

# Automating Image Science

John E. Hoot  
 SSC Observatory  
 San Clemente, CA 92672  
 jhoot@ssccorp.com

## Abstract

This paper presents methods and techniques that allow for the automatic calibration, stacking and analysis of astronomical images using royalty free and open source software. With the advent of low cost imaging techniques that produce large and long time sequences of images, data production in small observatories has outpaced the interactive tools employed by many observers. Presented are several scripting based approaches, with examples, on how science data can be extracted from these large data sets with little interaction. Introduction

## 1. Introduction

The past decade has seen the consumer photographic market undergo a transformation from emulsion based films to digital detectors. This revolution was nurtured by professional astronomy and biology's need for better detectors. The advances in charge coupled devices and photo-diode technology arose out of the detector research of the 1970s and 1980s.

The fruits of this effort found its way onto amateur telescopes starting in the early 1990's. The early detectors were small, pricey and the tools to handle the images were rudimentary. With nearly all imaging becoming digital, the variety of sensors available and the configurations in which they are packaged has led to a proliferation of choices for the science imager. At the highest price points of the market are cooled, temperature controlled, high quantum efficiency cameras. At the middle price range are uncooled science cameras, and general purpose digital SLR cameras. At the lowest price point are modified web and video cameras.

The advent of inexpensive uncooled cameras capable of producing science quality images allows anyone interested in pursuing quantitative work to do so from nearly any locale. While this is a boon to science, the consumer market forces that are driving sensor design and production are leading to a glut of data. High pixel count cameras with small dynamic ranges require the processing of ever more image data to yield science results. These factors are increasing the processing requirements so fast that they are overwhelming the GUI image processors that observers are using.

The professionals have adopted automated data reduction pipeline systems to handle the large volume of data produced by queued observing and automated surveys. Amateurs too need to embrace automated

data reduction methods to cope with the volume of raw observational data even a modest program can produce.

This paper explains the forces driving this exponential explosion of data, and presents an overview of how to develop an automated data reduction system using freeware or open source software. The pipeline design and development process shall be explored in the context of a variable cadence survey project underway at the SSC Observatory.

## 2. Science Imager Trends

Almost every commercially available imager is based upon a consumer electronics sensor. While the cost of processed silicon (the basis of most detectors) has remained stable or increased slightly over the years, the density of circuitry per unit area has grown geometrically. In the quest for higher image resolution, these two factors lead chip designers to shrink pixel size. Shrinking pixel size puts higher pixel counts on their detectors without escalating material costs.

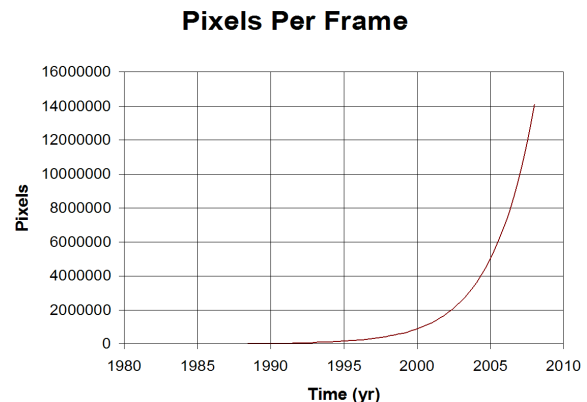
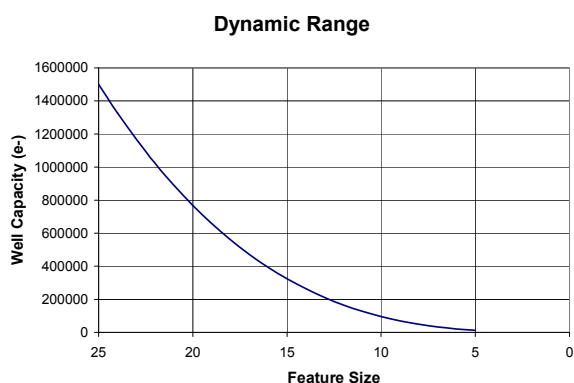


Figure 1. The Escalating Pixel Count Trend

The growth in pixel counts is a corollary to Moore’s law, which roughly states that: “Circuit complexity is doubling approximately every two year’s.

$$\frac{Pixels(yr)}{Frame} \approx e^{0.39 yr}$$

Thus, the number of pixels per imager double about every two years, but the full well capacity of the sensor decreases by a factor of approximately 2.8 every two years since the well is a volume, not an area.



**Figure 2. Well depth as a function of pixel size**

Faced with decreasing well capacity, maintaining a constant signal to noise ratio requires you to take multiple shorter exposures and combine them to compensate for the lower dynamic range. Since signal to noise (S/N) ratios decrease only in rough proportion to the square root of the exposure count, you need to take geometrically more exposures. The rough expression of the number of frames required to maintain a constant S/N is show below:

$$Frames(yr) \approx \left[ \frac{Pixels(yr)}{Frame} \right]^{\frac{3}{2}}$$

The product of the frames produced times the pixel per frame is the measure of the processing power required to reduce raw observations to science data. The result is an escalation of the number of pixels you must process that is advancing at a rate faster than computer power is increasing!

$$Pixels(yr) \approx \left[ \left( e^{0.39 yr} \right)^{\frac{3}{2}} \right]^2 \approx e^{0.78 yr}$$

Fortunately this pixel arms race cannot continue indefinitely. Below some threshold the read noise will render the image too noisy to be commercially viable. Nevertheless, optimally exploiting high pixel count sensors with limited dynamic range is getting to the point where a small observatory can easily produce raw data at a faster rate than it can be reduced.

### 3. Automating Science Images

Science imaging is fundamentally different from imaging for aesthetic purposes. In the latter, many images of the same target are combined to present a pleasing appearance, without too much regard for the preservation of the quantitative representation of physical phenomena. By contrast, each step of image processing for science research is carefully managed to preserve and measure the phenomena under study. Often many similar observations of a given type of target are performed, or many identical observations of the same target are made over time to track and measure change.

It is the very repetitive nature of science observing that lends the procedure to automation. Automating data reduction is not an easy feat. To be a candidate for automation, an observing program must be of sufficient duration, or the data reduction of sufficient complexity that the effort involved in automating the process, is more than compensated by time saved during the program. The other reward for automating a data reduction process is reduction of random procedure errors that corrupt the your results. Once an automated reduction process is running, it will treat all observations identically. The downside of automation is that any systematic error in your process will propagate rapidly through your experiment.

### 4. Process Design

If you are fortunate enough to be engaged in a science program, such as asteroid study, where automated tools are commercially available, automating your work is comparatively easy. Otherwise, you are faced with the task of developing your own process.

The design of an automated data reduction system is an extension of your experimental design. Rather than discuss this in abstract terms, this paper will follow the design and development process for automating the Variable Cadence Survey (VCS) currently in progress at the SSC Observatory.

The VCS is designed to monitor a collection of relatively large fields over a long period of time in the hope of discovering new variable stars, novae, high proper motion objects, supernovas and other

transient events worthy of targeted follow-up. The idea is to gather survey images as part of the normal observatory startup and shutdown procedure for every observing run.

The survey images a set of fields 160 arc minutes, by 106 arc minutes at a resolution of 2.66"/pixel utilizing a modified digital SLR camera. Accessible fields are imaged prior the beginning of any primary observing run, and again at the end of the primary observing run, provided weather did not terminate the primary observing session. The survey fields are imaged at exposure times of 15 seconds, 60 seconds and 240 seconds. A minimum of four frames are taken for each exposure duration. With this technique and equipment 2% photometry can be extracted from Magnitudes 4.0 through 14.0.

With a 10 mega-pixel sensor, the survey produces over 72 images per field per night. Or 2.88 Gigabytes of raw data per field per night. With these numbers in hand, it was clear automation was the only hope the program has for success.

Survey frames for each field are handled as follows:

1. The survey images separated into the R, G and B components FITs files.
2. The images are dark and flat calibrated.
3. The images are all co-registered, stacked and combined to preferentially use the unsaturated data with the highest S/N ratio from among the three different exposures duration at each pixel location.
4. The field's R, G, and B images are then rotated, scaled and aligned to match the coordinates of the master reference frame for that field.
5. Using reference standard stars, differential photometry is extracted for every source identified in the master frame. This data is added to database for that field.
6. All sources in the observed field are identified. These sources are then compared against the master source list for that field and any new sources are noted in an alert file.
7. Astrometry for all sources in the master field are measured and saved to the astrometry database.
8. The photometry database is scanned and stars with statistically significant brightness deviation are noted in the alert file.
9. Periodically, the astrometry database is scanned and objects with statistically significant position deviations are noted in the alert file.

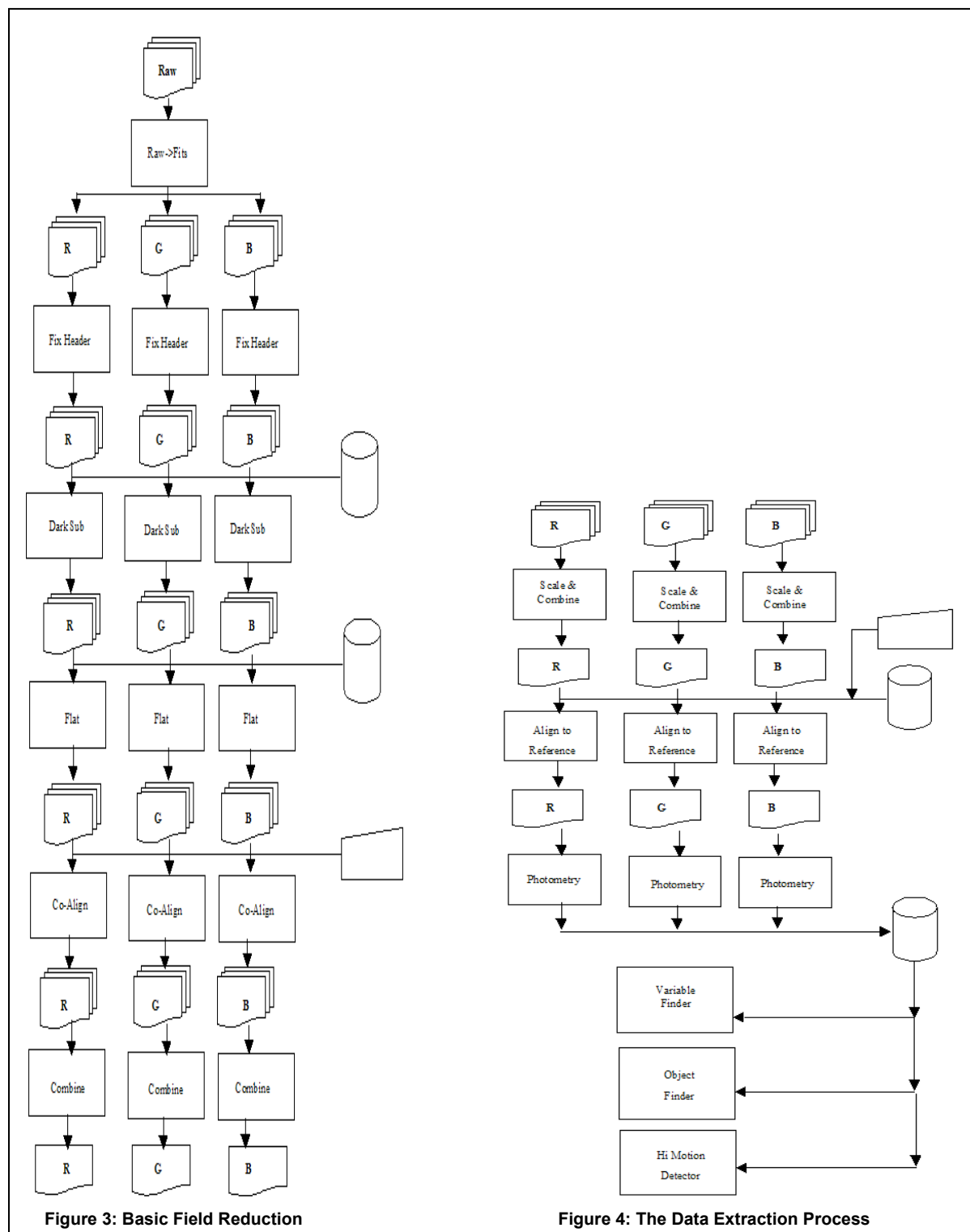
## 5. Survey Data Flow

In developing the data processing pipeline for the VCS, the first step was to graph out the flow of information and the processing steps in a flow chart. Flow charting is an old, but trusty, planning tools used when designing any automation system. At the highest levels it identifies the processing steps and the input and output at each phase of the operation.

The chart in Figure 3 summaries the steps that are taken to process the raw image sequence for a single frame. The boxes with the curved bottoms, represent image files, the rectangular boxes represent processing steps applied to data. The cylinder represents system archival file storage and the box with the slanted top represents user interaction.

The only manual input to the initial reduction process is the identification of the field alignment star in a single frame from the sequence.

Similarly, in the data extraction process, the only manual input required is the identification of two registration stars in a single reduced G frame.



## 6. Tool Selection

The first step in implementing the data reduction process was the selection of the appropriate tools. Given that I have a collection of Windows PCs and

that interactive tools were out of the question, I evaluated the least expensive tools capable of supporting my application. The two most robust image processing freeware or open source systems I located were IRIS and IRAF.

IRIS is powerful freeware astronomical image processing program developed and distributed by Christian Buil. IRIS features a graphical user interface, but this GUI is really an add-on front end to the actual command line driven application.

The IRIS program is constantly being updated and features over 125 separate processing commands from which you can then assemble macros to process your images. In addition to simple macro processing, the command processor can invoke a TCL interpreter so the more complex processing scripts can be handled.

IRIS, though originally written in French, has been translated to English and is supported by an active user community and a Yahoo interest group.

The other alternative considered was the Cygwin port of IRAF. IRAF has long been a staple of professional astronomers. IRAF, short for Image Reduction and Analysis Facility, originated in the 1980's at KPNO, and was subsequently supported and enhanced by the NOAO.

IRAF was originally developed for VMS and Unix systems. It has been ported to Solaris and Linux in the past, but more recently IRAF has been ported to Cygwin, an open source DLL library that runs Linux applications within Windows by converting Linux system calls to their Windows equivalents.

The choice of systems was difficult. IRIS features integrated support for digital SLR cameras. IRIS DSLR support includes handling the many manufacturer's proprietary "RAW" image formats and their conversion to FITs. It also streamlines the calibration of DSLR images by performing dark subtraction and flat fielding in the RAW format prior to splitting the image into its R, G and B components.

IRAF is a far more complex and extensible system than IRIS, but its complexity makes it more difficult to master and manage.

IRAF was ultimately selected for this project because IRIS represents pixels internally as 16 bit signed integers. Our desire to span 10 magnitudes of dynamic range in our data precluded using IRIS without a very complicated and lossy scaling system for our image data. Instead, all VCS image data is referenced and processed using pixels represented using 32 bit floating point numbers.

The selection of IRAF entailed solving the problem of converting DSLR image data from Raw format to FITs. The open source community's DCRAW software solved this problem. This open source program converts most popular digital camera formats to PGM or TIFF format. From earlier work, I had a converter that accepted TIFF and could convert it to FITS, preserving applicable tag values from the TIFF header. It was a straightforward effort to create both DOS and IRAF scripts to convert DSLR images to

FITs. In the freeware tradition, the software and scripts along with a brief usage summary are posted on the SSC Observatory web site.

## 7. Pipeline Implementation

One of the keys to making automated data handling scripts work smoothly is to embed information in the data so that it can be routed, calibrated and processed correctly. The easiest place to put this information is into the file name itself. All operating systems and scripting languages have powerful tools for file manipulation, so you get tremendous benefit from adopting a rational and regular set of conventions for naming files.

In the case of VCS the following naming convention are employed for image frames files:

### Raw Images:

<TARGET>\_<EXPOSURE>\_<GAIN>\_<INDEX>.<TYPE>  
Example: HVCS02\_240\_asa800-0071.CR2

### Dark Calibration Images:

DarkDslr<EXPOSURE><COLOR>.fts  
Example: DarkDslr240R.fts

### Flat Fields

FlatDslr<INSTRUMENT>\_<COLOR>.fts  
Example: FlatDslr30cmR.fts

### Calibrated Images:

<TARGET>\_<EXPOSURE>\_<GAIN>\_<INDEX>\_<COLOR>.fts  
Example: HVCS02\_240\_asa800-0071\_G.FTS

### Combined Science Images:

<TARGET>\_<YY><MM><DD>\_<INDEX>\_<COLOR>.fts  
Example: HVCS02\_080212\_01\_B.fts

### Field Reference Images:

<FIELD>\_Master\_<COLOR>.fts  
Example: HVCS02\_Master\_R.fts

Typically, scripts are constructed to take advantage of these lexical conventions using directory wildcards to create lists of files to be processed. For example the fragment below shows how the conversion from RAW image format to FITs is achieved.

```
procedure rawtofits.cl( fileid)
  string fileid

  begin
  sec (fileid // "*.CR2",> "list01")
    awk ("-f /Scripts/Iraf/rawtofits.awk",
         "list01", > "temp")
  cl <temp
  end
```

In the above example the “//” operator means concatenate, and the “sec” operator is an advanced directory listing command. The result of this first line will collect all RAW DSLR files starting with the

string passed and create a text file listing them one per line. The next line constructs command lines to convert each file in *list01*, and the third line process the commands sequentially.

Figure 5 below contains the highest level pipeline script in the stream. It shows how the lower level scripts are executed to implement the data reduction process. Each day following an observing run, this file is edited with the names of the field from the previous evening’s run and the date entries are updated. The file is then fired off to an IRAF command processor in its own window. Later in the day, the alignment stars are marked, and the process then runs to completion.

```
#####
# HVCS Daily Reduction Pipeline
#
# Use - Edit in todays field id's and date
# cl <pipeline
#####
### create todays darks #####
rawtofits dark
mkdlsrdark dark_015 015
mkdssrdark dark_060 060
mkdlsrdark dark_240 240
del darK_*.fts

### Calibrate todays fields ###
rawtofits HVCS
caldslr HVCS??_015 DarkDslr015 FlatDslr
caldslr HVCS??_060 DarkDslr015 FlatDslr
caldslr HVCS??_240 DarkDslr015 FlatDslr

### Coalign And Combine Todays observations ###
combdslr HVCS02 HVCS02_080212 # edit field and date
combdslr HVCS04 HVCS04_080212
combdslr HVCS08 HVCS08_080212

### Transform to master WCS ###
align HVCS02_080212 /cosmos/projects/HVCS/HVCS02_Master
align HVCS04_080212 /cosmos/projects/HVCS/HVCS04_Master
align HVCS08_080212 /cosmos/projects/HVCS/HVCS08_Master

### Extract Data ###
PhotDslr HVCS02 /cosmos/projects/HVCS/HVCS02_Master 080212
PhotDslr HVCS04 /cosmos/projects/HVCS/HVCS04_Master 080212
PhotDslr HVCS08 /cosmos/projects/HVCS/HVCS08_Master 080212
TransientDslr HVCS02 080212 >>/cosmos/projects/HVCS/Alert_080212.txt
TransientDslr HVCS04 080212 >>/cosmos/projects/HVCS/Alert_080212.txt
TransientDslr HVCS08 080212 >>/cosmos/projects/HVCS/Alert_080212.txt
NewObjDslr HVCS02 080212 >>/cosmos/projects/HVCS/Alert_080212.txt
NewObjDslr HVCS04 080212 >>/cosmos/projects/HVCS/Alert_080212.txt
NewObjDslr HVCS08 080212 >>/cosmos/projects/HVCS/Alert_080212.txt

### Dump Alert File ###
more /cosmos/projects/HVCS/Alert_080212.txt
```

Figure 5: Sample Daily Master Script

On a 2.4 GHz dual core PC, the script above takes about three hours to complete. The process could be accelerated by rewriting custom code to parse the database and perform the correlation analysis to look for transients, and the elimination of known stars in *NewObjDslr*. When the database expands to the point that processing cycles cannot complete within 8 hours this will become a priority.

The complete script library should be available on the SSC Observatory web site by the time of publication.

## 8. Conclusions

The VCS project was conceived about 12 months ago, after the observatory had acquired a

modern DSLR and evaluated its capabilities. The VCS project took a left turn after only a couple of nights of observing. Once the master field frames had been meticulously constructed, sources cataloged and a coordinate reference frame selected, it became clear that the level of manual effort required to process incoming VCS data was beyond the tools I was using and the time available. From this dilemma sprang this foray into data pipeline development. The survey has been on hold for almost a year as the pipeline has been developed, tested and revised. SSC Observatory is looking forward to commencing full survey operations this year. My hope is that it produces science results commensurate with the development efforts expended.

## 9. References

Anderson, C., Seaman, R. (1989). *An Introductory User's Guide to IRAF Scripts*. NOAO, <http://iraf.noao.edu>

Barnes, J., (1993). *A Beginner's Guide to IRAF*, NOAO.

Buil, C. (2007). IRIS

Buil, C. (2008). *IRIS, Version 5.53*  
<http://www.astrosurf.com/buil/us/iris/iris.htm>

Coffin, D. (2007). “DCRAW”,  
<http://www.cybercom.net/~dcoffin/dcrawl/>

Hanisch, R. *et al.* (1994). *STSDAS User's Guide*, Space Telescope Science Institute.

Hoot, J. (2008) “The Variable Cadence Survey Project”, <http://www.sscorp.com/observatory>, SSC.

Hoot, J. (2007). “Photometry With DSLR Cameras”, *Proceeding of the 24th Annual Conference of the Society For Astronomical Sciences*.

Massey, P, Davis, L. (1990). *A User's Guide to Stellar Photometry with IRAF*, NOAO.

Red Hat Inc. (2008) “Cygwin”,  
<http://www.cygwin.com>

Robbins, A.D. (2003). *GAWK: Effective AWK Programming*, Free Software Associates.

Roecklein, A, (2008). “Diamonds From The Rough: Primer on Processing Deep-Sky DSLR Images with IRIS”, <http://astro.ai-software.com>