Proceedings of the IADIS Multiconference on Computer Science and Information Systems, p. 83-90, IADIS Press, ISBN 978-972-8924-63-8, held in Amsterdam, 22.-27. July 2008

ASTROINFORMATICS – A STUDY ABOUT CONSTRAINTS AND REQUIREMENTS FOR NEXT GENERATION ASTRONOMICAL IMAGE PROCESSING

T. Bauer
bluewater multimedia concepts
D-53757 Sankt Augustin, Germany
tbauer@astro.bwmc.net

ABSTRACT

A concept of a new software architecture for astronomical image processing is presented. Current approaches of astronomical data extraction are explained: photometry, astrometry and high-resolution imaging. The methods seem incommensurate with each other and suffer from systematic errors caused by known issues. Hints are given, where and how to consolidate these algorithms into a more sophisticated approach to improve data retrieval from images. A detailed discussion of user experience and possible expectations is presented. The analysis yielded current astronomical image processing software based on a shell oriented task collection designed and implemented since the 80ies. From a historical perspective, requirements for any next generation software design can be identified as backward compatibility, platform independence, extensibility, a more intuitive user experience, documentation issues, and long-term reliability. A smooth software transition is proposed. Object-oriented and XML driven approaches may help to combine the existing software packages and future extensions within a common user interface. A unified, formal task description language for image processing is proposed to support generality.

KEYWORDS

Astronomical image processing, software architecture

1. INTRODUCTION

Astronomy science has a long-term tradition using and stepping over many different technologies from visual observations, manual drawings, the era of the photographic plate until the advent of modern imaging detectors like photon-counting devices, CCD and CMOS imaging sensors. The main procedures of current image data processing in the optical astronomy are: *Stellar photometry*, the measurement of irradiation intensities (especially: light), mainly expressed in logarithmic scales, *astrometry*, the measurement of positions and proper motions of celestial bodies, and *high-resolution imaging*, the extraction of image details.

Processing optical images and data digitally started in the 1970ies based on digitized photographic plates and the arrival of electronic photo detectors. A new revolution has begun in the 1980ies with the application of silicon image detectors. *Mackay (1986)* described the use of CCD detectors in astronomy. It is still a large field of interest. One gets the impression that development and application of algorithms is heavily affected by experimental development with every new telescope or observatory, and is finished with a set of software tools which will work as designed in anyway not regarding the user experience. *Brunner et al. 2001* report that imaging detectors nowadays record petabytes of astronomical images. Sources are ground-based and satellite observatories within the complete range of wavelengths. The many amateurs involved in astrophotography and astronomy cannot be ignored, but one hardly will be able to estimate the amount of their contribution. Objects of interest are the exploration of billions of stars, stellar clusters, galaxies and the distributed interstellar matter. With this complex scientific background and data amount it is no wonder, that informatics should play an important role, but only plays an ancillary role. There are mainly two accepted standard digital image processing packages: MIDAS *(Banse et al. 1983)* and IRAF *(Tody, 1986)*. They both combine a large set of algorithms, and standard image processing routines within a collection of tools, that have been implemented and integrated. These packages are provided as open source to the astronomical community.

2. ASTRONOMY: A SCIENCE BEYOND THE LIMITS

„Astronomers never seem to want to do anything easy" (Stetson, 1987).

What Stetson meant with his comment was one of the first trials in history to provide a robust way of automated pattern recognition and feature extraction from images. With shell-based scripting and the invention of tasks like DAOPHOT *(Stetson, 1987)*, automated data reduction was introduced within the professional astronomy. *Schechter et al. (1993)* presented a competing tool: DoPHOT. *Diolaiti et al. (1999)* and others using their own implementations and aproaches for high precision extraction of photometric data. In the meantime people are discussing technical issues seen from obtained results of stellar photometry and astrometry. Some problems within the data reduction occurred as systematic abberrations and have been discussed by *Anderson & King (2000), Mighell (2005)*, *Stubbs & Tonry (2006)* and others. These problems are mainly identified to rely on physical properties of the image formation process and algorithmic details within the data reduction pipeline. In other terms, attributes of the same objects of interest, which physically depend on each other, need different approaches with current software tools to achieve high quality measurements. However, the process does not seem to be understood completely. The expected outcome so far are high precision photometric and astrometric data interfered by systematic errors. Position measurement is a secondary outcome from photometric data extraction of the identified stars in stellar fields. Positions and proper motions are important properties to understand the evolution of star clusters and galaxies. The photometric packages have been identified to produce systematic errors within subpixel range, which restricts the use of these tools.

In principle the photometric packages fit stellar profiles in such a way, that they will be either modeled theoretically, or derived from the observed images itself. This will yield intensity information about the overall collected light obtained from the observed star (photometric property, collected light), as well as the information about the (x/y) coordinate where the profile has been detected (astrometric property; position of star). There are several approaches for the model of stellar profiles (in other terms: point spread functions), as they will be seen on an astronomical image. *Stetson et al. (1990)* give examples, like Gaussian star profiles or Moffat functions. The systematic aberrations of the position measurement are a consequence of fitting a continuous stellar profile on a discrete, sampled image. Detecting positions with a precision of subpixel range, effects of discretization show up from the spatial integration of the point spread function on a finite (rectangular) pixel area. Although *Anderson & King (2000)* noted this as a surprise, the effects have been seen earlier with CCD detectors. The author obtained the effect from any degree of sampling between undersampled, well-sampled and oversampled images (internal discussion and presentation at the Sternwarte of the University of Bonn, 1993-1997 unpublished). Fitting a stellar profile with a continuous function resulted in an approximately sinuoidal error curve. The error defines the displacement between the observed and calculated (O-C from simulation), or suggested real position (O-R from observation) of a detected the stellar profile. *Anderson & King (2000)* later observed the effect from real and undersampled images taken with the Hubble Space Telescope planetary camera. The error depends on the S/N of the star found on the detector, which causes an additional, statistical error. In practice a precision within the range of 1/100 of a pixel and better seems possible to obtain star coordinates.

In short terms, measuring precise star intensities is linked to the question where to find the observed star profile, and vice versa. The current approach of using separate tasks to extract both information does not seem to be an adequate solution to the problem. This might imply, to think about theory and practice of photometric and astrometric measurements and a possible new implementation in the sense of a unified feature extraction process.

Speaking about subpixel range, I may consider high angular image resolution. Point spread can be a result of atmospheric turbulence, aberrations of the telescope optics or telescope guiding errors, or all together. The latter case, having all effects, is the practical situation obtained from ground-based astronomical observations. The possibility of high angular resolution within subpixel range and the dependency from the signal-to-noise ratio has been discussed earlier by *Sementilli et al. (1993)*. Practical applications to achieve high angular resolution below the dimension of a pixel followed. This can be understood as a kind of undersampling in the sense of smaller structures found, than the pixel dimension. *Fruchter & Hook (2002)* invented the Drizzle algorithm to achieve smoother resolved pixel grids. The Drizzle method ignores the possibility of image deconvolution to improve separation of details. Again, the question of PSF retrieval leaves the same open questions, like how to precisely detect the positions and thus the shift between the images used for addition. *Willett et al. (2004)* proposed a wavelet based approach, which achieves high angular resolution, and presents details smaller than the detector would be able to resolve. While knowledge about super-resolution is not only of theoretical interest, new research methods have been invented to examine the properties of a real detector. *Piterman & Ninkov (2002)* published a method for the calibration of the spatial light response of CCD imagers within subpixel range. Here it is considered, that a real image sensor will not provide a uniform sensitivity of light within the small area of a single pixel as part of the silicon detector.

Astronomy certainly is one of the most impressive sciences, investigating the frontiers of the universe. Astronomy left over technical limits as seen from classical education in image processing. This survey shows, that astronomical image processing has not defined theoretical limits in the sense of having found a natural barrier with measuring light intensities or any limit of applicable image resolution. New approaches and the unification of the isolated methods of measuring intensities, positions and obtaining high resolution within subpixel range, might help to understand and move the limits of image processing beyond the experience of having seen many light on a single pixel with a discrete (x/y) coordinate.

3. ANALYSIS OF CURRENT TECHNOLOGY

Wells et al. (1981) defined the image format standards early: FITS, a flexible image transport system. It is a simple raw file format with metadata support and a low degree of complexity. FITS forms a standard and supports data arrays, like single images and sequences of frames. Long-term reliability in the sense of usability, simplicity of design, and simple interpretation of information content are consequences.

Software in astronomy science is of moderate complexity. IRAF and MIDAS have shell oriented software architectures. Separate task fragments are split into a collection of mini applications. The single applications range from simple image operators (e.g. add or subtract images), to complex algorithms like DAOPHOT which is included with both packages. Image I/O is provided either by simple file access, or piped streams. The user interface is a simple command line execution with a specific set of options for the respective applications. Ports existed and still exist for desktop PCs, mainframes and supercomputers. Examples are UNIX systems like Sun OS, Solaris, DEC Ultrix, HP/UX, or IBM/AUX. IRAF also supports vector computers and supercomputers, like CONVEX. Other platforms found with MIDAS are VAX/VMS or OpenVMS. Both packages now support Linux, Windows and Mac OS X. Windows support uses a Cygwin/ X11 environment in both cases to define the shell environment for the architecture. In principle, handling and application design was left untouched, except addition and improvement of algorithms. Both packages support primitive display software for image display and drawings. One clearly observes, that professional, astronomical software survived competing technologies and the evolution of operating systems and hardware with only some marginal refinement.

Modern operating systems and their applications rely on multiprocessing and multithreaded software architectures providing standardized graphical user interfaces. In contrast, IRAF and MIDAS rely on a command-line and shell based user interface with a little help from a handful of separate applications for displaying images, plotted curves, and graphics. Humans used to work within modern graphical environments not only have to understand astronomical data analysis, complex image processing techniques and algorithms. They also have a time travel back into the last century and to learn the use of software where shell oriented programs formed the user interface. Of course, one observes, that this architecture offers a certain degree of flexibility: the user is able to *(a) create and edit*, *(b) replay*, and *(c) compare* any *(d) batch process* to *(e) to formulate a problem solution* for image reduction steps in any way. The edited batch jobs can be treated as *a machine and human readable form of a problem solution* and they shall be *kept seriously for documentation purposes (f)*. These points will be referred later.

The ongoing development of professional grade CCD detectors and low-budget cameras later introduced modern GUI oriented approaches. One of the first GUI applications mentioned in literature was ArgusPro *(Bauer, 1992)*. The software was developed for telescope guiding and image processing. Applications published were the photometry of minor planets *(Magnusson et al., 1996)* and specific instrument designs, like the high-resolution imaging system SPICA, a speckle interferometry camera *(Weghorn et al. 1995* and *Bauer et al. 1996)*. *Berry & Burnell (2006)* provide their own contribution: AIP4WIN, a closed source astronomical software package supporting a GUI for Windows. There are certainly more software packages programs, which will not be discussed in detail within this paper. These GUI driven astronomical software lead to an ongoing revolution of astrophotography with overwhelming results within the last 15 years. However, GUI oriented architectures and their success still are restricted on the side of amateur astronomy and sometimes preferred by individual professionals. At this point, I cannot find any confirmation, that astronomers do not want to do anything easy, as things are kept simple for good reasons. My personal opinion is: nobody should even think about, nor should anybody really change this comfortable situation to gain much complexity as dead freight in the software development process. It seems, preservation of simplicity of design might be a challenge.

With the advent of modern digital SLR cameras for general purpose, there were new image file formats introduced from the amateur side, like TIFF *(Adobe, 1992)*. These cameras also provide several raw file formats, which gained focus in astronomy. Astronomical literature indicates at least, these cameras may also awake professional interests *(Hollow, 2000)*. First results from a separate analysis indicate, that rumors about raw file formats started a useless public discussion about proprietary standards. Instead, TIFF derivatives have been found with publicly available documentation *(Bauer, 2008)*.

4. USER EXPERIENCE AND EXPECTATIONS

Astronomical software will and obviously must survive decades of technical evolution in any way. I shall keep in mind, that long-term reliability should be one of our most important requirements in the software development process. The techniques of data extraction and their astrophysical interpretation have been developed and applied over centuries, and form the standards in statistical analysis, interpretation and comparability of results. These are common with the many different technological approaches used to be applied in the past and the present.

The art of programming is not a typical part of education in astronomy science. From the astronomical literature a minority of astronomers performs basic research on image processing, feature and data extraction. These astronomers accept not having any tool for their special purpose. From evaluated literature about observations of open stellar clusters, certainly the most referred tool, DAOPHOT, is an accepted standard in astronomy. As seen from earlier discussion (internal discussion and presentation at the Sternwarte of the University of Bonn in the years from 1993 until 1997; unpublished), astronomers told they struggle with DAOPHOT. The tasks are shell oriented. Reduction steps are invisible in the background. Parameters are hard to understand without in deep knowledge about algorithms. Properties and their dependencies have been discussed and systematic errors are well documented. An interrogation might help to clarify, whether such statements are valid. The standard process of the complete image reduction pipeline is well-defined, too. *Stuart (2004)* and *Howell (2006)* present and explain image reduction steps, and provide a description of several programs supporting a GUI. *Berry & Burnell (2006)* also present standard image processing procedures in detail and presented API4WIN. *Geyer (2007)*, a retired professional astronomer with a long-term experience in astronomy, reported he is satisfied with AIP4WIN. Explaining the use of the software, he noted missing support of reproducibility of the many reduction steps to be invoked manually. Measuring the properties of a single star within a frame, and if he did a mistake, he was forced to invoke many of the steps for a second time. He avoided use of IRAF or MIDAS for some reason. It shall be noted, that amateurs also use applications like Photoshop and Excel and a minority uses MIDAS or IRAF. Automation of the standard image reduction pipeline are not a real topic with GUI applications. Astronomers have to decide whether they use a tool with a graphical user interface or write scripts for blindly automated tasks with IRAF or MIDAS.

5. REQUIREMENTS AND ACCEPTANCE

I shall consider intuitive, GUI oriented software approaches as a preferred technology, where it is provided. On the other hand, there are serious requirements regarding issues on documentation and reproducibility of reduction steps, and verification of results. Software transition, platform independence, and long-term knowledge and data archiving are important topics. Complexity of use shall be improved while preserving flexibility and power of features. Requirements which seem to be hard to fulfill at the same time.

The above mentioned user interactions (a), (b), (c), (d), (e), and documentation issues (f), in the sense of a human and machine readable form a batch job which defines a requirement to be kept with a new software architecture. The modern astronomer is used to find his tools wherever he works on any platform on any observatory around the world. A platform independent software architecture consequently is claimed from the user experience. Human and machine readable batch job formulation shall provide a certain degree of technical robustness and long-term readability.

As it comes to robustness, perhaps, one has to clarify, what robustness means to the user itself. Robustness comes with different aspects, which may define further expectations and requirements. As seen from practical work in astronomy and literature, situations can be identified, where astronomers run into different types of problems. These are found as mistakes during observations and/or data analysis. Sources might be bad whether conditions, detector and/or telescope malfunction, wrong use, choose of wrong parameters with any image processing task, or abuse of tasks, like intensive image post-processing before the data analysis. The typical user shall be assumed as partially overstrained with in deep knowledge about so called well-known computing methods. There might be a future hope of any form of computer assistance to define task descriptions properly, and to support methods based on machine decision. It might be helpful to prove stable concepts of image analysis, if and how to decide about image processing steps automatically based on the collected material itself. Example: Are there regular patterns detected with dark frames? If yes, then they can be detected with Fourier analysis, finally removed automatically, where applicable.

Some of the image processing techniques, strongly break with conventional understanding of the image formation process. If we would like to improve algorithms, as hints found in Chapter 2 might indicate, I may argue, that astronomers do not seem to accept new technology, because it is state of the art. As seen from the competing approaches with photometric analysis, not every technique is accepted by the community. Even if it gives advantages in the sense of precision of measures, it might even be ignored. Maybe, this is caused by a subliminal pain not having understood details of the process. Maybe, it seems sufficient to have any tool, which works anyway to obtain any expected outcome with any precision. As found with DAOPHOT, the first published attempt may wins the competition and establishes over more than 20 years. An interrogation might help to find out, when and in how far a new approach of software architecture and algorithms will be accepted by the whole community. This might influence the adoption of such an approach.

6. A SMOOTH SOFTWARE TRANSITION SCENARIO

From the discussion above we are confronted with a complex situation. The survey found revolutionary approaches of image processing, which break with conventions and limits of understanding, technology and science. This is a matter of course and a natural consequence of how astronomy and astrophysics and the community is used to work within their experimental environment. Aspects of the current situation at least indicate, if not require, new approaches for modern software architectures. On the other hand, astronomers form a conservative user group. This will not mean an award to the people. However, the survey yielded standards and technology, established over decades and left untouched in principle. My personal intuition indicates to be very careful about proposing an approach of a completely new environment and software architecture quickly. A way out might be a smooth transition towards the future of astronomical image processing. Further integration and maintenance of existing architectures shall be a preferred way of choice, while modern methods may be integrated in a different kind of programming. The ability to compare new and old technologies from the image processing results at least has to be kept for documentation purposes.

7. PROPOSED SOFTWARE ARCHITECTURE

Ames et al. (2000) introduced XML and Java within astronomy for instrument control. With the astronomical digital image library (ADIL) *Guillaume & Plante (2001)* presented a method for image metadata, database storage and retrieval based on XML.

XML may also support the image processing stage, to hide the real processing behind the scenes of a graphical user interface without loosing transparency. This will fulfill our requirements: a description of a solution, and a documentation of working steps to be preserved. Imagine a typical image processing task, where 20 frames are processed from a sequence of selected files. This could be formulated with XML as given in Figure 1. Implementation details are negligible, as it is not important to have a final specification for the discussion. We also benefit from versioning for future software transitions, as versioning comes with the XML specification. From object-oriented aspects, properties and constraints of the object class model within the internal framework architecture can be identified. Inheritance and associations between tasks, properties and image sources are a consequence of the XML formulation and the necessary task execution steps. Figure 2 will give a partial impression of a possible object class model in UML notation. The machine may generate XML process descriptions internally from user interactions (GUI), the user may edit these descriptions manually or scripted, based on a well-formed scheme and XML syntax. In a next step the machine will interpret the XML documents, create objects and invoke methods to perform the task processing based on its definitions.

```
<source>
        <filesequence>
                <namepattern>IMG_#####.fits</namepattern>
                <firstindex integer="200" /> <lastindex integer="219" />
        </filesequence>
</source>
<compound_task>
        <compound_task>
                <annotate>Preprocessing stage</annotate>
                <task name="dark_subtraction"> ... </task>
                <task name="flatfield_division"> ... </task>
        </compound_task>
        <task name="DAOPHOT">
                <task_properties>
                        <param name="..." value="..."> ...
                </task_properties>
                ...
```

Figure 1. A possible XML formulation of an image processing task: Even with a basic idea of a process description language in mind, machines and users will be able to edit, read and perform complex image processing tasks

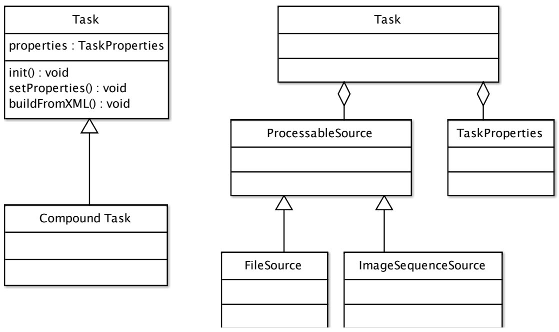

Figure 2. Object-oriented UML analysis of the problem based on the XML formulation: inheritance and associations of tasks, compound tasks, their properties and and relations to image sources to be processed

As platform independence is required, an answer could be the Java platform. However, the astronomical community might associate the Java platform with loosing performance. From historical reasons Java is known as an interpreter language. Astronomical software is written mostly in C, C++ or Fortran code, running natively with a high optimization of code during the compilation stage. Even with modern just-in-time compilers, Java will not have the same performance. Costs of migration have to be considered. Projects are payed on astronomical interests. The art of programming is neither a topic of interest, nor part of education. Benefit of a migration is hard to see. On the other hand, there are reasonable Pro‘s thinking of Java as a basic platform. Object-oriented languages like the Java platform (which might be thought of a complete operating system by itself) provide the possibility to create objects interacting between the internal processing, tasks, and high-level processes. This can be formulated in such a way, that a client-server bridge can be implemented to interact as an communication between the GUI and the processing unit, old and new software. There is one technique, which might help us here: The Java Native Interface *(Liang, 1999)* provides image processing, running natively in separate modules (shared libraries or DLLs). However, existing code must neither be rewritten as a library, nor is this a requirement resulting from such an architecture. The basic Java application programming interfaces provide the execution of separate tasks from an application framework within a separate thread. Thus, the complete collection of MIDAS and IRAF can be reused without modifications. Figure 3 will give an overview about the architectural design. Such a software architecture will support the ongoing development process and existing building block pieces like MIDAS or IRAF. Maintenance of these packages will not change. Adding features and integration do not require in deep knowledge and will not gain further complexity as it is already found in the existing development process. This shall be supported by simplicity of interface design. There remain questions how to integrate such environments, which do not provide Java with their operating system. However, such an architecture will even support a computing network and even a cluster of hardware and processors, like it is proposed by Grid computing approaches.

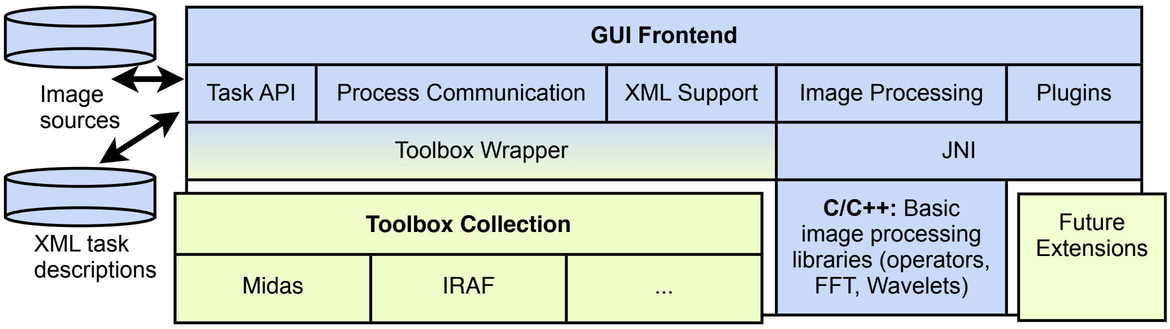

Figure 3. Basic design of a Java image processing environment with core APIs (blue) and external processes (yellow).

Maybe the reader has caught me at this point to asks: why add more complexity, while proposing to keep things simple. However, techniques, discussed so far, still support different existing software packages with a common user interface and extensibility. XML interfaces help to combine and arrange different software packages, which do not seem to be removed from earth. This is not gain of complexity. Instead a missing link was found: a unified, formal process description language based on XML, which provides human and machine readable process formulation of the image processing stage itself.

8. CONCLUSION

A concept for a new software architecture for image processing in astronomy was developed and presented. The proposed object-oriented approach offers an acceptable small solution to the problem and supports flexibility and extensibility. From the analysis important topics have been pointed out: user experience, possible expectations, analysis of the existing technology, requirements and acceptance. However, it is not quite clear, which expectations astronomers really might have about software and user interface. A preference of any GUI frontend is identified, so far. A specific GUI oriented approach shall be designed to grant intuitive handling and flexibility. The analysis yielded existing tools and data structures with a low degree of complexity. Therefore, it seems naturally to provide a conservative approach of simplicity. This is claimed from historical issues: long-term reliability, comparability and documentation of results. As software development is not part of education in astronomy, the architecture shall provide maintenance of software by simple means with basic knowledge of a programming astronomer. The missing link is proposed as a unified, formal task description language for the image processing. This shall be examined and specified. With astronomy science methods are beyond the limits of understanding of the image formation process. Progress in information technology might further improve quality of measurements. However, the survey indicates to carefully think about acceptance of any proposed architecture or new algorithm. Nonetheless, understanding of the image formation process and its analytical counterpart, feature extraction, seems a challenge by itself. The current situation of a having techniques beyond limits of knowledge, may indicate a good starting point for a related research project. A research at least shall include interrogation of astronomers, implementation of the application framework and a final examination by a different user group.

REFERENCES

Adobe Systems Inc., *TIFF Revision 6.0*, final, June 3, 1992

Ames, T. J. et al., 2000. Using XML and Java for Astronomical Instrument Control. *Astronomical Data Analysis Software and Systems IX, ASP Conference Proceedings*, Vol. 216, edited by Nadine Manset, Christian Veillet, and Dennis Crabtree. Astronomical Society of the Pacific, ISBN 1-58381-047-1, p.271

Anderson, J. and King, I. R., 2000. Toward High-Precision Astrometry with WFPC2. I. Deriving an Accurate Point-Spread Function. *The Publications of the Astronomical Society of the Pacific*, Volume 112, Issue 776, pp. 1360-1382

Banse, K. et al., 1983. MIDAS - ESO's new image processing system. *ESO Messenger (ISSN 0722-6691)*, March 1983, p. 26-28

Bauer, T., 1992. Pattern recognition in astronomy. *Sterne und Weltraum* (ISSN 0039-1263), Vol. 31, No. 12, p. 774-778

Bauer, T. et al., 1996. The young star cluster R64 in the OB association LH9 resolved with ground-based CCD observations. *Astronomy and Astrophysics*, Vol.305, p.135

Bauer, T., 2008. The RAW Image File Format Problem - Applications of Digital SLR Cameras in Astronomy and Science. *Proc. of the 4. Jahrestagung für Informationstechnik und Informatik der Berufsakademie in Baden-Württemberg*, ISSN 1614-2519, H. Weghorn (ed.), Stuttgart, Germany

Berry, R. and Burnell, J., 2006. *The Handbook of Astronomical Image Processing.* 2nd. english ed., Wilmann-Bell, Inc., Richmond, VA, USA

Brunner, R. J. et al., 2001. *Handbook of massive data sets.* Kluwer Academic Publishers Norwell, MA, USA

Diolaiti, E. et al., 1999. An Algorithm for Crowded Stellar Fields Analysis. *ESO Conference and Workshop Proceedings*, vol. 56, Edited by Domenico Bonaccini, p.175

Fruchter, A. and Hook, R. N., 1997. Novel image reconstruction method applied to deep Hubble space telescope images. *Proc. SPIE,* Vol. 3164, p. 120-125, Applications of Digital Image Processing XX, Andrew G. Tescher; Ed.

Geyer, E. H., 2007. Private personal and email communication.

Guillaume, D. R. and Plante, R., 2001. Declarative Metadata Processing with XML and Java. *Astronomical Data Analysis Software and Systems X, ASP Conference Proceedings,* Vol. 238. Edited by F. R. Harnden, Jr., Francis A. Primini, and Harry E. Payne. San Francisco: Astronomical Society of the Pacific, ISSN: 1080-7926, p.221

Hollow, 2000. The student as scientist: Secondary student research projects in astronomy. *Publications Astronomical Society of Australia*, Vol. 17, no. 2, p. 162-167

Howell, S. B., 2006. *Handbook of CCD Astronomy.* Cambridge University Press, second edition, Cambridge, UK.

Liang, S., 1999. *The Java Native Interface: Programmer's Guide and Specification*. Addison Wesley, USA.

Mackay, C. G., 1986. Charge-Coupled Devices in Astronomy. *Ann. Rev. Astron. Astrophys*, Vol. 24, pp. 255-283

Magnusson P.; et al., 1996. Photometric Observations and Modeling of Asteroid 1620 Geographos. *Icarus*, Sept. 1996, vol. 123, no. 1, pp. 227-244(18)

Mighell, K. J., 2005. Stellar photometry and astrometry with discrete point spread functions. *Monthly Notices of the Royal Astronomical Society*, Volume 361, Issue 3, pp. 861-878

Piterman, A. and Ninkov, Z., 2002. Subpixel sensitivity maps for a back-illuminated charge-coupled device and the effects of nonuniform response on measurement. *Optical Engineering,* Vol. 41(06), p. 1192-1202, Donald C. O'Shea; Ed.

Schechter, P. L. et al. 1993. DOPHOT, a CCD photometry program: Description and tests. *Astron. Soc. of the Pacific, Publications* (ISSN 0004-6280), Vol. 105, No. 693, p. 1342-1353

Sementilli, P. J. et al., 1993. Analysis of the limit to superresolution in incoherent imaging. *J. Opt. Soc. Am. A: Optics, Image Science, and Vision*, Vol. 10, Issue 11, November 1993, pp.2265-2276

Stetson, P. B., 1987. DAOPHOT - A computer program for crowded-field stellar photometry. *Astron. Soc. of the Pacific, Publications* (ISSN 0004-6280), Vol. 99, March 1987, p. 191-222.

Stetson, P.B. et al., 1990, Future development of the DAOPHOT crowded-field photometry package. *CCDs in astronomy; Proceedings of the Conference, Tucson, AZ, Sept. 6-8, 1989* (A91-45976 19-33). San Francisco, CA, Astronomical Society of the Pacific, p. 289-304

Stuart, A. M., 2006. *CCD Astrophotography*. Springer Science+Business Media, Inc., New York, USA.

Stubbs, C. W. and Tonry, J. L., 2006. Toward 1% Photometry: End-to-End Calibration of Astronomical Telescopes and Detectors. *The Astrophysical Journal,* Vol. 646, Issue 2, pp. 1436-1444

Tody, D., 1986. The IRAF Data Reduction and Analysis System. *Instrumentation in astronomy VI; Proceedings of the Meeting, Tucson, AZ, Mar. 4-8, 1986. Part 2 (A87-36376 15-35). Bellingham, WA, Society of Photo-Optical Instrumentation Engineers, p. 733*

Weghorn, H. et al., 1995, High-resolution imaging at the ESO Danish 1.5 in Period 54. *Astron. Gesellschaft Abstract Ser.*, 11, p.158

Wells, G. et al., 1981. FITS - a Flexible Image Transport System. *Astron. & Astrophys. Supp. Ser.*, Vol. 44, p. 363

Willett, R. et al., 2004. Wavelet-Based Superresolution in Astronomy. *Astronomical Data Analysis Software and Systems (ADASS) XIII, Proceedings of the conference held 12-15 October, 2003 in Strasbourg, France.* Edited by Francois

Proceedings of the IADIS Multiconference on Computer Science and Information Systems, p. 83-90, IADIS Press, ISBN 978-972-8924-63-8, held in Amsterdam, 22.-27. July 2008

Ochsenbein, Mark G. Allen and Daniel Egret. ASP Conference Proceedings, Vol. 314. San Francisco: Astronomical Society of the Pacific, p.107