Instrument Science Group
Royal Greenwich Observatory

The Design and Use of the RGO CCD Flatness Scanner.

RGO TECHNICAL NOTE 110.

Simon Tulloch.

10th June 1997

CONTENTS

1. Introduction.

With the development of mosaic CCD cameras and very large area CCDs, it becomes increasingly hard to maintain device co-planarity, which is particularly important in fast optical systems. For example a CCD with a corner to centre bow of 50um , in an f3 beam , may defocus stellar images to a diameter of 17μm at the device corners even if perfectly focused at the centre. It is therefore important to be able to measure chip flatness and mosaic co-planarity before delivery of the camera system to the telescope. The measurement system should obviously be non-contact and should measure devices at their operating temperatures. Such a system has been developed at RGO and has been in regular operation for the last year. It was designed primarily to test the co-planarity of the INT Prime Focus Mosaic camera but has proved equally useful in characterising new SITe, Loral and EEV single CCD cameras. This technical note describes the design of this instrument and gives detailed instructions for its use.
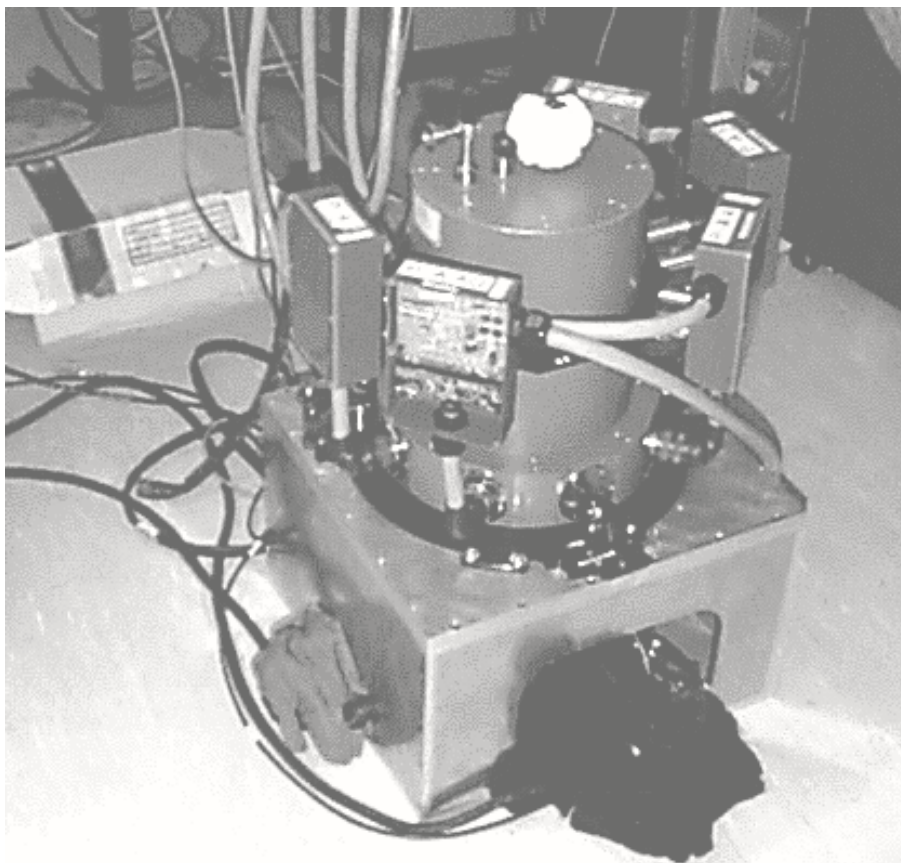

figure 1). The INT Prime Focus Mosaic Camera mounted on the Flatness Scanner

2. Principle Of Operation.

The scanner structure consists of a sturdy support box manufactured form 10mm aluminium alloy plate on top of which sits the CCD cryostat mounted window down. Inside the box are two orthogonal linear translation stages able to move a scan head in a horizontal plane with a repeatability of 1μm. The scan head projects a spot of light upward onto the CCD under measure. The translation stages have a range of 100mm in each axis so the projected spot can be positioned at any point of even the largest CCD mosaics planned by RGO. The top surface of the box is machined to directly accept the large 5 chip cryostat used in the INT PF mosaic project, however, standard RGO cryostats can also be mounted using a circular interface ring to which they are clamped using the normal 3 point kinematic mounts.
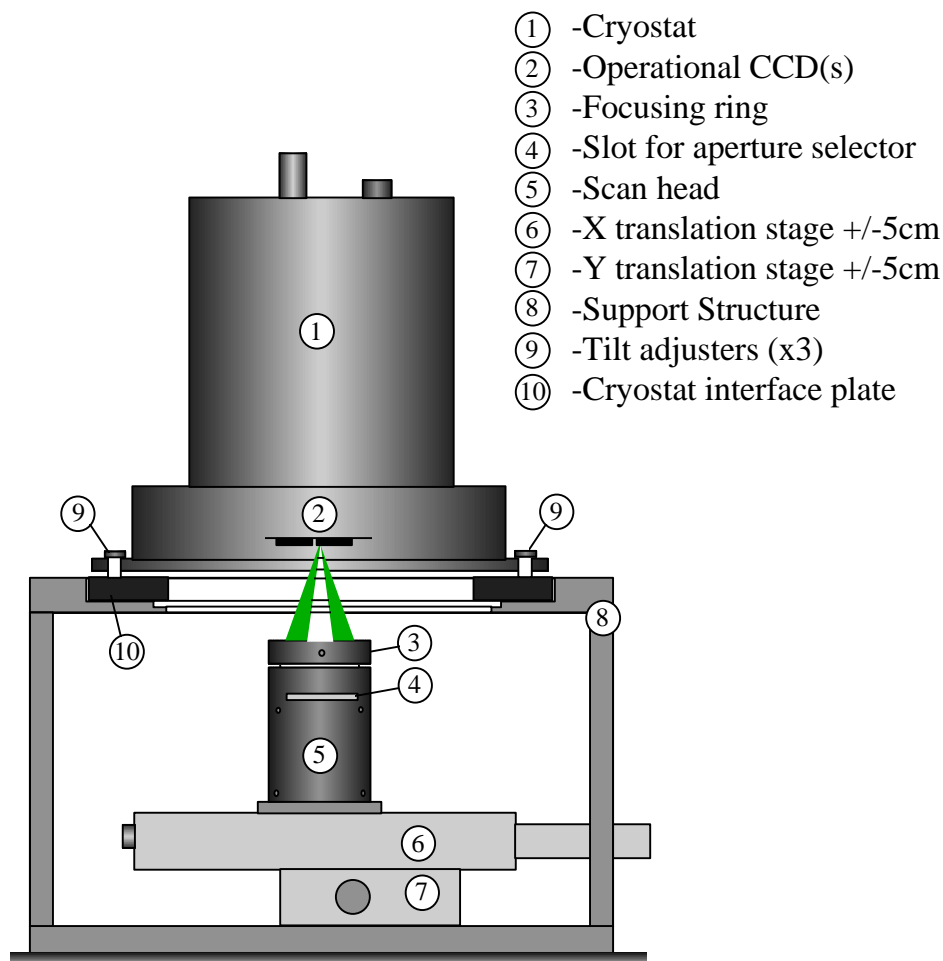
① -Cryostat
② -Operational CCD(s)
③ -Focusing ring
④ -Slot for aperture selector
⑤ -Scan head
⑥ -X translation stage +/-5cm
⑦ -Y translation stage +/-5cm
⑧ -Support Structure
⑨ -Tilt adjusters (x3)
⑩ -Cryostat interface plate

figure 2). Mechanical schematic of the flatness scanner

The principle of operation is derived from the Hartman Shutter technique used for focusing spectrographs. The scan head can project the spot image onto the CCD from two directions, depending on the position of a shutter inserted into the scan head. If the CCD does not lie precisely at the focal plane of the scanner then a double spot image will be recorded, the separation of the spots being proportional to the vertical distance by which the CCD is out of focus. The centroid of each

spot image can be determined using the IRAF 'splot' function to an accuracy of better than 0.02 pixels. The optical arrangement is shown in figure 3).

$\Phi = 28.5^0$

Plane A
Plane B
Plane C

58mm

40mm

LED illuminated pinhole

A CCD placed at each of these planes will record the following images:

A

B

C

x

h= the vertical distance of the CCD from the focal point of the scan head.

c1=centroid position along x axis of spot generated by aperture ①

c2=centroid position along x axis of spot generated by aperture ②

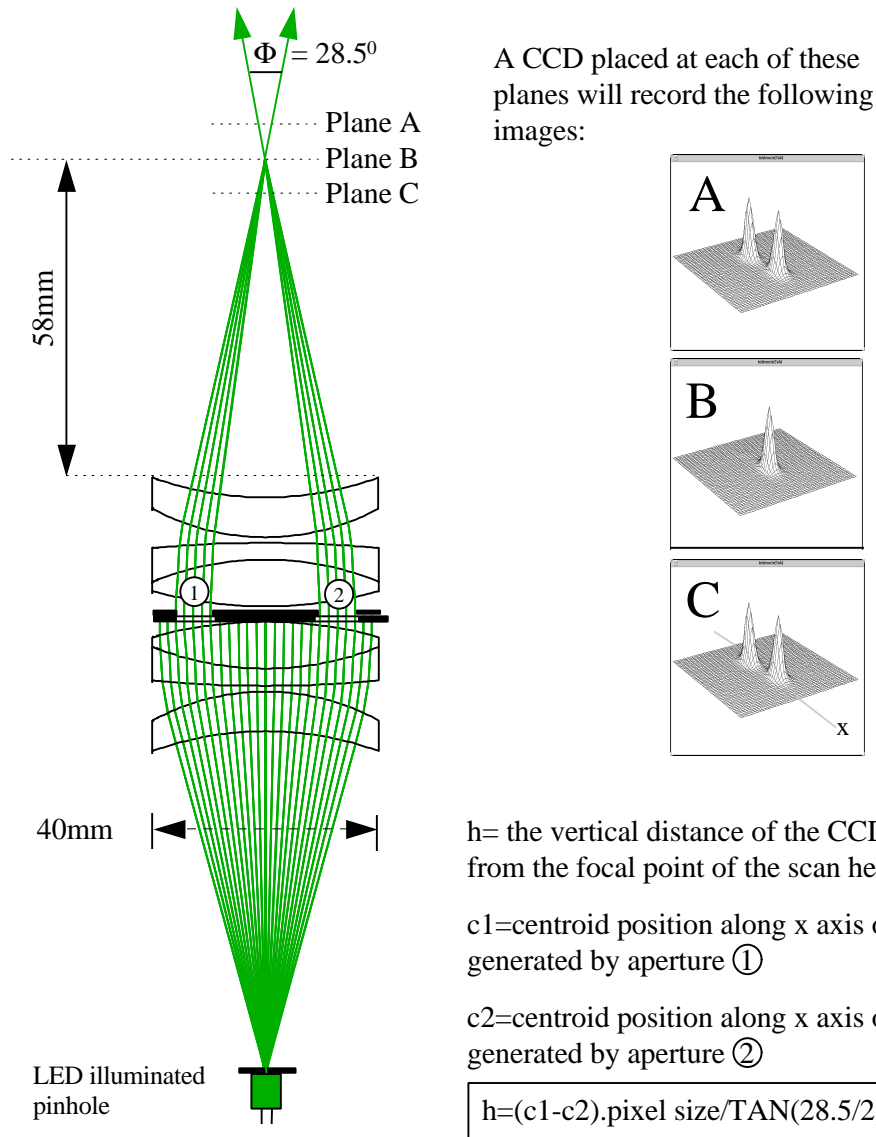$$h = (c1 - c2).\text{pixel size}/\text{TAN}(28.5/2)$$

figure 3). Optical layout of the scan head

In practice, centroiding can be done more accurately by recording each spot separately, this being essential if the two spots overlap. Surface maps can also be made by recording discrete spot images at many points on the CCD surface and measuring each image pair; a rather time consuming process. Far better is to scan the spot across the CCD surface during an exposure to produce a line image. Two such line images are equivalent to a string of individual spot images and contain all the information required to give height data right across a CCD to a horizontal resolution of 1 pixel. Clearly the orientation of the two shutter apertures with respect to the scan direction is important, in the above diagram the scan direction must be perpendicular to the page. An extension of this method is to record not just a single scan line but a whole series in a raster pattern, thus producing a surface map. Again, only two line images (or in this case multiple-line images) , are needed to record all the necessary information. The path scanned by the spot can be easily defined using a programmable translation stage controller.

3. Repeatability and Accuracy.

The centroiding algorithm in the IRAF function '**splot**' was able to locate the centre of the spot image with surprising accuracy. To quantify this a TEK1024 CCD was scanned. A small section of this profile at a point where the CCD surface was tangential to the plane of motion of the scan head is shown enlarged in figure 4).
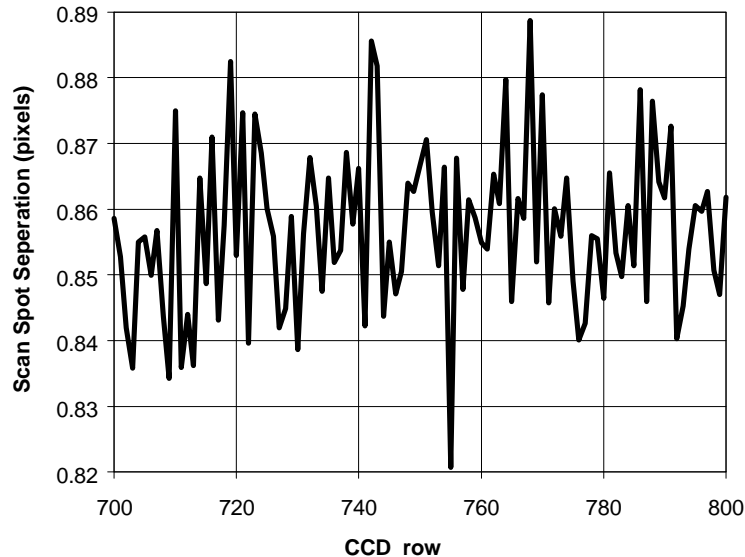

figure 4) . Error in Centroiding process

The vertical scale shows the *horizontal* separation of the two scan line centroids in the *plane of the CCD*. Since over this small region the CCD approximates to a plane, the variation in the data will show the error introduced by the centroiding process. Its value is 0.013 pixels, or 0.31μm RMS (in the *horizontal* plane). The vertical scale of the above graph indicates that the CCD at this point was very close to the focal plane of the scan head.

The absolute accuracy of the scanner relies very much on knowing the apex angle between the two scanner beams. This was measured by scanning an un-thinned GEC CCD that was known to be very flat. Using a slip gauge the CCD was then tilted by a precisely known amount relative to the scanner and a second scan performed. It was then possible to plot the graph shown in figure 5) giving the separation between the spot images as a function of distance from focus.

The following result was obtained by fitting a linear function to this graph :

   - If the CCD was above focus,
      its vertical displacement from the focal plane =**1.97**∗ Separation of spot centroids

   - If the CCD was below focus,
      its vertical displacement from the focal plane =**1.99**∗ Separation of spot centroids

In this measurement it was only possible to measure this relationship within about 100μm of the focal plane.
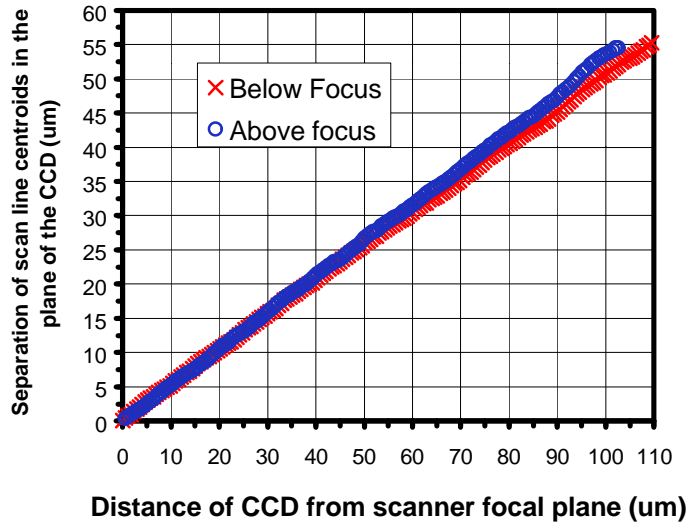
figure 5). Linearity of scanner : Centroid separation versus distance from focus.

The repeatability of the scanner was measured by profiling the same chip twice in the space of a few minutes. Figure 6) shows the differences between the two sets of data. The RMS deviation of this data is 0.9μm.
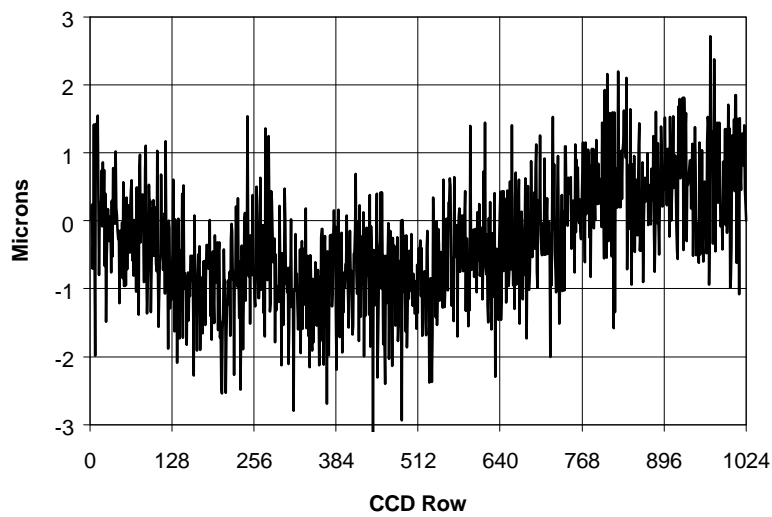

figure 6) . Scanner Repeatability : Difference in height data produced by consecutive scans.

4. How to Use the Scanner

4.1. Set-up.

The two translation stages should first be connected to plugs 1 and 3 on the back panel of the programmable controller. A BNC cable should then be plugged into the socket at the base of the scan head. This cable will carry the current for the internal LED illuminator. The other end of this cable is connected to the CCD controller interface box which is in turn coupled to the CCD controller using the custom temperature servo cable. This custom cable allows the preflash signal from the front panel temperature connector of the CCD controller to trigger the LED inside the scan head. For a full description of both the interface box and the cable, see appendix A.

The CCD cryostat is mounted on top of the scanner structure and the three kinematic clamps engaged. The top plate of the scan box should be freely rotated until the axes of the translation stages are approximately parallel to the rows and columns of the CCD. The three kinematic posts on the faceplate of the cryostat should be adjusted to give a gap of 3mm between the 3 'ears' on the cryostat lid and the top plate of the scan box. If the focusing ring on the scan head is then rotated anti-clockwise from its lowest position by three turns then the system will be roughly in focus assuming that the cryostat window-CCD distance is 11mm (Its nominal design value as measured optically).

The 'focus plate' aperture selector should then be inserted into the scan head slot. This plate contains two apertures and is used during fine focusing of the system.

The whole assembly can then be covered over with several black blankets to exclude ambient light, ready for the fine focusing procedure. The actual flatness scan should be performed with the room lights out as well.

4.2. Programming the Translation Stage Controller.

The Ealing Digital Positioning System (DPS) controller is able to simultaneously drive 3 translation stages although only two are used in the flatness scanner. The positional precision is 0.5μm, the repeatability 1μm. When moving across their full ranges the stages do not deviate from a straight line motion by more than +/- 1μm. Commands to the controller can be directly entered from a keyboard or can be in the form of programs which are stored in non-volatile memory. The battery that supplies this memory has a lifetime of several weeks so programs may be lost if the controller is not used for some time.

The translation stages have no absolute encoding so after power-up the controller defines their initial positions as zero. The stages have a total translation range of 10cms, microswitches indicate when these limits have been reached and automatically shut down the motors. There is also a hall effect sensor used to define the centre position of the stages but using it is very time consuming and not very useful for this application. A more useful convention is to define the optical centre of the CCD chip being scanned as the origin of the co-ordinate system, this being found by examination of the CCD images.

When switched on the controller takes about 10 seconds to configure after which the main menu will appear on the screen. This menu can be returned to at any time by using the escape key. Menu option 3 allows the user to directly input commands from the keyboard, option 4 allows programs to be created, selected, edited , deleted and run. Directly input commands have the same syntax as those used within programs. Multiple commands can be entered on one line if separated by commas.

These commands will generally be executed in parallel (although an extra command can be introduced to force sequential execution) so that both stages can be moved simultaneously. A selection of useful commands are summarised below :

Xmaccc : Move stage number X to absolute co-ordinate ccc

Xmrccc : Move stage number X to relative   co-ordinate ccc

XsvV : Set translation speed of stage X to velocity V (0<V<256, default=100)

wsX   : Wait for movement of stage X to stop before executing next command

Xdhccc : Re-define current co-ordinate of stage X =ccc

mdC   : Define a macro with identification number C

emC   : Execute a macro with identification number C

A macro can be defined to condense a number of commands that are often used together. Co-ordinates can be negative as well as positive. Co-ordinate units are 0.5μm. Co-ordinates increase positively as the stage moves away from its drive motor. With the translation speed=100, the stage will move at approximately 2.5mm/s.

Examples of Direct Commands

Move stage 1 by 1mm, wait for it to stop then move stage 2 by -1cm :

1mr2000,ws1,2mr-20000 <RETURN>

Move stage 1 to absolute position 0 whilst moving stage 2 to absolute position -5000 :

1ma0,2ma-10000 <RETURN>

Direct commands are suitable for doing simple single axis profile scans of CCDs , e.g. if all that is required is a surface profile of say the long axis of a chip. If more extensive 2D mapping of the CCD is needed it is best to control the stage motion using program mode.

To enter program mode select option 4 from the main menu. After selecting the subsequent 'create' option a program can then be typed in. The first line should contain a '$' sign followed by the name of the program. The rest of the program can then be entered line by line. Once editing is complete the escape key saves the program and returns the user to the program mode menu from where it can be executed.

Example of a Program

The program listed below was used to produce the scan pattern shown in figure 7. Prior to running the program the co-ordinate origin was adjusted using the 'dh' command to coincide with the centre of the CCD. Scan axis 1 was adjusted to be parallel with the CCD columns. Prior to running the program and exposing the line image, the scanner was positioned, using direct command, at the co-ordinates shown on line 2. Scans must start and stop away from the CCD imaging area , otherwise the spot velocity and therefore the depth of exposure of the line image will not be constant.

$scan1
2ma-24192,1ma-60000
md1,1mr120000,ws1,2mr6912,ws2,1mr-120000,ws1,2mr6912,ws2
em1,em1,em1,em1
1ma-60000
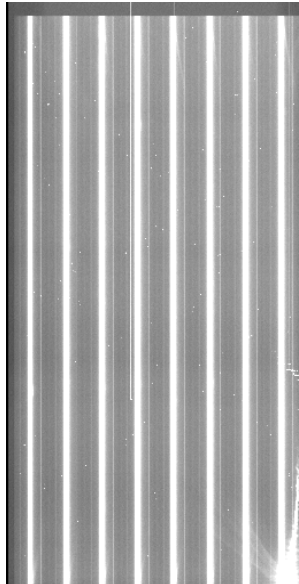
CCD measured   2048 x 4096 x 13.5 µm pixels.

figure 7). Scan line image produced by program shown above.

The programmable translation stage controller has two idiosyncrasies. The first is due to incorrect positioning of the motor drive cards in its internal slots. Three cards are installed which allows the user to write software to control motors designated 1,2 and 3. Control signals to motor 1 are correctly routed to rear panel connector 1, however, signals to motors 2 and 3 are routed to connectors 3 and 5 respectively. When using the flatness scanner connect the lower translation stage to connector 1 and the upper stage to connector 3. You should then use commands to control motors 1 and 2.

The second idiosyncrasy lies in the co-ordinate system used by the controller. The front panel screen keeps the user informed of the current co-ordinates of the translation stages. When the user inputs co-ordinates to move the stages, the required units are 0.5µm e.g. the command "**1ma2000**" will move translation stage 1 by 2000 x 0.5µm = 1000µm. The front panel display will then indicate the co-ordinates of this new position but the units will be in microns *not* 0.5 microns.

4.3. Focusing and De-tilting.

The scanner must be correctly focused and its plane of motion made approximately parallel to the CCD surface prior to flatness measurements being made. This is generally the most time consuming part of the whole measurement process.

The aperture plate labelled 'focus-plate' should already have been inserted into the scan head during set up. When the CCD is out of focus the spot image produced with this aperture plate resolves into two sub-images, the separation of which indicates the degree of focus error. This plate is *not* used for subsequent profile scans.

Ideally the CCD should lie within 100μm of the focal point of the scanner. Focus can be achieved by adjustments to the kinematic capstans on the cryostat faceplate and the focusing ring on the scan head. Before this can be done the orientation of the CCD relative to the axes of the translation stages must be determined. It is also necessary to determine from which corner the CCD is being read out. This can be done by recording two short (say 50ms) preflash exposures with the CCD in binned mode to save time. After the first of these exposures both translation stages should be moved by about 1mm. By observing the motion of the spot between the two images it will be possible to determine the physical orientation of the CCD.

The translation stages should then be further adjusted to bring the projected spot to the physical centre of the CCD and the '**dh**' command entered into the controller to define this as the origin of the scanner co-ordinate system.

It is now necessary make fine adjustments to the rotational orientation of the CCD to ensure that one of the translation stages lies precisely parallel to the CCD columns. If the horizontal co-ordinate of the centroid of a vertical scan line changes by more than 50 pixels between the top and the bottom of the image then the analysis software will be unable to produce a surface profile. In practice this degree of parallelism is easily achieved and the user should really aim to reduce it below 20 pixels. The parallelism can be measured by moving the scan head off the edge of the chip near the serial register and approximately in line with the central column. A 30s pre-flash exposure should then be commenced and the projected spot scanned up the full length of the CCD during integration. The tilt of the resulting line image will allow the user to determine in which sense the cryostat needs to be rotated to improve parallelism. The amount of rotation required is best found by trial and error. Generally after about two iterations the adjustment should be complete.

The scan head should then be brought back to the CCD centre and another spot image examined in greater detail using a windowed un-binned 50ms pre-flash exposure. Unless by chance the scanner is already perfectly in focus, what should be revealed is a double spot image from whose profile it should be possible to determine the sense and magnitude of the focus error. The separation of the two spots can be measured on the VME system using the cursor and the '**x-prof**' or '**y-prof**' commands. Once the separation is established in pixel units it should be converted to microns. The vertical error is then simply established by multiplying by 2, the approximate f-number of the scan head optics.

Both the kinematic capstans on the cryostat faceplate and the focusing ring on the scan head have a 40TPI thread. One rotation of each will produce a vertical motion of 635μm. It is easier to make height adjustments using the focus ring since when adjusting the capstans all three must be adjusted together if the tilt of the cryostat is not to be changed.

Three spot profiles recorded with an EEV42 are shown in figures 8,9 and 10. The first profile indicates that the CCD is beyond focus and should be lowered, the second shows perfect focus, the third shows that the CCD is inside focus and should be raised. Although the two out of focus images are similar, the bottom one shows characteristic sub-peaks lying *outside* of the two main peaks. If these are seen then the scan head to CCD separation should be increased.

figure 8). CCD Cryostat too high

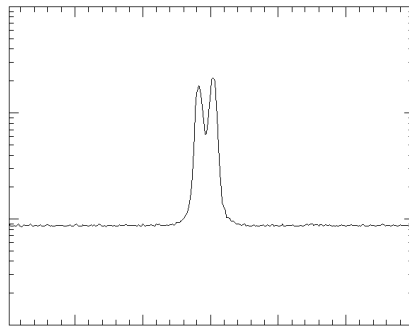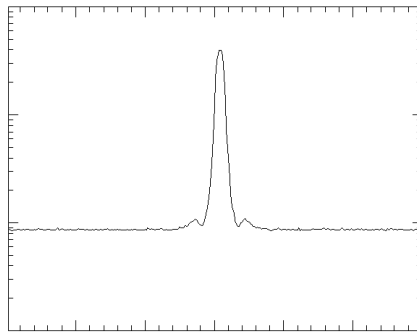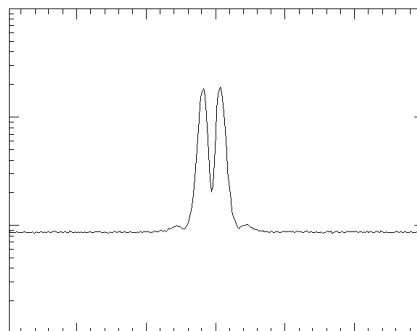**figure 9). CCD Cryostat in-focus**

figure 10). CCD Cryostat too low

Once the centre of the CCD has been focused the scan head should then be moved close to the mid-point of each edge of the device in turn and the approximate height calculated from the spot separation. This data can then be used in the next stage of the process : removing the tilt by kinematic capstan adjustment.

The diagram below shows the position of the cryostat faceplate mounting points and defines 2 axes for the de-tilting procedure.

Axis 2

Axis 1

115mm

57.5mm

A

B   C

CCD
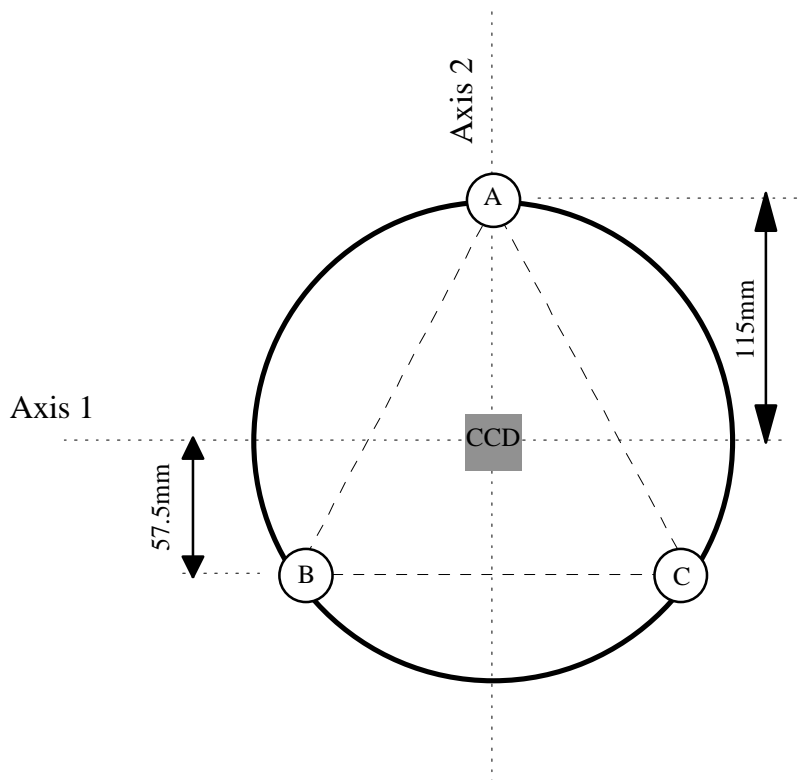
figure 11). The detector cryostat faceplate

The tilt in the system should be removed in a systematic way, first along one axis then the other. During this procedure the focus of the centre of the CCD should not be disturbed. If the system described below is adhered to, de-tilting the cryostat can be fairly straightforward. A small amount of residual tilt is not a problem since the analysis software is able to remove it from the data but ideally each edge of a CCD should be within 100μm of the focal plane of the scanner. Outside of this range the scanner will still function but its accuracy has not been checked beyond +/-100μm.

Rotating the CCD about axis 1.

Step 1 : Capstans B and C must be adjusted by **n** turns in the same sense.

Step 2 : Capstan A must be adjusted by 2x**n** turns in the opposite sense.

Rotating the CCD about axis 2.

Step 1 : Capstan B must be turned by **n** turns.

Step 2 : Capstan C must be turned by **n** turns in the opposite sense.

[In the above examples if **n**=1, the tilt of the cryostat will change by 0.623^{o} or 10.87mRAD.]

The table below shows what adjustment in height will be produced by capstan rotation at the edges of some commonly used CCDs.

CCD Type	Difference in height produced between top and bottom of CCD if n = 1.	Difference in height produced between left and right of CCD if n = 1.
TEK1024	267 µm	267 µm
SITe 2048	534 µm	534 µm
Loral 2K	334 µm	334 µm
EEV42-80	601 µm	301 µm

Example :

Spot measurements have been made at the mid point of each edge of an EEV42-80 CCD, the results are shown on figure 12. (Positive height readings indicate that the CCD is *above* the focal plane of the scanner).

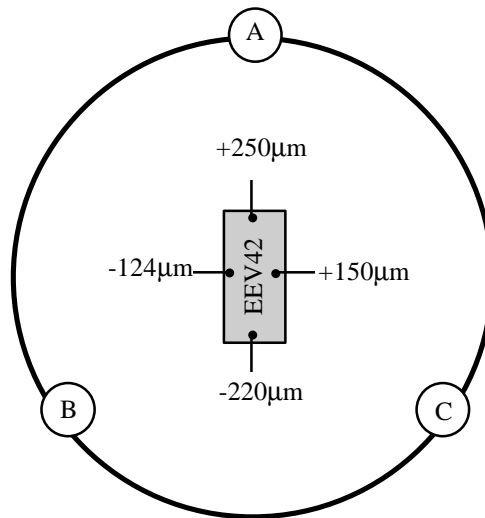**figure 12). An EEV42 CCD prior to de-tilting**

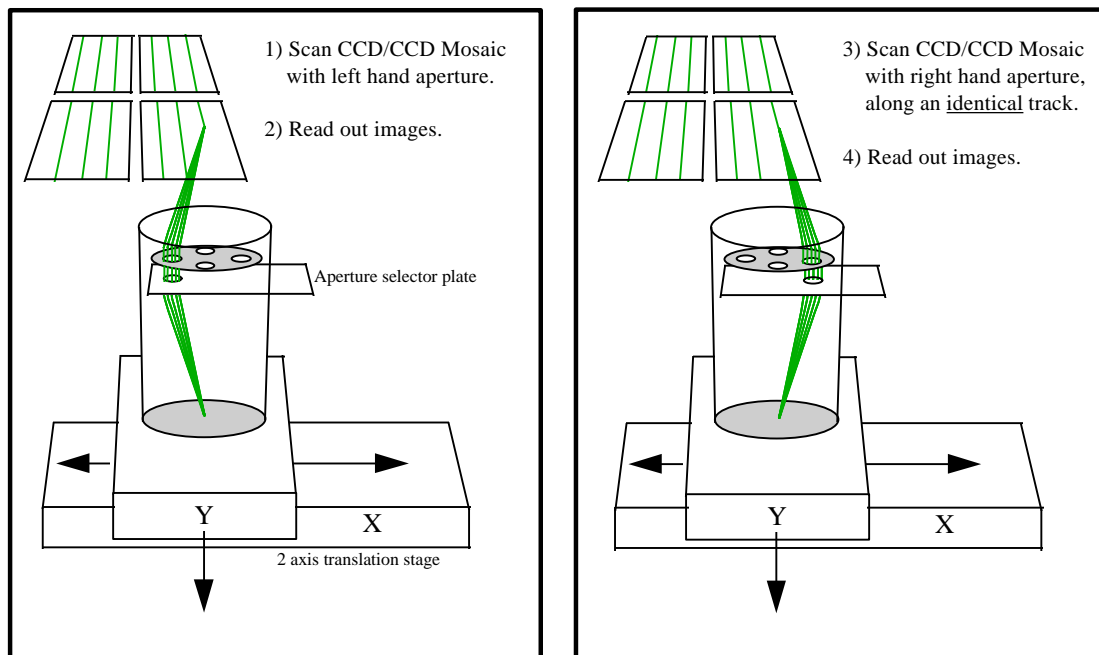The total tilt down a column is 250+220 = 470 µm. To correct this , capstans B and C must be rotated clockwise by 470/601 =0.78 turns and capstan A rotated 2*0.78= 1.56 turns anti-clockwise.

The total tilt along a row = 150+124 = 274 µm. To correct this, capstan B should be rotated clockwise by 274/301 =0.91 turns and capstan C rotated 0.91 turns anti-clockwise.

4.4. Performing a Scan.

A profiling run can involve a simple single axis scan down a CCD column or along a row, or it can involve a more thorough measurement where a surface map is created from a raster style motion of the scan head. The single axis scan is by far the simplest and is generally sufficient to predict possible focus problems when the CCD is used at the telescope. An IRAF script has been developed to analyse scan-line images, it is equally happy with single or multiple line images. Multiple scan lines must have uniform separation.

Once the required scan duration has been determined it should be converted into '*pre-flash*' units in preparation for a pre-flash exposure equal to the scan duration. An aperture selector plate is then inserted into the scan head ready for the first scan. This is then started by entering a '**fl**' command into the VME system and waiting for the red indicator light on the front of the CCD Controller interface box to come on. At this point the CCD begins integrating and the translation stage controller program can be run. After the first scan line image is safely stored on disc, the aperture selector plate can be changed to select the diametrically opposite aperture ready for the second scan. The line joining these two apertures *must* be perpendicular to the scan line axis.

1) Scan CCD/CCD Mosaic with left hand aperture.

2) Read out images.

Aperture selector plate

Y   X

2 axis translation stage

3) Scan CCD/CCD Mosaic with right hand aperture, along an identical track.

4) Read out images.

Y   X

For the case of a single CCD, two images result :

Taken with left aperture

Taken with right aperture

figure 13). Scan Procedure

The second scan line image should then be stored alongside the first. Note which image was taken with which aperture, more specifically which image was taken with the aperture closest to the readout amplifier of the CCD. Failure to do this will result in ambiguity as to whether the CCD has a concave or convex surface profile. The next stage of the analysis is shown in figure 14.
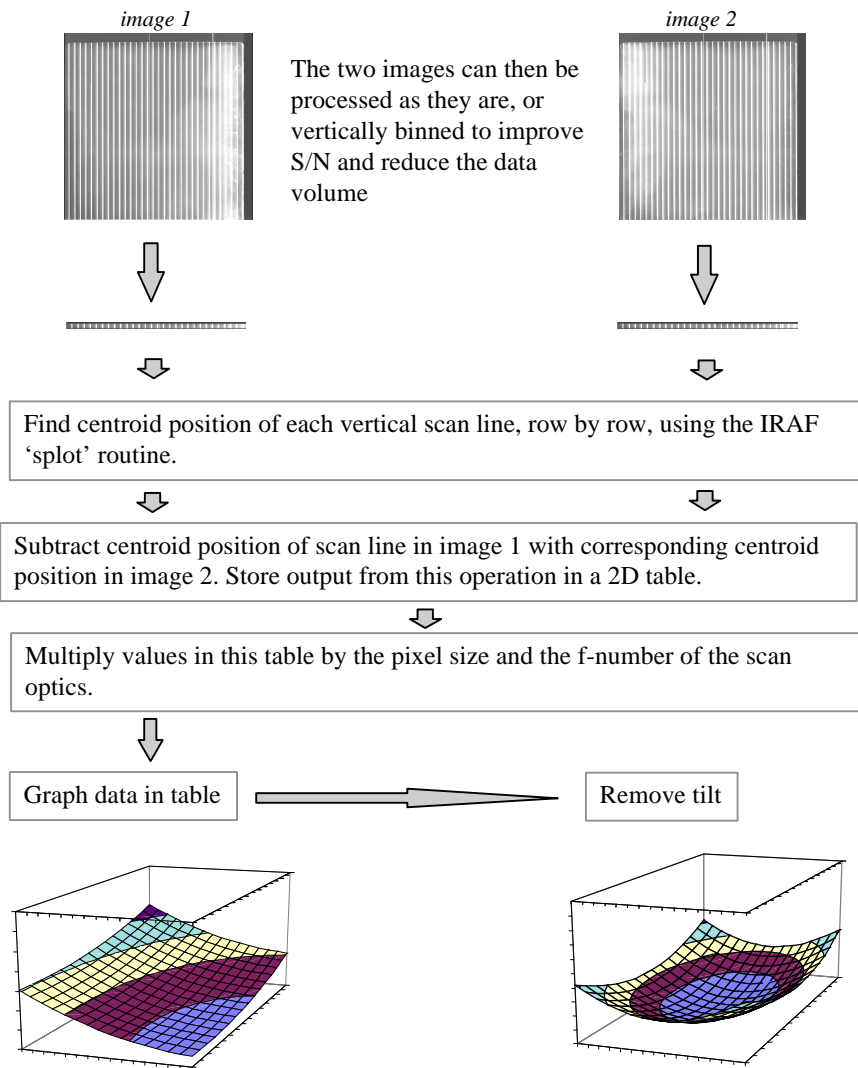
figure 14). Data Reduction Procedure
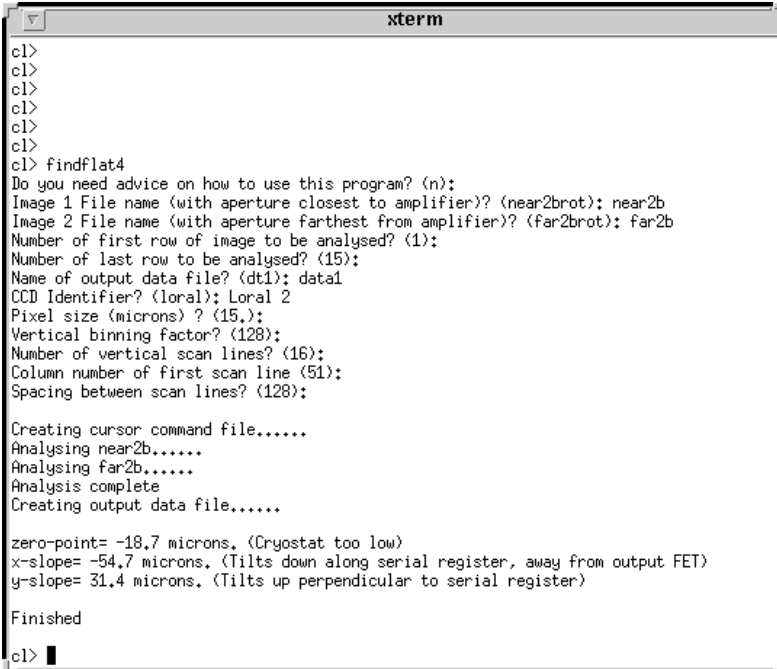
The first of these operations, image binning should be performed manually using the IRAF '**blkavg**' command. All the rest are performed automatically by the '/home/smt/iraf/flatness/findflat4' script in the case of scan lines parallel to image columns or by the '/home/smt/iraf/flatness/findflat5' script if the scan lines are parallel to image rows. Findflat4 is the preferred script since it operates faster and its results are easier to interpret. It takes the two scan line images and produces an EXCEL readable text file containing the height data of various points on the CCD surface. Final graphing of the results is best done from EXCEL. The use of 'findflat4' is described in the next section.

4.5. Data Analysis in IRAF using the 'findflat4' script.

Before running this script make sure that the '**noao'** and '**onedspec'** libraries are loaded into IRAF. The pair of line scan images should be first block averaged along an axis parallel to the scan lines to reduce the data volume and therefore the processing time. A binning factor of at least 16 is recommended. If the images contain scan lines parallel to image columns the following commands should be used from within IRAF :

<div align="center">

blkavg image1 image1 1 16 <RETURN>
blkavg image2 image2 1 16 <RETURN>

</div>

One of these images should then be displayed using '**saoimage'** or '**ximtool'** and '**findflat4'** executed. It should produce the following dialogue :

```
                              xterm
cl>
cl>
cl>
cl>
cl>
cl>
cl> findflat4
Do you need advice on how to use this program? (n):
Image 1 File name (with aperture closest to amplifier)? (near2brot): near2b
Image 2 File name (with aperture farthest from amplifier)? (far2brot): far2b
Number of first row of image to be analysed? (1):
Number of last row to be analysed? (15):
Name of output data file? (dt1): data1
CCD Identifier? (loral): Loral 2
Pixel size (microns) ? (15.):
Vertical binning factor? (128):
Number of vertical scan lines? (16):
Column number of first scan line (51):
Spacing between scan lines? (128):

Creating cursor command file......
Analysing near2b......
Analysing far2b......
Analysis complete
Creating output data file......

zero-point= -18.7 microns. (Cryostat too low)
x-slope= -54.7 microns. (Tilts down along serial register, away from output FET)
y-slope= 31.4 microns. (Tilts up perpendicular to serial register)

Finished

cl> █
```

The program finishes by graphing the profile of one of the scan lines as a check that all has gone well. The height data is left in the user specified file in ASCII format, in the above example it is file 'data1'. This file will contain one table of absolute height values at various points on the CCD and another table of 'de-tilted' data.

For scan line images with the lines parallel to image rows, the script '**findflat5'** should instead be used, to the user it will appear almost identical to '**findflat4'**. If this is used make sure that the images are pre-binned in the correct sense i.e. horizontally rather than vertically compressed. The correct syntax in this case would be :

<div align="center">

blkavg image1 image1 16 1 <RETURN>
blkavg image2 image2 16 1 <RETURN>

</div>

4.6. Using EXCEL to Present the Results.

The ASCII format data file produced by findflat4 or findflat5 can be imported into EXCEL using the 'open' option from the 'file' menu. Specify that the program should look for all file types rather than just EXCEL format spreadsheets and click on the name of the data file. Next specify that the data is delimited by commas. The data will then appear as a spreadsheet of the following format :

	149	405	661	917	1173	1429	1685	1941	2197	2453	2709	2965	3221	3477	3733	3989
FLATNESS DATA FROM IMAGES near1b AND away1b TAKEN WITH eev42																
Royal Greenwich Observatory. 5 Jun 1997																
Height data is in microns. A +ve bow is CONCAVE.																
Data produced by iraf script 'findflat5' from horizontal scan lines																
First column gives CCD column number of height data																
First row gives CCD row number of height data																
	149	405	661	917	1173	1429	1685	1941	2197	2453	2709	2965	3221	3477	3733	3989
1920	-306	-271.8	-259.2	-241.4	-230.9	-216.1	-198.6	-183.4	-165.6	-149.5	-130.4	-114.8	-101.5	-83.8	-78.1	-68.8
1664	-303.2	-270.7	-257.3	-239.3	-228.5	-214.3	-197.5	-182.1	-164.7	-148.7	-129.4	-113.8	-100.4	-83.3	-77.3	-67.8
1408	-300.3	-269.2	-255.2	-236.8	-226.2	-212.6	-195.8	-181	-163.3	-147.6	-128.7	-113	-99.3	-82.1	-76.4	-66.8
1152	-298.7	-268	-253.1	-234.8	-224.6	-211.6	-194.4	-180.1	-162.8	-147	-127.7	-112.5	-98.8	-81.2	-75.5	-66.3
896	-296.8	-266.2	-251.4	-234	-223.6	-210.7	-193.7	-179.1	-161.5	-146.7	-127.2	-112.5	-98.4	-80.9	-74.8	-66.2
640	-295.2	-264.9	-250.3	-233.7	-223.2	-210.3	-193.4	-178.7	-161.3	-147	-127.3	-112.5	-98.2	-81.3	-75.2	-66.6
384	-294	-264.2	-249.2	-233.4	-222.8	-210	-192.9	-178.5	-161.1	-146.8	-127.4	-112.6	-99	-81.6	-75.3	-66.8
128	-294.7	-265.2	-250.4	-233.4	-222.9	-211	-193.4	-179	-162	-147.8	-128.2	-112.5	-99.8	-81.9	-75.5	-66.6
FOLLOWING DATA HAS TILT REMOVED																
zero-point (absolute height of readout corner)= -294.7 microns																
x-slope (difference in height along serial register) = -11.2 microns																
y-slope (difference in height along an image column) = 228.0 microns																
	149	405	661	917	1173	1429	1685	1941	2197	2453	2709	2965	3221	3477	3733	3989
1920	0	18.9	16.3	18.9	14.2	13.8	16.1	16.1	18.6	19.6	23.4	23.9	21.9	24.5	14.9	9.1
1664	1.1	18.4	16.6	19.4	14.9	13.9	15.6	15.7	18	18.8	22.8	23.3	21.4	23.3	14.1	8.4
1408	2.4	18.3	17	20.3	15.6	14.1	15.7	15.3	17.8	18.3	21.9	22.4	20.9	22.9	13.5	7.8
1152	2.4	17.8	17.6	20.7	15.6	13.4	15.5	14.5	16.7	17.2	21.3	21.4	19.9	22.2	12.7	6.8
896	2.7	18.1	17.7	19.8	15.1	12.7	14.6	13.9	16.3	16	20.2	19.7	18.6	20.9	11.8	5.2
640	2.7	17.8	17.1	18.5	13.9	11.5	13.3	12.8	15	14	18.5	18.1	17.2	18.9	9.8	3.2
384	2.2	16.8	16.7	17.2	12.6	10.3	12.1	11.3	13.3	12.6	16.8	16.4	14.8	17	8.1	1.4
128	0	14.2	13.8	15.6	10.9	7.6	10.1	9.2	11	10	14.4	14.9	12.4	15.1	6.3	0

The first row and the first column of each data table give the co-ordinates of each height datum within the table. If we look at an entry close to the centre of the top table, for example, we see that the height of the chip at pixel location {column=1152, row=1941} is -180.1 microns relative to the focal plane of the scan head.

The lower data table is de-tilted and has the height of three of its corners set to zero. This de-tilting is done (within the IRAF script) by fitting a plane through three corners of the upper data table and then calculating the height difference between this plane and the rest of the data table entries.

A 3D surface plot of the data can be produced using the 'chart wizard' feature. Once the resultant graph is selected by double clicking , it can then be rotated and elevated to produce the clearest display. This is done using the '3D View' option from within the 'format' menu.

5. Acknowledgements.

I am grateful to Dave King who designed the optics, Jim Lewis who advised on the use of IRAF and the RGO Mechanical Workshop who manufactured the scanner.

APPENDICES

A.   CCD Controller Interface Box.

The standard cryostat to CCD controller heater cable has an RS404-468 cable mounting plug at the controller end and an RS404-474 cable mounting socket at the cryostat end. The cable used to connect the controller to the interface box shown below modified this design by connecting a co-ax to the controller end of the cable. The sheath was connected to pin D, the core to pin C. The other end of this co-ax was terminated in a co-ax plug that mated with J3 shown in the diagram below.
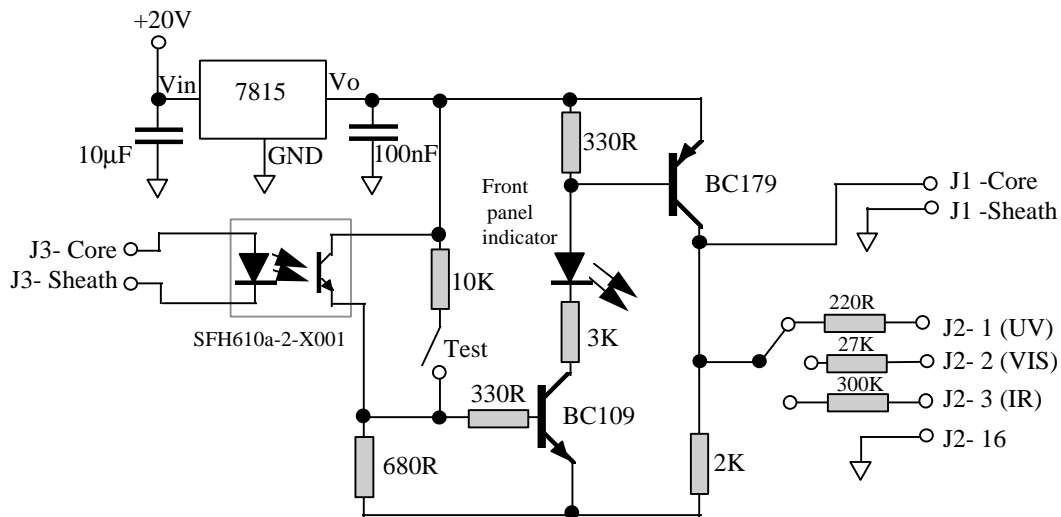

- J1 is connected to the scan head

- J3 receives the 'preflash' signal from the CCD Controller using a modified temperature servo cable

- J2 is used to power another piece of equipment : the flat field projector.

- A front panel indicator LED shows when the circuit is active. A front panel single pole non-latching switch allows the circuit to be activated manually.

B. Optical specification of scan head

The scan head measures 98mm in diameter and is 130mm high. Care is taken in the positioning of this scan head with relation to the translation stages to ensure that the centre of gravity of the top stage- scan head combination does not at any time lie outside the base of the lower stage, a situation that would compromise translation accuracy. The scan head consists of a symmetrical optical system containing two 40mm diameter achromatic doublets (Melles Griot 01 LAO 138) mounted in combination with two meniscus lenses (Melles Griot 01 LAM 138). One doublet-meniscus pair acts as a collimator for an LED-illuminated 50μm pin-hole , the second focuses this collimated beam to a 75μm FWHM spot image on the CCD. This second lens pair is mounted in a threaded barrel that allows it to be focused vertically by +/-2mm. The air gap between each doublet and its respective meniscus is 1mm. Between the two lens groups is fixed an aperture plate with a 14mm diameter on-axis hole surrounded by a ring of 4 equi-spaced 6mm diameter holes near the lens periphery on a 31mm P.C.D. A slot allows a brass aperture selector plate to be inserted to uncover one or more of these holes. The LED illuminator has a centre wavelength of 560nm.
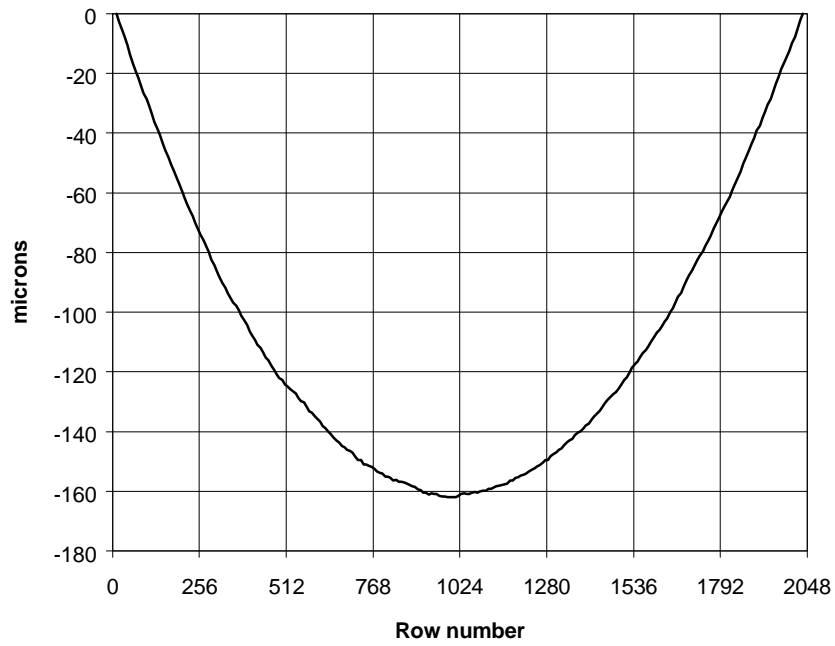
C. Some CCD surface scan results.
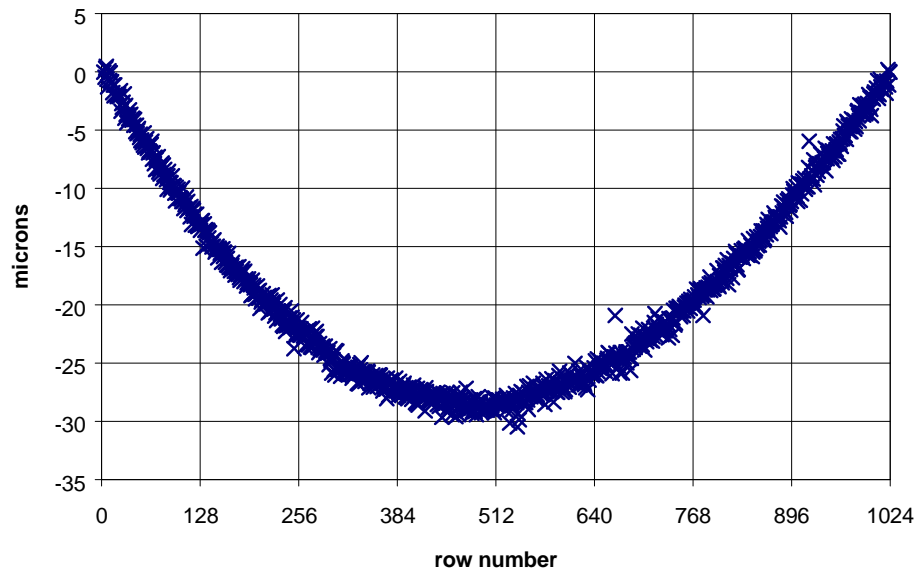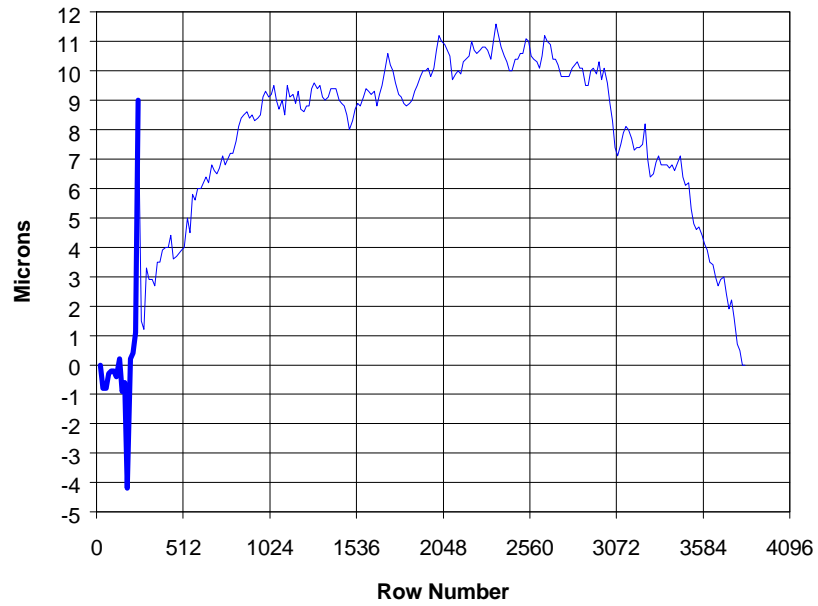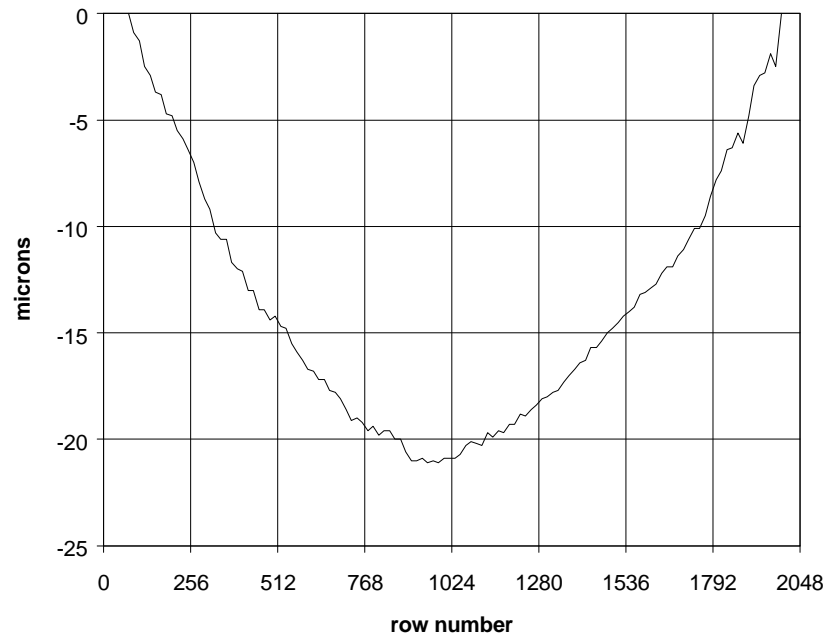

TEK1024 CCD
(convex bow)
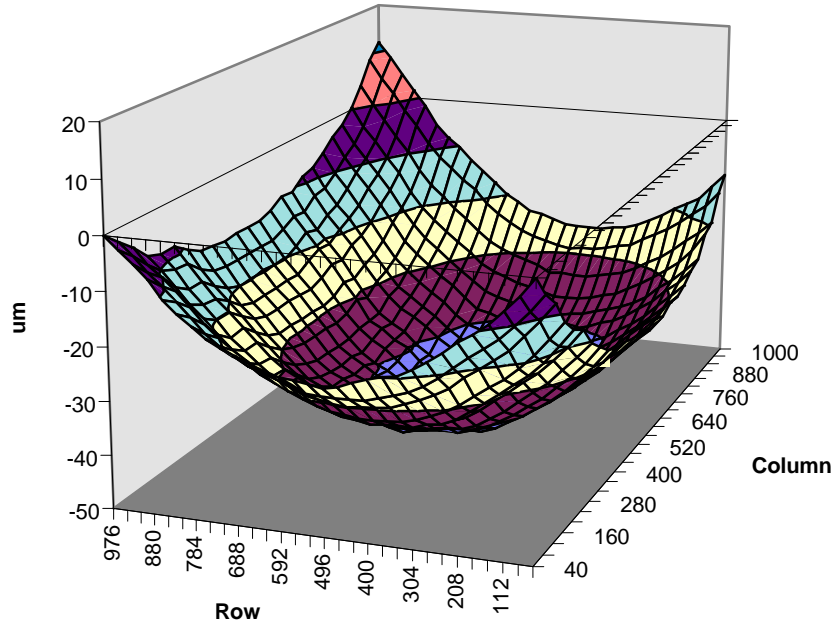
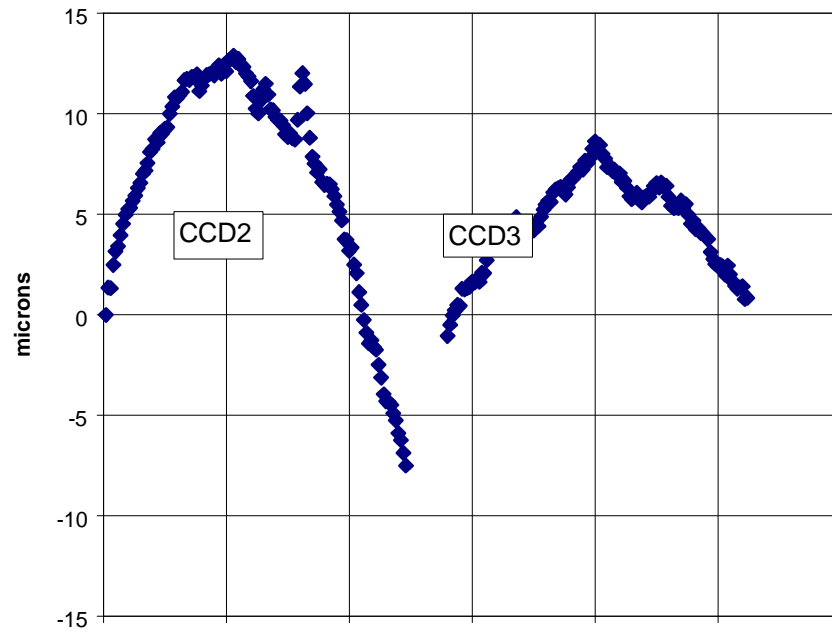**2D Scan of a TEK1024 CCD**

(convex bow)

SITe2048 CCD

EEV 42-80

(concave bow)

Lesser Thinned Loral 3EB

(Convex bow)

Scan across two adjacent CCDs in the INT Prime Focus Mosaic

D. Listing of the IRAF script 'findflat4'

```
procedure findflat4(p0,p1,p2,p3,p4,p7,p13,p8,p9,p10,p11,p12)

####### FOR IMAGES WITH VERTICAL SCAN LINES #########

# Takes two images with multiple or single vertical scan line(s).
# Scan lines must be evenly spaced and bias level should be <500
# Places height data in a 2D array output file, EXCEL readable.
# Can accept binned images with up to 32 rows and 32 vertical scan lines.
# Scan lines can be anywhere but must be parallel to columns and equally spaced.
# SMT 29 July 96

char p0 {prompt="Do you need advice on how to use this program?"}
string p1 {prompt="Image 1 File name (with aperture closest to amplifier)?"}
string p2 {prompt="Image 2 File name (with aperture farthest from amplifier)?"}
int p3 {prompt="Number of first row of image to be analysed?"}
int p4 {prompt="Number of last row to be analysed?"}
string  p7 {prompt="Name of output data file?"}
string  p13 {prompt="CCD Identifier?"}
real p8 {prompt="Pixel size (microns) ?"}
int p9 {prompt="Vertical binning factor?"}
int p10 {prompt="Number of vertical scan lines?"}
int p11 {prompt="Column number of first scan line"}
int p12 {prompt="Spacing between scan lines?"}

struct *clist1
struct *clist2

begin
char reply
int count,min,max,first_row,last_row,bin,lines,firstcol,spacing,c,lp1,x,y,idx,i,xoffset
string st1,st2,st3,st4,st5,dum1,dum2,dum3,dum4,dum5,ipf1,ipf2
string tfile0,tfile1,tfile2,tfile3,tfile4,tfile5,image1,image2,opf,dat3,ccd
string day,mnth,dte,tme,yr
real val,val1,val2,fwhm1,fwhm2,av,zero,xslope,yslope,pixsize
real dat1,dat2,dat4,dat5,diff,fnumneg,fnumpos
string str0
real darray1[9000]
real darray2[34,1030]
real darray3[34,1030]

date | scan(day,mnth,dte,tme,dum1,yr)
reply=p0
if ((reply=="y") || (reply=="Y"))
                {
                print(" ")
                print("GENERAL INFO")
                print("The two images should contain between 1 and 25 vertical scan lines. These lines should be exposed")
                print("to about 10000 ADU and should be parallel to the CCD columns. It is essential that these scan ")
                print("lines be equally spaced. The images should be binned down vertically to reduce processing time.")
                print("If a windowed image is used, the window should include the first row of the CCD or the data may be")
                print("incorrectly labelled. The centroiding algorithm uses data 35 pixels to the left and right of each")
                print("scan line, if a column defect occurs within this range the algorithm will not work properly.")
                print(" ")
                print("The data is output in two blocks in a format readable to EXCEL, if you import data delimited")
                print("by commas. The first block contains the raw height data in microns. The first row of data shows")
                print("the CCD column number, the first column shows the CCD row number. The second block of data shows")
                print("heights relative to the output corner of the CCD and  has the tilt of the cryostat removed. The x ")
                print("and y tilt values are also printed." )
                print("The middle scan line is graphed by the program as a check that the input images have given good data.")
                print(" ")
                print("PRECAUTIONS")
                print("Keep tilt small, if it exceeds a few hundred microns you should adjust the cryostat capstans.")
                print("For a 30mm square CCD in a standard cryostat, one turn of one capstan will change tilt by")
                print("about 110um and zero-point by about 220um.")
                print(" ")
                print("If the zero-point is >+-250 microns you should refocus using either the focussing ring on the")
                print("scan head or in more severe cases the capstans. One turn of the scan head or capstans =635um.")
                print(" ")
                print("N.B. If only a single capstan is adjusted the focus will be adjusted by 220um per turn.")
                print("Remember, increasing the scan head-cryostat distance will make the height readings more positive.")
                print(" ")
                print(" SMT Aug 96")
```

```
                print(" ")
                print(" ")
                }
image1=p1
image2=p2
first_row=p3
last_row=p4
if (last_row-first_row > 1024)
                {
                print("Too many rows, total number of rows truncated to 1024.")
                print("NB, only the first 1024 rows of the image will be analysed, suggest further binning")
                last_row=first_row+1023
                }
opf=p7
ccd=p13
pixsize=p8
bin=p9
lines=p10
firstcol=p11
if (lines>1) spacing=p12
                else spacing=10

fnumpos=1.97 # f number of scanning optics above focus
fnumneg=1.99 # f number of scanning optics below focus

#####################################################################################

print(" ")
print("Creating cursor command file......")

tfile2=mktemp("tmp$smt")
tfile3=mktemp("tmp$smt")
tfile4=mktemp("tmp$smt")

# Dummy operation, needed because of inconsistencies in output data format of 'splot'.######

tfile0=mktemp("tmp$smt")
print("10 0 1 k", >>tfile0)
print("50 0 1 k", >>tfile0)
print("0 0 1 q", >>tfile0)
splot(image1,1,cursor=tfile0, >G "dev$null", >> tfile2)
delete("splot.log",verify-)
delete(tfile0,verify-)


# Create cursor command file, for use by 'splot'.#######################################


tfile1=mktemp("tmp$smt")
for(lp1=0;lp1<lines;lp+=1)
                {
                min=firstcol+lp1*spacing-50
                max=firstcol+lp1*spacing+50
                for(count=first_row;count<=last_row;count+=1)
                        {
                        st1=str(min) + " 0 1 k"
                        st2=str(max) + " 0 1 k"
                        st3=" )"
                        print(st1,>>tfile1)
                        print(st2,>>tfile1)
                        print(st3,>>tfile1)
                        }
                # Now go back to bottom
                st3="("
                for(count=first_row;count<=last_row;count+=1)
                        print(st3,>>tfile1)
                }
print("0 0 1 q",>>tfile1)



# Find centroid of two images and write verbose data to temp files########################

st1="Analysing "+image1+"......"
print(st1)
splot(image1, first_row, cursor=tfile1, >G "dev$null", >> tfile2)
```

26

```
copy("splot.log",tfile3)
delete("splot.log",verify-)

st1="Analysing "+image2+"......"
print(st1)
splot(image2, first_row, cursor=tfile1, >G "dev$null", >> tfile2)
copy("splot.log",tfile4)
delete("splot.log",verify-)

delete(tfile2,verify-)
delete(tfile1,verify-)
print("Analysis complete")



# Extract only relevant data from temp files and write to 'darray1'. ######################

clist1=tfile3
clist2=tfile4

count=first_row*bin
print("Creating output data file......")
print(" ")
max=lines*(last_row-first_row+1)
for(c=1;c<=max;c+=1)
                {
                # skim off two empty lines that sperate each chunk of centroid data
                if (fscan(clist1,dum1) !=EOF)
                if (fscan(clist1,dum1) !=EOF)
                if (fscan(clist1,val1,dat1,dat2,dat3,dat4,dat5,fwhm1) !=EOF)

                # skim off two empty lines that sperate each chunk of centroid data
                if (fscan(clist2,dum1) != EOF)
                if (fscan(clist2,dum1) != EOF)
                if (fscan(clist2,val2,dat1,dat2,dat3,dat4,dat5,fwhm1) != EOF)

                diff=(val1-val2)*pixsize
                if (diff<=0) diff=diff*fnumneg
                else
                if (diff>0) diff=diff*fnumpos
                else
                print("error")
                darray1[c]=diff
                }


# Now sort the data into the two dimensional 'darray2'###################################

for(x=1;x<=lines;x+=1)
                for(y=1;y<=(last_row-first_row+1);y+=1)
                        {
                        idx=1+(x-1)*(last_row-first_row+1)+(last_row-first_row+1-y)
                        val1=darray1[idx]
                        if (abs(val1)<0.1) val1=0
                        darray2[x+1,y+1]=val1
                        }

# Now calculate the tilt of the CCD. Place height data with tilt removed into 'darray3'.####

zero=darray2[2,2+last_row-first_row]  # Height of chip at corner closest to output FET
xslope=darray2[1+lines,2+last_row-first_row]-zero
yslope=darray2[2,2]-zero
i=stridx(".",str(zero))
st1="zero-point= "+substr(str(zero),1,i+1)+ " microns. "
st2=" "
if (zero>10)
                st2="(Cryostat too high)"
                else
                if (zero<-10) st2="(Cryostat too low)"
st1=st1+st2
print(st1)
i=stridx(".",str(xslope))
st1="x-slope= "+substr(str(xslope),1,i+1)+ " microns. "
st2=" "
if (xslope>10)
                st2="(Tilts up along serial register, away from output FET)"
                else
```

```
            if (xslope<-10) st2="(Tilts down along serial register, away from output FET)"
st1=st1+st2
if (lines>1) print(st1)
i=stridx(".",str(yslope))
st1="y-slope= "+substr(str(yslope),1,i+1)+ " microns. "
st2=" "
if (yslope>10)
            st2="(Tilts up perpendicular to serial register)"
            else
            if (yslope<-10) st2="(Tilts down perpendicular to serial register)"
st1=st1+st2
print(st1)

# Remove offset     ####################################################

for(y=1;y<=(last_row-first_row+2);y+=1)
            {
            for(x=1;x<=(lines+1);x+=1)
                    darray3[x+1,y+1]=darray2[x+1,y+1]-zero
            }

# Remove slope along serial register, if more than 1 scan line #####################

if (lines>1)
            {
            for(y=1;y<=(last_row-first_row+1);y+=1)
                    {
                    for(x=1;x<=(lines);x+=1)
                            darray3[x+1,y+1]=darray3[x+1,y+1]-xslope*(x-1)/(lines-1)
                    }
            }

# Remove slope perpendicular to serial register  ################################

for(x=1;x<=lines;x+=1)
            {
            for(y=1;y<=(last_row-first_row+1);y+=1)
                    darray3[x+1,y+1]=darray3[x+1,y+1]-yslope*((last_row-first_row)-y+1)/(last_row-first_row)
            }

# Set tiny numbers ( such as 1.2e-12 ) to zero #################################

for(x=1;x<=lines;x+=1)
            {
            for(y=1;y<=(last_row-first_row+1);y+=1)
                    if(abs(darray3[x+1,y+1])<0.1) darray3[x+1,y+1]=0
            }

# Now print column and row information into both output arrays. ####################

darray2[1,1]=0
darray3[1,1]=0

# First the row headings
for(y=1;y<=(last_row-first_row+1);y+=1)
            {
            val=(last_row-y+1)*bin-bin/2
            darray2[1,y+1]=val
            darray3[1,y+1]=val
            }

# Then the column headings    ####################################################
for(x=1;x<=lines;x+=1)
            {
            val=firstcol+(x-1)*spacing
            darray2[x+1,1]=val
            darray3[x+1,1]=val
            }


# Dump contents of both arrays into EXCEL format file. #########################

st1="FLATNESS DATA FROM IMAGES "+image1+" AND "+image2+" TAKEN WITH "+ccd
print(st1,>>opf)
st1="Royal Greenwich Observatory.  "+dte+ " "+mnth+" "+yr
print(st1,>>opf)
```

```
st1="Height data is in microns. A +ve bow is CONCAVE."
print(st1,>>opf)
print("Data produced by iraf script 'findflat' from vertical scan lines",>>opf)
print("First column gives CCD row number of height data'",>>opf)
print("First row gives CCD column number of height data'",>>opf)


print(" ",>>opf)

# Print out raw data from darray2 #############################################

for(y=1;y<=(last_row-first_row+2);y+=1)
            {
            st1=""
            for(x=1;x<=(lines+1);x+=1)
                        {
                        st2=str(darray2[x,y])
                        i=stridx(".",st2)
                        if ((y==1) || (x==1)) i=i-2
                        st3=substr(st2,1,i+1)
                        st1=st1+","+st3
                        if((x==1) && (y==1)) st1=","
                        }
            print(st1,>>opf)
            }

print(" ",>>opf)
st1="FOLLOWING DATA HAS TILT REMOVED"
print(st1,>>opf)
i=stridx(".",str(zero))
st1="zero-point (absolute height of readout corner)= "+substr(str(zero),1,i+1)+ " microns"
print(st1,>>opf)
i=stridx(".",str(xslope))
st1="x-slope (difference in height along serial register) = "+substr(str(xslope),1,i+1)+ " microns"
if (lines>1) print(st1,>>opf)
i=stridx(".",str(yslope))
st1="y-slope (difference in height along an image column) = "+substr(str(yslope),1,i+1)+ " microns"
print(st1,>>opf)
print(" ",>>opf)

# print out 'de-tilted' data from darray3 #############################################

for(y=1;y<=(last_row-first_row+2);y+=1)
            {
            st1=""
            for(x=1;x<=(lines+1);x+=1)
                        {
                        st2=str(darray3[x,y])
                        i=stridx(".",st2)
                        if ((y==1) || (x==1)) i=i-2
                        st3=substr(st2,1,i+1)
                        st1=st1+","+st3
                        if ((x==1) && (y==1)) st1=","
                        }
            print(st1,>>opf)
            }

# Graph a single column of the de-tilted data   #######################################

if (lines>1)
            {
            x=lines/2
            xoffset=firstcol+x*spacing
            }
else
            {
            x=1
            xoffset=firstcol
            }
tfile5=mktemp("tmp$smt")
for(y=2;y<=(last_row-first_row+2);y+=1)
            {
            val1=darray3[1,y]
            val2=darray3[x+1,y]
            print(val1,val2,>>tfile5)
            }
```

```
dum1=" CCD Row Number "
dum2=" microns "
dum3=ccd+". Chip profile down column "+str(xoffset)+". RGO"
fields(tfile5,fields="1,2") | graph(po+,marker="box",title=dum3,xlabel=dum1,ylabel=dum2)
delete(tfile3,verify-)
delete(tfile4,verify-)
delete(tfile5,verify-)
print("")
print("Finished")
print("")
end
```