DIFFERENTIAL ASTROMETRIC PROCESSING FOR LSST

Dave Monet – dgm@nofs.navy.mil

I. Introduction and Statement of the Problem

The Requirements for the astrometric processing of LSST data flows from the SRD. Please note the distinction between *absolute* astrometry, which refers to the calculation of positions with respect to the reference frame of J2000, and *differential* astrometry, which deals with computing the positions and their time derivatives in distinct, small patches of sky. It is the nature of the calculation that the differential solution, which provides measures for the stellar parallax, proper motion, and perturbations, can be substantially more accurate than the knowledge of the absolute (i.e., J2000 coordinates) of the object. This discussion addresses the differential astrometric pipeline for LSST.

The core of the astrometric algorithm is the simultaneous solution for two types of unknowns, the coefficients that transform the coordinates measured in exposure into some common coordinate system, and the positions and motions of each star. Whereas a direct solution exists, it involves the inversion of relatively large matrices and is rarely used. Instead, the solution is based on an iterative improvement given the prior knowledge of the star positions (from a reference catalog or similar) and iteration. All observations for all stars in a small area of sky are extracted from the database. Using the catalog positions for the stars as a first guess, the transformations from each observation to the catalog system are computed, and then all measures for each star are use to compute the new values for position, motion, etc.

Perhaps the biggest unknown is the size of the solution patch. The camera has several "natural" sizes: the whole camera, each raft, each CCD, and the area of each CCD read through a single amplifier. Given the circumstances of each observation, particularly the jitter between the requested and actual boresight, the intentional dither of each observation with respect to a system of boresights on the sky, and the rotation angle of the camera, there are no obvious "natural" sizes on the sky. Instead, a scheme for tessellation of the sky can be adopted, and this makes an unambiguous assignment of each star into a specific tile. Currently, the size of the solution patch is derived from the outer scale of the seeing. Various experiments done with precursor LSST data indicate that the short exposure time sets an upper limit on the size of the solution patch. If the patch is small enough, the astrometric impact of the unaveraged turbulence can be mapped with a simple polynomial, and the differential astrometric accuracy approaches that set by the photon statistics. If the patch is too large, the residual turbulence does not have a simple model, and the resulting astrometric accuracy is substantially degraded.

The current expectation is that the LSST patch ought to be at least a few arcminutes in size, but may not be much larger than perhaps 10 arcminutes. I have adopted the HEALPix tessellation scheme, and these define tiles based on the number of sides. For example, $N = 64$ sides gives tiles of an approximate size of 55 arcminutes on a side, whereas $N = 512$ sides gives tiles of about 7 arcminutes. Given this approach, the complexity of the astrometric solution can be estimated. The area of the $N = 512$ tile is about 0.014 square degrees and the area of the LSST camera is about 8 square degrees. This means that there are about 600 tiles in each observation. Sky tiles can cross CCD boundaries, so perhaps there are 1000 tile solutions per exposure. LSST is expected to take 300,000 exposures per year which means that we can expect 30,000,000 tile solutions per year. While this isn't a huge number, it does suggest that some effort could be spent optimizing the calculation as well as understanding intermediate storage is needed, etc. Note that several scientific programs are enabled by doing several all-sky astrometric solutions per year.

II. Walk-through of the Existing Solution Algorithm

I have tried to code the algorithm described above several ways, but I have made only modest progress. My current development effort is more successful than previous ones, but there is much room for examination and improvement. I have tried to strike a balance between presenting enough detail to explain the issues, and omitting unnecessary complexity. Comments are more than welcome, and more explanation can be provided on request. The following steps were needed before doing the astrometric processing.

0) Perhaps the most important single decision is the algorithm for computing image positions. In an ideal world, we might expect the same algorithm to be optimal for both astrometric and photometric analysis.

I really worry about this. Photometric analysis needs to measure all the flux from an image whereas astrometric analysis concentrates on the portion of the image where the flux gradient is highest. Think about it: no gradient means no center. The algorithm currently used for the USNO Flagstaff astrometry program fits the image core to a Gaussian and ignores all flux external to 2.7 times the computed sigma. This algorithm seems to work very well for astrometry, but is terrible for doing photometry. It is not yet clear that LSST should assume that the photometric and astrometric positions are the same. A safe decision would allow for different algorithms and separate columns in the database.

1) *X,Y,M,CCD* The pixel positions *(X,Y)* are computed by the pipeline, as is some internal magnitude *M*. The calibration of this magnitude is not critical because its only astrometric purpose is to weight solutions if photon statistics are an issue. Various metadata are assumed to be provided such as which CCD, the temperature, etc., and a unique identifier for each observation is presumed.

2) Approximate absolute astrometric solution: The nominal telescope and camera parameters should allow for the blind computation of the transformation between *(X,Y,CCD)* to (α,δ) with an accuracy better than one arcsecond.

3) Correlation with reference catalog: The detections from the various observations needed to be merged into objects, and the unique object identifier must to be associated with each detection.

4) Storage HEALPix: This might be an artifact of how I am doing the solution, but I found the need to tessellate the sky in tiles that contained a reasonable amount of data. If I stored all from a single observation in a single file, I ran out of memory when processing large numbers of observations. If I split each observation into sky HEALPix, I had so many disk accesses that the processing was unacceptably slow.

5) Storage by HEALPix: My simulator created a sequence of observations (CRONOS.92 was too big), and a single file for all simulated detections. Each file was read, the necessary data were extracted, the storage HEALPix for the correlated catalog star was computed, and the data were saved in the appropriate file.

6) Storage of Reference Catalog by HEALPix: The astrometric processing needs the current (best) initial guess, so the reference catalog needs to be stored in a form compatible with the observations.

7) First Guess for the New Catalog: The reference catalog is cloned as the first guess for the new catalog. The reference catalog is assumed to be static, but the astrometric processing should start with the current best estimation for all attributes.

8) Sky HEALPix: The size of the processing patch is specified.

9) Epochs for Sky HEALPix: I am still unsure if this step is needed, but it made sense at the time. The astrometric algorithm needs a coordinate system for each tile, and it makes sense to use a tangent plane projection instead of doing all processing in spherical coordinates. Hence, I adopted the nominal $\alpha,\delta)$ for the sky HEALPix as the tangent point, and the instant of upper culmination in the first 24 hours after 2000.0 as the epoch. This aligns the axes of the tile with the cardinal directions, and means that the tile positions are a very good approximation of the positions at the epoch of the reference catalog. The calculation of the upper culmination is expensive and is needed for the large number of sky HEALPix tiles, but only needs be done once.

The items above are needed before the astrometric processing can start. Once all is ready, the following steps implement my current algorithm.

10) Outer Loop Over Storage HEALPix: For each tile to be examined, the reference and new catalogs are read, as are all of the observations for all of the stars in this tile. The initial guess is taken from the new catalog except for stars not yet measured whose positions must come from the reference catalog.

11) Inner Loop Over Sky HEALPix: Since the storage tiles hold a unique set of sky tiles, all stars will eventually be processed.

12) Isolate By CCD+Observation: Since a single CCD can contain stars on several sky HEALPix, separate transformations are needed for each CCD in each observation.

13) Solve For The X and Y Transformations: For each unique Observation+CCD, there is a list of nominal positions and the list of observed positions. Perhaps I worry too much, but I have made this step very complicated. Given the current estimators for the metadata, I transform the pixel coordinates to sky coordinates, transform sky coordinates to sky tile coordinates, and then do the least squares fits to polynomials in sky tile coordinate system. Doing this pair of transformations allows systematic effects to be modeled and removed, and uses the least squares fit to solve for the seeing- and telescope-based terms.

14) Isolate By Star: We need to process each star separately.

15) Solve For Position, Motion, etc.: Using the sky tile coordinates and the metadata for each observation (epoch, parallax factor, zenith distance, etc.), a separate solution is done in X and Y for each star. The equation is similar to

$$\bar{x} = a + b \times T + c \times P_x + d \times Z_x$$

where T is the epoch in fractional years so that the coefficient b is the annual proper motion, P_x is the parallax factor in X (RA) computed from the observation metadata so that the coefficient c is the annual parallax, and Z_x is the refractive effect in X also computed from the observation metadata. A similar equation is used in Y.

16) WARNING! There are some who argue that the X and Y solutions above (15) should be coupled because there is only one value for the parallax. I despise this practice, but encourage those who espouse this approach to make their case before the algorithm is coded.

17) Iterate 12 – 16: As noted above, this solution is approximate because proper motion, parallax, and refraction are assumed to be zero for the first iteration (assuming it was not contained in the reference catalog), so the transformations in (12) will be somewhat distorted. After a few iterations, least squares can sort this out.

18) Update New Catalog: Depending on the range of epoch, parallax factor, and refraction, new values for any or all of these parameters were computed for each star.

19) Update Predicted Position For Each Star: With new values for the star parameters, the predicted positions in the solution sky tile must to be updated.

20) Next Sky HEALPix:

21) Save New Catalog: Since stars are uniquely assigned to a catalog HEALPix, all possible stars have been examined.

22) Next Catalog HEALPix:

III. Issues and Discussion

What is the proper size for a sky HEALPix? As the size gets larger, the number of solutions decreases and the number of reference galaxies and QSOs increases. Hence, it is reasonable to use the largest size consistent with the scale length of the seeing, which is as yet unknown. If the seeing were not an issue, the stability of CCDs in a raft and rafts in the camera starts to enter for sizes much larger than a single piece of silicon, about 14 arcminutes for LSST. Given the telescope scale, 1 milliarcsecond is 20 nanometers in the focal plane. This lies well beyond any of the propose camera metrology systems, so fixing multiple CCDs or rafts in a single solution may not be a reasonable thing to do.

What about sky vs. storage HEALPix? I don't have much experience with database and my telescope simulator produces a single file containing *(X,Y,CCD)* for each observation. In the case of LSST, the output of the pipeline is ingested by one or more databases, and this is a step that I have not simulated. However, I should point out that astrometric processing uses every detection from every observation, and this is no small number of queries. It would be nice to spin up some of the database pundits to understand whether this is a significant load or not. I anticipate wanting to do a full astrometric solution a few times per year, at least as the survey is getting started.

How many intermediate solutions do we save? I see a clear case to save the full solution for each star because these are the parameters of scientific interest (parallax, proper motion, etc.), and the residuals might be processed in the search for unseen companions. I don't see a compelling reason to save the transformation for each sky HEALPix for each CCD for each observation so long as the uncorrelated detections are transformed and saved. My simulator doesn't worry about new stars, missing stars, stars that come and go as a result of variability, etc.

As noted above (16), some people advocate a single solution that combines X and Y so that only a single parallax is fit. Others such as myself, use the separate values and their uncertainty estimators as a sanity check (i.e., X and Y values must agree within their error estimators), and then compute the weighted mean for insertion into the final catalog.

The choice of tessellation scheme is open for discussion. I have not researched the issue, and I don't know the alternatives. HEALPix is nice because (a) big tiles hold all portions of smaller tiles, and (b) they are sqare-ish. The former is nice because it allows for the outer loop over catalog tiles to solve all stars and all sky tiles in it. The latter is nice because the least squares fits are done in X (RA) and Y (Dec) so there are no extra corners or big blank regions that might confuse simple polynomials.

Although not strictly an issue with the astrometric processing, the biggest single problem I have with my simulator is the assigning of detections to objects. Specifically, I use spatial proximity (i.e., choosing the detection closest to the predicted position), and this has many failure modes. It would be really nice to include a test of this part of the LSST pipeline in DC3, just to make sure that the LSST algorithm is a lot more robust than mine.

IV. Contents of Structures (so far)

Exposure Metadata: boresight RA; boresight Dec; MJD of mid exposure.

Sky HEALPix Metadata: central RA; central Dec; epoch of upper culmination.

Reference Catalog: (J2000) RA; Dec; magnitude; RA proper motion; Dec proper motion; parallax; refraction; catalog HEALPix tile; star id; catalog flag.

New Catalog: (J2000) RA; error of RA; Dec; error of Dec; RA proper motion; error of RA proper motion; Dec proper motion; error of Dec proper motion; RA parallax; error of RA parallax; Dec parallax; error of Dec parallax; RA refraction; error of RA refraction; Dec refraction; error of Dec refraction; total weight; number of observations; tangent plane ξ; tangent plane η; standard deviation of ξ fit; standard deviation of η fit; sky HEALPix tile; magnitude; catalog flag.

Raw Observation: pixel X; pixel Y; magnitude; star id; sky HEALPix tile; exposure id; observation flag.

Internal Observation: pixel X; pixel Y; camera X; camera Y; at epoch RA; at epoch Dec; tangent plane ξ; tangent plane η; epoch in years; RA parallax factor; Dec parallax factor; RA refraction; Dec refraction; magnitude; sky HEALPix tile; star id; exposure id; observation flag.

Least Squares Fit (X or Y): observed ξ; observed η; predicted value; weight; residual; epoch; parallax factor; refraction; magnitude; computed value; mean position; error of mean position; index into observation list; index into catalog list; star id; exposure id; sky HEALPix tile; camera ccd; observation flag; catalog flag.

Flag Bits: catalog galaxy; catalog qso; catalog blend; catalog variable; catalog motion; catalog parallax; catalog refraction; observation galaxy; observation blend.