

Franco Zavatti  
Dipartimento di Astronomia, Università di Bologna

# **D I S**

An MS-DOS Image Display System  
Version 2.1

## **User Manual**

Technical Report  
October 1992

# 1.INTRODUCTION

Dis is an MS-DOS Display System for digital astronomical images with maximum dimension 512 rows by 320 columns. It is based on PLOTPE and ASMPLOT, Fortran and Assembler libraries described elsewhere (Rinaldi and Zavatti, 1991). The included command parser can execute commands after they have been read, along with the respective parameters, from both the keyboard or already existing disk file. All commands enter into an internal vector (called '*command buffer*') which can be modified by dedicated commands (Read, Delete, Insert, Edit) or saved on disk. Command buffer can contain 250 commands as maximum.

Images can be read and written, only as *two-bytes integer data*, in two formats: Direct Access with no room for image header (hereafter DA) and FITS (tizio e Caio, 1999) with record length 2880 bytes. The first format does not maintain the knowledge of the number of columns per row, so user must enter this number or confirm the default one. Dis assumes that files with extension .CCD and .NTT have 320 columns per row, but this value can be interactively changed.

The only true Dis problem comes out from the initial choice of keeping the whole image in computer memory: this freezes 320 KiloBytes out of the (nominal) total amount of 640 KB and leaves a small place to the other Dis functions. The '*hunger*' of memory has brought to disrespect the starting statement that Dis could run with any PC processor, from 8088 to 80x86. Present Dis version requires 510,402 bytes at run-time, so, in practice, at least an 80286, 2 MB memory and MS-DOS 5.1 (or similar operating system) with its capability to allocate high memory for drivers or TSR programs, are an absolute need. A mathematical co-processor (80x87) is also strongly recommended.

Any image and data vector created by Dis can be saved on disk files.

Dis has 72 commands: a list is reported in Appendix A, with the number of parameters (9 means alphanumeric string) and one - line description. The commands are written into the file CMDS.DIS, and Dis routines check the existence of a given command by comparing the actual line of the command buffer to the content of this file. CMDS.DIS, so, should have '*hidden*' and '*read only*' attributes and the file should not be modified in any case. in any case.

On-line help is available for any command, within the limitation that shortcuts are not allowed in the names of help files. For example, PL or PLAY are normally used in place of PLAYBACK, while Help PL gives an error. The correct command will be Help PLAYBACK. The help files are also collected in the TEX file DISMAN.TEX. If help command is entered without any parameter, a list of all available commands is displayed.

A very crude and '*dirty*' hardcopy routine is present, but output on printers is committed to DOS Print Screen (version 4.1 or higher) or to commercial software.

For any problem or comments, the author can be reached at the following mail and e-mail addresses:

Franco Zavatti  
 Dipartimento di Astronomia  
 Via Zamboni 33  
 40126 BOLOGNA (Italy)

Internet: zavatti@alma02.cineca.it

Decnet: 3719::zavatti

## 2. INSTALLATION

Dis can be installed by putting the distribution diskette into drive A:, setting the computer actual disk also to A: and typing:

```
A: > INSTALL disk <return>
```

to install Dis into the hard disk *disk*, where *disk* may be any of C:, D:, ..., G:.

The batch program INSTALL.BAT provides the creation of both the directory \DIS\, containing the main program, its related files and external programs (see Appendix B), and the subdirectory \DIS\DISHLP\ with the help files.

A list of actually available VGA modes is then shown, and user asked to give a choice by typing the (decimal) code of the preferred graphical environment. The code INSTALL uses as default is 18 decimal (12 Hex), i.e. the standard VGA mode, with a resolution of 640 rows by 480 columns and 16 colors. Any VGA card knows this mode and the lower ones, while the higher modes (called Super VGA or SVGA) are typical of each manufacturer. So 18D mode is the better starting mode and its choice recommended. To accept the default choice a comma can be typed, followed by <return>: Fortran allows such a way to confirm formerly given numerical value(s) (18 <return> can be also typed, of course).

Some more words about SVGA codes: they are often listed in the VGA reference manual as a table of hexadecimal digits with related resolution, colors and text characters dimension. To change actually available SVGA codes, the file MODIVGA.PLT should be edited, referring to the above mentioned table, as far as the first column (and also the last one) is concerned. Also the file \dis\dishlp\ mode.hlp should be updated in the same way.

A new version of CONFIG.PLT is then written with the actual VGA code, and both .PLT files (i.e. Config and Modivga) are copied to the root of the chosen hard disk.

The file DSKDIS.INI, containing the name of the hard disk where Dis has been installed, is also created. This file **must** be copied to the directory Dis is called from and modified any time Dis directories are copied to another disk (only when copied, because Install provides by itself the creation of a correct DSKDIS.INI).

The command `DEVICE = path Ansi.sys` should be included in Config.sys file.

## 3. OPERATIONS

To start Dis from within its own directory \dis\, type `dis <ret>`. Running from another directory, Dis path should be known, either adding it to the PATH command in `autoexec.bat` or by a batch file (say `dis.bat`) containing the following lines:

```
COPY c: \dis\dskdis.ini
c:\dis\dis
```

This also automatically copies `dskdis.ini` in the working directory.

After starting, Dis clears the screen (no welcome or copywrite message appears), shows the prompt '>' and waits for a command.

Dis sessions may be interactive, batch and mixed: in the first case any command is executed just after <ret>. In the other situation, i.e. when a command file has been already created, the command `READ file - name` loads the existing list of commands and `PL(ayback)` execute it. During mixed sessions, a command file is loaded and interactive commands are given.

Commands can be entered with the minimum number of characters needed to distinguish them one from another. An ambiguous command will be interpreted with the first matching occurrence in the file `CMD5.DIS` (Appendix A). For example, `IN 1` (which should mean `INSERT 1` in user's mind) will be translated as `INFO`.

## 4. BUGS

Some bugs affect Dis: when known, they have been reported in the respective help files.

Two of them appear more intriguing and actually do not have an explanation: the first one (reported in `MODE` help file) refers to the failure in text positioning with the 800 x 600 modes. No particular software has been written for these modes, any setting being referred to the only one assembler subroutine which provides calls to BIOS interrupts.

The other bug, not reported in help files, concerns the incapacity of correctly drawing lines at higher resolutions, with some SVGA card (e.g. with the Trident TR4000, but the drawing routines were originally developed on a TR3000, i.e. on practically the same chip, and there worked well, of course). The last bug affects only plots, not pictures.

Some problem may also arise from the original choice of using alphanumeric characters, instead of bit-mapped fonts, mainly as far as correct positioning and variety of styles are concerned. Alphanumeric characters were chosen because they could be displayed at higher speed by the old Intel processors, without 80x87 chips.

## 5. COMMANDS

A complete description of Dis commands follows, but before that, three general statements :

- 1) Parameters may be separated by either comma or *one* space.
- 2) The symbol `%` in first column acts as a Comment, i.e. the command parser does not interpret what follows.
- 3) A command line is analysed by the parser up to column 40. The remaining columns can be used for comments.



Format: ! n

Parameters: command ordinal

Function: Executes the command # n in the command buffer

Bugs: none, but note the blank space between ! and n

**ADD**

**ADD**

Format: ADD p1

Parameters: P1, constant to be added

Function: Adds p1 to the image matrix, within the last defined box.

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE,  
INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT

**ALPHA**

**ALPHA**

Format: ALPHA

Parameters: none

Function: Sets video to alphanumeric mode. As MODE 3, but doesn't enter in  
the command buffer.

Bugs: none

**BCOLOR**

**BCOLOR**

Format: BCOLOR ncol

Parameters: ncol, color code (0-15)

Function: Fills area with color ncol. This command works only with 640x480x16  
VGA standard mode (18 dec). For any other VGA mode, the screen is  
set to black.

Bugs: none

**BOX**

**BOX**

Format: BOX p1 or BOX ?

Parameters: p1 is the number of a pre-defined box. ? asks for actual box #.

Function: Makes available the area number n (n=1-12). Next commands like  
MODIFY or INTERP refer to this area.

Bugs: none

See also: DBOX, GBOX, LBOX, VBOX

**CHANGE**

**CHANGE**

Format: CHANGE P1

Parameters: P1, constant

Function: Changes to p1 the pixel whose intensity is between two levels, extrema included. The operation holds within the last defined box. Levels l1,l2 are asked for; if l1+l2=0, the whole box is set to const.

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE, INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES,SUBTRACT

<b>CLA</b>	<b>CLA</b>
------------	------------

Format: CLA

Parameters: none

Function: In a graphical mode, clears an area defined by cursor

Bugs: none, but it is very slow. Use CLA only for small areas.

<b>CLG</b>	<b>CLG</b>
------------	------------

Format: CLG

Parameters: none

Function: Clears actual graphic screen. Like ERASE, but not included in command buffer

Bugs: none

<b>CLS</b>	<b>CLS</b>
------------	------------

Format: CLS

Parameters: none

Function: Clears actual alphanumeric screen.

Bugs: none, but be sure to be in alpha mode, otherwise screen is cleared to white

<b>CLUT</b>	<b>CLUT</b>
-------------	-------------

Format: CLUT n

Parameters: color set n (n=1,2,3,4)

Function: Loads, on color registers of VGA card the R,G,B codes for 256 colors. Of course, colors immediately appear on the screen. Look Up Tables (LUT) actually available are: 1- Gray levels; 2- Rainbow; 3- Sixteen (16x16); 4- Vax (random). Default 16 and 256 colors LUTs are not directly accessed by this command: use MODE n, LUT 0 (or 1) and TV commands to redraw image with a default lut. CLUT command should be given before XLUT, to circularly rotate actual LUT.

Bugs: none

See also: LUT, XLUT

**CLW**

**CLW**

Format: CLW

Parameters: none

Function: Clears the last 5 lines of any graphic mode. These lines are the dialog window.

Bugs: none

**CONTOUR**

**CONTOUR**

Format: CONTOUR nlev or CONTOUR

Parameters: number of levels to be drawn. If nlev is 0, 1 or doesn't appear, levels must be supplied by user, following the dialog. Maximum value of nlev is 50.

Function: Computes and draws isophotes. User can select the area to be analysed and the area where isophotes will be traced (they may be the same area).

Bugs: none

**COSMIC**

**COSMIC**

Format: COSMIC P1

Parameters: P1, constant

Function: Selects and averages pixels whose value is P1 times the mean of the 8 pixels around. Suitable values for P1 may be 1.8-2.0. Like SPIKES, but slower because it computes spikes coordinates. It uses the same algorithm as the external program COSMIC.EXE, but does not write output files.

Bugs: none

See also: SPIKES

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE, INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT

**CSUB**

**CSUB**

Format: CSUB c1 c2

Parameters: c1= column to be changed; c2= column which substitutes c1.

Function: Substitutes column c1 with intensity values of column c2. Such substitution changes the image matrix in memory. To save on disk the modified image, use WDIRECT or WFITS commands.

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE, INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT

**CURSOR**

**CURSOR**

Format: CURSOR or CURSOR Y

Parameters: none or flag

Function: Sets graphic cursor to ON. Use arrows to move, also along diagonals. Hit numbers (on the keyboard first line) to change speed. For max speed (20) hit ' (ASCII 39D). X and Y position (pixels) and the corresponding pixel value (intensity) continuously appear on the bottom right corner of the screen. After cursor is set, you can hit - (ASCII 45D) to change actual pixel value. Hit ESC to exit from the routine. If flag Y is given, you can move the cursor and then hit ' (ASCII 96D) to write row, column and intensity into a disk file whose name is asked for. The first line of this file is the image file name.

Bugs: none, but no control is made on actual screen physical limits; cursor can disappear on the right side and reappear on the left one.

**DBOX**

**DBOX**

Format: DBOX p1 p2 p3 p4 p5 or DBOX p1

Parameters: p1,p2,p3,p4,p5 are box #; first and last row; first and last column

Function: Defines a sub-area (box) from the whole original frame. Any sub-area is characterized by a number (1-12 max) and four coordinates. If only area number is given, limits of the area are selected by cursor. RDIRECT and RFITS commands initialize box #1 to actual image limits.

Bugs: none

See also: BOX, GBOX, LBOX, VBOX

**DELETE**

**DELETE**

Format: DELETE L1,L2 or L1

Parameters: L1, L2 first and last line, or the only line to delete

Function: Deletes from the command buffer the lines from L1 to L2 (extrema included) or the line L1 and re-organizes the command buffer.

Bugs: none

**DIVIDE**

**DIVIDE**

Format: DIVIDE p1

Parameters: P1, constant image is divided by

Function: Divides the image matrix by p1, within the last defined box

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE,  
INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT

**DOS**

**DOS**

Format: DOS

Parameters: none

Function: Opens a child process which allows one DOS commands a time. If the first DOS command is the word COMMAND, several commands can be given; EXIT allows to leave child process and return to Dis.

Bugs: none, but a child process requires enough available memory to run a copy of COMMAND.COM.

**EDIT**

**EDIT**

Format: EDIT Line-number

Parameters: line-number; the line number, in the command buffer, to be edited

Function: Edit a line. Very, very raw editor. Use right arrow to set the cursor over the character to be changed. Space bar writes blanks (ascii 32). Use left arrow to go backward. Enter terminates the editing.

Bugs: left arrow doesn't work. In practice, carefully avoid the use of this key. Re-use edit if you are wrong.

**END**

**END**

Format: END

Parameters: none

Function: Exit to DOS. End does not save the actual command buffer (use WRITE for that).

Bugs: none

**ERASE**

**ERASE**

Format: ERASE

Parameters: none

Function: Clears the actual graphic screen.

Bugs: none

**FITSH**

**FITSH**

Format: FITSH

Parameters: none

Function: shows the first 36 records of FITS header. To see the next record hit  
< *RET* >; to exit hit X (or x) < *RET* >.

Bugs: none

**GBOX**

**GBOX**

Format: GBOX p1 or GBOX ?

Parameters: p1 is the number of a pre-defined box. ? asks for actual box

Function: Makes available to plots the area number n (n=1-12). Next command  
SCAN or OSCAN refers to this area.

Bugs: none

See also: BOX, DBOX, LBOX, VBOX, SCAN, OSCAN

**GRAPH**

**GRAPH**

Format: GRAPH

Parameters: none

Function: Sets video to actual graphic mode. As MODE n, but doesn't enter in  
the command buffer. It should be used to re-enter graphic mode after  
LIST command.

Bugs: none

**HARDCOPY**

**HARDCOPY**

Format: HARDCOPY

Parameters: none

Function: sends video memory to an Epson MX-80 printer. Uses low density, high  
speed codes so that print is quick. For more accurate reproductions use  
the GRAPHICS DOS 4.XX/5.XX command, or specialized software  
like PIZAZZ PLUS (R) or HPSCREEN (R).

Bugs: the routine doesn't handle gray levels, so it is valid only for plots, not for  
pictures.

**HEADER**

**HEADER**

Format: HEADER

Parameters: none

Function: Writes on the right side of the screen some data of the image (file name,  
dimension, zoom, max and min intensity). See also NOHEADER.

Bugs: none

**HELP****HELP**

Format: (1) HELP or (2) HELP command-name

Parameters: none or the name of a command for detailed help.

Function: (1) lists all commands available in Dis; (2) gives the help of the specific command. Use the COMPLETE name of the command, no abbreviations.

Example: HELP ALP is wrong (Help routine cannot find help file); HELP ALPHA is correct

Bugs: none

**IDENT****IDENT**

Format: IDENT or IDENT cc

Parameters: none or a 2-characters identification

Function: writes, at the left bottom of the image, actual date and time. Please note that date format is *dd-mm-yyyy*. If the two characters 'cc' are given, they are added to the above string.

Bugs: none

**INFO****INFO**

Format: INFO

Parameters: none

Function: Displays general information on Dis (author address, references).

Bugs: none

**INSERT****INSERT**

Format: INSERT line-number

Parameters: line-number, the ordinal which identifies the command in the command buffer.

Function: Allows insertion of new command(s) AFTER the command defined by line-number. INSERT prompt is I. END terminates insertion.

Bugs: none

**INTERP****INTERP**

Format: INTERP p1,p2

Parameters: p1, p2; x and y number of pixels.

Function: computes linear least-squares interpolation along rows or columns (or both in two steps) within the last defined box. Parameters give how

many pixels must be used as interpolation basis. Interp does not perform interpolation for zero-valued parameters. This command **MODIFIES THE MATRIX IN MEMORY**.

Bugs: none

To change data in memory use: **ADD, CHANGE, COSMIC, CSUB, DIVIDE, INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT**

<b>LABEL</b>	<b>LABEL</b>
--------------	--------------

Format: LABEL string

Parameters: string, a text of 40 characters max.

Function: Writes string (in alphanumeric mode) at the last-defined position of the graphic cursor. See also RELOCATE.

Bugs: Due to the transformation in row and column of the cursor position (in pixels), positioning may be difficult. Try RELOCATE with various coordinates to find out the best for you. Strictly speaking, this is not a bug, but depends on the author's choice to use alpha- characters in order to improve drawing speed.

<b>LBOX</b>	<b>LBOX</b>
-------------	-------------

Format: LBOX or LBOX p1

Parameters: p1 is box number in the direct access file boxes.lis

Function: lists in the dialog window number and limits of the actual 12 boxes or show the same for box # p1.

Bugs: none

See also: BOX, DBOX, GBOX, VBOX

<b>LEVELS</b>	<b>LEVELS</b>
---------------	---------------

Format: LEVELS nlev Xcenter Ycenter

Parameters: number of levels, row and column of the center. If LEVEL.DAT exists, nlev is overwritten by the number of levels found in such file. Maximum value of nlev is 50.

Function: Writes on the files ISO.NN, the coordinates of pixels whose intensity is within (extrema included) min and max given levels. Iso-levels are computed within the last-defined box. Any ISO.nn file contains in its first 4 lines: name of input image; tentative position angle of the isophotes (45 degrees), row and column of the centre, regularization parameter, useful for off-line fitting of the isophotes (fixed value= 0.05); minimum and maximum intensity the file refers to; one blank line. Row and column of the selected pixels follow.

Bugs: none, but be sure you are using the right box.

Examples:

DBOX 2 30 200 18 346  
 MAXMIN  
 BOX 2  
 LEVELS 15 85 163

!box levels must be computed in  
 !another box will be probably defined  
 !now box #2 is again the active one  
 !15 levels, center at 85,163, within box #2

**LIST****LIST**

Format: LIST

Parameters: none

Function: Lists on the screen the entire command buffer. Use Ctrl S or Pause key to stop the listing. List doesn't return to actual graphic mode. User needs give explicitly the MODE n (or GRAPH) command to re-enter graphics.

Bugs: none

**LOG****LOG**

Format: LOG p1 or LOG

Parameters: P1, constant to be multiplied or 1, if omitted

Function: Computes decimal logarithm of the image times p1 (e.g.  $-2.5 \cdot \log_{10}$ )  
 Multiply the results by 100 to avoid truncation of numbers between -1 and 1 due to integer\*2 data. Then:

$$\text{OUTPUT} = \text{INT}(P1 \cdot \text{LOG}_{10}(\text{INPUT}) \cdot 100 + 0.5)$$

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE, INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT

**LUT****LUT**

Format: LUT or LUT n or LUT ?

Parameters: none: displays lut; n: sets lut #n; ?: shows actual lut #

Function:

a) Displays actual color Look Up Table, starting from screen coordinates 1,322.

Lut can be cleared by CLA command.

b) Makes available LUT #n. Presently only 0 or 1 can be selected.

c) Shows actual LUT #

This command refers only to default LUTs. Use CLUT and XLUT for the other colors maps.

Bugs: Lut's fixed position may conflict with multi-image or zoomed displays.

See also: CLUT, XLUT

**MAXMIN****MAXMIN**

Format: MAXMIN or MAXMIN box

Parameters: none or box number

Function: Computes maximum, minimum and respective coordinates. Asks for area definition by cursor if no parameter is given, otherwise uses the area defined by the number after LBOX command.

Bugs: none

See also: DBOX, LBOX

Examples:

```

dbox 3 100 200 25 75    !define box #3
dbox 4 160 190 104 175 !define box #4
dbox 12 300 451 325 475 !define box #12
maxmin 4                !maxmin over area #4
maxmin                  !area will be defined by cursor

```

## METRIC

## METRIC

Format: METRIC scale or METRIC

Parameters: scale of actual image, in unit/pixel. If scale is not given, it will be set to 1 unit/px.

Function: Computes and shows distance between two points selected by cursor.

X, Y and diagonal distance are shown in both pixel and unit.

Bugs: none

## MODE

## MODE

Format: MODE n

Parameters: decimal value of the VGA emulation code.

Function: Sets VGA video card to one of the modes listed in the file MODI-VGA.PLT. This file can be edited in order to add your own graphic card or mode. The main VGA emulation modes are listed below:

Mode	Rows	Cols	Physical Window	Char.h.	Maxcol	Description
4	320	200	100 300 50 170	8	4	CGA
5	320	200	100 300 50 170	8	4	CGA
6	640	200	100 600 50 170	8	4	CGA
13	320	200	100 300 50 170	8	16	EGA
14	640	200	100 600 50 170	8	16	EGA
15	640	350	100 500 50 290	14	2	EGA MONO
16	640	350	100 600 50 290	14	16	EGA
17	640	480	100 600 130 430	16	2	VGA MONO
18	640	480	100 600 130 430	16	16	VGA default
19	320	200	100 300 50 170	8	256	VGA
64	640	400	100 600 100 350	16	2	OLIVETTI
41	800	600	100 500 100 500	8	16	VGA ET4000

42	640	400	100	500	100	350	16	256	VGA ET4000
46	640	480	100	500	100	430	16	256	VGA ET4000
55	1024	768	100	650	100	650	16	16	VGA ET4000
56	1024	768	100	650	100	650	16	256	VGA ET4000

---

Bugs: 800x600 modes fail in text positioning

**MULTIPLY**

**MULTIPLY**

Format: MULTIPLY p1

Parameters: P1, constant to be multiplied by the image

Function: multiply the image matrix by p1, within the last defined box.

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE,  
INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT

**NOHEADER**

**NOHEADER**

Format: NOHEADER

Parameters: none

Function: Clears (sets to black) the area used to show image info.

Bugs: none

See also: HEADER

**NOISE**

**NOISE**

Format: NOISE P1

Parameters: P1, constant

Function: edge enhancement and noise filtering by local average. A GAIN FACTOR is required. Use a value like 3 or 2. This gain is the ratio between the initial and final noise level. Run the external program NOISE.EXE to have more information and references.

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE,  
INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT

**OSCAN**

**OSCAN**

Format: OSCAN p1 p2 p3 or OSCAN

Parameters: flags which control [p1] scan direction (row, column or diagonal if p1=0; through a rectangular window if p1=1), [p2] writing data of the scan on disk file (p2=0=no; p2=1=yes); [p3] plotting of scan data

(p3=0=no; p3=1=yes). No parameter is the same as p1=0, p2=0, p3=1 (i.e. 1-pixel-high scan, no file written, plot of data).

Function: produces one-pixel-high (line) or slit scan of the image, can write disk files with and/or show OVER-plots of scan data. Plot limits can be defined only interactively, by the cursor. Screen area that will contain the graphs is defined in advance by the command GBOX n (no default exists). User coordinates can be extracted by cursor, by hitting X (or x) after the end of plot. Dialog asks for drawing color. The last defined color is used also to draw the box. Depending on parameters p1 and p2, this command outputs the disk files RECT.OUT, LINEX.OUT, LINEY.OUT, LINEXY.OUT. OVERSCAN leaves the last defined x and y scales unchanged, so that different plots can be drawn on the same scale. This command must be used AFTER at least one SCAN command.

Bugs: none

See also: SCAN, GBOX

**PAUSE**

**PAUSE**

Format: PAUSE

Parameters: none

Function: Stops the execution of the next command until a key is pressed. On the dialog window "*DIS paused: hit a key to continue*" appears, and a beep outlines the presence of this command.

Bugs: none

**PCOM**

**PCOM**

Format: PCOM

Parameters: none

Function: displays the values of the variables in both plot and Dis libraries. This command has been written for author's use, so not all may be clear; nevertheless PCOM can be useful in a lot of cases.

Bugs: none

**PLAYBACK**

**PLAYBACK**

Format: PLAYBACK

Parameters: none

Function: Executes the command buffer from the first to the last command. No jump is allowed.

Bugs: none

**POSITION****POSITION**

Format: POSITION P1 P2 or POSITION

Parameters: P1 x-position, P2 y-position of the upper left corner of image.

Function: Defines the upper left corner of the image display in screen coordinates. If no parameter is present, position will be given by cursor. The variables XPOS and YPOS in common block are modified.

Bugs: none

**QUIT****QUIT**

Format: QUIT

Parameters: none

Function: Exit from Dis to DOS. Does not save the actual command buffer (use WRITE for that). This command is the same as END

Bugs: none

**RDIRECT****RDIRECT**

Format: RDIRECT file name or RDIRECT

Parameters: file name, 40 characters max string containing the [disk:path]file-name.ext of image file.

Function: Reads an integer\*2 (i.e. 2-bytes) direct access, no header file containing the image. If file name is omitted, this command asks for it as *File:*. Due to the lack of the header, the user MUST know the # of data-per-row NC. This number is normally written as file name extension by DIS output routines. If ext is CCD or NTT then NC is assumed to be 320 (RDIRECT shows this number as default). Box #1 in the file *boxes.lis* is initialized to start-row, end-row, start-column, end-column.

Bugs: none

**READ****READ**

Format: READ file-name

Parameters: file-name, 40 characters max string containing a list of Dis commands.

Function: Reads a list of commands from a disk file and initializes the command buffer.

Bugs: none

**RELOCATE****RELOCATE**

Format: RELOCATE row column

Parameters: x and y screen coordinates

Function: re-defines the current point position.

Bugs: none

**RESET**

**RESET**

Format: RESET

Parameters: none

Function: Resets Dis graphics to default parameters, as listed in MODIVGA.PLT and defined by CONFIG.PLT. Clears the screen. This command is added to the command buffer.

Bugs: none

**RFITS**

**RFITS**

Format: RFITS file name or RFITS

Parameters: file name, 40 characters max string containing the [disk:path]file-name.ext of image file.

Function: Reads an integer\*2 or integer\*4 (i.e. 2 or 4-bytes) FITS file containing the image. If file name is omitted, this commans asks for it as *File:*. Box #1 in the file *boxes.lis* is initialized to start-row, end-row, start-column, end-column.

Bugs: none, but intepretation of fits record is a slow process here.

**RSUB**

**RSUB**

Format: RSUB r1 r2

Parameters: r1= row to be changed; r2= row which substitutes r1.

Function: Substitutes row r1 with intensity values of row r2. Such substitution changes the image matrix in memory. To save the modified matrix use WDIRECT command.

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE, INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES,SUBTRACT

**SCAN**

**SCAN**

Format: SCAN p1 p2 p3 or SCAN

Parameters: flags which control [p1] scan direction (row, column or diagonal if p1=0; through a rectangular window if p1=1), [p2] writing data of the scan on disk file (p2=0=no; p2=1=yes); [p3] plotting of scan data (p3=0=no; p3=1=yes). No parameters is the same as p1=0, p2=0, p3=1 (i.e. 1-px-high scan, no file written, plot of data).

Function: produces one-pixel-high (line) or slit scan of the image, can write disk files with and/or show plots of scan data. Plot limits can be defined only interactively, by the cursor. Screen area that will contain the graphs is defined in advance by the command GBOX n (no default exists). User coordinates can be extracted by cursor, by hitting X (or x) after the end of plot. Dialog asks for drawing color. The last defined color is used also to draw the box. Depending on parameters p1 and p2, this command outputs the disk files RECT.OUT, LINEX.OUT, LINEY.OUT, LINEXY.OUT.

Bugs: none

See also: OSCAN, GBOX

## SPIKES

## SPIKES

Format: SPIKES

Parameters: none

Function: Reads the coordinates of one-pixel spikes which deviate more than XM times from the average intensity of its (3x3) neighbourings, computes again the mean values and changes any central pixel to average. Spikes positions and constant XM have been formerly recorded by means of the external program COSMIC.EXE or by the command CURSOR Y and saved into the disk files COSMIC.OUT or CURSOR.OUT, respectively. Anyway, a file name is asked for.

Bugs: none

See also: COSMIC

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE, INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES, SUBTRACT

## SQUARES

## SQUARES

Format: SQUARES cr,cc,ndata,istep or SQUARES cr,cc,ndata

Parameters: cr,cc central row and column; ndata # of squares involved (max 100); istep growing step.

Function: Computes the total intensity within *istep - pixels* increasing squares. Produces the output file DIS.Q containing a comment, ndata, calibration constant (=0), sky background (=0), regularization parameter(=0.002), istep. This file is the input for the external program BECCD which computes brightness profile of the (extended) object centered at cr,cc.

Bugs: none. Be sure BECCD.EXE is available, otherwise this is a useless command

## STAT

## STAT

Format: STAT n or STAT

Parameters: a box number as listed after LBOX command. Also no parameter  
 Function: computes and shows max,min,total intensity, average, standard deviation of data within a pre-defined box or an area defined interactively by cursor.

Bugs:none

See also: DBOX, LBOX

Examples:

```

dbox 3 100 200 25 75      !define box #3
dbox 4 160 190 104 175   !define box #4
dbox 12 300 451 325 475  !define box #12
stat 4                    !statistics over area #4
stat                      !area will be defined by cursor

```

## SUBTRACT

## SUBTRACT

Format: SUBTRACT p1

Parameters: P1, constant to be subtracted

Function: Subtracts p1 to the image matrix, within the last defined box

Bugs: none

To change data in memory use: ADD, CHANGE, COSMIC, CSUB, DIVIDE, INTERP, LOG, MULTIPLY, NOISE, RSUB, SPIKES,SUBTRACT

## TEXT

## TEXT

Format: TEXT

Parameters: none

Function: Writes a string (in alphanumeric mode) at cursor position. Dialog asks for input string and cursor positioning. At 'Enter string:', hit <RET> to exit from TEXT.

Bugs: Due to the trasformation in row and column of the cursor position (in pixels), positioning may be difficult. Try with different coordinates and screen resolution to find out the best for you. Strictly speaking, this is not a bug, but depends on the author's choice to use alpha characters in order to improve drawing speed. See also LABEL.

## TV

## TV

Format: TV or TV min max

Parameters: none or minimum, maximum intensity

Function: Displays image on the screen, following the actual video mode. If no parameter is given, default (or last defined) minimum and maximum intensities within which color codes are spread will be assumed. Image position and screen erasure are defined by commands POSITION and CLG or ERASE (see).

Bugs: none

**VBOX**

**VBOX**

Format: VBOX n or VBOX

Parameters: n is the number of a pre-defined box, as appears after LBOX command.

Function: Defines a (sub)-area from the whole original frame. Next display operations like TV refers to this 'vision' box. After image loading, it is defaulted to image dimension (i.e. box 1). If no parameters are given, limits of the area will be selected by cursor.

Bugs: none

See also: BOX, DBOX, GBOX, LBOX

Examples:

```

dbox 2 120 220 100 180      ! now box 2 is defined and active
box 1                       ! now box 1 is active
vbox 2                      ! vision box like box 2
mode 46                     ! 640x480x256 colors
tv 0 1280                   ! display within box 2; cuts 0 - 1280

vbox                        ! use cursor to define vision box
                           BOX: left up corner then <ESC>
```

**WDIRECT**

**WDIRECT**

Format: WDIRECT file name

Parameters: file name, 40 characters max string containing the [disk:path]filename of image file. DON'T USE EXTENSION.

Function: Writes an integer\*2 (2-bytes) direct access, no header file containing the image. Due to the lack of the header, number of data per row is computed and used as file name extension. It will be used by RDIRECT. The area to be written depends on the last defined BOX or is the whole image.

Bugs: none

**WFITS**

**WFITS**

Format: WFITS file name

Parameters: file name, 40 characters max string containing the [disk:path]filename of image file. DON'T USE EXTENSION; it will be set to .FTS by the program.

Function: Writes a 16 bit/pixel FITS file containing the image. The area to be written depends on the last defined BOX or is the whole image.

Bugs: none

**WPIXEL****WPIXEL**

Format: WPIXEL p1 p2 p3 p4

Parameters: first and last line, first and last column

Function: shows numerical values of actual image data within the limits defined by parameters, extrema included.

Bugs: none

**WRITE****WRITE**

Format: WRITE file-name

Parameters: [*path*]output file name

Function: Saves on a disk file the actual command buffer. If file-name exists, it will be overwritten.

Bugs: none

**XLUT****XLUT**

Format: XLUT

Parameters: none

Function: Circularly rotate the starting color of actual LUT, by means of keypad left and right arrows. Hit *ESC* to exit from this command. A CLUT n command should have been given before XLUT. Follow the same rules used for CURSOR: numerical keys set the jump along color scale. Vertical position of the cursor is set to 100 by default and can be changed by vertical arrows.

Bugs: none

See also: LUT, CLUT

**ZOOM****ZOOM**

Format: ZOOM n

Parameters: zoom value

Function: Sets zoom value to n. Maximum zoom value is undefined.

Bugs: none

## EXTERNAL PROGRAMS

Dis includes some external programs which execute less used tasks or too much computer resources requiring algorithms. Some of them are also embedded in Dis, also if are slightly different in both form or output products. Any of external programs get a Dis-produced input or outputs Dis-suitable data and images. Actually available programs are:

BECCD.EXE - Radial brightness profile with constant axial ratio

CORONA.EXE - Smaller image centered in a larger one and zero-filling of the area around it

COSMIC.EXE - One-pixel spikes detection and averaging

EXTRACT.EXE - Extraction of a smaller image from a larger one

IMACOL.EXE - Color index image in magnitudes, from two given images

MATOP.EXE - Add or subtract two images and transpose a square image

MOVEM.EXE - Translation of an image. Integer steps

NOISE.EXE - Edge enhancement and noise filtering at given gain

PULCCD.EXE - Bias and Flat Field correction

**BECCD**

The radial brightness profile of an (extended) object is computed by solving an integral equation. The small point-by-point difference between the intensity distribution integrated over a square domain or over a circular one with the same area, allows to express the integrated luminosity  $L(x_k)$  of a CCD image over the square with  $x_k = (2k - 1)p$  as

$$L(x_k) \simeq L(r_k) = \int_0^{r_k} 2\pi r I(r) dr + \pi r_k^2 b, \quad (1)$$

where  $p$  is the pixel size,  $I(r)$  the radial intensity,  $r_k = (2k - 1)p/\sqrt{\pi}$  and  $b$  a constant sky background ( see Bendinelli and Gatti, 1991). When the brightness profile of a star or an extended object is needed, the integral equation (1) must be solved for  $I(r)$ , getting the luminosity  $L(x_k)$  within squares of growing side.  $L(x_k)$  is the output of the Dis command SQUARES (i.e. the file DIS.Q), so a constrained mean squares (in Tikhonov sense; see e.g. Tikhonov and Arsenine, 1977) solution of Eq.(1) can be obtained by BECCD. Constrained mean square solution requires a numerical parameter ( the regularization parameter  $\gamma$ ) which dumps the numerical instabilities due to *ill-posed* nature of Eq.(1). Regularization introduces a *bias* which adds to variance to form the final error of the solution (see Marquardt, 1970)

Suitable values of  $\gamma$  should be set by trial and errors, looking for the minimum residual of the first 2 or 3 data. A possible starting value for  $\gamma$  may be about 1.0E-11.

BECCD outputs the file BE.OUT, without control on the existence of a formerly created file with the same name; user should provide to change file name in order to avoid overwriting.

An example of BE.OUT follows:

```

Comment is taken from DIS.Q
G58 DEC LUCY 35IT
these 7 data are the input parameters
Calibration Const.= .000 zero point for magnitudes scale (optional)
Background/px... = .000 !sky background
Kmax..... = 60 !dimension of data sample
Axial ratio..... = 1.000 !constant axial ratio of the image
Scale ("/px).....= .022 !this value refers to HST FOC f/96
Gam.....= .1000E-10 !regularization parameter  $\gamma$ 
Istep.....= 2 !step of squares growth, in pixels

```

**Output parameters****Statistical Info on Output**

```

Sum of Squared Residuals= .64227E+03
Stand. Dev. of Residuals= .32718E+01
Gamma [gam*w(1)].....= .10730E-13 ! w(1) = 1/ $\sqrt{\text{data}(1)}$ 
Background/arcsec2..... = .00000E+00
Chi Square = .16884E-01
Bias..... = .94192E-12 ! Bias due to regularization

```

Variance.. = .78247E+10

Two tables follow: the first one has 8 columns which contain

- (1) Ordinal of data (I)
- (2) Mean radius in arcsec (RM)
- (3)  $L(x_k)$ , the integrated intensity over squares (OBS)
- (4)  $L(x_k)$ , as computed after Eq.(1) has been solved (COMP). It should be as similar as possible to OBS
- (5) Integrated luminosity  $L(R)=2\pi \int RMMU(r)dr$
- (6) Brightness profile MU in *intensity/(arcsec)<sup>2</sup>*
- (7) Like column (6) but corrected for sky background (MUC)
- (8) Standard deviation (SIGMA) of MUC. Given by both Bias and Variance.

The second table has 4 columns:

- (1) Ordinal of data (I)
- (2) Mean radius in arcsec (RM)
- (3) Brightness profile in magnitudes per square arcsec, normalized to the central value (BLOG).
- (4) Standard deviation (ELOG) of BLOG.

## CORONA

From an input, smaller image, this code generates a larger one, the original image being centered on it. The area around the smaller image is set to zero. The maximum allowed dimensions of input and output images are, respectively, (128x128) and (400x300). CORONA reads Fits files (whose extension is .FITS) and Direct Access (DA) files (with any other extension). On output, only DA files can be written.

At run, CORONA asks for input file name (small matrix), output file name and its dimensions (rows and columns).

## COSMIC

Search of 'cosmic rays', referred as pixels whose intensity is greater than the average flux of its neighbourings, times a given constant (XM); so, cosmic rays are selected if  $\text{Intensity} > \text{Ave}(3 \times 3 \text{ flux}) * \text{XM}$ . Events which agree with the above criterium, are registered on a disk file as position (row and column), intensity and average flux of the 3x3 box around.

Default output file name is COSMIC.OUT, but its existence is checked at any run, and user asked to append new data to the old file or give new file name. At run COSMIC asks for XM and input file name.

As input images, COSMIC can read fits or Direct Access (DA) files. The output file will be used, without changes, by the Dis command SPIKES, which substitutes the average flux to 'spikes'.

This external program has a correspondence in the Dis command COSMIC, which performs in-place search and substitution of the spikes, without saving their position and intensity. The command COSMIC should be used for quick look and to choose a proper value of XM.

## EXTRACT

An image with max dimensions 512x320 is extracted from a larger frame. Input and output data may be in both Direct Access (DA) or Fits format. The last format writes 16 bits-per-pixel integer data, with byte swapping.

At run, EXTRACT asks for input file name, shows input rows and columns and bit/pixel; then waits for start row, start column, number of rows, number of columns of the output image (SR,SC,NR,NC). The output file name is then asked for, followed by the format (DA or Fits) request. If the choice is DA the file name extension will be 3-digits number of data-per-row; Fits files will have .FITS extension.

## IMACOL

Computes color index from two (512x320 max) CCD images. Output colors are in magnitudes and multiplied by 100, in order to partially avoid 16-bits integer truncation. Given the input images (shifted, if necessary) IM1 and IM2, the output image COLOR will be:

$$COLOR = INT(2.5 * LOG10(IM1/IM2) * 100)$$

If the actual pixel of IM1 or IM2 is less than or equal to *zero*, COLOR is set to *zero*. At run, IMACOL asks for the number of rows and columns, common to both input and output images and then the names of all files involved. The output file is written as two-bytes integers in direct access, no header format. Output file name extension will be the number of data-per-row. IMACOL reads only I\*2, direct access input files.

## MATOP

Add or subtract two matrices and transpose a matrix. Only integer\*2 operations.

At any step the same row of both matrices is read, operation computed and resulting row written on the output file. This procedure implies a lot of work for hard disks heads, so, if possible, use a RAM disk, where both images have been copied in, as working area. At run, MATOP asks for input file names (only one name if a matrix must be transposed), dimensions, operation (+, -, T) and output file name (give also extension). All frames are read and written in Integer\*2, direct-access, no header format.

## MOVEM

MOVE *Matrix* shifts an image along rows, columns or both at integer steps without rebinning and/or rotation. At run, after the input file name, signed shifts, in pixels, are asked for. New areas derived from shifts are filled with zeroes. The Direct Access (DA) output file will be named MOVED.xxx, where xxx is the number-of-data-per-row (NX). An output row, used as control, shows on the screen NX, the bytes per row, start and end row and column of the output image (I1,I2,J1,J2).

*Example:* two frames of, say, NGC 1068 in B and R colors are available and a color index image must be computed by means of IMACOL. The objects in the frames should have the same coordinates, of course. MAXMIN command gives 267,188 and 268,184 for central row and column of B and R frames respectively. Superposition of, e.g., B over R frame means that the row of the origin of B frame must be diminished of 1 pixel while the column must be increased of 4 pixel. So, the shifts will be -1, +4.

## NOISE

The local average is used to filter noise and enhance contrast at a given level of *gain*, following the formula

$$OUTPUT = INT(\langle INPUT \rangle + \frac{INPUT - \langle INPUT \rangle}{GAIN} + 0.5), \quad (3)$$

where  $\langle INPUT \rangle$  is the mean value of a 3x3 box centered on the pixel to be filtered. If GAIN is zero, OUTPUT is set to  $INT(\langle INPUT \rangle + 0.5)$ . The basic assumption for this algorithm is that the sample mean and variance of a pixel is equal to the local mean and variance of all pixels within a fixed range surrounding it (Lee, 1980).

NOISE reads Fits and Direct Access (DA) files. At run input file name and gain are asked for. Filtered image is written on the output DA file OUT.NNN, where NNN is the number of data per row.

The Dis command NOISE does the same of NOISE.EXE.

**PULCCD**

Correction of 512x320 CCD frames from flat-field and bias. Output image is:

$$OUTPUT = INT\left(\frac{INPUT - BIAS}{FLAT - BIAS} * \langle FF \rangle + 0.5\right), \quad (2)$$

where  $\langle FF \rangle$  is the mean value of the flat-field image.

At any step the same row of data, bias and flat field frames is read, Eq. (2) computed and the resulting row written on the output file. This procedure implies a lot of work for hard disks heads, so, if possible, use a RAM disk, where all images have been copied in, as working area.

At run, PULCCD asks for data, bias, flat field and output file names and for the mean value  $\langle FF \rangle$  of the flat field frame. This value can be derived by DIS command STAT 1.

All frames are read and written in Integer\*2, direct-access, no header format.

## Appendix A. List of DIS commands

Command	Params	Description
!	1	re-execute command <i>#n</i> . Add it to buffer
ADD	1	add <i>constant</i> to matrix within box
ALPHA	0	enter alphanumeric mode. Like MODE 3. Not in buffer
BCOLOR	1	in standard VGA clear screen to color <i>n</i> . Otherwise CLG.
BOX	1	make available the area <i># n</i>
CHANGE	3	change pixel values to a constant, within given limits
CLA	0	clear area selected by cursor. Graphic
CLG	0	clear graphic screen. Set to color <i>#0</i>
CLS	0	clear alphanumeric screen
CLUT	1	change external luts: available lut <i>codes</i> are (1,2,3,4)
CLW	0	clear dialog window (last bottom 5 lines)
CONTOUR	1	compute & draw a given <i>number</i> of isophotes in the last def. box
COSMIC	1	select pixels with intensity larger than <i>c1* ave</i>
CSUB	2	substitution of column <i>c1</i> with data of column <i>c2</i>
CURSOR	9	set cursor on. ESC to set it off. <i>Yes</i> writes coords
DBOX	5	define a picture box <i>#</i> by <i>corners coordinates</i> . Write boxes.lis
DELETE	2	delete rows <i>n1</i> to <i>n2</i> (extrema included) in command buffer
DIVIDE	1	divide matrix by <i>constant</i>
DOS	0	open a DOS child process (spawn). Need enough memory
EDIT	1	edit a <i>line</i> in the command buffer
END	0	end of insertion or exit to DOS
ERASE	0	erase graphic screen
FITSH	0	show first 32 lines of FITS header
GBOX	1	make available the area <i># n</i> to graphics
GRAPH	0	enter actual graphic mode. Like MODE n, but not in c. buffer
HARDCOPY	0	send video to Epson printer. Very dirty routine.
HEADER	0	show some info about the image
HELP	9	display list of commands or a command help <i>file</i>
IDENT	9	write <i>date</i> and <i>user code</i> (2 chars) if any
INFO	0	show some info about Dis
INSERT	1	allow insertion after the row <i>#n</i> in command buffer
INTERP	2	linear interpolation for <i>rows</i> and <i>columns</i>
LABEL	9	write <i>text</i> at cursor position
LBOX	1	display all boxes or <i>given box #</i> coordinates
LEVELS	3	write on disk file row & column of <i>n</i> levels centered at <i>XC, YC</i>
LIST	0	list all commands in the buffer
LOG	1	decimal log(image) * <i>const</i> * 100
LUT	1	show LUT <i>n = 0,1</i> or write LUT <i>#</i>
MAXMIN	1	maximum and minimum within <i>box #n</i> or cursor defined
METRIC	1	give measures on image if <i>scale</i> $\neq 1$
MODE	1	select graphic mode <i>#n</i> (decimal)
MULTIPLY	1	multiply matrix by <i>constant</i>
NOHEADER	0	clear image header
NOISE	1	noise filtering with given <i>gain</i>
OSCAN	3	draw data in existing frame and scale. See SCAN
PAUSE	0	pause execution. Hit a key to restart
PCOM	0	print variables in common blocks

PLAYBACK	0	execute a command list
POSITION	2	define image position $X$ , $Y$
QUIT	0	exit to DOS ( like END)
RDIRECT	9	open and read a direct access image <i>file</i>
READ	9	open and read disk <i>file</i> with command list
RELOCATE	2	locate the current point at $X$ , $Y$ (screen coordinates)
RESET	0	reset all as defined in the file MODIVGA.PLT
RFITS	9	read image <i>file</i> in FITS format
RSUB	2	substitution of row $r1$ with data in row $r2$
SCAN	3	plot data within the last defined GBOX
SPIKES	0	average spikes. Read output of COSMIC.EXE
SQUARES	4	compute & write on file integrated flux within growing squares
STAT	1	max, min, average, $\sigma$ , total flux in a given <i>box</i>
SUBTRACT	1	subtract <i>constant</i> from matrix
TEXT	9	write <i>string</i> at cursor position
TV	2	display image according to MODE, within <i>min</i> , <i>max</i>
VBOX	1	define box $\#n$ as image display box
WDIRECT	9	write <i>file</i> on disk in DA format
WFITS	9	write <i>file</i> on disk in FITS format
WPIXEL	9	show values within $xstart$ , $xend$ , $ystart$ , $yend$
WRITE	9	write on disk <i>file</i> , actual command buffer
XLUT	0	shift starting color register of actual lut (rotate lut)
ZOOM	1	set zoom value to $n$

---

**Appendix B.** List of files in DIS installation diskette

---

File	Description
INSTALL.BAT	Installation batch file
DIS.EXE	Main program
BECCD.EXE	Brightness profile
CORONA.EXE	shell of zeroes around a matrix
COSMIC.EXE	identifies one-pixel wide spikes
EXTRACT.EXE	extract a 512x320 max matrix from a larger one
IMACOL.EXE	color image from two monochrome images
MATOP.EXE	Integer matrix operations
MOVEM.EXE	shift of an image (no rebin)
NOISE.EXE	noise filtering
PLINST.EXE	Writes installed Config.plt
PULCCD.EXE	Bias and flat field correction
DSKA.EXE	outputs DSKDIS.INI for disk A:
DSKB.EXE	outputs DSKDIS.INI for disk B:
DSKC.EXE	outputs DSKDIS.INI for disk C:
DSKD.EXE	outputs DSKDIS.INI for disk D:
DSKE.EXE	outputs DSKDIS.INI for disk E:
DSKF.EXE	outputs DSKDIS.INI for disk F:
CMDS.DIS	List of commands and parameters
BOXES.LIS	list of boxes coordinates
LEVEL.DAT	list of intensity levels for Contour
CONFOG.PLT	Starting config file, changed to Config.plt
MODIVGA.PLT	List of VGA graphic modes and behaviours
DISMAN.TEX	Tex version of dis user manual (this manual)
DISHLP	Directory with the <i>command.hlp</i> help files

---

**Appendix C.**

Examples of DIS command files

a)

```

% 640x480x256 colors display mode Tseng Lab VGA chip
MODE 46
% read a 16-bits integer, direct access, no header, image file
RDIRECT focpsf.172      (ALSO RD FOCPSF.172)
% FITS files can also be read
RFITS snr2m.fts        (ALSO RF SNR2M.FTS)
% image will be positioned at x=1 y=1 (default)
POSITION 1,1          (ALSO POS 1 1)
% set normal lut (default)
LUT 0
% spread colors between intensity 0 and 25000
TV 0 25000
%define box #2
DBOX 2 80 130 90 160
%activate original dimension of the image
BOX 1
% compute and draw 30 isophotes; follow dialog
CONTOUR 30
%subtract a constant
SUBTRACT 183.4
% if you want to subtract the same constant from box#2 only
BOX 2
SUBTRACT 183.4
% try new image position (x=180,y=1) with inverse color map
POS 180 1
% change lut: use inverse. These are native luts
LUT 1
% spread colors between 0 and 17000
TV 0 17000
% compute and draw 20 isophotes; follow dialog
CONTOUR 20
% profile along a 1-px high scan. Doesn't write data into a disk file
scan 0 0 1            (ALSO SCAN)
% as before, plotting with the same scale and
% over the above graph (overplot)
oscan 0 0 1          (ALSO OSCAN)
% save on fits file subf.fts the sub-frame within box #2
BOX 2
WFITS subf      dont'give extension!      (ALSO WF SUBF)
%...or save it on the direct access file subf.071
WDIRECT subf    don't give extension !     (ALSO WD SUBF)

```

b)

```

% 7 256x256 images of the same object (a globular cluster in M31 as
% observed by the Hubble Space Telescope)
% are shown on 1024x768x256 colors screen
mode 56 !hi-res mode on Tseng Lab.  chip ET4000 (decimal)
% this is the external lut #4 (Random)
CLUT 4
pos 1 1 !position of left top corner of the image
rf g305a.fts !reads fits file
tv 0 8000 !display intensities between 0 and 8000
RELOCATE 125 280 !position of cursor below the image
LABEL G305A !writes object name
%... repeat for any image ....
pos 258 1
rf g305b.fts
tv 0 8000
RELOCATE 385 280
LABEL G305B
%
pos 514 1
rf g305ab.fts
tv 0 8000
RELOCATE 640 280
LABEL G305AB
% pos 772 1
rf g305am.fts
tv 0 26000
RELOCATE 897 280
LABEL G305AM
%
pos 1 330
rf g305al.fts
tv 0 8000
RELOCATE 125 600
LABEL G305AL
%
pos 258 330
rf g305bl.fts
tv 0 8000
RELOCATE 385 600
LABEL G305BL
%
pos 514 330
rf g305abl.fts
tv 0 8000
RELOCATE 640 600
LABEL G305ABL

```

## REFERENCES

- Bendinelli, O. and Gatti, M.G., 1991. *Numerical and parametric PSF extraction from CCD frames: application to ground-based and HST observations*. Bologna Astrophysics Preprints, BAP 11-1991-039-DDA
- Lee, J.-S., 1980. *Digital Image Enhancement and Noise Filtering by Use of Local Statistics*, IEEE Trans. Patt. Analys.Mach.Int., **PAMI-2**, 165.
- Marquardt, D.W., 1970. *Generalized inverses, Ridge regression, Biased linear estimation, and Nonlinear estimation.*, Technometrics, **12**, no. 3, p.591.
- Rinaldi, W. and Zavatti, F., 1991. *PLOTPE e ASM PLOT*, Dipartimento di Astronomia, Bologna, *Internal Report*, November.
- Tikhonov, A. and Arsenine, V., 1974. *Méthodes de résolution de problèmes mal posé*, Editions MIR, Moscow.