FORTH
USER GUIDE
To programs for interactive image processing
and microdensitometry on the RGO PDS.

JONATHAN MCDOWELL.
KEITH TAYLOR.

RGO 26/07/78
INTRODUCTION

[0.1] Conventions:

i) User typed FORTH instructions in capitals.

ii) Terminal response in italics

iii) Comments and descriptions in lower case.

iv) In user typed FORTH instructions, lower case symbols such as: n, m, x, y
stand for user typed numbers and are defined in accompanying comments.
Numbers are to be assumed to be in 16-bit format (integers, unpunctuated)
unless stated otherwise.

v) PDS coordinate directions defined by x and y.

vi) Size of 2-D data arrays on disk or tape defined by p (for Rows) and k (for columns).

vii) Pixel identification within those arrays given by i, specifying row number and j,
specifying pixel in row (column number).

viii) Some parameters are specified by 2 components. In this case they will be identified
using vector notation in the comments. For example, WN is a parameter representing
the dimensions of a 2-D array or scan. Its components are 2, 16-bit numbers
referenced by WN (k - direction) WN 1+ (p - direction).

[0.2] Instructions:

FORTH instructions are single words, delimited by spaces. The words will be
executed on typing a carriage return. (represented here by (cr)). On completion
of execution, if execution has not caused any errors, output will be OK.
Several instructions may be typed on line, separated by spaces. They will be
executed in turn. FORTH uses a push-down stack for number operations. If an
instruction is encountered which is not recognised, the word is echoed back followed
by a question mark, and execution is interrupted.

[0.3] Number entry:

FORTH uses single-precision (16-bit) and double precision (32-bit) signed integers.
16-bit integers are in the range -32768 ≤ n ≤ 32767. These are used for most purposes
but sometimes, 32-bit numbers are needed, (for example, PDS co-ordinates). 16-bit
integers are entered with no punctuation except a leading minus sign where applicable;
that is the same as FORTRAN integers (but there must be no leading plus sign). 32-bit
integers are entered by including punctuation either , - or /. Any number which
includes punctuation (except for a single leading minus sign which means a negative
number for both 16- and 32-bit numbers) will be treated as a 32-bit number, but it
will be treated as an integer, ie position of the punctuation will be ignored, hence
the numbers 12.34 1.234 1234 are all taken to be the same number, a double-precision
1234. To avoid confusion, it is probably simplest just to terminate with a point
(eg 1234.)
Basic operations:
Arithmetic is reverse polish; integer only.
Typing a number puts it on the stack.
Typing \texttt{.} prints out the top of the stack.
Typing \texttt{(number) (address) \_ stores (number) in (address).
Constants: To put value of constant, R on stack just type;
\texttt{R}
n \_ \texttt{R!} stores n in R
Variables: Typing the name of a variable, DX puts address of DX on stack.

To put value of DX on stack type;
\texttt{DX @}
n DX \_ stores n in DX
The command \texttt{?} is equivalent to \texttt{@}. hence it prints out value in address
currently on stack; eg, DX \texttt{?} prints current value of DX.
To set number base, type DECIMAL, OCTAL or HEX.
For more detailed explanation of subjects in [0.2] to [0.4], see FORTH

Note on FORTH words:
The FORTH compiler only stores the first three letters of each word and its length
thus, \texttt{FILE} is treated the same way as \texttt{FILL}; FORTH cannot tell the difference.
\texttt{FILE1, FILE2, FILE3, FILA!} are treated the same, but \texttt{FILEAB} is not (different length).
This is useful when a long word is spelt incorrectly but can lead to confusion.
Also note: CENTRE = CENTER, DISK = DISC etc. This applies to 'words', ie FORTH
commands, but not to alphanumeric strings. These strings are usually used as identifiers
or file names. They are only valid if used after a command requiring such a string;
such commands include \texttt{FIND, WRITE, INTER, ENTER}. In this documentation, notation
of the form (name) or (filename) will refer to one of these strings (eg; \texttt{FIND (filename)
might be FIND GALAXY}). These strings should not have any spaces in them, except where
mentioned in the documentation.

Interrupt:
"Control C" acts as an interrupt. This may not work when tape is being used.

PDS control:
For user to move PDS stage, the SELECT (blue and white) button must be on 'manual',
for the computer to move it, it must be on 'auto' (eg when doing a scan). Remember
to turn control over when using a command which requires stage movement, as the doc-
umentation does not mention this. Control can be turned over before or after the
relevant command has been sent.
The same basic instruction, PASS, is used to transfer scan data from PDS to
tape, or from disk to terminal display, etc, ie from one representation to
another. The format of the instruction is (unit1) SOURCE (unit2) DESTINATION
PASS to transfer data from (unit1) to (unit2); the format (location, array
size, scan sample interval, etc.) is determined by a set of parameters; each
unit has its own set. The whole instruction may not always be typed explicitly;
e.g. in plotting routines the word PLOT may be defined as SOURCE SCREEN DESTINATION
PASS, and so DISK PLOT will transfer the data from the disk to the screen in
graphic form.

Each unit name is called an 'image'. A list of 'images' follows.

Images

<table>
<thead>
<tr>
<th>Images</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLATE</td>
<td>The PDS stage; PLATE SOURCE (unit2) DESTINATION PASS does a scan</td>
</tr>
<tr>
<td>DISK</td>
<td>A file on disk. The word DISK is used implicitly in the instruction</td>
</tr>
<tr>
<td></td>
<td>FIND (filename) which also selects a specific file.</td>
</tr>
<tr>
<td>TAPE</td>
<td>A file on tape</td>
</tr>
<tr>
<td>TERMINAL</td>
<td>The terminal; for digital output</td>
</tr>
<tr>
<td>SCREEN</td>
<td>The Tektronix terminal; for graphic output</td>
</tr>
</tbody>
</table>

Also:

INPUT )
OUTPUT )
OPERA

Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
<td>Disk block number for DISK image; not used for other images.</td>
</tr>
<tr>
<td></td>
<td>Value is 500 ( \leq n \leq 2435 ).</td>
</tr>
<tr>
<td>CN</td>
<td>Size of scan in pixels; CN contains number of pixels in a row</td>
</tr>
<tr>
<td>CN1+</td>
<td>CN1 contains number of rows in the scan</td>
</tr>
<tr>
<td>DX</td>
<td>Sample interval in microns:</td>
</tr>
<tr>
<td></td>
<td>DX contains sample interval in X</td>
</tr>
<tr>
<td>DX1+</td>
<td>DX1 contains sample interval in Y</td>
</tr>
<tr>
<td>C0</td>
<td>Origin of scan in PDS co-ordinates. This is the first pixel in</td>
</tr>
<tr>
<td></td>
<td>the array, designated pixel 0, row 0, (for an nxn array pixels</td>
</tr>
<tr>
<td></td>
<td>and rows are numbered 0 through n-1.) C0 contains 2 double-precision</td>
</tr>
<tr>
<td></td>
<td>numbers: C0 contains the X co-ordinate, C0 2+ contains the Y co-ordinate.</td>
</tr>
<tr>
<td>W0</td>
<td>A 'window' or subset of the array (on disk, tape, etc.) may be defined.</td>
</tr>
<tr>
<td></td>
<td>The origin (first pixel of first row) in pixels relative to the origin</td>
</tr>
<tr>
<td></td>
<td>of complete array is contained in W0. W0 contains origin in pixels,</td>
</tr>
<tr>
<td></td>
<td>W0 1+ contains origin row number. i.e &quot;i&quot;, &quot;j&quot; of origin respectively,</td>
</tr>
<tr>
<td></td>
<td>see [0.1]. W0 and W0 1+ are always positive.</td>
</tr>
</tbody>
</table>
WN Size of window in pixels; WN contains number of pixels/row of window, WN 1+ contains number of rows in window.

KIND Specified direction of scan, i.e., scan in X or scan in Y.

Current Image

The parameters listed above are not ordinary variables because each of the 8 images mentioned above has its own set of parameters. For example, typing DX will reference the DX of only one of these images, called the "current image". An image is made the current image by typing its name (or using a word which itself used an image name). For instance, TAPE DX ? will print out the current DX value for TAPE, while PLATE DX ? will print out that for PLATE, which may be different. Words which change the current image are:

FIND makes DISK the current image
TWIDE makes TAPE the current image
PASS makes OUTPUT the current image
PLOT makes OUTPUT the current image after setting OUTPUT to be equal to SCREEN
SOURCE Copies parameters form current image to INPUT and leaves the latter as current image.
DESTINATION copies parameters from current image to OUTPUT and leaves the latter as current image.
SECOND copies parameters from current image to OPERAND and leaves the latter as current image.
USING FORTH

[1]
Bootstrap, Loading

[1.1]
Follow 'starting up' instructions in black PDS manual up to point of pressing HALT switch on console; use FORTH SYSTEM DISK 1

[1.2]
Disk Bootstrap

Set up correct address on data switches, as in PDS manual, and as indicated on console.
Press console switches: HALT (ie depress HALT/ENABLE)
LOAD ADRS
ENABLE (ie raise HALT/ENABLE)
START

If
(Otherwise
HELLO? Go to [1.4]
continue on to [1.3])

[1.3]
Bootstrap Fails

If
(Otherwise
Hit "Return" key
OK Go to [1.3.1]
Try reloading the disk; see 'closing down'
and then 'starting up' again in black PDS manual,

[1.3.1]
Block Buffers Corrupted

33350 4010 ERASE (should come back OK)
Return to [1.2]

[1.4]
Loading

JCM LOAD 20/07/78 14:12 OK

If you do not wish to correct date and time (this is completely optional) go to [1.5]
Otherwise, eg
5/09/78 NOW OK
18:07 UT OK

Response is the system date and time, these may be corrected if desired.
Sets date to be 1978 Sep 5
Sets time to be 18:07; GMT and BST are alternatives to UT and are identical, computer does not know about time zones.
If required, load magnetic tape as described in black PDS manual when signing log book, put 'FORTH' in comments column. To close down see PDS manual. In the following documentation, the response OK will be assumed after all commands.

**Options**

**List of Options**

Select an option; loading is described under the paragraph number given. Subsequently loading another option removes the programs for the first option.

<table>
<thead>
<tr>
<th>Option name</th>
<th>Brief description</th>
<th>Go to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigscan</td>
<td>Scan a plate; store data on tape, or on disk</td>
<td>[3]</td>
</tr>
<tr>
<td></td>
<td>Maximum row size is 30000 pixels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Additional option-result may be averaged down by factor of $2^n$ in x and y, while scanning.</td>
<td></td>
</tr>
<tr>
<td>Taylor</td>
<td>Scan a plate, store data on disk</td>
<td>[4]</td>
</tr>
<tr>
<td></td>
<td>Maximum row size is 512 pixels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Align a plate, by creating coordinate transform between it and a reference plate.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Graphic output; range of additional sub-options are available, eg averaging down in x and y as in Bigscan, adding, subtracting and dividing two images; contour plots, moving arrays between disk and tape, defect removal, disk file graphic display</td>
<td></td>
</tr>
<tr>
<td>Smoothing</td>
<td>Averaging digitized scan stored on disk, without shrinking array (boxcar convolution)</td>
<td>[5]</td>
</tr>
<tr>
<td>3D</td>
<td>3D plot of a scan stored on disk or tape (several versions)</td>
<td>[6]</td>
</tr>
<tr>
<td>Photometry</td>
<td>Fitting 2D Gaussians to digitized scan (on disk or tape)</td>
<td>[7]</td>
</tr>
</tbody>
</table>

**Additions**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Go to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very long scans</td>
<td>More than 30000 pixels/row</td>
<td>[8.1]</td>
</tr>
<tr>
<td>Multiple scans</td>
<td>Reflecting arrays in x and y axes</td>
<td>[8.2]</td>
</tr>
<tr>
<td>Array flipping</td>
<td></td>
<td>[8.3]</td>
</tr>
<tr>
<td>Tape file record-by-record graphic display</td>
<td></td>
<td>[8.4]</td>
</tr>
<tr>
<td>Coarse file</td>
<td></td>
<td>[8.5]</td>
</tr>
<tr>
<td>Density histogram</td>
<td></td>
<td>[8.6]</td>
</tr>
</tbody>
</table>

*Except for options covered in [8.1], [8.2] and [8.5] which are to be loaded "on top of" BIGSCAN and [8.3], to be loaded with either BIGSCAN or TAYLOR. This is made clear in the relevant documentation.*
If you wish to use the capability to average down in x and y, as opposed to simple scan onto tape, go to [3.3]

Otherwise:

Loading BIGSCAN

\[ \kappa \text{ BIGSCAN LOAD} \]

\[ \kappa \text{ is the maximum row size in pixels that you will be using. It must be} \leq 3000. \text{ For larger scan lines see [8.1]} \]

Go to [3.4]

Loading BIGSCAN with Averaging Capability

When averaging, a scan of \( p \) rows of \( \kappa \) pixels each will be stored in an array on tape of \( p/a \) rows with \( \kappa/a \) pixels each, where \( a (\neq 0) \) is the value of a constant called \( AV \). However, working storage must be reserved for the full \( \kappa \) pixels.

\[ \kappa \text{ BIGSCAN LOAD} \]

\[ 435 \text{ LOAD} \]

Continue

Scanning

Set up over desired scan

If you wish to zero PDS coordinates at this point,

\[ \text{ZILCH} \]

If you wish to scan in x no special command is needed, this is default. However if you scan in y and later wish to scan in x, you must then use

\[ \text{HORIZONTAL} \]

If you wish to scan in y \[ \text{VERTICAL} \]
HELLO ? JCM LOAD 26/07/78 33:52 OK
19:48 UT OK
1824 BIGSCAN LOAD OK
ZILCH OK
3 MM+S OK
0 " MOD 1 OK
PLATE 64 64 WIDE 10 10 RXDY! OK
CENTER OK
REWOI OK
NTP OK
WRITE JCM TEST SCAN 1978 JUL 26 OK

Example of a simple scan using BIGSCAN

VERTICAL OK
5 VEL 11 ODD 1 OK
PLATE 32 29 WIDE 12 5 RXDY! OK
LL CORNER OK
WRITE VERTICAL SCAN, ODD=1 OK

Example of a further scan using non-standard parameters
To set PDS velocity

Either
or
n MM/S
m VEL !

Sets velocity in mm s⁻¹
Sets velocity in PDS units
(m = 255/60n)

To set scan mode:

Either
or
Ø ' ODD !
1 ' ODD !

Default. Scans in 1 direction only
(increasing X, or Y if VERTICAL)
Scans back-and-forth, and flips rows in
working storage so final result is same

Then set up scan parameters as follows:

PLATE
p × WIDE

øx øy DXDY !

Makes PLATE the current image (see [0.8])
Size of scan; p rows with x pixels in each
row.

Sets parameters WN(=CN)
Sets sampling interval in microns
dx is interval in x-coordinate
dy is interval in Y-coordinate
(sets parameters DX)
Ignore output column marked 'ORIGIN'

If you wish to check
the above parameters,
If you wish to define the scan by its center:

STATUS
Position over center
CENTER

Stores origin (ie coordinates of most -ve
Corner) by setting parameters CW, using
current PDS position (x,y) and WN and DX,
such that \(x^* = x - \frac{1}{2} W_x 6x; y^* = y - \frac{1}{2} W_y 6y \)
where \((W_x, W_y)\) are \((k,p)\) or \((p,k)\) depending
on scan direction.

Go to [3.5]

Otherwise define the appropriate corner

Position over corner

Either
or
UR
UL

Upper right corner; default
Upper left corner
or
or

Then

CORNER

[3.5] Scanning

For scanning to disk, see the description under TAYLOR LOAD, section [4.7.2]

If tape is a new tape

REWIND
NTP

Otherwise the tape may be positioned anywhere before the final double filemark.

Then: If averaging option is loaded and you wish to use it on this scan, go to [3.6]

Otherwise WRITE (identifier)

 Rewinds tape, initializes variables, writes 1 record and 2 filemarks, then backspaces twice. The record is for compatibility with RT tapes which have no filemark before the first filemark.

(identifier) is a string of ≤ 40 characters (including spaces). Follow this by carriage return.

(i) Tape will run forward until a double filemark is found, and will then backspace so that the second filemark is overwritten.

(ii) The scan will be performed using the PLATE parameters set previously. The data will be written on to tape in Scansalot format.

(iii) After the scan is completed, a double filemark will be written at the end of the file, and the tape will then be backed spaced twice, so that a second WRITE will not require any repositioning of the tape.

(For an example of user commands for a scan using BIGSCAN, see p )
Go to [3.7]

Averaging to tape

To average on $\kappa \times \rho$ scan into an array of size INT ($\kappa/\alpha \times \rho/\alpha$)

```plaintext
a ' AV !
R-UP TURN-OFF
```

AVWRITE (identifier)

Stores $a (= 2^n)$ in constant AV.

Prints out warning message on other terminal

(VDU or DECWRITER) if either is switched on.

The other terminal cannot be used while

AVWRITE is in use since this uses the other terminals' program area as working storage.

As WRITE (see [3.5]) but averages.

eg If there are 18 pixels/row and AV=4, first 4 rows of 18 pixels each will be averaged down to one row of 4 pixels and then written on to tape, similarly for subsequent sets of 4 rows, until the 16th row, $\kappa, \rho \geq 17$ are ignored.

Continue

Further scans

You may return to section [3.4]; remember to type PLATE so that any parameters modified will be those of PLATE.

Reset any parameters by using MM/\alpha, WIDE, DXDY! VERTICAL, ODD. You may leave any of these unchanged. Then

specify origin using CENTER or UR, UL, LL, LR CORNER

To Read Files Created by BIGSCAN

R-DOWN

Go to [4.12] (Reading tape files)
[4.1] Loading

TAYLOR LOAD

Loads FDS scanning routines, Tektronix graphics routines, least squares fit routines. STARS file routines, and disk file index. Max no of pixels/scan row is 512.

[4.2] Sub-options

There are a number of additional facilities which may be loaded as well as the programs in TAYLOR LOAD. Only one of these options may be loaded at a time, subsequently loading another one will remove the first.

Either (i) PROCESSING LOAD

Go to [4.3]

or (ii) Either

removal load ) see [4.9] before

or

SLICING OPT LOAD ) loading

Go to [4.9]

or (iii) CONTOUR LOAD

Go to [4.12]

or (iv) SCANSALOT LOAD

[4.3] If you wish to align a plate before a scan, so that corresponding arrays may be obtained from different plates for the purpose of adding two plates of same object, or for the normalization of electronographs using a photocathode map go to [4.4]

Otherwise If you just wish to scan to disk without using the plate alignment facility, go to [4.7]
ALIGN

Description of method: 3 or more points (fiducial marks or stars) are entered into a file, called the STARS file, followed by the scan origin coordinates. A scan is done. Corresponding points are then entered in file for a second plate. Least-squares fitting is used to determine angle of rotation needed for the plates to be aligned (the transitive component of the coordinate transformation is taken care of by the software but the rotation must be made zero by physically rotating the stage.) After stage is rotated, re-enter points and iterate until angle is acceptably small. (Usually 2 or 3 iterations are enough.) Then the corresponding scan is done.

(Note: the stars file is stored on disk so the data will not be lost until they are overwritten by the user.)

Continue

[4.5]

LINK-CLEAR

Position over the first fiducial mark (or star)

If gaussians to be fitted:

1 ENTER (name)

Otherwise

Center over mark
1 $ENTER (name)

Clears out stars file.
If there is a set of data already in it (eg you are aligning different sets of plates) omit this instruction, and place the new data later in the file.

If it is a star, you may wish to fit gaussians to it to determine the center point.
The word 'mark' will be used (for this section and the next) to mean fiducial mark or star

(name) is ≤ 6 characters long and is for user identification only, not used in programs. This will so a scan 640 µ long (64 pixels) in Y (center of scan line is the current PDS position), fit gaussian to the scan line data, plot the data points and the fitted curve on the Tektronix screen. Do similar scan in X at Y value determined by first fit, fit gaussian, type out co'ords of fitted center and enter them and (name) into record 1 of STARS file. If you are keeping old data in records 1 through n of stars file, you can start at record n + 1 thus N+1 ENTER (name)

(name) is ≤ 6 characters long and is not used in programs. Enters (name) and current PDS coordinates.
TAYLOR LOAD OK
PROCESSING LOAD OK
LINK-CLEAR OK
1 ENTER SMALL

Example of ENTER
<table>
<thead>
<tr>
<th>#</th>
<th>NAME</th>
<th>X</th>
<th>Y</th>
<th>X</th>
<th>Y</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FLU</td>
<td>-24778</td>
<td>2</td>
<td>-24767</td>
<td>-11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>FRU</td>
<td>-24634</td>
<td>-24626</td>
<td>-24627</td>
<td>-24604</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>FRL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>GALAXY</td>
<td>-9478</td>
<td>-11933</td>
<td>-9547</td>
<td>-11918</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>STAR1</td>
<td>1426</td>
<td>-4209</td>
<td>1356</td>
<td>-4182</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>STAR2</td>
<td>-22551</td>
<td>-5337</td>
<td>-22626</td>
<td>-5319</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>STAR3</td>
<td>-21128</td>
<td>-10525</td>
<td>-21190</td>
<td>-10493</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>HII#1</td>
<td>-10325</td>
<td>-11999</td>
<td>-10891</td>
<td>-11983</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OK

These columns not used.

Example of STARS ALL
Then

If you wish to check contents of STARS file at any time

STARS ALL

Proceed to next mark

If gaussian centering required

2 ENTER (name)

Otherwise

2 %ENTER (name)

Then carry on, using ENTER, or %ENTER to enter consecutive records of the STARS file, until all marks have been entered. Some marks may be entered using ENTER and others using %ENTER.

After all marks have been entered, move the PDS to the position of the scan (or the first scan, if more than one) that you are going to do, and enter the position in the STARS file in the next record, using %ENTER (e.g. 4 ENTER SCAN). Enter all the scans that you wish to re-scan on the second plate in consecutive records of the STARS file using %ENTER. The point that defines a scan can be its center or its center or any of its corners.

To do the scans go to [4.7] or [4.12]
for tape and then after scanning go to [4.6].

[4.6.]

Aligning second plate

You wish to do a scan or scans which will correspond pixel for pixel to a scan done on another plate, whose fiducial or stars have been entered in the STARS file. The new plate must be already roughly aligned by eye with the first one, within a few degrees, (by aligning fiducial marks or stars it should be possible to rotate the plate to within about 5° or the other: this is quite adequate.) The program uses the approximation sin θ ≈ θ.

[4.6.1]

Position over first mark ENTERed on first plate

If gaussians to be fitted 1 FIRST

Otherwise

1 %FIRST

Initializes rotation variables. Scans in Y and X as in ENTER, displays results as in ENTER, but stores final coordinates in second set of X, Y columns in STARS file. If for first mark on first plate you used a record other than record 1 (i.e. n+1 ENTER (name)) use the same record here (i.e. n+1 FIRST).

Initializes rotation variables Enters current coordinates in second set of
NOTE: The records in the file used for the alignment must be consecutive, but if the first mark cannot be found, find another and use m FIRST where m is the appropriate record number, then type Ø R# ! (or n R# ! if first mark is record n+1) before using NEXT (see below).

Then [4.6.2] NEXT

Move the PDS to be correctly over the mark

If gaussian fit required

OK

Otherwise $OK

Then loop back to [4.6.2] until all marks have been stored.

Then n m ROTATE

X,Y columns in STARS file. The record number used should be the same as for the corresponding mark on the first plate, i.e. if (n+1) ENTER (name) or n+1 $ENTER (name) was used, n+1 $FIRST should be used here.

PDS will move to a position corresponding to next record in the STARS file, plus an offset of the coordinate origin calculated from the 2 sets of coordinates in the record for which FIRST or $FIRST was used; i.e. where the next mark ought to be if the angle of rotation were zero.

In the first iteration the error may be quite large but in later iterations the position reached after NEXT should be close enough to require only a little 'tweaking up'.

Scans, fits, displays, stores in STARS file (second set of columns). Record number was incremented by NEXT. Stores current coordinates in STARS file (second set of columns). Record number was incremented by NEXT, so will be correct automatically.

STARS file: records n through m inclusive are used to fit a transformation between the two sets of coordinates. If first record was record 1 and 3 marks were entered, 1 3 ROTATE
ROTATION
X2 Y2 ANGLE
23.4 -14.2 3.780
12.0 15.0 0.218
137

Rotate PDS stage through the appropriate angle

Output from ROTATE
The number immediately under ANGLE represents the angle of rotation in degrees.
X2, Y2 are offsets and second, third rows are residuals. The only number the user needs is the angle.
If angle is large, accuracy in rotation is not important due to the approximation in the transformation.

Iterate by returning to [4.6.1], repeat until angle is sufficiently small.

Then

NEXT

[4.6.3] Additional ALIGN commands

y x 00
n 1G0
n 2G0

Graphics mode: The data points for the gaussian fit in ENTER, FIRST, OK are plotted out as crosses. ("POINTS mode")
Alternatives are listed below, using the word sets the plotting mode until another of the words is used or the graphics are re-loaded.

POINTS
FOLLOW
HISTOGRAM

The commands AXES and BIG GRID will plot axes and a grid on the screen.
Scanning

Position PDS over point defining first scan (if ALIGN has been used, this will already have been done.) This point may be the center or any corner.

If you wish to zero the PDS coordinates at any time, this can be done by the command ZILCH or by pressing the appropriate buttons on the console displaying the coordinates.

If you wish to scan in Y

VERTICAL

To return to scanning in X later,

HORIZONTAL

Otherwise: Scanning in X is default; it is not necessary to type HORIZONTAL unless VERTICAL has been used earlier.

To set PDS velocity

Either

n MM/S
or
m VEL !

Sets velocity in mm s⁻¹
Sets velocity in PDS units (m = 255/60 n)

To set scan mode:

Either

0 ' ODD !
or
1 ' ODD !

Default. Scans in 1 direction only (increasing X, or Y if VERTICAL)
Scans back-and forth, and flips rows in working storage so final result is same.

Then set up scan parameters as follows:

PLATE
ρ = WIDE

δx δy DXDY!
If you wish to check the above parameters

STATUS

If you wish to define the scan by its center

Position over center
CENTER

\[4.7.2\]

Otherwise define the appropriate corner.

Position over corner

<table>
<thead>
<tr>
<th>Either</th>
<th>UR</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>or</td>
<td></td>
<td></td>
</tr>
<tr>
<td>or</td>
<td>LR</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td>LL</td>
<td></td>
</tr>
</tbody>
</table>

Then

CORNER

If you are scanning to tape using SCANSALOT LOAD, return to \[4.12\]

Otherwise go to \[4.7.2\]

\[4.7.2\]

Scanning to disk

SOURCE

Ignore output column marked 'ORIGIN'

Stores origin (ie coordinates of most -ve corner) by setting parameters $C_0$, using current PDS position $(x, y)$ and $\delta X$ and $\delta Y$.

\[
x_0 = x - \frac{1}{2} w_x \delta x
\]

\[
y_0 = y - \frac{1}{2} w_y \delta y
\]

where $(w_x, w_y)$ are $(\kappa, \rho)$ or $(\rho, \kappa)$ depending on scan direction.

Upper right corner: default
Upper left corner
Lower left corner
Lower left corner

These define corner of scan, as seen on PDS viewing screen, not on plate itself. They modify the signs of PLATE parameters $DX$.

Stores current PDS coordinates in PLATE parameters $C_0$. This position will be the first pixel $(i = 0, j = 0)$ of the scan

Specifies PLATE (the current image) as the source for the pass (scan operation). The parameters set up in PLATE will be copies to INPUT. SOURCE leaves INPUT as the current
<table>
<thead>
<tr>
<th>#</th>
<th>NAME</th>
<th>LOC'I N</th>
<th>ORIGIN:X</th>
<th>Y DIR.</th>
<th>CORN.</th>
<th>SIZE:X</th>
<th>Y</th>
<th>+X</th>
<th>+Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CART2</td>
<td>485</td>
<td>0</td>
<td>0</td>
<td>H</td>
<td>UR</td>
<td>128</td>
<td>128</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>516</td>
<td>640</td>
<td>640</td>
<td>H</td>
<td>UR</td>
<td>64</td>
<td>64</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>CART3</td>
<td>520</td>
<td>-1920</td>
<td>-1920</td>
<td>H</td>
<td>UR</td>
<td>64</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>527</td>
<td>0</td>
<td>0</td>
<td>V</td>
<td>LL</td>
<td>64</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>COPY</td>
<td>479</td>
<td>0</td>
<td>0</td>
<td>V</td>
<td>LL</td>
<td>64</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>477</td>
<td>-320</td>
<td>-320</td>
<td>V</td>
<td>LL</td>
<td>64</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>JCM</td>
<td>480</td>
<td>-2562</td>
<td>-2560</td>
<td>H</td>
<td>UR</td>
<td>128</td>
<td>128</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>481</td>
<td>-2</td>
<td>0</td>
<td>V</td>
<td>UR</td>
<td>512</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>LONG</td>
<td>600</td>
<td>0</td>
<td>0</td>
<td>V</td>
<td>UR</td>
<td>512</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>603</td>
<td>20</td>
<td>2560</td>
<td>V</td>
<td>UR</td>
<td>512</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>MGD01</td>
<td>457</td>
<td>-1000</td>
<td>1000</td>
<td>H</td>
<td>UL</td>
<td>32</td>
<td>32</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>458</td>
<td>-1160</td>
<td>1160</td>
<td>H</td>
<td>UL</td>
<td>32</td>
<td>32</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>N1068#1</td>
<td>436</td>
<td>-436</td>
<td>-501</td>
<td>H</td>
<td>UR</td>
<td>52</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>434</td>
<td>14</td>
<td>-1</td>
<td>V</td>
<td>UR</td>
<td>52</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>PK</td>
<td>461</td>
<td>0</td>
<td>0</td>
<td>H</td>
<td>LR</td>
<td>64</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>468</td>
<td>320</td>
<td>-320</td>
<td>H</td>
<td>LR</td>
<td>64</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>TEST</td>
<td>459</td>
<td>-1000</td>
<td>1000</td>
<td>H</td>
<td>UL</td>
<td>32</td>
<td>32</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>460</td>
<td>-1160</td>
<td>1160</td>
<td>H</td>
<td>UL</td>
<td>32</td>
<td>32</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>TWIN</td>
<td>432</td>
<td>0</td>
<td>0</td>
<td>V</td>
<td>UR</td>
<td>64</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>439</td>
<td>320</td>
<td>320</td>
<td>V</td>
<td>UR</td>
<td>64</td>
<td>64</td>
<td>10</td>
</tr>
</tbody>
</table>

OK

Example of LISTFILE

(Name) Starting block X origin Y origin
Finding block X center Y center

Which area? X pixels Y pixels 6x 6y
Vertical beam? X center Y center

Example of LISTFILE
LISTFILE (cr) (cr)

Find highest block number used

Store your image starting in a block higher than this (usually the next block)
Allocate the file by:
  n  INTER (filename)

If there are old files in the index that are no longer used,
  REMOVE (filename)

If you have done a PROCESSING LOAD (or do one at this point) and wish to use the averaging capability
for this scan, go to [4,73]

Otherwise
  FIND (filename) EQUAL DESTINATION

PASS

If you are using ALIGN and now wish to align second plate, go to [4,6].

If you wish to do further scans, go to [4,7]
For further capabilities in PROCESSING LOAD see [4.8]

[4.7.3]

Averaging

FIND (filename) a 'AV'!

AVBY DESTINATION

AVERAGE

Sects disk file
Stores a (number to be averaged by, a = 2^a) in AV
AVBY takes INPUT parameters, multiplies or divides by AV where appropriate, and puts them in DISK's parameters. DESTINATION specified DISK as the OUTPUT (copies DISK's parameters to OUTPUT)

Modified PASS - averages in i and j scans a line, puts this in working storage. Takes first AV numbers, adds each into first location of "sum buffer"; next AV numbers into second location, etc. Before being added in, numbers are divided by AV^2 2nd, 3rd... AVth scan line added in and then sum buffer stores as lst row of output array, and sum buffer zeroed. Repeated for every group of AV scan lines, so final array will have each number being the average of AV^2 numbers; eg a scan 64 64 WIDE with 4 in AV will result in an output array of size 16 by 16 with sample size multiplied by 4 in each direction (ie 16 16 DXY! Will have an output dx, dy of 40, 40).

If you are using ALIGN and now wish to align second plate, go to [4.6]
If you wish to do further scans, go to [4.7]

For further capabilities in PROCESSING LOAD see [4.8]

[4.8]

Further capabilities in PROCESSING LOAD - combination of disk images

For two disk files (file 1) and (file 2) to be arithmetically combined to form a third file (file 3) use the following instructions

FIND (file 1) SOURCE
FIND (file 3) DESTINATION

(file 1) is identified with INPUT
(file 3) is identified with OUTPUT
FIND (file 2) ( ADD SUBTRACT DIVIDE
PASS

(file 2) is identified with OPERAND by using one of the words.
PASS combines data in INPUT with data in OPERAND and places the result in (file 3)

The commands defining the OPERAND are

ADD giving (file 3) = (file 1) + (file 2)
SUBTRACT giving (file 3) = (file 1) - (file 2)
DIVIDE giving (file 3) = NUM x (file 1)/(file 3)

where NUM is a variable (default 1000) which may be set by n NUM !

Each element in the INPUT array is combined with the corresponding element in the OPERAND array row by row in working storage. The INPUT and OPERAND files are unaltered.

If you wish to overwrite (file 1) with the result then

FIND (file 1) SOURCE DESTINATION
) ADD
FIND (file 2)) SUBTRACT PASS
) DIVIDE

If, however you wish to overwrite (file 2) with the result, then

FIND (file 1) SOURCE
FIND (file 2) DESTINATION
ADD )
SUBTRACT) PASS
DIVIDE )

[4.8.2] Sum and average of an array

SBAR will calculate the average value of an array and both print it out and store it in the variable NUM. The sum of the array is stored in the double precision (32-bit) variable XSUM, to see value: XSUM 2@ D.

To subtract a constant background value n throughout the calculation (while not altering the array itself) use the command

n BKG !

before using SBAR; default BKG is zero, after changing it, it will stay at current value until altered again or until PROCESSING is reloaded.

FIND (filename) SBAR AVERAGE VALUE = 1025 Calculates sum and average
[4.8.3] Normalization of arrays

For example, for removing variations due to photocathode sensitivity on an electronograph, divide scan array by scan photocathode map and multiply through by average of latter scan.

Suppose first scan is in a disk file called STAR, the photocathode map in a file called SKY, and you wish to place the result in a file called FINAL, which you first need to create with size equal to those of STAR and SKY, starting at block n.

```
LISTFILE
FIND STAR
n INTER FINAL

FIND SKY SBAR AVERAGE VALUE = 952
FIND STAR SOURCE FIND FINAL DESTINATION
FIND SKY DIVIDE PASS
```

If you wish to do further DIVIDE's with NUM at the default value

```
10000 NUM !
```

See [4.7.2] for details
Makes disk current image with STAR's parameters.
Enter FINAL in index file with parameters equal to STAR.
Stores average value of SKY by NUM.
Divides STAR by SKY
multiplying through by NUM and putting
the result in FINAL.

[4.8.4] NOTES
(i) If you wish to use only part of the image, ie a subset of window, see notes on windowing in [4.11]

(ii) Words like ADD, DIVIDE, SUBTRACT do not do anything themselves except assign an operation called PROCESS which is executed for each row in PASS. At the end of a PASS, a default operation which does nothing is re-assigned so that subsequent PASSes will not perform any operation. This default operation is called NOTHING; using it immediately after ADD, etc will cancel the effect of the latter.

These words also assign the current image to be the OPERAND (they include the word SECOND).
REMOVAL LOAD/SLICING LOAD

SLICING OPT LOAD loads basic display and defect removal routines. (disk files only)
REMOVAL LOAD loads the above, plus routines to
   (1) Makes PDS move to appropriate location for direct
       visual examination of defects
   (2) Make available a file of defect coordinates
   (3) Fit gaussians to scan lines

If you only want the basic SLICING routines

   SLICING OPT LOAD
   Go to [4.9.1]

Loads limited options

Otherwise, for REMOVAL

   REMOVAL LOAD
   Go to [4.9.1], continuing through to [4.9.2]

Loads complete package

[4.9.1] SLICING

To view an array line by line:-

   FIND (filename)
   FIRST
   N

Select appropriate disk file.
Displays first line, sets graphics
scalings and row sets no R to 0.
Displays next line. Repeat N
for consecutive lines clearing
pages as required.

[If response is OK with no lines plotted, file is complete]

Subsidiary Display words:-

   PLOT
   n ' R + !
   (n-1) ' R ! N
   m n X SIZE
   m n Y SIZE

Replots current line. Useful after
clearing page.
Skips n lines
Plots line n
Resets x-scale so that pixels m ≤ i ≤ n
are plotted
Resets Y-scale so that density values m
through n are plotted. (Default to 0.4000)
POI NTS
HISTOGRAM
FOLLOW

To removed defect, plot line and

[4.9.1.1]  
FIX
Position cursor on L.H.S. of defect
Tap any key (except RETURN)
Position cursor on R.H.S. of defect
Tap key

[New interpolated line is stored in the disk file and replotted on screen.]
Old line is lost].

These are all the available words in SLICING.
If you have loaded REMOVAL continue to [4.9.2]

[4.9.2]  
REMOVAL

This assumes that source of disk file is still on PDS and co-ordinate system
has not been changed.

[4.9.2.1]  
Plot lines until possible defect appears

If you wish to enter removed-defects co-ordinate into DEFECTS file to initialize
that file then

?DEFECTS (or) (or)

LINK-CLEAR

Otherwise

NOX
Centre cursor over apparent defect

Sets Plotting mode +  +  +  +
Sets Plotting mode
Sets Plotting mode
(Default.)

Puts up cursors

Program does linear interpolation
between 2 points.

If desired clear page and PLOT

Displays cursor and contents of
DEFECTS
Initializes file

Puts up cursor
This position is not used in any
subsequent defect removal and hence
position of cursor is not critical.
Tap any key (except RETURN)

If defect to be removed go to [4.9.2.2]
Otherwise go back to [4.9.2.1] and locate another defect candidate

[4.9.2.2] If defect to be entered into DEFECTS file go to [4.9.2.3]
Otherwise use FIX as described before in [4.9.1.1]

[4.9.2.3] NIX

To view DEFECTS file

?DEFECTS

To remove further defects go to [4.9.2.1]

If a second plate is then set up using ALIGN so that the co-ordinate translation has been calculated using ROTATE

n @DEFECT

PDS will move to co-ordinate corresponding to that point and the user may decide from visual appearance of area whether removal of feature is desired.

Identical to FIX except, stores current PDS co-ordinates in DEFECTS file. (Do not move PDS between NOX and NIX otherwise wrong co-ordinates will be entered.)

Defects are entered sequentially numbered after last defect in file.

This will move PDS to position corresponding to coordinates in record n of defects file, with the translation added.

In this way the user can see if the same defects are present on different plates.

If no ROTATE has been performed since loading TAYLOR then this instruction will take PDS to nth defect of existing plate.
Fitting a Gaussian to a Scan Line

Plot Line by using (N or PLOT)

APPROX FIT

CURVE

RESULT

CENTER  PEAK  BASE  HW

1.52  3000  257  3.42
0.00  0.00  0.00  0.00

Calculates 1st approximation and does least squares fit.
Plots fitted curve.
Outputs values of parameters \( x_0 \)
(value in column CENTER) is distance from center of curve in units of \( \frac{1}{10} \) \( \mu \) (i.e. if result is \( x \), pixel number in line is \( x \div 10 + \frac{1}{2} K \) where \( K \) is pixel length of row).

PEAK Is height of the peak above the base.
BASE Is the base. Units for these are the same as the scan data

(0 \( \leq \) density \( \leq \) 4095)

HW Halfwidth, units \( \frac{1}{10} \) \( \mu \). Second line gives residuals.

Repeat FIT, CURVE, RESULT as required, for example:

FIT CURVE FIT CURVE FIT RESULT CURVE etc

Selects disk image to be plotted.

CONTOUR LOAD

FIND (filename)

If only a part (window) of the file is to be plotted, see [4.11] to do this.
SCREEN n WIDTH

sets up the screen to be n wide by \( \frac{3}{4} \) n high (in pixels)

So, for an \( m \times m \) image-
\( \frac{3}{4} \) WIDTH will set the scale so that all the file will be plotted, and the right hand quarter of the screen will be blank.

\( m \) WIDTH would fill the whole screen but only plot the first \( \frac{3}{4} m \) rows

Optional: default 100
Sets contour interval to be n (i.e. difference of n in the numbers in the array.)

Optional: default -32768 32767 CLIP "low" and "high" are numbers defining cutoff points, for example 200 4000 CLIP will treat all numbers <200 as 200, and all >4000 as 4000

Plots contour map

[4.10.2] Contour plotting directly from tape

512 MAXIMUM SCANSALOT LOAD CONTOUR LOAD

This unloads TAYLOR so if you subsequently wish to use programs in TAYLOR you must repeat TAYLOR LOAD.

Select tape file; described in [4.12]
Example of a plot using CONTOUR
If windowing is desired, see [4.11]

SCREEN n WIDTH
n SPACE : low high CLIP

TAPE PLOT

Disk file manipulation

[4.11]

Windowing

[4.11.1]

Given a 2D array $A_{ij}$ of pixel dimension $p \times n$ (such that $0 \leq i < k$ and $0 \leq j < p$) a subset $B_{ij}$ of pixel dimension $a \times b$ (such that $m \leq i < (m + a)$ and $n \leq j < (n + b)$) may be accessed, provided $m$ and $n \geq 0$, $(m + a) < k$ and $(n + b) < p$.

This is used for plotting part of a large array, combining part of an array with a smaller array fitting a function to part of an array, etc.

Windowing words are:

\begin{align*}
n & \hspace{1cm} m \hspace{1cm} \text{ORIG} \\
& \text{Sets } W0: \text{ defines window origin at row } n, \text{ pixel } m
\end{align*}
[4.11.2] Digital display

FIND (filename) m n VOILA

Sets \( \text{WN} \), defines size of window to be \( b \) rows of \( a \) pixels each.

Sets \( \text{WN} \) to \((0,0)\) and \( \text{WN} = \text{CN} \)
ie; makes the window the default value, the whole array.

C clears page and displays window of file with origin at pixel \( n \), row \( m \). Resets \( \text{WN} \), \( \text{WN} \) to default at end.

[4.11.3] Examples of Copying Files

FIND (file 1) SOURCE
FIND (file 2) DESTINATION PASS
FIND (file 1) 32 32 WW SOURCE
FIND (file 2) 32 32 WW DESTINATION PASS

Copies (file 1) to (file 2)
Copies "top left" (1st 32 pixels of 1st 32 rows of file 1) to corresponding area of (file 2)

[4.12] SCANSALOT LOAD

Taping by SCANSALOT LOAD, uses the second terminal's working storage area so the second terminal may not be used while you are using the tape.

[4.12.1] To set up parameters for a scan to tape, go to [4.7.1]

If tape is a new tape (no files on it to be kept)

REWIND

Rewinds tape
Initialises tape variables

NTP

Writes 1 dummy record and 2 filemarks then backspaces twice. The record is for compatibility with RTSCAN tapes which have no filemark before the first file.
Otherwise ensure tape is positioned at some point before the final double filemark. Program will look for a double filemark before starting to write.

Then

\[ \text{WRITE (identifier) (cr)} \]

(Identifier) is a string of \( \leq 40 \) characters (including spaces). Tape will run forward until a double filemark is found, and will then backspace so that the second filemark will be overwritten. The scan will be performed using the PLATE parameters set previously. The data will be written onto tape in Scansalot format.

After the scan is completed, a double filemark will be written at the end of the file, and the tape will then be backspaced twice, so that a second WRITE will not require any repositioning of the tape.

The rest of this section [4.12] can be used without doing a TAYLOR LOAD, by doing a 512 MAXIMUM SCANSLOT LOAD instead.

[4.12.2] Disk-to-tape pass

\[ \text{FIND (filename)} \]
\[ \text{WRITE (identifier)} \]

Disk-to-tape pass

Selects appropriate disk file. Looks for end of tape (double filemark) backspaces once and copies disk file onto tape. Finishes by writing a new double filemark and backspaces twice.

(Identifier) is \( \leq 40 \) character string including spaces terminated by carriage return.
### Tape positioning

**REWIND**
Rewinds tape, initialises tape variables EOFS and REC.

**n SKIPS**
- \( n > 0 \) Passes forward over \( n \) filemarks. Remains positioned immediately after the last (i.e. at the beginning of a file.)
- \( n < 0 \) Goes backward over \( n \) filemarks then forward 1 space, so positioning tape at the beginning of a file.
- \( -1 \) SKIPS Returns to beginning of current file.

**TWIDE**
Returns to beginning of current file, goes through file checking number of records and coordinates in header, uses these to set TAPE parameters \( \text{TN}, \text{CN}, \text{CF}, \text{DX} \). Types first identifier, returns to beginning of the file.

**n SKIPS TWIDE** is the taping equivalent of FIND (filename) for disk files.

**T-INDEX**
Prints directory of files on tape; rewinds tape and goes forward until double filemark found, then backspaces twice.

**n FILE**
Positions at end of first record of file \( n \), \( n \) is absolute file number with dummy header file being zero. However in some circumstances when interrogating individual records in a file, one goes past the end of file. Subsequent use of FILE may give the wrong result.
T-INDEX
0 DUMMY FILE ; BEGINNING OF TAPE
1 JCJ TEST SCAN 1978 JUL 26
2 VERTICAL SCAN, ODD=1
3 ANOTHER VERTICAL SCAN
4 SIMPLE SCAN
5 AWRITE TEST

END OF FILES OK

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>10</td>
<td>10-26860</td>
<td>-3398</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>-12</td>
<td>-5-14010</td>
<td>-11834</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>-12</td>
<td>-5-14010</td>
<td>-11834</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>10</td>
<td>10-26860</td>
<td>-3398</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

Example of T-INDEX
LABEL
n RECS

SHOW

[4.12.4] Tape to disk pass

n SKIPS TWIDE

SOURCE

If disk array not yet allocated

LISTFILE
n INTER (filename)
FIND (filename)
DESTINATION
FIND (filename)
EQUAL DESTINATION
PASS

Then

Reads and prints next records identifier.
Moves forward n-1 records and reads next record
1 RECS reads next record
2 RECS reads next record but 1
0 RECS reads record just read
-1 RECS reads record before one just read
Types identifier and dumps scan data currently in working storage. To be used after RECS.

Selects file, prints identifier, sets TAPE parameters
Copies TAPE parameters to INPUT
See [4.7.2] for explanation, lists disk fil
See [4.7.2] for explanation, enters disk fi
Selects disk file
Copies DISK parameters to OUTPUT
Selects disk file
Sets DISK and OUTPUT parameters
Copies tape file to disk file
SMOOTHING (boxcar convolution)

Averaging in i and j by constant factor a; disk arrays only.
Description: let contents of pixel i, row j in array A be $A_{ij}$

$A$ is source array; $B$ is smoothed array
Each array consists of $\rho$ rows of $K$ pixels each
$[x]$ means integer part of $x$

Action of program:

$B_{ij} = A_{ij}$ for (i) \[ \emptyset \leq j < \lfloor \frac{3}{2}a \rfloor, \emptyset \leq i < K \]

(ii) \[ \emptyset \leq i < \lfloor \frac{3}{2}a \rfloor, \emptyset \leq j < \rho \]

(iii) \[ K - \left\lfloor \frac{a + 1}{2} \right\rfloor + 1 \leq i < K, \emptyset \leq j < \rho \]

Partially smoothed s.t. $B_{ij} = \frac{1}{\sum_{k=i-\left\lfloor \frac{a}{2} \right\rfloor}^{\rho-1} \sum_{l=j-\left\lfloor \frac{a}{2} \right\rfloor}^{\rho-1} A_{kl}}$

for $\lfloor \frac{3}{2}a \rfloor \leq i < K \left\lfloor \frac{a + 1}{2} \right\rfloor + 1, \rho - \left\lfloor \frac{a + 1}{2} \right\rfloor + 1 \leq j < \rho$

$B_{ij} = \frac{1}{a^2} \sum_{k=i-\left\lfloor \frac{a}{2} \right\rfloor}^{\left\lfloor \frac{a + 1}{2} \right\rfloor - 1} \sum_{l=j-\left\lfloor \frac{a}{2} \right\rfloor}^{\left\lfloor \frac{a + 1}{2} \right\rfloor - 1} A_{kl}$

Smoothed s.t. for $\lfloor \frac{3}{2}a \rfloor \leq i < K - \left\lfloor \frac{a + 1}{2} \right\rfloor + 1, \lfloor \frac{3}{2}a \rfloor \leq j < \rho - \left\lfloor \frac{a + 1}{2} \right\rfloor + 1$
Inside fully smoothed region:

For odd values of a - Bij is the average of the a x a square of points in array A with center \((i, j)\)

For even values of a - Bij is the average of the a x a square of points in array A with center \((i - \frac{1}{2}, j - \frac{1}{2})\) so for even a the image is translated by \(\frac{1}{2}\) pixel in i and j. If an array is to be repeatedly smoothed by an even factor it is recommended that the array be 'flipped' each time before smoothing so that \(A_{ij} \rightarrow A_{(K-i-1)(p-j-1)}\). Thus the successive translations will cancel instead of adding. How to achieve this is described below.

[5.2] Tape files

This program only works on disk files, to smooth a tape file first copy it to disk, either 512 MAXIMUM SCANSALOT LOAD or TAYLOR LOAD SCANSALOT LOAD, and then go to [4.12.4] for instructions on tape-to-disk pass.

[5.3] Loading

a SMOOTHING LOAD 512

a is the full width of the boxcar in p and k \((a \geq 2)\).
Output is maximum row size allowed. This is 512 or less depending on value of a. For \(a > 8\) the buffer sizes must be < 512 to fit in core.

[5.4] Disk files

Allocate all disk files you will require using LISTFILE

FIND (file 1)

Lists images index, for explanation see [4.7.2]

Sets current image parameters to those of (file 1). (file 1) is the source file i.e. the one you wish to smooth.
n INTER (filename)

[5.5] Smoothing

(i) Single smoothing
FIND (file 1) SOURCE
FIND (file 2) CONVOLUTE

(ii) Smoothing into same file
FIND (file 1) %CONVOLVE
FIND (file 1) %CONVOLVE %CONVOLVE %CONVOLVE

(iii) Flipping and then smoothing
This is for use with repeated smoothing by even values etc.
FIND (file 1) SOURCE
FIND (file 2) RCONVOLUTE
SOURCE FIND (file 1) RCONVOLUTE

n is block number, select as described in [4.7.2]. INTER allocates a file called (filename) the same size as (file 1). Use INTER to allocate all the files you will need.

Defines (file 1) as the INPUT file, i.e.: file to be smoothed
Copies (file 1) to (file 2) and smooths (file 2). CONVOLUTE is equivalent to DESTINATION PASS %CONVOLVE (see below).
Smoothes (file 1), overwriting original data. If you wish to save original data, copy it into another file or use CONVOLUTE as in (i) and then FIND (file 2) %CONVOLVE.
Repeatedly smooths (file 1).

(file 1) will be overwritten later, to save original first copy it.
Copies (file 1) to (file 2) reflecting it on the way so that pixel (i,j) of (file 1) is put in pixel (k-i-1,p-j-1) of (file 2), i.e. reflected in the center point of the array. Then smooths (file 2). Flips (file 2) into (file 1) and smooths (file 1).
SOURCE FIND (file 2) RCONVOLUTE
SOURCE FIND (file 1) RCONVOLUTE
SOURCE FIND (file 2) RCONVOLUTE

If the final file is (file 1), the array will be the right way round, but
if it is (file 2), one more flip must be done to correct it.

SOURCE FIND (file 1) RC-FLIP PASS

3D plot

There are 3 variants loaded by:

Either
or
or
3D LOAD
3D/J LOAD
3ISLCW LCAD

6.1]

Form of plot

Orientation of axes

Repeatedly smooths array, with the
½-pixel translation caused by each
smoothing cancelling.

Takes current file, flips into (file 1)

Basic FORTH 3D plot
Hidden line 3D plot
Alternative hidden line plot

Loads taping routines

X represents pixels/row
Y represents density units (numbers in array)
Z is direction of decreasing row number i.e.
later rows are plotted towards the more
negative Z.
n m X SIZE sets the X scale, n being the abscissa of the left hand side of the screen, m being the abscissa of the right hand side.

Thus $\varnothing \text{ WN } \varnothing \times \text{ SIZE}$ will cause a plotted row to fill the screen from left to right. ($\text{WN } \varnothing$ gives the value of K)

n m Y SIZE sets the Y scale
n m Z SIZE controls the spacing between the rows.

Default values: $\varnothing \ 32 \times \text{ SIZE}$
$\varnothing \ 8192 \ Y \text{ SIZE}$
- 64 64 Z SIZE

The pitch and yaw of the image can also be altered.

Pitch inclines the Z axis from the horizontal
Yaw inclines the Y axis from the vertical, in the plane normal to Z.
Default pitch is $60^\circ$, yaw is zero.

[6.2]

Commands

Commands for the three variants are identical except in the case of PITCH and YAW.

SCREEN $p \ K \ WIDE$

No more than $p$ rows and no more than $K$ pixels/row will be plotted.
For a disk file: FIND (filename)
For a tape file: n SKIPS TWIDE

Then [6.2.1]
If only a part of the file is to be plotted:
\[
\begin{align*}
  n & \text{ m ORIGIN} \\
  b & \text{ a WW} \\
  n & \text{ m X SIZE} \\
  n & \text{ m Y SIZE} \\
  n & \text{ m Z SIZE}
\end{align*}
\]

If 3D variant being used
\[
\begin{align*}
  S & \text{ c PITCH} \\
  S & \text{ c YAW}
\end{align*}
\]

Otherwise (ie 3D/J or 3DSLOW variants)
\[
\begin{align*}
  \text{nn.nn PITCH} \\
  \text{nn.nn YAW}
\end{align*}
\]

Selects disk file to be plotted.
n is relative file number; see [4.12] for explanation

see [4.11] for explanation

see [4.11] for explanation

Sets up X axis co-ordinates
n = abscissa of L.H.S.
m = abscissa of R.H.S.
(Default 0 32 X SIZE)

( Default 0 8192 Y SIZE)

( Default -64 64 Z SIZE)
For explanation see [6.1]

Default 866 500 PITCH

Default 0 1000 YAW
S = 1000 \sin \theta \quad c = 1000 \cos \theta

Default 60.00 PITCH

Default 00.00 YAW
nn.nn is angle in degrees. There must be exactly 2 decimal places - do NOT omit trailing zeros.
Then For a disk file: DISK PLOT
For a tape file: TAPE PLOT
Then to replot using different window, scales, pitch or yaw: go to [6.2.1]

Photometry

Description: Least squares fitting of a 2-D gaussian to scan data from a disk file. The file may be windowed (see [4.11]).

Consider a 2-D array of data $A_{ij}$ where $0 \leq i < k$ and $0 \leq j < p$.

<table>
<thead>
<tr>
<th>Fitting Parameters</th>
<th>Parameter definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0 = 6400 \frac{i}{k}$</td>
<td>$(x_0, y_0)$ is centre of gaussian</td>
</tr>
<tr>
<td>$y_0 = 6400 \frac{j}{\delta_j / \delta_i}$</td>
<td>$\frac{\delta_j}{\delta_i}$ ratio of sample intervals in microns of the $j$ and $i$ directions.</td>
</tr>
</tbody>
</table>

PEAK Height of peak above base (units as in $A_{ij}$)
BASE Base (units as in $A_{ij}$)
HX Half width at half height in $i$ direction (units as in $x_0$)
HY Half width at half height in $i$ direction (units as in $y_0$)
SR Correlation. Stored as integer. Value multiplied by 1000.

Function:

$f(i,j) = (BASE) + (PEAK) \cdot 2^\xi \leq C$

where

$\xi = \frac{(X-x_0)^2 + SR(X-x_0)(Y-y_0)}{Hx} + \frac{(Y-y_0)^2}{HY}$
\[ X = \frac{6400i}{k} \quad Y = \frac{6400j}{k} \quad 6i \]

C is a cut-off value (Default 4095)

[7.1]
PHOTOMETRY LOAD

[7.2]
\( n \) CUTOFF !
FIND (filename)

If windowing required, see [4.11]

APPROX
FIT
RESULT

PHOTOMETRY

<table>
<thead>
<tr>
<th>X0</th>
<th>Y0</th>
<th>PEAK</th>
<th>BASE</th>
<th>HX</th>
<th>HY</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.0</td>
<td>16.0</td>
<td>3000</td>
<td>200</td>
<td>8.00</td>
<td>8.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.03</td>
<td>0.23</td>
<td>14</td>
<td>2</td>
<td>0.23</td>
<td>0.04</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Loop to [7.2.1] until residuals are sufficiently small.

[7.3]
Creating 'Faked' Gaussians

To create a file containing a digitized gaussian which has been defined by a FIT, immediately after FIT or RESULT.

loads relevant programs
Sets cutoff value, default 4095
Selects disk file
Calculates a first approximation
Performs the fit
Outputs results.

\[ \sqrt{S^2} \text{ where } S^2 \text{ is sum of squared deviations of previous fit} \]

4 FITS are usually enough; a single FIT takes about 3 mins. for a 32 x 32 file, 9 mins. for a 64 x 64 file.
If cutoff value to be changed

\[ c \text{ CUTOFF} ! \]

If you wish to overwrite the fitted data in a file by its fitted gaussian

\[ FAKE \]

Otherwise create a new file by:-

\[ LISTFILE \]
\[ n \text{ INTER (file 2)} \]
\[ FIND (file 2) \text{ FAKE} \]

[7.3.1] \textbf{Test gaussians}

These can be created by setting up the appropriate parameter values as follows:

\[ \text{FIND (filename)} \]

Either

\[ x_0 \text{ RSU x0 !} \]
\[ y_0 \text{ UM y0 !} \]
\[ h_x \text{ W0 @ + RSU HX !} \]
\[ h_y \text{ W0 l + @ + RSU HY !} \]

32767 CUTOFF ! gives c its maximum value.

For every point in the file the function is calculated and the value is stored in that point of the array.

Explained in [4.7.2]

" " "

Puts gaussian in (file 2)
or

xo XO! yo Y0!

hx HX! hy HY!

Then

p PEAK! b BASE! r SR!

If cutoff to be reset:
c CUTOFF!

To check the above entries

RESULT

If you wish to add normally distributed noise:

s SD!

Then

FAKE

8]

Additional Facilities

8.1]

Very long scans

For scans with row length too long to fit in the buffer (working storage) the rows must be split up into several sections, each section being scanned and stored as a single record on tape (or disk) Scans as follows:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

numbers are record (section) numbers.

xo, yo, hx, hy in scaled units described at beginning of [7]

Sets PEAK, BASE, SR

where \( r = \text{correlation} \times 1000 \)

Ignore 2nd and 3rd rows of output

Multiples each value to be stored in the array by \((1 + \frac{\delta(s)}{1000})\) where \(\delta(s)\) is normally distributed between \(-3\) with mean zero.

Creates gaussian, stores in disk file.
Records 0-3, 5-8, 10-13 are of size equal to the buffer size. Records 4, 9 and 14 are shorter since the total row length is in this case, not an integral multiple of the buffer size. (This example might be: 1924 RLENGTH !3 4500 LWIDE). For scans to tape the scanslot header for records other than record 0 will be incorrect.

\$ BIGSCAN LOAD
427 LOAD

K RLENGTH !

Sets up parameters as in [3.4] except replace WIDE by LWIDE such that

\( \rho K LWIDE \)

For scanning to tape: Scan as in [3.5] except replace WRITE by LWRITE such that

LWRITE (identifier)

For scanning to disk: Scan as in [4.7.2] except

Replace: FIND (filename) EQUAL DESTINATION PASS
by: FIND (filename) LEQUAL DESTINATION LPASS

\[\text{Multiple Scans}\]

Capability to set up for a number of scans to tape as in BIGSCAN, but instead of doing each scan before setting up for the next, store the parameters of each scan in a file called SCANS file, then a single instruction will cause all the scans to be done in turn.

\(i (<2 . . . ) \text{ is maximum buffer size, loads very-long-scan programs}\)

Buffer size to be used, i.e. standard section length, \(K \leq \rho \). For scans to disk \(K \leq 512\).

\(\rho \) is actual number of rows to be scanned, of \(K \) pixels each. LWIDE will automatically divide each row into sections.

scans onto tape as described above
The following parameters may be different for each scan:

The entire "image parameter field": \( \hat{CN}, \hat{DX}, \hat{CQ}, \hat{WQ}, \hat{WN}, \text{KIND} \)

i.e.

(i) size of scan, set by WIDE

(ii) sample intervals, set by DXDY!

(iii) origin, set by either CENTER or c CORNER

where c is UR, UL, LL or LR. Scans may be identified by different corners.

(iv) direction of scan, set by VERTICAL or HORIZONTAL

and also

(v) The Scansalot 4-byte-character identifier

\[ [8.2.1] \]

\[
\begin{align*}
K & \text{ BIGSCAN LOAD} \\
428 & \text{ LOAD} \\
& \text{ CLEAR} \\
\text{Set up as in [3.4] for first scan}
\end{align*}
\]

\[ [8.2.2] \]

\[ n \text{ ENTER (identifier)} \]

\[ n \text{ is scan no. (1 for first scan)} \]

\[ \text{Enters parameters and identifier in file} \]

\[ \text{Set up PDS for } n \text{th scan:} \]

Either

\[ HORIZONTAL \]

or

\[ VERTICAL \]

or nothing if previous scan was the same direction

Then

\[ c \text{ K WIDE} \]

\[ \delta x \delta y \text{ DXDY!} \]

Either

\[ CENTER \]

or

\[ \text{either UR or UL or LL or LR CORNER} \]
Then Loop to [8.2.2] incrementing n by 1 each time until all scans (≤ 16) have been entered.

If tape is a new tape

   REWIND
   NTP

Then

   SCAN

   CLEAR

[8.3]  

Array flipping

Capability to flip data in an array on disk
in i : flips each row end for end
in j : flips each column end for end
or in both i and j

[8.3.1]

TAYLOR LOAD

431 OPT LOAD

FIND (file 1) SOURCE FIND (file 2) DESTINATION

Either

   ROW-FLIP

or

   COL-FLIP

or

   RC-FLIP

see [3.5] for explanation

Does each scan in turn, writing onto tape, with the appropriate parameters; with a filemark between each scan and two at the end.

Clears scans file for next user.

Loads flipping option

Allocate file 2 by INTER as in [4.7.2]

Sets up for flip in i

" " " " " j

" " " " " i and j
Then KEEP PASS

To flip an image scanned with origin at one corner to be as if it was scanned from another, use RECTIFY:

FIND (file 1) SOURCE FIND (file 2) UR DESTINATION RECTIFY

To transform an array scanned in Y to one scanned in X:

FIND (file 1) SOURCE FIND (file 2)
DESTINATION TURN

If (file 1) is A and (file 2) is B, the effect of the command TURN is

Aij becomes Bji where subscripts represent row and column numbers.
Due to the simplicity of the method, the operation is prohibitively slow for arrays > (64 by 64).

Tape file graphic record display

This utility combines SCANSALOT with the display words in SLICING (see [4.9], [4.12]).

454 LOAD
n SKIPS TWIDE
FIRST

Updates index file and copies array from (file 1)
to (file 2) flipping on the way.

Copies (file 1) to (file 2) flipping it approximately to make the scan as if it was scanned from the UR corner.
Using UL, LL or LR instead of UR would cause the final file to have the specified corner as its origin.

Selects tape file, see [4.12]
Sets scales, plots 1st line. See [4.9]
The following commands are available, and are described in [4.9]

N
PLLOT
AXES
BIG GRID
POINTS
FOLLOW
HISTOGRAM
n m X SIZE
n m Y SIZE

There is one additional command:

SAMPLE

[8.5] coarse file

Description: Used most often to compare scans from an electronograph with the corresponding area on a cathode map. Disadvantage is that it is not as accurate as ALIGN, but advantage is that cathode map is only scanned once, and file is reused for different high-resolution scans. ALIGN is generally used to eliminate fine variations within a single scan area while the coarse-file method is used to eliminate grosser variations across a larger area of photocathode.

The whole or a large part of the photocathode map is scanned at a coarse spatial resolution and stored on tape or disk.
A second scan at a higher spatial resolution is stored on disk or tape. The program creates a third file, on disk, containing photocathode map points corresponding to those of second scan, the numbers being the weighted average of the 4 nearest-neighbour points in the coarse file.

Creating the coarse file

Do a normal scan as in [3.4], [3.5] or [4.7.1], [4.7.2] except: ZILCH over a standard mark common to all electronographs to be scanned, (for example, upper left fiducial mark). In addition electronographs must be approximately aligned with one another using the fiducial marks or photocathode defects. The scan must be in X and must be specified by either the CENTER or the UR CORNER.

Creating the nearest-neighbour file from the parameters of a scan (called "source scan") done from a corresponding electronograph.

The source scan is done either onto tape or disk. The sample intervals must be less than those of the coarse file. The source scan must be in X and the origin must be defined by either CENTER or UR CORNER. ZILCH must be done at the same standard point as was done for the coarse file.

The nearest neighbour file may now be created using the data from the coarse file and the parameters of the source scan.

```
 n BIGSCAN LOAD
  457 LOAD

 If source scan on disk:  n FIND (filename) SOURCE
 Otherwise, i.e. on tape:  n SKIPS TWIDE SOURCE

 If destination file not allocated in index. use  LISTFILE n INTER (file 2)
 FIND (file 2) DESTINATION
```

n is buffer size
loads coarse file routines
Selects source scan file
Selects source scan file
see [4.12] for explanation
See [4.7.2] for explanation
Selects destination file
If coarse file on disk: FIND (coarse file) SECOND

If coarse file on tape: n SKIPS TWIDE SECOND

Then NEAREST

To use (file 2) created by NEAREST to take out effects of photocathode response from (file 1) do TAYLOR LOAD PROCESSING LOAD and see section [4.8]

Density histogram
(Disk files only)

DHIST LOAD
FIND (filename)
DENSITY

[8.6]

Sets up coarse file parameters
Sets up coarse file parameters
Calculates and stores nearest neighbour file in (file 2)

Sets up array of 'buckets', goes through file counting number of points in file with densities in range of each bucket, and plots results as a histogram of number against density. The default bucket size is 10 (minimum value is 0, set by n 'QUAN !)
i.e. an ordinate of 200 means that there are 200 points between the abscissa density value and that value +10 (or + the value of QUAN)
Data may be replotted as follows:

- a b ROI PLOT
- a b SEE PLOT
- min max Y SIZE
- ENTIRE PLOT

Plots density values a to b only, without change of scale.

Plots density values a to b, expanding scale to fill screen.

Sets Y scale

Equivalent to Ø 4095 SEE PLOT

DENSITY plots all values from Ø to 4095

1978 July 26