Software Development for the Reduction of Astronomical Data

Steve Kollmansberger

Mentor: Dr. Glenn Orton

Abstract

Space missions are returning increasingly large sets of data. In order to derive scientific hypotheses and conclusions from this data, astronomers need ways to *reduce* the data to manageable, concise forms. One piece of software which performs this task is the *Data Reduction Manager*. Our work extends the Data Reduction Manager software to increase automation, accuracy, and to provide functionality that better accentuates important and interesting features of astronomical data. In particular, our work provides extended support for research using ground based telescopes as well as spacecraft imaging systems for planetary analysis of Jupiter and Saturn. Our work has brought to light planetary features unnoticed in data as old as fifteen years, while at the same time accelerated and streamlined analysis of newly acquired images.

1 Introduction

The Data Reduction Manager (DRM) is a user-interactive system for reducing astronomical data. The DRM consists of two primary visual interfaces: a file manager for selecting files, and an image tool for viewing and modifying individual files. The DRM also includes numerous modules called from either of these interfaces. Most of the modules acquire simple input through a dialog box or the console and then perform some operation on one or more files.

The DRM is written using the Interactive Data Language (IDL). IDL is a proprietary, vector procedural programming language designed for scientific applications. To this end, IDL supports straightforward image manipulation and display, and includes a substantial library of scientific functions. IDL is extended by an astronomy library of functions from NASA. From a computer science perspective, IDL is a low-abstraction, strongly and dynamically typed procedural language which emphasises array processing. IDL passes parameters by reference and allows named parameters, called keywords. The most recent versions of IDL add object-oriented programming, although the DRM does not yet take advantage of this.

The data reduction process consists of transforming a series of input images into one final image and cylindrical map. A manual, written by Dr. Paul Parrish, describes thirteen steps for data reduction. These steps can roughly be divided into phases: the first step is data acquisition, steps 2 through 6 are multi-image processing, steps 7 and 8 reduce the multiple images into one image, steps 9 through 13 process and annotate the resulting image.

2 Developments

The thirteen step process of data reduction can be quite time consuming. Much of our work involved automating portions of the data reduction so that it could be more performed both faster and more accurately.

2.1 Multi Image Processing

Steps two through six of data reduction consist of subtracting image pairs, selecting and subtracting the sky frames, despiking, and depatterning (removing systematic read noise). Of these, selecting the sky frames is the only part that actually may vary, and sometimes the sky frames are known in advance.

Originally, users had to order, in sequence, each of these operations and perform a variety of detailed selections, for example, to ensure that the A sky image is subtracted only from the A data images and the B sky image is subtracted only from the B data images. This process is now automated, stopping only to prompt the user to select the sky images if the user does not indicate them in advance. This automation provides a user reported 20% decrease in time required.

2.2 Alignment

Aligning positive and negative images on a single image to co-add them is a significant and time consuming step in image processing. First, a vertical slice of the image is taken and graphed (see Figure 1). The user must identify the high and low points on the graph and determine the distance between them. This distance is then entered numerically to co-add the images. If the resultant image is blurry, the user must undo and repeat.

We automated this process to find a co-add fit. The user first places a horizontal divider between the two images. The program then adds the two images together and computes the standard deviation of the result. Since the images are positive and negative, when properly aligned they will cancel out and the standard deviation will minimize. The program shifts the alignment until a minimum standard deviation is found.

2.3 Limb Fitting

Limb fitting is the act of aligning a model of the planet with the actual image. This allows different parts of the image (body, rings, sky) to be automatically identified, and planetary latitude and longitudes correlated to specific image

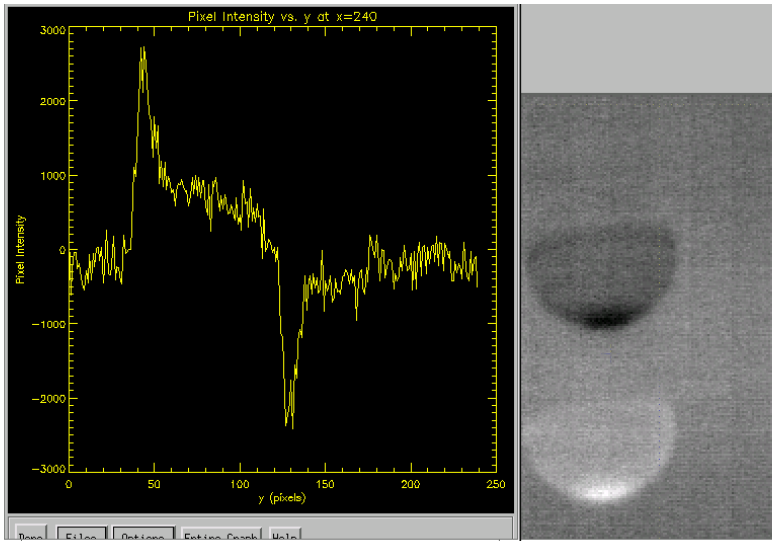

Figure 1: Aligning Images with I vs Y Graph

pixels. This process, like the image alignment, is a tedious manual process of navigating a model until it visually fits well with the planet. To accelerate this process, a previous programmer implemented an auto-alignment guess designed to position the model nearly correct. However, this alignment process was often wrong and needed improvement.

The original fitting algorithm was based on trying to fit a model disc to the image. We created a new algorithm which takes all super-median points (the sub-median points being considered sky or noise) and finds the center-most point by minimizing sum of the squared distance from that point to all super-median points. This algorithm performs very well, fitting the limb correctly about half the time, and requiring only small adjustments another quarter of the time time. A comparison of the original algorithm (left) and our improved fitting algorithm (right) is shown in Figure 2.

2.4 Blacking out Rings

In order to study details of Saturn's atmosphere, researchers project the body of the planet onto a cylindrical map. However, part of the surface is partially obscured by the rings of the planet. These rings can be easily identified before a cylindrical map is created, but it is very difficult to find them afterwards. Therefore, we introduced a feature which blacks out the rings where they pass in front of the planet before a cylindrical map is created. Any places in the cylindrical map which would be influenced by the rings is then rendered black.

The ring black out functions automatically, using the information derived

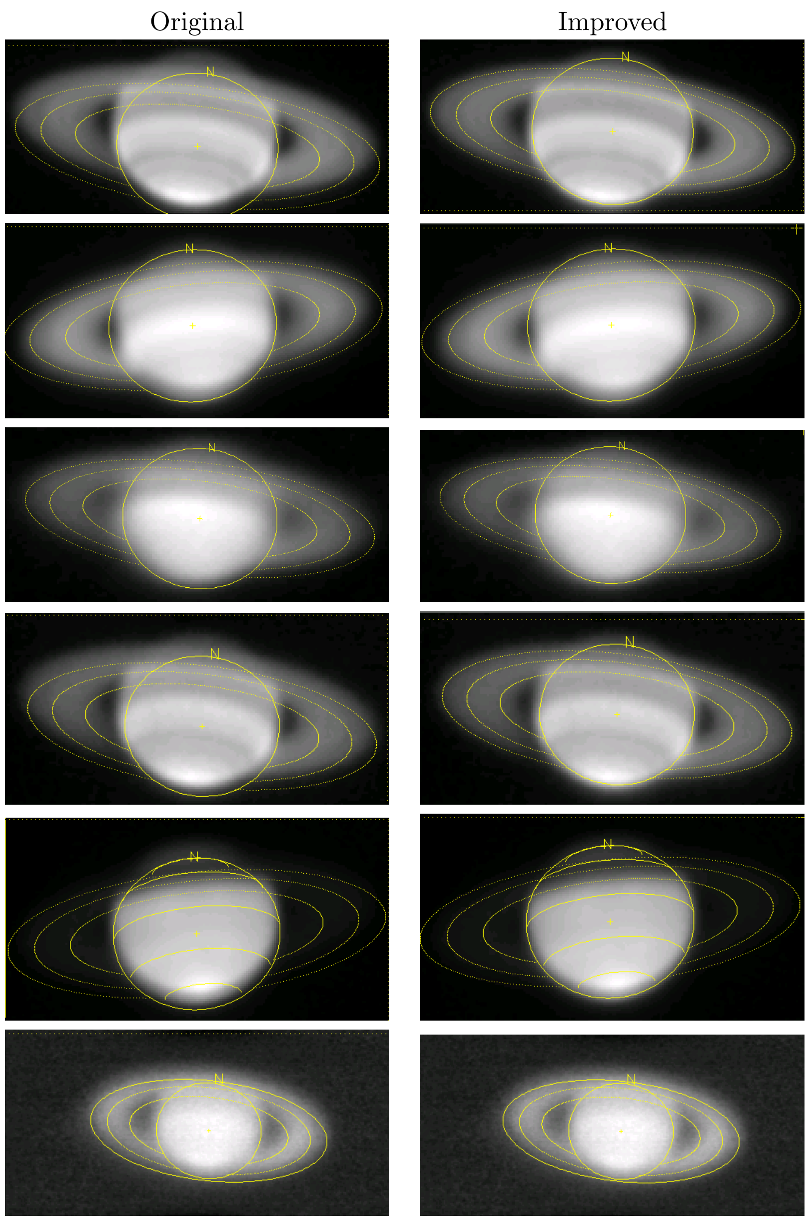

Figure 2: Examples of Auto-Alignment Guess

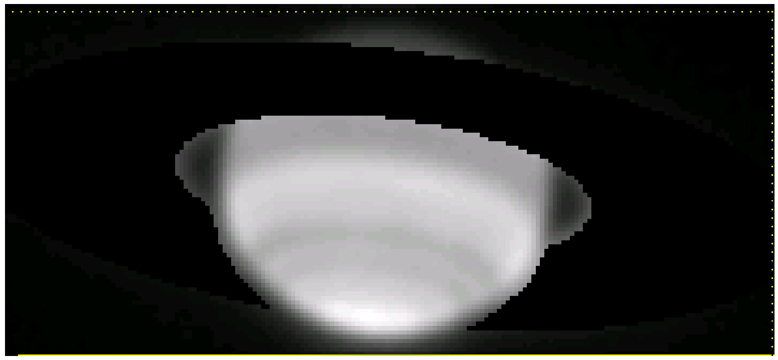

Figure 3: Blacked out Rings of Saturn

from fitting the limb. Blacked out rings are shown in Figure 3.

2.5 Center Finding

Small bodies, such as moons and stars, do not have a limb fitting procedure. Operations involving these objects require the user to click in the center of the object. However, the objects are quite small and so even a pixel or two deviation can be substantial.

To this end, we devised a similar routine to the limb-fitting guess to find the center of small objects. This routine sets the cursor to the centermost point of the object in a selected region, as several small objects may be present on one frame. The result of this procedure is shown in Figure 4.

2.6 Image Deconvolution

In collaboration with Dr. Richard Puetter (UC San Diego), we applied his image deconvolution system PIXON to a variety of images. The images acquired by ground-based telescopes are affected by atmospheric disturbance, which causes them to be slightly blurry even with ideal focus. Deconvolution allows us to retrieve, in part, the image before atmospheric blurring and telescope diffraction effects take their toll.

However, this process is by no means trivial or automated; it is heavily dependent on a very precise understanding of what kind of noise is introduced. Thus we spent a great deal of time trying to determine exactly what these parameters were in such a way that the routine could be automated for almost any image, and the system could look up the necessary parameters.

We initially received some very promising results from older, very noisy, data taken by Dr. David Gezari (NASA Goddard Space Flight Center). See Figure 5, original left, processed with PIXON right. However, later attempts to process other images resulted in increasing difficulties and less spectacular results.

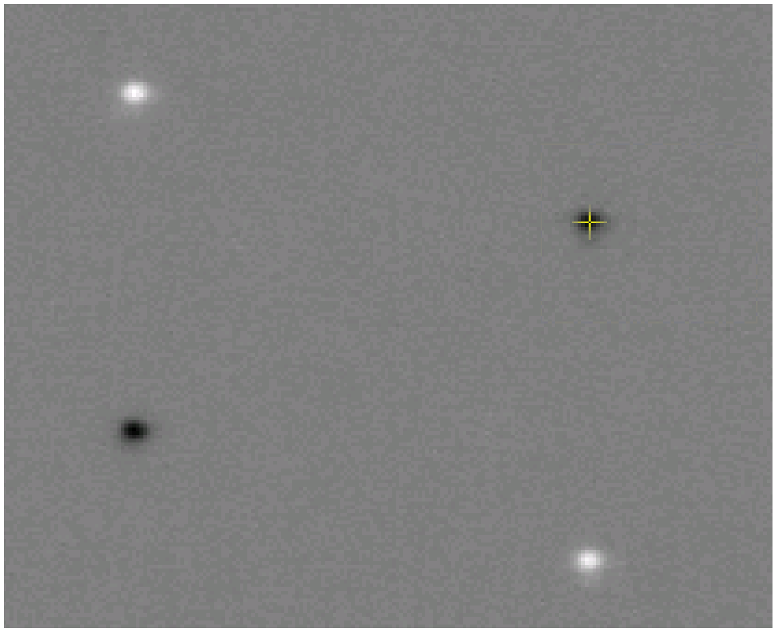

Figure 4: Selecting the Center of a Small Body

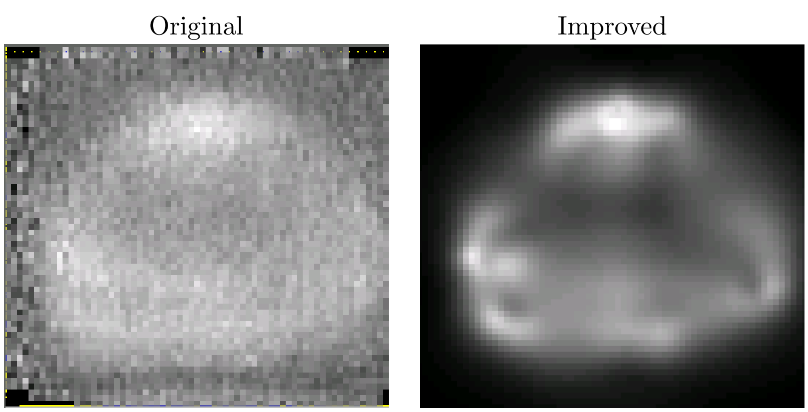

Figure 5: PIXON Deconvolution

3 Conclusion

Data reduction is an essential part of the astronomical observing process. However, much of the scientists' time is spent performing unnecessarily trivial operations, taking time away from science or observing. Our improvements to the data reduction process frees scientists to spend less effort on rote data reduction and more time on scientific discovery.

4 Acknowledgements

Thanks to Glenn Orton, my mentor, and Paul Parrish his post-doc. Together these two really hooked me up with everything I needed. Thanks to my co-interns who provided amusement. Thanks to Space Grant for providing funding. Thanks to Brother Brent for housing. Thanks to the people who paid for the free lunches on Fridays.