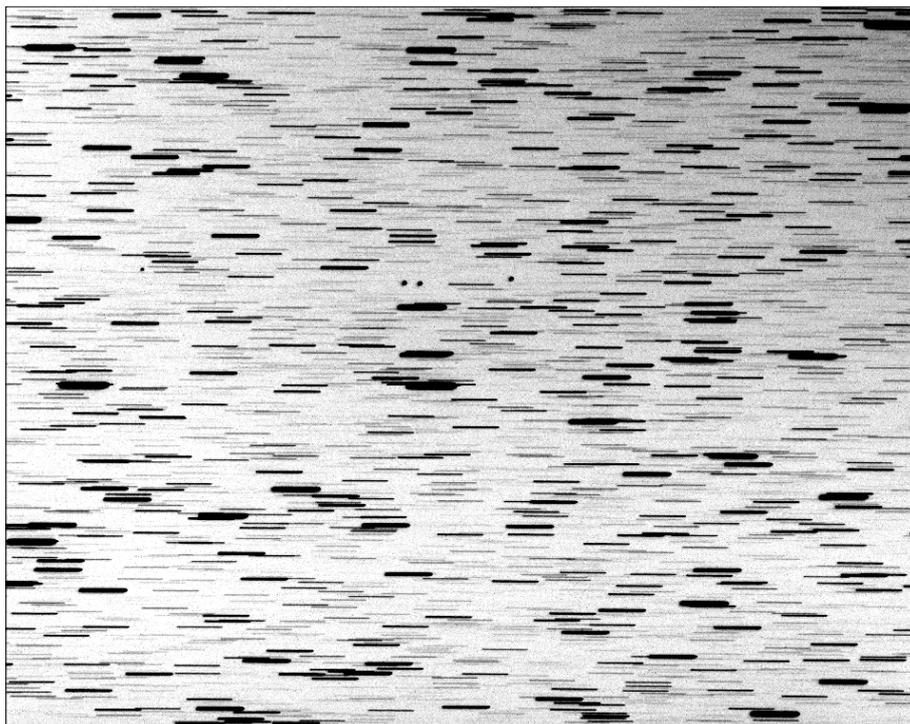


**Automated detection and analysis of object motion
in CCD sky images**



Project

**Report & Software Documentation
for Lockheed Engineering & Sciences Company
Subcontract No. 02N0106599**

Prepared by

**Michael Ashley, Joe Liske, Colin Bembrick,
Bradley Carter and Peter Mitchell**

Department of Astrophysics and Optics
School of Physics
University of New South Wales
Sydney NSW Australia 2052
Phone: 61-2-385-5465; Fax: 61-2-385-6060
e-mail: mcba@newt.phys.unsw.edu.au

February 19, 1996

Contents

1	Summary	4
2	Introduction	4
3	The appearance of moving objects on CCD images	4
3.1	Examples of satellite trails	5
4	This project	6
4.1	Search algorithm	6
4.2	Optimum observing strategy	7
4.3	Optimum data reduction strategy	7
5	Description of the software	8
5.1	The software distribution	9
5.2	Application to simulated data	9
5.3	Application to actual data	10
6	Conclusion	10
7	Supplementary information on the Automated Patrol Telescope	10
7.1	Specifications	10
7.2	Observer's Guide	11
7.3	History of the telescope	11
7.4	Recent scientific highlights using the APT	12
8	Software installation guide	14
9	Software documentation	23
9.1	User's Guide to the DEBRIS Package	23
9.2	debm _v — <i>shifts and combines images to make orbital debris visible</i>	27
9.3	debfind— <i>searches images for orbital debris</i>	32
9.4	debmv _f — <i>combination of debmv and debfind</i>	35
9.5	debreport— <i>reports results of search for orbital debris</i>	37
9.6	debsearch— <i>finds the best place to look for orbital debris</i>	40
10	Figures	43

Automated detection and analysis of object motion in CCD sky images

1 Summary

A search algorithm for the automated detection of object motion in digital sky images is described. The motivation for the development of the algorithm is discussed, and the implementation of the algorithm in software is given. The results from analysis of a series of sky images taken with the Automated Patrol Telescope (a 0.53 m Schmidt telescope at Siding Spring Observatory in Australia) are given, as verification that the software works on real data.

2 Introduction

Sky images taken for astronomical and space research frequently show objects that are not moving at the sidereal rate, for example, satellites in low-Earth orbit or asteroids. Images taken by telescopes with a wide field-of-view and sensitive electronic Charge-Coupled Device (CCD) sensors are particularly affected by such objects (see Figures 1 through 10). Automated detection of moving point sources in CCD sky images would be beneficial in sky survey work, especially where extensive surveys make identification of such objects by eye impractical. One way to detect point sources which do not move at the sidereal rate is to detect motion by comparing the position of point sources in successive sky exposures.

3 The appearance of moving objects on CCD images

Moving objects have different appearances on a CCD image depending on:

- **Their angular velocity.** Objects that move many pixels during the exposure time will leave a trail, slowly moving objects may remain in a single pixel, thereby making it difficult to distinguish them from stars.
- **Their brightness.** Bright objects are obvious, faint objects may be so faint that they aren't visible above the noise unless multiple images are co-added. It is also possible that faint, fast moving objects are not detectable above the noise in a single pixel, but they do become visible by integrating along their

trail. The eye is very good at identifying such faint trails—doing it in software is a challenge, particularly when the trail may curve slightly during the exposure time, as can be the case with objects in near geostationary orbits.

- **Their variation in brightness.** Some objects remain relatively constant in brightness, others (e.g., rotating pieces of space debris) show large variations, easily a factor of 10 or more. A single 5-minute 2×3 degree exposure on the sky will often show satellite trails, ranging from uniform lines to regular pulsations. Occasionally, more complex repeating patterns of dots may be seen, presumably due to tumbling debris with multiple reflective surfaces per revolution.

Moving objects can also be categorized according to their motion with respect to the earth and solar system:

- **Objects in orbit about the Sun.** Asteroids and comets in orbit about the Sun generally show only small variations from the sidereal rate (i.e., they are nearly fixed with respect to the stars). These objects are best detected by taking multiple images with the telescope tracking at the sidereal rate, and looking for motion relative to stars.
- **Objects in geostationary orbit.** Geostationary satellites and space debris have orbits which are almost fixed in altitude and azimuth at a given observing site on the Earth. Over a 24 hour period these objects will show a periodic motion with an amplitude depending on their maximum declination and any orbital corrections that may be applied. These objects are best detected by taking exposures with the telescope tracking turned off.
- **Objects in low-Earth orbit.** These objects move so quickly across the sky that only the brightest ones can be detected unless the telescope is tracked at precisely the right rate to make the objects appear stationary on the image.

3.1 Examples of satellite trails

Figures 1 through 10 are examples of satellite trails as observed by the Automated Patrol Telescope with 300 second exposures. The field of view is 2 by 3 degrees. While most of the trails appear to be of uniform brightness along their length, this is mostly due to the grey-scales chosen for the plots, and if you look at detail at the trails (e.g., Figures 7b, 7c, 8b, and 8c) you will often see quite striking regular variations.

These figures are included as Postscript files on the Exabyte tape delivered with this report. The raw FITS images are included also, to facilitate more detailed examination.

Figures 11 through 14 are examples of geostationary satellites (i.e., the telescope tracking was turned off, the stars are therefore trailed). These images were produced by running the software provided (see section §8 for details).

4 This project

In this project we concentrate on a subset of the categories in section §3. Specifically, we have created algorithms which make it possible to detect very faint near-geostationary objects—objects that are so faint that they are not detectable in a single exposure. Our technique, described in detail in the next section, is to take multiple short exposures and to stack them in software to make slowly moving objects visible. The images have to be taken with the telescope's sidereal drive off, so that a near-geostationary object will be fixed in position. The choice of exposure time is critical—too long an exposure time will blur the object and produce so many star trails that the object will be obscured; too short an exposure will cause the CCD readout noise to interfere with the signal from the object, and will reduce the observing efficiency due to the increased fraction of time being spent reading-out the CCD. These issues are discussed in more detail later.

4.1 Search algorithm

The algorithm uses a series of sky exposures to amplify point sources which move in a particular fixed direction between frames taken when the telescope is stationary. The digital images are all co-added after each digital image is shifted in one or two directions by a specified number of pixels. Because the motion of the object is not known beforehand, a large number of shifts are tried. Consequently a large number of co-added result frames are generated. Each one of these frames is analyzed by software for the presence of point source images. If the motion of an object has been matched, the possibly faint point source in each frame, when co-added, will be amplified. Otherwise, the image will be suppressed by the repeated additions with sky. The method therefore suppresses point sources such as stars or galaxies which have a precisely East-West motion. At the same time, the method allows objects not possessing sidereal motion but within a chosen range of linear speeds in an arbitrary direction to have their point-source images co-added.

The method can be extended to allow for non-linear object motion, although

this greatly increases the phase-space which must be covered. In practice we have found that quadratic terms in right ascension and declination are sufficient to approximate geostationary orbits over periods of an hour.

4.2 Optimum observing strategy

The optimum observing strategy depends very much on what sort of objects you wish to detect. In the case of geostationary debris, it is probably best to image objects when they are close to their maximum deviation from zero declination (as observed from the center of the Earth—parallax is important for such objects), since their relative motion on the sky is then a minimum. If you do not know the orbit in advance, you will have to observe in a range of declinations at each right ascension. It is also important to choose a time when the objects will be optimally illuminated by the Sun—the DEBSEARCH program within the DEBRIS package (included as part of this project) will perform the necessary calculations.

It is also wise to consider the density of stars in the direction of the planned observations, and to avoid times when the Milky Way is present, since the star density will then be so great as to make detecting faint objects almost impossible.

Avoid times when the Moon is present (since the increase in background signal will wash-out faint objects), and times when the sky illumination is changing, either due to a rising Moon or proximity to twilight (since the change in background will make it difficult to accurately combine the individual images).

A filter should be chosen to maximize the contrast between the sky and the reflected sunlight from the object.

When choosing the exposure time, it should be long enough that the CCD readout noise is insignificant with respect to the sky noise, but short enough so that objects of interest remain in a single pixel (or resolution element) during the exposure and do not saturate the detector.

4.3 Optimum data reduction strategy

The optimum data reduction strategy also depends on the sort of objects you wish to detect. After the usual bias subtraction, linearity correction, and flat-fielding, the images are ready for processing with the DEBRIS package. The crucial numbers to determine are the range and increments of velocity and acceleration parameters that you will use in the search. Too great a range and/or too small an increment will lead to very long execution times (days) even on fast workstations. To determine the range in velocity and acceleration over which to search, you need to estimate the likely range of orbital parameters of the objects in question. To de-

termine the increments, simply calculate the minimum increment in each velocity or acceleration vector that would result in a shift of one pixel (or in general, one resolution element) over the course of the observation run. Using finer increments is unnecessary.

The ability to detect faint objects depends critically on how well stars are removed when combining the individual images. It is worth experimenting with the input parameters to the IRAF `imcombine` routine, to minimize the noise from stars. The basic idea is to use a median filter, so that points in the final image are not greatly biased by having had stars being present in one or more of the input images. It can help to use median filtering with the additional of sigma-clipping (i.e., do not include in the calculation pixels that fall more than a user-specified number of standard-deviations away from the median). There are numerous other options that can be tweaked to produce better results (see the on-line help for `imcombine` for a complete description).

5 Description of the software

All the software has been developed in the IRAF (Image Reduction and Analysis Facility) environment. IRAF is supported by the NOAO (National Optical Astronomical Observatories), and is freely available and supported for most popular UNIX workstation environments.

More information about IRAF and how to install it can be obtained on the World Wide Web from the URL <http://www.noao.edu>.

A package of four `cl` (command language) scripts has been written that performs the tasks of shifting and co-adding the original frames and searching the co-added frames for possible detections. The result of such an analysis is presented in text form. The package is delivered complete with a package script and on-line documentation.

The four scripts of the package include the tasks `DEBMV`, `DEBFIND`, `DEBMVF` and `DEBREPORT`. `DEBMV` shifts and combines the original frames using the `IMCOMBINE` task of the IRAF `IMAGES` package. `DEBFIND` employs the `DAOFIND` algorithm of the `DIGIPHOT.DAOPHOT` package to detect sources exposed by `DEBMV`. `DEBMVF` is a combination of `DEBMV` and `DEBFIND`. As described above the algorithm produces a large number of co-added frames. The searching of these frames for point sources results in an equally large number of coordinate files. `DEBREPORT` scans these files and presents the results of the analysis in a manageable format.

In addition to these scripts the FORTRAN program `DEBSEARCH` has been in-

cluded. For a given night of the year DEBSEARCH tells you where the Earth's shadow is, thereby allowing you to choose times to observe at any particular azimuth and altitude. DEBSEARCH is currently "hard-wired" for the particular case of the Automated Patrol Telescope at Siding Spring Observatory, but it is straightforward to alter the code for other observing sites.

A detailed description of all the software components can be found in the on-line documentation.

5.1 The software distribution

The software is provided as a number of UNIX `tar` savesets on a 5 Gbyte 8-mm Exabyte data cartridge. The software has been verified to work on a Sun SPARCstation running Solaris 2.4, a DECstation 2000/260 running ULTRIX 4.4, and an HP735 workstation running the latest release of HP-UX.

Section §8 contains detailed instructions for installing the software and in testing it with trial data.

The Exabyte tape contains the following `tar` savesets, in the order shown:

<code>debris/</code>	The DEBRIS software and on-line documentation
<code>report/</code>	The source for this Report, including Postscript versions of the Figures
<code>trails/</code>	The raw images used to generate Figure 1 through 14
<code>demo/</code>	Demonstration images and results
<code>data/</code>	35 raw images which can be used to test the DEBRIS package

To provide some measure of redundancy, the above savesets are then repeated an additional two times. So if you can't read one of the savesets, you can try skipping five filemarks to the next copy of it.

5.2 Application to simulated data

In order to quantify the performance of the software, we generated fake data by adding artificial moving point sources to real data. The results show that a 15th magnitude moving object can be detected when using a stack of thirty-five 30-second exposures. Figure 13 shows the object added to a real exposure. Figure 14 shows the recovery of the object after shifting and stacking the exposures using the DEBRIS package.

5.3 Application to actual data

The algorithm has been applied to actual data taken with the Automated Patrol Telescope. Six objects with magnitudes between 10.4 and 12.0 have been found. Their motion clearly identifies these objects as satellites. At this stage of the analysis of the data no other objects with non-sidereal motion have been found.

6 Conclusion

Further development is necessary in order to derive a more practical algorithm (in terms of run-time). But our results from simulation and actual data show that our method is suitable for automatically detecting point sources that do not move at the sidereal rate using a telescope with a wide field of view.

7 Supplementary information on the Automated Patrol Telescope

The optical design of the APT resembles that of a Schmidt camera, but uses a 3-element lens to achieve a wide, corrected field of view. Telescope motion and operation of the CCD have been placed under computer control, allowing automated observations for long-term survey and monitoring projects. The APT has 0.5m aperture f/1 optics which produce a 5 degree flat field, of which a 2 by 3 degree field is utilised by the CCD currently installed. Imaging can be done either unfiltered or through B, V R and I broad-band filters.

7.1 Specifications

LOCATION (Siding Spring Observatory, near Coonabarabran, NSW, Australia)
 Latitude.....31d 16' 30'' South
 Longitude.....149d 03' 40'' East
 Altitude.....1140 metres above sea level

MOUNTING
 Declination South limit.....-75 degrees (software limit)
 Declination North limit.....+29d 55' (software limit)
 RA West and East limits.....Approx 4h from meridian (software limit)

ENCODERS
 RA incremental.....0.045 arcsecs per step, 2500 steps per
 revolution of the RA motor
 Dec incremental.....0.045 arcsecs per step, 1000 steps per
 revolution of the Dec motor

of any part of the sky. The CCD camera used in the APT was purchased from Wright Instruments, Enfield, U.K., and features an EEV CCD with 1152 by 770 pixels. The optics were redesigned at UNSW to allow the use of standard 5-mm thick astronomical filters in front of the CCD, and also to produce a flat-field. The optics were refigured by James Optics Ltd, Melbourne, Australia. In 1993, final installation was done of the telescope's modified optics, and observations for research with the completed telescope started in early 1994.

7.4 Recent scientific highlights using the APT

In reverse chronological order. Note: MCBA is Michael Ashley, CSB is Colin Bembrick, BDC is Brad Carter, JBWC is Jack Cochrane, PM is Peter Mitchell:

- **20-Feb-1995 Chris Young (Beijing Astronomical Observatory), MCBA**—*B* and *V*-band surface photometry of Virgo cluster galaxies.
- **18-Dec-1995 Ian Price (University of Newcastle; Stromlo Vacation Scholar), Brian Schmidt (MSSSO), MCBA**—Observations continue on the supernova project.
- **23-Aug-1995 Bruce Grossan, MCBA**—Work commences on using the APT for a gamma ray burst detection experiment, in a collaboration between the Lawrence Berkeley Laboratory (Grossan, Perlmutter), and UNSW (Ashley, Storey). Supernovae observations continue. An image was taken of Comet Hale-Bopp, currently at about 10th magnitude.
- **1995 Andrew Jacob**—An experimental program to test a new H alpha filter concept.
- **20-Apr-1995 Brian Schmidt (MSSSO), MCBA**—Observations commence for a supernova search. Over 800 galaxies can be observed in two nights (40 fields per night, each field is observed for two consecutive 300 second exposures, each field contains on average 10 galaxies).
- **04-Mar-1995 to 20-Apr-1995 MCBA, Oliver Prouton, and CSB**—Observations of the Delta Scuti variable star FG Vir were made as part of an international campaign coordinated by Michel Breger in Austria. 8000 images were taken over 163 hours of observations over 44 nights, using 60 second integrations separated by 15 seconds. An rms accuracy of 0.003 magnitudes was reached during each 60 second V-band observation of the 6.5 magnitude star. This accuracy was achieved by defocussing the APT so

that images were 15 pixels across, thereby reducing the effect of intrapixel sensitivity variations. The advantage of the APT for this observation is that bright standard stars could be measured on the same exposure. A subset of the images is being used for an undergraduate experiment in image analysis by Michel Breger in Austria.

- **Nov-1994, Joe Liske, MCBA, BDC, CSB, PM**—Observations of geostationary objects for a Lockheed contract.
- **1994, BDC, MCBA, CSB**—Time-resolved photometry of a flare star—part of an international multi-observatory campaign.
- **1994, MCBA, BDC**—High-speed imaging of Jupiter during the Shoemaker-Levy 9 collision. Nothing detected, but 40,000 blurry images obtained!
- **1994, Takehiko Wada, Ohno, Munetaka Ueno (University of Tokyo), MCBA, BDC, CSB**—A 20 cm infrared (JHK) camera was strapped to the side of the APT and used for 2.5 months to take images of the Galactic centre and LMC for a MACHO experiment.
- **1994, BDC, MCBA**—Observations of stars in open clusters for variability.

8 Software installation guide

INSTALLING THE DEBRIS PACKAGE:

If you have problems with the installation, you can e-mail Michael Ashley (mcba@newt.phys.unsw.edu.au) for advice. The package was written by Joe Liske (JOL@newt.phys.unsw.edu.au).

Before beginning installation:

To install the DEBRIS package you will need the following:

- The Exabyte cartridge containing the DEBRIS package.
- A UNIX workstation with a 8-mm Exabyte drive capable of reading 5 Gbyte cartridges. The software has been verified to work on a Sun SPARCstation running Solaris 2.4, a DECstation 2000/260 running ULTRIX 4.4, and an HP735 workstation running the latest release of HP-UX.
- IRAF V2.10 (or more recent) must be installed. IRAF may be obtained by anonymous ftp from ftp.noao.edu.
- The DIGIPHOT package for IRAF. This may also be obtained from ftp.noao.edu. In the past, the most recent version of DIGIPHOT has been called DIGIPHOTX - check carefully that you have the most recent version.
- A FORTRAN compiler. This is only required for the DEBSEARCH program, which is not an essential part of the DEBRIS package.

Important note:

Since the DEBRIS package uses other routines (such as DIGIPHOT) within IRAF, and since these other routines are subject to revision from time to time, it is possible that small incompatibilities will occur which prevent DEBRIS from working. If you have such problems, please e-mail Michael Ashley (mcba@newt.phys.unsw.edu.au) for assistance.

Installation:

These installation notes are contained in the file ``debris/README`` on the Exabyte tape.

1. Read the DEBRIS package off the tape:

The DEBRIS package requires about 1 Mbyte of disk storage. The logical place to put it is in your IRAF login directory, although it can be placed anywhere on the system. Note: in what follows, do not type the exclamation mark or the comments that appear to the right.

```
cd ~/iraf                ! go to your IRAF login directory
tar xvpf /dev/nrmt0h     ! read the first tar saveset off the tape
```

Your IRAF login directory is not necessarily ~/iraf, so substitute whatever is correct in your case. If you haven't used IRAF before, you can set it up by typing ``cd; mkdir iraf; cd iraf; mkiraf``. Note that you will have to replace ``/dev/nrmt0h`` with whatever your system uses for the device name of the Exabyte drive. Be sure to choose a "no-rewind" device so that you can read the next saveset without having to reposition the tape.

2. Optional: read the "report/ and trails/" savesets off the tape:

The next two savesets contain copies of this report (including the figures), and the raw images used to generate the figures. If you would like these on disk, you can read them using a command like "tar xvpf /dev/nrmt0h" for each saveset. Alternatively, skip over them using "mt -f /dev/nrmt0h fsf 2".

3. Optional: read the "demo/ and data/" saveset off tape:

The next saveset on the tape contains some demonstration images and some raw data that you can practice with after installing the DEBRIS package. You will need 110 Mbytes of spare disk space to unload the images, and a further 100 Mbytes or so to process the data.

```
cd /scratch              ! a scratch directory somewhere
tar xvpf /dev/nrmt0h     ! read the first tar saveset off the tape
```

Once again, you will need to specify the appropriate device for your system.

4. Optional: compile DEBSEARCH:

DEBSEARCH is a program to help you choose appropriate times to observe debris so that it is illuminated by the sun. The program has been hard-wired for the Automated Patrol Telescope, so you can skip this step if you are not using this telescope.

```
cd ~/iraf/debris          ! go to the top level of the DEBRIS package
f77 -o debsearch debsearch.f ! compile the FORTRAN program
```

Your system may use a different name and/or options for the FORTRAN compiler.

5. Edit "debris.cl":

The script "debris.cl" in the toplevel directory of the DEBRIS package needs to know the name of the directory in which it is stored. You have to edit this file in two places, as shown below:

```
# Package script for the DEBRIS package

# Load the digiphot and daophot packages:

digiphot
daophot

# Change ???/iraf/debris to the pathname of the toplevel
# directory of the DEBRIS package:

set      debris = "???/iraf/debris/"

package debris

task     debmv      = "debris$debmv.cl"
task     debfind    = "debris$debfind.cl"
task     debmvf     = "debris$debmvf.cl"
task     debreport  = "debris$debreport.cl"

# Change ???/iraf/debris to the pathname of the toplevel
# directory of the DEBRIS package (leave the $!):

task     $debsearch = "$!???/iraf/debris/debsearch"

clbye()
```

6. Edit your IRAF login:

You now need to let IRAF know about the existence of the DEBRIS package. The best place to do this is in your "loginuser.cl" file in your IRAF login directory. If you don't have such a file, then create it with the following four lines in it (if you do have the file, append these lines to it):

```
# The following three lines are needed by the DEBRIS package
task  $debris = "??*/iraf/debris/debris.cl"
set    helpdb = (envget("helpdb")//", ??*/iraf/debris/lib/helpdb.mip")
keep
```

Where you must replace "??*/iraf/debris" with the pathname of the toplevel directory of the DEBRIS package.

7. Compile the on-line help database:

In order to access the on-line documentation for the DEBRIS package you will need to compile the help database. This is done as follows:

```
cd ~/iraf                ! go to your IRAF login directory
cl                       ! start up IRAF
softools                 ! load the SOFTTOOLS package
debris                   ! load the DEBRIS package
cd debris$lib            ! go to the debris/lib directory
mkhelpdb root.hd helpdb.mip ! compile the database
```

TESTING THE DEBRIS PACKAGE:

This information is included in the file "demo/README" on the Exabyte tape.

1. Loading the DEBRIS package in IRAF:

The first test is simply to check if the DEBRIS package can be loaded by IRAF (presumably it can, since otherwise you would have been unable to complete the installation...):

```
cd ~/iraf                ! go to your IRAF login directory
cl                       ! start up IRAF
debris                   ! load the DEBRIS package
```

2. Check the availability of the on-line documentation:

After loading the DEBRIS package, try

```
help
```

```
phelp debguide          ! gives a paged guide to the DEBRIS package
```

3. Testing the software with actual data:

During the installation process, you would have loaded the "demo" and "data" directories at some location (called "/scratch" above).

The "demo" directory contains some four demonstration images and four text files showing the operation of the DEBRIS package.

The "data" directory contains the raw CCD images that were used to generate the "demo" results. You can process the "data" images yourself, and compare your answers with those in "demo".

At this point you might like to read the Tutorial and Demonstration Pages which can be found at the WWW DEBRIS Page at <http://www.phys.unsw.edu/~JOL/demo/html/debris.html>. Alternatively, continue with these instructions...

3.1 Start up IRAF, load the DEBRIS package, and read in the raw images:

```
cd ~/iraf              ! go to your IRAF login directory
cl                    ! start up IRAF
dataio                ! load the DATAIO package
debris                ! load the DEBRIS package
cd /scratch/demo      ! go to the demo directory
unlearn rfits         ! restore the "rfits" defaults
rfits demo0* * demo1,demo2,demo3,demo4 ! read in all the demo images
cd ../data            ! go to the data directory
rfits data00* * data  ! read in all the data images
```

If you are running out of disk space, you can remove the FITS images from the "demo" and "data" directories. If necessary, you can read in one FITS image at a time, and then delete the FITS image before reading in the next one.

3.2 Examine the images in the "demo" directory:

This is an optional exercise, which you may find informative:

After starting IRAF, go to the "demo" directory, and display the example images (how you do this depends on your local system, typically you use the "display" command within IRAF).

Here is a description of the images:

Figure 11 "demo01":

This image is an example of the data that can be analyzed by the DEBRIS package. The image is CCD-processed (i.e., it has been bias-subtracted and flat-fielded). The image shows four bright, geostationary objects which are probably satellites since they are all neatly lined up at zero degrees declination and do not move (as can be seen from the rest of the data set).

Figure 12 "demo02":

This image was created by the task DEBMV. It is a combination of 35 images like "demo01", added using a rejection algorithm. It shows the four satellites from "demo01" in front of a more or less flat background. The thousands of uniformly distributed very small dots are to be ignored. They are an artifact of an imperfectly applied flat-field. These dots are only visible in this particular image, and they do not affect the rest of the analysis.

Figure 13 "demo03":

This is the same image as "demo01" only that it has a point source artificially added to it at the position indicated by the circle. It is hardly visible. This image is actually the same as "data/data0022" (without the circle of course). In the rest of the data set contained in "data" this point source moves in both x- and y-directions at a speed of 2 pixels/image.

Figure 14 "demo04":

This image shows what DEBMV made out of the data set represented by "demo03". Note how the simulated moving object is now clearly visible as a point near the coordinate (270,572) (the object is easily visible on a computer screen, but does not reproduce well on Figure 14). Note also that the four previously identified satellites have now all but disappeared due to the shift and stack operation.

In order to test the software we will now attempt to remake the demo04 image from the given data set in "data".

3.3 Run the DEBRIS package to reproduce the "demo" results:

With IRAF, type

```
epar debmv
```

and then edit the fields to make the screen look like this:

```

-----
                                I R A F
                          Image Reduction and Analysis Facility

PACKAGE = debris
      TASK = debmv

input  =          data*.imh  List of input images
output =          demo      Root name of output images (must not contain '.')

(maxspee=          -2.83) Maximum trial speed (in pixels/image)
(minspee=          -3.)  Minimum trial speed (in pixels/image)
(nspeed =          1)    Number of trial speeds
(maxacc =          0.)   Maximum trial acceleration
(minacc =          -1.)  Minimum trial acceleration
(nacc   =          1)    Number of trial accelerations
(rcone  =          45.)  Angle of right side of cone with N
(lcone  =          0.)   Angle of left side of cone with N
(ndir   =          1)    Number of trial directions

(reject =          crreject) Type of rejection
(lthresh=          INDEF) Lower threshold
(hthresh=          INDEF) Upper threshold
(nlow   =          1)    minmax: Number of low pixels to reject
(nhigh  =          1)    minmax: Number of high pixels to reject
(mclip  =          yes)  Use median in sigma clipping algorithms?
(lsigma =          3.)   Lower sigma clipping factor
(hsigma =          5.)   Upper sigma clipping factor
(rdnoise=          rdnoise) CCD readout noise image header keyword
(gain   =          gain)  CCD gain image header keyword
(pclip  =          -0.5) pclip: Percentile clipping parameter

(mode   =          ql)
(flpar  =          no)  flush pfile on assign?
-----

```

You can then type ":go" to start DEBMV, or else exit "epar" with ":q" and run DEBMV by typing "debmv". Just type carriage returns for any questions that "debmv" asks. You must run "debmv" from the directory containing the data images (i.e., you would type "cd /scratch/data", assuming you unpacked the images starting at the "/scratch" directory).

You will receive a warning about image orientation; don't worry about this, read the on-line documentation for more information.

DEBMV should run for a few minutes before completing. The result

should be an image called "demo0001.imh" and two logfiles, "logdemo" and "logdemo0001". These last two files should be identical to the files of the same name in the directory "demo" (except for minor differences due to IRAF versions, dates, etc).

"demo0001.imh"

This image should be identical to the image "../demo/demo4.imh". It is the result of shifting and combining all 35 images in the ../data directory.

"logdemo":

A minimal logfile produced by DEBMV.

"logdemo0001":

A logfile produced by DEBMV with every output image listed.

We will now search "demo0001" for any objects using DEBFIND. Begin by loading the DAOPHOT package:

```
digiphotx          ! may be called digiphot
apphotx           ! may be called apphot
epar debfind
```

I R A F

Image Reduction and Analysis Facility

PACKAGE = debris
TASK = debfind

```
input      =          demo0001  List of input image(s)

(logfile=          default) name of the DEBMV logfile
(fwhmpsf=          2.28) FWHM of the PSF
(sigma =          100.) Standard deviation of background in counts
(thresh =          2.) Threshold in sigma for feature detection
(roundlo=          -1.2) Lower bound on roundness for feature detection
(roundhi=          0.5) Upper bound on roundness for feature detection
(rdnoise=          rdnoise) CCD readout noise image header keyword
(gain =          gain) CCD gain image header keyword
(trim =          yes) Trim images?

(mode =          ql)
(flpar =          no) flush pfile on assign?
-----
```

```
:go
```

The result will be a file called demo0001.coo.1. It should be the same as demo/demo4.coo

You will notice that besides the one detection we expected (which is by far the brightest, i.e., its entry in the MAG column is a magnitude brighter than any other) this coordinate file contains a number of unwanted detections all of which are very faint. To discard these (and to search the thousands of other coordinate files that would need processing were this a detailed analysis of the data) we will now run DEBREPORT:

```
epar debreport
```

```
-----
                                I R A F
                        Image Reduction and Analysis Facility
PACKAGE = debris
  TASK = debreport

input      =      demo0001.coo.1  List of input files

(logfile=      default) name of the DEBMV logfile
(boxsize=      10) Max difference in coordinates in neighbours
(fmag   =      -0.6) Only events with MAG < fmag are listed
(fround =      INDEF) Only events with ROUNDNESS < fround are listed
(x1     =      0) x1<=x2: Only events with x1<=XCENTER<=x2 are listed
(x2     =      INDEF) x1>x2: Only events with x1<=XCEN or XCEN<=x2 are listed
(y1     =      0) y1<=y2: Only events with y1<=YCENTER<=y2 are listed
(y2     =      INDEF) y1>y2: Only events with y1<=YCEN or YCEN<=y2 are listed

(list1 =      )
(list2 =      )
(list3 =      )
(list4 =      )
(mode  =      h)
(flpar =      no) flush pfile on assign?
-----
:go
```

The resultant output should be as shown in demo/results: possibly with a different ID number

If you get this far, you have successfully used the DEBRIS package to find a very faint simulated geostationary object.

9 Software documentation

9.1 User's Guide to the DEBRIS Package

1. Introduction

This guide provides a description of the procedures involved in detecting geostationary/geosynchronous orbital debris using an untracked observation method. It is a general outline of how data is acquired and how it is subsequently analyzed using the tasks of the DEBRIS package. A detailed description of the individual tasks can be found in the help documentation of the package.

2. The Method

In order to observe geostationary objects there are two obvious methods:

Tracking

If the telescope drives are on during the observations then of course all geostationary objects will appear as straight lines in the east-west direction. This means that the already faint objects get blurred out over many pixels, making them even harder to detect.

Untracked

However, if the telescope drives are switched off during the observations then all truly geostationary objects would appear as point sources. Unfortunately, orbital debris cannot be assumed to be truly geostationary due to changes in the orbit's inclination and ellipticity. Therefore it must be assumed to change both its HA and DEC. Although this motion is comparatively small it is obvious that this poses a limit on the exposure time if one keeps in mind that the whole purpose of this observing method is to keep the object if possible in one pixel.

The DEBRIS package was designed as a facility to analyze data acquired using the untracked method. The idea is to compensate for the above mentioned limit on the exposure time by taking a whole stack of frames and combining them later thus making objects visible that were undetectable in a single image. The following will identify the two major problems with this idea:

- Since the observer has no idea how fast and in what direction a particular piece of debris is moving it is impossible to observe this piece in the same pixel in every image of the series.
- Even if the first problem was somehow solved how can a detection

program (we are interested in an automated search algorithm!) distinguish between a star and orbital debris? (The roundness parameters of the DAOFIND task alone are not enough.)

The first problem is unsolvable. One has to live with it. This means that the telescope stays fixed with all drives switched off for the entire series of observations. Before combining the images they will be shifted with respect to each other in many different trial directions and at many different trial "speeds" in an attempt to simulate the motion of a particular object and to bring all its light into a single pixel. At the end of this, one is left with very many combined images to be searched. A particular object will only appear in very few (one or two) of these images, namely the ones which simulate its motion best.

The second problem can be dealt with in a more elegant manor simply by using an appropriate rejection algorithm when combining the images. In this way one can make the stars almost disappear altogether. So the problem of how to distinguish between stars and orbital debris does not pose itself in the first place.

3. The Data

As described above, the input to the DEBRIS package consists basically of a stack of images that were taken with all telescope drives switched off. It is important to note that all images should have the same exposure time and must be equally spaced in time. All image reduction should be performed before entering an analysis, i.e., the images should be bias-subtracted, corrected for dark current, flat fielded and if necessary illumination corrected. At present, the images should also have a particular orientation: north <--> left, east <--> up, south <--> right, west <--> down. The images do not have to be oriented in this way but it makes life a lot easier if they are.

4. The Analysis

The first step is the shifting and combining of the images. This is performed by the task DEBMV. The shifting for a particular image is performed according to three parameters which represent a trial motion. The output consists of the combined images and various logfiles. The logfiles will be used at later stages of the analysis and should not be deleted or tampered with light heartedly. The same goes for file names: The DEBRIS package uses easy to understand naming conventions for image-, log- and coordinate-files. In this way a task will automatically find the files it needs without having to bother the user. DO NOT rename files unless you know exactly what

you are doing.

The second step is the detection of orbital debris. The task DEBFIND scans the output images of DEBMV for local intensity maxima using the task DAOFIND in the APPHOTX package. Remember that stars have mostly been removed from the image yet residuals may still lead to unwanted detections. In many cases however these can be eliminated by using the roundness parameters of DAOFIND. The output of this step consists of one coordinate file per input image. Most of these will be empty (except for the header). Since there are a great number of coordinate files it would be a tedious task to check these files for detections one by one.

The third step is therefore the reporting of the results by DEBREPORT. This task reports in which images objects have been found, the magnitude of the objects and in how many "neighbours" a particular object was found. A "neighbour" of an image is an image with neighbouring shifting parameters. This quickly shows the user whether detections in two different images refer to the same object or not. It is also some sort of measure of the reliability of a detection.

4.1 DEBMV and DEBFIND vs DEBMVF

The task DEBMVF is a combination of the tasks DEBMV and DEBFIND. It serves exactly the same purpose but there is an important difference in the order in which the steps are executed. If DEBMV and DEBFIND are used for an analysis then first all images are created and then they are searched. This is impractical since it requires a large amount of available memory. Instead, DEBMVF searches every image right after its creation and is then free to delete the image if the user so wishes. So the actual analysis is best performed using DEBMVF deleting every image after it has been searched. Any image can be created again at any time using the information in the logfiles; if one wishes to create an image again just for the purpose of examination DEBMV should be used. DEBFIND is best used for searching images repeatedly with different parameter settings.

Throughout all documentation on the DEBRIS package almost all references to the DEBMV and DEBFIND tasks apply equally to the DEBMVF task.

5. Summary

The "course of action": The stack of input images are shifted with respect to each other in order to simulate the movement of a supposed piece of orbital debris. Then the images are combined using a rejection algorithm so that stars disappear. The combined images

are searched for remaining brightness maxima and the result is reported in a manageable fashion.

Detailed descriptions of the individual procedures can be found in the respective help pages. Additionally, the help pages of the tasks IMCOMBINE and DAOFIND provide detailed information on the rejection and detection algorithms employed.

9.2 debmv—shifts and combines images to make orbital debris visible

USAGE

debmw input output

PARAMETERS

List of images to be shifted and combined. The images must appear in the same order in which they were taken.

output

Root name of output images and logfiles. The image names will be of the form "output.nnnn", where "n" represents a digit. The logfile names will be of the form log"output" and log"output"nnnn for the individual IMCOMBINE logfiles. If the same root name has been used in a previous run of DEBMV then DEBMV will only use presently completely unused image and individual logfile names in order to avoid confusion. "output" must not contain the character "!"!

maxspeed = , minspeed =

The maximum and minimum values to be tried for the shifting parameter SPEED. The unit is pixels per image. Positive values of SPEED simulate northward motion, negative values simulate southward motion.

nspeed = 1

The number of values to be tried for SPEED. If "nspeed" = 1 then only "maxspeed" is tried.

maxacc = , minacc =

The maximum and minimum values to be tried for the shifting parameter ACCELERATION. The unit is pixels per image**2. Positive values of ACCELERATION simulate northward acceleration and vice versa.

nacc = 1

The number of values to be tried for ACCELERATION. If "nacc" = 1 only "maxacc" is tried.

rcone = , lcone =

The maximum and minimum values to be tried for the shifting parameter DIR. The unit is in degrees from the north-south axis. For northward motion (positive values of SPEED), positive values of DIR simulate westward motion and vice versa. For southward motion, positive values of DIR simulate eastward motion and vice versa.

ndir = 1

The number of values to be tried for DIR. If "ndir" = 1 only

"rcone" is tried for DIR.

reject = "crreject" (none|minmax|ccdclip|crreject|sigclip|avsigclip|pclip)

The rejection algorithm to be applied when combining the images. The rejection is applied after thresholding. For a detailed description of the algorithms see the help page of the IMCOMBINE task in the IMAGES package. The rejection choices are:

- none - No rejection
- minmax - Reject the nlow and nhigh pixels
- ccdclip - Reject pixels using CCD noise parameters
- crreject - Reject only positive pixels using CCD noise parameters
- sigclip - Reject pixels using a sigma clipping algorithm
- avsigclip - Reject pixels using an averaged sigma clipping algorithm
- pclip - Reject pixels using sigma based on percentiles

In this application "crreject" is the most suited rejection algorithm because it rejects the stars.

lthreshold = INDEF, hthreshold = INDEF

Low and high thresholds to be applied when combining the images. This is done before rejection. If INDEF the thresholds are not used.

nlow = 1, nhigh = 1 (minmax)

The number of low and high pixels to be rejected by the "minmax" algorithm. These numbers are converted to fractions of the total number of input images so that if no rejections have taken place the specified number of pixels are rejected while if pixels have been rejected, by thresholding or nonoverlap, then the fraction of the remaining pixels, truncated to an integer, is used.

mclip = yes (ccdclip, crreject, sigclip, avsigclip)

Use the median as the estimate for the true intensity rather than the average with high and low values excluded in the "ccdclip", "crreject", "sigclip", and "avsigclip" algorithms? The median is a better estimator in the presence of data which one wants to reject than the average. However, computing the median is slower than the average.

lsigma = 3., hsigma = 5. (ccdclip, crreject, sigclip, avsigclip, pclip)

Low and high sigma clipping factors for the "ccdclip", "crreject", "sigclip", "avsigclip", and "pclip" algorithms. They multiply a "sigma" factor produced by the algorithm to select a point below and above the average or median value for rejecting pixels. The lower sigma is ignored for the "crreject" algorithm.

rdnoise = , gain = (ccdclip, crreject)

CCD readout noise and gain image header keywords used with the

"ccdclip" and "crreject" algorithms. The keywords should contain the actual values in units of electrons and electrons/adu respectively.

pclip = -0.5 (pclip)

Percentile clipping algorithm parameter. If greater than one in absolute value then it specifies a number of pixels above or below the median to use for computing the clipping sigma. If less than one in absolute value then it specifies the fraction of the pixels above or below the median to use. A positive value selects a point above the median and a negative value selects a point below the median. The default of -0.5 selects approximately the quartile point.

DESCRIPTION

This task is the first step in the search for orbital debris. It uses the IMCOMBINE task in the IMAGES package to shift and combine the "input" images in order to make objects visible that were undetectable in any one of the "input" images.

The "input" images should be orientated in the following way:

N <-> left W <-> down S <-> right E <-> up

First, the images are shifted with respect to each other because it cannot be assumed that pieces of orbital debris are truly stationary. Therefore, the shifting of the images must attempt to simulate the movement of the debris. This is why the shifting parameters are called SPEED, ACCELERATION and DIR. SPEED is a linear motion parameter, ACCELERATION a quadratic one which is only applied to the north-south motion, and DIR represents the direction in which the debris is moving. E.g. the shift of the *i*th image with respect to the first image of the series is calculated according to:

$$\begin{aligned} \text{N-S: } x\text{shift} &= [(i-1) \times \text{SPEED} + (i-1)**2 \times \text{ACCELERATION}] \times \cos(\text{DIR}) \\ \text{E-W: } y\text{shift} &= (i-1) \times \text{SPEED} \times \sin(\text{DIR}) \end{aligned}$$

The true north-south motion of a piece of orbital debris is sinusoidal since it is due to the inclination of the orbit. The movement is symmetric around 0 degrees DEC with a 24 hour period and an amplitude equal to the orbit's inclination. So at 0 degrees DEC all debris should move linearly in the north-south direction which justifies the linear coefficient. The quadratic coefficient represents the admixture of a cosine term to the pure sine (phase shift). This is helpful when using a telescope with a very wide field of view. The edges of the field of view may be as far away from 0 degrees DEC as 1 or 2 degrees. Pieces of debris with an amplitude of 1 or 2 degrees that are at or close to the turning point of their motion at the time of observation will surely not

move in a linear fashion. In fact, they will move quadratically (because the first term in the cosine expansion series is quadratic).

The east-west motion is not considered to be strongly accelerated which is why in this case only the linear coefficient is used.

The ranges and step sizes over which the three shifting parameters are scanned should be considered carefully before entering a major analysis, since "nspeed" x "nacc" x "ndir" is equal or slightly greater than the number of output images.

After having calculated and applied the shifts the thresholds are applied. Finally, the images are averaged using the "reject" algorithm. Since the stars have different positions in every image, a particular pixel will only be illuminated by stars in very few images whereas a piece of debris, whose motion has successfully been simulated, will illuminate the same pixel in every image. Thus, the cosmic-ray reject algorithm will treat the stars as cosmic rays and will reject them leaving the piece of debris exposed and easy to detect.

In addition to the combined images, the output consists of different logfiles. The first, log"output", will contain a list of all output images and "their" values of the shifting parameters. A re-run of DEBMV with the same value of "output" will append its list of images and shifting parameter values to the existing file log"output". A second logfile is kept for every individual image, which is actually the logfile produced by IMCOMBINE.

EXAMPLES

1. Shift and combine the images 1_run0001 - 1_run0010 which are the only files in the current directory. Create three output images:

```
cl> eparam debmv

... edit the parameters

cl> debmv 1_run00*.imh out nspeed=1 nacc=1 ndir=3

cl> dir

1_run0001.imh  1_run0006.imh  logout          out0002.imh
1_run0002.imh  1_run0007.imh  logout0001     out0003.imh
1_run0003.imh  1_run0008.imh  logout0002
1_run0004.imh  1_run0009.imh  logout0003
1_run0005.imh  1_run0010.imh  out0001.imh
```

TIME REQUIREMENTS

BUGS

SEE ALSO

images.imcombine, debuguide

AUTHOR

Joe Liske (JOL@newt.phys.unsw.edu.au)

E-MAIL HELP

Michael Ashley (mcba@newt.phys.unsw.edu.au)

9.3 **debfnd**—searches images for orbital debris

USAGE

debfnd input

PARAMETERS

input

The images to be searched. Usually these will be the output images of the DEBMV task.

logfile = "default"

The name of the DEBMV logfile. If left at "default" the task assumes the image names to be of the form "debm.outputnnnn", where "n" represents a digit. Furthermore, it assumes the logfile name to be of the form logdebm.output. If either the image(s) or the logfile have been renamed then logfile must be set appropriately. In this case though, the task loses its ability to automatically find the corresponding logfile for every individual image in "input": The file named in "logfile" will be used as logfile for all images! The contents of "logfile" must, of course, be of the same format as the logfile kept by DEBMV.

fwhmpsf = 1.0

Full width half maximum of the point spread function. This parameter is crucial to successful detection and should be well known. It can be determined using e.g. IMEXAMINE in the PROTO package.

sigma =

The estimate of the standard deviation of the sky pixels. This parameter should either be determined from the images to be searched themselves or alternatively it can be inferred from the images that were the input to the DEBMV task.

threshold = 2.0

The object detection threshold above local background in units of "sigma".

roundlo = -1.2, roundhi = 0.2

"roundlo" and "roundhi" are numerical cutoffs on the image roundness statistic originally designed to eliminate brightness maxima which are due to bad rows or columns, rather than to astronomical objects. Here, "roundhi" is used to eliminate brightness maxima due to stars, rather than to orbital debris, which tend to be elongated in the direction of the movement of the stars across the original frames (input images to the DEBMV task).

rdnoise = , gain =

CCD readout noise and gain image header keywords. The keywords should contain the actual values in units of electrons and electrons/adu respectively.

trim = no

If this parameter is set the images in "input" will be trimmed to the section that is searched.

DESCRIPTION

This task is the second step in the search for orbital debris. It searches the "input" images for brightness maxima using the DAOFIND task of the APPHOT package with its parameters appropriately set. (See the help page for DAOFIND for a detailed description of the detection algorithms).

The input images are usually the output of the DEBMV task. These images are averages of images that have been shifted with respect to each other. Therefore, only certain central parts of those images are actually useful. DEBFIND uses the information about every individual image which is stored in the logfile produced by DEBMV to determine those parts. Only these sections are actually searched by DAOFIND. If the "trim" flag is set then the images will also be trimmed to the search sections. This will be recorded in the image header. If an already trimmed image is searched again DEBFIND will recognize that it has been trimmed to the appropriate section and will therefore search the entire image and will not attempt to trim it again.

The output consists of the coordinate files produced by DAOFIND. The output file names are of the form "imagename.coo.?", where "?" stands for the highest available version.

EXAMPLES

1. Search the image test0005 for for orbital debris and trim the image:

```
cl> debfind test0005 trim+
```

2. Search the same image again with different search parameters:

```
cl> eparam debfind
```

```
... edit the parameters
```

```
cl> debfind test0005
```

Note: the setting of "trim" is irrelevant.

3. Search the images test2070 - test2099 and the image search0376:

```
cl> debfind test20![7-9]*.imh,search0376 logfile="default"
```

TIME REQUIREMENTS

BUGS

SEE ALSO

apphot.daofind, debmv, debguide

AUTHOR

Joe Liske (JOL@newt.phys.unsw.edu.au)

E-MAIL HELP

Michael Ashley (mcb@newt.phys.unsw.edu.au)

9.4 debmvf—combination of debmv and debfind

USAGE

debmvf input output

PARAMETERS

See the DEBMV and DEBFIND help pages for most of the parameters.

delete = no

If this flag is set then the combined images will be deleted after they have been searched for objects.

trim = no

If this flag is set and "delete" = no then the output images will be trimmed to the sections that were actually searched for objects.

DESCRIPTION

This task is a combination of the tasks DEBMV and DEBFIND. It works in the same way as running first DEBMV and then running DEBFIND on the output images. But there is one important difference: Instead of producing all the images first and then searching them, DEBMVF searches every image right after its creation and is then free to delete the image if the "delete" flag is set. So DEBMVF uses considerably less memory than DEBMV and DEBFIND.

For further description of the task refer to the help pages of the DEBMV and DEBFIND tasks.

EXAMPLES

1. Shift and combine the images 1_run0001 - 1_run0010 which are the only files in the current directory. Create three output images and search them for objects:

```
cl> eparam debmvf
... edit the parameters
cl> debmvf 1_run00*.imh out nspeed=1 nacc=1 ndir=3 delete-
cl> dir
1_run0001.imh  1_run0006.imh  logout          out0002.coo.1
1_run0002.imh  1_run0007.imh  logout0001     out0003.coo.1
1_run0003.imh  1_run0008.imh  logout0002     out0001.imh
1_run0004.imh  1_run0009.imh  logout0003     out0002.imh
```

1_run0005.imh 1_run0010.imh out0001.coo.1 out0003.imh

TIME REQUIREMENTS

BUGS

SEE ALSO

debm, debfind, debuguide

AUTHOR

Joe Liske (JOL@newt.phys.unsw.edu.au)

E-MAIL HELP

Michael Ashley (mcba@newt.phys.unsw.edu.au)

9.5 debreport—reports results of search for orbital debris

USAGE

debreport input

PARAMETERS

input

The coordinate files that are to be checked for objects.

logfile = "default"

The name of the DEBMV logfile. If left at "default" the task assumes the "input" file names to be of the form "debmv.output"nnnn.coo.n, where n represents a digit. Furthermore, it assumes the logfile name to be of the form log"debmv.output". If either the coordinate files or the logfile have been renamed then "logfile" must be set appropriately. In this case however, the task loses its ability to automatically find the corresponding logfile for every individual coordinate file in "input": The file named in "logfile" will be used as logfile for all coordinate files in "input"! The contents of "logfile" must, of course, be of the same format as the logfile kept by DEBMV.

boxsize = 10

Size of a coordinate box around an object. If an object is found within that box in a neighbour it is presumed to be the same object.

fmag = 0.

Brightness filter. Only objects that are brighter than "fmag" will be included in the output.

fround = INDEF

Roundness filter. Only objects with ROUNDNESS < "fround" will be included in the output. If left at INDEF fround will automatically be set to the highest value of the ROUNDHI keyword in the headers of all coordinate files in "input".

x1 = 0, x2 = INDEF, y1 = 0, y2 = INDEF

Positional filter. If $x1(y1) \leq x2(y2)$ then only objects with x(y)-coordinates between $x1(y1)$ and $x2(y2)$ will be included in the output. If $x1(y1) > x2(y2)$ then only objects with x(y)-coordinates outside the range defined by $x1(y1)$ and $x2(y2)$ will be reported. If $x2(y2)$ is left at INDEF it will be set to a very large number.

DESCRIPTION

DEBREPORT reports the results of an orbital debris search. For every non-empty coordinate file in "input" DEBREPORT determines the number

of neighbours for the corresponding image and how many of the coordinate files of these neighbours are present in "input". The neighbours of an image "x.imh" are those images listed in x.imh's logfile with two of the three shifting parameters equal to the corresponding ones of x.imh and the third one closest to the corresponding one of x.imh, being either greater or smaller. Therefore, any image can have up to $3 \times 2 = 6$ neighbours. Every object in a non-empty coordinate file is then taken through the filters. The purpose of the filters is to discard unwanted detections due to poorly chosen search parameters of DEBFIND, dead pixels, etc. Once an object has made it through the filters its id-number, position, magnitude and roundness are listed. In addition, DEBREPORT checks whether the same object was found in any of the neighbours. A detection in a neighbour is presumed to be the same object if its coordinates are within "boxsize" of the object which is currently being reported. Such a detection will still be counted even if it will not be reported by itself because it has not made it through the filters. This gives the user a fairly good idea of whether two reported detections in two different coordinate files refer to the same object. Although it is quite possible to detect a real object only in one single image, this feature is also some indication of how reliable the detection is.

EXAMPLES

1. Check all coordinate files (version 1) beginning with "deb":

```
cl> debreport deb*.coo.1
```

2. Check the same files again in addition to all files beginning with " search". Report only objects that are brighter than -0.2:

```
cl> debreport deb*.coo.1,search*.coo.1 logfile="default" \  
>>> fmag=-0.2
```

3. Check the file search0354.coo.1 again after having renamed the logfile to "todays_log":

```
cl> debreport search0354.coo.1 logfile="todays_log"
```

TIME REQUIREMENTS

BUGS

If the logfile entry for an image in which objects have been detected is faulty, then the neighbour statistics of the image's neighbours should not be trusted.

SEE ALSO

debfind, debuguide

AUTHOR

Joe Liske (JOL@newt.phys.unsw.edu.au)

E-MAIL HELP

Michael Ashley (mcba@newt.phys.unsw.edu.au)

9.6 debsearch—*finds the best place to look for orbital debris*

USAGE

debsearch

INPUT

day of the year

The day of the year on which the observations are to take place.

three character identifier

Identifies the output file: debsearch"xxx".dat

UT hours, UT min

The Universal Time at which the observations are to begin.

decl1, decl2, decl3

The declinations to be used in the search. Since the DEBRIS package requires data taken at low declinations "decl,..." are limited to the range -4 to 4 degrees.

Number of images to be taken per field

Of every field that is being observed a whole stack of exposures need to be taken (see debguide).

Time between images in seconds

This includes integration time, readout time and any possible dead time.

DESCRIPTION

DEBSEARCH is a stand alone FORTRAN program that determines the coordinates of the best search area for a given day of the year. The "best search area" is defined as the area that gives the smallest reflection angle for sunlight.

First the program determines whether or not the declination band of -4 to 4 degrees is partly obscured by the earth's shadow. If so the program will calculate the coordinates of the edges of the shadow and will instruct the user to observe at the western edge before the sun's "anti-culmination" (azimuth of 180 degrees on the southern hemisphere) and at the eastern edge thereafter. If the sun is at high declinations and the -4 to 4 band is not obscured the program will find the point opposite the sun. The output coordinates are corrected for the fact that the geostationary ring is not infinitely far away compared with the earth's radius. The program assumes an hour angle of 4 hours to be the limit of the telescope's RA drive. If the best area for observation is beyond that limit the program will simply give an RA which corresponds to

an hour angle of 4 hours. The program assumes 2 minutes of maneuvering time if only the declination needs to be changed and 10 minutes if both declination and RA are changed (re-calibration might be needed).

The output file contains the corrected coordinates, the UT at which to begin the observation of these coordinates (i.e. at which to turn the telescope drives off), the true longitude of the area that is being observed and the reflection angle of the sunlight.

EXAMPLES

Make an observation plan for the 19th of February (50th day of the year) starting at UT 10:00 (dusk), taking 20 images per field at declinations +1.0 and -1.0 degrees and 80 seconds between images. The output file would look like this:

CORRECTED RA	CORRECTED DEC	UT	LONG	SPA
HR MIN SEC	DEG MIN SEC	HR MIN SEC	DEG MIN SEC	DEG
9 0 16	6 27 37	10 0 0	170 27 31	25.9
	4 9 16	10 28 40	170 27 31	26.8
8 50 23	6 27 37	11 5 20	-173 9 48	25.9
	4 9 16	11 33 60	-173 9 48	26.8
8 39 11	6 27 37	12 10 40	-156 47 7	25.9
	4 9 16	12 39 20	-156 47 7	26.8
8 27 33	6 27 37	13 15 60	-140 24 26	25.9
	4 9 16	13 44 40	-140 24 26	26.8
10 56 13	6 27 37	14 21 20	-158 3 45	14.4
	4 9 16	14 49 60	-158 3 45	15.9
10 44 35	6 27 37	15 26 40	-141 41 4	14.4
	4 9 16	15 55 20	-141 41 4	15.9
10 33 22	6 27 37	16 31 60	-125 18 23	14.4
	4 9 16	17 0 40	-125 18 23	15.9
10 23 27	6 27 37	17 37 20	-108 55 42	14.4
	4 9 16	18 6 0	-108 55 42	15.9
10 35 54	6 27 37	18 42 40	-97 7 53	17.9
	4 9 16	19 11 20	-97 7 53	19.1
11 41 24	6 27 37	19 47 60	-97 7 53	32.5
	4 9 16	20 16 40	-97 7 53	33.3
12 46 55	6 27 37	20 53 20	-97 7 53	48.2
	4 9 16	21 22 0	-97 7 53	48.7
88 88 88	888 88 88	88 88 88	8888 88 88	888.8

Notice that the reflection angle (SPA) is constant at first (for a given declination) then changes at UT 14 21 20 and then is constant again. On the 50th day of the year the declination band from -4 to 4 degrees is already partly covered by the earth's shadow. At the

beginning of the night the program tells the user to observe at the western edge of the shadow. Since each observation lasts about 26 minutes the program calculates the coordinates so that the telescope will point to the edge of the shadow at the end of a particular observation. After the sun's "anti-culmination" around UT 14:30 the program of course tells the user to point the telescope right at the eastern edge of the shadow at the beginning of the observation. In the morning hours the best spot for observation has moved beyond the telescope's RA limit and the program calculates the coordinates at that limit. Notice how the SPA gradually worsens. Notice also that LONG is now constant meaning that the last three observations are made of the same segment of the geostationary ring which obviously means that only the first of these is of value and the rest can be discarded.

So the course of the observations would be thus:

Point the telescope at 09 00 16 06 27 37 and switch off both drives at UT 10 00 00. Take 20 images 80 seconds apart from each other. Switch the DEC drive back on and move the telescope to DEC 04 09 16. Switch it off again at UT 10 28 40 and start taking another series. Switch both drives back on and move the telescope to 08 50 23 06 27 37 after having made sure that the telescope controlling computer knows where the telescope is pointing. Switch off both drives at UT 11 05 20

TIME REQUIREMENTS

BUGS

SEE ALSO

 debuguide

AUTHOR

 Joe Liske (JOL@newt.phys.unsw.edu.au)

E-MAIL HELP

 Michael Ashley (mcba@newt.phys.unsw.edu.au)

10 Figures