	int	number of times
N_MEAS	int	number of measurements

Table 14: Distortion History

Column Name	**Datatype**	**Description**
Camera		
Telescope		
distortion terms		
time/date		
residuals / error		
N stars		
N images		
astrom ref set		

Table 15: Database Hosts

Column Name	**Datatype**	**Description**
machine name		
machine ID		

Table 16: Solar System Objects

Column Name	**Datatype**	**Description**
SSO ID		
$N_{\rm det}$		