

ANALYSES OF REQUIREMENTS FOR COMPUTER CONTROL  
AND DATA PROCESSING EXPERIMENT SUBSYSTEMS

IMAGE DATA PROCESSING SYSTEM  
IDAPS  
SOFTWARE DESCRIPTION (7094 VERSION)

VOLUME II

15 SEPTEMBER 1973

NASA-CR-124447) ANALYSES OF REQUIREMENTS  
FOR COMPUTER CONTROL AND DATA PROCESSING  
EXPERIMENT SUBSYSTEMS: IMAGE DATA  
PROCESSING SYSTEM (System Development  
Corp.) 134 p HC \$8.75

N73-33123

CSCL 09B

G3/08

Unclas  
15727

Reproduced by  
NATIONAL TECHNICAL  
INFORMATION SERVICE  
US Department of Commerce  
Springfield, VA. 22151

**SYSTEM DEVELOPMENT CORPORATION**

135

## TABLE OF CONTENTS

| <u>Subroutine</u> | <u>Page</u> | <u>Subroutine</u> | <u>Page</u> |
|-------------------|-------------|-------------------|-------------|
| AREAC             | 3           | FRXFM             | 39          |
| ARITH             | 6           | FRX2V             | 43          |
| ARROW             | 7           | FSIZE             | 49          |
| ASSIGN            | 8           | FUNGN             | 50          |
| BESJ              | 10          | FXTOFL            | 51          |
| CENTER            | 11          | HISTOW            | 52          |
| CKFMT             | 12          | HRDCPY            | 53          |
| CONV              | 13          | IDAPS2            | 55          |
| CONVRT            | 15          | IDENT             | 56          |
| CRLATE            | 13          | INIMAG            | 57          |
| DATAIN            | 16          | INIT              | 58          |
| DCONV             | 13          | INSERT            | 61          |
| DELCLS            | 58          | INSRT             | 62          |
| DISKC             | 18          | IPARAM            | 63          |
| DISKD             | 18          | ISOPLT            | 64          |
| DISKO             | 18          | IXFORM            | 13          |
| DISKR             | 21          | KONVL             | 66          |
| DISKW             | 21          | LABL              | 67          |
| DRIVTP            | 22          | MERG              | 101         |
| DSPLIN            | 24          | MESS              | 69          |
| DSPLN             | 25          | MLTAVS            | 70          |
| DSTANC            | 26          | MPTORI            | 72          |
| DUMPF             | 29          | MULDSP            | 73          |
| DUMPFD            | 30          | NOSFUG            | 75          |
| ERROR             | 31          | NSA               | 76          |
| FFIL              | 32          | OPA-OP66          | 58          |
| FLIP              | 34          | ORAND             | 79          |
| FLTOFX            | 35          | OTIMAG            | 57          |
| FRAME             | 36          | PACK              | 80          |
| FRAMWR            | 38          | PDGSA             | 81          |

TABLE OF CONTENTS (continued)

| <u>Subroutine</u> | <u>Page</u> | <u>Subroutine</u> | <u>Page</u> |
|-------------------|-------------|-------------------|-------------|
| PL6               | 119         | SPRED3            | 105         |
| PRERUN            | 83          | STAT              | 106         |
| PSFGEN            | 84          | STKPIC            | 107         |
| RDC               | 58          | SZCK              | 108         |
| READF             | 29          | TFGEN             | 109         |
| RESTOR            | 85          | TIME              | 112         |
| REWND             | 86          | TRANS             | 113         |
| RITOMP            | 72          | TRNPS             | 114         |
| RSLGEN            | 87          | TSFSZ             | 116         |
| SCAL              | 88          | UNPAC             | 117         |
| SCK               | 90          | UNPACK            | 118         |
| SEGMNT            | 91          | UPL6              | 119         |
| SHADE             | 95          | XFORM             | 13          |
| SHAVE             | 96          | XPAND             | 120         |
| SHRINK            | 97          |                   |             |
| SLICE             | 98          |                   |             |
| SPINT             | 99          |                   |             |
| SPLT              | 101         |                   |             |
| SPRED1            | 102         |                   |             |
| SPRED2            | 103         |                   |             |
|                   |             |                   |             |
| Appendix A .....  |             |                   | 122         |
| Appendix B .....  |             |                   | 124         |
| Appendix C .....  |             |                   | 130         |

This document contains a description of each of the software modules of the Image Data Processing System (IDAPS) developed under NASA contract NAS8-25471 during the period September 15, 1972 to September 15, 1973. Much of this software is derived from an earlier effort which was reported in SDC TM-(L)-HU-033/006/00 "ATM Experiment S-056 Image Data Processing System Software Development - Volume II". The changes which this document reflects are the result of additions to the application software of the system and an upgrade of the IBM 7094 Mod I computer to a 1301 disk storage configuration. The addition of the disk to the system eliminates most of the manual steps associated with storing intermediary data files on magnetic tape, thus greatly improving ease of use of the system.

The purpose of this document is to supply necessary information about IDAPS software to the computer programmer who desires to make changes in the software system or who desires to use portions of the software outside of the IDAPS system. Generally, each software module is documented in the following manner:

- Module Name
- Purpose
- Usage
- Common Block(s) Description
- Method (algorithm of subroutine)
- Flow Diagram (if needed)
- Subroutines Called
- Storage Requirements
- Required by: (IDAPS Operator)

A general description of the common blocks used by the IDAPS system is included as Appendix A to this document. Appendix B is an explanation of the diagnostic features of the system, and Appendix C describes in detail the procedure for adding new IDAPS operators to the system.

For the convenience of the user, subroutines herein documented are arranged in alphabetical order.

AREAC (Area Calculations)

**Purpose:** AREAC calculates the area on a sphere represented by pixels of a specific value. Aspect ratio is accounted for as a function of the distance to each pixel from the center of the sphere. Also calculated are the centroid of the area, the center of radiation intensity over the area, and the center of total released energy over the area.

**Usage:** CALL AREAC

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| IN(5)       | JIN - Input file containing subsetted data (output from SUBSET operator)   |
| SN(5)       | KIN - Original input image (used with MOMENT operator only)  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns   |
| K1          | CASE - Pixel value for which the area is calculated  |
| K2          | LC } - Line and column location of the center of the sphere<br>KC } with respect to the upper left corner of the original input image                  |
| K3          |  |
| K4          | LSF } - Location of the upper left corner of the data<br>KSF } frame being processed with respect to the upper left corner of the original input image |
| K5          |  |
| K6          | NTAB - Indicator of the type of aspect correction desired:   |

|    |   |
|----|---|
| K7 | NTAB = 0 - No aspect corrections are applied<br>NTAB = 1 - Corrections based on table supplied by user<br>NTAB = Other - Internal aspect ratio table used<br>IORB - Indicator of sphere<br>IORB = 1 - Earth<br>IORB = Other - Sun |
| K8 | IOP - Operator indicator<br>IOB = 1 - AREA calculated only<br>IOB = 2 - MOMENT package  |
| A1 | SPOT - Size in micrometers of scanning aperture used  |

Method:

The centroid of a thin plate on the surface of the hemisphere is located in rectangular dimensions by:

$$\bar{X}_s = \frac{\sum c \Delta s}{\sum \Delta s} \quad \bar{Y}_s = \frac{\sum g \Delta s}{\sum \Delta s} \quad \bar{Z}_s = \frac{\sum \ell \Delta s}{\sum \Delta s}$$

The values of  $c$  and  $\ell$  can be obtained by direct methods and the values of  $g$  can be computed as:

$$g = R^2 - \ell^2 - c^2$$

$\Delta s$  is the elemental area corresponding to one picture element and located by  $\ell$ ,  $c$ , and  $g$ .

For calculating the center of radiation intensity instead of the area, the above equations must be modified by replacing the elemental area  $s$  with an intensity factor. Thus:

$$\bar{X}_i = \frac{\sum c i}{\sum i} \quad \bar{Y}_i = \frac{\sum g i}{\sum i} \quad \bar{Z}_i = \frac{\sum \ell i}{\sum i}$$

Likewise, for the center of total released energy, both the rate of energy release (intensity) and the area over which it is released must be taken into account so that:

$$\bar{X}_e = \frac{\sum ci\Delta s}{\sum i\Delta s} \quad \bar{Y}_e = \frac{\sum gi\Delta s}{\sum i\Delta s} \quad \bar{Z}_e = \frac{\sum li\Delta s}{\sum \Delta s}$$

From the observer's point of view, the centroid of a feature on the sun may more easily be located by spherical coordinates of latitude and longitude ( $\bar{\phi}$  and  $\bar{\theta}$ ).

$$\bar{\phi} = \cos^{-1}(\bar{Z}/R) \quad \text{and} \quad \bar{\theta} = \tan^{-1}(\bar{Y}/\bar{X})$$

Converting the formulas for the centroid, center of intensity, and center of energy:

$$\bar{\phi}_s = \cos^{-1}\left(\frac{\sum l\Delta s}{R\sum \Delta s}\right) \quad \bar{\theta}_s = \tan^{-1}\left(\frac{\sum g\Delta s}{\sum c\Delta s}\right)$$

$$\bar{\phi}_i = \cos^{-1}\left(\frac{\sum li}{R\sum i}\right) \quad \bar{\theta}_i = \tan^{-1}\left(\frac{\sum gi}{\sum ci}\right)$$

$$\bar{\phi}_e = \cos^{-1}\left(\frac{\sum li\Delta s}{R\sum i\Delta s}\right) \quad \bar{\theta}_e = \tan^{-1}\left(\frac{\sum gi\Delta s}{\sum ci\Delta s}\right)$$

Subroutines: DISKR, DATAIN, CONVRT

System library routines: ARCOS, SQRT, ATAN

Storage: 12227<sub>(8)</sub> words

Required by: AREA, MOMENT

ARITH (Arithmetic Functions)

Purpose: ARITH performs the following point by point operations on two frames of data: A+B, A-B, A\*B, A/B. Also A+C, A\*C, C/A and A\*\*C, where C is a constant, may be performed.

Usage: CALL ARITH

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| IN(5)       | I1, I2 - Input file names for A and B  |
| ON          | IO - Output file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns   |
| K1          | KF - Operation flag<br>KF = 1 - A+B<br>KF = 2 - A-B<br>KF = 3 - A*B<br>KF = 4 - A/B<br>KF = 5 - A+C<br>KF = 6 - A*C<br>KF = 7 - C/A<br>KF = 8 - A**C |
| A1          | CN - Constant to be used if KF is greater than 4   |

Subroutines: DISKR, DISKW

Storage: 14243 (8) words

Required by: MATH

ARROW

Purpose: ARROW creates the image of an arrow in a 9x9 matrix.

Usage: CALL ARROW (MAT,INDIC)

MAT - The given 9x9 matrix

INDIC - Arrow direction indicator

INDIC = 1 - SW to NE

INDIC = 2 - SE to NW

INDIC = 3 - NW to SE

INDIC = 4 - NE to SW

Storage: 511<sub>(8)</sub> words

Required by: FRAME

ASSIGN (Assign Name)

**Purpose:** ASSIGN designates a unique name for every input, scratch, and output disk file for an operator.

**Usage:** CALL ASSIGN (I,IN,SN,ON,NF,LOF,CARD)

I - Operator index into array NF  
IN - Input file name array  
SN - Scratch file name array  
ON - Output file name  
NF - Operator parameter table containing the first three characters of each operator's name, and the number of input, scratch, and output files to be assigned for that operator  
LOF - Last opened output file  
CARD - Operator card image

**Method:** The operator card image is scanned for the presence of multiple or non-standard input file names enclosed in parentheses. If any are found, they are stored in array IN and replaced by blanks in the card image array. Otherwise, the file found in LOF is stored in IN(1). Standard output file names are established by the following convention:

The first three characters of the operator's name are preceded by an integer constant which represents the numerical order of the file's use. For example, the first time operator CONVOLVE is encountered, the output file is named 1CON; the second time CONVOLVE is encountered, the output file is named 2CON, etc.

Scratch files are named in a similar manner with the addition of the letter S before the operator name. For example, operator TRANSPOSE may have scratch files 1STRA, 2STRA, 3STRA, etc. The integer digit always begins at one for each operator's scratch files because the scratch files are deleted immediately after use.

15 September 1973

9

System Development Corporation  
TM- (L) -HU-033/008/00

Subroutines: UNPACK, ERROR

System library routines: CTOBIN, FLD, MOVST

Storage: 1305<sub>(8)</sub> words

BESJ (Bessel Function Generator)

**Purpose:** BESJ generates all Bessel functions of the first kind having integral order and real argument subject to the restrictions below.

**Usage:** CALL BESJ (X, N, BJ, D, IER)

X - Argument of the J Bessel function desired  
 N - Order of the J Bessel function  
 BJ - Resultant Bessel function  
 D - Required accuracy  
 IER - Error code  
     IER = 0 - No error  
     IER = 1 - N is negative  
     IER = 2 - X is negative or zero  
     IER = 3 - Required accuracy not obtained  
     IER = 4 - Range of N compared to X not correct

**Method:** The Bessel functions of the first kind of integral order are computed by recurrence relation techniques described by H. Goldstein and R. M. Thaler, Recurrence Techniques For The Calculation Of Bessel Functions, MTAC, V. 13, pp. 102-108 and A. Stegun and M. Abramowitz, Generation of Bessel Functions On High Speed Computers, MTAC, V. 11, 1957, pp. 255-257.

**Restrictions:**  $N < 20. + 10. x - x^{2/3}$   
                   for  $x \leq 15.$   
 $N < 90. + x/2.$   
                   for  $x > 15.$

**Storage:** 456 (8) words

CENTR (Center)

Purpose:           CENTR multiplies each pixel of an input image by  $(-1)^{i+j}$  where  $i$  is the line position of the point and  $j$  is its column position.

Usage:           CALL CENTR

Common block COM1 parameters:

| Common name | Local name                  |
|-------------|-----------------------------|
| IN (5)      | IN - Input file name        |
| ON          | IO - Output file name       |
| L           | NL - Number of lines        |
| C           | NC - Number of columns      |
| K1          | KF - Data flag              |
|             | KF = 1 - Integer            |
|             | KF = Other - Floating point |

Restrictions:   The number of columns in the image must be less than or equal to 1024.

Subroutines:   DISKR, DISKW

Storage:       2206 (8) words

Required by:   CENTER

CKFMT (Check Format)

**Purpose:** CKFMT compares the format of an input file for an operator with the format required by that operator, and converts the input format if necessary.

**Usage:** CALL CKFMT (KIF)

KIF - Input file format desired

KIF = 0 - Floating point

KIF = 1 - Integer

**Common block COM1 parameters:** (See Appendix A)

**Method:** The format of each input file is determined from the disk directory. If format conversion is necessary, a temporary input file is opened. The name of the original input file is saved in array PN, while the temporary file name replaces it in array IN. The converted data is stored in the temporary file and the original input file is closed.

**Subroutines:** DISKO, DISKC, IDENT, DISKR, DISKW

**System library routine:** SSWTCH

**Storage:** 12246<sub>(8)</sub> words

CONV (Convolution Driver)

Purpose: CONV has five entry points which perform the following two-dimensional Fourier transform operations on an input image:

CONV - Convolution  
 DCONV - Deconvolution  
 CRLATE - Correlation  
 XFORM - FFT  
 IXFORM - IFFT

Usage:

CALL CONV (IP,MSIZE,IN,NTSIZE,KFLAG,II)  
 CALL DCONV (IP,MSIZE,IN,NTSIZE,KFLAG,II)  
 CALL CRLATE (IP,MSIZE,IN,NTSIZE,KFLAG,II)  
 CALL XFORM (IP,MSIZE,NTSIZE,NOR,II)  
 CALL IXFORM (IP,MSIZE,NTSIZE,II)

IP - Point spread function file name  
 MSIZE - Size of point spread function  
 IN - Input image file name  
 NTSIZE - Size of output file  
 KFLAG - Point spread function control flag  
     KFLAG = 1 - Point spread function is being input  
                   in frequency domain  
     KFLAG = Other - Point spread function is being  
                   input in spatial domain  
 II - Scratch disk file name

Common block COM1 parameters:

| Common name | Local name                                       |
|-------------|--|
| CARD(13)    | CARD1(6), RMAG, PHA, CARD9(5) - Input card image |
| IN(5)       | IA - Input disk file names                       |
| SN(5)       | IW, IWW, SN3, IWWW, SN5 - Scratch file names     |
| ON          | IO - Output file name                            |
| L           | NSIZE - Size of input file                       |

Notes: If the point spread function is input in the spatial domain (KFLAG  $\neq$  1), an extra call is made to FRX2V to transform and normalize the PSF.

Restrictions: The point spread function and input image arrays must be square.

Subroutines: FRX2V

Storage: 3456<sub>(8)</sub> words

Required by: FFT, IFFT, CORRELATE, CONVOLVE, DECONVOLVE

CONVRT (Convert to Degrees)

**Purpose:** CONVRT converts a floating point number which represents an angle to integer degrees, minutes and seconds.

**Usage:** CALL CONVRT (HDG, IDEG, MIN, ISEC)

HDG - Number to be converted

IDEG - Degrees

MIN - Minutes

ISEC - Seconds

**Storage:** 110<sub>(8)</sub> words

DATAIN (Data Input)

**Purpose:** DATAIN reads free field, tabular data from cards and stores it in a specified array.

**Usage:** CALL DATAIN (X,N,K)

X - Array into which data will be stored

N - Number of data values to be input

K - Data format

K = 0 - Integer

K = 1 - Floating point

**Method:** Data is input in the following general format:

A = 9.,8.1,6.3,4.2,2.7,1.6,.8,.09,12\*0.\$

DATAIN looks for the equal sign and isolates each number separated by commas until a \$ is encountered at which point the input scan is terminated.

Multiple values may be input by using an asterisk, as shown in the example above where 12\*0. indicates that 12 zeros are to be input following the value .09.

Integer values are input without decimal points.

Continuation cards are required if data cannot be contained in columns 1 - 78 of the first card. Continuation cards must start in column 2 with column 1 blank.

A check is made for the correct number of input values, as well as the correct format of the data.

**Subroutines:** UNPACK, PACK, INSERT

**System library routines:** CTOBIN, MOVST, SSWTCH

**Storage:** 1021 (8) words

15 September 1973

17

System Development Corporation  
TM-(L)-HU-033/008/00

Required by: ALTER, DEPENDENT ALTER, PSF GENERATOR, FEATHER

DISKO (Disk Open)

- Purpose:** DISKO has three entry points:
- DISKO - Opens a disk file and initializes the file directory
  - DISKC - Closes a disk file
  - DISKD - Deletes a disk file
- Usage:**
- CALL DISKO (ID, NL, NC, KD, KU, JU)
  - CALL DISKC (ID)
  - CALL DISKD (ID)
- ID - File identification
  - NL - Number of lines
  - NC - Number of columns
  - KD - Data type
    - KD = 0 - Floating point
    - KD = 1 - Integer (6-bit)
  - KU - Disk unit number to be assigned at time file initially opened (0 or 1)
  - JU - Disk unit number that was assigned to a previously opened file
- Method:** A disk file may be in one of the following states:
- Open and active - The file has been opened but not closed or deleted: it can be written or read.
  - Open and inactive - The file has been opened and closed but not deleted: it must be reopened before written or read.
  - Deleted - The file has been opened, closed, and deleted: it cannot be written or read.

When a file is initially opened, the following entries for as many as 100 files are made in the file directory:

IDT - File identification  
NLT - Number of lines in the input image  
NCT - Number of columns per line  
KST - Starting track number  
NLPT - Number of lines per track  
NT - Number of tracks  
KDT - Data type  
    KDT = 0 - Floating point  
    KDT = 1 - 6-bit integer packed  
KP - Pointer to buffer  
    KP = 0 - Buffer closed  
    KP = 1-7 - Buffers 1-7  
KUT - Unit specification (0 or 1)  
NAF - Number of assigned files  
NOB - Number of open buffers  
LTU - Last track used - LTU(1) = Unit 0, LTU(2) = Unit 1  
ITKADR - Track address of data currently in buffer (KP)

In addition to the entries made in the file directory, when a file is initially opened a core buffer may be set aside for it. A buffer is required if the number of columns per line requires less than 233 words of storage. Thus, an integer file will use a buffer if a line contains less than 1393 columns since each pixel is 6 bits and 6 pixels are packed into a 36-bit word. The use of buffers reduces the number of accesses required when reading or writing disk, since a full track of data is read into core or written from core for each read or write operation. For example, to read or write a 128 by 128 integer image requires only 7 disk accesses. The following table portrays the layout of various sized images when stored on the disk.

| Number of lines<br>per image | Number of columns<br>per line | Number of lines<br>per track<br>(integer format) | Number of tracks<br>per image |
|------------------------------|-------------------------------|--|-------------------------------|
| 1024                         | 1024                          | 2  | 512                           |
| 512                          | 512                           | 5  | 103                           |
| 256                          | 256                           | 10   | 26                            |
| 128                          | 128                           | 21   | 7                             |
| 64                           | 64                            | 43   | 2                             |

Buffers are used only during an operator's execution. Each active file is either closed or deleted following execution of the operator.

Input file(s) - Closed

Scratch file(s) - Deleted

Output file - Closed

If an output file is to be used as the input file for a succeeding operation, it must be reopened.

When a file is closed, the assigned buffer is released but the entries in the file directory for the closed file and the file itself on disk are not changed. When a file is deleted, the buffer is released, the entries for the deleted file in the file directory are deleted, and the file is purged from the disk.

Subroutines: IDENT

System library routines: SSWTCH, EXIT

Storage: 3134<sub>8</sub> words

DISKR (Disk Read)

Purpose: DISKR contains two entry points:

DISKR - Reads a section of an image from disk  
DISKW - Writes a section of an image to disk

Usage:

CALL DISKR (ID, LN, LL)

CALL DISKW (ID, LN, LL)

ID - File identification

LN - Line index. If negative, write backwards (i.e., bottom  
to top of image)

LL - Line to be read or written

Method:

Pixels are stored on the disk in packed format (6 characters/  
word) for integer files and unpacked format (1 character/word)  
for floating point files. Lines are buffered when the  
column size is less than 1393 (integer) or 233 (floating point).

Subroutines: UPL6, PL6, IDENT

System library routines: SSWTCH, DTAKE, DCHEK, DPUT, EXIT

Storage: 11310<sub>(8)</sub> words

DRIVTP (TRNPS Driver)

**Purpose:** DRIVTP calculates the variable dimensions of the core buffer used in subroutine TRNPS, and calls TRNPS with the correct parameters.

**Usage:** CALL DRIVTP (REVOLV)

REVOLV - Rotation option indicator  
 REVOLV = 1 - TRANSPOSE  
 REVOLV = 2 - +90 DEGREE ROTATE  
 REVOLV = 3 - -90 DEGREE ROTATE  
 REVOLV = 4 - MINOR TRANSPOSE  
 REVOLV = 5 - VERTICAL ROTATE  
 REVOLV = 6 - HORIZONTAL ROTATE  
 REVOLV = 7 - 180 DEGREE ROTATE

Common block COM1 parameters:

| Common name | Local name  |
|-------------|---|
| L           | NL - Number of lines  |
| C           | NC - Number of columns  |
| K1          | FMT - Format of the data<br>FMT = 1 - Integer<br>FMT = Other - Floating point |

**Method:** The maximum number of columns which can be in core is determined by dividing a single 10240 word array by the number of lines in the input image. This calculated dimension, INDEX, is passed to TRNPS where the core buffer is dimensioned dynamically (INDEX, NL).

**Subroutines:** TRNPS

**Storage:** 24157<sub>(8)</sub> words

15 September 1973

23

System Development Corporation  
TM-(L)-HU-033/008/00

Required by: TRANSPOSE, +90 DEGREE ROTATE, -90 DEGREE ROTATE, MINOR TRANSPOSE,  
VERTICAL ROTATE, HORIZONTAL ROTATE, 180 DEGREE ROTATE

DSPLIN (Display Line)

**Purpose:** DSPLIN converts a string of BCD characters into a 9x171 array for display of alphanumeric characters on the Photowriter or DICOMED display unit.

**Usage:** CALL DSPLIN (IN, NC, KC, KOUT)

- IN - String of characters to be converted to display format
- NC - Number of characters in string to be converted, excluding trailing blanks
- KC - Starting column position in output array
- KOUT - Output array (9x171 packed format)

**Method:** Each BCD character in the input string is interpreted and a corresponding 9x7 bit configuration is placed in the output array.

**Restrictions:** A maximum of 102 characters can be converted for display at one time.

**Subroutines:** UNPAC

**System library routines:** FLD

**Storage:** 22566<sub>(8)</sub> words

**Required by:** LABEL

DSPLN (Display One Line at a Time)

**Purpose:** DSPLN converts a string of BCD characters into an array for display of alphanumeric characters on the Photowriter or DICOMED display. This subroutine is used for the line-at-a-time output of labels required by the FRAME and MULTIPLE DISPLAY operators.

**Usage:** CALL DSPLN (IN,NC,KC,OUT,TP,NUM,LN)

IN - String of characters to be converted for display  
in BCD format  
NC - Number of characters in string to be  
converted  
KC - Starting column position in output array  
OUT - Output array unpacked  
TP - Maximum number of columns in the output array  
NUM - Index to first character in the input array  
LN - Line number (1-9)

**Method:** Each BCD character in the input string is interpreted and a corresponding bit configuration is placed in the output array. DSPLN must be called in a loop, 9 times, with LN incremented each time, to effect the correct 9x7 bit character configuration.

**Restrictions:** A maximum of 80 characters can be converted for display at one time.

System library routines: FLD

Storage: 436<sub>(8)</sub> words

Required by: FRAME, MULTIPLE DISPLAY

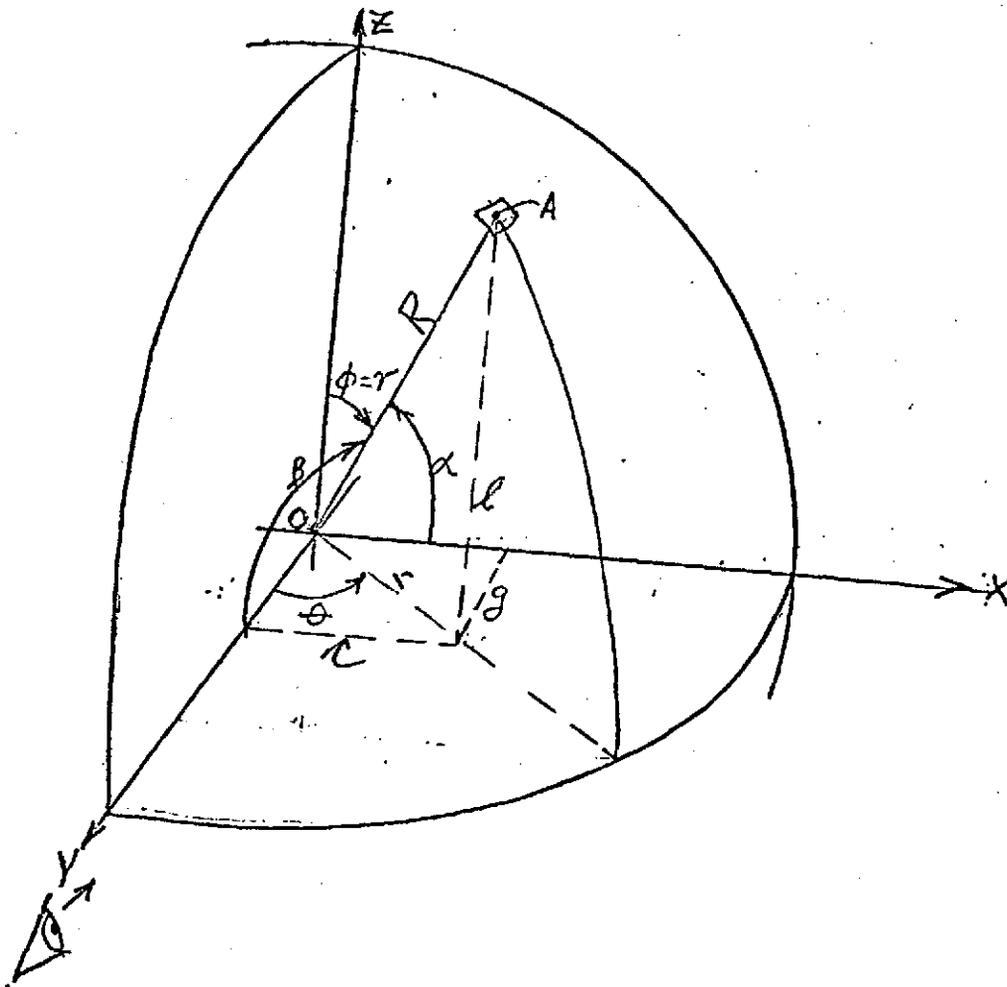
DSTANC (Distance)

**Purpose:** DSTANC calculates the distance on a sphere from one point to another, given the latitude and longitude of each point.

**Usage:** CALL DSTANC

**Common block COM1 parameters:** (See Appendix A.)

**Method:**



A point (A) is located on the surface of a hemisphere of radius  $R$ . Two sets of coordinates are chosen to describe the location of A - spherical coordinates  $\theta$ ,  $\phi$ , and  $R$ , and rectangular coordinate angles  $\alpha$ ,  $\beta$ , and  $\gamma$ .

$\phi$  = The angle that OA makes with the Z axis.

$\theta$  = The angle between the projection of OA on the XY plane and the Y axis.

$\alpha$  = The angle between OA and the X axis.

$\beta$  = The angle between OA and the Y axis.

$\gamma$  = The angle between OA and the Z axis.

If the sphere is viewed at a great distance by an observer situated on the Y axis, then all points on the hemisphere will appear as a perpendicular projection on the X/Z plane. All lines which are parallel to a radius vector in the X/Z plane will appear in true perspective while all others will appear foreshortened by some aspect ratio.

From the viewer's observations, the following calculations can be made:

$R = 1/2$  the diameter of the apparent solar disk

$$\phi = \cos^{-1} \left( \frac{\ell}{R} \right)$$

$$\theta = \sin^{-1} \left( \frac{c}{R \sin \phi} \right) = \sin^{-1} \left[ \frac{c}{\sqrt{R^2 \ell^2}} \right]$$

$$\alpha = \cos^{-1} \frac{c}{R}$$

$$\beta = \sin^{-1} \frac{\sqrt{c^2 + \ell^2}}{R}$$

$$\gamma = \phi = \cos^{-1} \frac{\ell}{R}$$

$$\alpha = \cos^{-1} (\sin \phi \sin \theta) \quad \phi = \gamma$$

$$\beta = \cos^{-1} (\sin \phi \cos \theta) \quad \text{and} \quad \theta = \cos^{-1} \left( \frac{\cos \beta}{\sin \gamma} \right)$$

$$\gamma = \phi$$

For two vectors  $0 \rightarrow A_1$  and  $0 \rightarrow A_2$ , the cosine of the angle between them is:

$$\cos \angle \begin{matrix} OA_1 \\ OA_2 \end{matrix} = \cos \alpha_1 \cos \alpha_2 + \cos \beta_1 \cos \beta_2 + \cos \gamma_1 \cos \gamma_2$$

Translating this to spherical coordinates:

$$\begin{aligned} \angle_{OA_2}^{OA_1} &= \cos^{-1} (\sin\phi_1 \sin\phi_2 \sin\theta_1 \sin\theta_2 + \sin\phi_1 \sin\phi_2 \\ &\quad \cos\theta_1 \cos\theta_2 + \cos\phi_1 \cos\phi_2) \text{ or} \\ &= \cos^{-1} (\sin\phi_1 \sin\phi_2 (\sin\theta_1 \sin\theta_2 + \cos\theta_1 \cos\theta_2) + \cos\phi_1 \cos\phi_2) \end{aligned}$$

since,  $\sin\theta_1 \sin\theta_2 + \cos\theta_1 \cos\theta_2 = \cos(\theta_1 - \theta_2)$

Then:

$$\angle_{OH_2}^{OH_1} = \cos^{-1} (\sin\phi_1 \sin\phi_2 (\cos(\theta_1 - \theta_2)) + \cos\phi_1 \cos\phi_2)$$

and the distance between  $OA_1$  and  $OA_2$  is  $R \left[ \angle_{OA_2}^{OA_1} \right]$ .

Subroutines: CONVRT

System library routines: CTOBIN, SIN, COS, ARCOS, SQRT

Storage: 514<sub>(8)</sub> words

Required by: DISTANCE

DUMPF (Dump File)

Purpose: DUMPF has two entry points:

DUMPF - Dumps a disk file to tape

READF - Loads a file from tape to disk

Usage: CALL DUMPF  
CALL READF

Common block COM1 parameters:

| Common name | Local name                 |
|-------------|----------------------------|
| IN (5)      | IN - Input disk file names |
| ON          | ON - Output disk file name |
| L           | NL - Number of lines       |
| C           | NC - Number of columns     |
| K1          | KT - Tape unit number      |
| K2          | KF - File number           |

Subroutines: DISKR, DISKW

System library routines: SKFBIN

Storage: 4275<sub>(8)</sub> words

Required by: DUMP FILE, LOAD FILE

DUMPFED (Dump File Directory)

Purpose: DUMPFED saves a current copy of the disk file directory on the disk.

Usage: CALL DUMPFED (LOF,NF)

LOF - Last opened output file

NF - Operator parameter table

System library routines: DPUT, DCHEK

Storage: 2743<sub>(8)</sub> words

ERROR

Purpose: ERROR causes the printing of a message on the on-line printer.  
After a pause, the next card is read in from the card reader.

Usage: CALL ERROR (CARD)

CARD - Array of 13 words containing the packed (6 ch/wd)  
card image

Storage: 103<sub>(8)</sub> words

FFIL (Fourier Filter)

Purpose: FFIL filters an input image in the frequency domain.

Usage: CALL FFIL

Common block COM1 parameters:

| Common name | Local name  |
|-------------|---|
| IN (5)      | IN (1) - Image file name<br>IN (2) - Filter file name |
| SN (5)      | SN - Scratch files 1 through 5                        |
| ON          | ON - Output file                                      |
| L           | NL - Number of lines                                  |
| C           | NC - Number of columns                                |

Method: The following sequence of operations is performed:

- 1) The image is transformed using subroutine FFT.
- 2) The transform is converted from real/imaginary to magnitude/phase format.
- 3) The magnitude terms of the transform are multiplied by corresponding terms of the filter.
- 4) The resulting transform in magnitude/phase is converted back to real/imaginary.
- 5) The transform is inverse transformed.

Restrictions:

- 1) The filter must be the same size as the image.
- 2) The image and filter must be square.

15 September 1973

33

System Development Corporation

TM-(L)-HU-033/008/00

Subroutines: FRX2V, DISKR, DISKW

Storage: 3362<sub>(8)</sub> words

Required by: FOURIER FILTER

FLIP (Vertical Rotate)

Purpose: FLIP inverts the order of the elements in a line of data so that column NC becomes column 1, column (NC-1) becomes column 2, etc.

Usage: CALL FLIP (NC,IBUF)  
NC - Number of columns in a line  
IBUF - Input buffer for one line

Storage: 2070<sub>(8)</sub> words

Required by: TRANSPOSE, +90 DEGREE ROTATE, -90 DEGREE ROTATE, MINOR TRANSPOSE, HORIZONTAL ROTATE, VERTICAL ROTATE, 180 DEGREE ROTATE

FLTOFX (Floating to Fixed Point Conversion)

Purpose: FLTOFX converts an input image from floating point to fixed point format.

Usage: CALL FLTOFX

Common block COM1 parameters:

| Common name | Local name             |
|-------------|------------------------|
| IN(5)       | IN - Input file name   |
| ON          | IO - Output file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns |

Method: Each pixel within the image is rounded to the nearest whole number and is output in integer format.

Subroutines: DISKR, DISKW

Storage: 10147 (8) words

Required by: FIX

FRAME (Standardized Frame Mask)

Purpose: FRAME provides a standard frame for a processed image consisting of the following information:

- 1) Gray scale step wedge
- 2) Tic marks along the edges of the image
- 3) Optional grid
- 4) Optional labels, both above and below the image
- 5) Histogram of gray scale distribution
- 6) Border around the image
- 7) Optional arrow to point at an area of interest
- 8) Optional inversion of gray scale values

Usage: CALL FRAME

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| CARD(13)    | KARD(13)   |
| IN(5)       | ID - Input file  |
| L           | TNL - Number of lines  |
| C           | TNC - Number of columns  |
| K1          | TL - Upper label<br>TL = 0 or blank - No top label<br>TL > 0 - Number of lines in top label                    |
| K2          | BL - Bottom label<br>BL = 0 or blank - No bottom label<br>BL > 0 - Number of lines in bottom label             |
| K3          | G - Grid option<br>G = 0 or blank - No grid<br>G = 1 - Automatic grid<br>G > 1 - User supplied grid increments |
| K4          | OP1 - Gray scale inversion option which inverts the image only   |

|    |   |
|----|---|
| K5 | OP2 - Gray scale inversion option which inverts the entire output frame   |
| K6 | AL - Arrow point line coordinate  |
| K7 | AC - Arrow point column coordinate  |
| K8 | INDIC - Arrow direction indicator. If 0, then SW to NE direction is used unless image borders indicate otherwise.<br>INDIC = 1 - SE to NE<br>INDIC = 2 - SE to NW<br>INDIC = 3 - NW to SE<br>INDIC = 4 - NE to SW |

Method: The designated information is overlaid on the input image, one line at a time, and written onto an output tape with the gray scale wedge lined up vertically along the right side of the image and the histogram below all specified bottom labels. The completed frame width will be TNC + 106.

Restrictions: (1) The input image may not exceed 2048 x 2048.  
(2) Only one arrow may be specified.  
(3) The labels are centered, one 80 column card image per line, over the entire width of the framed image.  
(4) The label may not exceed (TNC + 106)/10 characters.

Subroutines: FRAMWR, HISTOW, ARROW, DISKR, DSPLN

Storage: 7022<sub>(8)</sub> words

Required by: FRAME

FRAMWR (Frame Write)

**Purpose:** FRAMWR packs a buffer and writes it on the output tape, A6, a designated number of times (NUM). If the gray scale inversion option is indicated, the inversion is performed before the record is written.

**Usage:** CALL FRAMWR (RECORD,LEN,IOP,NUM,PL,PACKED)

RECORD - The record to be written out on tape

LEN - Length of the record

IOP - Gray scale inversion option indicator

IOP = 0 - No inversion

IOP = Other - Invert

NUM - The number of times this record is to be written

PL - The packed line buffer

PACKED - The length of the packed buffer

**Subroutines:** PL6

**System library routines:** WRITER

**Storage:** 4127<sub>(8)</sub> words

**Required by:** FRAME

FRXFM (Fast Fourier Transform Routine)

**Purpose:** FRXFM computes the Fast Fourier Transform of a complex vector using a Radix 4 plus 2 algorithm.

**Usage:** CALL FRXFM (N2POW,X,Y)

N2POW -  $\log_2 N$ , where N is the number of elements in the complex array

X - Array containing the real part of the complex vector  
Y - Array containing the imaginary part of the complex vector

**Notes:** The array need be dimensioned only in the calling routine. If a real array is to be transformed, set the Y array elements to zero and proceed as usual.

The transform is performed in place; therefore, the input data is destroyed as the transform is computed.

**Method:** The routine is a modified implementation of a procedure originally developed by Cooley and Tukey and later improved by Gentleman and Sande.

If N is the size of the vector to be transformed, then the equations describing discrete Fourier transformations are given by:

$$X(j) = \sum_{k=0}^{N-1} A(k) \exp\left(\frac{i2\pi jk}{N}\right) \quad j = 0, 1, \dots, N-1 \quad (1)$$

$$A(k) = \sum_{j=0}^{N-1} X(j) \exp\left(\frac{-i2\pi jk}{N}\right) \quad k = 0, 1, \dots, N-1 \quad (2)$$

If  $W = \exp\left(\frac{i2\pi}{N}\right)$  and  $W^\alpha = W^{nk}$ , then equation (1) can be written in matrix form as

$$\begin{bmatrix} X(0) \\ X(1) \\ \cdot \\ \cdot \\ \cdot \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & \dots & W^0 \\ W^0 & W^1 & & W^{N-1} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ W^0 & W^{N-1} & \dots & X^{(N-1)^2} \end{bmatrix} \begin{bmatrix} A(0) \\ A(1) \\ \cdot \\ \cdot \\ \cdot \\ A(N-1) \end{bmatrix} \quad (3)$$

or  $X = TA$  (4)

In general, equation (4) requires  $N^2$  multiplications and divisions. If, however,  $N$  has  $M$  factors  $r_1, r_2, \dots, r_m$  such that  $N = \prod_{i=1}^M r_i$ , then  $T$  can be decomposed into the product of  $m$  elementary transformations, followed by a permutation  $P$  of the result:

$$T = P T_m T_{m-1} \dots T_1 \quad (5)$$

Each step  $T_j$  is in turn composed of  $N/r_j$  transforms of dimension  $r_j$ . Since the number of arithmetic operations for a transform of dimension  $r_j$  is of the order  $r_j^2$ , the total number of operations for the transformation  $T$  is less than  $N \sum_{j=1}^m r_j = mNr$  but  $N = r^m$  or  $\log_r N = m$ , therefore computation requires  $r_N \log_r N$  arithmetic operations compared with the  $N^2$  operations required by a standard transform. Also by reduction of arithmetic operations, accuracy is improved.

The recursive equations needed to implement the Fast Fourier Transformation routine for arbitrary length records ( $r = 2^\alpha$  for all integer  $\alpha$ ) were derived by G. D. Bergland and are shown below.

$$\begin{aligned}
 & A_p(j_0, j_1, \dots, j_{p-1}, k_{n-p-1}, \dots, k_0) \\
 &= \sum_{k_{n-p}=0}^{r_p-1} A_{p-1}(j_0, j_1, \dots, j_{p-2}, k_{m-p}, \dots, k_0) \cdot \\
 & W_N [j_{p-1}(r_1 r_2 \dots r_{p-1}) + \dots + j_0] k_{m-p} (r_{p+1} \dots r_m) \quad (6)
 \end{aligned}$$

The last entry thus calculated will yield the Fourier sum as

$$X(j_{m-1}, \dots, j_0) = A_m(j_0, \dots, j_{m-1}) \quad (7)$$

in such an order that the index of an X must have its binary bits put in reverse order to yield its index in the array  $A_m$ .

In some cases the computation required to evaluate equation (6) can be reduced by regrouping the equation as

$$\begin{aligned}
 & \hat{A}_p(j_0, j_1, \dots, j_{p-1}, k_{m-p-1}, \dots, k_0) \\
 &= \left[ \sum_{k_{n-p}=0}^{r_p-1} \hat{A}_{p-1}(j_0, j_1, \dots, j_{p-2}, k_{m-p}, \dots, k_0) W_{r_p}^{j_{p-1} k_{m-p}} \right] \cdot \\
 & W_N^{j_{p-1} (k_{n-p-1} (r_{p+2} \dots r_m) + \dots + k_1 k_n + r_0) (r_1 r_2 \dots r_{p-1})} \quad (8)
 \end{aligned}$$

The bracketed term represents a set of  $r_p$ -point Fourier transforms and the complex exponential weights outside the brackets reference each set of results to a common time origin.

The term

$$W_{r_p} = W_N^{N/r_p} = e^{2\pi i/r_p}$$

forms the basis for the complex exponential weights required in evaluating each  $r_p$ -point transform, and  $j_{p-1}, k_{m-p}$  are the indices of the transform.

If the  $r_1, r_2, \dots, r_m$  are all equal to some integer, say  $\alpha$ , then the transformation described by the resulting recursive equations is referred to as a Radix  $\alpha$  transformation. Cooley and Tukey stated in their original paper that the FFT algorithm is formally most efficient when the number of samples in a record is a power of 3 (i.e.,  $r_j = 3$ ) but such an algorithm is only slightly more efficient than a Radix 2 or a Radix 4 routine. Later it was recognized that the symmetries of the sine and cosine weighting functions made the Radix 4 algorithm more efficient than either the Radix 2 or Radix 3 algorithm. Radix 4 algorithms, however, suffer from the severe limitation on acceptable record sizes. To overcome this difficulty Gentleman and Sande developed an algorithm which performs as many iterations of the transform as possible with a Radix 4 routine, and then, if required, performs the last iteration with a Radix 2 routine. A modified version of Gentleman and Sande's algorithm is implemented as the subroutine FRXFM.

If  $N$  is the size of the array such that  $N = 2^{N2POW}$ , then  $N2POW$  is divided by two (integer division) to compute the required number of transformations of dimension four ( $N4POW$ ).

Restrictions: The dimension ( $N$ ) of the input array must be a power of two. If it is not, zeros may be used to buffer it up to a length which is a power of two.

System library routines: COS, SIN

Storage: 1224<sub>(8)</sub> words

Required by: FFT, IFFT, CORRELATE, CONVOLVE, DECONVOLVE, FOURIER FILTER

FRX2V (Two-dimensional Fast Fourier Process Driver)

**Purpose:** FRX2V performs a two-dimensional Fourier transformation by calling the fast Fourier subroutine FRXFM first by rows then by columns. FRX2V also serves as the main processor for the convolution, deconvolution, correlation, transform, normalized transform, and inverse transform processes.

**Usage:** CALL FRX2V (NSIZE, NPSIZE, IN, IW, IO, IM, IAPPLY, IP, II, RMAG, PHA, ITRUNC, TMAX, TMIN)

NSIZE - Input image dimension  
NPSIZE - Output image dimension  
IN - Input image disk file name  
IW - Scratch disk file name  
IO - Output image disk file name  
IM - Forward/Inverse transformation flag  
IM = 0 - Forward transformation  
IM = 1 - Inverse transformation  
IAPPLY - Process control flag  
IAPPLY = 1 - Convolve  
IAPPLY = 2 - Deconvolve  
IAPPLY = 3 - Transform and normalize  
IAPPLY = 4 - Correlate  
IAPPLY = Other - Transform  
IP - Transformed point spread function disk file name  
II - Scratch disk file name  
RMAG - Limit on magnitude amplification -- See SPRED2  
PHA - Limit on phase amplification -- See SPRED2  
ITRUNC - Truncation flag  
ITRUNC = 1 - No truncation  
ITRUNC = 2 - Truncate values outside of TMIN and TMAX  
ITRUNC = Other - Truncate values outside of 0. and 63.

TMAX - Maximum truncation limit

TMIN - Minimum truncation limit

## Method:

A two-dimensional Fourier transformation is achieved by first performing a one-dimensional transformation of each of the rows of the original array and then performing a one-dimensional transformation of each of the columns of the array resulting from the first set of transformations. To better understand this procedure consider the continuous two-dimensional transform of equation (1).

$$h(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(X,Y) e^{-12\pi ux} e^{-12\pi vy} dx dy \quad (1)$$

If an interval T is divided into N equal parts both horizontally and vertically such that  $N\Delta X = N\Delta Y = T$ , then the maximum frequency  $F = \frac{1}{\Delta X} = \frac{1}{\Delta Y}$ . The incremental frequencies are then

$\Delta U = \Delta V = F/N$  so that  $\Delta U\Delta X = \Delta V\Delta Y = \frac{1}{N}$ . Using the indicies j,k, l,m equation (1) in the discrete case becomes

$$h(l\Delta u, m\Delta u) = \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} H(j\Delta x, k\Delta y) e^{\frac{-2\pi i}{N} j l} e^{\frac{-2\pi i}{N} k m} \quad (2)$$

Let

$$TR(j,m) = \sum_{k=0}^{N-1} H(j\Delta x, k\Delta y) e^{\frac{-2\pi i}{N} k m} \quad (3)$$

then

$$h(l\Delta u, m\Delta u) = \sum_{j=0}^{N-1} TR(j,m) e^{\frac{-2\pi i}{N} j l} \quad (4)$$

but a one-dimensional discrete Fourier transform over an interval T, sampled N times ( $N\Delta X = T$  and  $F = \frac{1}{\Delta X}$ ,  $\Delta U = F/N$ ) is given by

$$T(M\Delta X) = \sum_{k=0}^{N-1} V(k) e^{\frac{-2\pi i}{N} k m}$$

Therefore, equation (3) is just N ordinary one-dimensional transforms, one for each row of the picture, and equation (4) is N ordinary one-dimensional transforms, one for each of the columns resulting from the N applications of equation (3).

**Image buffering:** If the input picture(s) and/or point spread function, have dimensions less than the smallest power of two greater than NPSIZE they are buffered with zeros equally on all sides up to this power. Indeed, for convolution, deconvolution, or correlation this is necessary in order to prevent image overlapping due to the cyclic nature of the Fast Fourier Transform. In general if M is the size of the input picture and L is the size of the point spread function, the NPSIZE should be at least the smallest power of two greater than M + L.

**Point spread function transformation:**

If the PSF is input in the spatial domain, an extra call to FRX2V is made by CONV to transform and normalize the PSF (IAPPLY = 3). The normalization consists of summing the gray levels of all elements of the PSF and dividing each frequency element (both real and imaginary) by this amount. This procedure insures that a unit amount of light will be dispensed for each unit present in the input picture.

**Image Centering:**

The transform of the PSF (real and imaginary) is also multiplied by  $(-1)^{i+j}$ ,  $i=1, 2, \dots, NPSIZE$ ;  $j=1, 2, \dots, NPSIZE$ , for convolution and deconvolution so that the output picture will be centered in the frame.

**Clipping:**

Due to improper amplification of certain frequency terms due to noise, improper PSF, etc., the gray levels obtained from deconvolution may extend over a large dynamic range. The subroutine will determine the maximum and minimum values encountered and will print these on-line. If the

64 levels were divided over this large range one would obtain an output image with little contrast. A much improved image will result if the output is clipped both above and below, with the remaining levels quantized to 64 gray levels. Presently the subroutine clips at 0. and 63.; that is, all values below zero are set to 0. and all values above 63 are set to 63. The remaining range is truncated to 64 gray levels.

Data storage: The figure on the next page represents the data storage layout for a 128 by 128 frame of data. Since the whole picture will not fit into core at the same time, the frame is divided into four sections and each section processed in turn. Frames of data which are 256 by 256, and 512 by 512 are similarly handled to minimize data transfer.

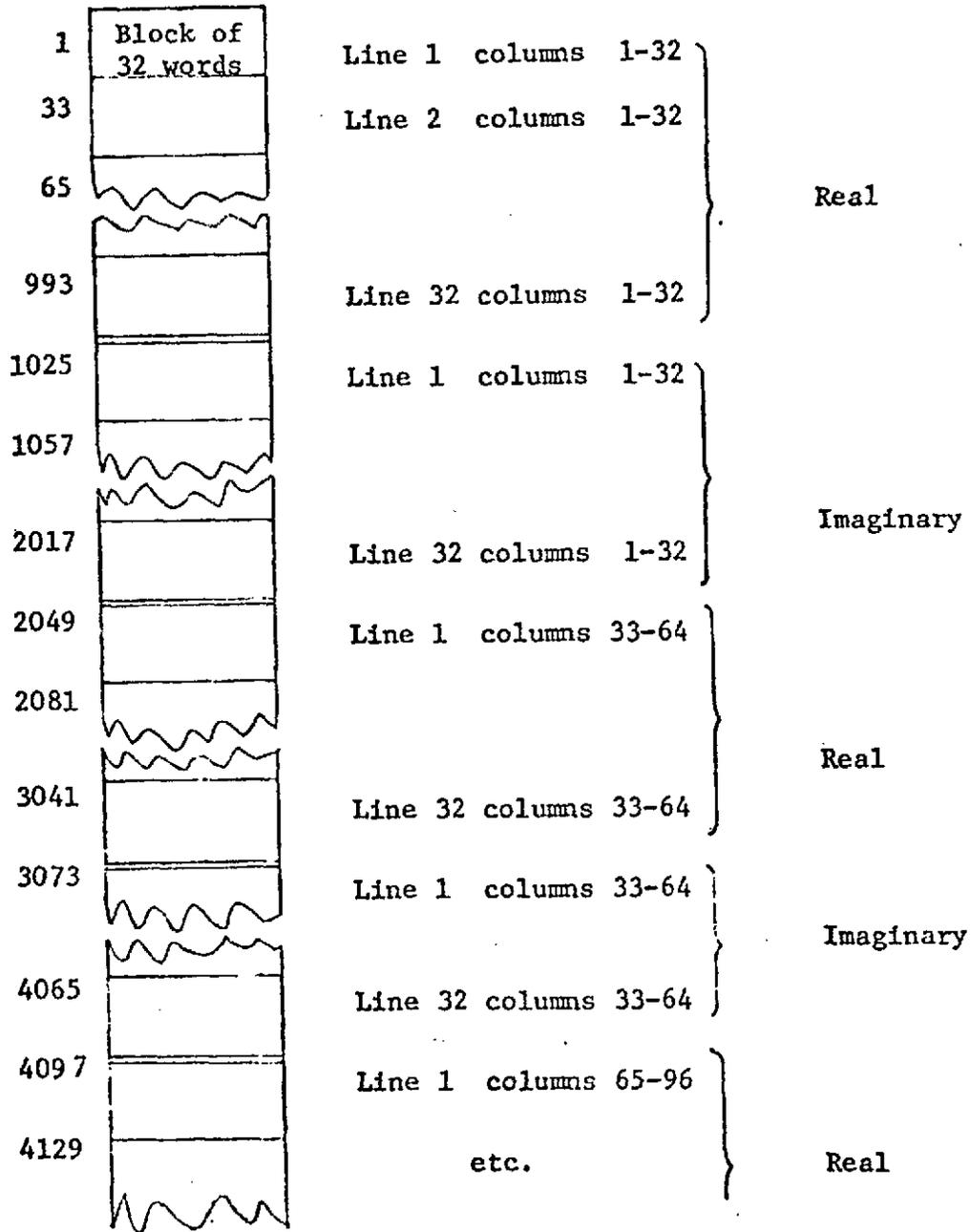
Restrictions:

- 1) The image size must be greater than or equal to 16 by 16 and must be less than or equal to 512 by 512. This should be considered the maximum output image size.
- 2) The output image dimension must be square and must be greater than or equal to NSIZE.
- 3) The inverse operator assumes the input image is square.

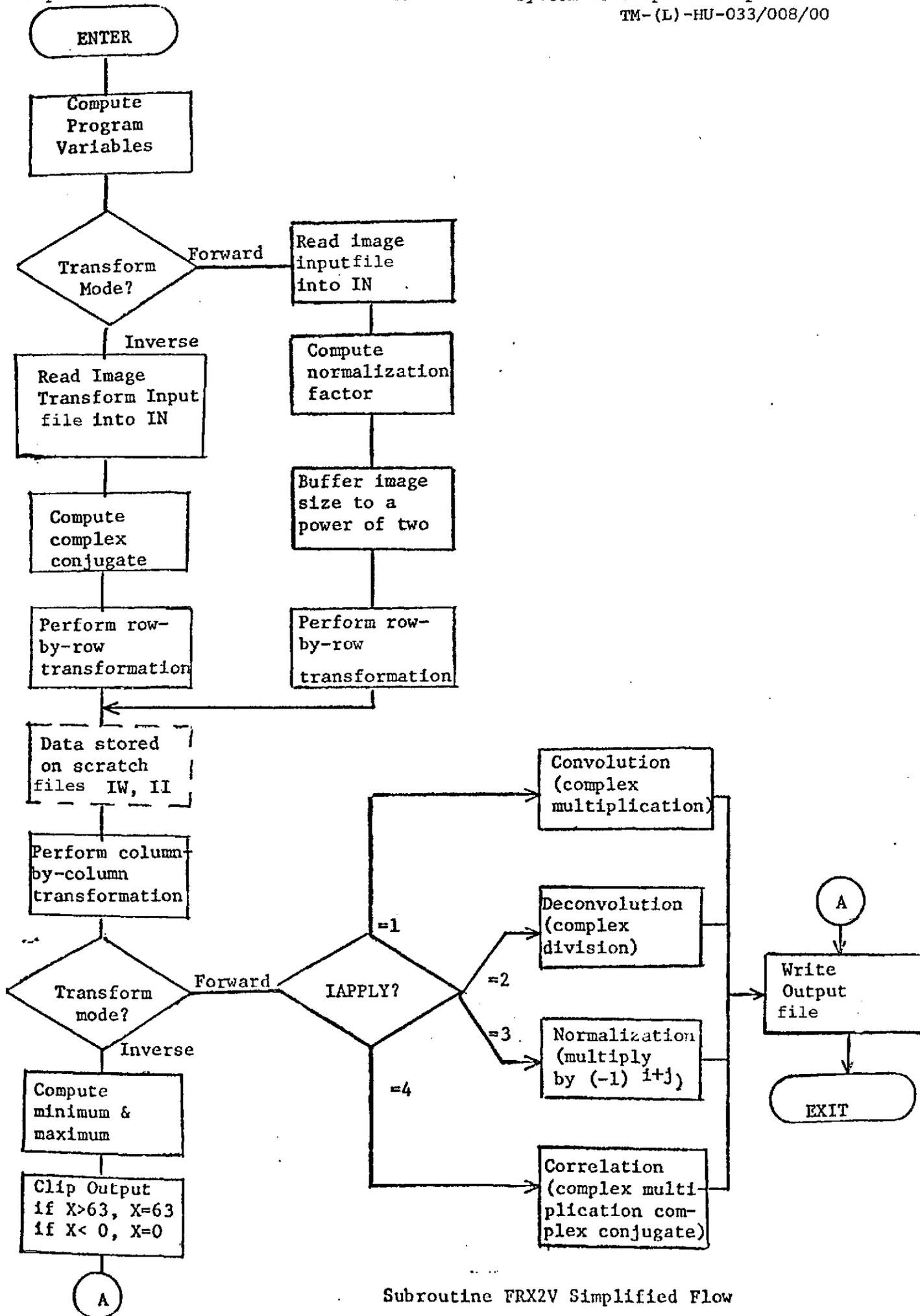
Subroutines: DISKR, DISKW, FRXFM, SPRED1, SPRED2, SPRED3

Storage: 25,520<sub>8</sub> words

Required by: FFT, IFFT, CORRELATE, CONVOLVE, DECONVOLVE, FOURIER FILTER



128 x 128 Data Storage Layout



Subroutine FRX2V Simplified Flow

FSIZE (File Size)

Purpose: FSIZE calculates the size of the scratch file needed by the two-dimensional FFT operator, according to the following table:

| Image Output Size | Scratch File Size |        |
|-------------------|-------------------|--------|
|                   | Records           | Length |
| 32 x 32           | 1                 | 1024   |
| 64 x 64           | 1                 | 4096   |
| 128 x 128         | 16                | 1024   |
| 256 x 256         | 256               | 256    |
| 512 x 512         | 4096              | 64     |
| 1024 x 1024       | 65536             | 16     |

Usage: CALL FSIZE (NPSIZE, NLINES, NCOLMS)

NPSIZE - Output Image Size

NLINES - Number of lines in scratch file

NCOLMS - Number of columns in scratch file

Storage: 112<sub>(8)</sub> words

FUNGN (Function Generator)

Purpose: FUNGN produces a symmetrical function in a square array from an input curve.

Usage: CALL FUNGN

Common block COM1 parameters:

| Common Name | Local Name   |
|-------------|--|
| ON          | IOUT - Output file name  |
| L           | L - Number of lines  |
| C           | C - Number of columns  |
| K1          | K1 - Input option<br>K1 = 1 - Input three horizontal line segments<br>K1 = Other - Input curve profile |

Method: The distance from the center of the array is computed for each point in a square array. The distance is used to interpolate a value from the input curve, which is placed into the square array. For distance values outside the range of the input curve corners, the point in the square array is set to the last value found in the curve.

Subroutines: DATAIN, DISKW

System library routines: LAGRNG, COS, SQRT

Storage: 10644 (8 words)

Required by: FUNCTION GENERATOR

FXTOFL (Fixed to Floating Point Conversion)

Purpose: FXTOFL converts an input image from fixed point format to floating point format.

Usage: CALL FXTOFL

Common block COM1 parameters:

| Common name | Local name             |
|-------------|------------------------|
| IN(5)       | IN - Input file name   |
| ON          | IO - Output file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns |

Subroutines: DISKR, DISKW

Storage: 10150<sub>(8)</sub> words

Required by: FLOAT

HISTOW (Histogram Write)

Purpose: HISTOW generates a histogram from an input array of gray scale totals for output by the FRAME operator.

Usage: CALL HISTOW (GSD, TP, NQ, IOP, PL, PACKED)

GSD - Array of totals for the 64 gray scale values

TP - Total number of columns in the final frame

NQ - Buffer used for generating the gray scale indicators and tics

IOP - Gray scale inversion option indicator

PL - Packed line buffer

PACKED - Length of PL

Subroutines: FRAMWR, DSPLN

Storage: 675 (8 words

Required by: FRAME

HRDCPY (Hardcopy Printout of Images)

**Purpose:** HRDCPY prints a frame of image data in 1) fixed or floating point format, 2) sixteen levels of gray, or 3) a map of values above a specified threshold.

**Usage:** CALL HRDCPY

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| IN(5)       | JIN - Input file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns   |
| K1          | NO - Control flag<br>NO = -4 - Print gray level values on-line in floating point<br>NO = -3 - Print gray level values off-line in floating point<br>NO = -2 - Prints gray level values on-line in integer format<br>NO = -1 - Prints gray level values off-line in integer format<br>NO = 0 to 63 - Prints blanks for values less than or equal to NO and X for values greater than NO<br>NO = 64 - Prints 16 levels of gray using overstrike method |
| K2          | ITOP - Starting line number  |
| K3          | JLFT - Starting column number  |
| K4          | LINES - Number of lines to be printed  |
| K5          | KLMNS - Number of columns to be printed  |

Method: When printing gray level output, HRDCPY divides each gray level value (integer values 0 to 63) by four, adds one and then prints up to nine overstrikes of selected characters to produce one of sixteen different gray levels.

Subroutines: DISKR

Storage: 15104 (8 words

Required by: PRINT

IDAPS2 (Executive Control For IDAPS)

Purpose: IDAPS2 is the main control program for the system and performs, or controls the performance of, the following functions:

- 1) Initialization of the system
- 2) Input and interpretation of IDAPS operators and their associated parameters
- 3) Assignment of input, output, and scratch file names
- 4) Calling the necessary subroutines to execute the operator
- 5) Deletion of the assigned scratch files at the end of execution of each operator
- 6) Closing of input and output files

Common block COM1 parameters: (See Appendix A.)

Subroutines: INIT, SCK, OP1-OP66, OPA-OPH, XPAND, SHADE, PDGSA, MLTAVS, DCONV, RSLGEN, HRDCPY, PSFGEN, CONV, XFORM, IXFORM, CRLATE, SHAVE, FXTOFL, FLTOFX, SPINT, VARY, MAC, LABL, CENTR, MERG, SPLT, RITOMP, MPTORI, ARITH, SCAL, NOSFUG, INIMAG, OTIMAG, STKPIC, MULDSP, FRAME, REWND, DRIVTP, SEGMNT, DUMPF, READF, ISOPLT, ORAND, DELCLS, STAT, INSRT, RESTOR, SHRINK, AREAC, TFGEN, DSTANC, FUNGN, KONVL, SLICE, FFIL, NSA, TIME, DUMPF, DISKD, RDC

Storage: 4572 (8) words

IDENT (Identify File Index)

Purpose: IDENT is a function which returns the index to the disk file directory for a specified disk file name.

Usage: INDEX = IDENT (ID)

ID - Disk file name

INDEX - Pointer to disk file directory

Storage: 1663<sub>(8)</sub> words

INIMAG (Input Image)

Purpose: INIMAG has two entry points:  
 INIMAG - Reads an input image in scanner format from  
 tape unit A5.  
 OTIMAG - Writes a file onto tape unit A6 in display format.

Usage: CALL INIMAG  
 CALL OTIMAG

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| IN(5)       | IN - Input file name   |
| ON          | ON - Output file name  |
| L           | L - Number of lines  |
| C           | C - Number of columns  |
| K1          | KF - File number on input tape (for entry INIMAG)<br>Complement flag (for entry OTIMAG)<br>KF = 1 - Complement<br>KF = Other - No complement |
| K2          | SL - Starting line   |
| K3          | SC - Starting column   |

Subroutines: UPL6, DISKW, DISKR, PL6

System library routines: SKFBIN, SKRBIN, REDTPR, WRITER

Storage: 5046<sub>(8)</sub> words

Required by: INPUT IMAGE, OUTPUT IMAGE

INIT (Initialize Operator)

Purpose: INIT contains the following entry points:

INIT - Initializes

RDC - Read card

DELCLS - Delete/Close disk files

OPA-OP66 - Individual operator parameter and disk file  
manipulation

|        |                                | <u>Purpose</u>                    |
|--------|--------------------------------|-----------------------------------|
| Usage: | CALL INIT (I,LOF,NF,NO,KREP)   | System Initialization             |
|        | CALL RDC (I,LOF,NF,NO,KREP)    | Read and interpret operator card  |
|        | CALL DELCLS (I,INX,SNX,PNX,NF) | Delete and/or close disk files    |
|        | CALL OPA (J,K)                 | Initializes ALTER, FIX, FLOAT     |
|        | CALL OPB (J)                   | " DEPENDENT ALTER, FEATHER, SPLIT |
|        | CALL OPC (J)                   | " BAR CHART, PSF GENERATOR        |
|        | CALL OPD                       | " R/I TO M/P, M/P TO R/I          |
|        | CALL OP1                       | " SIZE                            |
|        | CALL OP2                       | " EXPAND                          |
|        | CALL OP5                       | " AVERAGE                         |
|        | CALL OP6                       | " DECONVOLVE                      |
|        | CALL OP7                       | " STOP                            |
|        | CALL OP9                       | " PRINT                           |
|        | CALL OP11                      | " EXTRACT                         |
|        | CALL OP12                      | " CONVOLVE                        |
|        | CALL OP13                      | " FFT                             |
|        | CALL OP14                      | " IFFT                            |
|        | CALL OP15                      | " CORRELATE                       |
|        | CALL OP19                      | " SMOOTH                          |
|        | CALL OP20                      | " COMMENT                         |
|        | CALL OP24                      | " LABEL                           |
|        | CALL OP25                      | " CENTER TRANSFORM                |
|        | CALL OP26                      | " MERGE                           |
|        | CALL OP30                      | " MATH                            |

|                        |             |   |
|------------------------|-------------|---|
| CALL OP31              | Initializes | SCALE   |
| CALL OP33              | "           | NON-SYMMETRICAL PSF                                   |
| CALL OP34              | "           | INPUT IMAGE   |
| CALL OP35              | "           | OUTPUT IMAGE  |
| CALL OP36              | "           | STACK   |
| CALL OP37              | "           | MULTIPLE DISPLAY                                      |
| CALL OP38              | "           | FRAME   |
| CALL OPF               | "           | TRANSPOSE, +90 ROTATE, -90 ROTATE,<br>MINOR TRANSPOSE |
| CALL OPG               | "           | VERTICAL ROTATE, HORIZONTAL ROTATE,<br>180 ROTATE     |
| CALL OP47              | "           | SUBSET  |
| CALL OP48              | "           | DUMP FILE   |
| CALL OP49              | "           | LOAD FILE   |
| CALL OP50 (KQ)         | "           | ISOGRAM   |
| CALL OPH               | "           | DELETE FILES  |
| CALL OP53 (NF, ID, NO) | "           | INCLUSIVE OR, AND                                     |
| CALL OP54              | "           | HISTOGRAM   |
| CALL OP55 (KREP)       | "           | REPEAT  |
| CALL OP56              | "           | INSERT  |
| CALL OP57 (KQ1)        | "           | RESTORE   |
| CALL OP58              | "           | SHRINK  |
| CALL OP59              | "           | AREA, MOMENT  |
| CALL OP61 (A, B, E, D) | "           | MTF   |
| CALL OP63              | "           | FUNCTION GENERATOR                                    |
| CALL OP64              | "           | FILTER  |
| CALL OP65              | "           | SLICE   |
| CALL OP66              | "           | FOURIER FILTER  |

## I - Operator Index

LOF - Last opened output file name

NF - Operator parameter table

NO - Number of operators in NF

KREP - Repeat flag

INX - Input file table

SNX - Scratch file table

ONX - Output file name

A,B,E,D - Independent variable parameters for MTF operator  
J - Format desired in files  
K - Variable format  
KQ - Logical variable used with ISOGRAM operator  
JD - Array used with DELETE FILES operator  
KQ1 - Logical variable used with RESTORE operator

Common block COM1 parameters: (See Appendix A.)

Method: The entry for each individual operator handles reading any parameters into the block COM1, and the opening of all required input, scratch, and output disk files.

Subroutines: ASSIGN, CKFMT, FSIZE, TSFSZ, DISKD, DISKO, IDENT, PRERUN, DISKC, IPARM.

System library routines: SSWTCH, STRTMR, FLD, MOVST, CTOBIN

Storage: 11342<sub>8</sub> words

INSERT (Insert Character)

Purpose: INSERT is used by subroutine DATAIN to insert a character into a string of characters.

Usage: CALL INSERT (CHAR, STRING, POS)

CHAR - Character to be inserted

STRING - String into which character is to be inserted

POS - Position in string to insert character

System library routines: FLD

Storage: 100<sub>(8)</sub> words

INSRT (Insert Array)

**Purpose:** INSRT inserts a sub-frame into a specified position in a larger file.

**Usage:** CALL INSRT

**Common block COM1 parameters:**

| Common Name | Local Name   |
|-------------|--|
| IN(5)       | IN1 - Name of larger disk file<br>IN2 - Name of disk file to be inserted   |
| ON          | ON - Output disk file name   |
| L           | NL - Number of lines in large file   |
| C           | NC - Number of columns in large file   |
| K1          | KL - Number of lines in sub-frame  |
| K2          | KC - Number of columns in sub-frame  |
| K3          | SL - Starting line to start insertion  |
| K4          | SC - Starting column to start insertion  |
| K5          | KF - Edge modification flag<br>KF = 1 - Smooth edges<br>KF = 2 - Border around sub-frame<br>KF = Other - No modification |

**Subroutines:** DISKR, DISKW

**Storage:** 21016 8) words

**Required by:** INSERT

IPARAM (Parameter Check)

**Purpose:** IPARAM is a function which checks the format of all parameters on an input operator card. If the formats are correct, the parameters are converted to binary and stored in the common parameter area (K(1)-K(8), KV, A1, A2) in COM1. The function returns a -2 if any format discrepancies are discovered, a 0 if all parameters are correctly input.

**Usage:** IERR = IPARAM (CDNO,JFMT,NO)

CDNO - Index to packed input card  
 JFMT - Expected parameter format  
       JFMT = 0 - Integer  
       JFMT = 1 - Floating point  
 NO - Number of consecutive parameters  
       to be checked  
 IERR - Parameter check  
       IERR = 0 - All parameters input correctly  
       IERR = -2 - Format error discovered

Common block COM1 parameters:

| Common name | Local name                |
|-------------|---------------------------|
| CARD(13)    | CARD - Packed card image  |
| IN(5)       | IN - Input disk files     |
| L           | L - Number of lines       |
| C           | C - Number of columns     |
| K1-K8       | K(9) - Integer parameters |
| KV          |                           |
| A1          | A12(2) - Real parameters  |
| A2          |                           |

Subroutine: UNPACK

System library routines: CTOBIN

Storage: 365 (8 words)

ISOPLT (Isodata Plot)

Purpose: ISOPLT produces an isodata plot of the gray values from an input image.

Usage: CALL ISOPLT (SMOOTH)

SMOOTH - Option flag

SMOOTH = .FALSE. - No spatial integration

SMOOTH = .TRUE. - Spatial integration (averaging)  
is performed on each pixel of  
input image.

Common block COM1 parameters:

| Common name | Local name             |
|-------------|------------------------|
| IN (5)      | IDP - Input file name  |
| ON          | IDO - Output file name |
| L           | NL - Number of lines   |
| C           | NC - Number of colums  |

Method:

|      |      |      |
|------|------|------|
| i(7) | i(8) | i(1) |
| i(6) | x    | i(2) |
| i(5) | i(4) | i(3) |

The above is an example of the numbering scheme used in the iso-data contouring routine for identifying the eight neighbors (i) of element x.

The entire picture is smoothed by replacing each neighbor with the average of it and its eight neighbors, i.e.,

$$x = \frac{x + \sum_{n=1}^8 i(n)}{9}$$

Each of the eight neighbors i (1 through 8) are searched to find at least one which is in the same contour plane or in a higher level plane as the center element (x).

Each of the four neighbors (2, 4, 6, and 8) are searched to find at least one which is in a lower contour plane than the center element (x).

Elements for which the neighbor conditions stated above are found to be true are marked as border elements and all others are set to zero.

Subroutines:           DATAIN, DISKR, DISKW

Storage:               20774<sub>(8)</sub> words

Required by:           ISOGRAM

KONVL (Convolve)

**Purpose:** KONVL applies a 3 by 3 moving window filter to an image by numerically convolving the input array with the 3 by 3 filter.

**Usage:** CALL KONVL

Common block COM1 parameters:

| Common name | Local name               |
|-------------|--------------------------|
| IN (5)      | JIN - Input file name    |
| ON          | JOUT - Output file name  |
| L           | NLIN - Number of lines   |
| C           | NCLM - Number of columns |

**Method:** With filter W (J,K), input array I(C,L) and output array O(C,L) where C and L are the column and line coordinates respectively:

$$O(M,N) = \sum_{j=1}^3 \sum_{k=1}^3 W(j,k) I(m-2+j, n-2+k)$$

**Subroutines:** DATAIN, DISKR, DISKW

**Storage:** 20407 (8) words

**Required by:** FILTER

LABL (Label)

Purpose: LABL inserts labels, which are read from input cards, into an image.

Usage: CALL LABL

Common block COM1 parameters:

| Common name | Local name                                |
|-------------|---|
| L           | NL - Number of lines in output frame      |
| C           | C - Number of columns (must be 1024)      |
| K1          | NP - Number of label lines to be inserted |
| K2          | JIN - Input tape                          |
| K3          | JOUT - Output tape                        |

Method: A packed display format tape is read down to the line at which the label is to be inserted. The label is overlaid and the process repeated until all labels have been inserted. The format of the label specification is:

|      |      |      |      |      |
|------|------|------|------|------|
| Col. | Col. | Col. | Col. | Col. |
| 6    | 12   | 18   | 25   | 72   |

|   |   |   |                    |
|---|---|---|--------------------|
| J | K | L | (Any desired label |
|---|---|---|--------------------|

J - Number of characters in the label  
 K - Starting line position on the image  
 L - Starting column position

The label begins in column 25 and continues for J columns. If the label is longer than 48 characters, it is continued on a second card beginning in column 1.

Each character is a 9x7 matrix with a 9x3 matrix of trailing blanks; therefore, each character occupies 9 lines and 10 columns. Line spacing is controlled by the user - a minimum of 5 lines between labels is recommended.

The character set is:

0 through 9, = ' - ( ) . , \* / , A through Z

The label is added to the original image so that the labels essentially overlay the image thus allowing overprinting, if desired.

Subroutines: DSPLIN

System library routines: REDTPR, WRITER

Storage: 3724<sub>(8)</sub> words

Required by: LABEL

MESS (Message)

Purpose: MESS is used by the diagnostics package to store and output specific error diagnostics.

Usage: CALL MESS (FLAG)

FLAG - Message indicator

Storage: 426<sub>(8)</sub> words

MLTAVS (Multi-Copy Average)

Purpose: MLTAVS computes the average of up to five images.

Usage: CALL MLTAVS

Common block COM1 parameters:

| Common name | Local name                          |
|-------------|-------------------------------------|
| IN(5)       | T - Input file names                |
| ON          | O - Output file name                |
| L           | L - Number of lines                 |
| C           | M - Number of columns               |
| K1          | S - Number of images to be averaged |

Method: For corresponding pixels of the images to be averaged, the mean gray scale value is calculated. The standard deviation is computed as

$$s = \sqrt{\frac{\sum_{i=1}^n x^2 - \frac{(\sum_{i=1}^n x_i)^2}{n}}{n-1}}$$

Those gray scale values outside 1 standard deviation are eliminated and a new mean value is calculated using only the gray scale values within 1 standard deviation of the original mean.

Restrictions:

- (1) A maximum of 5 images may be averaged
- (2) The maximum size of the images to be averaged is 2048 by 2048.

15 September 1973

71

System Development Corporation  
TM-(L)-HU-033/008/00

Subroutines: DISKR, DISKW

Storage: 3014<sub>(8)</sub> words

Required by: AVERAGE

MPTORI (M/P to R/I)

Purpose: MPTORI contains two entry points:

MPTORI - Converts magnitude/phase complex file  
(interleaved line by line) to a real/  
imaginary complex file (interleaved line  
by line).

RITOMP - Converts from a R/I file to a M/P file.

Usage: CALL MPTORI

CALL RITOMP

Common block COM1 parameters:

| Common name | Local name             |
|-------------|------------------------|
| IN (5)      | IN - Input file name   |
| ON          | IO - Output file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns |

Restrictions: The number of columns (NC) must be less than or equal to 1024.

Subroutines: DISKR, DISKW

System library routines: SIN, COS, SQRT, ATAN2

Storage: 10400<sub>(8)</sub> words

Required by: M/P TO R/I, R/I TO M/P

MULDSP (Multiple Display Format)

**Purpose:** MULDSP arranges multiple images into a single frame in display format. Options include the expansion of each image by an expansion factor, and labelling under each row of images.

**Usage:** CALL MULDSP (INTP, OUTP)

INTP - Input tape logical unit number  
OUTP - Output tape logical unit number

Common block COM1 parameters:

| Common name | Local name  |
|-------------|---|
| SN(5)       | SCRACH - Scratch disk file name   |
| L           | NL - Number of lines  |
| C           | NC - Number of columns  |
| K1          | NTP - Total number of images to be on output frame<br>NTP < 0 - no white spaces, butt images together<br>NTP ≥ 0 - 100 lines of blanks between rows |
| K2          | NPR - Number of images in one row across frame  |
| K3          | IOP - Option to invert gray scale values before output<br>IOP = 1 - invert<br>IOP = Other - No inversion  |
| K4          | IEXP - Expansion factor (a positive integer)  |
| K5          | NLAB - Total number of label cards following operator card  |

**Method:** The images are taken from successive files on the input tape, expanded if required, and formatted into a single  $n \times 1024$  frame in display format, where  $n$  is the variable length made up of  $NL * (\text{Number of rows} + 100)$ . Designated labels are written underneath the rows of images. A maximum of 102 characters may be written per line.

15 September 1973

74

System Development Corporation  
TM-(L)-HU-033/008/00

Restrictions: All input images must be the same size.

Subroutines: DISKR, DISKW, PL6, FRAMWR, DSPLN

System library routines: WRITER

Storage: 11161<sub>(8)</sub> words

Required by: MULTIPLE DISPLAY

NOSFUG (Non-Symmetrical Function Generator)

**Purpose:** NOSFUG produces a non-symmetrical two dimensional function for use as a point spread function in deconvolution or for general filter operations where non-symmetry is necessary.

**Usage:** CALL NOSFUG

Common block COM1 parameters:

| Common name | Local name              |
|-------------|-------------------------|
| ON          | JOUT - Output file name |
| L           | N - Number of lines     |
| C           | M - Number of columns   |

**Method:** The user specifies the non-symmetrical function base in polar coordinates and the function profile as a function of the distance along any radial through the Z-axis extending to the base boundary. The base boundary is defined in polar coordinates (R and Theta). The function profile is defined as a percentage of any radial.

**Subroutines:** DATAIN, DISKW

**System library routines:** SQRT, ATAN2, LAGRAN

**Storage:** 4360<sub>(8)</sub> words

**Required by:** NON-SYMMETRICAL PSF

NSA (Non-Symmetrical Alteration)

**Purpose:** NSA performs a non-symmetrical alteration of the gray scale of an image based on correction planes which are input on tape unit A5 (8).

**Usage:** CALL NSA

Common block COM1 parameters:

| Common name | Local name                                   |
|-------------|--|
| IN(5)       | IN - Input file name                         |
| ON          | ON - Output file name                        |
| L           | NL - Number of lines                         |
| C           | NC - Number of columns                       |
| KV          | NP - Number of planes ( $3 \leq NP \leq 8$ ) |

**Input Data:** The actual value of each correction plane is input following the operator card in table form as follows:

CURVE=5., 10., 20., 32., 45., 60.\$

**Tape Format:** The values in CURVE represent the actual value of the gray scale of each plane. For example, if six planes are to be used for correction, they may be generated by photographing 6 uniform fields of gray scales 5, 10, 20, ... as indicated above. The digitized images resulting from photographing the uniform fields are the planes used for correction and are formatted on tape as follows:

|              |                   |
|--------------|-------------------|
| RECORD 1     | Line 1, Plane 1   |
| RECORD 2     | Line 1, Plane 2   |
| .            | .                 |
| .            | .                 |
| .            | .                 |
| RECORD NP    | Line 1, Plane NP  |
| RECORD NP+1  | Line 2, Plane 1   |
| .            | .                 |
| .            | .                 |
| .            | .                 |
| RECORD NL*NP | Line NL, Plane NP |

The tape, with the correction planes properly formatted, must be generated prior to operation.

Method:

A three-point Lagrange interpolation is performed to find the corrected gray scale value for each pixel in the image. Assume the correction planes are designated  $X_1, X_2, \dots, X_n$ . The corrected value,  $G(X_{ij})$  is then computed by interpolation as follows: Given the dependent variable table,  $X_1(ij), X_2(ij), \dots, X_n(ij)$ , as a function of the independent variable table, CURVE, find a value,  $G(X_{ij})$ , as a function of given argument,  $X_{ij}$ .

Restrictions:

- 1)  $3 \leq NP \leq 8$
- 2) Number of columns in image must be no greater than 1024.
- 3) Number of columns in correction planes must be equal to number of columns in image.

Subroutines:

DISKR, DISKW, DATAIN

System library routines: LAGRAN, EXIT

Storage:

22406<sub>(8)</sub> words

15 September 1973

78

System Development Corporation  
TM-(L)-HU-033/008/00

Required by:

FIELD CORRECTION

ORAND (OR/AND)

**Purpose:** ORAND performs a logical AND or an inclusive OR on two input images.

**Usage:** CALL ORAND (KO)

KO - Option flag

KO = 1 - Inclusive OR

KO = 2 - AND

Common block COM1 parameters:

| Common name | Local name             |
|-------------|------------------------|
| IN(5)       | IN - Input file name   |
| ON          | ON - Output file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns |

**Subroutines:** DISKR, DISKW,

**Storage:** 10222 (8) words

**Required by:** AND, INCLUSIVE OR

PACK (Pack 6 Characters Per Word)

**Purpose:** PACK transfers N characters which are stored as the least significant 6 bits of N words into an array where the characters are packed 6 per word.

**Usage:** CALL PACK (A, N, B)  
A - Array of characters to be packed  
N - Number of characters  
B - Array into which characters will be packed

System library routines: FLD

Storage: 130<sub>(8)</sub> words

Required by: ALTER, DEPENDENT ALTER, PSF GENERATOR, FEATHER

PDGSA (Position Dependent Gray Scale Alteration)

**Purpose:** PDGSA performs a position dependent alteration of the gray scale value of each pixel of an input image.

**Usage:** CALL PDGSA

Modification tables A and B, of up to 100 values each, represent the coefficients of the linear expression  $A + BX$ .

Common block COM1 parameters:

| Common name | Local name                                |
|-------------|---|
| IN (5)      | I - Input file name                       |
| ON          | O - Output file name                      |
| L           | L - Number of lines                       |
| C           | C - Number of columns                     |
| KV          | N - Length of modification tables A and B |

**Method:** The radial distance of a pixel from the center of the image is computed and is used as the index (m) to tables A and B in calculating a modified value as

$$\text{Modified value} = A_m + B_m (\text{original value})$$

The modified value replaces the original value in the image.

**Restrictions:** All alterations are symmetrical about the center point of the image.

15 September 1973

82

System Development Corporation  
TM-(L)-HU-033/008/00

Subroutines:           DATAIN, DISKW, DISKR

System library routines: SQRT

Storage:               20342<sub>(8)</sub> words

Required by:           DEPENDENT ALTER

PRERUN (Pre-Run Diagnostics)

**Purpose:** PRERUN checks an input card image for the presence of an equal sign (to indicate a table) or a blank in card column 1 (to indicate a continuation card). This subroutine is only called in the case that sense switch 3 is down (signifying a pre-run diagnostic run) and the scan routine in entry RDC of subroutine INIT has not found a legal operator name in columns 1 - 3.

**Usage:** CALL PRERUN (CARD, I3C)  
CARD - Packed (6 ch/wd) card image  
I3C - Card columns 1 - 3

**Subroutines:** UNPACK

**Storage:** 101<sub>(8)</sub> words

PSFGEN (Symmetrical Point Spread Function Generator)

**Purpose:** PSFGEN accepts a curve which defines a point spread function half-profile and creates a three-dimensional PSF array in floating point format.

**Usage:** CALL PSFGEN

The desired set, of up to 50 floating point values representing the profile of the PSF to be generated, is input in tabular format.

Common block COM1 parameters:

| Common name | Local name                             |
|-------------|--|
| ON          | JOUT - Output file name                |
| KV          | KSIZ - Size of the PSF to be generated |

**Method:** The three-dimensional PSF is generated by rotating the input curve about its left-hand axis. The distance from the center is calculated for each point and a gray value is calculated by interpolating between the input curve points using library routine LAGRNG.

**Subroutines:** DATAIN, DISKW

System library routines: SQR, LAGRNG

**Storage:** 24166<sub>(8)</sub> words

**Required by:** PSF GENERATOR

RESTOR (Restore Disk File Directory)

**Purpose:** RESTOR restores the copy of the file directory which was saved on disk by operator DUMPFDD.

**Usage:** CALL RESTOR (LOF, NF, DISK)

LOF - Last opened output file

NF - Operator parameter table

DISK - Restore flag

DISK = .TRUE. - Restore from disk

DISK = .FALSE. - Restore from tape

Common block DISKFD parameters:

| Common name | Local name |                        |
|-------------|------------|------------------------|
| NLT (100)   | ITKO (464) | } Copy of Disk Track 0 |
| NCT (100)   |            |                        |
| KST (100)   |            |                        |
| NLPT (100)  |            |                        |
| KDT (100)   |            |                        |
| KP (100)    | ITK1 (464) | } Copy of Disk Track 1 |
| KUT (100)   |            |                        |
| IDT (100)   |            |                        |
| NAF         |            |                        |
| NOB         |            |                        |
| LTU (2)     |            |                        |
| NT (100)    |            |                        |
| ITKADR (7)  |            |                        |

System library routines: DTAKE, DCHECK, DPUT

**Storage:** 4211<sub>8</sub> words

**Required by:** RESTORE

REWND (Rewind)

Purpose: REWND rewinds and optionally unloads a tape drive. The tape logical unit number and load/unload flag are parameters read from the IDAPS operator card.

Usage: CALL REWND (CARD)  
CARD - Packed (6 ch/wd) representation of the operator card image

System library routines: CTOBIN, CLOSE

Storage: 66<sub>(8)</sub> words

Required by: REWIND

RSLGEN (Bar Chart Generator)

**Purpose:** RSLGEN produces a two-dimensional bar chart consisting of alternating groups of black and white bars with spatial frequencies which range from one cycle/line to one-half cycle per column.

**Usage:** CALL RSLGEN

Common block COM1 parameters:

| Common name | Local name               |
|-------------|--------------------------|
| ON          | JOUT - Output file name  |
| KV          | SIZE - Size of bar chart |

**Subroutines:** DISKW

**Storage:** 4353<sub>(8)</sub> words

**Required by:** BAR CHART

SCAL (Scale)

Purpose: SCAL scales an input image to values between 0. and 63.

Usage: CALL SCAL

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| IN (5)      | IN - Input file name   |
| SN (5)      | IS - Scratch file name   |
| ON          | IO - Output file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns   |
| K1          | KF - Curve flag<br><br>KF = 1 - Logarithmic (base e)<br>KF = 2 - Square root<br>KF = 3 - Cube root<br>KF = 4 - Square<br>KF = Other - Linear |
| K2          | LF - Clipping flag<br><br>LF = 1 - Clip outside of BC and TC<br>LF = Other - No clipping   |
| A1          | BC - Lower clipping level. All values below BC will be set to zero.  |
| A2          | TC - Upper clipping level. All values above TC will be set to 63.  |

15 September 1973

89

System Development Corporation  
TM-(L)-HU-033/008/00

Method:                   The minimum (MIN) and maximum (MAX) value is determined  
by comparing each pixel in the image.

The spread is computed as

$$\text{SPREAD} = \text{MAX} - \text{MIN}$$

and the step size, or increment, is

$$\text{STEP} = \text{SPREAD}/64.$$

The file is then rescanned to scale each pixel between 0.  
and 63. by the desired curve.

Subroutines:             DISKR, DISKW

System library routines: ALOG, SQRT

Storage:                4476<sub>(8)</sub> words

Required by:            SCALE

SCK (System Check)

Purpose: SCK is the main program in the diagnostic package. It contains individual checkout sections for each operator. General types of diagnostics made are:

- 1) Check for mandatory non-zero, non-blank parameters
- 2) Check size restrictions on disk files
- 3) Check unique restrictions of each operator

Usage: CALL SCK (ID, \*, \*,)

ID - Operator index  
\* - Statement numbers in subroutine INIT, preceded by \$, to which RETURN 1 and RETURN 2 refer in subroutine SCK.

Common block COM1 parameters: (See Appendix A)

Subroutines: MESS, SZCK

System library routines: SQRT, SSWTCH, NOW

Storage: 2334<sub>(8)</sub> words

SEGMNT (Segment Subsets)

Purpose: SEGMNT investigates an input image and identifies groups of pixels within a common subset. If a point within one subset is given, all other subsets are ignored.

Usage: CALL SEGMNT

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| IN(5)       | JIN - Input file names   |
| SN(5)       | JSKT, JKST - Scratch file names  |
| ON          | JOUT - Output file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns   |
| K1          | LL - Lower limit for thresholding. Pixels less than LL are excluded from subset calculation.                                   |
| K2          | LU - Upper limit for thresholding. Pixels greater than LU are excluded from subset calculation.                                |
| K3          | NULL - Size limit. Subsets smaller than NULL are not included in the subset ID.  |
| K4          | LN - Complement tag.<br>LN = 1 - Input file is complemented before processing.<br>LN = Other - Input file is not complemented. |
| K5          | IL - Line coordinate of a point within a subset.   |
| K6          | IC - Columns coordinate of a point within a subset.  |

Method: The SEGMNT subroutine consists of five major sections:

- Pixels which are members of a subset are determined by virtue of their being within the specified upper and lower

gray scale limits (LL and LU) and their having at least one neighbor within those limits.

- All members of a single subset are identified by assigning each pixel of the subset a unique identification number. If only one subset is desired (IL and IC > 0), all other subset members are set to zero and the desired subset members are set to 63.
- The number of subsets thus identified and the number of pixels which are members of each subset are counted and any subsets smaller than the specified size limit (NULL) are thrown out.
- Subset numbers are reassigned to reflect the relative size of all remaining subsets.
- An output array of subset ID numbers is prepared, and the operator statistics are printed.

The basic algorithm involved is that which identifies members of a single subset. Assuming that a pixel  $\gamma_0$  has been identified as being a member of a subset and it has neighbors in the following locations:

|            |            |            |
|------------|------------|------------|
| $\gamma_7$ | $\gamma_8$ | $\gamma_1$ |
| $\gamma_6$ | $\gamma_0$ | $\gamma_2$ |
| $\gamma_5$ | $\gamma_4$ | $\gamma_3$ |

which may or may not be within a subset, the following flow diagram describes the algorithm in general terms.

BEGIN

Set all pixels that are within the subset = 1. and all that are outside = 0. All border pixels set = 0.

Is  $\gamma_0 > 0$  ?

Is  $\gamma_6, \gamma_7, \gamma_8, \text{ or } \gamma_1 > 1$  ?

Set each of the four neighbors (above > 1) equal to the lowest greater-than-one member.

Set  $\gamma_0 = \text{KST}$  where KST is an integer tag > 1

Set  $\text{KST} = \text{KST} + 1$ .

For each neighbor ( $\gamma_6, \gamma_7, \gamma_8, \gamma_1$ ) modified above, enter the value that that the neighbor was changed to in a table (LSEG) in the location corresponding to the original value of the neighbor.

Is  $\gamma_0$  the next-to-last pixel in the next-to-last line?

Is  $\gamma_0$  the next-to-last pixel in line?

For each entry (n) in the LSEG Table, set --  
 $M = \text{LSEG}(n)$   
 $\text{LSEG}(n) = \text{LSEG}(m)$

Go to the next pixel in the line.

Write modified line onto scratch file then go to the 2nd pixel of next line

For each pixel stored in the scratch file (scratch(K)) create output file by setting --  
 $\text{output}(K) = \text{LSEG}(\text{scratch}(L))$

END

15 September 1973

94

System Development Corporation  
TM-(L)-HU-033/008/00

Subroutines: DISKR, DISKW

Storage: 22234<sub>(8)</sub> words

Required by: SUBSET

SHADE (Position Invariant Gray Scale Alteration)

**Purpose:** SHADE modifies the gray scale values of an image by using the original gray value as an index to a table of modified gray values and replacing the original value with the modified one.

**Usage:** CALL SHADE

The desired modification set of 64 values is input in tabular format.

Common block Com1 parameters:

| Common name | Local name              |
|-------------|-------------------------|
| IN(5)       | JIN - Input file name   |
| ON          | JOUT - Output file name |
| L           | L - Number of lines     |
| C           | C - Number of columns   |

**Subroutines:** DATAIN, DISKR, DISKW

**Storage:** 465 (8) words

**Required by:** ALTER

SHAVE (Frame Edge Rounding)

Purpose: SHAVE rounds off the edges of an input image.

Usage: CALL SHAVE

A table of up to 25 real values, representing percentages by which the elements on the edge of the frame will be modified, is entered in tabular format.

Common block COM1 parameters:

| Common name | Local name                                  |
|-------------|---|
| IN(5)       | IN - Input file name                        |
| ON          | IO - Output file name                       |
| L           | NL - Number of lines                        |
| C           | NC - Number of columns                      |
| KV          | NP - Number of points in modification table |

Subroutines: DATAIN, DISKR, DISKW

Storage: 7361 (8) words

Required by: FEATHER

SHRINK

Purpose: SHRINK reduces the size of an image by an integer factor.

Usage: CALL SHRINK

Common block COM1 parameters:

| Common name | Local name  |
|-------------|---|
| IN(5)       | IN - Input file names   |
| ON          | ON - Output file name   |
| L           | L - Number of lines in the picture  |
| C           | C - Number of columns in the picture  |
| K1          | IFACT - Shrinkage factor  |
| K2          | KOP - Shrinkage method<br>KOP = 1 - Every IFACT'th point<br>KOP = Other - Average of consecutive IFACT x IFACT arrays |

Method: An image that is  $(\frac{1}{\text{IFACT}})$  as large as the original image is formed by first calculating the average of consecutive IFACT x IFACT subarrays from the original image, and then inserting the average value into consecutive pixels in a smaller array.

Restriction: The maximum column size is 1024.

Subroutines: DISKR, DISKW

Storage: 10360<sub>(8)</sub> words

Required by: SHRINK

SLICE

Purpose: SLICE plots the gray level values found on a line between two points of an image.

Usage: CALL SLICE

Common block COM1 parameters:

| Common name | Local name                      |
|-------------|---------------------------------|
| IN(5)       | IN - Input file name            |
| L           | L - Number of lines             |
| C           | C - Number of columns           |
| K1          | IY1 - Line coordinate point 1   |
| K2          | IX1 - Column coordinate point 1 |
| K3          | IY2 - Line coordinate point 2   |
| K4          | IX2 - Column coordinate point 2 |

Method: The point slope form of the straight line equation is used to find the index of points found on a line between two points.

Subroutines: DISKR

Storage: 10624<sub>(8)</sub> words

Required by: SLICE

SPINT (Spatial Integration)

Purpose: SPINT performs a spatial integration of an image by averaging the eight immediate neighbors of each element and adjusting the element by some percentage.

Usage: CALL SPINT

Common block COM1 parameters:

| Common name | Local name  |
|-------------|---|
| IN(5)       | JIN - Input file name   |
| ON          | JOUT - Output file name   |
| L           | NL - Number of lines  |
| C           | NC - Number of columns  |
| K1          | LMT - Allowable variance within which an element will not be adjusted |
| K2          | KDT - Data type   |
| A1          | AJST - Correction factor (stated as a percentage)                     |

Method: Each element E of the frame of data within a 1 element border is surrounded by 8 immediate neighbors N; i.e.,

|   |   |   |
|---|---|---|
| N | N | N |
| N | E | N |
| N | N | N |

The mean ( $\bar{X}$ ) of the 8 neighboring elements is computed for each element within the frame. If the center element is outside a specified limit of the mean of its neighbors, it is replaced by adjusting it as follows:

$$\text{New E} = E - [E - \bar{X}] \cdot \text{AJST}$$

For example, if  $E = 20$ ,  $\bar{X} = 28$ ,  $\text{LMT} = 2$ , and  $\text{AJST} = .8$  (80%):

$$\begin{aligned} \text{New E} &= 20 - [(20 - 28) (.8)] \\ &= 20 - (-6.4) \\ &= 26.4 = 26 \end{aligned}$$

Restriction: The outer border of the image is unaltered.

Subroutines: DISKR, DISKW

Storage: 20526<sub>(8)</sub> words

Required by: SMOOTH

SPLT (Split Data)

Purpose:                   SPLT has two entries:

SPLT - Splits complex data which is interleaved line-by-line onto real (magnitude) components or imaginary (phase) components.

MERG - Merges the separated complex data line-by-line into one file.

Usage:                   CALL SPLT  
                          CALL MERG

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| IN(5)       | IN - Input file name   |
| ON          | IO - Output file name  |
| L           | NL - Number of lines   |
| C           | NC - Number of columns   |
| KV          | NP - Split flag<br>NP = 1 - Split imaginary (or phase)<br>NP = Other - Split real (or magnitude) |

Restriction:            The number of columns must be less than or equal to 1024.

Subroutines:            DISKR, DISKW

Storage:                2234<sub>(8)</sub> words

Required by:            SPLIT, MERGE

SPRED1

Purpose: SPRED1 performs the multiplication of two complex vectors.

Usage: CALL SPRED1 (N, IP, X, Y, GR, GI, MOL)

N - Size of the arrays

IP - Name of PSF file

X - Real part of iamge transform array

Y - Imaginary part of iamge transform array

GR - Real part of PSF transform array

GI - Imaginary part of PSF transform array

MOL - Line control

Method: Let  $I = \text{image} = X + j Y$

$P = \text{PSF} = GR + j GI$

$O_R =$  real part of resulting image

$O_I =$  imaginary part of resulting image.

then  $(I)(P) = (X + jY)(GR + j GI) =$

$(X)(GR) - (Y)(GI) + j[(Y)(GR) + (X)(GI)]$

$O_R = (X)(GR) - (Y)(GI)$

$O_I = (Y)(GR) + (X)(GI)$

Restriction: The arrays must be the same size and less than or equal to 1024.

Subroutines: DISKR

Storage: 142<sub>(8)</sub> words

Required by: CONVOLVE

SPRED2

**Purpose:** SPRED2 performs the division of two complex vectors while allowing the limitation of the amount of magnitude change or phase change of the numerator resulting from this operation.

**Usage:** CALL SPRED2 (NC, IN, IMR, IMI, PSFR, PSFI, NMLIM, NPLIM, MOL )

NC - Number of columns  
 IN - PSF file name  
 IMR - Real part of the numerator (image transform) array  
 IMI - Imaginary part of the numerator array  
 PSFR - Real part of the denominator (PSF transform) array  
 PSFI - Imaginary part of denominator array  
 NMLIM - Limit on magnitude amplification (integer)  
     NMLIM = Blank - No limit  
     NMLIM = 0 - Allow no magnitude change  
     NMLIM = Other - Limit all amplifications greater than NMLIM to NMLIM  
 NPLIM - Limit on phase amplification (integer)  
     NPLIM = Blank - No limit  
     NPLIM = 0 - Allow no phase change  
     NPLIM = Other - Limit all phase changes whose magnitude is greater than NPLIM to NPLIM.  
 MOL - Line control

**Method:** The real/imaginary components are converted into magnitude/phase components.  
 Let A = image magnitude vector  
 $\theta$  = image phase vector

B = PSF magnitude vector

$\emptyset$  = PSF phase vector

C = resulting magnitude vector

$\alpha$  = resulting magnitude vector

$$\frac{C/\alpha}{B/\emptyset} = \frac{A/\theta}{B} = \frac{A}{B} \frac{1}{\theta - \emptyset}$$

If NMLIM = Blank - operation above is carried out for all PSF points not equal to zero. If a PSF point is equal zero, the corresponding point of the image magnitude vector is unaltered.

If NMLIN = 0 - each point of B is set to one.

If NMLIN = Other - for those points such that  $1/B_i$  is greater than NMLIN, set  $1/B_i$  to NMLIN.

If NPLIM = Blank - operation above is carried out completely.

If NPLIM = 0 -  $\emptyset$  is set to zero.

If NPLIM = Other and  $|\emptyset|$  is greater than MPLIM,  $\emptyset$  is set to sign ( $\emptyset$ ) times NPLIM.

Restriction: The arrays must be the same size and less than or equal to 1024.

Subroutines: DISKR

System library routines: CTOBIN, SQRT, ATAN2, COS, SIN

Storage: 450<sub>(8)</sub> words

Required by: DECONVOLVE

SPRED3

**Purpose:** SPRED3 performs the multiplication of two complex vectors, and computes the complex conjugate of the result.

**Usage:** CALL SPRED3 (N,IP,X,Y,GR,GI,MOL)

N - Size of the arrays

IP - Name of PSF file

X - Real part of image transform array

Y - Imaginary part of image transform array

GR - Real part of PSF transform array

GI - Imaginary part of PSF transform array

MOL - Line control

**Method:** Let  $I = \text{image} = X + j Y$

$P = \text{PSF} = GR + j GI$

$O_R = \text{real part of resulting image}$

$O_I = \text{imaginary part of resulting image}$

then  $(I) (P) = (X + jY) (GR + j GI) =$

$(X) (GR) - (Y) (GI) + j [ (Y) (GR) + (X) (GI) ]$

$O_R = (X) (GR) - (Y) (GI)$

$O_I = -(Y) (GR) - (X) (GI)$

**Restrictions:** The arrays must be the same size and less than or equal to 1024.

**Subroutines:** DISKR

**Storage:** 142<sub>(8)</sub> words

**Required by:** CORRELATE

STAT (Statistics)

Purpose: STAT produces a histogram of the distribution of the gray scale values of an image and computes the following statistics:

Sum of gray scale values  
 Mean gray scale value  
 Average deviation  
 Variance  
 Standard deviation  
 Isolated maximum and minimum points

Usage: CALL STAT

Common Block COM1 Parameters:

| Common name | Local name           |
|-------------|----------------------|
| IN          | IN - Input file name |
| L           | NL - Number of lines |
| C           | NC - Number of colum |

Subroutines: DISKR

System library routines: SQRT

Storage: 5063<sub>(8)</sub> words

Required by: HISTOGRAM

STKPIC (Stack Frames of Data on Tape)

Purpose: STKPIC transfers an image from disk and stacks it on tape.

Usage: CALL STKPIC (IO)  
IO - Output tape logical unit number

Note: The output tape, IO, is not rewound, and no end of file is written after the last record.

Common block COM1 parameters:

| Common name | Local name             |
|-------------|------------------------|
| IN          | IN - Input file name   |
| L           | NL - Number of lines   |
| C           | NC - Number of columns |

Subroutines: DISKR

Storage: 1153<sub>(8)</sub> words

Required by: STACK

SZCK (Size Check)

**Purpose:** SZCK is used by the diagnostic package to provide size information about existing input files (found in the disk file directory).

**Usage:** CALL SZCK (IN, ISIZE)  
IN - Input file array  
ISIZE - Array containing the number of lines and number of columns of each file in IN.

**Storage:** 1716<sub>8</sub>) words

TFGEN (Transfer Function Generator)

**Purpose:** TFGEN generates a transfer function from user-provided independent variables.

**Usage:** CALL TFGEN (A,B,C,D)

A - Independent variable  
B - Independent variable  
C - Independent variable  
D - Independent variable

**Common block COM1 parameters:**

| Common name | Local name  |
|-------------|---|
| CARD(13)    | CARD - Input card image (packed)                    |
| ON          | ON - Output disk file name                          |
| L           | NL - Number of lines                                |
| C           | NC - Number of columns                              |
| K1          | K1 - Transfer function code                         |
|             | K1 = 1 - Polynomial                                 |
|             | K1 = 2 - Delta function                             |
|             | K1 = 3 - Gaussian                                   |
|             | K1 = 4 - MTF of defocussed perfect lens             |
|             | K1 = 5 - Quarter wave of zero order Bessel function |

**Method:** Based on the basic transfer function profile specified by the user and any required independent variables that must be provided for the selected profile, a one dimensional table is generated from the appropriate mathematical expression. The table length is selected equal to the number of picture elements from the center of the two dimensional output array to its outer edge.

The function profile curve is then "rotated" about the center point to generate a two dimensional output array having such rotational symmetry that any plane passing through the center point, normal to the base, will contain the original one dimensional profile curve and its mirror image. This is done by calculating the distance of every picture element in the output array from the center element and then finding the value for that element on the profile curve. A Lagrangian interpolation technique is used to find values for those point-to-center distances which are not an integer number of picture elements.

The two dimensional array of real data values is written as an output file and the values of the one dimensional profile from which the array was produced are printed on the on-line printer.

The transfer functions  $T(x)$  have the following mathematical formulae --

$$K = 1 \quad T(x) = A + Bx + Cx^2 + Dx^3$$

$$K = 2 \quad T(x) = A \quad \begin{array}{l} \text{where line} = B \\ \text{and column} = C \\ \\ = 0 \quad \text{where line} \neq B \\ \text{and column} \neq C \end{array}$$

(If no B and C are specified, the center point = A)

$$K = 3 \quad T(x) = Be^{-\frac{1}{2}\left(\frac{3x}{A}\right)^2}$$

$$K = 4 \quad T(x) = \frac{2}{\pi} \left[ \cos^{-1} \left( \frac{x}{A} \right) - \frac{x}{A} \sqrt{1 - \left( \frac{x}{A} \right)^2} \right] \left[ \frac{2J_1(K)}{K} \right]$$

$$0 \leq \frac{x}{A} \leq 1.$$

$$= 0 \quad \frac{x}{A} > 1.$$

$$\text{where } K = 8\pi B \left( \frac{x}{A} \right) \left( 1 - \left( \frac{x}{A} \right) \right)$$

$J_1(K)$  = First order Bessel function of K

$$K = 5 \quad T(x) = 2.405 J_0 \left( \frac{x}{A} \right) \quad \begin{array}{l} 0 \leq \frac{x}{A} \leq 1. \\ \\ = 0 \quad \frac{x}{A} > 1. \end{array}$$

15 September 1973

111

System Development Corporation  
TM-(L)-HU-033/008/00

Subroutines: DISKW, BESJ

System library routines: SQRT, ATAN2, LAGRAN

Storage: 11046<sub>(8)</sub> words

Required by: MTF

15 September 1973

112

System Development Corporation  
TM-(L)-HU-033/008/00

TIME

Purpose: TIME prints out the time required to execute the preceding operator, if sense switch 5 is not down.

Usage: CALL TIME (KSS)

KSS - Time at beginning of operator's execution.

System library routines: SSWICH, NOW

Storage: 72<sub>(8)</sub> words

TRANS (Transpose)

**Purpose:** TRANS performs a transposition, and associated operations, on a core buffer, and writes the final image out on the output file (A6).

**Usage:** CALL TRANS (OUTB, COREB, OUTF, NL, NUM, F, MAXC, IT, INDEX, REVOLV)

OUTB - Output buffer  
COREB - Core buffer  
OUTF - Output file on disk  
NL - Number of lines in the image  
NUM - Number of columns to transpose as rows  
F - Format  
F = 1 - Integer  
F = 0 - Floating Point  
MAXC - Maximum number of lines to write (actual number of unpacked characters in original line)  
IT - Current line on output disk file. If negative, then a horizontal rotation has been selected.  
INDEX - Variable dimension indicator for COREB  
REVOLV - Transposition option indicator  
REVOLV = 1 - Transpose only. (IT = 1).  
REVOLV = 2 - Transpose and vertical rotate. (IT = 1).  
REVOLV = 3 - Transpose and horizontal rotate.  
(IT = -NC).  
REVOLV = 4 - Transpose and both. (IT = -NC).

**Subroutines:** FLIP, DISKW

**System library routines:** FLD

**Storage:** 363<sub>(8)</sub> words

**Required by:** TRANSPOSE, +90 DEGREE ROTATE, -90 DEGREE ROTATE, MINOR  
TRANSPOSE, VERTICAL ROTATE, HORIZONTAL ROTATE, 180 DEGREE ROTATE

TRNPS (Transposition Control)

Purpose: TRNPS controls the rotations required by the following operators:

- 1) TRANSPOSE
- 2) +90 DEGREE ROTATE
- 3) -90 DEGREE ROTATE
- 4) MINOR TRANSPOSE
- 5) VERTICAL ROTATE
- 6) HORIZONTAL ROTATE
- 7) 180 DEGREE ROTATE

Usage: CALL TRNPS (COREFL, INDEX, REVOLV, NL)

COREFL - Buffer of core storage

INDEX - Variable which dynamically dimensions COREFL to maximum advantage

REVOLV - Rotation options indicator

REVOLV = 1 - Transpose (used for TRANSPOSE operator)

REVOLV = 2 - Transpose and Vertical Rotate (used for +90 DEGREE ROTATE operator)

REVOLV = 3 - Transpose and Horizontal Rotate (used for -90 DEGREE ROTATE operator)

REVOLV = 4 - Transpose and both Vertical and Horizontal Rotate (used for MINOR TRANSPOSE operator)

REVOLV = 5 - VERTICAL ROTATE

REVOLV = 6 - HORIZONTAL ROTATE

REVOLV = 7 - Both Vertical and Horizontal Rotate (used for 180 DEGREE ROTATE operator)

NL - Number of lines in input image

Common block COM1 parameters:

| Common name | Local name  |
|-------------|---|
| IN(5)       | INFILE - Input file name  |
| SN(5)       | SCRACH - Scratch file names   |
| ON          | OUTFIL - Output file names  |
| L           | NL - Number of lines  |
| C           | NC - Number of columns  |
| K1          | FMT - Format of the data<br>FMT = 1 - Integer<br>FMT = Other - Floating point |

**Method:** The input image is stored in COREFL and up to 5 scratch files. The core memory portion is transposed and written on the output file. Each scratch file is transferred to memory, one core buffer section at a time, and transposed and output in the same way. If possible, the scratch files are dimensioned the same size, or a multiple of the size, of the core buffer to minimize disk transfer time.

**Restrictions:** The maximum image size is 2048 x 2048.

**Subroutines:** DISKR, DISKW, FLIP, PL6, TRANS

**Storage:** 3145<sub>(8)</sub> words

**Required by:** TRANSPOSE, +90 DEGREE ROTATE, -90 DEGREE ROTATE, MINOR TRANSPOSE, VERTICAL ROTATE, HORIZONTAL ROTATE, 180 DEGREE ROTATE

TSFSZ (Transpose Scratch File Size)

**Purpose:** TSFSZ calculates the optimum sizes for the scratch files used by the TRANSPOSE and other related operators.

**Usage:** CALL TSFSZ (NL, NC, FMT, NO, KFN)

NL - Number of lines in original image

NC - Number of columns in original image

FMT - Format of original image

FMT = 1 - Integer

FMT = 0 - Floating point

NO - Dummy integer. Returns number of scratch files to be opened.

KFN - Dummy array. Returns size (number of columns) calculated for each scratch file.

**Method:** The optimum size of the scratch files to be opened is determined by the following considerations:

- 1) There may be a maximum of 5 scratch files opened.
- 2) The disk files should fit into the core buffer, with one core block per disk file, if possible. If not, the size of the disk file should be a multiple of the size of the core buffer (10240).
- 3) If unequal sizes are necessary between the five files, the last, or fifth, file should contain the overflow.

**Storage:** 272<sub>(8)</sub> words

**Required by:** TRANSPOSE, +90 DEGREE ROTATE, -90 DEGREE ROTATE, MINOR TRANSPOSE, VERTICAL ROTATE, HORIZONTAL ROTATE, 180 DEGREE ROTATE

UNPAC (Unpack 1 character from packed array)

Purpose: UNPAC is a function which extracts the  $j^{\text{th}}$  character from a packed array.

Usage: B = UNPAC(A,J)

A - Array of packed data

J - Index of character to be unpacked

B - Desired character right justified with leading zeros

System library routines: FLD

Storage: 103<sub>(8)</sub> words

UNPACK (Unpack 1 character from packed array)

Purpose: UNPACK is a function which extracts the  $j^{\text{th}}$  character from a packed array.

Usage: B = UNPACK(A,J)

A - Array of packed data

J - Index of character to be unpacked

B - Desired character, right justified with leading blanks

System library routines: FLD

Storage: 103<sub>(8)</sub> words

Required by: ALTER, DEPENDENT ALTER, PSF GENERATOR, FEATHER

UPL6 (Unpack Line - 6 Characters/Word)

Purpose: UPL6 contains two entry points:

- UPL6 - Unpacks 6 bit characters (6 char/word to 1 char/word).
- PL6 - Packs 6 bit characters (1 char/word to 6 char/word).

Usage: CALL UPL6 (LP, SC, NC, LU)  
CALL PL6 (LU, SC, NC, LP)

- LP - Packed line
- SC - Starting character to be packed/unpacked
- NC - Number of characters to be packed/unpacked
- LU - Unpacked line

System library routines: FLD, PCKINT, UNPINT

Storage: 543<sub>(8)</sub> words

XPAND (Frame Expansion)

**Purpose:** XPAND performs an expansion or an extraction of a selected portion of an image.

**Usage:** CALL XPAND (NP)  
NP - Expansion factor

Common block COM1 parameters:

| Common name | Local name   |
|-------------|--|
| IN (5)      | JIN - Input file name                                |
| ON          | JOUT - Output file name                              |
| L           | L - Number of lines in original frame                |
| C           | C - Number of columns in original frame              |
| K1          | LUL - Left upper line element number                 |
| K2          | LUC - Left upper column element number               |
| K3          | NE - Number of lines in extracted portion of image   |
| K4          | NC - Number of columns in extracted portion of image |

**Method:** Each line of the original image is read from the input tape and each element within the area to be expanded is duplicated NP times. Each line of data is then output NP times on the output tape. To extract rather than expand a selected portion of an image, NP should be set to 1.

**Restrictions:** The product of the size of the area to be expanded or extracted and NP may not exceed 2048.

**Subroutines:** DISKR, DISKW

15 September 1973

121

System Development Corporation  
TM-(L)-HU-033/008/00

System library routines: EXIT

Storage: 10302<sub>(8)</sub> words

Required by: EXPAND, EXTRACT

## APPENDIX A

## COMMON BLOCKS

COMMON/COM1/CARD,IN,SN,ON,PN,L,C,K1,K2,K3,K4,K5,K6,K7,K8,KV,A1,A2

Common block COM1 is used by the IDAPS system to store information used throughout the system. Although called by various local names, the information can be generally defined as follows:

- CARD(13)        - Input operator card image, packed six characters per word
- IN(5)           - Array of assigned input file names
- SN(5)           - Array of assigned scratch file names
- ON              - Output file name
- PN(5)           - Temporary file names used for format conversion of input files
- L               - Number of lines, as defined by the last processed SIZE operator
- C               - Number of columns, as defined by the last processed SIZE operator
- K1-K8          - Integer parameters from operator card
- KV              - Integer parameter from operator card
- A1,A2          - Real parameters from operator card

COMMON/DISKFD/NLT,NCT,KST,NLPT,KDT, KP,KUT, IDT,NAF,NOB,LTU,NT,ITKADR

Common block DISKFD is used by the IDAPS system as a disk file directory containing information about all disk files (up to 100) which are opened. The directory is divided into arrays which can be defined as follows:

- NLT(100)       - Number of lines in the image
- NCT(100)       - Number of columns in the image
- KST(100)       - Starting track number
- NLPT(100)      - Number of lines per track
- KDT(100)       - Data type
  - KDT = 0 - Floating point
  - KDT = 1 - 6-bit integer packed

KP(100) - Pointer to buffer  
           KP = 0 - Closed  
           KP = 1-7 - Buffer 1-7

KUT(100) - Unit specification (0 or 1)

IDT(100) - File identification

NAF - Number of assigned files

NOB - Number of open buffers

LTU(2) - Last track used

NT - Number of tracks

ITKADR(7) - Track address of data currently in buffer KP

COMMON/BREAK/KBREAK

Common block BREAK is used to relay the information that the break key (sense switch 2) has been pressed by the user.

COMMON/PRD/IERR

Common block PRD is used to relay the information that an error condition has been discovered by the diagnostic package.

COMMON/TIMS/KSS

Common block TIMS is used to store the starting time for an operator's execution, when the timing option (sense switch 5) is discovered.

COMMON/COM2/Y,GR,GI

Common block COM2 is used to store intermediate results and to share common buffers between the Fourier filtering subroutines (FRX2V, CONV, and FFIL). The names and dimensions defined for COM2 in the subroutines are as follows:

| <u>FRX2V</u> | <u>CONV</u> | <u>FFIL</u> |
|--------------|-------------|-------------|
| Y(512)       | DUM(1536)   | FIPIC(512)  |
| GR(512)      |             | EIPIC(512)  |
| GI(512)      |             | FILTER(512) |

APPENDIX B  
DIAGNOSTICS

IDAPS' diagnostics package produces error messages which are output on the on-line printer. One group of these messages, primarily concerned with syntax errors, originates from the pre-run diagnostics subroutine, SCK, and is stored in subroutine MESS. Following printout of one of this group of messages, two paths of operation may be followed. If the system is in the pre-run diagnostic mode (sense switch #3 down), the checkout of the remaining input cards is continued. During normal execution, however, the system pauses and awaits input of a corrected card before continuing execution. No execution of the incorrect operator is performed, any opened input disk files are closed, and output and scratch files for that operator are deleted. The following chart lists the non-fatal error messages in this group, the IDAPS operators to which they apply, and a brief explanation of each.

| DIAGNOSTICS                                   | OPERATORS  | EXPLANATION  |
|---|--|--|
| NON-BLANK PARAMETER REQUIRED                  |  | A blank occurs where a number must be.   |
| TAPE LOGICAL UNIT NUMBER MUST BE 8, 10, or 11 | LABEL<br>DUMP FILE<br>LOAD FILE  | Wrong number given for tape parameter.   |
| INPUT FILE PARAMETER INCORRECT                | AVERAGE<br>LOAD FILE   | Number of files parameter is either too many or missing.   |
| WRONG NO. INPUT FILE NAMES SPECIFIED          | AVERAGE<br>CONVOLVE<br>CORRELATE<br>MERGE<br>MATH<br>INCLUSIVE OR<br>AND<br>INSERT<br>MOMENT | Number of files listed not equal to number of files designated by a parameter, or by definition of the operator. |
| INPUT FILES MUST BE THE SAME SIZE             | AVERAGE<br>MOMENT<br>FOURIER FILTER  | Lines and columns of two or more input files are not equal.  |
| INPUT FILES MUST BE SQUARE                    | DECONVOLVE<br>CONVOLVE<br>CORRELATE<br>FOURIER FILTER  | Lines are not equal to columns in input file(s).   |
| DIMENSION MUST BE A POWER OF 2                | DECONVOLVE<br>BAR CHART<br>CONVOLVE<br>CORRELATE   | Dimension of input file is not a power of 2 between 32 and 512.  |
| DECONVOLVED IMAGE LESS THAN INPUT IMAGE       | DECONVOLVE   | Parameter designated size of square deconvolved image is less than the number of lines in the input file.        |
| PSF GREATER THAN INPUT IMAGE                  | DECONVOLVE   | Parameter designated the size of the square PSF is greater than the number of lines in the input file.           |

| DIAGNOSTICS  | OPERATORS                 | EXPLANATION   |
|--|---------------------------|---|
| LINES AND COL. OF 2ND FILE MUST = COL. OF 1ST                                      | FOURIER FILTER            | Number of lines of the square second input file does not equal the number of columns of the first input file.                       |
| N CANT BE LESS THAN HALF THE PICTURE DIAGONAL                                      | DEPENDENT ALTER           | Parameter N is less than $\sqrt{\left(\frac{NL}{2}\right)^2 + \left(\frac{NC}{2}\right)^2}$   |
| MOD CONTAINS MORE THAN 25 VALUES   | FEATHER                   | Parameter designated the number of values in table MOD is greater than 25.  |
| FILE MUST HAVE 1024 COLUMNS PER LINE   | LABEL                     | Input file has less than or more than 1024 columns.   |
| NON-ZERO PARAMETER REQUIRED  | SIZE                      | A zero number of lines or columns is illegal.   |
| OPERATION FLAG OUT OF BOUNDS   | MATH                      | Parameter is either less than 1 or greater than 8.  |
| FRAMES ON ROW GREATER THAN TOTAL NO. FRAMES  | MULTIPLE DISPLAY          | Parameter indicating the number of pictures/row must be less than the total number of pictures.                                     |
| PSF IS GREATER THAN 100  | PSF                       | Parameter indicating the size of the PSF is greater than 100 or less than zero.   |
| RESOLUTION CHART IS LARGER THAN 2048   | BAR CHART                 | Parameter indicating the size of the bar chart is greater than 2048.  |
| PARAMETER OUT OF BOUNDS  | FIELD CORRECTION<br>SCALE | Parameter less than 3 or greater than 8 for FIELD CORRECTION. Minimum clipping value greater than maximum clipping value for SCALE. |
| LARGER FILE MUST PRECEDE SMALLER ONE   | INSERT                    | Second input file is larger than first input file.  |
| SUBFRAME MUST FIT +2 WITHIN LARGER ARRAY<br>SUBFRAME OVERLAPS EDGES OF LARGE ARRAY | INSERT                    | Size of second input file too big to allow insertion into first input file.   |
| NO. LINES OF 1ST FILE MUST BE 2*NO. COLS.  | FOURIER FILTER            | Incorrect size for input file.  |
| NO. COLUMNS MUST BE NO GREATER THAN 1024   | FIELD CORRECTION          | Number of columns of input file greater than 1024.  |

Special disk file diagnostic messages handle errors which occur when the disk is being read or written on and during the file directory bookkeeping. These messages generally indicate irrecoverable error conditions and an exit from the IDAPS system occurs.

Other execution time diagnostics are not fatal. Either execution is continued following output of the error message, or instructions to the user are printed for re-entrance of the erroneous card.

A list of these diagnostics, whether they are fatal or not, the subroutine from which they originate, and a brief explanation of each is given in the following chart.

| DIAGNOSTICS   | FATAL | SUBROUTINE     | EXPLANATION  |
|---|-------|----------------|--|
| MULTIPLE INPUT FILE NAMES MUST BE SPECIFIED<br>FOR OPERATOR xxx               | x     | ASSIGN         | Operator xxx requires more than one input file and the names have not been provided in parentheses on the operator card. |
| BAD FILE NAME SPECIFICATION   | x     | ASSIGN         | No comma or close parentheses was found for multiple input file names specified on operator card.                        |
| IDENTICAL FILE NAMES NOT ALLOWED  | x     | ASSIGN         | A file name has occurred more than once in a multiple input file name specification.                                     |
| INCORRECT NUMBER OF FILE NAMES SPECIFIED                                      | x     | ASSIGN         | Either too many or too few names have been provided in the multiple input file name specification in parentheses.        |
| ERROR-IMAGE ARRAY TOO LARGE. MUST BE 2048<br>COLUMNS OR LESS FOR INTEGER FILE | x     | DISKO          | Maximum image size has been exceeded for an integer array.   |
| ERROR - TOO MANY OPEN FILES   | x     | DISKO          | An attempt has been made to open more than the maximum of 7 image files at one time.                                     |
| ERROR - FILE WITH ID OF xxxxxx NOT ASSIGNED                                   | x     | DISKD          | An attempt has been made to delete a non-existent file.  |
| ERROR - FILE WITH ID xxxxxx NOT OPEN  | x     | DISKC          | An attempt has been made to close a file that has been previously deleted or one that has not been opened.               |
|   | x     | DISKR<br>DISKW | An attempt has been made to read/write on a file which has not been opened (i.e., a buffer has not been assigned to it). |
| ERROR FILE DIRECTORY ID xxxxxx CANNOT BE FOUND                                | x     | DISKR<br>DISKW | An attempt has been made to read/write on a file which does not have an entry in the disk file directory.                |

| DIAGNOSTICS   | FATAL | SUBROUTINE       | EXPLANATION   |
|---|-------|------------------|---|
| ERROR - SUBSET NUMBER ASSIGNMENT HAS EXCEEDED AVAILABLE BUFFER SPACE                    |       | SEGMNT           | More than 2048 subsets have been found which fall within the specified thresholds.  |
| TOO MANY CHARACTERS IN DISPLAY LINE - EXCESSIVE CHARACTERS IGNORED                      |       | DSPLIN           | More than 102 characters provided for display.  |
| DATA CARD FORMAT ERROR - CLEAR CARD READER, CORRECT AND RELOAD CARDS, READY CARD READER |       | DATAIN           | No equals sign was found on first input data table card.  |
| INCORRECT NUMBER OF DATA VALUES PROVIDED - CLEAR CARD READER, CORRECT AND RELOAD DATA   |       | DATAIN           | The count of the input values in the tables is not equal to the number of values expected.  |
| ONE OR MORE PARAMETERS IN WRONG FORMAT  |       | INIT             | No decimal point found where real parameter required, or decimal point found where integer parameter required.  |
| xxx IS AN ILLEGAL OPERATOR  |       | PRERUN           | A card input during a pre-run diagnostic check is not part of a table or a continuation card and the first three characters are not a legal IDAPS operator. |
| xxx NOT LEGAL OPERATOR - CLEAR CARD READER, CORRECT AND RELOAD CARDS, READY CARD READER |       | RDC(INIT)        | First three characters of name on operator card (card columns 1-3) are not recognized as an IDAPS operator.   |
| ERROR IN INTERPOLATION ROUTINE  |       | FUNGN            | System error has occurred in system library routine LAGRNG.   |
| INTERPOLATION ERROR   |       | NSA              | System error occurred in system library interpolation routine LAGRAN.   |
| ERROR - xxxxxxxx DATA TRANSFER MAY BE INCOMPLETE OR WRONG                               |       | DISKW            | System error xxxxxxxx has occurred while attempting to write to disk (system library routine DCHEK).  |
| DATA TRANSFER ERROR - xxxxxxxx  |       | DUMPFD<br>RESTOR | System error xxxxxxxx has occurred while attempting to dump a disk file or write on disk (system library routine DCHEK).                                    |

## APPENDIX C

## PROCESS FOR ADDING AN OPERATOR TO THE IDAPS SYSTEM

## A. Subroutine IDAPS2

The array NF must be altered to include information about the new operator.  
Add

```
      ,6HXXX  ,A,B,C/
```

to the DATA declaration for array NF, where XXX are the first three characters in the operator's name, and A,B, and C are defined as follows:

- A - Number of input disk files to be assigned
- B - Number of scratch disk files to be assigned
- C - Number of output disk files to be assigned

The DATA declaration for integer NO and the implied DO delimiter in the DATA statement for array NF, must be incremented to reflect the additional operator.

The section of subroutine IDAPS2 pertaining to the new operator should be in the following format:

```
      nn      CALL OPnn
              CALL SCK (I,$nn,00,$6 00)
nn00      CONTINUE
              CALL NEWOP
              GO TO 600
```

where nn by convention indicates the new operator's index into array NF (also contained in integer I). OPnn is the entry for the operator in subroutine INIT, SCK is the driver for the diagnostic package, and NEWOP is the new execution subroutine or subroutine package.

At statement number 200, the GO TO switch must contain the statement number nn.

## B. Subroutine INIT

The new operator, if it requires the opening of any disk files or the input of any parameters, must have an entry in subroutine INIT. Here, the parameters are read in by calls to IPARAM, format conversion is performed by a call on CKFMT, and any needed disk files are opened by calls to DISKO.

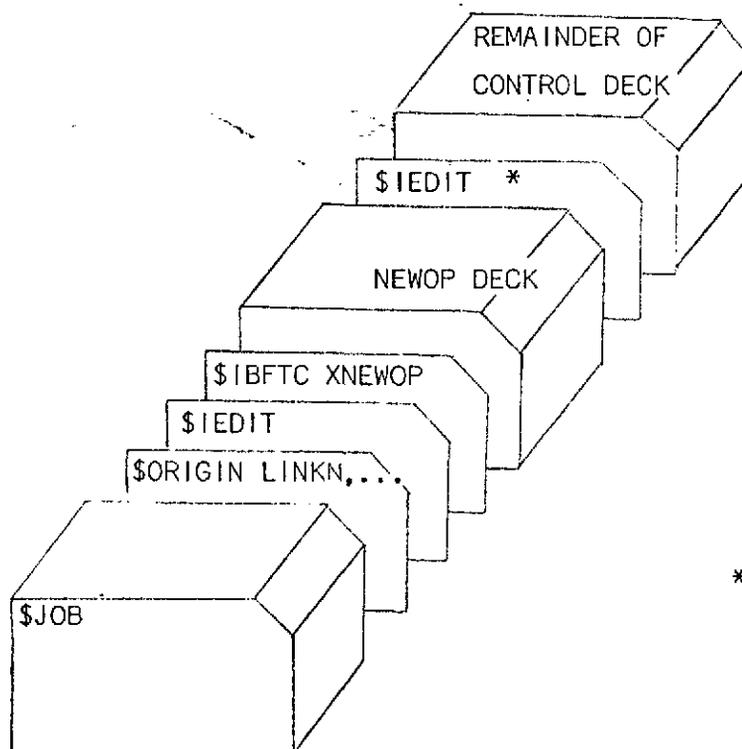
## C. Subroutine SCK

Any diagnostics to be made on the new operator should be checked in subroutine SCK. As in IDAPS2, the operator index is used as a statement number and must be included in the initial GO TO switch.

Checks on dimensions of disk files may be made by calls on SZCK. The diagnostic messages are stored in array B in subroutine MESS. Any additions made to this array must be reflected in the dimensions of the array. The maximum number of characters in a diagnostic message is 46.

## D. Subroutine "NEWOP"

After its proper linkage, N, is determined, the subroutine (or subroutine package) NEWOP is inserted in the run deck in the following manner:



\*SYSCK2, SCHF01