BMON2 — BUFFER MEMORY MONITOR SYSTEM FOR INTERACTIVE IMAGE PROCESSING

NCI/IP Technical Report #21b

December 14, 1976 (TR21)
Revised June 27, 1977 (TR21a)
Revised March 17, 1978 (TR21b)

Peter Lemkin

National Cancer Program

U.S. Department of Health, Education, and Welfare / National Institutes of Health / National Cancer Institute

BMON2

BUFFER MEMORY MONITOR SYSTEM FOR INTERACTIVE IMAGE PROCESSING

NCI/IP Technical Report #21a

Peter Lemkin

Image Processing Unit
Division of Cancer Biology and Diagnosis
National Cancer Institute
National Institutes of Health
Bethesda, Md. 20014

December 14, 1976 (TR21)

Revised June 27, 1977 (TR21a)

Revised March 17, 1978 (TR21b)

Abstract

BMON2 is a PDP8e image processing system running on the Image Processing Units' Real Time Picture Processor (RTPP) used for biological image processing. The RTPP consists of a PDP8e computer, Quantimet 720 TV image analyzer, axiomat microscope with stage, focus and zoom stepping motor control, interactive control desk, GraphPen tablet, and image buffer memory. Control of the system is effected by interaction through the teletype or control desk. Teletype commands are entered via the OS8 command decoder (with the implication that BMON2 may be run under OS8 BATCH). About 100 operations are available. Results are converted to microns for user convenience. Teletype commands may be dynamically assigned to the 12 command keys on the control desk which may be saved and restored from disk files. This allows individually selected subsets of BMON2 commands to be called directly from the control desk.

T A B L E O F C O N T E N T S

1. INTRODUCTION

BMON2 is a PDP8e image processing system running on the Image Processing Units' Real Time Picture Processor (RTPP) used for biological image processing ([1], [2], [3], [4]).

Interactive image processing systems such as BMON2 and earlier ones such as TICAS [19], SCANIT [20], and SCANCANS [21] facilitate the design and implementation of experiments involving automated image analysis of biologic microscope images.

The RTPP consists of a PDP8e computer with four RK05 disks and two 9-track magtape units, Quantimet 720 TV image analyzer, axiomat microscope with stage, focus and zoom (0.8X:3.2X) stepping motor control, interactive control desk, GraphPen tablet, and image buffer memory. Control of the system is effected by interaction through the teletype or control desk. Teletype commands are entered via the OS8 command decoder (with the implication that BMON2 may be run under OS8 BATCH [8]). About 100 operations are available. Results are converted to microns for user convenience. Teletype commands may be dynamically assigned to the 12 command keys on the control desk which may be saved and restored from disk files. This allows individually selected subsets of BMON2 commands to be called directly from the control desk.

Eight image buffer memories (BM) each store two 8-bit 256x256 pixel images and may be accessed at video frame rates for image acquisition and display, or randomly accessed. The BMs may be posted on the Quantimet (QMT) display, unposted, loaded with QMT camera video images and moved around the QMT screen. Data in the memories may be randomly accessed by the PDP8e and processed with the resultant transformed images stored in the memories. I/O operations may be performed between the memories and the PDP8e's RK05 disk files or MAG10 [10] formatted 9-track magtape files.

Rapid feature extraction of objects segmentable by threshold may be performed using the QMT "function computer" hardware with a special data acquisition interface on the PDP8e to acquire density, area, perimeter and projections at TV frame rates. QMT video images are the result of either QMT camera video or of BM synthesized video inserted into the image in place of the QMT camera video.

The RTPP digitizes QMT camera video (720 line by 880 pixel non-interlaced frame) using an 8-bit 125 nanosecond A/D conversion. The digitized video is then multiplexed with digital video from the BMs. Camera video may be stored in the BMs at this point in one TV time frame. Alternatively, BM digital video may be output from the multiplexer instead of the digitized camera video. This digital output is then converted back to an analog video signal using a high speed D/A converter. In the above scheme, it is also possible to display

the output of either the QMT camera or a single 8-bit BM at any single pixel in the image. It is also possible however, as an option, to use the other half of the BM being displayed at a particular pixel as a mask. This is done by generating an effective QMT live frame from this mask BM using the sign bit of the data to denote a mask pixel. Only video within this mask is used for QMT measurements.

A GraphPen tablet is used to enter and edit (x,y) constant gray value line drawings in BMs. It also permits the deliniation and extraction of gray scale regions of a BM.

Although most operation parameters are specified in microns, some of the operations which include posting BMs on the QMT screen are specified as pixel coordinates. The pixel coordinate system is the RTPP Logical Coordinate System [4] or LCS. The LCS has (0,0) as the upper left hand corner and (1023,1023) as the lower right hand corner. The visible QMT screen size is effectively (0:860,0:680).

The pixel/micron conversion factor is derived using the axiomat zoom position to interpolate a 3rd order polynomial fitting magnification to zoom position and objective lens micron/pixel calibration. Thus if the zoom is changed after an image is acquired, the pixel/micron conversion factor for that image will be incorrect. The zoom is calibrated by turning off the stepping motor power (via a switch on the control desk), manually turning the zoom to 0.8X (minimum zoom), and then pressing the decrease zoom control until all slack is taken up at which time the stepping motor power may be turned on and the zoom will be recalibrated. Care must be taken to set up calibration, using the INIT,STATE command.

This document discusses the use of chained operators in the command syntax; lists the BMON2 operators; list alphabetically the teletype operator switches. It also documents, in the Appendix, the steps necessary to add a new operator to BMON2; the source files necessary to create the BMON2 system and its operators. An example of a sequence of operations in BMON2 is given in the Appendix in the form of a OS/8 Batch input file.

BMON2 is derived from an earlier buffer memory monitor program called BMON1.

1.1 Chained operators

Various operators such as SEGBND, EXTRACT, ZOOM, etc. are implemented as chained OS8 ".SV" core image files. /when one of these is called, the state of BMON2 (including the parsed command line) is saved in system (SYS:) disk files (SVDDTG.DA, SVBMON.DA) before the CHAIN is performed. The operator segment, on being started, has the option of restoring the state of BMON2 from these files or assuming that the state is left in COMMON. It then uses the parsed argument

specifications in COMMON. After it performs the operation, it has the option of saving the new state of BMON2. It then chains back to BMON2. The chain operation is parsed by BMON2 as follows:

[1] The operator is checked against a list of internal BMON2 operators. If it is a internal operator, it is executed within BMON2.

[2] An unknown operator (potential CHAIN operator) X is looked up on the SYS: as "SYS:X.SV". If it is found, then the state of BMON2 is saved, and the system chains to SYS:X.SV. If X is not found, then BMON2 searches the rest of the disks in the following order: DSKB, DSKC, DSKD, DSKE, DSKF, DSKG, and DSKH. If it is found on any of these disks then X.SV is copied to SYS:JUNK.SV file and the system chains to SYS:JUNK.SV. If X is not found, an error message is printed. Any disks off line are not searched.

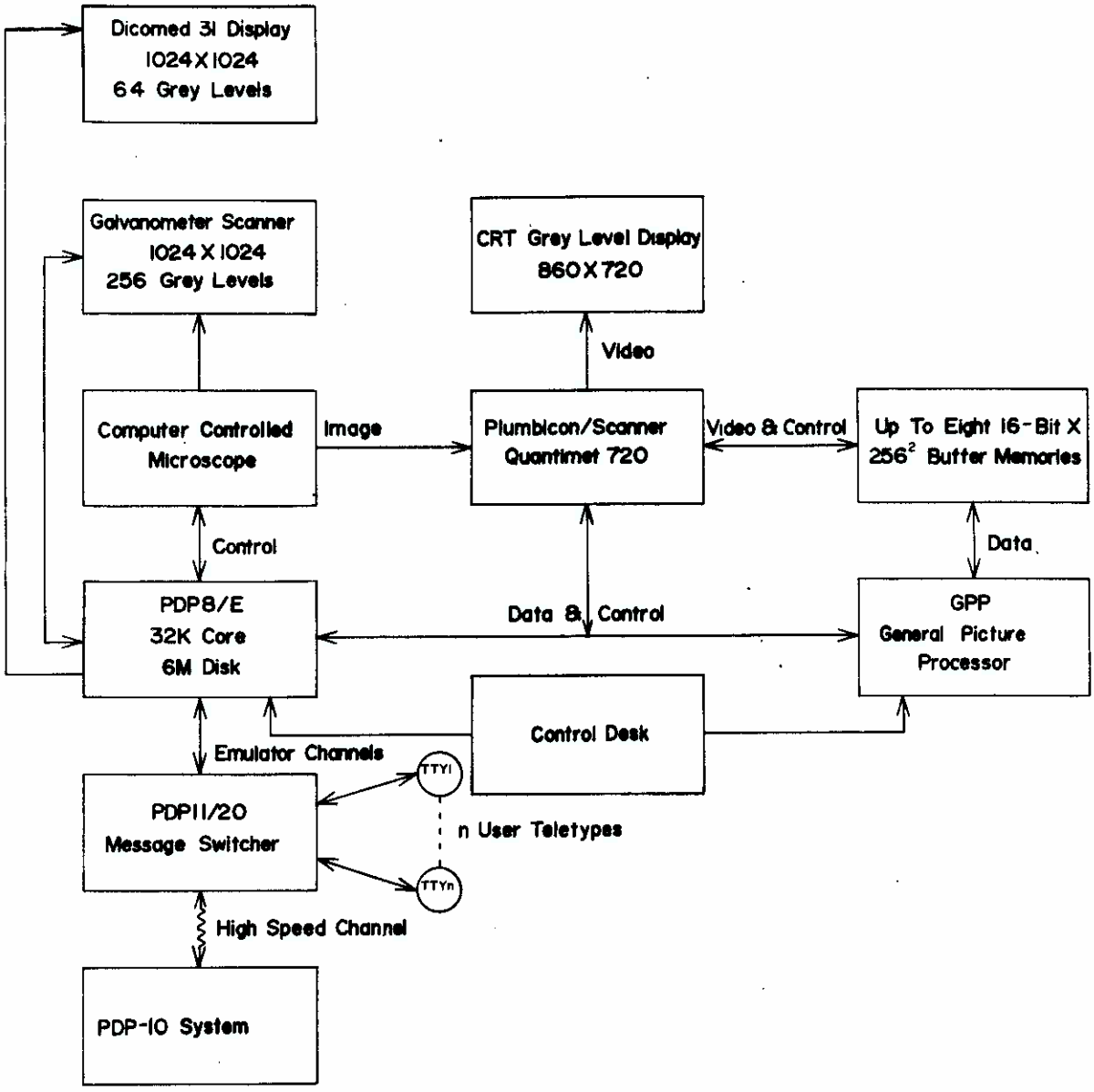
This mechanism thus has the property that operators may be added or deleted from the BMON2 system by simply adding or removing disk packs with the operator program segments on them. The system has a total of four disk drives.

1.2 Command syntax

Buffer memories are specified as "BMnh" where n is the buffer memory number and h denotes an optional byte selector (h=null for low byte, h="H" for high byte). The notation <BMi'> denotes the other half of <BMi>. So-called switches are denoted by "/" followed by an alphanumeric character. Other parameters, denoted <args>, may also be specified. Note that <args> includes decimal numbers up to 4095, physical devices such as: the 8 knob pots Pi, the control desk switches FBW1:9, and the keypad KPD (numbers 0:999). Twenty-six specially named registers called Q-registers are available for the user procedures (or user) to pass integer parameters between operations. The registers are named QR<letter>. The symbol GENSYM may be used instead of a filename (but not its extension) to indicate that BMON2 should generate a filename when referenced. Numbers and <args> may not be mixed in command lines because the parser confuses them.

Commands are entered, one per line as is specified in the Backus Normal Form (BNF) syntax given in the Appendix. Section 2 lists the actual commands and gives some examples of their use.

Figure 1 shows a block diagram of the RTPP and Figure 2 illustrates the two components of the interactive control desk.



**** Insert Figure 1 - Block diagram of the RTPP ****

Figure 1. Block diagram of the RTPP

The PDP8e computer directs the microscope stage to positions determined either manually by the operator (or automatically by the PDP10 system in the future). Images may be acquired by the buffer memories for processing by the BMON2 system at present and the General Picture Processor (GPP) in the future. Raw images as well as processed images may be displayed on the Quantimet 720. Precision scanning and display are implemented by the galvanometer mirror scanner and Dicomed display respectively. The user may interact with the system either through a control desk or a teletype.

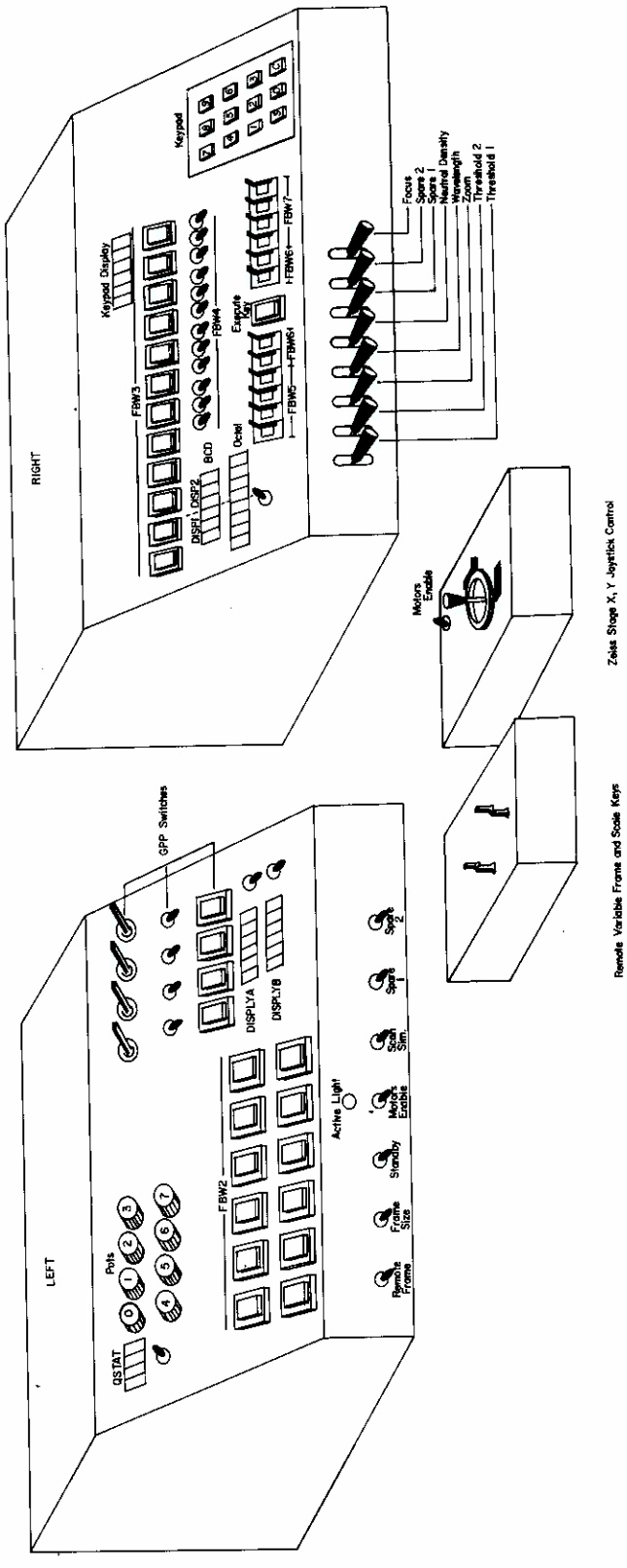

RIGHT
Keypad
Keypad Display
FBW3
BCD
Octal
DISPLY DISP2
FBW4
Expnsn Slot
FBW5
FBW6
FBW7
Focus
Spare 2
Spare 1
Neutral Density
Wavelength
Zoom
Threshold 2
Threshold 1
Motors Enable
Zeiss Stage X, Y Joystick Control
Remote Variable Frame and Scale Keys
LEFT
GPP Switches
Pots
QSTAT
FBW2
DISPLYA
DISPLYB
Active Light
Remote Frame
Frame Size
Standby
Motors Enable
Scale Sim.
Spare 1
Spare 2

Figure 2

Figure 2. Interactive control desk

The RTPP interactive control desk is situated next to the RTPP with the Quantimet video display to the rear of the control desk and the axiomat microscope off to the side. A Graf-Pen spark tablet is located immediately in front of the operator with the pushbuttons and lights mounted in two large boxes to the left and right. The remote Quantimet variable frame and scale keys are located in a small box with a movable cable as is joystick for the Zeiss axiomat (x,y) stepping stage. The latter has a long cable and may be used at the microscope for control of the stage while viewing thru the eyepiece of the microscope. The control desk controls are listed as follows going from left to right and top to bottom (for the left box first): the QSTAT lights indicate the status of the Quantimet interface; the pots are connected to A/D channels in the PDP8e; FBW2 are lighted "command" keys for the PDP8e; the remote frame switch enables the remote frame and scale switches even when the PDP8e has not enabled them; the frame size switch freezes the frame and scale sizes so that a frame of fixed size may be moved around; the standby switch places the Quantimet display and system control in standby mode; the motors enable switch (also in joystick box) enables the stepping motors when the light above it is on; the scan simulation switch moves the galvanometer scanner independently of the PDP8e so that the gain/baseline controls may be setup with using the PDP8e. For the right control box, the controls are: keypad display of keypad input for the PDP8e; FBW3 "classification" keys for the PDP8e; DISP1/2 PDP8e display lights which are decoded as BCD in the top lights and as octal in the bottom lights; FBW4 PDP8e toggle switches; keypad to input 6 BCD digits to the PDP8e; FBW5/6/7 PDP8e octal digiswitches; Execute key used to execute (interpretively by the PDP8e) instructions given in the digiswitches; eight 5-position spring loaded toggle switches control various stepping motors with a fast and slow speed in both forward and reverse directions.

2. BMON2 OPERATORS

The BMON2 commands are listed below with the specific syntax and semantics for each command. Descriptions of operators implemented via chained files have a "*" after the command name. The "*" is not typed by the user in specifying the operator. In general, <BMj> is a "sink" BM and <BMi> (or <BMi1> or <BMi2>) is a "source" BM. The "_" is a leftarrow symbol and denotes assignment of <BMi> related data to <BMj>. All pixel computations are clipped to the range [0:255]. Buffer memories are named 0 to 7. The high image of a BM is denoted by a H postscript while the low image is denoted by L or null. For example the following are legal specifications: BM2, BM2L (= BM2), BM3H.

Occasionally, operators which perform fairly complex operations (such as PIXMTA) are described in terms of other BMON2 operators using an ALGCL like description language.

2.1 Image display and acquisition commands

TV images may be acquired and stored in the image buffer memories or they may be displayed or posted form these memories using the following set of commands. These operators control these processes including the positioning of the memories and the Frame and Scale (F&S) for defining a computing window over a buffer memory. The command prefix POS- refers to positioning a BM in the LCS while SET- refers to seting the position and size of the F&S in the LCS.

2.1.1 GET

GET, <BMi> (opt. /B, /M; /A for all BMiL) - Get a Video scan from the QMT camera at the current position of the specified <BMi> into <BMi>. The /A switch causes all low byte memories to acquire data.

2.1.2 SMPGET*

<BMjL>_SMPGET (Opt. /N) - get a 512x512 image at the current Frame and Scale position on the full QMT screen into BM0H, BM1H, BM2H, BM3H (destroying their previous contents). Then copy every other pixel in every other line from these four memories arranged adjacently into <BMjL>. The effect of this operation is to acquired a (sampled) image of a 512x512 region of the TV camera.

If /N is specified, then do not do the GET operation before collapsing (BM0:3H==>BMj).

2.1.3 POST

POST, <BMi> (opt. /B, /M; /A for all BMiL) - Post the specified <BMi> on the QMT display at its current location. That is, display the specific BM contens as a gray scale (on line) image on the QMT screen in the position determined by previous initialization or the positioning commands given below. The /A switch causes all low byte memories to post data.

2.1.4 UNPOST

UNPOST, <BMi> (opt. /B, /M; /A for all BMiL) - unpost (remove) the specified BMi from the Quantimet display. It does nothing it the image was not previously posted. The /A switch causes all low byte memories to post data. UNPOST does not affect the contents of the BMs.

The information is retained until explicitly over-written or the power is turned off.

2.1.5 POSXY

POSXY,<BMi>,x,y - position <BMi> at the x,y LCS coordinates specified in pixels.

2.1.6 POSFS

POSFS,<BMi> - position <BMi> according to the position of the Quantimet Frame and Scale (F&S) computing window. The F&S position is defined as the locus of the upper left hand corner of the F&S. It is independent of F&S size.

2.1.7 SETFSXY

SETFSXY,x,y,(Opt. horizontal size, vertical size) - Set the F&S computing window to the specified LCS position and size (if the size arguments are zero, then do not change the size).

2.1.8 SETFSBM

SETFSBM,<BMi> - Set the F&S computing window position to the LCS coordinates of <BMi> and set the F&S size to 256x256.

2.1.9 SETFSREL

SETFSREL,<BMi1>,<Opt. BMi2> - Set the F&S computing window position over <BMi2> to be over <BMi1> in the same relative position as it was over <BMi2>. If the 2nd BM is not specified, then the offset relative to the upper left hand corner of <BMi1> is (0,0). This translation is given as:

```
Horiz' = If <BMi2>
                Then (Horiz-Xbmi2) + Xbmi1
                Else Xbmi1,
Vert'  = If <BMi2>
                Then (Vert-Ybmi2) + Ybmi1
                Else Ybmi1.
```

The computing window is not changed if <BMi2> is specified. If <BMi2> is not specified, then the size is set to 256x256 pixels.

2.1.10 FINDFS*

(Opt. <BMj> or QRj)_FINDFS,(Opt. <BMi> or QRi) (/F /P or /R) - if /F then find the minimum window around BMi such that pixels outside of the window but insode of the current computing window (F&S) are zero. Put the F&S at the new minimum window over BMi (over BMj if specified). If QRj is specified then save it in [QRj:QRj+3].

If /P is specified, then save the current F&S computing window over <BMi> into [QRj:QRj+3].

If /R is specified, then restore the current computing window (F&S) contained in [QRi:QRi+3] over <BMj>.

2.1.11 STDBM

STDBM (Opt. /A /B) - Set all buffer memories to the standard LCS positions which are up to 3 BMs/row with 3 rows as:

```
0 | 1 | 4
2 | 3 | 5
6 | 7 | -.
```

STDBM is also executed as part of the INIT command. BMs (4,5,6,7) are partially outside of the Quantimet video frame. The other memories (0,1,2,3) are included in the window "seen" by a Tektronix video terminal camera when used with the 3x5 Polaroid back. This is a reasonable method for photographing BM data in the QMT display.

If the /A switch is specified, then center the STDBM on the center of the Quantimet display so as to minimize the distortion of photographing the curved surface of the display.

If the /B switch is specified in place of /A then set the STDBM at the F&S position.

2.1.12 ALL384

ALL384 - Set all BM positions and the F&S computing window to the LCS position (384,384), the center of the axiomat field of view, and the F&S size to 256x256. Note that BM0 has greater priority than BM1 (in general BMi>BMi+1). This means that if any two memories occupy the same window, the memory with the lower number (i.e. higher priority) which is posted will be displayed or acquire data. The ALL384 command is used to

frame memories and the F&S, for among other uses, setting a window for the galvanometer scanner, acquiring several iamges at the same position over time, positioning frames for a movie.

2.1.13 SHOWMOVIE*

SHOWMOVIE - requests a list of up to 16 ordered movie frames at their current LCS positions. The /M switch used in a particular frame specification causes the other half of a BM to be posted in masked video (sign bit is the mask) mode. A movie frame is a <BMi> name. After the list is entered (terminated by a null frame name) the movie is shown.

The sequence is repeated at a frame rate determined by the setting on knob pot 0. The minimum rate is 0.1 seconds/frame, the maximum rate 3.1 seconds/frame. The movie may be restarted at any time by typing ^O. It may be stoped at the current frame by typing ^S and continued by typing ^Q. The movie frame number currently being shown is posted in the right QMT display register when it is temporarily stopped with ^S. The movie is terminated by pressing CLASSKEY[11].

An example is given of a movie made from 4 BMs where it goes to the end of the movie and then reverses ("pogo-stick" motion). Another movie method is the "cliff-hanger" which simply restarts at the first frame after the movie is finished.

```
*ALL384
*SHOWMOVIE
*BM0
*BM1
*BM2
*BM3
*BM2
*BM1
*

*ALL384
*SHOWMOVIE
*BM0
*BM1
*BM2
*BM3
*
```

2.2 Input/Output operations

The set of I/O operations below permit complete flexibility in moving data between buffer memories and auxiliary storeage such as disks, Dectape and magtape. The READ and WRITE built-in operations are mapped to the XMITBM chained segment. Images may also be copied to the Dicomed display and windows in the BMs dumped on the lineprinter. Images may be acquired from the galvanometer scanner. Provision is made for running various automatic cameras.

2.2.1 WRITE*

<filespec>_WRITE,<Opt. BMi> (Opt. /B, /L, /M, /I, /G, or /Q data mode switches) (/C, /K, /A modifier switches) - Write data from the specified data source to the file. The /C switch causes the program to request a file header comment from the user (which may not be specified under Batch). The /K switch causes it to request a keypad class number in the range of [0:999] before continuing. The following cases are allowed.

<filespec>_WRITE,<BMi> (Opt. /I, /A) - BM gray scale image data. The /A switch specifies that both halves of the BM will be transfered.

<filespec>_WRITE,<BMi>/M - BM binary image image data generated from a BM by taking all pixels with values geq 128 as 1 and all pixels < 128 as 0.

<filespec>_WRITE,<BMi>/B,(Opt. boundary number) - Dump boundary(ies) contained in the BM created with SEGBND pointed to by set pointer QRZ.

<filespec>_WRITE,<BMi>/L - Dump line drawing data structure contained in the BM created with EXTRACT pointed to by QRZ.

<filespec>_WRITE,<BMi>/G, (sampling d) - Dump a 256x256 galvanometer scanner image (8-bit data) centered at the logical center of the microscope (LCS coordinates (384,384)). A sampling parameter d specifies (if non zero) how many samples are to be taken at each position and averaged (up to 16 samples). The ALL384 command should be used before doing a scan in order for you to see the frame of the scan window.

<filespec>_WRITE,<BMi>/Q (programming switch /n) - Run and dump Quantimet function computer data programmed according to one of the (/1,/2,/3,/4,/5,/6,/7 switches - see QDATA for details).

*** note: /B, /L, /G switches have bugs or not implemented***

2.2.2 READ*

<Opt. BMj>_READ,<filespec> (/B, /L, /M, /I, /A, /D, /R or /H data mode switches) - Read data from the file into the specified data structure (or device). The comment field of the file header is always printed. The following cases are allowed:

<BMi>_READ,<filespec> (Opt. /I, /A) - BM gray scale image. The /A switch specifies that the both halves of the BM will be read from the file.

<BMi>_READ,<filespec>/B - boundary data will be loaded as black (255) into the BM as specific pixels. The remainder of the image will not be changed. Also the boundary will be loaded as a list into the node data structure pointed to by the set of boundaries QRZ.

<BMi>_READ,<filespec>/M - BM binary image generated by setting pixels in the image to 255 which correspond to file 1's and pixels to 0 in the image which correspond to 0's in the file.

<BMi>_READ,<filespec>/L - load the line drawing data structure into the BM list structure pointed to by QRZ.

READ,<filespec>/D (Opt. /E to erase before drawing), (Xlcs,Ylcs if /F) - draw either a boundary (/B generated) or gray scale image (/I or /G generated) data structure on the Dicomed display. Note that boundary data is drawn as black and that image data is displayed as the most significant 6-bits of the 8-bit gray scale data. If /F is specified then specify the position of the image at the (Xlcs,Ylcs) specified values. Otherwise, the image position is that of the computing window (F&S).

READ,<filespec>/R - restore the state of the stepping motor devices of BMON2 to that contained in the file header. I.e. (x,y,f,z,w,n,tb,tc).

READ,<filespec>/H - print the contents of the file header. See details on header contents in the appendix.

(<BMj> or /A)_READ,MTAi:<Opt. filespec> - read in a single file in <BMi> or 4 files into BM0,1,2,3 (if /A specified) from the specified magtape unit MTAi:. The READ is actually doing a REVIEW/X. See REVIEW for more details.

*** note: /B, /L, /G switches have bugs or not implemented***

2.2.3 WINDMP*

WINDMP,<BMi>/U,(Opt. numerator, denominator scale factor) - Dump the <BMi> pixel values in the specified F&S window (to a maximum width of 18 pixels and a height up to 256 pixels). The pixel data is scaled by: numerator/denominator before being printed. If the OPENFILE is active, then the data is also dumped on the LPT: file WINDMP.DA.

2.2.4 DRWDICOMED*

DRWDICOMED,<BMi> (Opt. /E to erase the screen before writing), (Xlcs,Ylcs if /F) - Display <BMi> on the Dicomed at the current F&S position. The Dicomed uses 6-bit data so that the most significant 6-bits of the 8-bit BM data is used. Therefore, it may be necessary to SCALE or CONTRAST stretch some images before displaying them on the Dicomed. If the /F switch is specified then position the image on the Dicomed at the specified (Xlcs,Ylcs) positions, otherwise position it at the computing window (F&S) position.

2.2.5 MAG10*

MTAi:_MAG10,<filespec>,BMON2 (Opt. /C, /D, /V) - Dump the specified file(s) according to the file specification (with wild card characters *, ? legal) onto MTAi: (where i is unit 0 or 1) and return to BMON2 when finished. In this case, the specific mention of "BMON2" in the call informs MAG10 that it is to chain back to BMON2 after it is finished. If it is not mentioned, it will request another magtape transfer operation. If the /D switch is specified, then post delete the files after they are transfered to magtape. The /V switch causes each tape record to be verified by being read-compared after it is written onto the tape. Files may also be transfered the other way using:

*<DSK:filespec>_MAG10,<MTAi:filespec>,BMON2.

MAG10 is documented in [10]. For example, to dump and delete all picture (*.PX) files on DSKG: with the current date (/C) onto MTA0:,

MTA0:_MAG10, DSKG:.PX/D/C, BMON2

Only OS8 files may be transfered to/from magtape. Thus BM data transfer to/from magtape must be done via the use of disk file intermediary data files (using WRITE/READ commands).

2.2.6 PIXMTA*

MTAj:(Opt. file)_PIXMTA,BMi (Opt. /C /X /R /H /M /A /F /T), (Opt. focus serial section size in steps k if /F (or time increment in seconds if /T), number of serial sections m if /F or /T specified) - dump a set of images to be acquired from BMi to MTAj:GENSYM.PX. If /R is specified then rewind MTAj: first. If /H is specified, then print the header after each file is written. If /X is specified, then do not wait for the keypad to be pressed but write the specific BMi (without doing a get) and return immediately after it is written. The algorithm is described as follows using a BMON2 like notation:

```
[1] "Initialization"
        Save post status;
        If not /X
                Then
                Begin "unpost"
                UNPOST/A;
                If /A
                   Then STDBM/A;
                End "unpost";
old!time_GET!TIME;
[2] "Data acquisition loop"
        If (X,Y,FOCUS,ZOOM,TH1,TH2,ND,WV) switch
                Then MANUAL control;
        If CLASSKEY[0] Or FBW4[0] switch
                Then
                Begin "flicker camera/BM data"
                POST; wait 0.1 seconds;
                UNPOST; wait 0.1 seconds;
                End "flicker camera/BM data";
        If keypad data
                Then If KPD data > 0
                        Then Goto [3];
        If /A
                Then STDBM/B "position BM0,1,2,3 from F&S"
                Else POSFS,<BMi> "position BMi from F&S";
        If /X
                Then goto [3];
        If ^O
                Then Goto [4];
        If /F
                Then If (m_m-1) < 0
                        Then Goto [4];
                        Else
                        Begin "get next serial section"
                        MOVSTATE, SF,k,1/N;
                        If /A
                                Then GET/A
                                Else GET,<BMi>;
                        Goto [3];
                        End "get next serial section";
        If /T
```

```
                Then If (m_m-1) < 0
                        Then Goto [4];
                        Else
                        Begin "get next time increment"
                        Until GET!TIME Geq old!time+k)
                                Do Continue;
                        old!time_GET!TIME;
                        Print(old!time and date);
                        If /A
                                Then GET/A
                                Else GET,<BMi>;
                        Goto [3];
                        End "get next time increment";
        "Continue polling loop"
        Goto [2];
[3] "Get data and write it on MTA"
        If not /X
                Then
                If /A
                   Then
                   Begin "get 512x512 BM data"
                   STDBM/B "position BM0,1,2,3 from F&S";
                   GET,BM0H;
                   GET,BM1H;
                   GET,BM2H;
                   GET,BM3H;
                   End "get 512x512 BM data"
                   Else
                   Begin "get single BM data"
                   POSFS, BMi;
                   GET, BMi;
                   End "get single BM data";
        If /C
                Then get picture header file comment;
        If Opt. ext = null
                Then ext_PX;
        If Opt. file = null
                Then
                file_If 3rd Opt. file
                        Then
                        GENSYM(3rd file, gensym)
                        Else
                        Begin "increment gensym"
                        gensym_gensym+1;
                        GENSYM(gensym name, gensym);
                        End "increment gensym";
        If Not /A
            Then MTAj:file.ext_write MAG10 format, BMi;
        If /A
            Then
            For i_0 step 1 until 3 Do
                Begin "write out 4 files"
                file_file[1]&CVSTRING(i)&file[3 to 6];
                MTAj:file.ext_write MAG10 format, BMiH;;
                End "write out 4 files"
        If /H
                Then print picture file header on LPT:;
```

```
        UNPOST, BMi;
        If /X
                Then Goto [4];
        Goto [2];
"[4] Restore state and return to BMON2"
Restore frame and scale;
Restore post status;
Restore BM positions;
Return to BMON2 preserving the state;
```

If the /M switch is specified, PIXMTA makes continued requests for transfer commands (as for example:

```
        *MTAj:_PIXMTA/X/M,BMi).
```

until a null command or no /M is specified. It then returns to BMON2. If the optional 3rd file specification is used, it uses the first 2 characters to specify the GENSYM name and does NOT increment the gensym counter.

If an output file extension is specified, then it will be used otherwise the .PX extension is assumed.

The CLASSKEY[0] or FBW4[0] switch is used to flicker the current contents of the BMs on and off in order to align sequential serial sections (such as on film negatives or microtome slices etc.). Note that the Quantimet TV camera may be rotated as well to aid in alignment.

2.2.7 REVIEW*

BMj_REVIEW,MTAj:(Opt. file) (Opt. /R /H /X /M /A) - review a series of picture files on MAG10 formated magtape (generated either with MAG10 or PIXMTA) by sequentially reading them from the MTAj: to BMi. If a specific file is to be loaded, then the file name is specified. If the next sequential file is to be used then no file specification is required. Pressing the red "execute" key on the right control desk causes the next image to be read into the BMj.

The /R rewinds on initial entry. If the logical end of tape (EOT) is encountered, the tape is rewound and control transfered back to BMON2. The /H prints the picture file header. The /X switch causes the file to be read without pressing the "execute" key and to return immediately afterwards to BMON2.

The /M switch used with the /X switch gets additional requests from the command decoder.

The /A switch indicates that the next 4 magtape files meeting the file specification are to be read into

BM0L, BM1L, BM2L, and BM3L.

2.2.8 BNDPRINT*

(Opt. <BMj>)_BNDPRINT,<.DA filespec>, (Opt lower CC, upper CC) (Opt /H /L /A /T /B /M /D /1 /2 /V), (Opt. averaging window width w) - (with no switches) print the SEGBND produced .DA boundary file on the line printer for the connected component numbers specified (CC#) with [2:255] being the default range of allowable CC numbers. Control/O stops the printing. The feature data in the file is also printed on the lineprinter before the (x,y) coordinate data.

In addition to the (x,y) data, the Freeman chain code (CCD), chain code difference (CCDF) (point i and i-1), average chain code (AVC) over the window w (range [5:13] default 13), chain code difference (AVD) over the averaging window w [16] (where the middle 3 points of the window are not considered in the difference) and RLM semantic labeling for each point (at a 0 and 90 degree orientation) is also printed. After the data is printed (/H omits the individual x y chain code semantic label data), the bending energy [15], a frequency histogram of the chain codes, a histogram of the chain code differences and a joint frequency histogram (0 and 90 degree) semantic labels is printed. If /B is specified, then no point data or histograms are printed - only the CC's features and its bending energy.

Chain code difference specification

If the /D switch is specifed, then it requests a difference specification from the command decoder

*TH1,TH2,(Opt. scale,cornerity thr,heuristic thr)

where: low thr=TH1/scale (or 1)
high thr=TH2/scale (or 1)

In addition, /1 negates low thr, and /2 high thr CCDF's in [low thr : high thr] are printed. Thus the maximum range is [-8.0:+8.0] specified by *8,8/1. Concavites are denoted by positive AVD's and convexities by negative AVD's. Initial values for the range may be specified by noting that AVD is the angular differance divide by 45 degrees. Thus most corners involve at least a 45 degree difference (or a minimum threshold of 1).

If /L is specified along with <BMj>, then the (x,y data is used to label points (meeting the specified label criteria) in <BMj> with black (255). The /T switch traces points not meeting the criteria as gray scale

50. Switches /A (accept all labels), /M (request semantic label names from the command decoder terminating the list of 2-tuples with a null entry), and /R (rotate x,y data -90 degrees before labeling) are operational when /L is specified. In addition, /D may be used to label points with the chain code difference within the specified range. If /C is specified with /D, then the corner detector is turned on.

Semantic Label specification

The /M switch causes additional requests for semantic labels to be made from the CD with a null request terminating the list. That is, a set of 2-tuples is specified as:

```
*<semantic name 0 deg>,<semantic name 90 deg>
```

Where "?" denotes any name will match. For example,

```
*NEWRUN,CPNRHT
*CPNLFT,NEWRUN
*EXNLFT,?
*?,EXNRHT
*
```

will label a point if the following Boolean expression is true.

```
    (NEWRUN(0) and CPNRHT(90))
or  (CPNLFT(0) and NEWRUN(90))
or  EXNLFT(0) or EXNRHT(90)
```

The semantic labels are generated by the RLM algoritm part of the SEGBND operation and are listed here for reference. See [14] for more information on the semantics of the specific labels.

Index	Code	Semantics of boundary point
1	CGLHSP	explicit split change HAIREP to left split
2	CGRHSP	explicit split change HAIREP to right split
3	CPALFT	complete left adjacent run
4	CPARHT	complete right adjacent run
5	CPNLFT	complete left non-adjacent run
6	CPNRHT	complete right non-adjacent run
7	CMPUNY	complete unary run
8	DUP-PT	duplicate point
9	EXALFT	extend left adjacent run
10	EXARHT	extend right adjacent run
11	EXNLFT	extend left non-adjacent run
12	EXNRHT	extend right non-adjacent run
13	HAIREP	new run hair end point
14	IMPLFT	implied split left CCW split
15	IMPRHT	implied split right CCW split
16	IMPLFH	implied split left hair

```
17      IMPRHH  implied split right hair
18      INCDPT  explicit split included point
19      MRGLFT  merge run left
20      MRGRHT  merge run right
21      NEWRUN  new run started
22      NULPNT  null point
23      SPCW-L  explicit split CW left
24      SPCW-R  explicit split CW right
25      SPCCWL  explicit split CCW left
26      SPCCWR  explicit split CCW right
27      NEWOVF  new run overflow (>128 runs)
```

Corner measure

A corner measure defined here is extends some concepts of [17]. A corner is a 3 segment edge (t1,extremum,t2) where the extremum CCDF is within the chain code (/D) difference range. The side lobe segments are defined to be those points between the extremum and the last point before the AVD changes sign from that of the extremum.

The cornerity as defined here differs from [17].

```
cornerity = Max(AVD of extremum)/
            length(Max(1,t1))*length(Max(1,t2)),
```

As the two side lobes may not be the same length, a measure of skewness is computed:

```
tmax=Max(length(t1),length(t2))

t12skew=tmax/tmin,
te12skew=tmax/length(exe)
tmin=Max(1,Min(length(t1),length(t2)))
```

where: length(x)=Max(1,x).

The aperture of the corner is defined in terms of the average chain codes of the begining of t1 and end of t2.

```
t1angle=(If t1=0
                Then first AVC of extremum
                Else first AVC of t1).

t2angle=(If t2=0
                Then last AVC of extremum
                Else last AVC of t2).
```

The aperture is computed using modulo 8 arithmetic as:

```
aperture = |Min((t1angle-t2angle),(t1angle-(8-t2angle)))|.
```

The normal (center of the aperture) is defined (using modulo 8 arithmetic)

```
normal = t1angle + aperture/2.
```

The curvature contribution of each segment is computed as follows:

t1avd=Sum AVD over t1,

t2avd=Sum AVD over t2,

extavd=Sum AVD over the extremum.

The curvature/segment is computed as:

```
t1c = (If t1=0
         Then 0
         Else t1avd/length(t1));

t2c = (If t2=0
         Then 0
         Else t2avd/length(t2));
```

exc=exeavd/length(exe)

The percent of extremum curvature to total corner curvature is:

pcntcrv=exe*100/(t1c+t2c+exe).

Pairing corners

Each corner (labeled by the maximum AVD point in the extremum) which has a cornerity > the specified threshold is queued for analysis when the entire boundary has been scanned. At that time, all corners are compared, two at a time, to see if they are complementary corners. Several correlation like functions are compute of the above corner data between the two corners. A heuristic summary function is then computed. Pairs whose heuristic function > the specified threshold are candidates for spliting. If the /V switch is also specified, then the candidate splits are passed to BSPLIT.SV (via chain) through a queue in COMMON. The following functions are computed for two corners i and j.

The correlation functions are as follows:

Euclidian distance between corners:

bp=boundary length

rsq=(bp+(Xi-Xj)**2+(Yi-Yj)**2)/(bp**2)

The relative balance between the two segments where the corners are specified by index(.)

kd1=|index(i)-index(j)|

kd2=bp-kd1

balance=Max(1,Min(kd1,kd2))/bp

In order to give added weight to boundaries of normal size for the 256x256 image, a weighting function 'optsize' takes this into account.

optsize = 200/(200+bp).

The optimum balance can now be computed.

optbalsq=(balance/(balance+optsize))**2

The normal correlation would be a maximum if the two normals are pointing directly at each other. Therefore,

dnorm = |4 - (normal(i)- normal(j) Mod 8)|.

The relative correlation is:

apwsq=aperture(i)*aperture(j)+0.001

The normal difference is then weighted by the . A large aperture negates well fitting normals.

dnorm' = (0.5+dnorm)/(1.0+apwsq).

The cornerity correlation expects similar high cornerities:

corn=cornerity(i)*cornerity(j)/(cornerity(i)+cornerity(j)+1).

A "nick" is a small concavity generally not associated with another corner. For small windows it generally has a high percentage of curvature concentrated on the extremum (t1 and/or t2 are 0).

If pcntcrv(i)=99% or pcntcrv(j)=99%
Then nick<==1 else nick<==0;

The percent curvature correlation is:

pcnt=pcntcrv(i)*pcntcrv(j)/(pcntcrv(i)+pcntcrv(j)+1)

The angular differences are computed using modulo 8 arithmetic:

d1=t1angle(i)-t2angle(j)

d2=t1angle(j)-t2angle(i)

capw=|d1*d2/apwsq|.

Pairing heuristic

Finally, the heuristic function is:

$$h(i,j) = \frac{(1-nick)*(corn*(pcnt+capw)*optbalsq)}{(rsq*dnorm'*10)}$$

Values of h range from 0 to several hundred although values around 10 and up are found for a range of cusps generated from touching circular objects such as white blood cell nuclei.

BSPLIT split region verification using texture

As we mentioned above if there exist corner pairs for which h > heuristic threshold, then the BSPLIT function is invoked by BNDPRINT (when /C/D/V are specified). BSPLIT requests the following information:

*BMj_BMi1,BMi2,n,d,scale

Where: BMj is the buffer memory in which to store the resultant split image,
BMi1 is the original gray scale image,
BMi2 is the connected component image,
texture threshold = (n/d)*scale.

The BMj' is used as a scratch memory in which to compute the texture properties of the potential cell junction. Two texture samples are taken at the intersection of the two corners for a 10x10 and 30x30 (less the 10x10 center region) pixel window. The gray scale co-occurrence matrix Pij (for a distance of 1) are computed and then the texture function Ht is then computed as follows:

Ht=10*(Txt(center)+Txt(outside-center))/Txt(center),

Where: Txt(k)=T15(k)*T19(k) for region k,

T15(k) is the coefficient of variation of Pij for region k,

T19(k) is the 2nd diagonal moment of Pij for region k.

Both T15 and T19 are discussed in the BNDPRINT function and are taken from [18]. Values of Ht need to be experimentally determined for each problem domain. For bone marrow nuclei at 800X (green-blue normalized image) touching nuclei have a Ht about 20 or greater while regions which might other constricted regions of a single object range between 10 and 20.

If the texture heuristic function is > texture threshold then the copy of BMi1 (previously stored in BMj) is split at the junction.

The actual splitting using the following algorithm.

Foreach (x,y) on the line [x1:y1] to [x2:y2] Do
Begin "zero junction"

```
If x=xold or y=yold
        Then zero pixel at (x,y)
        else zero 3x3 neighborhood at (x,y);
x_xold, y_yold;
End "zero junction";
```

If /D is specified (without /L) then information about the corner points is printed. This includes:

[boundary index, (x,y) max AVD of extremum, length t1, length extremum, length t2, cornerity, t1c, exc, t2c, %curve, t12skew, te12skew, t1angle, t2angle, width and normal angle].

If /C/D and not /L is specified, then corner pairs print the following:

[(i,j), rsq, balance, dnorm, corn, pcnt, capw, nick, optbalance, dnorm', heuristic].

If /C/D/L is specified, then corner pairs meeting the heuristic value have 3x3 black squares drawn over the corner points in BMj.

Evaluating corner pairs

After the corner pairs > heuristic threshold are found, they are sorted by heuristic value and then the highest valued pairs are saved in the message stack to BSPLIT. In addition they are printed on the lineprinter.

2.2.9 CAMERA*

CAMERA (Opt. /A to take picture automatically),(Opt. time exposure in seconds) - print a message on the teletype and lineprinter, and ring the teletype bell until a carrage return is typed. If /A is specified, then print the message but do not pause for user response. Instead take a picture by signaling the automatic 35MM camera (or 16 MM movie camera) by closing the camera relay (EXOUT channel 0 bit 11) for 1 second (unless a time exposure value is specified).

2.3 Control desk operations

The control desk has various control keys ([3],[4]). The 12 "command keys" 0 to 11 may be assigned to BMON2 command lines typed on the teletype by appending the syntax "/S=nn" to the command. The nn is the octal value of the assigned command key. Pressing an assigned command key is then equivalent to typing the assigned command. The other control keys (which include switches, pots, keypad, joystick and other stepping motor controls) are discussed in the section on Normal Operating Conditions.

2.3.1 CMDKEYS

CMDKEYS - List the user assigned command keys. ^S, ^Q, ^O are active.

2.3.2 SAVCMD

<filespec>_SAVCMD - save the 12 command key commands in the specified OS8 output file. If the SAVCMD is used on the command keys, it is possible to incrementally save the key assignment, especially if the file is specified using the GENSYM file specification. For example,

```
*DSKG:GENSYM.CD_SAVCMD.
```

2.3.3 RSTCMD

RSTCMD,<filespec> - Restore the 12 command key commands from the OS8 input file (previously created with the SAVCMD operation). If the RSTCMD <file> command is used on one or more command keys, it is possible to build a command tree of many more than 12 commands. Such a command tree might have commands at different levels. For example, keys might be assigned at the top level to MAINKEYS.CD, then some of its key asssignments might look like:

```
*RSTCMD, DSKG:GRADKEY.CD/S=2
*RSTCMD, DSKG:SMOOTHKEYS.CD/S=3
*RSTCMD, DSKG:HISTKEYS.CD/S=4.
```

Each of these command key sets could include a return to the top (or if we were nesting the sets, to the previous) level by

```
*RSTCMD, DSKG:MAINKEYS.CD/S=0.
```

It is useful to print the contents of the keys at any level, so that a particular key (eg 11) could be use to print the current keys:

```
*CMDKEYS/S=1.
```

2.4 Initialization and state inquiry commands

BMON2 contains various parameters which corresond to the state of the buffer memories, the stepping motor positions, lens magnification etc. This set of commands allows the user to initialize, interrogate and change them.

2.4.1 INIT

INIT,(Opt. STATE word to clear 8 stepping motors or /H to zero histogram only) - Power clear the PDP8e and initialize BMON2 parameters as follows:

(a) Set the magnification of the axiomat [3] according to the following switch assignment:
  /1 for 1X
  /2 for 2.5X
  /3 for 5X
  /4 for 10X
  /5 for 25X
  /6 for 50X dry
  /7 for 50X oil
  /8 for 100X oil where 100X is the default;
(b) Unpost all images;
(c) Set F&S to 256x256 image at LCS (384,384);
(d) Freeze F&S size keys when the left control desk "frame size" switch is changed from "manual" to "auto";
(e) Zero assigned command key list;
(f) Set all BMS to STDBM configuration (/A active);
(g) If the word STATE is specified, then reset the stepping motors;
(h) Clear the BM "freestore" active switch for BM7 (the freestore is re-initialized by any program using it if the switch is clear). This permits the freestore to be used between operators.
(i) Clear Q-registers QRA:QRZ.
(j) Zero the histogram array. If /H then don't initialize anything else.

2.4.2 EXIT

EXIT - Return to OS8 after saving the state of BMON2 (i.e. COMMON) in the SYS: files (SVDDTG.DA, SVBMON.DA). This allows the return of control to essentially the same conditions on the next call to BMON2. Exit should always be used at "*" level while the ^C may be used while waiting for input at non-"*" BMON2 level. For example:

```
.R BMON2
*INIT
*POST,BM4
*EXIT

.... OS8 operations

.R BMON2
* (continues where left off)
```

2.4.3 PARAMETERS

PARAMETERS (Opt. /P) (also Opt. /R, /M, /K, /G or /E) - print the state of BMON2. The state consists of:

(a) GENSYM file name generator value - print only GENSYM if /G.
(b) Lens X, Zoom X, microns/pixel calibration.
(c) F&S positions and sizes and area.
(d) BM posting active and mode status - omit if /P.
(e) Active BM x,y positions - omit if /P.
(f) Values of 8 stepping motors:
(stage-x, stage-y, focus, zoom, threshold-1, threshold-2, wavelength, neutral-density)
- print only stepping motors if /M.
(g) Values of the 8 control desk knob potentiometers - omit if /P, print only knobs if /K.
(h) Values of the 26 Q-registers QRA:QRZ - print only if /R else omit.

If the /E switch is used, then print the value of the argument listed after the PARAMETERS operators. Argument variables include: Pi (pots), FBWi (control desk switches), KPD (keypad), QRi (Q-registers). For example,

```
*PARAMETERS/E,P3
```

prints the value of knob pot 3.

The /P appended to any built-in (non-"*") operator will cause the PARAMETER command to be executed after the built-in command is finished. The parameter switches may also be passed as long as they do not conflict with those of the built-in.

The /8 appended to any PARAMETER command will print the results on the LPT: whether the spooler is on (OPENFILE) or not.

The /9 switch causes the second alphameric file name in the command sequence to be printed inside of [...] on the lineprinter. No other parameters are printed.

2.4.4 SETGENSYM

SETGENSYM, (two character name), (initial value for generator) - Initialize the GENSYM name to be used for automatic file name generation. The name consists of a two character prefix followed by a 4 digit decimal number (0000 to 4095). For example PX0572 would be specified by:

```
*SETGENSYM, PX, 572.
```

The current GENSYM value may be obtained from the PARAMETERS. When GENSYM is used, it will be replaced in the syntax entered with the generated symbol name where the value has been incremented by 1 (4095 wraps around to 0000). Thus if the GENSYM is PX0572, then

```
*DSKG:GENSYM.PX_WRITE,BM3
```

would write out BM3 into file PX0573.PX.

Only one letter-number generator series may be active at any one time. Thus to alternate between different series it is necessary to keep resetting GENSYM between using the different series.

2.4.5 LOADQR

<Q-register>_LOADQR,value - load the specified value into the Q-register. If the Q-register name is illegal, then the operation does nothing. Note that Q-registers QRA:QRD are used as a message area for various operators to pass results (e.g. HISTOGRAM and COMASS) and are thus volatile. QRZ is used to pass a list pointer back to BMON2 from the SEGBND operator. A complete list of implied Q-register usage is given in the appendix.

2.4.6 EVAL

<Opt. Q-register>_<arg1>,EVAL,<arg2>,<Opt. n if condition> (arithmetic switches: /A, /M, /P, /D, /X, /N; conditional switches: /G, /L, /E) - evaluate the numerical arguments for the arithmetic switches and print the result. If an output Q-register is specified, then store the results in it. If a conditional switch is specified instead, then if it fails, it skips the next n calls to the command decoder ("*") for BMON2 running under Batch. If Batch is not running it does nothing.

The switches correspond to the following operations:

Switch	Operation
/A	ADD
/M	MINUS
/P	PRODUCT
/D	DIVIDE
/X	MAX
/N	MIN
/G	.GT.
/L	.LT.
/E	.EQ.

For example, to start another batch job when a counter QRG overflows, the following sequence might be used.

```
*QRT_LOADQR,value
*QRO_LOADQR,1
*QRG_EVAL/A,QRG,QRO
*EVAL/L,QRT,QRG
*BATCH,NEWJOE.BI
```

2.4.7 HELP*

HELP, (Opt. command name> - print the BMON2.HL file on the TTY: (LPT: if /L) with control/S, /Q and /O active. When a specific command is mentioned, it searches BMON2.HL for the paragraph which discusses that command and prints only that text relating to that command. The /C switch lists the first lines of paragraphs which begin with a number. Thus a list of all commands may be requested.

2.4.8 OPENFILE

<Opt. LPT: output file>_OPENFILE - open the specified output file. If no output file is specified, then the LPT: is opened. Since the output is buffered, it may not immediately appear on the LPT: on being generated. The spooling status is maintained in the state of BMON2 so that it is prevserved when exiting and restarting.

The LPT: may be assigned to the DSK: for generating files rather than printing on the line printer. This is done via OS8 as follows:

```
.ASSIGN DSKG: LPT:
```

2.4.9 CLOSEFILE

CLOSEFILE - The line printer device output buffer is emptied, then it closes the output spooling file previously opened with the OPENFILE command.

2.4.10 MOVSTATE*

MOVSTATE, <state name>, <distance numerator n, denominator d> (Opt. /A for absolute otherwise relative) (/M for a minus distance) (Opt. /N for distance to be specified in steps otherwise it is specified in the rational units for the specific motor) - move the x or y stage, focus, zoom, light source wavelength or ND filter, or QMT threshold B or C the specified distance (n/d) in rational units for the specified device.

<state name>::= SX | SY | SF | SZ | SW | SN | SB | SC.

Normally, the state is moved a relative number of steps unless /A is specified. The /N switch overides the rational unit conversion and allows the user to specify the distance in steps. The rational units for x, y, focus are in microns while the zoom is in X, etc. Negative distances are specified by the "minus" /M switch.

For example,

```
*MOVSTATE, SF, 5, 3
```

moves the focus motor 5/3 of a micron while,

```
*MOVSTATE, SY, 100, 1 /M /A
```

moves the y stage stepping motor to -100 steps from the initial stage fiducial mark (set by INIT,STATE).

2.5 Auxiliary program commands

Various auxiliary commands are available for communicating with the OS8 Batch system and with the special hardware interfaces of the RTPP.

2.5.1 BATCH

BATCH,<Batch ".BI" file spec> - submit the specified OS8 Batch file to OS8 Batch. This permits one to assign sequences of BMON2 operations to command keys by running BMON2 in the Batch job and then supplying it with a sequence of operations to be performed. Note that the NOBATCH operator will turn off Batch and return control to the user teletype. Thus, insert a NOBATCH command at the end of the list of BMON2 operations to be performed in order to return control back to the user.

For example, the Batch program in the Appendix would be submitted as follows:

```
*BATCH,EXPR34.BI
```

2.5.2 NOBATCH

NOBATCH - Turn off OS8 Batch in case it is on. This is necessary at the end of a Batch job which is using BMON2 if the user wishes to regain teletype/control desk control of BMON2. For example,

```
$JOB
.R BMON2
*INIT
*POST,BM1
*POST,BM3
*NOBATCH
$END
```

2.5.3 APPLY*

<Opt. BM0>_APPLY,<.BT template file>,<Opt. BM1,BM2,BM3> (Opt. n1,n2,n3 arguments) - Apply the Batch .BT template file to the specified argument list to generate a specific instance Batch input file "SYS:JUNKTM.BI". This latter file is then submitted to Batch. The APPLY operation copies the template file into JUNKTM.BI. In the process, it searches for the patterns "?i" and "%j". The "?i" template corresponds

to the pattern for BMi and the "%j" to the pattern for nj. The actual values are then copied into the JUNKTM.BI file instead thus allowing dynamic argument evaluation for complex operations at run time.

2.5.4 SETIOT

SETIOT,<output IOT name>=nnnn - output nnnn (octal) using the specified PDP8e Input Output Transfer instruction (IOT). The following IOTs are implemented:

DET34, DETB, DETC, GETMSK, HPL, HSL, LDXP, LDYP, LFBW2 LGALX, LGALY, LQDT1, LQDT2, LQDT3, MSKADR, MSTAG, QPROG1, QPROG2, QPROG3, QPROG4, QPROG5, QPROG6, QPROG7, QPROG8, QSTAT, SIZEA, SIZEC, SIZEM, SIZES, STEP, STQMT, VPL, VSL, EXADR, EXOUT.

The PDP8e IOTs mentioned here are discussed in [3,4].

2.6 Synthetic image operators

Synthetic image operators are used to generate image data based on external (non-image) parameters. The /C switch complements the gray value for all operators in this group.

2.6.1 COLOR

<BMj>_COLOR, (Opt. density) (Opt. /C for black) - color entire image specified density. If no arg is given, it will zero the memory, If /C and no arg then it colors it black.

2.6.2 ZERO

<BMj>_ZERO (/C for black) - zero BMj by coloring it white. The entire image is zeroed, thus /U is inoperative. Note that ZERO is a special case of the COLOR command with a default density of 0.

2.6.3 GRAYBAR

<BMj>_GRAYBAR, (Opt. graybar width w, default=256) (Opt. /C) - display horizontal graybar where the bars are 16 pixels wide with the first bar =255 are related by g(y=0)=255, g(y+16)=g(y)/sqrt(2); If the /1 switch is specified, then a linear scaling is used: g=255-y. If the width is specified, overwrite the graybar on the right side of <BMj> a width w.

2.6.4 TEXT

<BMj>_TEXT, x,y, (Opt. gray value) (Opt. /C /Z) - add text to <BMj> starting at (x,y). If /G is specified, then (x,y) is specified from the Graphpen when the tip is pressed. The default text density is black. Note that the text display overwrites data in <BMj> so that <BMj> will probably not be usable for further processing.

2.6.5 GRID

<BMj>_GRID, (Opt. gray value),(Opt. size in microns) (Opt. /C) - write a (default black) grid with squares corresponding to 10x10 microns (default) at the current

axiomat zoom setting. If /N is specified, then make the squares 10x10 pixels.

2.6.6 WHITENOISE

<BMj>_WHITENOISE, std dev. density, mean dens. (Opt. /C, /U) - Generate Gaussian noise in <BMj>.

2.7 Unary image point operators

Unary image operators take one input image and map it to one output image using a point operation. The /U switch for defining the computing window using the F&S is active for all operators of this group except COPY and COMPLEMENT.

2.7.1 COPY

<BMj>_COPY,<BMi> - copy the entire image. Note that the /U switch is inoperative.

2.7.2 COMPLEMENT

<BMj>_COMPLEMENT,<BMi> - complement the entire image by computing g=255-g. Note that the /U switch is inoperative.

2.7.3 CONTRAST

<BMj>_CONTRAST,<BMi> - Contrast stretch <BMi> according to the last histogram computed (using the HISTOGRAM operator), as follows: If hi and lo are the maximum and minimum values of data in the range of the histogram, g the gray value at a pixel, then stretch the image according to the linear interpolation:

$$g' = ((hi-lo)/255)*(g-lo).$$

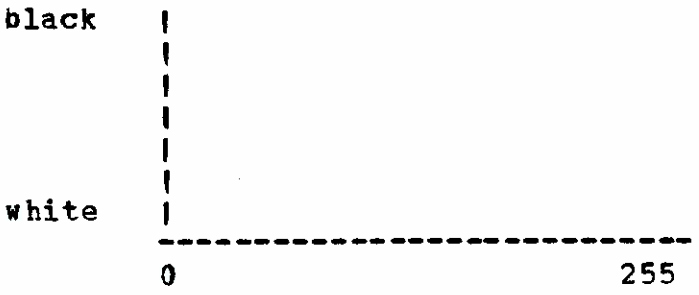
Note that hi is stored in QRD and lo in QRC when the HISTOGRAM function is used.

2.7.4 DEFCONTRAST*

<BMj>_DEFCONTRAST,(Opt. <BMi>/U) - define the contrast function (CF) in <BMj> (storing it in the histogram array for use by DEFCONTRAST and FNCONTRAST). The initial contrast function is set to:

$$contrast(g) = g.$$

The GraphPen is used to input the contrast function where the coordinate system is as follows:



The class keys constitute a menu of commands available to the operator:

Class key	function
0	print current contrast function on LPT:.
1	<BMj>_apply CF to <BMi>/U.
2	<BMj>_apply CF to <BMi>.
3	<BMj>_redisplay current contrast function.
4	restart DEFCONTRAST and print help message.
11	return to BMON2 saving last defined CF.

2.7.5 FNCONTRAST*

<BMj>_FNCONTRAST,<BMi> - contrast stretch pixel data in <BMi> by the contrast function stored in the histogram and previously defined by DEFCONTRAST. The data is mapped as follows:

BMj(x,y)=h(BMi(x,y)).

2.7.6 SCALE

<BMj>_SCALE,<BMi>, n, d, b (Opt. /C) - scale <BMi> by ((n*<BMi>/d)+b) Mod 2047. The result is then clipped to [0:255]. Negative b may be expressed as 2's complement numbers, (i.e. -1=4095, -2=4094, etc.).

2.7.7 SLICE

<BMj>_<BMi1>,SLICE,(Cpt. <BMi2>), dmin, dmax (opt. /C) - Threshold slice an input image onto an output image. The operator may be used with SEGBND to extract connected components since the data which is passed may come from another source (<BMi2>). It uses the following algorithm:

```
BMj(x,y)=If (dmin leq Bmi1(x,y) leq dmax)
        Then
        If BMi2 specified
                Then BMi2(x,y)
                Else BMi1(x,y)
        Else 0;
```

The SLICE operator may be used to extract pixels from <BMi2> based on the results of the thresholded data in <BMi1>.

The /C complements the data after the slice function is evaluated.

2.7.8 SHIFT

<BMj>_SHIFT, <BMi>, delta x, delta y (Opt. /C) - Shift pixels from <BMi> by (delta x, delta y). Pixels shifted outside of (0:255,0:255) are ignored. Shifts in a negative direction must be specified as 2's complement numbers.

2.7.9 ROTATE*

<BMj>_ROTATE, <BMi>, angle in degrees, xoffset, yoffset (Opt /X /Y) - rotate image pixels from <BMi> the specified angle about the center of of the computing window (127,127) if no /U is specified. After the rotation transformation is performed the pixels may be translated by an amount (xoffset,yoffset) (defaults to (0,0)). The /X and /Y switches make xoffset and yoffset negative if specified. The rotation transformation is as follows:

Let [xl:xh, yl:yh] be the computing window. The center of the window is:

```
xc = (xh+xl)/2,
yc = (yh+yl)/2.
```

Then convert the LCS coordinates to the Cartesian

coordinate system,

```
        mx = x - xc,
        my = yc - y.
```

Then for angle t,

```
        xp = (xc - (mx*Cos(t) + my*Sin(t))) + xoffset,
        yp = (yc  (-mx*Sin(t) + my*Cos(t))) + yoffset.
```

Then the following transformation is applied:

```
For y1_1 step 1 until 256 Do
  For x1_1 step 1 until 256 Do
        Begin "transform by testing all points in BMj"
        compute (xp,yp);
        If (xl leq xp leq xh) and (yl leq yp leq yh)
                Then Gj(x,y)_Gi(xp,yp);
        End "transform by testing all points in BMj";
```

2.8 Neighborhood unary image operators

The following neighborhood definition is used in many of the following operations where I8 is the central pixel of reference:

```
I3 I2 I1
I4 I8 I0
I5 I6 I7.
```

This notation is used because of the ease of specifying the direction of the pixel relative to the central pixel. The angle formed between I8 and Ij is j*(45 degrees).

None of the operators in this group should use the same BM for both the source neighborhood and the output BM as they would propagate changes in a non-uniform manner due to the way neighbhborhoods are simulated using a triple-line buffer.

The /U switch may be used for the operators of this group to indicate the computing window using the F&S.

2.8.1 ZOOM*

<BMj>_ZOOM,<BMi>, (magnification (m=N/D) specified as N,D) (Opt. /A to use 512x512 specified by BMj as the output image) - Zoom <BMi> by the specified magnifaction m by repeating pixels if m > 1.0, or sampling pixels if m < 1.0. The resulting output image is centered in <BMj>. If the output image is magnified so that it is larger than 256x256, then only the upper left hand corner of the magnified image is displayed such that it fits into a 256x256 pixel window. If /A is specified, then map it to a 512x512 instead of a 256x256 image.

2.8.2 AVG8

<BMj>_AVG8,<BMi> (Opt. /C) - compute the eight neighbor average by:

q = (I0+I1+I2+I3+I4+I5+I6+I7+I8)/9.

2.8.3 AVGN*

<BMj>_AVGN,<BMi>,<Opt. N> - Compute the NxN neighbor average of <BMi> as follows (for N odd, default N=3):

$$g'(x,y) = (1/(N*N))\ \mathrm{SUM}\ (i=[x-N/2:x+N/2], j=[y-N/2:y+N/2]\ \mathrm{of}\ g(i,j)).$$

2.8.4 MIDPOINT*

<BMj>_MIDPOINT,<BMi> - compute the midpoint gray value of each 3x3 neighborhood by finding the maximum and minimum gray values and taking the average of the two.

$$g'(x,y) = [MAX(gi(x,y)) + MIN(gi(x,y))]/2.$$

2.8.5 MEDIAN*

<BMj>_MEDIAN,<BMi> - compute the median gray value of each 3x3 neighborhood by sorting the gray values in the neighborhood and taking the middle value.

2.8.6 LAPLACIAN

<BMj>_LAPLACIAN,<BMi> (Opt. /C) - compute the 8 neighbor Laplacian by:

$$g = |8*I8 - (I0+I1+I1+I3+I4+I5+I6+I7)|.$$

2.8.7 GRAD4

<BMj>_GRAD4,<BMi>, (Opt numerator, denominator scaling) (Opt /D to store direction values instead of the magnitude, Dx<==>1, Dy<==>3, D45 or D225<==>2, D135 or D325<==>4) - Compute the 4 major axis gradient. The four differences are computed using a filter emphasising the central points discussed in [5].

$$D0 = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1, \end{matrix}$$

$$D45 = \begin{matrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0, \end{matrix}$$

$$D90 = \begin{matrix} -1 & 0 & 1 \\ 2 & 0 & -2 \\ -1 & 0 & 1, \end{matrix}$$

$$D135 = \begin{matrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2. \end{matrix}$$

2.8.8 EDGE

<BMj>_EDGE, <BMi>, gradientthreshold, Laplacian threshold - compute GRAD4 and the LAPLACIAN for each (x,y) and test whether |grad| > gradient threshold AND |Laplacian| > Laplacian threshold. If the boolean is true then store BMi(x,y) else 0; The EDGE operator is discussed in [5].

2.8.9 GRADN*

<BMj>_GRADN,<BMi>,<Opt. N> - Compute the NxN neighbor gradient of <BMi> as follows (for N odd, default N=3):

$$g' = GNxN(x,y) = MAX(|Dh|,|Dy|),$$

where

Dh = (SUM of the top N/2 rows) - (SUM of the bottom N/2 rows),

Dv = (SUM of the left N/2 columns) - (SUM of the right N/2 columns).

2.8.10 MTV*

<BMj>_MTV,<BMi> - compute the minimum total variation (MTV) from the minimum of the horizontal and vertical differences as follows:

$$HTV = |I13-I12|+|I12-I11| + |I14-I18|+|I18-I10| + |I15-I16|+|I16-I17|,$$

$$VTV = |I13-I14|+|I14-I15| + |I12-I18|+|I18-I16| + |I11-I10|+|I10-I17|,$$

$$g = Min(HTV,VTV).$$

This function was developed by [11] and is a type of edge detector.

2.8.11 VARIANCE*

<BMj>_VARIANCE,<BMi> - Compute the variance of a 3x3 neighborhood at each pixel of <BMi> and store it in <BMj>.

2.8.12 GRAD8*

<BMj>_GRAD8,<BMi>, (Opt. numerator, denominator scaling) (Opt /D to store direction values instead of the magnitude, D0=1, D45=2, D90=3, D135=4, D180=5, D225=6, D270=7, D325=8) (Opt /C) - Compute the 8-neighbor gradient used by Kirsch.

$$Fi = \sum_{j=0}^{7} (5*(I(j)+I(j+1)+I(j+2)) - (3*(I(j+3)+...+I(j+7))$$

Then,
$g = Max(F0, F1, ..., F7)$.

2.8.13 FILTER*

<BMj>_FILTER, <BMi>, (Opt. numerator n, denominator d (scale factor n/d), b (offset)) (Opt /C) - Requests from the command decoder a list of up to 256 direction list points with their associated scale factors (a null entry terminates the list). A direction list entry is a 4-tuple consisting of

[Di, Ni, XRELi, YRELi].

The switches /N, /X, and /Y indicate that the respective entries are negative.

The resultant pixel is computed as:

$g'(x,y) =$ SUM [over all i of $g(x+XRELi, y+YRELi)*(Ni/Di)$].

For example, to apply the following filter:

$$(1/11)\begin{vmatrix} 10001 \\ 01010 \\ 01110 \\ 01010 \\ 10001 \end{vmatrix},$$

specify the following direction list sequence:

```
*1,11,2,2/Y
*1,11,2/X,2/Y

*1,11,1,1/Y
*1,11,1/X,1/Y

*1,11,1/X,0
*1,11,0,0
*1,11,1,0

*1,11,1/X,1
*1,11,1,1

*1,11,2/X,2
*1,11,2,2
*
```

2.8.14 FILLPINHOLES

<BMj>_FILLPINHOLES,<BMi>, density difference d (Opt. /C) - Fill pinholes in the image as follows:

$$g' = \text{If } |g - AVG8| > d \text{ Then } AVG8 \text{ Else } g;$$

2.8.15 CIRCLE

<BMj>_CIRCLE, <BMi>, xcenter, ycenter, radius (Opt./C) - Extract a circular region from <BMi> and insert it into <BMj>.

2.8.16 RECTANGLE

<BMj>_RECTANGLE, <BMi> (Opt /C) - Extract a retangular region from <BMi> and insert it into <BMj>. Additional args are requested with another command decoder call:

```
*xcenter, ycenter, xside, yside
```

2.8.17 PROPAGATE*

<BMj>_PROPAGATE,<BMi> (Opt /U) - propagate 1's in a copy of <BMi> into gray values > 1 in <BMj>. Pixels are considered 8-neighbor connected. The algorithm assumes a preprocessed image which has regions not intended to be propagated together to be separated by 8-neighbor

0's. The ISOLATE operator is one mechanism for forming such an image.

The algorithm converges fairly rapidly because of the above assumption that regions of 1's belonging to separate components are disjoint. The algorithm extends lines from the right to the left, left to the right, top to the bottom and bottom to the top simultaniously to optimize propagation.

```
While (any propagations from previous pass) Do
        Begin "prop a pass"
        For i_2:255 Do
           For j_2:255 Do
                Begin "prop pixel"
                If g(i,j) > 1
                        Then Prop(i,j);
                If g(i,256-j) > 1
                        Then Prop(i,256-j);
                If g(256-i,j) > 1
                        Then Prop(256-i,j);
                If g(256-i,256-j) > 1
                        Then Prop(256-i,256-j);
                End "prop pixel";
        End "prop a pass";
```

2.8.18 PROP2*

<BMj>_PROP2,<BMi>, (Opt. number of passes, 384 default) - first copy BMi into BMj, then propagate 1's from 8-neighbor CC pixels > 1 in BMj in the number of passes specified. The tesselation is successive rasters.

2.8.19 RUNFILTER*

<Opt. BMj>_RUNFILTER,<BMi>,(Opt. delta),(Opt. Rmin,Rmax) - compute the run length histogram of BMi into the BMON2 histogram array ignoring zero runs. A run is started/terminated if there is a horizontal difference between the previous pixel and the current pixel:

$$|G(x,y) - G(x-1,y)| > \text{delta}.$$

The default for delta is 1. In addition, runs outside of the range of run lengths [Rmin:Rmax] are ignored. The default range of runs is [1:255]. The /C switch complements the sense of the run length sizing. The /L switch prints the run length distribution on the lineprinter. If <BMj> is specified, then pixels of a run just terminated cf length q are set to value q. If

/M is also specified, then the pixels of the run are set to black (255). Pixels outside of the run (in either case) are set to 0.

2.8.20 NGHSE*

<BMj>_NGHSE,<BMi>,/X for max else min) - compute the 8-neighborhood minimum (maximum if /X) of BMi and store it into BMj. It has been pointed out that the local max corresponds to a gray level expand and the local min to a gray level shrink (where connectivity is not taken into account) [22].

2.8.21 NOTCH*

<BMj>_NOTCH,<BMi>,(Opt. n),(Opt. B)(Opt. /A) - compute the nxn notch filter of <BMi> and store it in <BMj>. The parameter n (defaulted to 32) is the window size and is an even integer between 2 and 62. The basic notch filter algorithm was suggested by Jack Sklansky. An offset, B, may be added to the filter value of may be computed in two passes of the algorithm automatically using the /A switch. The filter is computed as follows:

$$Gj(x,y) = Gi(x,y) - AVGnxn,i(x,y) + B.$$

The result is then clipped to [0:255] before being stored in BMj.

Let $n1 = (n/2) - 1$.

$$AVGnxn,i(c,r) = \frac{1}{n*n} * \sum_{x=c-n1:c+n/2} \sum_{y=r-n1:r+n/2} Gi'(x,y).$$

In order to handle the edge conditions, the image is reflected about the boundaries as follows.

$$Gi'(x,y) = Gi(R(x),R(y)).$$

Where R(.) is the reflection function.

```
R(p) = If p < 0
         Then -p
         Else If p > 255
                Then 255-(p-255)
                Else p.
```

If the /A switch is specified, then automatic computation of B is performed in two passes as follows. During the first pass B is computed. The finial output image BMj is computed during the second pass.

$$B = - \min_{\text{All } x,y} (Gi(x,y) - AVGnxn,i(x,y)).$$

In order to increase the efficiency, AVGnxn,i is computed iteratively as follows:

$$AVGnxn,i(0,r) = \frac{1}{n*n} * \sum_{y=r-n1:r+n/2} COLSUM(0)$$

and,

$$AVGnxn,i(c,r) = AVGnxn(c-1,r) - COLSUM(c-n/2,r) + COLSUM(c+n/2,r).$$

$$COLSUM(c) = \sum_{y=r-n1:r+n/2} G'(c,y).$$

2.8.22 FILGAP*

<BMj>_FILGAP,<BMi>,Threshold t - Fill gaps in narrow dark lines. The filter looks for a light gap of 1 to 2 pixels in length in a narrow dark line of 1 to 2 pixels wide. The filter consists of a 5x5 neighborhood in 8 different orientations (actually 16 but 8 are symmetric).
Let Fi be one of the eight filters. Then the Gj(x,y) is computed as:

$$Gj(x,y) = \begin{cases} \text{If } \max\{Fi(x,y)\} > t \\ \quad \text{Then } \max(25\text{-neighborhood of } (x,y)) \\ \quad \text{Else } Gi(x,y); \end{cases}$$

The eight filters are:

a

0	0	+4	0	0
0	0	+2	0	0
-2	-1	-6	-1	-2
0	0	+2	0	0
0	0	+4	0	0

b

0	0	+2	+2	0
-1	0	+1	+1	0
-1	-1	-6	-1	-1
0	+1	+1	0	-1
0	+2	+2	0	0

c

-2	0	0	0	+4
0	-1	0	+2	0
0	0	-6	0	0
0	+2	0	-1	0
+4	0	0	0	-2

```
          0 +2 +2  0  0
          0 +1 +1  0 -1
d        -1 -1 -6 -1 -1
         -1  0 +1 +1  0
          0  0 +2 +2  0
```

The filters a', b', c' and d' are filters a, b, c, and d rotated 90 degrees.

```
          0  0 -2  0  0
          0  0 -1  0  0
a'       +4 +2 -6 +2 +4
          0  0 -1  0  0
          0  0 -2  0  0

          0  0 -1 -1  0
         +2 +1 -1  0  0
b'       +2 +1 -6 +1 +2
          0  0 -1 +1 +2
          0 -1 -1  0  0

         +4  0  0  0 -2
          0 +2  0 -1  0
c'        0  0 -6  0  0
          0 -1  0 +2  0
         -2  0  0  0 +4

          0 -1 -1  0  0
          0  0 -1 +1 +2
d'       +2 +1 -6 +1 +2
         +2 +1 -1  0  0
          0  0 -1 -1  0
```

2.9 Point binary operators

The point binary operators operate on the corresponding pixel at each (x,y) point in the computing window for two source images. The term binary is being employed here in the algebraic rather than gray scale sense. The results of operations are clipped to 255 if the value is greater than 255 and 0 if the value is less than 0.

If the <BMi2> argument for binary operators (ADD) to (DIFF) is a scalar value rather than a BM specification, then the scalar value is used rather than pixel data from in the BM. The /U switch used to define the computing window over <BMi1> is active for all of the operators in this group.

2.9.1 ADD

<BMj>_<BMi1>, ADD, <BMi2> (Opt. /C) - clips values > 255 to 255.

2.9.2 SUB

<BMj>_<BMi1>, SUB, <BMi2> (Opt. /C) - clips values < 0 to 0.

2.9.3 MUL

<BMj>_<BMi1>, MUL, <BMi2> (Opt. /C) - clips values > 255 to 255.

2.9.4 DIV

<BMj>_<BMi1>, DIV, <BMi2> (Opt. /C) - Computes <BMi1>/<BMi2>.

2.9.5 AND

<BMj>_<BMi1>, AND, <BMi2> (Opt. /C) - bitwise AND.

2.9.6 OR

<BMj>_<BMi1>, OR, <BMi2> (Opt. /C) - bitwise inclusive OR.

2.9.7 MAX

<BMj>_<BMi1>, MAX, <BMi2> (Opt. /C)

2.9.8 MIN

<BMj>_<BMi1>, MIN, <BMi2> (Opt. /C)

2.9.9 DIFF

<BMj>_<BMi1>, DIFF, <BMi2>, threshold value (Opt. /C) - Set <BMj> to the difference between the two input images if the difference is > the threshold value.

2.9.10 ISOLATE*

<BMj>_ISOLATE,<BMi1>,<BMi2>,(threshold t) (Opt /T for trace in BMj') - isolate 1 regions associated with component regions with values > 1 by painting 0's to isolate the regions. Such a resulting image may then be used to propagate the components into the surrounding 1's regions. <BMi1> is the image to be isolated and the isolated resultant image is stored in <BMj>. <BMi2> is the gray scale image associated with the component labeled image <BMi1>.

The isolation is made based on a decison process which looks for a lightened texture region between pairs of connected component (CC) regions. This would correspond to the possible diffraction region between two lightly touching objects. Unfortunately, if there is very tight contact between two objects such a lightened texture region may be difficult to find or may not exist, being a darkened texture region due to overlapping of the objects' edges.

The <BMi1> coding scheme

pixel value	semantics of the pixel
0	- background pixels
1	- propagatable pixels
2:255	- connected component regions pixels

The algorithm works as follows: the minimum rectangles are found for all connected component regions. Then all components are analyzed two at a time. The line connecting the centers of each pair of CC's is computed and the <BMi1> and <BMi2> data are analyzed along this

line and nearby regions. First, all possible pairs of i and j are computed:
A CC pair [i,j] is rejected if:

(1) There is a 0 valued 3x3 neighborhood on the line [i,j].
or
(2) There is a pixel of value k on [i,j] such that (k neq 1) and (k neq i) and (k neq j).
or
(3) The mean diameter of the enclosing rectangle of CCi and CCj is < 1.5 times the number of external 1's on [i,j] (i.e. those pixels of value 1 between the last i and first j valued pixels seen in moving from i to j in <BMi1>).

If the Euclidian distance of the 1's on the line [i,j] is < 10% of the length of [i,j] then the split region is assumed to be half way in the 1's region.

After the set of possible [i,j] are computed from <BMi1> the corresponding regions of <BMi2> (the corresponding gray value image) are analyzed for lightened texture regions by running a set of variable length perpendicular line filters along the line looking in the 1's region of the line for a lightend region. The line [i,j] is at angle theta where:

theta = Arctan((Yj-Yi)/(Xj-Xi)).

The perpendicular angle to [i,j] is eta.

eta = theta + 90.

A heuristic feature Hk (at point k on [i,j]) of these filters is computed for each k. At each point k along the line [i,j] at angle eta, the symmetric line filter of length r=3,5,7,9,11 pixels is computed. The average density of each line segment in <BMi2> Davg is computed for those points of the filter having 1's in the corresponding pixel in <BMi1>.

Hrk=Davg(r,k,eta).

Then, a composite heuristic function Hk is computed from the Krk.

Hk = Log(H3k*H5k*H7k*H9k*H11k).

Then, the global heuristic is computed:

(Hm | Hk ^ k=m ^ Hk < Hp ^ p neq m).

Then for Kmin=m, Hm is a minimum density filter heuristic and point Kmin is probably the index of the lightened texture region. The relative minimum value is

then computed as follows.

$$Hmin = Hm.$$

$$Havg = \frac{\text{SUM Hk for k an external 1 of [i,j]}}{\text{number of external 1's of [i,j] in <BMi1>}}.$$

Then,

$$Hrelmin = 100*(Havg-Hmin)/Havg.$$

If Hrelmin > t
Then split propagatable region belonging to both CC i and j at angle eta point Kmin;

Having decided to split the propagatable region of CCs' i and j, a correction is made to eta to better fit the angle at which the split is to take place. The correction is performed as follows by finding the minimum density line at angle eta' from point Kmin.

```
Dmin_255;
For alpha_eta-22 Step 5 Until eta+22 Do
        If (d_Davg(10 pixels, Kmin, alpha) < Dmin)
                Then eta'_eta, Dmin_d;
```

The 1's region in <BMj> is now split by painting 3x3 zero neighborhoods (to guarentee non-adjacency - although a 2x2 would suffice) along the line passing through point Kmin on [i,j] at angle eta'. The painted line extends on either side of [i,j] until it reaches a 3x3 neighborhood with all zeros or all pixels > 1 (i.e. a CC), or it reachecs the end of the BM frame.

2.9.11 SHADE*

<BMj>_SHADE,<BMi1>,(Opt. BMi2> - compute the point for point shade corrected image of BMi1 and save it in BMj. The algorithm requires two other images which are either specified in both halves of BMi2 or defaulted to BM3 if BMi2 is not specified. These two additional images are called the neutral density images Ind1 and Ind2. Ind(1 and Ind2 are acquired previous to doing the shade correction with Ind1 being the darker of the two filtered images. Ind1 is stored in BMi2L (BM3L) and Ind2 in BMi2H (BM3H). These images are acquired after setting up the QMT control to "manual" and the sensitivity to between 0.3 and 0.4 to get a data image with good dynamic range. All images are acquired from the same position on the QMT camera. The algorithm computes the shade corrected image as follows. Let Gi1, Gnd1, Gnd2, Gj be the gray values of the corresponding images.

Let A=256.0/Log(256.0)

Then,

$$Iavg(k) = \frac{1}{A*65536} * \sum_{All\ x,y} (A*Log(Gndk(x,y)+1)).$$

Let K = Iavg1-Iavg2.

Then,

$$Gj(x,y) = (Log^{-1}(K*\frac{(Gi1(x,y)-Gnd1(x,y))}{(Gnd2(x,y)-Gnd1(x,y))} + Iavg1) -1.$$

2.10 Statistical display operators

The buffer memories may also be used for plotting 2-D graphics. Several statistical measures operators are computed and generate displays.

2.10.1 HISTOGRAM

(opt. <BMj>) _ HIST, <BMi>, (Opt. parameter value d)
(Opt./X or /Y for col or row grayscale projections)
(Opt. /R grayscale run length/line)
(Opt. /Q to scale the display to absolute value of 1000.)
(Opt. /L to print histogram), (Opt. 2nd, 3rd values are the optional size of the histogram range to be displayed or printed).
- The histogram is computed for the specified BMi with parameter d and the maximum and minimum range of gray values and their associated sums are printed. If BMj is not specified then the histogram is not displayed. If it is displayed, then a scale is also displayed at the bottom of the histogram with short scale lines every 10 points and larger ones every 50.

For /R, d is used to
specify the magnitude difference between the start of any run gray value Gi and successive points in the run Gj such that [Gj | Gj in Run Gi and |Gj-Gi|<d]. For the grayscale projections, (including /X and /Y) d (default 1) is the spatial sampling distance.

The parameters [lower range, upper range, minimum gray value, maximum gray value] are stored in Q-registers [A,B,C,D].

Normally, the background of the histogram is displayed at gray value 50 and the line drawings at 150. The /J switch causes the background to be set to 0 and the line drawing to 70.

2.10.2 SHOWHISTOGRAM

<BMj>_SHOWHISTOGRAM - show the histogram already in histogram COMMON storage array IH[1:512] in EAE double precision format.The /L switch lists the histogram on the teletype.

2.10.3 SMOOTHISTOGRAM*

<Opt. BMj>_SMOOTHISTOGRAM, <Opt. Window width L>,<Opt. Noise immunity DELTA>, <Opt. Number of iterations to smooth the histogram> - smooth the histogram in memory using disaster analysis discussed in [13]. The disaster analysis computes for all x in [0:255] a noise figure

$$N(x) = |H(x) - Havg(x,L)|.$$

Where: $Havg(x,L) = (1/L)\,SUM[i=(x-L/2):(x+L/2)\ \text{of}\ H(i)]$.

If $N(x) > DELTA$
Then $H(x) <== Havg'(x,L)$;

Where: $Havg'(x,L) = ((L*Havg(x,L)) - H(x))/(L-1)$.

The end points at [-L/2:-1] and [256:255+L/2] are duplications of the end points 0 and 255 respectively. The /L switch causes non-zero points to be printed with the H(x), |N(x)/Total H|*100% values. DELTA is specified as multiples of 0.1% (i.e. 10 =1.0%) of the total area under the H(x) curve. The default values are L=10, DELTA=5 (0.5%), number of iterations=1. If <BMj> is specified, then the smoothed histogram is displayed in <BMj>.

2.10.4 PLOT2D*

<BMj>_PLOT2D,<BMi1>,<BMi2> - compute the scatter plot of the data in the two input images where the data in <BMi1> is plotted in the x direction and data in <BMi2> in the y direction in <BMj>. The background is set to 75 to facilitate photographing it. If /I is specified, then the background is set to 0 and each occurrence of a scatter point causes that pixel to be incremented to a maximum of 255.

2.11 Line drawing operators

Two line drawing input devices are available in the RTPP system. One is the GraphPen spark tablet and the other is the so called "mouse" which is a modified 2-D joystick with a push-button in the center. Both the GraphPen and the mouse are used in EXTRACT and BDEDIT while only the GraphPen is used in GRAPHPEN and DRW512. The position of the drawing device is contstantly tracked using a QMT cursor displayed under control of the "scale" intensity control. Note that the left edge of the cursor points to the actual position. Pressing the pen tip or the mouse button causes line data to be taken and written into the memory.

2.11.1 GRAPHPEN

<Opt. BMj>_GRAPHPEN, (Opt. gray value), (Opt. brush size) (Opt. /B /C) - read the GraphPen (x,y) coordinates relative to the F&S and put them into (DISP1,DISP2) displays. If <BMj> is specified, then position the F&S over <BMj>. When the pen tip is pressed, if a BM is specified then also write a black (255) pixel at that location. The F&S will be used to position the specified BM. If POST,BMj/B/M was specified for BMj then treat the <BMj> as a QMT live frame mask data <BMj>, thus permitting editing of mask type data.

The class keys are used as a menu for selecting editing functions.

CLASS KEY	FUNCTION
0	While pressed complements the ink density (white=0 default).
1	While pressed causes the "ink" to be "brushed" on using a nxn neighborhood at the pen point. The /B switch reverses the sense of this key. The value n=(2*FBW5)+1.
2	Causes a 6 neighbor gradient to be displayed in the right QMT display, Gr=MAX(|(I0+I1+I2)-(I4+I5+I6)|, |(I2+I3+I4)-(I6+I7+I0)|).
3	Causes the Laplacian to be displayed, computed as: Lp=|8*I8-(I0+I1+I2+I3+I4+I5+I6+I7)|.
4	Requests a new brush size from the keypad.
5	Requests a new gray value from the keypad.
6	Clears <BMj> by zeroing it.
11	Or typing ^O returns control to the BMON2 monitor.

When the pen is not writing, it will display the grayscale value of the pixel pointed to by the cursor in the right QMT display. It is possible to draw into the opposite half of a BM that is being displayed by drawing into the half being displayed as a mask. This permits editing the data in the memory by writing zeros and writing lines (black). Such a mask may be used with QMT data acquisition. The POST must be setup as follows:

```
*POST,BMi/B/M
*BMiH_GRAPHPEN.
```

The mask generated is the logical AND of the F&S and the BMiH data (>127).

2.11.2 EXTRACT*

<BMj>_EXTRACT,<BMi> (Opt. /Y /N /I) - Position the F&S which is expanded to 256x256 pixels around <BMi>. The EXTRACT command sets the QMT display to the equivalent

of "POST/B/M,<BMi>", "POST,<BMj>". It then enables the GraphPen editor for writing into <BMi'>. The editor has the capability of extracting data from <BMi> under a boundary drawn in <BMi'> and storing the data into <BMj>. GraphPen data is entered into <BMi'> only when the pen tip is pressed. Otherwise, the pen may be used to direct the QMT display cursor over <BMi> where the left edge of the cursor is the actual pointer.

Boundary data points are displayed in the mask BM, <BMi'>, as well as being stored in a circular linked list which may be edited. The drawing is available to other BMON2 operators through the list pointer in QRZ. The list consists of the actual (x,y) coordinates. The /I switch causes the list freestore to be re-initialized when EXTRACT is first entered. This permits saving the freestore between operations.

A "mouse" joystick is available as an alternative input instead of the GraphPen for x-y cursor input to the EXTRACT program. The mouse is invoked by specifying the /Y switch. Pressing the mouse data active button is equivalent to pressing the GraphPen tip or FBW12[11] "execute" key.

A region extraction feature is available. The latter, in traceing a boundary, generates a run length map (RLM) [14] which is translated to a list of run lengths indexed by row value for rapid inside/outside predicate evaluation. This method is similar to that of [9], except that he stores the sorted boundary pixels instead of run lengths and searches them differently.

When a region is extracted, the minimum enclosing rectangle, its area A, perimeter P are computed in microns (pixels if /N) and (P*P)/A, and density/A is also computed. It is critical for the validity of the RLM algorithm that the boundary be drawn clockwise and starting at the topmost pixel of the boundary. This is guaranteed by the algorithm by searching the boundary linked list to find the topmost point of the list.

The CLASS keys are used as a menu for selecting editing functions.

CLASS KEY	function
0	Toggle <BMi> POST/UNPOST status.
1	Perform a GET,<BMi> after asking: "ARE YOU SURE(Y/N):".
2	Erase boundary from <BMi'> and boundary list backwards.
3	--Not used--
4	Zero <BMj> (sink extraction memory).
5	Zero <BMi'>, reset boundary.
6	Position the cursor at the current "last" point in the line.
7	Rotate the current "last" point in the line clockwise and set the cursor to this point.
8	Rotate the current "last" point in the line counter-clockwise and set the cursor to this point.
9	<BMj>_Extract,0 pixels using <BMi'> RLM and store them in <BMj>. It also computes and prints the minimum window, area, perimeter edge points, density/area, and perimeter squared/ area of the extracted data.
10	<BMj>_Extract,<BMi> pixels using <BMi'> RLM and store them in <BMj>. It also computes and prints the minimum window, area, perimeter edge points, density/area, and perimeter squared/ area of the extracted data.
11	Exit to BMON2.

2.11.3 DRW512*

DRW512* - using the graphpen, draw a line in a 512x512 image consisting of BM0, 1, 2 and 3. Various options are activated by the CLASS keys.

Class key	Function
0	Erase all BMs 0 to 3.
1	Write white instead of black.
3	UNPOST/A while down.
11	Exit back to BMON2 (^O returns as well).

2.11.4 BDEDIT*

(Output file dev default DSK:) <BMj>_BDEDIT*, (If not /G then input boundary data file ".DA"), (output filespec prefix p with default "BD"), (/G /M /L /1), (Opt. BMi) - Edit boundaries from the input boundary data file into a set of single boundary data files numbered sequentially from the file name composed as: p&CVS(n)&".DA". If BMi is specified then the boundary is overlayed on a copy of BMi into BMj to give the user a better reference for editing. Otherwise, BMj is zeroed before editing each boundary. Note: only boundaries less than 2046 points in length can be edited. The /G switch permits the user to define a boundary from the graphpen (with CLASS keys 7, 8, 9, 10, 11 activated). If /G is specified then the following switches are activated: /M (for using the mouse instead of the GraphPen), /L to label up to 15 pairs of points from the keypad, and /1 to draw and mark only 1 file before returning to BMON2. After the boundary is defined or the boundary marking sequence is entered, CLASS keys 0, 1, 2, 3, 4, 5, and 6 are activated. It is possible to go back and forth between the marking and drawing parts of the function. The algorithm is given as follows:

```
[1] If not /G
        Then open the specified boundary data file for
            editing if it exists
        Else Goto [3];
[2] Mark editor.
   [2.1] If BMi exists
                Then BMj_COPY,BMi
                Else BMj_ZERO;
   [2.2] If not /G
        Then For each boundary b in the input file Do
             read next boundary header==>H, and
             boundary (x,y) data b==>list==> BMj
        Else draw boundary in list ==> BMj;
   [2.3] Edit according to CLASS keys:
        0 - Reject, Goto [2.5];
        1 - Rotate boundary pointer CW;
        2 - Rotate boundary pointer CCW;
        3 - Mark 1st label (index or KPD if /G);
        4 - Mark 2nd label (index or KPD if /G);
        5 - Accept boundary, Goto [2.4];
        6 - Edit existing boundary at current cursor.
```

```
   [2.4] Accept boundary.
        [2.4.1] Open output file on output device with
                 file name: p&CVS(n)&".DA"; n_n+1;
        [2.4.2] Dump header H;
        [2.4.3] Dump boundary list;
        [2.4.4] Dump eof:
                -1,0, index 1, index 2
                -1,2, index, mark # 1, mark #2
                  .
                  .
                  .
                -1,-1
        [2.4.5] Close the file;
   [2.5] Continue;
[3] Edit boundary.
        7 - Point to last point in "mark" mode.
        8 - Restart drawn boundary using GraphPen.
        9 - Delete drawn boundary backwards.
        10 - Close drawn boundary and start
                mark editor.
        11 - Return to BMON2.
```

2.12 Segmentation operators

Two types of segmentation operations are performed which result in similar types of connected component images but use different algorithms. The first SEGBND forms connected components by tracing the boundaries of objects in a sliced input image. The other algorithm, SEG2PS keeps track of components to be merged and merges them on a 2nd pass.

2.12.1 SEGBND*

<BMj>_SEGBND,<BMi>,<Opt. BMi2>,lower boundary size, upper boundary size, (Opt. slice threshold) (Opt /H /T /U /C /N /I /B /M /X /G) (Opt feature selection switch /i where i is 0 to 9) - Segment the thresholded image <BMi> into a connected component image <BMj> and a boundary trace image <BMj'> is the /T switch is specified. Isolated pixels and pixels outside of the specified segment feature size (default 0:2047) are labeled in the output image as background value (zero). The components are labeled as pixels having component values of 2 through 255 (253 components). The input image can have background pixels set to zero (i.e. be thresholded) or a threshold may be optionally specified.

If the /G switch is specified, after the binary image is generated, point the GraphPen to the left of the object you wish to segment and press the pen to start the segmentation of that object. After it is segmented, this process it repeated until either ^O or the execute key on the control desk is pressed.
The /H switch causes holes inside segments to be filled. The /T switch traces the boundaries as black (255) and filled holes as gray (70) in <BMj'>.

The /C reverses the sense of the sizing criteria. The /N switch causes the sizing and printout to be in pixels rather than microns. If features with a value greater than 2047 are to be accepted, then set the upper sizing limit to 2047. The /i feature selection switch (default to /0 is none specified) is used to select the feature to be used in the sizing decision. The switches are as follows:

/i	feature
/0	area
/1	perimeter
/2	average density/pixel
/3	horizontal size of enclosing rectangle
/4	vertical size of enclosing rectangle
/5	area of filled regions
/6	perimeter squared/area
/7	density
/8	number of boundary points
/9	number of boundary edges intersecting the BM edge

The /M switch causes the specified size parameters (if any) to be ignored and a set of ranges and feature types to be specified via the command decoder input (with a null specification terminating the list). These are used as a conjunction filter during segmentation. The command decoder input is of the format:

*lower size, upper size, (Opt. n, d)/i

which defines the range as:

[(n/d)*lower size : (n/d)*upper size].

For example, to size accoring to area [Al:Au] and perimeter [Pl:Pu] the following would be specified:

```
*Al,Au/0
*Pl,Pu/1
*
```

establishes the conjunctive filter

[Al leq area leq Au] and [Pl leq perimeter leq Pu].

The /U segments only those parts of <BMi> inside of the frame and scale window. The histogram array is used as a scratch area and is destroyed.

This algorithm is derived from one given in [7]. It first creates a (0,1) binary image in <BMj> from the image in <BMi>. <BMi> is assumed to be sliced such that background pixels are set to 0. Optionally the slice may be done as part of the segmentation by specifying the slice threshold value.

It then scans in a top down raster looking for a 1-value edge. This outside edge boundary is then traced and relabeled as the next connected component number (starting at 2). After it is traced, if the /H switch is on, it fills the 0's in the region inside of the boundary with the connected component number. It then changes all 1's inside of the boundary in <BMj> to the connected component number.

Components which meet the sizing criteria are saved in a linked list in the node freestore buffer memory as well as having that connected component (and optional trace) saved. The freestore is initialized by specification of the /I switch. The pointer to the connected component descriptor list is pointed to by QRZ. The list data structure is discussed in the appendix. The number of components (total) is saved in QRC.

Each connected component which meets the sizing conditions causes information on its parameters to be printed. The teletype is used unless the /L switch is specified in which case the LPT: is used. The information printed is:

connected component number,
minimum window upper left hander corner:
horizontal position, vertical position in pixels,
horizontal size, vertical size in microns or pixels,
object hole area filled in computed in microns or pixels.
object density (gray value) of pixel data from <BMi2> if it is specified otherwise <BMi> is used.
object area in square microns or pixels.
object boundary perimeter in microns or pixels computed by taking 45 degree pixel changes as Sqrt(2).
number of boundary points of the object.
object density/area (area is either microns of pixels (/N)).
perimeter squared/area global shape feature.
number of boundary edges intersecting the BM edge.

If the /B switch is set and the sizing criteria is met, each accepted boundary is written out into an ASCII file DSK:GENSYM.DA in the following format. An additional boundary trace is performed with the cursor after the feature list is printed as the boundary data is written out into the data file.

```
CC#, [Hor pos, Vert pos, Hor size, Vert size], filled area
        density, area, perimeter
        # bnd pts, density/area, (perim**2)/area, #edges/BM
X1 Y1
X2 Y2
.  .
.  .
.  .
Xk Yk
-1 0
.  .
.  .
.  .
CC#, [Hor pos, Vert pos, Hor size, Vert size], filled area
        density, area, perimeter
        # bnd pts, density/area, (perim**2)/area, #edges/BM
X1 Y1
X2 Y2
.  .
.  .
.  .
X1 Y1
-1 0
-1 -1  (logical end of file)
```

If the /X switch is specified in conjunction with the /B switch then each line containing the (Xi,Yi) data also contains the Freeman chain code and the RLM semantic labeling index (range [1:27]) of the boundary point for the boundary (0 degrees) and its 90 degree rotation. If /X is used, an addition trace of <BMj> not over the current CC boundary (but over where the 90 degree rotation is) will be observed while the RLM for the rotated image is being computed. The file may be printed with BNDPRINT.

2.12.2 SEG2PS*

<BMj>_SEG2PS,<BMi>,(Opt. lower, upper area sizing) - computes the connected components image of <BMi> using a two pass segmentation algorithm discussed in [7].

Assuming a thresholded image, the algorithm looks for non-zero edges and assigns them connected component numbers if they are not adjacent to another component. The case where they are adjacent causes the adjacent point with an existing component number to be propagated into the new point. These cases are listed below.

```
- b -          - b -
- a -   ==>    - b -

- - b          - - b
- a -   ==>    - b -

- - -          - - -
b a -   ==>    b b -
```

The 2nd pass merges those components which were discovered to be connected during the later part of the first pass.

2.12.3 SEGMRG*

<BMj>_SEGMRG*,<BMi>,(lower sizing R1, upper sizing R2), (Optional Threshold t) - segment BMi into BMj connected component image in two passes removing objects whose area is outside of [R1:R2]. The optional range of [R1:R2]=[2:2047]. The optional threshold is used to slice the image in the creation of the initial binary image (default t = 1).

2.12.4 RMVBLOBS*

<BMj>_RMVBLOBS*,<BMi>,(Opt. diameter d), (Opt. slice threshold t), (Opt /A to do rules (a.1:4)&(b.1:4)&(c.1:4) rather than just rules (b.1:4)&(c.1:4). - Remove noise blobs which are small compact regions (> threshold t) of size leq d. The default values for (t,d) are (5,0). The algorithm is as follows:

[1] Generate a binary image <BMj> from <BMi>.
<BMj'>_<BMj>_SCALE(SLICE,<BMi>,t,255) to [0:1];

[2] For i_1 Step 1 Until d Do
Shrink(<BMj>_COPY,<BMi>) using 3x3 neighborhood connectivity to create a new BMj. If a pixel is isolated, then add (x,y) to the delete!list. Up to 2047 points can be deleted. The shrink algorithm uses up to 12 rules applied to the image at each pixel. Only center 1's may be changed to 0's. If a rule has one or more X's, then the sum of the corresponding points have to be greater than 1. The 12 rules are broken into three groups a, b, c.

FACE ERODE

```
a.1 011     a.2 110     a.3 111     a.4 000
    011         110         111         111
    011         110         000         111
```

EXTREMITY ERODE

```
b.1 000     b.2 0XX     b.3 000     b.4 XX0
```

```
    01X          01X          X10          X10
    0XX          000          XX0          000
        SUBEXTREMITY ERODE
        ------------------
c.1 001      c.2 0XX      c.3 100      c.4 XX0
    011          011          110          110
    0XX          001          XX0          100
```

[3] Insert connected components from the list of delete!list points as gray value 255.

```
    For all (x,y) in delete!list Do
            <BMj'>(x,y)_255;
    <BMj>_COPY,<BMj'>;
```

[4] Propagate <BMj> until all 1's which are 8-neighbors of 255 valued pixels are changed to 255.

[5] Extract a mask of 255 from <BMj> and subtract to from the gray value image <BMi> as:

```
    <BMj>_<BMj>,SLICE,<BMi>,0,1;
```

Note: the iteration number appears in the low 3 digits of the right QMT display. The number of isolated pixels is given in the left 4 digits.

2.13 Measurement operators

These operators compute scalar valued functions of images.

2.13.1 AREA

AREA, <BMi>, (Opt. min density, max density) - Computes the total area of the image pixels within the specified density range. The measurement is in microns unless /N is specified in which case it is in number of pixels.

2.13.2 DENSITY

DENSITY, <BMi>, (Opt. min density, max density) - Computes the total normalized density of the image pixels within the specified density range. The density value is the sum of gray values divided by area.

2.13.3 PERIMETER

PERIMETER, <BMi>, (Opt. min density, max density) - Computes the total perimeter of the image for pixels within the specified density range. The perimeter is computed in microns unless /N is specified in which case it is in pixels. Diagonal edges are counted as square root 2 pixels.

2.13.4 SUMDIFF

<BMi1>,SUMDIFF,<BMi2> - Computes the sum of the absolute value of the pixel differences/(image area).

2.13.5 COMASS*

COMASS,<BMi> - computes the center of mass of the single (assumed) object in <BMi> using the maximums of the horizontal and vertical line density projections. After the center of mass is computed, the frame and scale window is positioned at the center of mass of <BMi>. The relative center of mass is returned in Q-registers [A,B]; the absolute (LCS) center of mass is returned in [C,D].

2.14 Quantimet data acquisition commands

The Quantimet 720 image analyzer (discussed in [3,4]) can measure some features of threshold segmented objects very rapidly. These commands facilitate that type of data acquisition. Note that the thresholds can be set manually using the threshold keys on the control desk. The values of the thresholds are printed using the PARAMETERS/M command.

2.14.1 QDATA

QDATA (OPT. /C /F /H /Q /N /i /W /K), (Opt. lower, upper sizes), (Opt. histogram scale factor numerator, denominator) - run the Quantimet (QMT) on the current detected video (which need not be that of a BM). The QMT is explained in more detail in [3]. It always prints the full field data, lens and zoom magnification, Detector module thresholds B (max) and C (min) values.

The /F switch causes the data to be taken from the two function computers. If the sizing is specified, it will accept data which matches function computer #1 such that (lower size < data < upper size) and the upper size limit is optional.

The /C switch accepts features outside of the specified range rather than inside of the range.

If /N is used then convert the output to pixels otherwise convert the output to microns.

If /Q is used, then compute FC1*FC1/area rather than FC1/area.

The /H switch causes the FC1/area data to be used in generating a frequency distribution histogram. The value FC1/area is scaled by the optional numerator/denominator arguments in the call (default of 1/1) and is then used to increment the standard built-in histogram which can store values up to 16,000,000 over the range of 0:255. The histogram may be zeroed with an "INIT/H" command and displayed with a "BMj_SHOWHISTOGRAM" command. Note that if the /Q switch were specified then (FC1*FC1/area) would be used in the frequency distribution.

If /W is used, then wait after every entry for either CLASSKEY[0] to "accept" the data for the object and print it, or CLASSKEY[1] to reject the data for this object. CLASSKEY[11] may be pressed to terminate the search.

If switch /K (possibly instead of or in addition to /W)

then it waits after every cursor movement for you to enter the cell class number via the key pad by doing a (clear, enter number, send). If 0 is sent, then reject the cell.

If switches /1 through /7 are used, then the function computers are programed as follows with the result of FC1/FC2 being printed. The assumption is that FC2 is set to "area".

Switch	FC1	FC2
/1	DENSITY	AREA
/2	AREA	AREA
/3	PERIMETER	AREA
/4	VERT.PROJ	AREA
/5	HOR. PROJ	AREA
/6	HOR. FERET	AREA
/7	VERT.FERET	AREA

2.14.2 LOADTHRESHOLDS

LOADTHRESHOLDS, threshold B, threshold C - load the QMT detector module with thresholds B (maximum) and C (minimum)
over the density range of 0:255 (white to black). Note the QMT Detector module switches should be set to "slice" mode with the select switch on "3".

2.15 Texture measures

Two classes of texture measures of gray scale images are computed: run length distributions and joint gray scale distributions (co-occurrence matrices).

2.15.1 RLTXTURE*

RLTXTURE,<BMi> - Compute the five texture measures discussed in [6]. The histogram of the entire image is first acquired with a sampling of 5 in order to normalize the image data. The /L switch will print the Pk run lengths as well as the statistics on the logical LPT: file RLTXTURE.DA. The histogram is used as a scratch area and is destroyed.

Notation

Ng=# distinct gray levels (16 levels=g(x,y)/16)

Nr=# distinct runs (8):
(0, 1, 2:3, 4:7, 8:15, 16:31, 32:63, 64:127, 128:255)

Pk(i,j)=run length histogram matrix for angle k where
i=gray level, j=run length.

Let: $S(m:N)$ denote $\sum_{m=1}^{N}$

$$\text{alpha}(k) = S(i:Ng)\, S(j:Nr)\, Pk(i,j)$$

Pt=# points in picture.

Five measures of Pk(i,j) are computed as follows:

RF1k (short run emphasis)
=S(i:Ng)S(j:Nr)(Pk(i,j)/(j*j))/alpha(k);

RF2k (long run emphasis)
=S(i:Ng)S(j:Nr)(Pk(i,j)*(j*j))/alpha(k);

RF3k (gray level distributions of runs)
=S(i:Ng)([S(j:Nr)(Pk(i,j)]**2)/alpha(k);
(RF3 is low when runs are equally distributed throughout the gray levels. High run length values contribute most to this function.)

RF4k (run length distribution)
=S(j:Nr)([S(i:Ng)(Pk(i,j)]**2)/alpha(k);

RF5k (Run %=(total # runs/total # runs if all runs were of length 1)
= alpha(k)/Pt.

2.15.2 JGSTXTURE*

JGSTXTURE,<BMi>,(distance d in pixels) (/L to dump Pk(i,j): k=0, 45, 90, 135 degrees and sum of the 4 angles) - Compute the Joint Gray Scale (JGS) co-occurrence density distributions, Pk(i,j), where k is an angle 0, 45, 90 or 135 degrees.

Before the Pk(i,j) are computed, the histogram of the entire image is acquired with a sampling of 5 in order to normalize the gray values. Index i is the central pixel gray value/32, The other index j is computed as the gray value/32 of a pixel distance d away in direction k.

After the Pk(i,j) are computed, the mean(i), mean(j), stddev(i), stddev(j) and covar(i,j) are computed. The JGS is discussed in [7]. The /L switch will dump the Pk JGS distributions as well as the other parameters of the Pk. If the QPENFILE command was active, then the statistics are also dumped on the logical LPT: file JGSTXT.DA. The histogram is used as a scratch area and is destroyed.

2.15.3 JGSPLOT*

<BMj>_JGSPLOT,<BMi>,(distance d in pixels) (Opt. /1, /I /T switches) - Compute the Joint Gray Scale (JGS) co-occurrence density distributions, Pk(i,j), where k is an angle 0, 45, 90 or 135 degrees.

If the /1 switch is not used, then before the Pk(i,j)

are computed, the histogram of the entire image is acquired with a sampling of 1 in order to normalize the gray values. Index i is the central pixel gray value/2, The other index j is computed as the gray value/2 of a pixel distance d away in direction k.

After the Pk(i,j) are computed, they are ploted in the output <BMj> in four quadrants as (+i down, +j right):

```
---------------
| P0   | P45  |
| P90  | P135 |
---------------
```

The background is set to 75 to facilitate photographing it. If /I is specified, then the background is set to 0 and each occurrence of a scatter point causes that pixel to be incremented to a maximum of 255, i.e. if the value Pk(i,j) > 255 then it is clipped to 255.

If the /1 switch is specified, then all Pk(i,j) are computed without normalization (i.e. over the range of [0:255]) and are added together in one plot.

The histogram is used as a scratch area and is destroyed.

The /T switch causes the /I switch to be forced on and several features of Pk(i,j) to be computed after Pk is computed. The features are discussed in an paper by Pressman [18].

T15 (coefficient of variation) =

$$\text{std dev (Pk)/mean (Pk)}.$$

T19 (2nd difference moment) =

$$\sum_i \sum_j (0.5*|i-j|*Pk(i,j))$$

2.16 3D reconstruction operations

Given a sequence of serial sections of an object it is possible to reconstruct another view from these images by computing the image from parts of the given images.

2.16.1 RECONSTRUCT*

RECONSTRUCT (/X or /Y line specification) (Opt. /N to acquire image data directly from the microscpe TV camera rather than from MTAi:) (Opt. /A /i), MTAi:PREFIX.EXT, number of sections (s), line of row or column to reconstruct (v), repeat width in pixels (w) ,(Opt. step size for focus if /N) - given a sequence of 256x256 (or 512x512 if /A is specified) images on magtape, read in s sets of 256x256 or 4 256x256 images (making up a single 512x512 image) with PREFIX and EXT file names into BM0H to BM3H. Then reconstruct in BM0L to BM3I the specified horizontal (/X) or vertical (/Y) section. This is done for all s sections.

If /N is specified, then sections are acquired on line rather than from the MTAi: which is now not required in the specification. The microscope moves down through the section on each scan. The step size is optional (default is 1 step).

If a /i switch is specified then use every i'th magtape file (i is 1 to 9 with /1 the default).

The PIXMTA/F operator may be used to acquire such a set of images.
The reconstruction is done on either a 256x256 or 512x512 pixel image and uses all of BM0 to BM3. Let g(x,y,k) be the pixel gray value (x,y in [0:255] or [0:511] if /A) for section k. Then the reconstructed image is:

$$Gx(i,j) = g(v,j,\text{TRUNCATE}(i/w)),$$

$$Gy(i,j) = g(i,v,\text{TRUNCATE}(j/w)).$$

3. TELETYPE CONTROL CHARACTERS

Various special teletype characters are activated during different times in BMON2 during command decoder input, processing, and waiting for a command.

1. Command decoder input:
   - carriage return - execute the command.
   - rubout - erase last character typed.
   - ^U - erase input line.
   - line feed - type cleaned up line.
   - ^C - return immediately to OS8 (do not use, use EXIT).
2. During processing:
   - ^S - stop processing until ^Q is typed.
   - ^Q - continue processing (after having previously typed ^S).
   - ^O - terminate processing.
3. Waiting for command:
   - ^C - save DDTG command and return to OS8 (same as EXIT above).

4. BMON2 OPERATOR SWITCHES

Various software switches may be included in the teletype commands. The switches are entered by preceeding the letter name switch with a "/". The semantics of the switches is mentioned in the individual command descriptions as well as repeated here alphabetically.

/A switch is used with GET, POST, and UNPOST to specify all lower buffer memories. /A is used by MOVSTATE to specify an absolute rather than a relative change. /A used with READ/WRITE causes 16-bit BM data to be transfered. /A is used to specify the add operation in EVAL. The /A in CAMERA causes the camera to take a picture without teletype prompting. /A used with STDBM or INIT positions the BM0,1,2,3 at the center of the QMT screen. /A is used with PIXMTA, REVIEW, and RECONSTRUCT operators to indicate that a 512x512 size image consisting of BM0,1,2,3 is to be used rather than a 256x256 image size. /A used with BNDPRINT causes all points to be labeled if /L is specified. /A used with NOTCH causes the baseline of the notch filter to be computed during a first pass and then added to the notch filtered image during the second pass.

/B switch is used with GET, POST, UNPOST to swap binary and gray scale halves of the specified memory. Normally, if BM0 is specified, then BM0L is the gray scale image and BM0H is the binary mask if the /M switch is specified. If /B is specified, then this correspondence is reversed. /B is used with READ/WRITE to specify a boundary data structure. /B used with the GRAPHPEN operator reverses the sense of CLASSKEY[1]. /B used in STDBM or INIT positions the STDBM upper left hand corner from the position of the F&S window. /B with SEGBND causes an ASCII boundary data file to be generated. /B used with BNDPRINT causes only bending energy to be computed.

/C switch is used to complement output resulting from raster operations using the function:
gray value = 255-gray value. The /C switch used with HELP lists numbered lines. /C used with WRITE and PIXMTA causes a header comment to be requested from the user. /C with SEGBND causes the sizing criteria to be negagated before being used.

/D switch saves the direction of the gradient coded as an integer i (in the range [1:8] where the angle is (i-1)*45 degrees) for the GRAD4 and GRAD8 commands instead of the magnitude of the gradient. See the specific commands for the coding assignment. /D in READ specifies that the output device is to be the Dicomed. /D in MAG10 specifies that a file being writen on the tape from a disk file will be deleted from the disk after it is written on the tape. /D in EVAL specifies the divide operation.

/E in PARAMETERS is used to evaluate the argument variable. /E in DRWDICMED or in READ with /D is used to erase the DICOMED. /E used with EVAL specifies the equal conditional test.

/F switch is used with the QDATA command to acquire data from the QMT function computers rather than from the MS3 computer. As the data is printed out, the QMT cursor is moved to the ACP coordinates (with ^S,^Q,^O active) where the /W and /K switches are active if specified. /F with PIXMTA causes a set of serial microscope sections to be generated.

/G is used with the WRITE command to acquire a galvanometer scan. /G is used in PARAMETERS to print the value of GENSYM. /G is used with EVAL to indicate the greater than conditional test. Use /G to individually select objects with SEGBND.

/H is used with the SEGBND operator to fill holes in objects after finding the boundaries of those objects. /H used with READ causes the file header to be printed. /H used with INIT clears the histogram array. /H used with QDATA causes the data to be stored in the histogram array as a frequency distribution. /H with PIXMTA causes the picture header to be printed after the file is transfered. /H with REVIEW causes the picture file headers to be printed after the file is transfered.

/I is used with PLOT2D and JGSPLOT to compute the actual 2-D distribution (clipped at 255) which may possibly need to be scaled to be seen. /I initializes the freestore in EXTRACT and SEGBND.

/J used in the HISTOGRAM command causes the image to display the histogram with relatively whiter intensities.

/K is used with QDATA to classify or reject (if class=0) objects scanned in function computer mode by entering class numbers in the range of 0:999 from the keypad. /K is used in EXTRACT to assign a class number to an extracted region. /K is used in WRITE to specify a class number. /K is used with PARAMETERS to print the values of the knob pots.

/L switch is used with the HISTOGRAM operator to print the histogram on the teletype (or file if OPENFILEd). /L is also used in SMOOTHISTOGRAM to specify that the individual H(x), noise figure(x) be listed. /L is used in RLTXTURE and JGSTXTURE to specify that the Pk distributions should also be printed. /L in READ/WRITE specifies that the BM data is in the form of a line data structure. /L used in EVAL specifies the less than conditional test. /L in SEGBND prints the results on the lineprinter. /L with BNDPRINT requests semantic labeling.

/M switch turns on the QMT mask generated by taking the logical AND of (BM mask) and (QMT variable frame) /V is the inverse of /M. /M is used with PARAMETERS to print only the values of the stepping motors. /M is used in MOVSTATE to specify that the number of steps is minus. /M in EVAL specifies the minus operation. /M with PIXMTA and REVIEW causes control to remain in that function while another command is acquired from the command decoder. The /M switch with SEGBND permits multiple feature sizing to be used in the segmentation. /M used with BNDPRINT requests a list of semantic labels from the command

decoder.

/N is used with QDATA to cause QMT data to be presented in terms of pixels rather than being converted to microns (which takes the lens and zoom into account). In FILTER, /N specifies a negative Ni. /N in EVAL specifies the minimum arithmetic operator. /N with SMPGET collapses data in BM0:3 to <BMj> without doing a get. /N in reconstruct gets data from the microscope rather than from the magtape.

/O omits printing the command key specification when the key is pressed.

/P is used with all commands to specify that the PARAMETERS with /P be printed after the command operation is finished. If specified with with the PARAMETERS command, it prints the parameters without the status of the buffer memories. /P in EVAL specifies the product (multiply) operation.

/Q switch is used with the QDATA command to compute FC1*FC1/AREA rather than FC1/AREA. /Q with READ/WRITE output specifies that QMT function computer data is to be acquired. /Q used with HISTOGRAM causes the histogram data to be plotted on a full scale equal to 1000.

/R used with HIST causes the histogram to be a runlength histogram of the gray values in each line of the image. /R used with READ causes the state of the stepping motors to be restored from the data file header. /R with PIXMTA and REVIEW causes the magtape unit to be rewound before the operation. /R with PARAMETERS prints the 26 Q-registers.

/S is used to save the current command line and associate it with a control desk command key 0 through 11. The command key is specified by an octal number nn by "=nn" where nn is 0:13 (0:11 decimal).

/T switch used with the SEGMENT operator traces the boundary while the image is being segmented in the opposite half of the output <BMj> memory. Use POST,<BMj>/M to see it. /T used with BNDPRINT translates labeled points -90 degrees. /T used with JGSPLOT computes several texture features of the joint gray scale density function of an image. /T used with ISOLATE traces the possible pairs of components eligible for isolation in BMj'. /T used with PIXMTA causes the system to sample images from the specified BM to magtape at the required time increment.

/U switch is used with image operations requiring an input image BMi1 which is posted. The frame and scale when positioned over the image may be used to delimit a computing window for the operation.

/V switch turns off the QMT mask generated by the BM anded with the F&S so that the so that the QMT frame is the (normal) variable frame. /M is the inverse of /V. The INIT command will also set the mode to variable frame. /V used with BNDPRINT

causes the image to be split using the BSPLIT part of the BNDPRINT function after acceptable corners were found.

/W is used with QDATA to accept, reject or terminate processing of function computer data by pressing CLASSKEY[0] to accept an object, CLASSKEY[1] to reject an object, or CLASSKEY[11] to terminate processing. The cursor stops at the current object in question. Note: /W and /K may be used together, in which case /W has its effect first.

/X in HIST causes a row density profile to be generated. /X in FILTER specifies a negative x offset. /X in EVAL specifies the maximum arithmetic operator. /X with PIXMTA causes data currently in the BM to be transfered to a magtape file and control to return to BMON2 when done. /X in REVIEW causes the tape to read 1 file into a buffer memory and then to return to BMON2. /X used with RECONSTRUCT reconstructs a horizontal section. /X is used in SEGBND with /B to dump the boundary point chain code and RLM point semantics as well as the (x,y) coordinates. The /X used with NGHSE causes a neighborhood maximum to be computed while omitting it computes the neighborhood minimum.

/Y is used to specify that the mouse should be used instead of the GraphPen in EXTRACT. In FILTER, /Y specifies a negative y offset. /Y used with HIST causes a column profile to be generated. /Y used with RECONSTRUCT reconstructs a vertical section.

The /1 switch is used with the GRAYBAR command to select a linear (versus the sqrt(2) related) test pattern. /1 is used in JGSPPLOT to compute the JGS distribution with nor normalization of the data. /1 is used with INIT to select the 1X lens.

/2 is used with INIT to select the 2.5X objective lens.

/3 is used with INIT to select the 5X objective lens.

/4 is used with INIT to select the 10X objective lens.

/5 is used with INIT to select the 25X objective lens.

/6 is used with INIT to select the 50X dry objective lens.

/7 is used with INIT to select the 50X oil objective lens.

/8 is used with INIT to select the 100X oil objective lens. /8 is used with PARAMETERS to print the parameters on the lineprinter.

/9 is used with PARAMETERS to print a file name on the lineprinter.

Switches /1 through /7 are used with QDATA to select the function computer program (see QDATA operator).

Switches /0 through /9 are used with SEGBND to select the sizing criteria feature (see SEGBND).

Switches /1 through /9 are used in RECONSTRUCT to determine the tape sampling function.

5. REFERENCES

1. Lemkin P, Carman G, Lipkin L, Shapiro B, Schultz M, Kaiser P:A Real Time Picture Processor for Use in Biological Cell Identification - I Systems Design. J Hist Cyto, Vol 22, 1974, 732:740.

2. Carman G, Lemkin P, Lipkin L, Shapiro B, Schultz M, Kaiser P:A Real Time Picture Processor for Use in Biological Cell Identification - II Hardware Implementation. J Hist Cyto, Vol 22, 1974, 732:740.

3. Lemkin P, Carman G, Lipkin L, Shapiro B, Schultz M: Real Time Picture Processor - Description and Specification. NCI/IP Technical Report #7. March 31, 1976, (revised Report #7a, June, 1977, NTIS PB269600/AS).

4. Lemkin P:Functional Specifications for the RTPP Monitor - Debugger DDTG. NTIS PB250726 (or DECUS 8-823) NCI/IP Technical Report #2, Feb 1976.

5. Wechsler H, Sklansky J:Automatic Detection of Rib Contours in Chest Radiographs. 4IJCAI, 1975, pp.688:694.

6. Galloway M: Texture Analysis using Gray Scale Run Lengths. Comp Graph Image Proc, Vol (?), 1975(?), ?:?.

7. Rosenfeld A, Kak A:Digital Picture Processing. Academic Press, 1976.

8. Digital Equipment Corp.:OS8 Handbook. Maynard, Mass., 1974.

9. Merrill R D:Representation of Continuous Regions for Efficient Computer Search. CACM, Vol 16, 1973, 69:82.

10. Lemkin P:MAG10 - A PDP8e File Based Magtape Utility. NTIS PB261534/AS (or DECUS 8-879), NCI/IP Technical Report #20, 1976.

11. Schacter B, Davis L, Rosenfeld A: Some Experiments in Image Segmentation by Clustering of Local Feature Values. Comp Graph Image Proc, Vol ?, 197?, ?:?.

12. Lemkin P:BMOMNI - Fortran Interface Program for the RTPP Buffer Memory, Quantimet and Control Desk. NTIS PB261538/AS, NCI/IP Technical Report #23. December, 1976.

13. Smith K T, Solmon D C, Wagner S L:Practical and Mathematical Aspects of the Problem of Reconstructing Objects from Radiographs. Address to Far West Sect Meeting of American Math Soc, Monterey, Calif., April, 1975.

14. Lemkin P:The Run Length Map - a representation of contours and regions for efficient search and low level semantic encoding. In preparation.

15. Young I T, Walker J E, Bowie J E: An Analysis Technique for Biological Shape I. Inf Control, Vol 25, 1974, 357:370.

16. Freeman H: Computer Processing of Line Drawing Images. Comput Surveys, Vol 6, March 1974, 57:97.

17. Freeman H: Shape Description Via the Use of Critical Points. Proc Pattern Recog and Image Process Conf. Troy, NY., 1977, 168:174.

18. Pressman N J:Optical Texture Analysis for Automatic Cytology and Histology: A Markovian Approach. Ph.D. Dissertation, Lawrence Livermore Lab, Univ. Cal., Report # UCRL-52155, Oct 1976.

19. Wied G, Bahr G F, Bartels P H: Automatic Analysis of Cell Images by TICAS. In "Automated Cell Identification and Cell Sorting", Wied G, Bahr G F (eds), Academic Press, NY, 1970, 195:356.

20. Brenner J F, Dew S B, Horton J B, King T, Neurath P W, Selles W D: An Automated Microscope for Cytoplgic Research. J Hist. Cyto., Vol 24, 1976, 100:?.

21. Bengtsson E, Holmquist J, Olsen B, Stenkvist B: SCANCANS - An Interactive Scanning Cell Analysis System. Comput Prog Biom, Vol 6, 1976, 39:49.

22. Nakagawa Y, Rosenfeld A: A note on the use of local min and max operations in digital picture processing. U Md CMSC TR-590, Oct, 1977.

APPENDIX A

ADDING NEW OPERATORS

It is possible to add new operators to the BMON2 system via the CHAIN feature of OS8. This feature permits programs (such as BMON2 or the operators themselves) to run other programs by requesting the OS8 monitor to run another program. These programs are core image (.SV) files created by compiling, loading and saving a program. The critical component of interfacing these new operator programs to BMON2 is in saving and restoring as well as using the state information generated in BMON2 and the operator programs. The state of BMON2 is contained in two files SYS:SVDDTG.DA (which also contains the state of DDTG) and SYS:SVBMCN.DA (which contains additional state information used by BMON2). A single procedure BSCOMMON can be used to either save or restore the state from these files and thus enables easy communication between BMON2 and the set of operator programs.

Note that BMON2 always restores COMMON on entry and saves it on exit. Auxiliary ".SV" operators, however, do not need to restore COMMON (file: BMCMN.FT) since it is preserved across the Chain OS8 operation. If the operator does not change the state of BMON2 (as would be reflected by a change in COMMON) then it does not have to save the state before chaining back to BMON2 (thus resulting in faster operation).

Most operators are created by taking the skeleton procedure "BMBODY".FT and changing the occurrence of "BMBODY" to the new procedure name throughout the file. It has a prologue to restore the state and an epilogue to save it and chain back to BMON2. It uses BMCMN so that any procedures using COMMON will have it available. Actual BM I/O may be performed either using BMIO.FT (which is a set of procedure calls passing arguments through COMMON) or BMOMNI.FT (which is a single procedure passing arguments via the subroutine argument list). The latter or course is much slower but is also easier for the novice to use.

BMOMNI also facilitates writing operator programs because it has a nearly complete set of interface functions for: BM I/O, posting, display, and movie generation; GraphPen; Quantimet control and data acquisition; Control Desk keys, pots and lights. It also has a clipping vector generator for use in drawing and/or generating vectors in the [0:255] pixel square domain.

A.1 Command line variables in COMMON

BMON2 saves the arguments, entered at the command level, in COMMON. These arguments are available to the chained segment for further analysis and use. The table below

summarizes some of the more useful COMMON data structure name and its contents.

COMMON variable	Contents
KDEVOUT	- output OS8 device number.
KOUTFILE[1:3]	- output file name 3A2 format.
KOUTFILE[4]	- output file extension A2 format.
KDEVIN[1:5]	- 5 input OS8 device numbers.
KINFILE[1:3+(i-1)*4]	- 5 input file names 3A2 format.
KINFILE[4+(i-1)*4]	- 5 input file extensions A2 format.
ISW[1:36]	- 36 alphameric switches ("/i").
ICNUM[1:5]	- Up to 5 magnitude integers used in place of input file names.
(JBM,JHGH)	- output BM specification.
(IBM1,IHGH1)	- 1st input BM specification.
(IBM2,IHGH2)	- 2nd input BM specification.
ITMPSTK[1:26]	- 26 Q-registers, single precision.
IQREG[1:26]	- 26 Q-registers, double precision most significant word.
LSFREESTORE[1:2]	- Node freestore pointer.
IFILTER	- current lens number.
IH[1:512]	- 256 double precision histogram array.
IOUTSPOOL	- lineprinter spooling switch (1=on, 0=off).
LSAVE[1:12,1:80]	- command key partially parsed saved command lines.
LSAVE[13,1:8]	- X positions of BM0 to BM7.
LSAVE[14,1:8]	- Y positions of BM0 to BM7.
LSAVE[15,1]	- lens number.

A.2 Spooling BMON2 teletype output

The OPENFILE command causes all output which is performed on the teletype to be repeated on the output spooling device. The CLOSEFILE command must be used to close the file. If no output file is mentioned with the OPENFILE, then the OS8 lineprinter (LPT:) device is assumed. If one chains to a program with the spooler active, then the spooler is closed and upon restarting BMON2, the OPENFILE command is simulated with the LPT: device specified. The spooler switch IOUTSPOOL (in COMMON area BMCMN.FT) remains active during the CHAIN and thus can be accessed by chained programs.

APPENDIX B

Q-REGISTER USAGE

Certain operators will return one or more values in some of the Q-registers for later use. These operator "side-effects" are listed here.

HISTOGRAM

QRA - minimum gray value range.
QRB - maximum gray value range.
QRC - minimum gray value.
QRD - maximum gray value.

CONTRAST

QRA - minimum gray value range.
QRB - maximum gray value range.

COMASS

QRA - relative X center of mass.
QRB - relative Y center of mass.
QRC - absolute X center of mass.
QRD - absolute Y center of mass.

SEGBND

QRZ - double precision pointer to set of connected components.
QRC - total number of connected components.

EXTRACT

QRZ - double precision pointer to line drawing.

APPENDIX C

BNF COMMAND LINE SYNTAX

The BMON2 command line syntax is presented here in the form of a BNF-like grammar. Note that the "_" is an underline character and is equivalent to a backarrow on some teletypes. It is used to indicate assignment. The "<" character is equivalent to the "_" in the OS8 envirnment. Furthermore, if no output specification is required "_<null>" is equivalent to "<null>".

```
<cmd line>::= <cmd> <switches>
<cmd>::= <op>, <args>
<cmd>::= <filespec> _ <op>, <args>
        ::= <filespec> _ <op>, <BM>
        ::= <BM> _ <op>, <filespec>
<cmd>::= <op>, <BM>, <args>
        ::= <BM> _ <op>,<args>
        ::= <op>, <BM>, <BM>, <args>
        ::= <BM>, <op>, <BM>, <args>
        ::= <BM> _ <op>, <BM>, <BM>, <args>
        ::= <BM> _ <BM>, <op>, <BM>, <args>
<filespec>::= <disk device> : <file name> . <extension name>
<disk device>::= SYS | DSKB | DSKC | DSKD | DSKE | DSKF | DSKG |
                 DSKH | DTA0 | DTA1 | LPT
<file name>::= GENSYM | (6 character alphameric identifier
                         beginning with letter)
<extension name>::= (2 character alphameric identifier)
<args>::= <args>,<arg> | <arg>
<arg>::= decimal number up to 4095
        ::= <knob>
        ::= <control desk switch>
        ::= <keypad>
        ::= <Q-register>
        ::= null
<knob>::=P<octal digit>   (Knobs 0:7 with values [0:511])
<control desk switch>::=FBW<decimal digit>   (control desk
                         switches 0:9 with values [0:4095])
<keypad>::= KPD  (keypad values [0:999])
<Q-register>::= QR<letter>
<letter>::= A | B | C | ... |Y | Z
<op>::= legal command.
<BM>::= BM<octal digit><byte>
<octal digit>::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
<decimal digit>::= <octal digit> | 8 | 9
<byte>::= H | L | null
<switches>::= /<letter> | /<octal digit> |
        ::= /S=<2 digit octal number>
```

APPENDIX D

LIST OF FILES REQUIRED FOR BMON2

Documentation files

BMSRC.BI	- Batch job to print BMON2 files.
BMON2H.PUB	- PUB (PDP10) input file to produce BMON2.HL.
BMON2.PUB	- PUB (PDP10) input file to generate complete .INDEXed listing.
BMON2.HL	- help file.
BMON2.BI	- Batch file to compile and build the BMON2.SV file.
BMLOAD.BI	- Batch file to load and build the BMON2.SV file.

COMMON file

BMCMN.FT	- Common area for all BMON2 files using COMMON.

BMON2 proper files

BMON2.FT	- BMON2 main.
BMAX1.FT	- Auxiliary procedures for BMON2.FT
BMAX2.FT	- Auxiliary procedures for BMON2.FT
BMAX3.FT	- Auxiliary procedures for BMON2.FT
BMAX4.FT	- Auxiliary procedures for BMON2.FT
BMAX5.FT	- Auxiliary procedures for BMON2.FT
BMAX6.FT	- Auxiliary procedures for BMON2.FT
BMAX7.FT	- Auxiliary procedures for BMON2.FT
BMAX9.FT	- Auxiliary procedures for BMON2.FT
BMAP1.FT,BMAP2.FT	- Zoom interpolation procedure for microns/pixel calibration.
BMIO.FT	- Buffer memory I/O through COMMON.
BCDSPEC.FT	- OS8 Command Decoder interface procedure.
GETDEVICE.FT	- Procedure to return the Ascii device name from its number.
FINDOPR.FT	- Procedure to lookup built-in function names and PDP8e IOTs.
BSCOMMON.FT	- Save and restore the state of BMON2 COMMON via files.
TIMER.FT	- One second timer and 200Hz clock procedures.
MATCH.FT	- Pattern matcher used in command recognition.
QUESTION.FT	- Guesser used in MATCH for guessing commands.

SEGBND function

SEGBND.BI	- Batch file to compile and build SEGBND.SV.
SEGBND.FT	- SEGBND main.
SEGB1.FT	- Top level raster driver for segmenter.
SEGB2.FT	- Boundary follow, fill and pack.
SEGB3.FT	- Sizing procedure
BFOLLOW.FT	- Next boundary pixel finite state machine.
BWINDOW.FT	- Minimum window and Min(entrance,exit,y)
GETNGH.FT	- 3x3 neighborhood fetching fresh copy. procedure.

```
BNRUN.FT        - Run Length Map procedure using linked lists.
STPROP.FT       - Set property list processing package.
BNODE.FT        - Node list processing package.
FREEST.FT       - BM node list processing package.

EXTRACT function
----------------
EXTRACT.BI      - Batch file to compile and build EXTRACT.SV.
EXTRACT.FT      - EXTRACT main.
MOUSE.FT        - (x,y,switch) input from GRAPHPEN or MOUSE.
GPEDIT.FT       - Major polling loop for EXTRACT.
BNRUN.FT        - Run Length Map procedure using linked lists.
BNODE.FT        - List processing package.
FREEST.FT       - Node freestore package.

RLTXTURE function
-----------------
RLTXTURE.BI     - Batch file to compile and build RLTXTURE.SV.
RLTXTURE.FT     - RLTXTURE main.
RLTX1.FT        - Compute run-length distributions procedure.
RLTX2.FT        - Print run-length distributions procedure.


JGSTXTURE function
------------------
JGSTXTURE.BI    - Batch file to compile and build JGSTXTURE.SV.
JGSTXTURE.FT    - JGSTXTURE main.
JGST1.FT        - Compute joint-gray-scale distributions.
JGST2.FT        - Print joint-gray-scale distributions.

WINDMP function
---------------
WINDMP.BI       - Batch file to compile and build WINDMP.SV.
WINDMP.FT       - WINDMP main.

BNDPRINT function
-----------------
BNDPRINT.BI     - Batch file to compile and build BNDPRINT.SV.
BNDIO.FT        - boundary structure acquisition procedure
IBNDREAD.FT     - SEGBND produced .DA file I/O with OS8.
BNDBE.FT        - bending energy computation.
BMPARAM.FT      - fast set of BMIO specs in common.
BNDCORNER.FT    -  compute  cornerity from avg chain code diff.
                  window
BNDCSTAT.FT     - save corners found and compute pairing heuristic
BNDHEM.FT       - compute chain code difference using MOD 8 arith.
BNDPRINT.FT     - BNDPRINT main.

BSPLIT function
---------------
BSPLIT.BI       - Batch file to compile and build BSPLIT.SV.
TXTGRAD.FT      - Compute Pij's and texture heuristic
CMPTXT.FT       - Compute T15 and T19 given Pij in COMMON
BSPLIT.FT       - BSPLIT main.

PROPAGATE function
------------------
```

```
PROPAGATE.BI     - Batch file to compile and build PROPAGATE.SV.
PROP1.FT         - propagation procedure
PROPAGATE.FT     - PROPAGATE main.

PROP2 function
---------
PROP2.BI         - Batch file to compile and build PROP2.SV.
PROP2.FT         - PROP2 main.

RUNFILTER function
------------------
RUNFILTER.BI     - Batch file to compile and build RUNFILTER.SV.
RUNFILTER.FT     - RUNFILTER main.

NGHSE function
--------------
NGHSE.BI         - Batch file to compile and build NGHSE.SV.
NGHSE.FT         - NGHSE main.

NOTCH function
--------------
NOTCH.BI         - Batch file to compile and build NOTCH.SV.
NOTCH.FT         - NOTCH main.

SHADE function
--------------
SHADE.BI         - Batch file to compile and build SHADE.SV.
SHADE.FT         - SHADE main.

FILGAP function
---------------
FILGAP.BI        - Batch file to compile and build FILGAP.SV.
FILGAP.FT        - FILGAP main.

DRW512 function
---------------
DRW512.BI        - Batch file to compile and build DRW512.SV.
DRW512.FT        - DRW512 main.

ISOLATE function
-------------
ISOLATE.BI       - Batch file to compile and build ISOLATE.SV.
ISOL1.FT         - compute pairs of CC's [i,j] to be isolated.
ISOL2.FT         - compute split heuristic and angle eta.
ISOL3.FT         - correct eta and paint zeros.
ISOLATE.FT       - ISOLATE main.

ZOOM function
-------------
ZOOM.BI          - Batch file to compile and build ZOOM.SV.
ZOOM.FT          - ZOOM main.

ZOOM function
-------------
ROTATE.BI        - Batch file to compile and build ROTATE.SV.
ROTATE.FT        - ROTATE main.
```

```
SETGET function
---------------
SETGET.BI                 - Batch file to compile and build SETGET.SV.
SETGET.FT                 - SETGET main.

VARIANCE function
-----------------
VARIANCE.BI       - Batch file to compile and build VARIANCE.SV.
VARIANCE.FT       - VARIANCE main.

GRAD8 function
--------------
GRAD8.BI          - Batch file to compile and build GRAD8.SV.
GRAD8.FT          - GRAD8 main.

COMASS function
---------------
COMASS.BI         - Batch file to compile and build COMASS.SV.
COMASS.FT         - COMASS main.

MEDIAN function
---------------
MEDIAN.BI         - Batch file to compile and build MEDIAN.SV.
MEDIAN.FT         - MEDIAN main.

MTV function
------------
MTV.BI            - Batch file to compile and build MTV.SV.
MTV.FT            - MTV main.

MIDPOINT function
-----------------
MIDPOINT.BI       - Batch file to compile and build MIDPOINT.SV.
MIDPOINT.FT       - MIDPOINT main.

PLOT2D function
---------------
PLOT2D.BI         - Batch file to compile and build PLOT2D.SV.
PLOT2D.FT         - PLOT2D main.

VARIANCE function
-----------------
VARIANCE.BI       - Batch file to compile and build VARIANCE.SV.
VARIANCE.FT       - VARIANCE main.

SMOOTHISTOGRAM function
-----------------------
SMOOTHISTOGRAM.BI  - Batch file to compile and build SMOOTHISTOGRAM.SV.
SMOOTHISTOGRAM.FT  - SMOOTHISTOGRAM main.
DISASTER.FT       - Routine to actually do the disaster analysis.

FILTER function
---------------
FILTER.BI - Batch file to compile and build FILTER.SV.
FILTER.FT - FILTER main.

JGSPLOT function
```

```
------------------
JGSPLOT.BI      - Batch file to compile and build JGSPLOT.SV.
JGS1.FT         - computes the Pk and plots them
JGSPLOT.FT      - JGSPLOT main.

GRADN function
------------
GRADN.BI        - Batch file to compile and build GRADN.SV.
GETNXN.FT       - Get an NxN neighbrhood.
GRADN.FT        - GRADN main.

AVGN function
------------
AVGN.BI         - Batch file to compile and build AVGN.SV.
AVGN.FT         - AVGN main.

MOVSTATE function
------------
MOVSTATE.BI     - Batch file to compile and build MOVSTATE.SV.
STAGE.FT        -Routine to convert rational stage units to steps
                 and move the stage via MANUAL package.
MOVSTATE.FT     - MOVSTATE main.

FILLBD function
---------------
FILLBD.BI       - Batch file to compile and build FILLBD.SV.
FILLBD.FT       - FILLBD main.

HELP function
---------------
HELP.BI         - Batch file to compile and build HELP.SV.
HELP.FT         - HELP main.

APPLY function
---------------
APPLY.BI              - Batch file to compile and build APPLY.SV.
APPLY.FT              - APPLY main.

CAMERA function
---------------
CAMERA.BI       - Batch file to compile and build CAMERA.SV.
CAMERA.FT       - CAMERA main.

XMITBM function
---------------
XMITBM.BI       - Batch file to compile and build XMITBM.SV.
BDOAUX.FT       - Specific I/O for BGETPUT
BGETPUT.FT      - GET/PUT for RTPP devices/Files for BMON2
PHEADER.FT      - File header printout procedure.
DICMED.FT       - Dicomed display procedure.
XMITBM.FT       - XMITBM main.

PIXMTA function
---------------
PIXMTA.BI       - Batch file to compile and build PIXMTA.SV.
MAGTAP.FT       - Specific MTA: I/O for PIXMTA
BMTOMTA.FT      - BM to MTA I/O procedure
```

```
BDOAUX.FT        - Specific I/O for BGETPUT
PHEADER.FT       - File header printout procedure.
PIXMTA.FT        - PIXMTA main.

REVIEW function
---------------
REVIEW.BI        - Batch file to compile and build REVIEW.SV.
MAGTAP.FT        - Specific MTA: I/O for REVIEW
MTATOBM.FT       - MTA to BM I/O procedure
PHEADER.FT       - File header printout procedure.
REVIEW.FT        - REVIEW main.

RECONSTRUCT function
---------------
RECONSTRUCT.BI   - Batch file to compile and build RECONSTRUCT.SV.
MAGTAP.FT        - Specific MTA: I/O for RECONSTRUCT
MTATOBM.FT       - MTA to BM I/O procedure
RECONSTRUCT.FT   - RECONSTRUCT main.

DEFCONTRAST function
---------------
DEFCONTRAST.BI   - Batch file to compile and build DEFCONTRAST.SV.
DEFCONTRAST.FT   - DEFCONTRAST main.

FNCONTRAST function
---------------
FNCONTRAST.BI    - Batch file to compile and build FNCONTRAST.SV.
FNCONTRAST.FT    - FNCONTRAST main.

BDEDIT function
---------------
BDEDIT.BI        - Batch file to compile and build BDEDIT.SV.
BDEDIT.FT        - BDEDIT main.

SEGMRG function
---------------
SEGMRG.BI        - Batch file to compile and build SEGMRG.SV.
SEGMRG.FT        - SEGMRG main.

RMVBLOB function
---------------
RMVBLOBS.BI      - Batch file to compile and build RMVBLOBS.SV.
RMVBLOBS.FT      - RMVBLOBS main.

NGHSE function
---------------
NGHSES.BI        - Batch file to compile and build NGHSES.SV.
NGHSES.FT        - NGHSES main.

Hardware interface procedures
-----------------------------
BMOMNI.FT        - BM I/O, QMT, Control Desk, GrafPen control
                   procedure package.
MANUAL.FT        - Stepping motor (x,y,focus,zoom,thB,thC,wv,nd)
                   control procedure package.
FREEST.FT        - BM node freestore procedure package.
BNODE.FT         - Node list processing package.
```

BSET.FT - Set list processing package.

Skeleton function program mains

BMBODY.FT - Skeleton of pixel or neighborhood raster function.
BMFAST.FT - Skeleton of fast raster function.

Auxiliary procedures

IOB.FT - OS8 general access file interface function IO procedure package.
DPCVRT.FT - Double precision integer to/from floating point conversion.
OCT.FT - Octal to/from double precision integer conversion.
IBCD.FT - BCD to/from decimal conversion.
MAX.FT - Return the Max of two integers.
MIN.FT - Return the Min of two integers.

APPENDIX E

EXAMPLE OF BATCH JOB FOR BMON2

The following text is a sample of an OS/8 Batch job which contains a sequence of operations on an image in BMON2.

```
$JOB EXPR34.BI
/The  assumption  is made that several images previously
/scanned exist  on  DSK  3.  The  images  GN0003.PX  and
/BL0003.PX  were  scanned  at 560 ad 420 nm respectively
/using a vidicon camera with an arc light source at 100X
/oil (0.8X zoom).

.DATE


.R BMON2
/       [0] Initialize BMON2
*INIT,STATE/A
*POST,BM0
*POST,BM1
*POST,BM2
*POST,BM3


/       [1] Read the green image from DSKG:.
*BM0_READ,DSKG:GN0003.PX
*BM1_HISTOGRAM,BM0
*BM2_CONTRAST,BM0

/       Overlay the image with 10 micron grid.
*BM3_COPY,BM0
*BM3_GRID

/       Save green original image and contrast
/       stretched images in BM0H and BM1H.
*BM0H_COPY,BM0
*BM1H_COPY,BM2

/       Take a picture for Figure E.1
*CAMERA


/       [2] Read the blue image from DSKG:.
*BM0_READ,DSKH:BL0003.PX
*BM1_HISTOGRAM,BM0
*BM2_CONTRAST,BM0

/       Overlay the image with 10 micron grid.
*BM3_COPY,BM0
*BM3_GRID

/       Save blue image and contrast stretched blue
/       image in BM2H and BM3H.
```

```
*BM2H_COPY,BM0
*BM3H_COPY,BM2
/       Take a picture for Figure E.2
*CAMERA


/       [3] Compute the joint green/blue density
/       distributions and plot it (un-contrasted stretched).
*BM0_BM0H,PLOT2D,BM2H
/
/       2D plot of contrast stretched green/blue images.
*BM1_BM1H,PLOT2D,BM3H
*UNPOST,BM2
*UNPOST,BM3
/       Take a picture for Figure E.3
*CAMERA
*POST,BM2
*POST,BM3


/       [4] Compute clipping difference image, GB, between
/       green and blue images.
*BM0_BM1H,SUB,BM3H
*BM1_HISTOGRAM,BM0,1,1,255
/       Smooth the histogram.
*BM2_SMOOTHISTOGRAM
/
/       Slice the nuclei.
*BM3_SLICE,BM0,60,255
*BM0_COPY,BM3
*BM3_SCALE,BM3,255,1

/       Take a picture for Figure E.4
*CAMERA


/       [5] Segment the image at threshold 60
/       computing the connected component and boundary
/       trace images. Ignore objects < 10 and > 100
/       square microns in area. Measure object
/       density from the original green image rather
/       than from the GB (clipping difference image).
*BM1H_SEGBND,BM0,BM0H /T/H/L/I, 10,100/0
/
/       Note: BM3 contains the linked lists used
/       in representing the boundaries and the run length
/       map during the segmentation process.

/       Scale the connected component image so that it may
/       be viewed.
*BM2_SCALE,BM1H,15,1

/       Take a picture for Figure E.5
*CAMERA


/       [6] Extract three components and measure
```

```
/       their density.
*BM0_BM1H,SLICE,BM0H,2,2
*BM1_BM1H,SLICE,BM0H,3,3
*BM2_BM1H,SLICE,BM0H,4,4
*DENSITY,BM0
*DENSITY,BM1
*DENSITY,BM2
/
/       Take a picture for Figure E.6
*UNPOST,BM3
*CAMERA


/       [7] Compute the joint gray scale density texture plots
/       of the nuclei after saving them in the high memories.
/       The densities are taken three pixels apart.
*BM0H_COPY,BM0
*BM1H_COPY,BM1
*BM2H_COPY,BM2
*BM0_JGSPLOT,BM0H,3/1
*BM1_JGSPLOT,BM1H,3/1
*BM2_JGSPLOT,BM2H,3/1
/
/       Take a picture for Figure E.7
*CAMERA


/       [8] Compute the joint gray scale density texture
/       plots of the nuclei after saving them in the high
/       memories. Then densities are taken 7 pixels apart.
*BM0_JGSPLOT,BM0H,7/1
*BM1_JGSPLOT,BM1H,7/1
*BM2_JGSPLOT,BM2H,7/1
/       Take a picture for Figure E.8
*CAMERA


/       [8] Done.
*EXIT
.DATE
$END EXPR34.BI
```

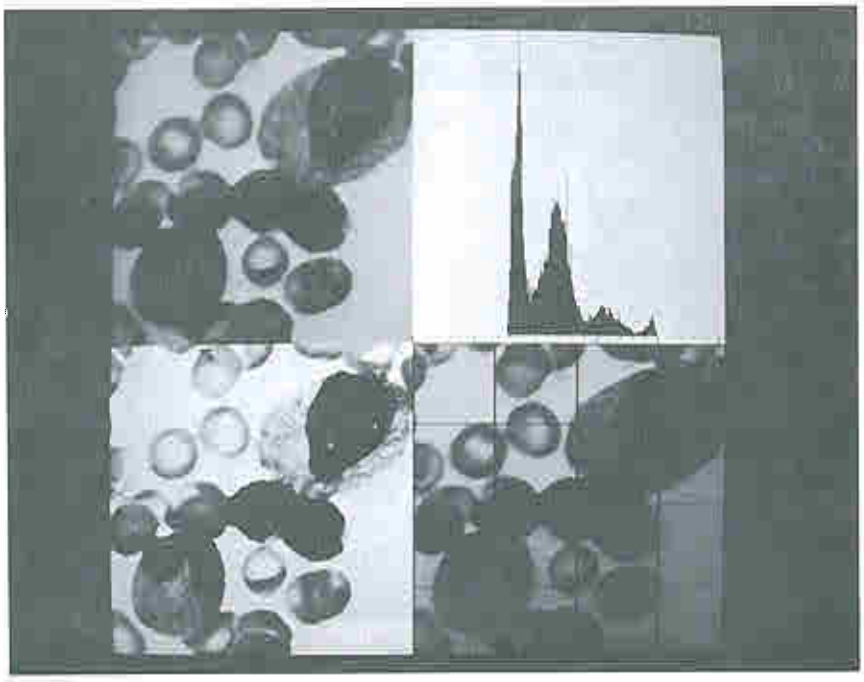


Figure E.1 (a) Green scan, (b) histogram of (a), (c) contrast stretched (a) i.e. CS(green), (d) green scan with 10 micron grid overlayed.

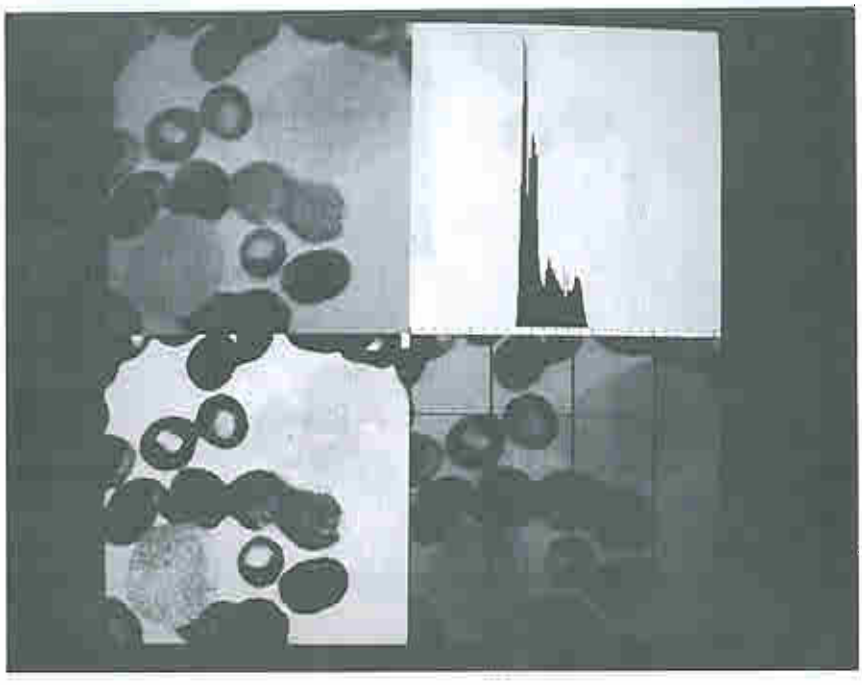


Figure E.2 (a) Blue scan, (b) histogram of (a), (c) contrast stretched (a) i.e. CS(blue), (d) blue scan with 10 micron grid overlayed.

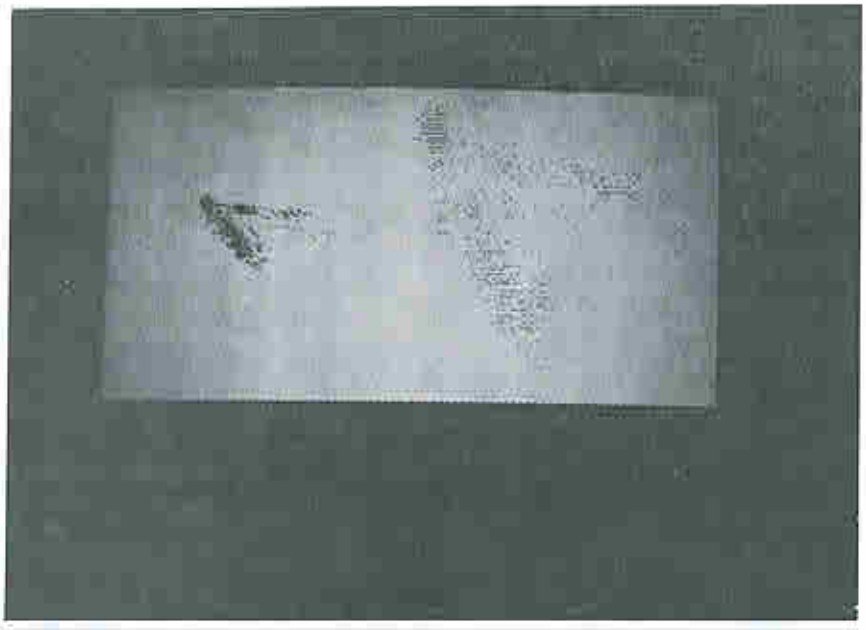


Figure E.3 (a) Joint gray scale distribution of green (x axis) and blue (y axis); (b) same but for contrast stretched images.

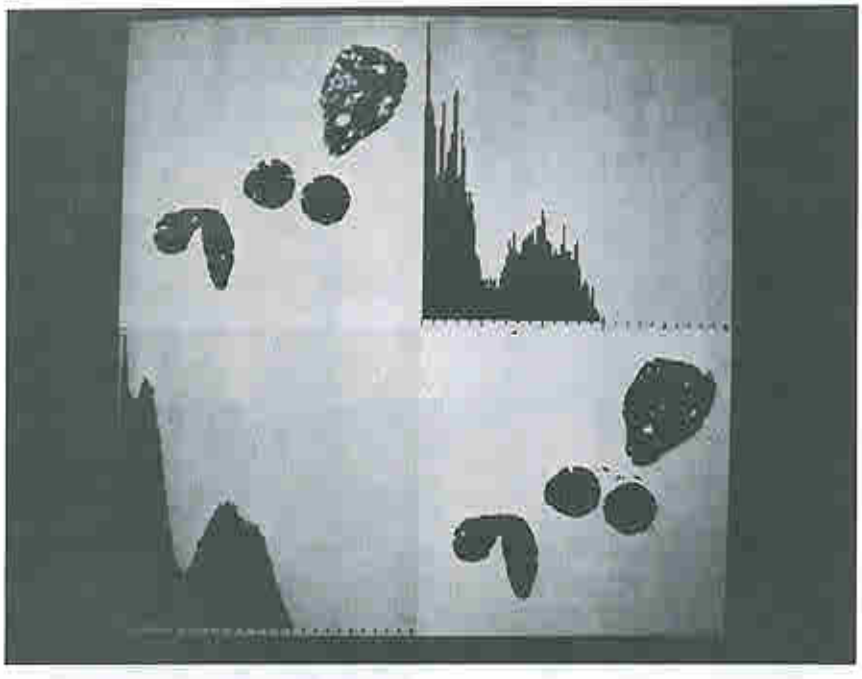


Figure E.4 (a) CS(green)-CS(blue) sliced at 60, (b) histogram of (d), (c) smoothed histogram of (b), (d) CS(green)-CS(blue) sliced at 60 and scaled to 0 (white) and 255 (black).

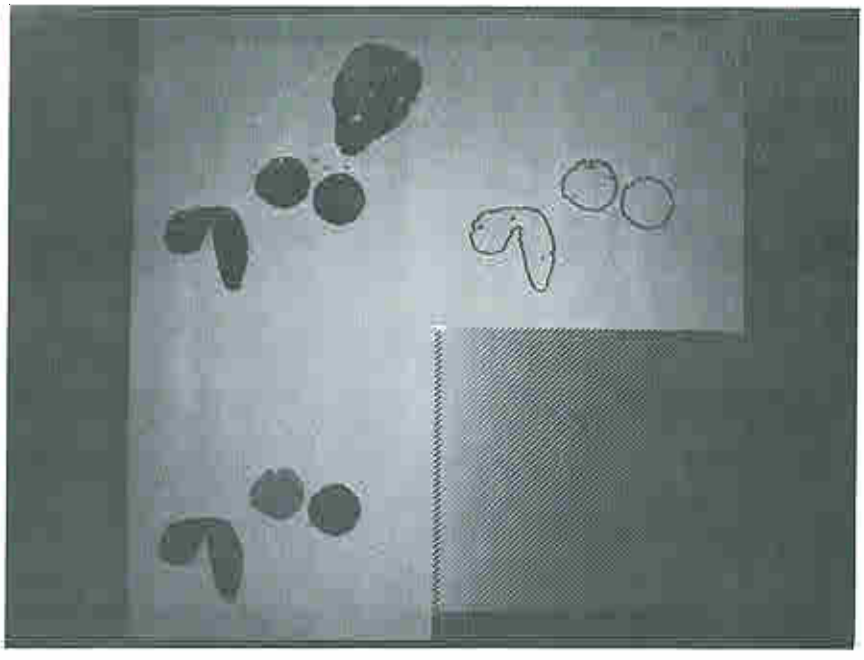


Figure E.5 (a) CS(green)-CS(blue) sliced at 60, (b) trace of boundaries of objects segmented, (c) connected components scaled by 15, (d) list space used by SEGBND.

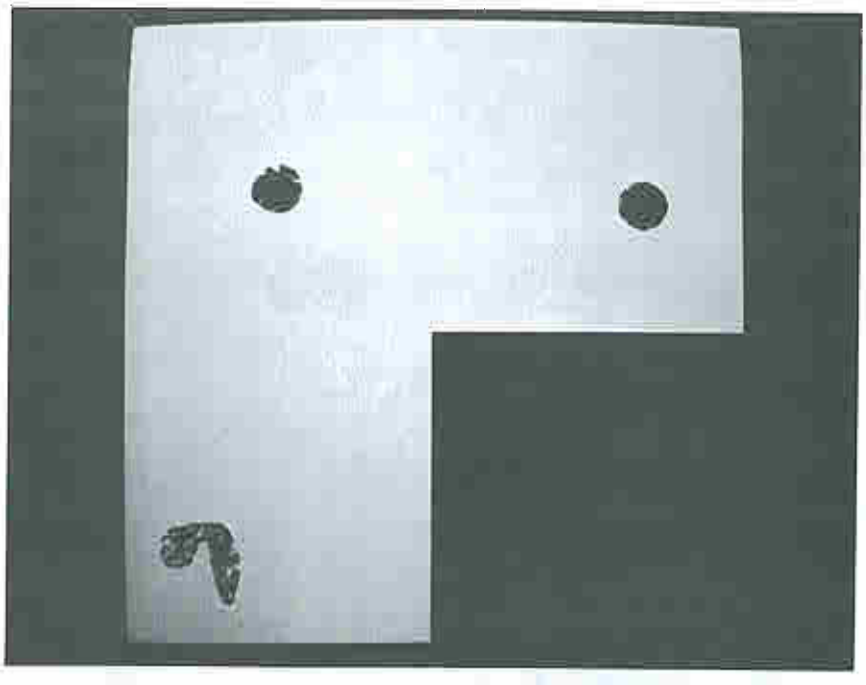


Figure E.6 (a) Connected component 2 masked with CS(green), (b) Connected component 3 masked with CS(green), (c) Connected component 4 masked with CS(green).

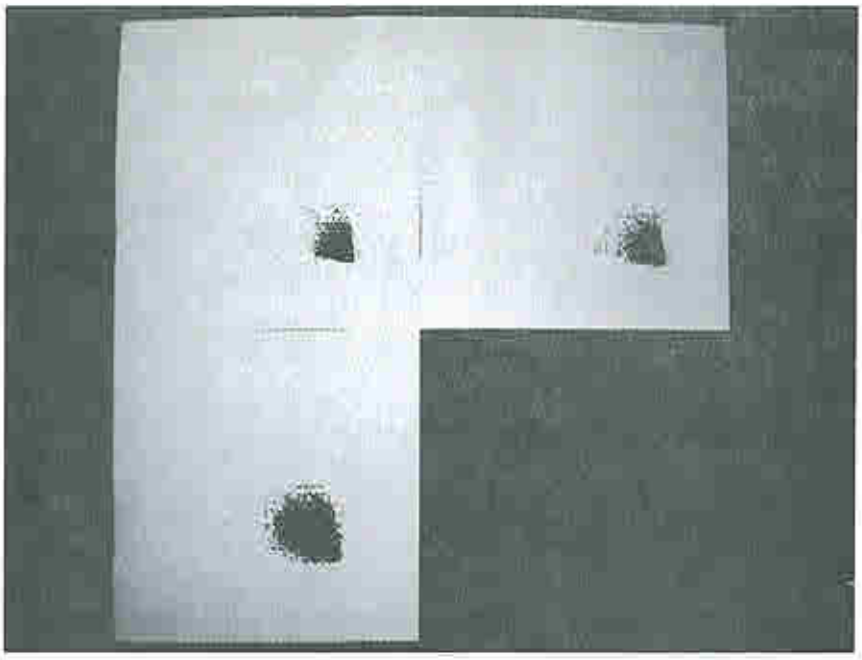


Figure E.7 (a) Joint gray scale distribution of connected component (CC) number 2 nucleus with itself taking gray values 3 pixels apart, (b) same for CC number 3, (c) same for CC number 4.

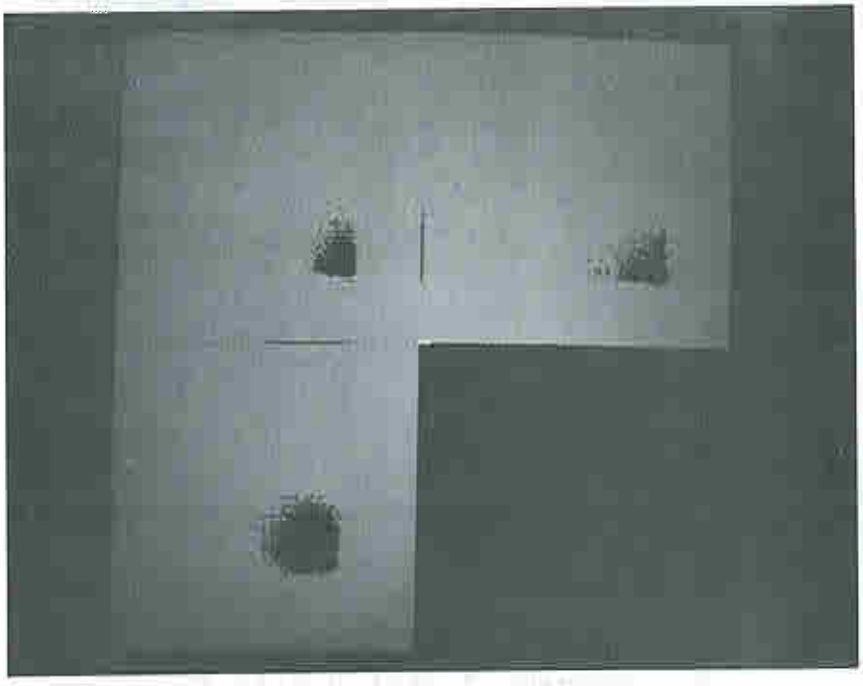


Figure E.8 (a) Joint gray scale distribution of connected component (CC) number 2 nucleus with itself taking gray values 7 pixels apart, (b) same for CC number 3, (c) same for CC number 4.

APPENDIX F

BMON2 DATA FILE FORMAT

A BMON2 data file consists of a file header followed by BMON2 data where the data format is described in the header. A block is 256 12-bit words or 384 8-bit packed bytes. Two 12-bit words are used to pack 3 bytes in OS8 as follows:

```
                 ------------------------------
word 1:          | byte 3 high 4 bits | byte 1|
                 ------------------------------
word 2:          | byte 3 low  4 bits | byte 2|
                 ------------------------------
```

```
-------------------------------------------
| Header | Variable length data segment   |
-------------------------------------------
1 block          n blocks
```

The header contains the following information:

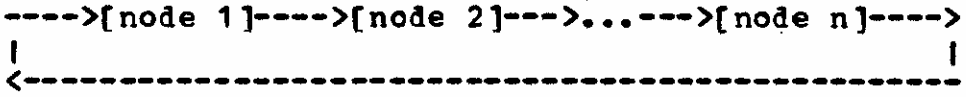
Header word number	field and function
1	(a) Data file space mode (7 to 15 decimal) - 1 word BM=7, MASK REG=10, QMT=11, BOUNDARY=12, GALSCAN=13, STATE=15, BINARY MASK=16, PROC10 VARIABLE SIZE IMAGE=17, PDP10 TRANSFORM DATA=18, LINE DRAWING=19.
2	(b) File length in blocks (0 means variable) - 1 word
3:4	(c) Number of data (0 means variable) - 2 words
5	(d) Number of words/datum (a fraction expressed by the numerator in the left 6-bit byte, and the denominator in the right 6-bit byte - 1 word.
6	(e) Data file submode number - 1 word. 1 = BM 8-bit packed (1 image) 2 = BM 16-bit packed (2 images) 3 = Mask register 4 = Quantimet function computers 5 = Boundary 10-bit, vector 6 = Boundary 10-bit, no vector 7 = Boundary 8-bit, vector 8 = Boundary 8-bit, no vector 9 = Galvanometer scanner 8-bit raster 10 = Galvanometer scanner 10-bit raster 11 = Galvanometer scanner 8-bit mask driven 12 = Galvanometer scanner 10-bit mask driven 13 = State 14 = Binary mask image 15 = PROC10 variable size image (see (r)) 16 = PROC10 36-bit data transform. 17 = Line drawing.
7	(f) Data length - 1 word (0=8-bit, 1=10-bit, 2=12 bit data, 3=16-bit, 4=1-bit)

8	(g)	Information type - 1 word (0=z data, 1=xy data, 2=xyz data (not used), 3=(x(e),x(x)) mask reg. data, 4=QMT function comp. data, 5=binary mask, 6=36bit PDP10 words)
9	(h)	Data packing mode - 1 word (0 means packed 8-bit, 1 means unpacked in 12-bit words).
10	(i)	OS/8 date word when file created - 1 word month - bits 0:3 day - bits 4:8 year (0 to 7) - bits 9:11.
11	(j)	Horizontal window coordinate before scan - 1 word
12	(k)	Vertical window coordinate before scan - 1 word
13:16	(l)	File name and extension (6-bit Ascii) - 4 words
17:52	(m)	72 character Comment (6-bit Ascii) with zero last character - 36 words
53:76	(n)	12-tuple double precision CURRENT position state vector, see MDPDATA[7:8,1:12] in COMMON - 24 words
77	(o)	gray value used for filling masks (8-bits lsb default is 255) - 1 word
78	(p)	Class key (0 if not used), 1 to 12 if used.
79:80	(q)	Time of day in two words packed 4A6.
81	(r)	Size of image for submode 15 as (2**n)-1
82	(s)	axiomat lens magnification as number (eg 100==>100X)
83:85	(t)	Microns/pixel conversion factor as PDP8e floating point number.
86:88	(u)	Zoom value (0.80x to 3.20x) as PDP8e floating point number.
89:90	(v)	Horizontal size, vertical size of F&S
91:92	(w)	<BMi1> LCS (Xlcs,Ylcs).
93:256		- currently unused but reserved.

APPENDIX G

BOUNDARY LIST DATA STRUCTURE

Boundaries are represented by circular linked lists containing (x,y) data. These lists are maintained in BM7 allocated for used as a linked list freestore. The freestore AVAIL pointer is kept in LSFREESTORE[1:2] in COMMON. The freestore will be reinitialized by an operation using it if the /I switch is specified in the operation using the FREESTORE or the INIT operation is performed previous to the operation. The pointer to the boundary data structure is in QRZ. A node consists of five 16-bit BM words. A list contains both backward and forward pointers. An optional right pointer is currently not used for lists but is reserved for trees and graphs. A list looks like:



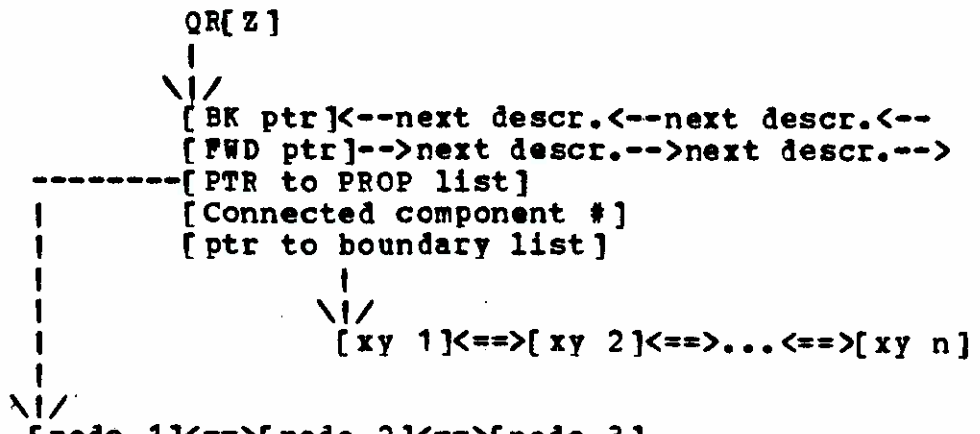
The field allocation of a node of five 16-bit words is:

```
[back pointer   ]
[forward pointer]
[right pointer  ]
[datum 1        ]
[datum 2        ]
```

A boundary node uses the last 3 fields as follows:

```
right ptr field: [4:15] Run Length Map (RLM) semantic
                        label, [1:3] Freeman type chain code.
datum 1 field: 8-bits of x coordinate data.
datum 2 field: 8-bits of y coordinate data.
```

A boundary descriptor list (which is pointed to by QRZ) allocates these fields as follows:



node 1
---

right ptr:	number of boundary points
datum 1:	true perimeter
datum 2:	area under the RLM

node 2
---

right ptr:	area filled of object under RLM
datum 1:	horizontal position of enclosing rectangle
datum 2:	vertical position of enclosing rectangle

node 3
---

right ptr:	density of object under RLM
datum 1:	horizontal size of enclosing rectangle
datum 2:	vertical size of enclosing rectangle

INDEX