Outputs of the Photometric Pipeline

Robert Lupton

March 10, 1997

1 Introduction

There's a note on mask structures at the end of this note for the not-yet-cognoscenti.

2 File Formats

1. Corrected Frames will be written to fits files containing images and masks, where the format of the corrected frames files will be:

   a PDU giving e.g. the frame number

   5*{

   An `XTENSION IMAGE` giving the corrected image

   A set of (currently 8) `XTENSION BINTABLE`s, each containing one of the 'bitplanes' (e.g. `MASK_INTERP`); in these the `OBJMASK` triples will be in the heap.

   }

   Note that this hierarchical structure enables us to correctly describe the data in heap, which could not be achieved if we used only a single binary table expressing the full SPANMASK. Dervish will provide the code to write the IMAGE extension, and I'll deal with the tables in photo.

2. OBJCs will be split into three structures for output; the main motivation for this is to write fields in each of the ncolor OBJECT1s together (e.g. rowc[5] not 5 rowc columns). The structs are:

`OBJC_IO`. The main tabular data. This table has no data in heap.

`ATLAS_IMAGE`. The masks and regions making up the atlas images. The structure of this binary table will be complex, but there will be a function to read a given row into a defined structure in memory. We shall provide a stand-alone programme that links this function and breaks atlas image files out into (many) fits binary images (or maybe one with IMAGE extensions; TBD). As of March 1996 this stand-alone programme is not written, although photo is able (and does) read atlas image files.

`struct test_info` Anything that the photometric pipeline testers want saved to disk. This product is *not* written out during production runs and it is not a saved product of the survey. An example might be the positions of all the peaks detected in each object in each band.

3. Parameters describing photo's performance are also written out.

3 `OBJC_IOs`: Photo's Main Tabular Output

First some outputs which describe the object as a whole:

`id` An id for the object within the frame; this id is the same as the object's row number in the table, and is used to tie the `OBJC_IO` and `ATLAS_IMAGE` tables together.

`parent` The `id` number of the object's parent, or -1 if it is a primary detection.

`ncolor` The number of colours present in this table; should always be 5 during normal operations.

`objc_type` An enum giving the object's overall classification. Current possibilities are:

UNK An object of unknown type (the default).

CR A cosmic ray

DEFECT Some indeterminate chip defect.

GALAXY A galaxy

GHOST A ghost produced by the 2.5m optics.

KNOWNOBJ A known object (from e.g. FIRST or ROSAT); its position and size are inputs to photo.

STAR A Star

TRAIL A satellite, aeroplane, meteorite, or asteroid trail.

SKY A pseudo-object; a location where no object was detected in any of the survey bands.

These categories are repeated below for each band; the algorithm used to arrive at the overall classigfication is TBD.

`catID` A catalog id number associated with KNOWNOBJs. We may want to revisit how this is done when some real known object catalogues are available.

`objc_flags` The union of certain of the flag bits set in each individual band (see discussion of `flags` below). Specifically, if the flags BLENDED, CHILD, EDGE, INTERP, MANYPETRO, NOPETRO, NOTCHECKED, or SATUR are set in any band, they are also set in `objc_flags`.

`objc_rowc, objc_colc, objc_rowcErr, objc_colcErr` The canonical position of the object (and 1-σ errors), in the r' coordinate system. If an object is detected in r', this is the r' centre; otherwise it's a suitable average of the bands where the object was detected (Pixels).

The following fields are calculated for each band:

`rowc, colc, rowcErr, colcErr` The position of the object (and 1-σ errors) in each band. In the case that an object is not detected in a given band (say f'), the position is taken to be the r' position offset to the f' coordinate system (if detected in r'), and a suitable average of the other bands otherwise. (Pixels).

We will describe the position assigned in the r' as the r' *position*, even if the object was not in fact detected in r'.

`sky, skyErr` The sky level (and the 1-σ), at the position of the object. (Counts/pixel^2).

`psfCounts, psfCountsErr` The PSF-flux (and the 1-σ error), at the position of the object (Counts).

`fiberCounts, fiberCountsErr` The 3"-counts (and the 1-σ error), at the r' position of the object. These counts are supposed to be corrected to a canonical seeing. $\sigma_{\mathrm{nominal}}$ (Counts).

`petroRad, petroRadErr` The Petrosian radius (and the 1-σ error), measured using the r' position of the object. See notes on Petrosian quantities in section 5 of this document (Pixels).

`petroCounts, petroCountsErr` The Petrosian counts (and the 1-σ error) within $f_3 r_{P,r'}$ of the r' centre. Suitable measures must be taken if the object is not detected in r' (Counts).

`petroR50, petroR50Err` The Petrosian 50%-light radius (and 1-σ error) (Pixels).

`petroR50, petroR90Err` The Petrosian 90%-light radius (and 1-σ error) (Pixels).

`Q, U, QErr, UErr` The values of $\langle \mathrm{col}^2/r^2 - \mathrm{row}^2/r^2 \rangle$ and $2\langle \mathrm{colrow}/r^2 \rangle$ (and their 1-σ errors), measured within $r_{P,r'}$. These are estimators of $(a-b)/(a+b)\cos 2\phi$ and $(a-b)/(a+b)\sin 2\phi$, and are named by analogy to the usual Stokes parameters. For more details, see *The Estimation of Object's Ellipticities*.

`nprof` The number of points in the three succeeding measures of the radial profile, each of which refers to annuli with fixed outer radii of { 0.56, 1.69, 2.58, 4.41, 7.51, 11.58, 18.58, 28.55, 45.50, 70.51, 110.5, 172.5, 269.5, 420.5, 657.5 } pixels, that is { 0.23 0.68 1.03 1.76 3.00 4.63 7.43 11.42 18.20 28.20 44.21 69.00 107.81 168.20 263.00 } arcseconds. These radii correspond to circular apertures enclosing an integral number of pixels: { 1, 9, 21, 61, 177, 421, 1085, 2561, 6505, 15619, 38381, 93475, 228207, 555525, 1358149 }.

`profMean` The mean surface brightness within the innermost `nprof` annuli, with fixed radii given above; these fluxes may be used to recover the annular counts exactly (counts/pixel).

`profMed` The 'median' surface brightness within the innermost `nprof` annuli, with fixed radii given above (counts/pixel). The exact definition of these medians is TBD.

`profErr` An estimate of the uncertainty in the profiles (counts/pixel); note that this is *not* the photon noise (which can be recovered from `profMean`), but an estimate of the true uncertainty allowing for contamination by stars, HII regions, etc.

`iso_rowc, iso_colc, iso_a, iso_b, iso_phi, iso_rowcErr, iso_colcErr, iso_aErr, iso_bErr, iso_phiErr, iso_rowcGrad, iso_colcGrad, iso_aGrad, iso_bGrad, iso_phi` The centre, major and minor axes, and position angle of a certain isophote (Pixels). These will be determined from the 2-dimensional extracted profile. The `Grad` quantities are correction terms allowing us to correct for errors in the photometric calibration.

`r_deV, I_deV, ab_deV, phi_deV, r_deVErr, I_deVErr, ab_deVErr, phi_deVErr` Parameters of the de Vaucouleurs profile that best fits the radial profile (as determined by the cell array), and errors. The `r` and `I` parameters are the effective radius and the surface brightness at that point.

`r_exp, I_exp, ab_exp, phi_exp, r_expErr, I_expErr, ab_expErr, phi_expErr` Parameters of the exponential profile that best fits the radial profile (as determined by the cell array), and errors. The `r` and `I` parameters are the effective radius and the surface brightness at that point.

`star_L, exp_L, deV_L` Likelihoods for the fits of the model by the PSF, an exponential disk, and a de Vaucouleurs profile. More specifically, the values quoted the probabilities of finding a value of χ^2 at least as large as that found for the model fits.

`fracPSF` The fraction of the total light in the profile that can be assigned to a point source. *Not currently calculated.*

`texture` A measure of the roughness of the object, based on the residuals after inverting the image and subtracting. *Not currently calculated.*

`flags` Some more information about how the processing went. Current possibilities are:

NOTDETECTED Object wasn't detected in this band

BRIGHT Object was found by findBrightObjects

EDGE Object was too close to edge of frame to be measured

BLENDED Object is/was blended

CHILD Object is a deblended child

PEAKCENTER Given centre is position of peak pixel, rather than an MLE fit

NODEBLEND No deblending was attempted, although the BLENDED flag is set

PETROFAINT; At least one possible Petrosian radius was rejected as the surface brightness at r_P was too low. If NOPETRO isn't set, an (different) acceptable Petrosian radius was found.

NOPETRO The object has no Petrosian radius

MANYPETRO The object has more than one Petrosian radius; the largest found is adopted

NOKENT_SMALL An object with no Petrosian radius has no Kent radius either because the central mean value is already below k1.

NOKENT_BIG An object with no Petrosian radius has no Kent radius either because the mean value never falls to k1.

MANYKENT An object with no Petrosian radius has more than one Kent radius.

MANYR50 An object has more than one 50% light radius

MANYR90 An object has more than one 90% light radius

BAD_RADIAL The radial profile extends beyond where its S/N first drops to (??) 1.

EXTRAP_STOKES The part of the object with measured Stokes parameters doesn't extend to $r_{P,r'}$, so photo was forced to extrapolate.

INTERP The object contains at least one pixel that has been interpolated

SATUR The object contains at least one saturated pixel

NOTCHECKED The object contains at least one pixel that is marked as having not been searched for objects

SUBTRACTED Bright wings were subtracted from this object (presumably a star).

NOSTOKES Object has no measured stokes parameters.

BADSKY The sky level is so bad that the highest pixel in the object is *very* negative; far more so than a mere non detection. No further analysis is attempted.

`type` The type assigned to the object in this colour; the possibilities are described for the `objc_type` field.

4 Usage of Photo's Output Parameters in Target Selection

The following table shows which of photo's outputs are used in target selection by the various groups assigned fibres.

	Stars	Galaxies	QSOs	Serendipity
`id`	X	X	X	X
`ncolor`				
`objc_type`	X	X	X	X
`catID`	X		X	X
`rowc, colc, rowcErr, colcErr`	X	X	X	X
`sky, skyErr`				
`psfCounts, psfCountsErr`	X		X	X
`fiberCounts, fiberCountsErr`	X	X	X	X
`petroRad, petroRadErr`		X		
`petroCounts, petroCountsErr`		X		
`petroR50, petroR50Err`		X		
`petroR50, petroR90Err`		X		
`Q, U, QErr, UErr`	X		X	
`nprof`				
`profMean`				
`profMed`				
`profErr`				
`iso_rowc, iso_colc, iso_ iso_b, iso_phi`				
`r_deV, I_deV, ab_deV, phi_deV`				
`r_exp, I_exp, ab_exp, phi_exp`				
`star_L, exp_L, deV_L`	X	X	X	X
`fracPSF`	X		X	
`flags`	X	X	X	X
`type`	X		X	X

Note: Large Scale Structure and Clusters have provided input to, and will use, the Galaxies target selection criteria.

5 Petrosian Quantities

(This discussion is stolen from and supersedes the corresponding parts of Michael Strauss' document *Galaxy Selection Algorithm for SDSS*).

Let $I(r)$ be (a spline fit to or other smooth representation of) the measured azimuthally averaged surface brightness profile[1] of an object in r'.

[1] The profile employed will be the `profMed` described above; `profMean` will be used in calculating fluxes. The profile used for calculating Petrosian quantities is cutoff at the first radial bin for which the S/N is below 1.0. To avoid the neccessity of interpolating to

Define the Petrosian ratio $\mathcal{R}_P(r)$ as the ratio of the local surface brightness at radius r to the mean within r:

$$\mathcal{R}_P \equiv \frac{\int_{0.8r}^{1.25r} I(r')2\pi r' dr'/[\pi(1.25^2 - 0.8^2)r^2]}{\int_0^r I(r')2\pi r' dr'/[\pi r^2]}.$$

Mark all the radii $r_i, i = 1, ..., N$ where $\mathcal{R}_P$ falls to a specified value f_1, and for which $I(r) > f_2$.

If there's at least one such radius ($N > 0$), the largest of the r_i will be taken as the Petrosian radius r_P; if $N = 0$, the adopted radius will be a Kent radius one given by the solution to $\langle I(r) \rangle = f_3$, the point where the mean surface brightness falls to some value f_3. If more than one such Kent radius exists, the smallest is adopted; if no such radius exists, there are two possibilities: that the lowest surface brightness in the object is above f_3 (in which case we adopt $r_P = r_{\max}$, the largest "good" radius in the profile), or that the highest surface brightness is below f_3 (in which case we take $r_P = f_5$).

The Petrosian flux F_P is defined as the total flux as measured within a certain number of Petrosian radii:

$$F_P = \int_0^{f_4 r_P} I(r')2\pi r' dr'.$$

The Petrosian half-light r_{50} is defined by the implicit equation:

$$\int_0^{r_{50}} I(r')2\pi r' dr' = 0.5 F_P.$$

The Petrosian 90% radius is defined by the implicit equation:

$$\int_0^{r_{90}} I(r')2\pi r' dr' = 0.9 F_P.$$

How should we set these surface brightness values? Michael Strauss suggests the following: For a Freemen disk (central surface brightness in r' of 20.85, using the $B - r'$ colors of disks from Frei and Gunn), the Petrosian ratio falls to 1/4 at 3.21 scale lengths, and to 1/8 at 4.43 (with corresponding surface brightnesses of 24.33 and 25.66). The Kent surface brightness at these radii are 22.83 and 23.40. If we decide that f_2 should be a magnitude fainter than the value for a Freeman disk, and take f_5 to be twice the fibre radius, we arrive at the values given in table 1.

determine fluxes, the actual radial profile will be recorded out to the adopted Petrosian radius.

f_1	Petrosian Ratio	0.25	0.125
f_2	Minimum Surface Brightness at r_P	25.3	26.7
f_3	Kent Surface Brightness	22.8	23.4
f_4	Multiple of r_P for Petrosian flux	2.75	2
f_3	Fallback Radius	3"	3"

Table 1: Two possible sets of strawman values for the parameters that photo uses to measure Petrosian quantities.

6 Note on Masks Structures

The standard shiva mask is a bitmask, with 8 bits for each pixel in the image. They have various disadvantages; they take up a lot of memory (a full mask is 3Mby) — not so much a concern per se, but a cache disaster; you cannot find all the saturated pixels (say) without running an object finder on the object mask; you have to check each pixel as you process it to see if it's OK. All of these concerns are addressed by the representation of each bitplane as a set of 'OBJMASK's, which are a structure consisting of row, col1, col2 for each line-segment where the bitmask would have been set (they also contain a bounding box and other book-keeping information). These sets, currently CHAINs, are then assembled into 'SPANMASK's, which are arrays of the chains of OBJMASKs, one for each bitplane in the old shiva masks.

Note that as our bitplanes are sparse, much less memory is involved in these data structures; you can ask for a list of all cosmic rays (it's a (possibly empty) chain of OBJMASKs); and you can ask for all the pixels in this object that aren't in the NOTCHECKED chain, and then simply loop over the spans in question, or ask for the pixels in this object that are saturated.

The downside is that a SPANMASK doesn't match a fits binary table very well. An chain of OBJMASKs isn't too bad — one row for each OBJMASK, with the r, c1, c2 triples in the heap as properly-byte-swapped-shorts — but to put a whole SPANMASK in requires that all of the OBJMASK data appear in the heap, and then there's a nightmare about byte order, as you have to know if the heap data is 1, 2, or 4 byte units.