The SDSS Imaging Pipelines

Robert H. Lupton[a], Željko Ivezić[a], Jim Gunn[a] Jill Knapp[a] Michael Strauss[a] Naoki Yasuda[b]

[a]Princeton University Observatory, Princeton, NJ 08544
[b]National Astronomical Observatory of Japan, Tokyo.

ABSTRACT

The SDSS project has taken 5-band data covering approximately 3000 deg^2, or 4Tby of data. This has been processed through a set of image-processing pipelines, and the resulting catalogues of about 100 million objects have been used for a number of scientific projects.

We discuss our software infrastructure, and outline the architecture of the SDSS image processing pipelines. In order to process this volume of data the pipelines have to be robust and reasonably fast; because we have been interested in looking for rare objects, the number of outliers due to deficiencies in the data and bugs in the software must be small. We have found that writing the codes has been one of the harder and more expensive aspects of the entire survey.

Keywords: Optical Surveys, Software, photometry

1. INTRODUCTION

The Sloan Digital Sky Survey (SDSS) is a digital photometric and spectroscopic survey which will cover 10,000 deg^2 of the Celestial Sphere in the North Galactic cap and produce a smaller ($\sim$ 225 deg^2) but much deeper survey in the Southern Galactic hemisphere.[1] The survey sky coverage will result in photometric measurements for about 50 million stars and a similar number of galaxies. About 30% of the Survey is currently finished. The flux densities of detected objects are measured almost simultaneously in five bands[2] (u, g, r, i, and z) with effective wavelengths of 3551 Å, 4686 Å, 6166 Å, 7480 Å, and 8932 Å, 95% complete for point sources to AB limiting magnitudes of 22.0, 22.2, 22.2, 21.3, and 20.5 in the North Galactic cap. Astrometric positions are accurate[3] to about 0.1 arcsec per coordinate (rms) for sources brighter than 20.5^m, and the morphological information from the images allows robust star-galaxy separation[4] to $\sim 21.5^m$. The SDSS is also taking spectra for about 10^6 objects (predominantly galaxies and quasars); we shall not discuss this dataset further in this paper.

2. THE IMAGING PIPELINES AND THEIR EVOLUTION

The software design for the SDSS is reasonably simple. We pass the imaging data through a series of pipelines: First the `SSC`* finds the per-column background levels and detects stars which are later used to model the point-spread function (PSF) and perform astrometry. Next the `PSP` estimates the flat-field and scattered light characteristics of the camera (as a function of time) and models the spatial/temporal structure of the PSF. Then `Astrom` ties the astrometry to the Tycho or UCAC catalogs[3] and estimates the astrometric offsets between the different bands. After that, `Photo` flat fields, finds and removes cosmic-rays, subtracts the sky, finds objects, reconciles detections in different bands, deblends complex objects, and measures the properties of every detection. Finally `nfcalib` photometrically calibrates the data which is then written to a database.

If this processing is simple, why is it an interesting topic for an SPIE meeting? The original processing model was rather different, and was forced to change by reality and the SDSS's scientific requirements. Initially we thought that the data acquisition (`da`) system on the mountain would perform the functions of the `SSC`, and that `astrom` would make use of its outputs. Then the `PSP` would be a simple, fast, piece of code to estimate the global properties of the sky and PSF, followed by `frames` to measure the object's properties. One desirable

E-mail: {rhl,ivezic,gk,jeg,strauss}@astro.princeton.edu, naoki.yasuda@nao.ac.jp

*the etymology of these names is not important feature of this structure is that the only slow step, `frames`, is manifestly parallelizable as disjoint patches of the sky can be processed independently on different CPUs.

The use of the `da` hardware to pre-process the data sounded like a good idea, as it meant that we'd be able to skip a processing step, and with it a pass through every pixel of the imaging data. We were initially forced to abandon it when it transpired that the `da` didn't have enough compute cycles available to perform all the desired calculations; furthermore the `da` runs under a real-time operating system (`vxWorks`) and we were not willing to take the the risk of making extensive modifications to a critical component — failing to write data to tape would have been a catastrophe. Once the `SSC` had been invented to take over some of the `da`'s duties it was easy to allow it to take over others, until at this point the `da`'s processed outputs are used solely for quality assurance (QA) on the mountain.

The SDSS's optical design delivers a 3° field of view with a constant scale (chosen to simplify the electronics required to support the the Time Delay Integrate [TDI] observing strategy), but at the cost of image quality that varies by up to 15% across a single CCD. When we first started to take data in 1998 the reduction codes did *not* take this into account, and we had not fully realised the combination of variable image quality (due to TDI, a function only of CCD column) with temporal variation leads to a 2-dimensional variation of the PSF; the variation is naturally different in the different bands, taken at slightly different times and with different atmospheric and optical PSFs.

The initial SDSS data was sufficiently much better than photographic surveys that the degradation due the neglect of the spatially variable PSF was not of crucial importance, but in retrospect it was one of the two limiting factors in the survey's performance (the other being our naïvety about how to flat-field the data). We were able to devise an algorithm to follow these PSF variations,[4] but it required more stars than the `da` was providing — this was one of the drivers for the development of the `SSC`. The calculations needed to model the PSF also meant that the `PSP`'s compute load increased significantly which had an impact upon the resources needed as the `PSP` does not parallelize well.

Once we had a good model of the PSF, we found that other aspects of the processing needed to, or at least could, be improved. For example, the star-galaxy separation algorithm makes explicit use of the PSF as does the centroiding algorithm.

2.1. Efficiency and Memory Usage

When we first designed the SDSS software systems the task was quite daunting. The peak data rate of 20 Gb/hr was large relative to available disks; computer memory was expensive; and CPUs were slow. The core of the SDSS imaging pipelines, `frames`, was therefore written with these limitations in mind.

Fig. 1 shows that processing of a 'field' (40s of data from one of the six dewars, or about 30Mby of data) takes about 150s on a single 1GHz Pentium II processor. The total memory usage is modest by todays standards, and doesn't grow without bound as it is fragmented. A more careful examination of this type of plot shows that about 15% of the time is taken performing routine operations (such as flat fielding and object detection), and the remaining 85% is spent in measuring the properties of the detected objects. When we first commissioned the SDSS telescope, the ratio was closer to 1:1 but as almost all the required algorithmic enhancements were concerned with measured properties rather than with routine CCD processing the balance understandably shifted.

In light of Moore's law it isn't clear that the effort involved in achieving this performance was well spent. In some cases the code could have been simplified if performance weren't an issue, although it is hard to quantify the gains.

2.2. Designing for Robustness; or, (some of) the Science is in the Tails

We have processed about 1.2 T-pixels of imaging data (3000 deg^2 in 5 bands); while processing this volume of data all the it-can't-happen-to-me singular cases do. We prepared for this in two ways: by the liberal use of assertions in the code and by allowing the pipeline to give up on particular objects providing that it had a way of communicating its decision to the end user. Of the 36 bug reports filed against `frames` in the last year,

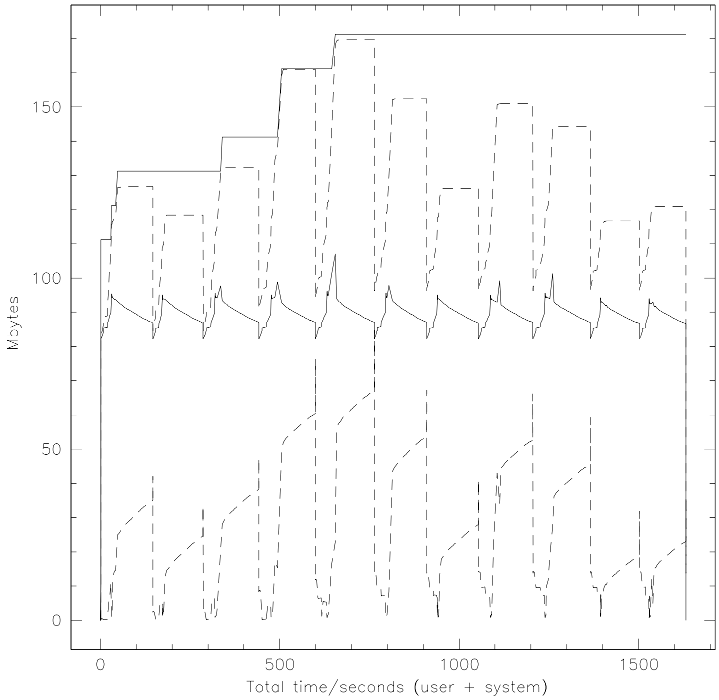

Figure 1. The memory usage while `frames` processed run 756, camCol 2, fields 700—710. The x-axis shows elapsed time on a 1GHz Linux box; the y-axis shows the total amount of memory consumed. The solid line at ~ 90Mby is the memory that is being used; the solid line that reaches ~ 170Mby is the total memory requested from the system. The two dashed lines show the free memory in the actively managed heap (at the bottom) and the memory currently in use or available. The region between the top dashed and solid lines could be returned to the operating system if desired.

15 were assertion failures; 9 were other ‘technical’ problems (e.g. a `NaN` in an output file); and the rest were scientific problems with the outputs (e.g. cores of bright stars were sometimes called cosmic rays). The pipeline currently uses 60 bits (in each of the 5 bands) to flag exceptions in the processing, and of these all but 2 or 3 have proved useful in practice.

In addition to causing problems to the routine processing of data, these rare conditions also contribute to the non-Gaussian tails of the distributions of measured properties. These show up in two ways; either as occasional outliers, or as occasional fields where e.g. the photometry is poor; these fields always, of course, seem to be the ones that happen to be of particular interest to ones colleagues (e.g. the Draco dwarf galaxy). It is as important to estimate errors correctly as is it is to correctly measure the quantities themselves. An example is given in Fig. 2, where the bottom-right panel shows the distribution of colours of a sample of stars (solid line). The distribution expected solely due to photometric errors is also shown (dashed), and the excess number of objects in the right-hand tail is interpreted in terms of the object’s gravities and metallicities. This signal would be swamped if the errors were even a little larger, or the tails a little fatter.

The SDSS was designed to study the large-scale structure of the Universe; rather surprisingly it has proved capable of answering questions about our Solar System. When we first started to take spectra of quasars, we found that a significant number of candidates had disappeared. Upon investigation, we found that these candidates were in fact asteroids — at opposition main belt asteroids move by about 2 arcsec in the 5 minutes between out g and r observations, and therefore look like a very exciting red object next to a very exciting green one. It was not too hard to generalise the deblender to include moving objects, and the missing quasars were

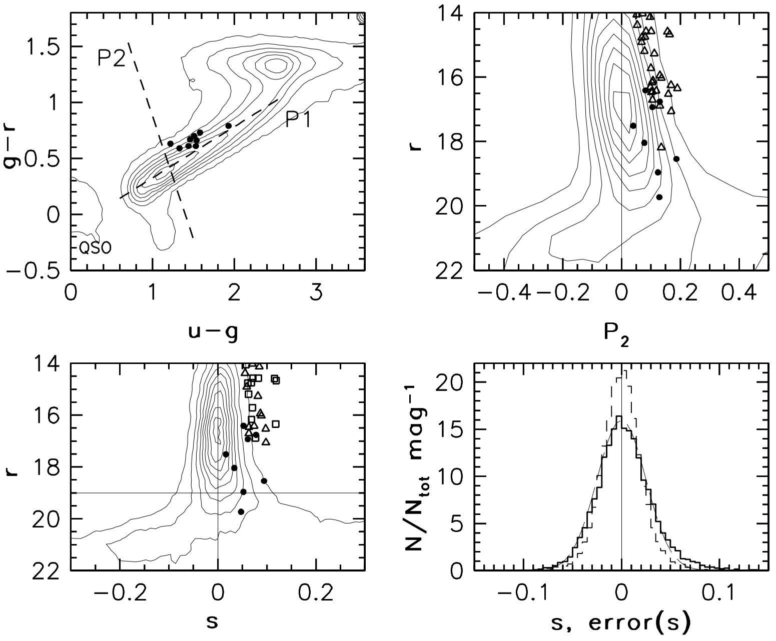

Figure 2. Using SDSS u-g-r colours to separate dwarf from giant stars.[5] The top left panel shows a 2-colour diagram, with the directions corresponding to the two colours P_1 and P_2 indicated. The solid points are giants discovered by the Spaghetti survey. The top-right panel shows the P_2 colour plotted as a function of magnitude; the points at $r \sim 18$ are the Spaghetti giants, while the points at $r \sim 16$ are candidate giants, chosen by their s colour (i.e. the P_2 colour corrected for the slope evident in the figure). The bottom left panel shows the results of spectroscopic followup; squares are confirmed giants, and triangles are dwarfs. Despite the relative rarity of giant stars (less than 5% of the sample), about 50% of the candidates are real. The bottom right hand panel shows the distribution of s (the solid line) together with a histogram of the errors in s. The excess tail of giant candidates on the right-hand side is clearly seen.

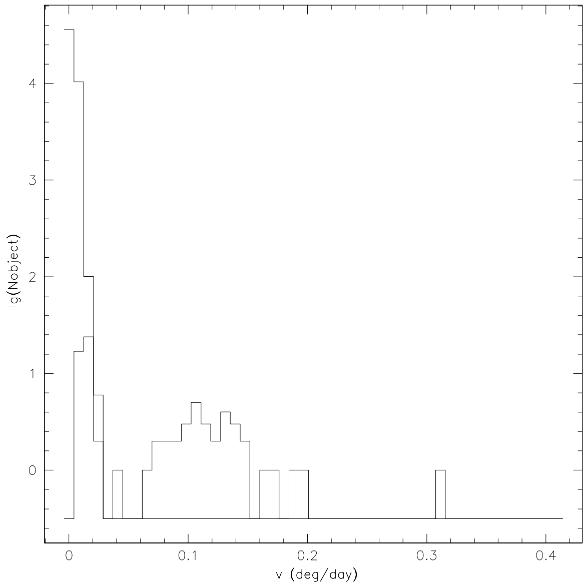

Figure 3. Distribution of velocities for 46480 objects with good velocities (e.g. detections in gri). All the objects with velocities > 0.04 deg/day are said to be moving at $> 4\sigma$, and almost are real.

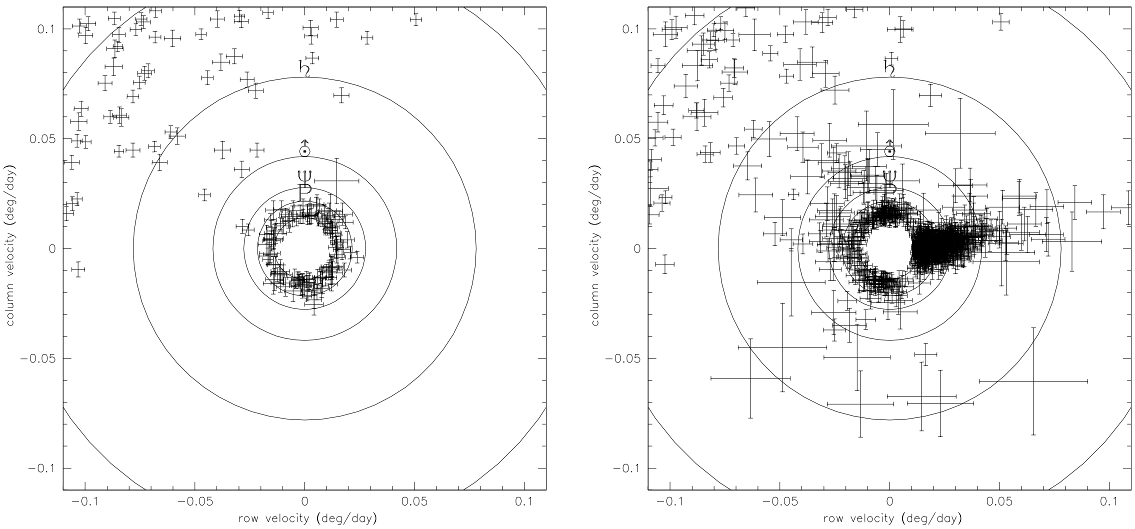

Figure 4. The velocities of the objects in Fig. 3 translated to distances assuming that they are in circular orbits at opposition. Points at the upper left of the figure are main-belt asteroids **These data were not in fact taken at opposition, so these distances are for illustration only**; the statistical properties are correct, but the apparent detection of objects beyond Saturn is not. Inspection of the data shows that objects near the orbit of "Neptune" are fictitious, but that three of four objects detected between "Uranus" and "Neptune" (but not shown here) are real. The plot on the right is based on older reductions of the same data; the improvement is readily apparent.

replaced by objects with a measured velocity. More interestingly, it was soon realised that these velocities could be used to study the asteroid population.[6,7] A back-of-the-envelope calculation indicates that SDSS should be sensitive to objects at distances up to 20–40 a.u., but realising this goal requires that there be only a very small contamination from bad velocities. The distribution measured velocities for a set of point sources is given in Fig. 3; translated into distances from the Sun these data are replotted in Fig. 4. The SDSS is apparently sensitive to objects with orbits between Saturn and Uranus; with our sensitivity limit, we would not only expect to see one Trans-Neptunian Object (TNO) per few-hundred square degrees even if we were able to improve our astrometry by another factor of two.

Another contaminant of the SDSS quasar sample is cosmic rays. About 90% of the sky is imaged only once in each band, so the usual method of cosmic ray rejection by comparing two or more images is not applicable. We were able to write code that found about 99.9% of the cosmic rays, but in the 1550 deg^2 that were searched for high-z quasars,[8] there are 15 million z-band detections above 6σ, and 6.5 million cosmic ray hits on the z-band CCDs; 0.1% of 6.5 million is 6500. Rather than attempt to tune the cosmic ray algorithm to find these objects (with the attendant risk of mistaking the cores of 0.1% of bright stars as cosmic rays), we chose to set a bit (`MAYBE_CR`) for suspicious objects — an example of the approach of communicating possible problems to the end user.

3. RANDOM SOFTWARE LESSONS THAT I LEARNED FROM SDSS

The *I* in this section means RHL. It hardly needs to be emphasized that these are his opinions which, for all he knows, may not be those of any other person, living or dead.

- Neither Science nor Software can be run as a democracy. Not all participants are equal, and it's folly to pretend that they are. This is not to say that the most senior (or smartest) individual should simply lay down the law.

- Standard software practices are necessary. This includes Source code control at the level of files (e.g. cvs); Code control at the level of releases that can be reconstructed, along with their dependent products (but probably not at the level of being able to recover old versions of the O/S, compilers, etc.); Enforced rules about tests associated with each new feature; Tools, preferably integrated with the code/release control system, to track problems and feature requests (we use Gnats). An insistence on adhering to standards; e.g. coding to ISO C89 and Posix 1003.1.

- Distribute Data and Information Freely

- Avoid single points of failure. OK, so this is totally obvious, but there are subtler aspects. If one person is allowed to become essential it implies that it's proved impossible to find someone else who could fill their role. In consequence, if they are on the critical path, and problems arise, it's hard to add resources to solve the problem. Another problem is that if someone with essential knowledge isn't very good, then an essential component of your system isn't going to work very well.

- Find some way to reward people working on the project. In SDSS we did this by promising them early access to the data via a proprietary period. Not only is this impossible for publically funded projects, but it doesn't really work very well. One problem is that the promise of data in the distant future doesn't help a post-doc much; another is that the community (at least in the US) doesn't value work on the technical aspects of a large project. I don't think that the solution 'Hire Professional Programmers' is viable (although hiring a significant number of *competent* software professionals is a good idea. My experience has been that we cannot afford to hire good programmers).

- Strive to ensure that the software takes full advantage of the hardware, even at the beginning of a project. This is partly a matter of principle, and partly because if you don't push to the instrumental limit you don't really know if things are working as well as they should. There is some tension in achieving this, and you need someone to keep the schedule.

ACKNOWLEDGMENTS

The Sloan Digital Sky Survey (SDSS) is a joint project of The University of Chicago, Fermilab, the Institute for Advanced Study, the Japan Participation Group, The Johns Hopkins University, the Max-Planck-Institute for Astronomy, the Max-Planck-Institute for Astrophysics, New Mexico State University, Princeton University, the United States Naval Observatory, and the University of Washington. Apache Point Observatory, site of the SDSS telescopes, is operated by the Astrophysical Research Consortium (ARC).

Funding for the project has been provided by the Alfred P. Sloan Foundation, the SDSS member institutions, the National Aeronautics and Space Administration, the National Science Foundation, the U.S. Department of Energy, Monbusho, and the Max Planck Society. The SDSS Web site is http://www.sdss.org/.

We thank Princeton University and the Fifth Author for generous financial support of the first two authors' research.

REFERENCES

1. D. York *et al.*, "The Sloan Digital Sky Survey: Technical summary," *Astronomical Journal* **120**, pp. 1579–1588, 2000.
2. M. Fukugita, T. Ichikawa, *et al.*, "The Sloan Digital Sky Survey photometric system," *Astronomical Journal* **111**, pp. 1748–1756, 1996.
3. J. R. Pier, J. A. Munn, *et al.*, "Astrometric calibration of the Sloan Digital Sky Survey," *Astronomical Journal* **in press**, 2002.
4. R. Lupton, J. Gunn, *et al.*, "The sdss imaging pipelines," in *Astronomical Data Analysis Software and Systems X*, F. R. H. Jr., F. A. Primini, and H. E. Payne, eds., *ASP Conference Proceedings* **238**, pp. 269–272, 2001.

5. A. Helmi, Željko Ivezić, *et al.*, "Selection of metal-poor giant stars using the Sloan Digital Sky Survey photometric system," *Astronomical Journal* **In Preparation**, 2002.
6. Ž. Ivezić *et al.*, "Solar system objects observed in the Sloan Digital Sky Survey commissioning data," *Astronomical Journal* **122**, pp. 2749–2784, 2001.
7. Ž. Ivezić *et al.*, "Color confirmation of asteroid families," *Astronomical Journal* **submitted**, 2002.
8. X. Fan, V. K. Narayanan, *et al.*, "A survey of z $>$ 5.8 quasars in the Sloan Digital Sky Survey i: Discovery of three new quasars and the spatial density of luminous quasars at z$\sim$6," *Astronomical Journal* **122**, p. 2833, 2001.