Beginner's Guide to SuperCOSMOS data

N C Hambly

July 30, 1998

1 Introduction

This document describes the nature and format of SuperCOSMOS data. No prior knowledge other than a little FORTRAN 77 is assumed. Information is given on the type of measurement jobs that are done for users, how the data relate to the original photograph scanned, the files and file format that are supplied, brief explanation of the 'Quality Control' plots, and finally simple examples of code showing how to read the data. Note that this document is expected to 'evolve' over time as the needs of users become clearer from feedback, so be sure to obtain the latest version. Also, email any comments, suggestions or bug reports to `N.Hambly@roe.ac.uk`.

Once your are familiar with the concepts described here, a set of guides are becoming available showing how to process SuperCOSMOS data; these are described in an introductory document which contains more information for users familiarised with this Beginner's Guide.

2 How SuperCOSMOS scans a plate

SuperCOSMOS digitises a photograph at 10μm resolution with 15–bit transmission values. A linear CCD images a line of pixels simultaneously; currently, the line is 1152 pixels (*i.e.* 11.52mm) wide. A complete image of the area required is then built up as a series of 'lanes', each is 1152 pixels wide. Figure 1 shows the usual orientation of a plate scan.

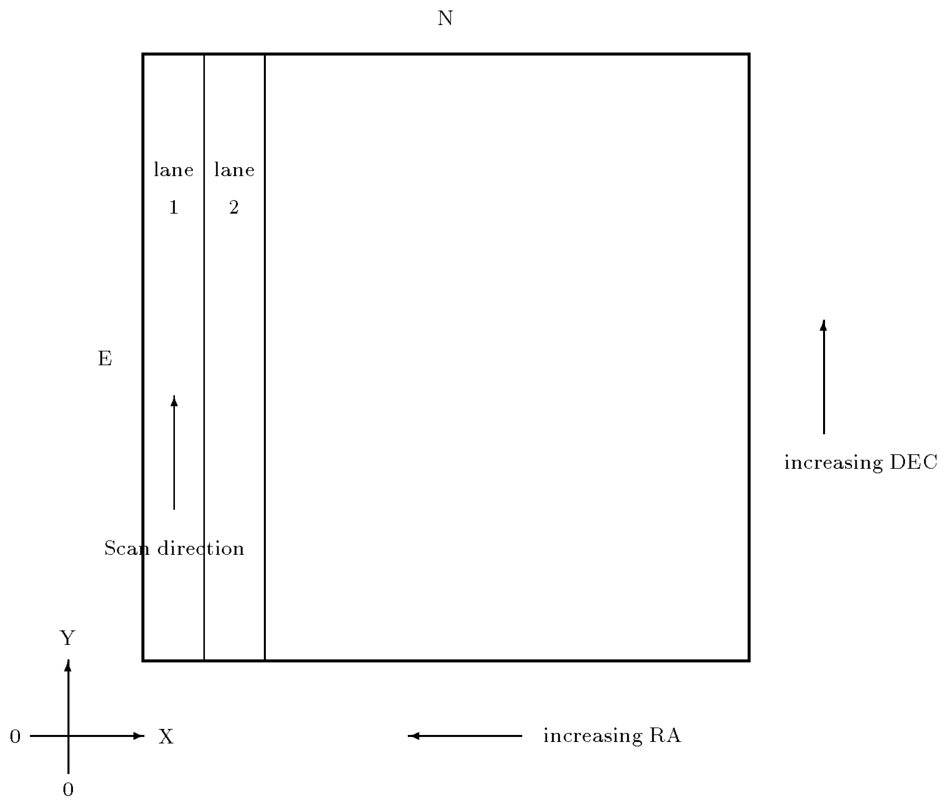

Figure 1: The usual orientation of a SuperCOSMOS measurement to the co-ordinate system

While SuperCOSMOS always operates in a pixel digitisation mode (the 'mapping mode', or MM), offline processing allows any of three options for a user job:

1. A small area scan yielding a 2–D image (up to 8192 × 8192 pixels)

2. A large area scan, between $3° \times 3°$ to $6° \times 6°$, which is then run through image detection and parameterisation software (the 'image analysis mode', or IAM), the image parameters being supplied to the user

3. As 2., but the pixel data also being supplied to the user (small area pixel images can be extracted from the large MM data set for users).

Digitisation is always done to a whole number of lanes, so the extent in X is always an integer number of 1152 pixels; in general a square map is produced so the extent in Y is the same. The default length for a whole–plate scan is 28 lanes$\times$1152 pixels = 322.56mm, or 6.0° at the 48 inch Schmidt Telescope plate scale of 67.14 arcsec mm^{-1}. For a complete technical description of the scanning hardware, see Hambly & Miller (1997). Details of the scanning procedure can be found in MacGillivray & Thomson (1998). Details of the processing of pixel data through to an object list can be found, for example, in MacGillivray & Stobie (1984) and Beard, MacGillivray & Thanisch (1990) and references therein.

3 The data tape

Jobs will in general be copied to DAT or EXABYTE tape and mailed to users (archive copies are kept at the Royal Observatory, Edinburgh). The tape format is a Unix tape archive, or tar file, at high or low density depending on user instructions (the default is high density). To list the contents of your tape:

```
% tar -tvf /dev/nrmt0h
./
./J12345
./J12345/hkfile.txt
./J12345/iam.srtrd
./J12345/mmdata
./J12345/mmdata/header
./J12345/mmdata/l1
    .
    .
    .
%
```

There are a few things to note:

- the device name `/dev/nrmt0h` is site dependent, but a trailing 'h' for high density (or 'l' for low density) is quite common
- the archive is a 'relative' tarfile[1] and upon extraction (see below) will be restored to the current directory in a subdirectory with the plate number (in this example J12345) for the subdirectory name.

A typical $6° \times 6°$ dataset, including the MM data, will be more than 2 Gbyte in size. To extract a specific file (for example, to get the small `iam.srtrd` and `hkfile.txt` files without the pixel data, if present):

[1] see, for example, Bourne (1990)

```
% tar -xvf /dev/nrmt0h ./J12345/hkfile.txt ./J12345/iam.srtrd
x ./J12345/hkfile.txt
x ./J12345/iam.srtrd
% cd J12345
% ls
hkfile.txt     iam.srtrd
%
```

Alternatively, to restore the whole archive (assuming the MM data are present and you have the disk space):

```
% tar -xvf /dev/nrmt0h
x ./J12345
x ./J12345/mmdata
   .
   .
   .
% cd J12345
% ls -R
hkfile.txt     iam.srtrd

./mmdata:
header     l3     .....
l1         l4
l2         l5
%
```

Note that small area imaging jobs can be supplied to users as a single image file (in Starlink NDF or disk fits format) via anonymous ftp. Limitations in network speed make transferring a 2 Gbyte pixel map via the Internet somewhat impractical.

4 The data format

Data files come in the following formats:

4.1 ASCII files

'Normal' text files (ASCII files) are generally small and can be inspected as follows:

```
% more hkfile.txt
      'JOB.JOBNO'              'UKJ128'
      'JOB.DATE-MES'           '1998:02:18'
      'JOB.TIME'               '16:48:10'
      'JOB.INSTRUME'           'SuperCOSMOS I'
      'JOB.ORIGIN'             'Royal Observatory Edinburgh'
      'JOB.SOFTWARE'           '/home/scosdev/v028'
      'JOB.OPERATOR'           'ebt'
      'JOB.USER'               'htm'
```

```
      'JOB.USERREF'            '1'
      'JOB.UORIGIN'            'ROE'
      'JOB.UCOUNTRY'           'uk'
      'JOB.COMMENT'            'An object catalogue of the Southern Sky'
      'JOB.IAM_FILE'           'iam'
      'PLATE.TELESCOP'         'UKST'
      'PLATE.TELTYPE'          'SCHM'
          .
          .
%
```

The global measurement parameters contained within the house–keeping file are described in more detail in 'The SuperCOSMOS data processing guides: Introduction'. Note that the command

```
% hdstrace hkfile > hkfile.txt
```

can be used to produce a text version of the file `hkfile.sdf` if required.

4.2 Small image files

Small area images can either be supplied as a STARLINK N–dimensional Data Format (NDF) with default filename `*.sdf`, or as a disk fits file, default filename `*.fits` if STARLINK software is not available at your site. The former allows input and manipulation in the host of STARLINK applications (*e.g.* KAPPA, Currie 1992). To inspect the contents of an NDF, use the command `hdstrace`:

```
% hdstrace Hale_Bopp

HALE_BOPP  <NDF>

   DATA_ARRAY     <ARRAY>         {structure}
      DATA(1024,1024)  <_REAL>       9987,9074,6984,3327,4424,6502,9730,
                                     ... 10475,11108,11150,12918,11767,13962
      ORIGIN(2)      <_INTEGER>      1,1

   TITLE          <_CHAR*64>      'Super-COSMOS MM data for job mmdata'
   LABEL          <_CHAR*32>      'SuperCOSMOS transmission'

End of Trace.

%
```

4.3 Binary files

The large IAM (up to ~ 100 Mbyte) and MM (up to 2 Gbyte) files are supplied as direct access, unformatted integer binary files, since ASCII versions would be prohibitively large. Unfortunately, Sun Microsystems 'SPARC' and Digital 'Alpha' workstations (the two most common systems in use in astronomy) store integers in opposite byte order: the Digital convention is least significant byte first in a two–or–more byte word; the Sun convention is most significant byte first (note that PCs running Linux use the same convention as Digital Alphas). Consequently, users may specify on which type of workstation they will be handling their data when requesting scans. The default is Digital Alpha format since the SuperCOSMOS workstations are of this type. To check if your data are in the correct format for your workstation and for further information (*e.g.* how to swap the byte order if required) see the introductory guide; information on FORTRAN compiler options and record lengths is also given there.

We will now discuss the arrangement of data in the IAM and MM files.

4.3.1 IAM data file

For every image detected in the IAM, 32 image parameters are written to a record in the file `iam.srtrd`. See Appendix A for a list of the parameters and their units. Each parameter is stored as a 4 byte word, consequently each record in the file is 128 bytes long. Furthermore, the file is sorted into lane order (see Figure 1). The first record in the IAM file will contain the parameters of the image in lane 1 with the smallest Y value; subsequent images will have increasing Y values until the last image in lane 1 will be followed by the first in lane 2, again having the smallest Y value, and so on. The following simple FORTRAN 77 program illustrates this:

```
      program dotplot
c
c  Plot a point for images in an iam.srtrd file
c
c  To compile: alpha% f77 dotplot.f -L/star/lib `pgplot_link` -o dotplot -vms
c                sun% f77 dotplot.f `pgplot_link` -lsocket -o dotplot -xl
c
      implicit none
      integer i,buff(32)
      real x,y

      open (unit=1,file='iam.srtrd',status='old',access='direct',
     :form='unformatted',recl=32,readonly)

      call pgbegin(0,'?',1,1)
      call pgscf(2)
      call pgenv(0.0,360.0,0.0,360.0,1,1)
      call pglabel('X\dSuperCOS\u / mm','Y\dSuperCOS\u / mm',
     :'Dotplot of file iam.srtrd')

      i=0
10    i=i+100
      read (1,rec=i,err=20) buff
      x=real(buff(11))/1.0E5
      y=real(buff(12))/1.0E5
      call pgpoint(1,x,y,-1)
      goto 10
20    close (1)
      call pgend
      end
```

There are several points to note about this program:

- The 'open' statement must specify the record length in 4 byte words for compilation on an Alpha, or in bytes (*i.e.* `recl=128`) for Sun compilation. However, if as instructed in the above example you use the `-xl` option on a Sun, the record length is specified in 4 byte words. This option is needed, for example, when running PGPLOT and specifying special characters in strings using the backslash (\) character, as is the case in this example program. See the introductory guide for more details.
- The program skips through the file and plots one point for images spaced every 100 records for speed; as it runs, you will see the scanning lanes being marked out in the plot.
- The program reads through the file until it encounters an error, and that error is assumed to be that an attempt has been made to access a record beyond the end of the file. Strictly speaking, a program such as this should check the input/output status at each read, watch for a 'non–existent' record error, and report any other unexpected errors.

4.3.2 The MM data files

Again, have a look at Figure 1. For a scan made up of n lanes, there will be n files called l1, l2, l3 ...ln in the subdirectory `mmdata/`. Also in this directory is a header file (called `header`) which describes the MM data:

```
% cd J12345/mmdata
% cat header
nlanes 28
lwidth 1152
lheight 32256
dtype I2D
%
```

The character string following `dtype` has the following valid types: I2D, I2S, I4D, I4S, R4D and R4S, where I and R refer to integer and real, the number is the number of bytes per pixel, and D and S refer to Digital and Sun byte ordering and floating point type. In practice, I2D and I2S will be the only types for user data. The numbers `lwidth` and `lheight` are both in pixels. Each direct access file l1 ...ln will therefore consist of `lheight` records, each record having length `lwidth`×2 bytes. Refering to Figure 1, the first 2 bytes of the first record of file l1 will be for the lower left–hand pixel of the map; the last two bytes of the 32,256$^{\mathrm{th}}$ record of file l28 would be the top right–hand pixel in the map *et cetera*. A FORTRAN 77 code would need the following lines to read from lane 1:

```
      integer i
      integer*2 buff(1152)
      .
      .
      open (unit=1,file='l1',status='old',access='direct',
     :form='unformatted',recl=576,readonly)
      .
      .
```

```
do i=1,32256
 read (1,rec=i) buff
  .
  .
enddo
.
.
```

Again, note the following:

- For Sun compilation, `recl=2304` in the 'open' statement unless using the `-xl` option, in which case `recl=576` is appropriate.
- The integer array `buff` has data type `integer*2` (for 2 byte integers) as opposed to the FORTRAN 77 default `integer*4` as in the previous example for reading the IAM file.

The 2 byte pixel values themselves are transmissions in the range 0 to 32767 ADU from the CCD detector. In the default orientation, within each lane file the records run from low to high DEC, subsequent lanes being west (*i.e.* at lower RA).

5 The quality control histograms

Supplied with your data tape will be a set of plots pertaining to the plate measurements. The title page gives some global parameters for the measurement. The following page contains four histograms showing the number of images having a given magnitude, area, ellipticity and orientation between minimum and maximum values for the measurement. The final three plots give details of the astrometric solution. Full details of the latter can be found elsewhere (Davenhall 1993); however we give a brief description here. The astrometric solution is based on a few hundred catalogue standards scattered over the plate. The first plot shows those included in the final fit (note that a large number are flagged as 'rejected' because the selection program chooses all objects within a radius of the plate centre that contains the plate corners). The size and direction of the 'lollipop' vectors indicates the residual of each included star in the fit. The next plot shows a histogram of the residuals; finally a plot of the SuperCOSMOS magnitude parameter versus the catalogue magnitude is plotted.

6 Example programs

In this Section, the following example programs are described:

- `brisel` selects the 1000 brightest images from an IAM file;
- `briplot` makes an ellipse plot of the images selected by `brisel`;
- `ncharts` makes ellipse plots of small areas around a specified list of objects;
- `pixplots` makes plots of pixel data around a specified set of objects;
- `nseq` searches an IAM file for a given list of objects;

- `ilist` lists the 32 image parameters for objects indexed in a pointer file (as produced by, for example, `nseq`);
- `lstcharts` makes numbered ellipse plots around a specified set of objects, and outputs a numbered list of parameters.

The example programs listed in this document, and others, are available by anonymous ftp:

```
% ftp ftp.roe.ac.uk
       .
       .
       .
Name (ftp.roe.ac.uk:xyz): anonymous
331 Guest login ok, send your complete email address as password
Password:
       .
       .
       .
ftp> ascii
ftp> cd pub/nch/code
ftp> mget *.f
ftp> close
ftp> quit
%
```

In all cases, compilation information is given at the start of the source listing. Feel free to take these codes, examine them and adapt them for your own purposes. Note that for convenience, the codes use the Starlink PGPLOT and SLALIB (Wallace 1992) libraries; in addition, some codes make use of NAG library calls.

Code `brisel` selects the 1000 largest (*i.e.* brightest) images in an IAM file, up to a maximum size specified:

```
% ./brisel
 File for input?
iam.srtrd
 File for output?
bright.dat
 Areal extent to search (xmin,ymin,xmax,ymax in mm):
0.0 0.0 400.0 400.0
 Maximum area (pixels), eg. 100000
10000
 No. of objects read from IAM file:      332422
 No. of bright images found:       4600
%
```

(note that `./` before the executable name is sometimes needed to run a program, depending on how the paths are set on a particular system). These images can then be plotted using program `briplot` (see Figure 2):

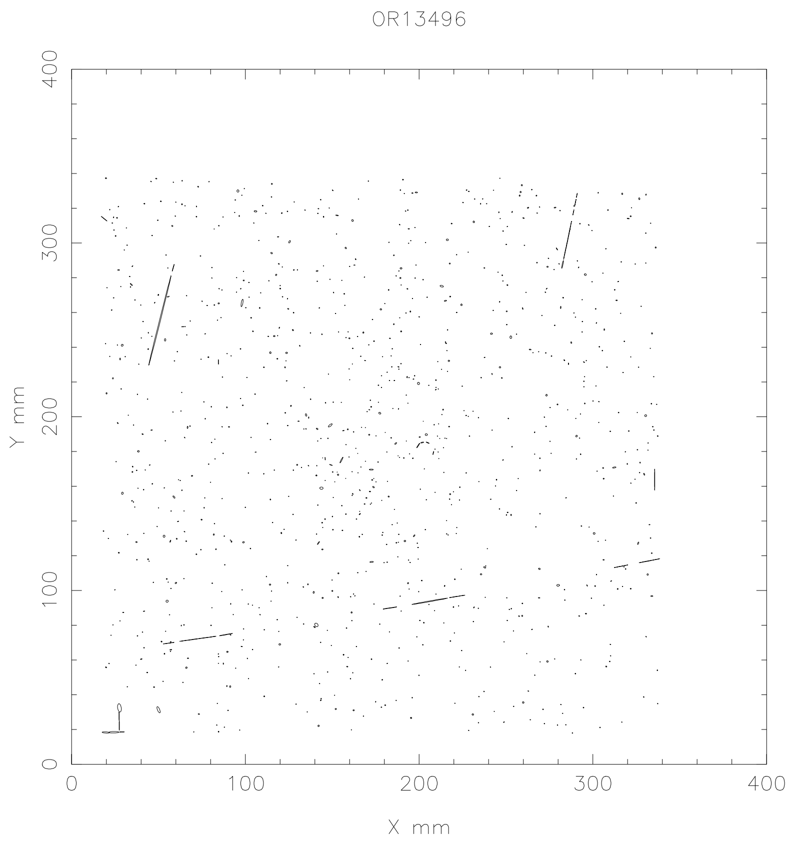

Figure 2: Example of displayed output from **briplot**.

```
% ./briplot
 Title for the plot?
OR13496
 Graphics device/type (? to see list, default xwindows):
xwindows
 Areal limits (xmin,ymin,xmax,ymax in mm):
0. 0. 400. 400.
 Flip plot? (0=no, 1=yes)
0
 Filename?
bright.dat
%
```

Two other useful programs are `ncharts` and `pixplots`. Both take a list file of RAs and DECs as input; the former makes ellipse plots of the image parameters in the specified area around each target, while the latter generates pixel plots from the MM data. Program `ncharts` takes an object list and can extract the RA and DEC (in the general form HH MM SS.S ±DD AM AS.S) from any position on each line in the file, one object per line (see Figure 3):

```
% cat objects.lis
  242587   IPMBD11     3 55 23.05 24 49  5.41    18.1
  319193   IPMBD20     3 49  4.82 23 33 39.65    18.1
  329044   IPMBD21     3 49 16.17 26 49  3.85    17.9  2.8
  329117   IPMBD22     3 49 33.06 26 50 43.17    18.0
                                  .
```

```
                                    .
                                    .
% ./ncharts
Filename containing co-ordinates:
objects.lis
Equinox (0=1900, 1=B1950, 2=J2000)
2
Start sexagesimal string decoding at which character?
20
IAM data filename:
iam.srtrd
Size of plot on a side (arcmin):
5.0
Graphics device/file:
xwindows
No. of plots across and down:
3 3
%
```

Program `pixplots` produces pixel plots around the RA and DEC centres in the input file (see Figure 4):

```
% ./pixplots
 Path-name of measurement:
/disk1/nch/UKR444
 Extent of scan (xmin,xmax,ymin,ymax mm):
16.22 338.78 16.22 338.78
 Filename containing centre co-ordinates of plots:
F444objects.lis
 Equinox of co-ordinates (0=1900, 1=B1950, 2=J2000):
2
 Start sexagesimal string decoding at which character?
1
 Size of square images to extract (arcmin):
5.0
 Graphics device:
xwindows
 No. of plots across and down:
3 3
 Gunsight (y/n):
y
%
```

Note that the extent of the scan can be obtained from the file `hkfile.txt`.

If you wish to search an IAM file for a set of objects (for example, to inspect the parameters for images you already know exist from another source) then use program `nseq`:

```
% ./nseq
 Input file containing co-ordinates:
objects.lis
```

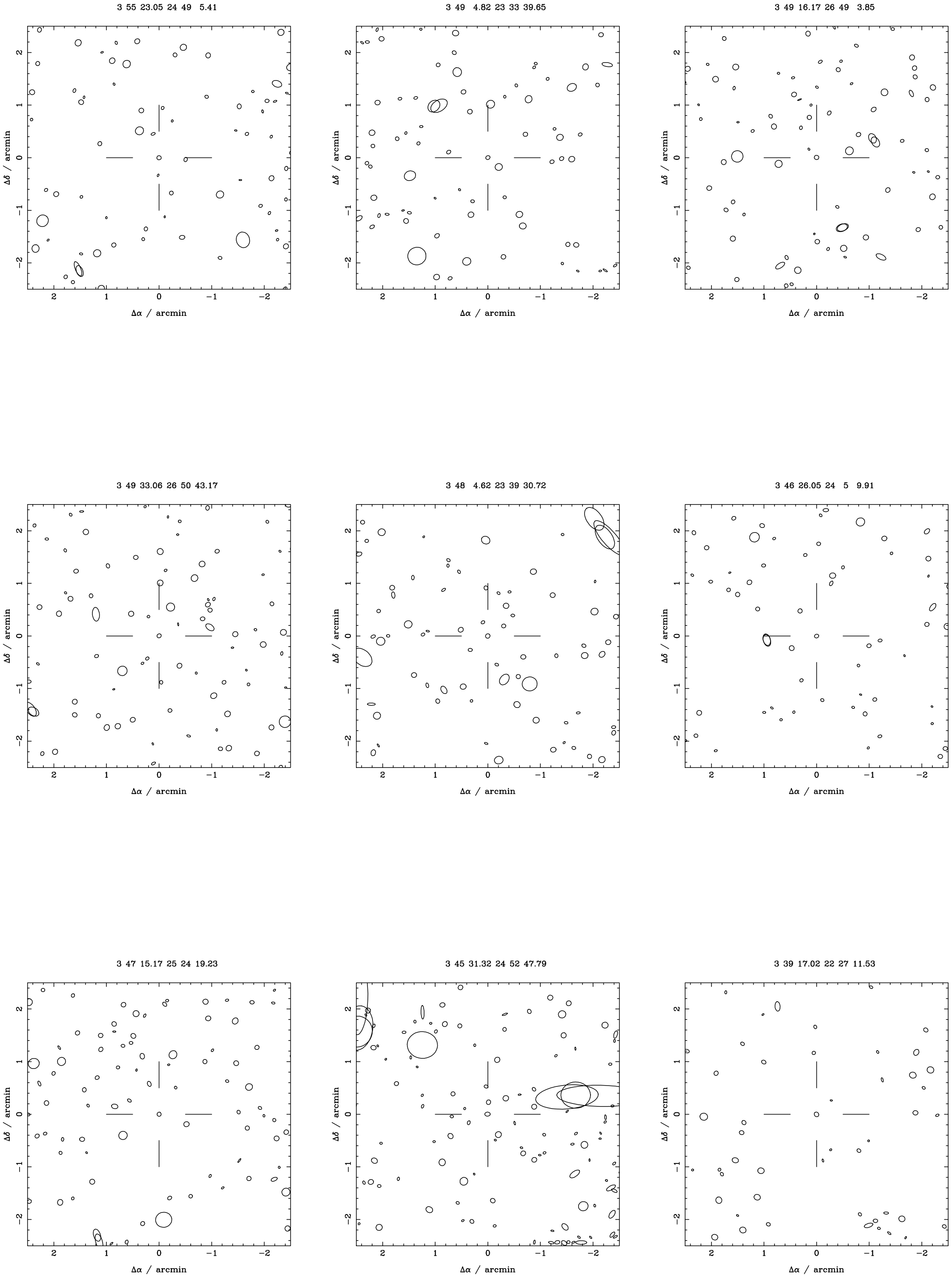

Figure 3: Example of displayed output from **ncharts**.

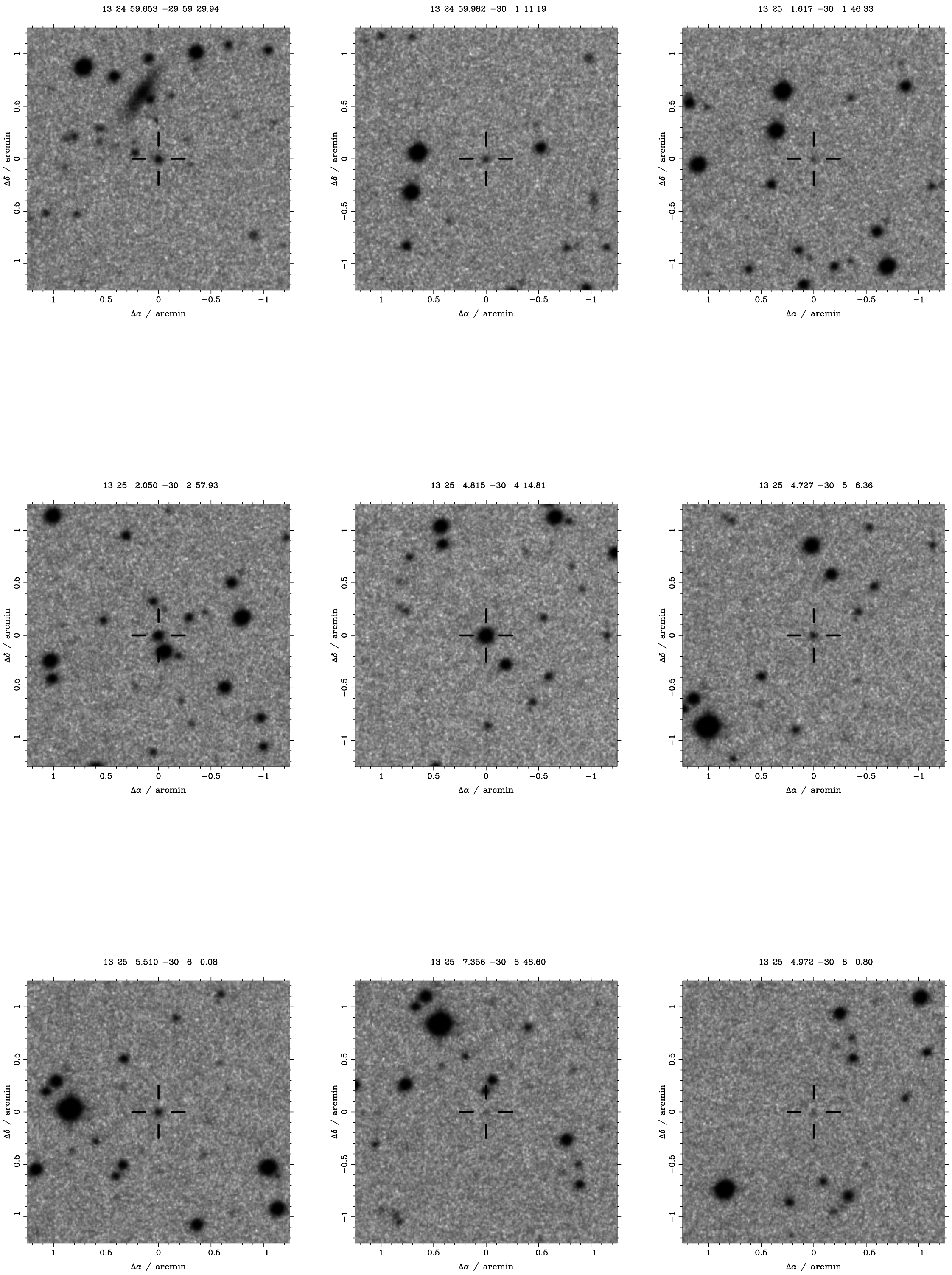

Figure 4: Example of displayed output from **pixplots**.

```
 Equinox (0=1900, 1=B1950, 2=J2000):
2
 Start decoding sexagesimal co-ordinates at which character?
20
 IAM file to be searched:
iam.srtrd
 Search radius (arcsec):
20.0
 Output index file:
objects.idx
 No. of objects to search for:          9
 Number of records read from IAM file:     363927
%
```

This will produce two files. First, a log of the run will be produced in filename `logfile` containing a record of the nearest images located within the search radius, and their co-ordinates (note that any existing file called `logfile` will be overwritten) – continuing from the last example:

```
% more logfile
Input file of objects to be searched for: objects.lis
IAM file searched: iam.srtrd
Search radius (arcsec):      20.00

 Number of records in IAM file:       363927

 Nearest matches:

 Object #       RA (J2000.0) DEC       Offsets to match     No.of   Pointer
                  (IAM file)              (arcsec)          finds   to IAM
       1    3 55 23.052 +24 49  5.33      -0.03     0.08        1     67515
       2    3 49  4.816 +23 33 39.79       0.05    -0.14        2    160291
       3    3 49 16.174 +26 49  4.15      -0.06    -0.30        2    170235
                                             .
                                             .
                                             .
%
```

Note that the program should run quickly, even if a large number of objects are searched for; also, the number of objects found within the search radius is indicated so as to flag any possible confusion problems due to inaccurate input co-ordinates. In addition to the logfile, a compact direct access binary file of pointers (those numbers in the final column of the logfile) is output, indicating where in the IAM file the nearest image (within the search radius) is located. This binary index can be subsequently used for input into other codes. For example `ilist` will list the parameters for those images (see Appendix A for an explanation of the 32 image parameters):

```
% ./ilist
 IAM file:
iam.srtrd
 Index of pointers:
objects.idx
```

```
 Image number:           1
 102705733  43315837   7430038   7440036  21796990  21805990        59
    823781    -18169  16600506   7434906  21801783      4605      4211
       164      3764      3454       177         0        75         0
         0         0         0         0         0         0         0
         0         0         0         0

 Image number:           2
         .
         .
         .
%
```

– this code could easily be modified to output selected parameters in a more readable form.

Finally, code `lstcharts` allows the user to produce charts like those from `ncharts` but with objects numbered and some parameters of the numbered objects output (again in a file called `logfile`):

```
% ./lstcharts
Filename containing co-ordinates:
objects.lis
Equinox (0=1900, 1=B1950, 2=J2000)
2
Start sexagesimal string decoding at which character?
20
IAM data filename:
iam.srtrd
Size of plot on a side (arcmin):
5.0
Graphics device/file:
xwindows
% more logfile
 Object list for chart number           1
001     3 55 12.665  +24 46 35.71 -19.609
002     3 55 27.804  +24 46 36.28 -23.382
003     3 55 30.126  +24 46 43.50 -18.954
004     3 55 18.306  +24 46 57.56 -18.627
005     3 55 29.813  +24 47  1.38 -21.725
                     .
                     .
                     .
```

– the numbered objects correspond to those in the plots that are produced (see for example Figure 5).

Further code examples are provided in the data processing guides, which also detail various data reduction procedures.

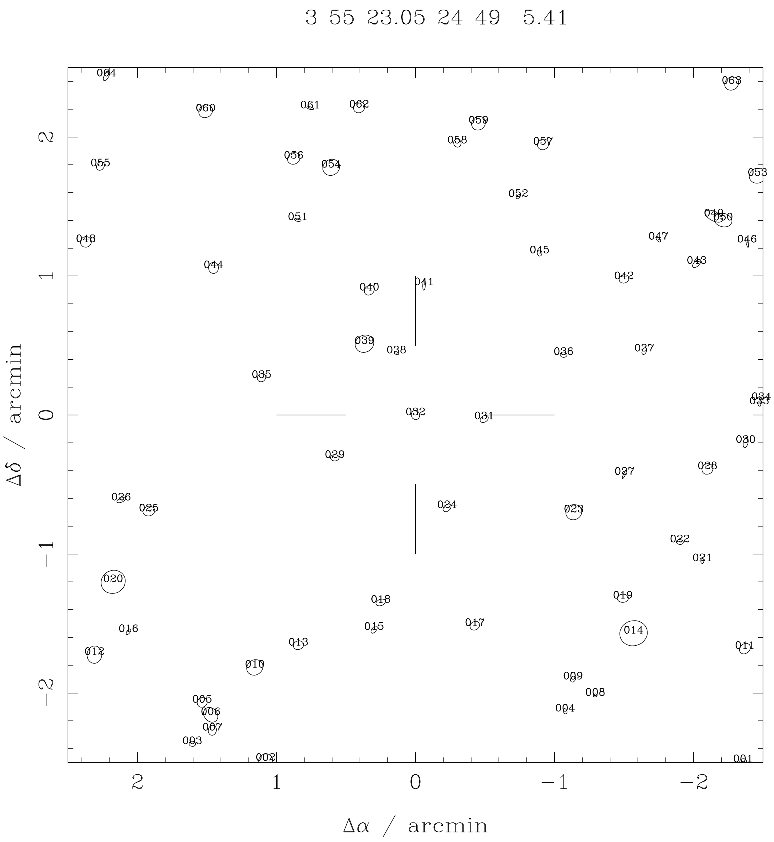

Figure 5: Example of displayed output from **lstcharts**.

7 Further reading

Much more detail concerning SuperCOSMOS data will be available in the set of data processing guides (currently in preparation) where information on astrometry, photometry, star/galaxy separation, plate pairing and plate stacking will be included. For a general introduction to science from digitised photographic sky surveys, see MacGillivray & Thomson (1992) and references therein. The UK Schmidt Telescope Handbook (Tritton 1983) provides much useful information on photographic Schmidt plates; more theoretical, technical and reference information on photography and imaging science in general can be found in James (1977) and Irwin (1985), and references therein, respectively.

References

Beard S.M., MacGillivray H.T., Thanisch P.F., MNRAS, 247, 311
Bourne P.E., 1990, 'UNIX for VMS Users', Digital Press
Currie M.J., 1992, Starlink User Note 95.8: Kappa – Kernal Application Package
Davenhall C., 1993, SuperCOSMOS System Note 27.1: Converting plate positions into equatorial co-ordinates
Hambly N.C., Miller L., 1997, 'SuperCOSMOS Technical Manual'
Irwin M.J., 1985, MNRAS, 214, 575
James T.H. (ed), 1977, 'The Theory of the Photographic Process', Macmillan
Kelly B.D., 1995, SuperCOSMOS User Note 22.0: Accessing SuperCOSMOS IAM data.
MacGillivray H.T., Thomson E.B., 1998, 'SuperCOSMOS Operator's Manual'
MacGillivray H.T., Stobie R.S., 1984, Vistas. Astron., 27, 433

MacGillivray H.T., Thomson E.B. (eds.), 1992, 'Digitised Optical Sky Surveys', Astrophysics & Space Science Libraries vol. 174, Kluwer
Tritton S.B., 1983, UK Schmidt Telescope Handbook, Royal Observatory, Edinburgh
Wallace P.T., 1992, Starlink User Note 67.17: SLALIB – a library of subprograms

A The 32 SuperCOSMOS image parameters

The following is taken from Kelly (1995):

No.	Name	Type	Description	Units
1	RA	INTEGER	Celestial Right Ascension	10^{-8} radians
2	DEC	INTEGER	Celestial Declination	10^{-8} radians
			(Note: the usual celestial co-ordinate equinox is J2000.0)	
3	XMIN	INTEGER	Left extent	0.01 micron
4	XMAX	INTEGER	Right extent	0.01 micron
5	YMIN	INTEGER	Bottom extent	0.01 micron
6	YMAX	INTEGER	Top extent	0.01 micron
7	AREA	INTEGER	Total area	Pixels
8	IPEAK	INTEGER	Brightest intensity above sky	
9	COSMAG	INTEGER	1000×COSMOS magnitude ($-2.5\log\sum_i$)	
10	ISKY	INTEGER	Sky intensity at (XCEN_I,YCEN_I)	
11	XCEN_I	INTEGER	Intensity weighted X centroid	0.01 micron
12	YCEN_I	INTEGER	Intensity weighted Y centroid	0.01 micron
13	A_U	INTEGER	Unweighted semi-major axis	0.01 micron
14	B_U	INTEGER	Unweighted semi-minor axis	0.01 micron
15	THETA_U	INTEGER	Unweighted orientation	Degrees
16	A_I	INTEGER	Weighted semi-major axis	0.01 micron
17	B_I	INTEGER	Weighted semi-minor axis	0.01 micron
18	THETA_I	INTEGER	Weighted orientation	Degrees
19	CLASS	INTEGER	Classification flag (0=none, 1=galaxy, 2=star, 3=faint, 4=junk 5,6=subcondensations)	
20	P_A	INTEGER	Celestial position angle	Degrees
21	AP(1)	INTEGER	Area above areal profile level 1	Pixels
22	AP(2)	INTEGER	Area above areal profile level 2	Pixels
23	AP(3)	INTEGER	Area above areal profile level 3	Pixels
24	AP(4)	INTEGER	Area above areal profile level 4	Pixels
25	AP(5)	INTEGER	Area above areal profile level 5	Pixels
26	AP(6)	INTEGER	Area above areal profile level 6	Pixels
27	AP(7)	INTEGER	Area above areal profile level 7	Pixels
28	AP(8)	INTEGER	Area above areal profile level 8	Pixels
29	BLEND	INTEGER	Deblending flag (0 if not deblended)	
30	QUALITY	INTEGER	Quality flag	
31	COSMAGCAL	INTEGER	1000 × COSMOS magnitude converted into surface brightness	Mag arcsec^{-2}
32	SPARE	INTEGER	Currently unused.	

More detail is provided in the intoductory guide. In outline: the sky noise for an MM data set is determined near the plate centre, and a threshold is set for the IAM run (typically, thresholding is done at 2.3σ). Pixels having values above the threshold are examined for 8–pixel connectivity, and the parameters are calculated from intensity–weighted moments. For more details, see MacGillivray & Stobie (1984) and Beard et al. (1990); the latter also describes the 16–level rethresholding procedure that deblends multiple objects.