UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

REMAPP - PC

Remote Sensing Image Processing Software for MS - DOS Personal Computers

VERSION 2.00

by

K. Eric Livo and Andrea J. Gallagher

Open-File Report 91-449 A-G

1991

91 - 449 A	Documentation (Paper Copy)
91 - 449 B	Installation and Executable Programs
91 - 449 C	Executable Programs and Source Code
91 - 449 D	Documentation and Landsat MSS Test Images (Canon City, Colorado)
91 - 449 E	Landsat 4 - TM Test Images (TM Disk 1 of 2) (Canon City and Central City, Colorado)
91 - 449 F	Landsat 4 - TM Test Images (TM Disk 2 of 2) (Canon City and Central City, Colorado)
91 - 449 G	AVIRIS (NASA) Imaging Spectrometry Test Data

This report is preliminary and has not been reviewed for conformity with the U.S. Geological Survey editorial standards. Use of trade names in this report is for descriptive purposes only and does not imply endorsement by the USGS. Although these programs have been extensively tested, the USGS makes no guarantee of correct results.

U.S. Geological Survey, Denver, Colorado 80225

Table of Contents

List of Figures

Introduction

The REMAPP - PC Remote Sensing Image Processing Software is a series of programs for processing satellite and aircraft digital image data, such as Landsat Multispectral Scanner (MSS) and Thematic Mapper (TM), Systeme Probatoire d'Observation de la Terre (SPOT), Airborne Visible and Infrared Imaging Spectrometry (AVIRIS), or other digital data. To use the REMAPP system an IBM-PC compatible computer with Video Graphics Array (VGA) color display and a hard disk are required. No additional software packages are needed to support REMAPP.

The REMote sensing Array Processing Procedures (REMAPP) package is useful for image display and enhancement, and other more specific uses, such as mineral alteration studies and basic spectral work. Image processing routines include mathematical operations, image statistics, contrast enhancement, spatial and color coordinate transformations, edge enhancement, and masking. Image files are supported with file management, file utility, and file import/export routines. All routines are stand-alone programs. New image processing algorithms may be incorporated readily with minor FORTRAN, C, or BASIC programming.

Image files may contain a maximum of 8192 pixels (picture elements) per scanline as 8 or 16-bits/pixel data, with the number of scanlines restricted only by the amount of hard disk storage capacity. Image file format consists of an **image subdirectory** containing a REMAPP ASCII header file (named HEADER) and a single band or band interleaved by line (BIL) binary image data file (named DATA). The image subdirectory name is the name passed to REMAPP as **the image filename.**

Images are displayed using up to 256 colors or 64 shades of gray, with 8 graphic plane colors available in either mode. Maximum screen resolution is 200 scanlines by 320 pixels; however, larger files may be displayed by windowing or subsetting the data within the display program, yielding larger spatial coverage with degraded resolution. All image processing routines however, retain full eight or 16-bits per pixel DN precision.

Hardware Requirements

This image processing system requires an MS-DOS IBM compatible PC computer with 640 kilobytes of system memory and a VGA graphics display. A VGA color graphics monitor is required for full use of the display options, though a VGA monochrome graphics monitor may be substituted (but not advised). Likewise, though the system may be run from floppy drives only, to gain usefulness, a hard disk is necessary. A 20 MB hard drive would be the minimum capacity needed to work with the test images provided and other small data sets. (Assuming the user keeps other software packages on the same hard drive.) For working with large data sets, larger hard drives (80 - 100 MB) would facilitate the processing.

Optional equipment are a math coprocessor, a Microsoft mouse, and the Microsoft program WINDOWS 3.0. All programs have been compiled to run on the entire family of Intel PC CPU chips, with or without math coprocessor chips. A math coprocessor will automatically be used if present. This series of programs are stored on 1.2 megabyte floppy disks; however, the programs may be transferred to any other size floppy for use.

The program DISPLAY is the only program requiring specific equipment. All the other programs are non-graphic and will run on PC's with any video display. DISPLAY requires VGA graphics and 640 kilobytes of system memory, of which something over 530K needs to be free (use MEM or CHKDSK to check). Multiprocessing with these DOS programs is available with 386 class PC's using Microsoft Windows 3.0.

Figure 1. Screen photograph of a TM band 3 image of Canon City, Colorado

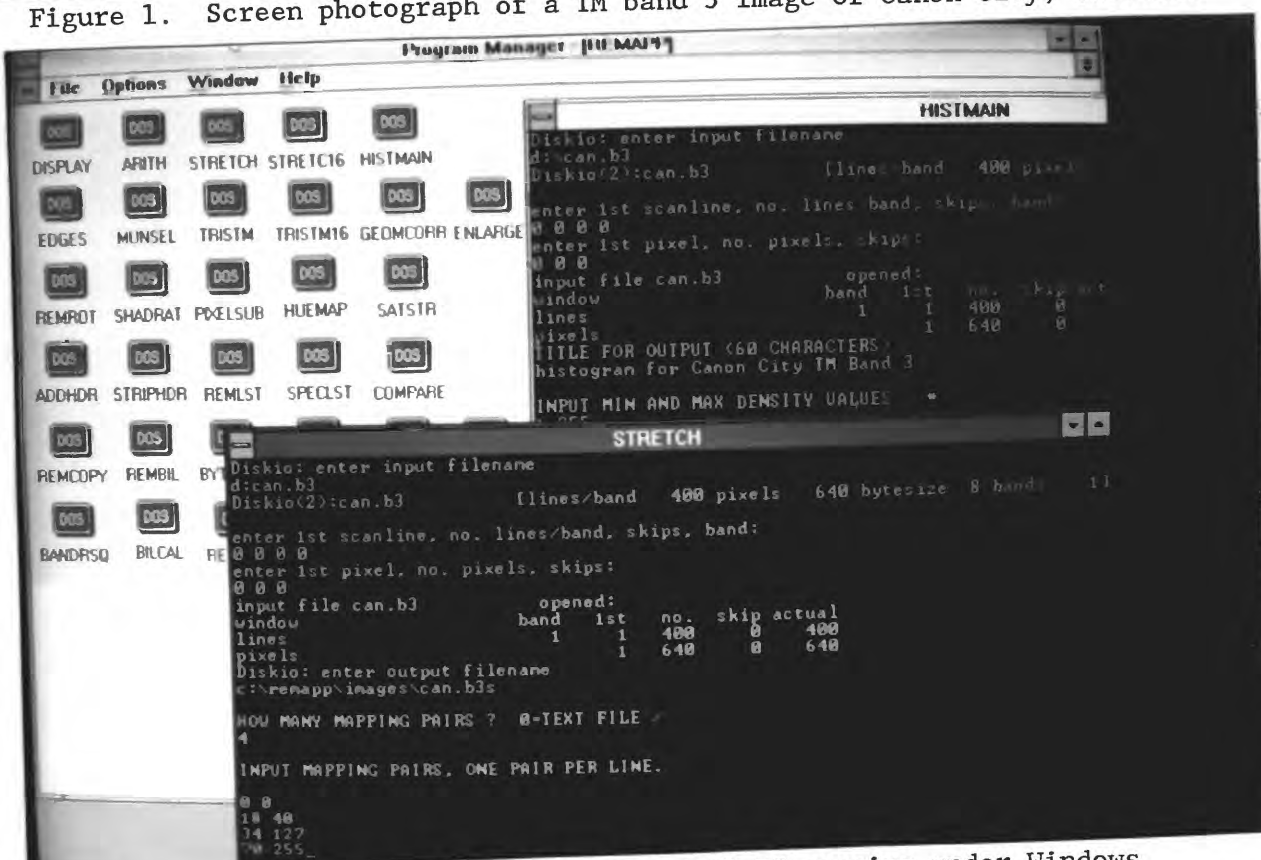


Figure 2. Screen photograph of REMAPP-PC running under Windows

Data Import/Export

Digital data may be imported and exported several ways. Tape drives are available for PC's from $3,000 - $10,000, or nine-track tape data may be transferred to floppy disk by media conversion service companies. A PC may be connected to a minicomputer and digital data transfered from the mini to the PC serially through a hard-wire using communication software on each computer with a binary dump. Ether net file transfer is another data transfer method. Landsat TM data are available on MS-DOS floppy diskettes from EOSAT. Data may be exported the same way it was imported. Files residing on hard disk may be transported or archived using the MS-DOS programs BACKUP and RESTORE. These two DOS programs allow a digital file to span several diskettes.

Processed data may be output to film, paper, or magnetic media. Film products can be generated through photographing the display, transferring the image to a professional film-writer through direct hard-wire or nine-track tape, or a slide may be produced from floppy disk data stored in a custom format by commercial slide imaging firms. All methods above would need to be developed by the individual user.

Images may easily be transferred to paper copy, though with reduced clarity. Commercial paint programs, such as Deluxe Paint II (Silva, 1988), can 'grab' a screen image, which then may be printed by the paint program through its proprietary printer routines. Deluxe Paint II, for example, outputs images on laser printers, and lower cost color and black-and-white ink-jet printers. Images have been displayed on the HP-PaintJet color ink-jet printer as well as the HP LaserJet series II with moderately good results. Dot matrix printers also produce modest quality prints. An Epson-MX compatible dot matrix print routine is included in this processing system. Larger images may be generated with finer detail making a mosaic of several paper print-outs. A side benefit of paint programs are their support of annotation on the images.

The most useful way of making detailed hardcopy with a minimal amount of equipment is to photograph the screen. Prints may be made from either slide film or negatives, depending on the local film developer's capabilities.

Software Development

The 'REMAPP' system was developed over a period of years by the U. S. Geological Survey, Branch of Geophysics, remote sensing group on several mini-computers. Some of the most useful stand-alone subroutines have been converted here to run in the MS-DOS environment. Tying the image processing system together is the display driver program DISPLAY, that enables digital image data to be displayed and analyzed. The driver package includes image buffers, graphics overlays, cursor, histogram stretch and windowing capabilities.

Programming languages for this system are principally Microsoft FORTRAN version 5.0 for the non-display processing routines, and Microsoft QuickC version 2.0 for the display driver. Microsoft QuickBasic version 4.0 is used in a couple of the routines.

Software Usage

The terms "scanline, pixel, DN, and image origin" will be briefly defined and terminology will be clarified. "Scanline" refers to the raster scan of a line of data, which in this system is one data record in an image file. This horizontal line of data is composed of data points referred to as "pixels". "DN" stands for digital number. Every pixel in a scanline has an associated digital number. Eight bits per pixel DNs range from 0 to 255 (2 to the eighth power), while 16-bits per pixel DNs range from 0 to 32,767 (2 to the sixteenth power, excluding negative values 0 to -32,768). The image origin refers to the upper left corner of an image, with increasing scanlines downward, and increasing pixels to the right. In this system, the image origin always starts at scanline 1, pixel 1 (1,1). All program names appear in capital letters within this report (i.e.: DISPLAY).

Image system usage relies on the 'REMAPP' file format (see Appendix C). This image file format is defined as a subdirectory, whose name becomes the image name, containing an ASCII header file and a binary image data file. All processing programs read this ASCII header file to learn the specifics about the image. This file format is created by the subprogram DISKIO, which is also the user interface for most of the programs. The various programs call DISKIO through function call statements. Open, read, write, scanline positioning, image windowing and subsetting, and close file functions are performed transparently to the user by DISKIO for a consistent user interface and compatible data structure. Programming with DISKIO is detailed in the USGS Open-File Report 85-231 (Sawatzky, 1985).

Appendix B describes the user interface operation of DISKIO for entering filename, and scanline and pixel subsets. It is very important to understand this operation for every image processing program uses this interface.

Installation

All files on the source floppy disks should be copied into subdirectories created on the hard disk. Subdirectories and the DOS file system tree structure should be organized by function. Though a single large subdirectory may be used to hold all files, it is less efficient.

An example structure would be:

subdirectory function	subdirectory
root (contains all the other subdirectories):	\remapp
executable (binary) programs:	\remapp\bin
source programs:	\remapp\source
M.S. Windows 3.0 files:	\remapp\pif
object code:	\remapp\link
images:	\remapp\images
additional images:	\remapp\images2

Standard Installation

Use the batch file INSTALL which resides on disk 1 for a standard installation. To install REMAPP-PC, invoke the batch file by typing **A:\INSTALL A: C:** and follow the instructions. Use the proper drive letters for your floppy and hard drives. Finally, modify your DOS path and add REMAPP.GRP to the MS-Windows PROGMAN.INI file for optional Windows usage as described in the Finish Installation section below.

Manual Installation

Create the subdirectories on the hard disk with the DOS MD (or MKDIR) command.

```
i.e.: md c:\remapp
      md c:\remapp\bin
etc. (as subdirectories above)
```

Once all the subdirectories are created, copy all the image processing files to their respective subdirectories using the DOS COPY or XCOPY command.

1) All executable program files (files with an .EXE extension) and batch files (.BAT) should be copied into the bin subdirectory. These files will be found in the \bin subdirectory of the distribution diskette.

```
i.e.: xcopy a:\bin\*.*  c:\remapp\bin\*.* /S /E /V
```

2) All source files (those with FORTRAN (.FOR), C (.C), and BASIC (.BAS)) should be copied into the source subdirectory. These source files are not necessary for program execution, but are useful for program modification or study. See Appendix D for program compiling and linking instructions. These files will be found in the \source subdirectory of the distribution diskette.

i.e.: xcopy a:\source*.* c:\remapp\source*.* /S /E /V

3) All images should be copied into image subdirectories. Several image subdirectories are desirable, though one large subdirectory may be used. Subdirectories for individual study areas may contain their corresponding images for better organization. Test image files will be found in the \images subdirectory of the distribution diskette.

i.e.: xcopy a:\images*.* c:\remapp\images*.* /S /E /V

4) All object code should be copied into the link subdirectory, or what ever subdirectory the user wants to compile programs in. These files will be found in the \link subdirectory of the distribution diskette.

i.e.: xcopy a:\link*.* c:\remapp\link*.* /S /E /V

5) If Windows 3.0 capability is desired with these MS-DOS programs, all PIF files and REMAPP.GRP should be copied either into the PIF subdirectory and/or into the WINDOWS 3.0 subdirectory. These files will be found in the \pif subdirectory of the distribution diskette.

i.e.: xcopy a:\pif*.* c:\remapp\pif*.* /S /E /V

Finish Installation

The installation is completed by adding the REMAPP program subdirectories to the MS-DOS PATH system variable (usually found in AUTOEXEC.BAT) and installing a group into the MS-Windows PROGMAN.INI file.

1) Set the DOS path to include the c:\remapp\bin and \remapp\pif subdirectories, either in the AUTOEXEC.BAT file, in some other startup batch file, or on the command line.

example: **path=c:\;c:\dos;c:\remapp\bin;c:\remapp\pif;c:\windows**

2) Optional, for windows compatibility, modify the PROGMAN.INI file within the \WINDOWS subdirectory. Using a word processor (such as EDIT), add the line:

Group#=C:\REMAPP\PIF\REMAPP.GRP

to the [Groups] section of the PROGMAN.INI file.

where the number # = the last group number + 1
example: Group8=c:\REMAPP\PIF\REMAPP.GRP

Insert this line just after the last Group definition line

NOTE: Please use the next available unused Group number in your PROGMAN.INI file; i.e.. if the last current group number used is 7, then use number 8.

Program Operation

To run any of the Image processing programs, make sure the path is set to the executable program subdirectory (see Installation above). Change directory (cd) to the image subdirectory containing the images to be processed (i.e.: cd c:\remapp\images1). Image subdirectories may be on different drives (even floppies). The image processing programs may then be run from the DOS prompt by typing their names (i.e.: type STRETCH to run the STRETCH program). Full path names must be used when inputting filenames of images not in the current subdirectory.

Successful image processing requires planning to achieve specific objectives. Objectives may be as simple as an image stretched for display, or a complex alteration theme map generated with single band pixel substitution (masking) for shadow, water, snow, and vegetation on a color ratio composite base, high-lighting iron oxides, clays, and carbonates. Examples of some image processing flow charts are presented in Appendix A.

Several important notes on operating the software are mentioned here. On running most REMAPP-PC programs, a program processing log is appended to the ASCII text-file named REMAPP.LOG. When REMAPP.LOG is not present in the current subdirectory, it is created. This text-file may be viewed using the DOS TYPE command and deleted at any time if desired. Upon running any of the processing programs, existing files will not be overwritten by new output files. The program will stop and a run time error will occur if the chosen output filename exits. File attributes of generated output files are inherited from the input files used. Unless specifically mentioned in the program description or changed by DISKIO subsetting, the number of scanlines and pixels, DN bits/pixel, and image format (BSQ, BIL, etc) default to the values of the first input file used.

All of the programs are individually described in the next section. Explore the programs and data output. The original input data will not be changed, so a program may be run several times to generate optimum results. The output image may be immediately displayed or examined with the file management programs. ASCII data may be examined on the screen by the DOS 'TYPE' command and/or sent to a printer.

A good beginning to learning this software processing package would be to try the DISPLAY program first (display an image). The various functions within DISPLAY will stimulate individual topics for further study by the user. Also look at the flow charts in Appendix A.

Public Distribution

This software package is offered into the public domain as research software. It is not a commercially supported software package and is made available for usage and modification, if need be, by the user. The REMAPP - PC Remote Sensing Image Processing Software may be copied and distributed freely.

Operating Notes

1.) REMAPP image format stores images in a subdirectory structure. Every image is stored in an individual subdirectory containing two files, named 'HEADER' and 'DATA'. **The subdirectory name will be used as the image name.** The HEADER file is an ASCII text file describing the image data while the DATA file is the actual binary image. See appendix C for details.

2.) To use the programs, change to the directory (cd or chdir) in which the images are stored (the subdirectory above the image name). If working with an image not in the current directory, give the full path.

3.) Once an image is open (or being accessed by a program) it cannot be accessed concurrently by the same program or another program.

4.) To start any of the programs below, type the name of the program at the DOS prompt.

5.) To start any batch files (scripts), type the name and any arguments on the same line at the DOS prompt.

6.) See Appendix B for information about Diskio prompts.

Note: DOS is not case sensitive

LIST OF PROGRAMS

Display

Video

DISPLAY	displays images on a VGA monitor

Printer

PRIMG6	prints coarse image on Epson dot matrix printer

Data Import/Export

ADDHDR	creates image subdirectory and REMAPP header
STRIPHDR	strips non REMAPP header and creates REMAPP image

File Management

REMHDR	displays REMAPP header information
REMLST	converts REMAPP image files from binary to ASCII
COMPARE	compares REMAPP images to check for identical data
REMFILE	creates test image file

File Utilities

RDEL	deletes image files
REMCOPY	copies REMAPP images or image subsets
REMBIL	assembles BIL files from single band files
SUBBAND	subtracts individual bands from a BIL file
BYTEFLIP	transposes byte pairs
APPENDX	appends image files top to bottom
CONCAT	concatenates image files left to right
BADLINE	replaces bad scanlines
GRDREM	converts USGS grid files to REMAPP images

Image Processing

Math

ARITH	applies math functions to image files
STRETCH	applies linear contrast stretch(es) to image files
STRETC16	same as above with 16-bit output

Statistics

HIST	calculates statistics and histogram for image files
TMCAL	generates Landsat TM calibration constants

Filters

EDGES	generates edge enhanced images
BOX	applies box smoothing function to image files

Color Coordinate System

MUNSEL	converts RGB images to hue, saturation, and value
TRISTM	converts Munsel HSV images to red, green and blue
TRISTM16	same as above with 16-bit output

Spatial Coordinate System

GEOMCORR	performs geometric correction of Landsat MSS data
ENLARGE	enlarges image files through pixel replication
REMROT	rotates image files 90 degrees clockwise
REMFLIP	flips an image top to bottom with no rotation

Masking

MASK	general purpose masking routine (masked images)
SHADRAT	masked image from background, overlay/mask images; +DN interval
PIXELSUB	masked image from background, overlay, mask images; +DN interval
HUEMAP	masked image from background, mask images; +defined DN,DN interval
SATSTR	masked image from overlay, mask images; +DN=0, DN intervals

Spectral

BANDRSQ	creates spectrally classified mineral theme images
CALFILE	generates calibration band multiplier ASCII file
SPECAVG	averages numerous CALFILE ASCII files into output
BILCAL	calibrates BIL image files using CALFILE files
ADDWAVES	merges 1st columns of 2 files into 2 columns out
SPECLST	extracts spectra from BIL image cube, ASCII output

Program Descriptions

ADDHDR

ADDHDR imports non-REMAPP raster image data and converts it into REMAPP style image data. The imported data must be binary integer raster data of 8 or 16 bits/pixel, as normally found in most remote sensing systems. ADDHDR creates an image subdirectory (image filename). In this subdirectory, a 'REMAPP' header file is generated from user supplied information and the original image file is copied and named 'data'. The data file can be either a single band image or a BIL binary array image. The header file provides descriptive information about the image to the 'REMAPP' programs.

To run, start program:

1) enter input non-'REMAPP' image filename

2) enter number of pixels per scanline for the input file

3) enter number of bits per pixel for the input file

4) enter number of bands in the scene
   (1 for a single band image or the number of bands [channels] in a BIL scene)

5) enter the output 'REMAPP' filename

NOTE: if answers are unknown, enter a guess for pixels per scanline (say 512, 1024, or 2048) and try 8-bits per pixel. Only a few types of images are 16-bits/pixel. Display image with DISPLAY using all 0's for scanline and pixel subsets, and look for a coherent pattern of scanline shifts. Try to determine a pattern, then run ADDHDR again using this new number of pixels with the original input file. For larger images, a larger pixels per scanline might be needed, with skips being used in DISPLAY, and number of pixel skips taken into account during the pixel count.

ADDWAVES

ADDWAVES merges ASCII data, such as wavelength files, from two input files together. ASCII data columns in each of the input data files is used to generate a two data column ASCII output file. The first column of the output file is copied from the first numeric column of the first input file, while the second column of the output file is copied from the first numeric column of the second input file. Any extra numeric columns in either input file is ignored by the program.

To run, start program:

1) enter the first ASCII input filename
   (for the first column of the output file)

2) enter the second ASCII input filename
   (for the second column of the output file)

3) enter the ASCII output filename

An example would be an ASCII wavelength file and an ASCII reflectance file, merged (concatenated) to form a wavelength - reflectance pair output file.

APPENDX

This program appends 'REMAPP' image files to each other, top to bottom, to form one image file. The first input file forms the top, with subsequent files being added to the bottom of the preceding files.

The scanlines in the appended file will all be the same length (pixels/scanline). To append files with scanlines of unequal length, fill pixels may be added to the beginning or the end of the scanline for any of the input files. Pixels per scanline = input file offset + DISKIO pixel subset + extra number of pixels. If the first image has longer scanlines than the subsequent images, the default will be for the right side of the subsequent the images to be padded with fill DN values to make up the difference. If the first image has shorter scanlines than the subsequent images, the default will be for the subsequent images to be truncated to the same scanline length as the first image. The first image also sets the bits/pixel of the output file to be equal to itself.

To duplicate an image file, but with more pixels per scanline, input 0 for the number of files to append and use extra offset and/or extra number of pixels.

To run, start program:

1) enter the number of files to append to the first file
   (i.e.: if there are four input files total, enter 3 - the
   first file is copied and three files are appended to it)

2) answer DISKIO for the first input image filename, and scanline and pixel subsets.

3) enter the input file offset, extra number of pixels, and fill DN of the
   offset and extra pixels
   (default set is 0 0 0).
   The input file offset is the number of pixels from the left image edge of the output file, from which the input image will start to be written.
   The extra number of pixels is the number of pixels from the right edge of the input image to the right edge of the output image file.
   The fill DN is the pixel DN (assigned by the user) for the offset and extra pixels, with 0 being black, to 255 being white.

If all appended input files have the same number of pixels per scanline and no fill is desired, use the default set of : 0 0 0.

4) enter the output image filename
5) answer DISKIO for the next input image filename, and scanline and pixel subsets
6) enter the input file offset, extra number of pixels, and fill DN.
7) repeat steps 5 and 6, until all of the input files are used.
8) check size of closed output file to be sure that input files were entered correctly.

EXAMPLE:

```
  1)     1
  2)     ANYWHERE.TOP
         0    0    0    0
         1  320    0
  3)     0   20    0
  4)     ANYWHERE.IMG
  5)     ANYWHERE.BOT
         0    0    0    0
         0    0    0
  6)     0    0    0
```

yielding an output file with 340 pixels/scanline.

ARITH

ARITH performs mathematical operations on image data. The operations available for one or two images and constants include addition, subtraction, multiplication, and division. Logarithm and exponential functions are also available.

This program is very useful for creating band-ratio images and calibrated data. Create band-ratio images using algorithm 3 below, or calibrate Landsat TM images using algorithm 4 with the ALPHA and BETA constants derived in TMCAL. ARITH takes 8 or 16-bit data input and outputs 16-bit data.

Six different equations are available within ARITH. The algorithms are:

1) (ALPHAA * FILEA) + (ALPHAB * FILEB) + BETA
2) (ALPHA * FILEA * FILEB) + BETA
3) (ALPHA * (FILEA / FILEB)) + BETA
4) (ALPHAA * FILEA) + BETA
5) (ALPHA * LOG(GAMMA * FILEA)) + BETA
6) (ALPHA * EXPONENTIAL(GAMMA * FILEA)) + BETA

where:

FILEA is the first image input file. FILEB is the second image input file (when required). ALPHA and GAMMA are user supplied real multiplication constants. BETA is a user supplied real additive constant.

To run, start program:

1) choose algorithm (1, 2, 3, 4, 5, or 6)

2) answer DISKIO for input image (FILEA) filename, and scanline and pixel subsets.

3) Answer DISKIO for input image (FILEB) filename, and scanline and pixel subsets (where applicable).

4) enter output image filename

5) enter ALPHAA (ALPHA), ALPHAB (if used), GAMMA (if used), and BETA (these constants are positive or negative real numbers)

   example:

ALPHA	BETA
1	-4.5

or

ALPHAA	ALPHAB	BETA
1.3	2.4	0

or

ALPHA	GAMMA	BETA
2	0.1	0

6) enter 'Y' (Yes) to use ARITH again or 'N' (No) to stop.

NOTE: the output numeric range is 0 - 32766; positive values only (16 bits/pixel integer).

BADLINE

BADLINE replaces bad scanlines in an image file through interpolation of the adjacent scanline above and below the bad line or lines. Linear interpolation of the scanline pixels immediately above the bad line or lines with the scanline pixels immediately below, creates the pixel values which then replace the bad scanline or scanlines pixels. Only adjacent single good scanlines are used above and below the bad zone. BADLINE is able to cosmetically correct multiple bad lines. Enter the bad lines of the image in order, from the top of the image, downward. This program will not correct bad first or last image scanlines.

To run, start program:

1) Answer DISKIO for input image filename, and scanline and pixel subsets.

2) enter output image filename

3) enter the first bad scanline
   (the top scanline of a bad zone if multiple lines)

4) enter the number of consecutive bad lines
   (1 if a single bad line, or the total number of bad lines within a bad zone)

5) enter more bad scanlines (Y/N - Yes or No)
   (press: <n> or <N> to terminate bad-line entry, or any other key to enter more)

NOTE: be sure to enter bad lines, ordered, from the top of the image, down

BANDRSQ

BANDRSQ classifies how well a BIL (band interleaved by line) image pixel spectra fits a given lab mineral or rock spectra. All pixels within the image are individually tested to generate two classified images, a goodness of fit (R-squared) image and a band absorption depth image. The spectral channels (bands) tested may be the whole spectra, or just a spectral interval; however, the lab mineral spectral interval must be of an absorption feature, with the starting and ending channel values being greater than the spectral values of the interval. Imaging spectrometry data such as AVIRIS (NASA) or GER (Geophysical Environmental Research) are usually used with this program to generate raw mineral theme maps.

The lab spectra file may contain the spectra of any desired material, but is usually a mineral, mineral combination, or rock spectra. This file contains two columns of ASCII data. The first column is the spectral wavelengths in real numbers, which at this time is read, but ignored. The second column is the spectral reflectance values in real numbers, ranging from 0.0 to 1.0 (0 - 100% reflectance).

Adjustments are made to both the lab spectra and the imaging spectrometry pixel spectra before classification of the pixel is performed. The continuum for both spectra are removed by dividing the channel spectral reflectance value by that spectra's continuum line value for that channel. The lab mineral spectra is then scaled to the same intensity as the imaging spectrometry spectra using a least squares method. Classification and band depth images are then generated.

Each pixel within the goodness of fit image is a raw classification of how well the corresponding imaging spectrometry data pixel fits the entered lab mineral spectra. When there is perfect overlap of the two spectral traces (both spectral shapes are identical) the pixel R-squared value is 1.000. When there is no correspondence between the two spectra, the pixel R-squared value is 0.000. Fits between no fit and perfect fit range between 0.000 and 1.000; and inverse fits are set to 0.000. The goodness of fit pixel is assigned this R-squared value times a multiplier scaling constant of 200 (RSQ * 200.0). Pixel values in the image range from 0 (no fit) to 200 (perfect fit). The standard mathematical formula for R-squared goodness of fit is used. This classification image is useful for identifying mineral occurrences within a data scene.

Pixel band absorption depths are also calculated during classification. Pixel band depths are calculated by subtracting the minimum intensity of the continuum removed spectral interval from 1.000 of the lab mineral spectra, which usually contains much less noise than the imaging spectrometry image. The band depth image is useful for yielding a very rough value of the intensity or quantity of the material per pixel.

To run, start program:

1) enter the continuum starting channels
   EXAMPLE: 4 6

   NOTE: the two channels for starting the spectral interval may be the
   same, or form an interval which then is averaged

2) enter the continuum ending channels
   EXAMPLE: 22 24

   NOTE: the two channels for ending the spectral interval are averaged as above

3) answer DISKIO for the input image filename (observation spectra), scanline and pixel subsets.
   EXAMPLE: 0 0 0 -1
   0 0 0

   NOTE: -1 must be used; selects all channels in BIL image
   (the starting and ending channels control the bands used)

4) enter library spectra filename
   (Make sure in the last column of screen output (continuum removed spectra), numbers do not exceed 1.000 within the spectral interval - elsewise start over, the continuum line is falling below the spectral curve.)
5) enter bad channel number
   (if any channels are to be excluded from test, keep entering bad bands until done, terminate loop with band of 0)
6) enter the goodness of fit output image filename
7) enter the band absorption depth output filename

BILCAL

BILCAL is used to calibrate band interleaved by line (BIL) imaging spectrometry data. BILCAL applies multiplicative and additive constants from an ASCII text file to the raw input BIL file. Every pixel of each band of the raw input file is multiplied by its corresponding multiplicative band constant and summed with its additive band constant, generating an output DN range from 0 to 20,000. DN 0 equals 0 percent reflectance, while DN 20,000 equals 100 percent reflectance.

Values in - range: (Raw Data * mult constant) + add constant = 0 to 1.00
Values out - range: I*2 0 - 20,000 [0 - 100% reflectance]

cal DN = ((raw DN * mult constant) + add constant) * 20,000

The ASCII text file is composed of 2 columns of ASCII data, the first being the multiplication real number constant column and the second being the additive real number constant column. Band one calibration constants form the first line of the file, while the last band forms the last line of the file. Input file real numbers may be stored in any format.

All ASCII real numbers must be in the numeric range of 0.0 - 1.0. Pixel DN output is limited to a numeric range of 0 - 20,000 (DNs < 0 are set to 0; DNs >20,000 are set to 20,000). The maximum number of bands used are 256.

To run, start program:

1) answer DISKIO for BIL image name, and scanline and pixel subsets (use -1 for band).

2) enter ASCII calibration filename.

3) enter output BIL filename

(the mult. and add. constants for each band will be displayed)

NOTE: the ASCII text file may be generated with SPECAVG and/or CALFILE.

BOX

BOX applies a box smoothing function to the input image. It does this by using a running average box of N pixels by N scanlines, where N is the number of pixels or scanlines per edge of the box. The average of the DNs within the box is placed in the center pixel of the output file. (If N is even, the average will be placed in the pixel below and to the right of the box center.)

At the image edges where the box is not filled by data, the pixels are not filtered, but directly copied from the input image. This unfiltered edge may be calculated as: # lines (or pixels) = (N-1)/2. When N is an even integer, the top and left margins contain one extra scanline or pixel of unfiltered data compared to the bottom and right margins.

Box's averaging function takes the form of a hi-pass smoothing filter (spike suppressor). This smoothing function is formed by three arguments.

1) **Averaging function.** This function sums up all pixel DNs within the box, divides this sum by N squared (N * N), and assigns this average DN value to the center pixel of the box in the output image. This averaging function is modified by steps 2 and 3 below.

2) **Minimum pixel DN threshold.** Any pixel DN below this threshold is not counted (summed) within the average of step 1. (The sum is still divided by N*N.)

3) **Minimum number of pixels per box threshold.** This function uses the pixels counted in step 2 to decide whether the DN of the center pixel of the output image is assigned an averaged DN or a DN of 0. When the pixel count is less than the cutoff, the output DN is set to 0. Where the pixel count is equal to or greater than the cutoff value, the output DN is assigned an average value of all the pixels which were counted in step 2 (i.e. - only the pixels which had pixel DNs greater than or equal to the minimum pixel DN threshold in step 2 have their pixel DNs summed and averaged).

Steps 2 and 3 are very useful for smoothing salt and pepper images but may be turned off by using values of 0.

To run, start program:

1) answer DISKIO for input image filename, and scanline and pixel subsets.

2) enter the number of pixels per side of the box.

3) enter the minimum pixel DN threshold cutoff value (0 to select all pixel DNs to average). [values range from 0 to maximum DN]

4) enter the minimum number of pixels per box cutoff value (0 to select all pixels within the box). [values range from 0 to N squared]

5) enter an output image filename

BYTEFLIP

BYTEFLIP transposes byte pairs of 16 bit data. Byte one is replaced by byte two, while byte two is replaced by byte one, byte three by byte four, etc. The position of the low and high bytes of 16-bit integer data are reversed. This program is useful for converting data between computer systems. HP, SUN, and Perkin Elmer computers use low byte then hi byte data format. VAX and PC computers use high byte then low byte format.

To run, start program:

1) answer DISKIO for image filename, and scanline and pixel subsets.

2) enter output filename

CALFILE

CALFILE generates the multiplicative constants that are used for calibrating BIL imaging spectrometry data to absolute reflectance. Input files are a raw BIL imaging spectrometry data file and an ASCII laboratory measured spectrum of a physical sample from a calibration site. Spectra of the pixels of the calibration site are extracted from the raw BIL image file and averaged. The ASCII lab spectrum is divided by the average site image spectrum to make the multiplicative constant (or multiplier). This will produce a multiplier for each band of data.

The sample site ASCII input (calibrated lab spectra) file contains two columns of real numbers, the first column contains the wavelength for the band, while the second column is the spectral reflectance value. The wavelength values are read, but not used within the program. The spectral reflectance values range from 0.0 (0% reflectance) to 1.0 (100% reflectance).

The ASCII calibration output file is composed of two columns of real numbers, the first is the multiplier band value, while the second column is the additive band column. CALFILE sets the additive column values to zero. For both the input spectra and output files the first row of data corresponds with band 1 and the last row to the last band. Output file real numbers are stored in scientific notation (FORTRAN E format).

To run, start program:

1) answer DISKIO for raw BIL image name, and scanline and pixel subsets

2) enter BIL image sample site coordinate (scanline and pixel)
   (enter only one pixel at a time)

3) enter 0 0 to end sample site coordinates
   (elsewise program loops back to step 2 - this loop allows multiple pixel spectra to be read and averaged for a sample site)

4) enter ASCII lab spectra (calibration site) filename

5) enter ASCII calibration output filename

Note: the program accepts three different types (styles) of subsetting.
The program will automatically calculate and start the BIL image file on band one of any image scanline.

Example 1: full BIL image file used:

```
     all bands entered -                   0 0 0 -1
                                           0 0 0

     or single band (band 20) entered -    0 0 0 20
                                           0 0 0
```

(either subset above generates the same results)

Example 2: image subsetting, 2 skips, 224 band image: [671 = (224*(2+1))-1]

```
     all bands entered -                   0 0 671 -1
                                           0 0 2

     or single band (band 20) entered -    0 0 2 20
                                           0 0 2
```

(either subset above generates the same results)

```
Example 3: SPECIAL CASE: scanline and
           pixel skips identical
           (here skips = 2):                  0 0 2 -1
                                              0 0 2
```

(this example generates the same result as example 2, single band)

In all cases, scanline and pixel subsetting is allowed, but full BIL (all channels) must be used.

CALFILE is used as input for SPECAVG and/or BILCAL

COMPARE

This program compares two 'REMAPP' image files or subsets of files to check for identical data. The 'REMAPP' headers are not compared and may be different. This program is useful for prototyping processing procedures or windowing and verifying the results. File subsets are selected through DISKIO during file selection below. Both files or file subsets must have the same number of bytes per scanline.

To run, start program:

1) answer DISKIO for first image name, and scanline and pixel subsets.

2) answer DISKIO for second image name, and scanline and pixel subsets.

If both image files or file subsets are identical, byte for byte, the message below is displayed:

```
DISKIO:END OF FILE ON first image filename
FILES SUCCESSFULLY COMPARED
```

If the two images or image subsets are not identical, the program returns a message stating in which record the first difference is detected.

CONCAT

This program concatenates 'REMAPP' image files to each other, from left to right, creating one image file. A maximum of 10 input files may be concatenated at one time, with a maximum output image of 8192 pixels/scanline. The number of scanlines of the concatenated output image file is truncated to the number of scanlines in the shortest input file. The bits/pixel of the output file is set to be identical as the first input file.

To run, start program:

1) Enter the number of input image files to concatenate.

2) Answer DISKIO for the first input image filename, and scanline and pixel subsets.

3) Answer DISKIO for the next input image filename, and scanline and pixel subsets.

4) Repeat step 3 until all of the image files are used.

5) Enter concatenation output image filename

6) Answer 'Y' (Yes) or 'N' (No) for further file concatenation

DISPLAY

This program displays images on any IBM compatible PC with 640 kilobytes of system memory and a VGA graphics card and monitor. Color and black-and-white images are displayed in a 320 pixel by 200 scanline mode, with 64 shades of gray or 256 colors. Six image buffers allow flicker comparison for geometric and radiometric differences. Cursor control displays scanline and pixel coordinates and DN value of a selected pixel. Eleven graphic planes with eight display colors enable theme mapping of DN intervals. Multiple images may be placed anywhere on the screen and/or overlaid by the windowing feature. Images may be enlarged by a factor of two (zoomed by 2X) as many times as desired. Enlargements may be saved in any of the image buffers, along with their scanline and pixel coordinates. Unlimited image size (up to 8192 pixels/scanline) may be displayed through interactive pixel subsetting within DISKIO.

Full 8-bit DN range (0 - 255) for gray scale images is maintained, though only 64 shades of gray are visually displayed on the screen. DN values displayed with color images are from the color palette (color coded) for the selected pixel. 16-bit image data is divided by a constant which reduces the numeric DN range to fit within the 8-bit 0 - 255 range. The default constant is 100, so that 16 bit DNs of 25,500 become 255, and linearly, 0 remains 0. This 16 bit DN range (0-25,500) may be rescaled using the <N> function.

Image files must be in REMAPP-PC format. A non-'REMAPP' single band image or a band interleaved by line image file may be converted to REMAPP-PC format using ADDHDR or STRIPHDR.

IMPORTANT - several points need to be clarified.

a) the <enter> key is used ONLY where listed here, when running DISPLAY.
b) DISPLAY should be started with the computer displaying white characters on a black screen (default mode). If a character color modifying program has been run, the text dialog will not be viewable.
c) When memory is limited, image buffers are disabled, one at a time, until the memory limits are honored. Disabled image buffers are listed during the display of the first image.
d) If a nonexistant image name is used, the screen will display a blank (black) image. Press <enter>, then use any subroutine as normal, such as the image buffer <I> to call up any previously stored image or display <D> to read in a new image.

To run, start program: (type: DISPLAY)

1) answer DISKIO for image name, and scanline and pixel subsets, terminating each input line with <enter>. A gray single band image is then displayed on the screen. The program is in the display mode, where various options may now be selected, such as using the cursor, graphics overlays, or windowing subroutines. When scanline/pixel defaults <0 0 0 0>, <0 0 0> are used, only the upper left corner of the image is displayed. (Description of options follow this page.)

2) to quit the DISPLAY program, press the <esc> key while in the display mode. If the program is in a subroutine, such as the cursor subroutine, finish completing any data entry, then press the <enter> key to cancel the subroutine and return to the display mode, then press the <esc> key.

3) Program options (subroutines) are image buffers, cursor, graphics overlays, image display, image windowing, image zooming, color image generation, gray - color palette control, and temporary exit to DOS. More options include masking, contrast stretching, scatter plots, spectra plots and scaling, image histogram display, and 16 bit pixel DN scaling. These subroutines are called by pressing the designated keys on the keyboard. The case of the key is not important (except for <f-F>). Do **not** press <enter> after pressing the subroutine selection key. Subroutine selection keys are:
<esc>,1-6,A,B,C,D,E,f,F,G,H,I,K,M,N,O,P,R,S,V,W,X,Y,and Z.

List of Subroutines

<esc>	terminate DISPLAY program
<1>-<6>	display image buffer <#>
<A>	Apply a graphics plane
<B>	reset Black and white palette
<C>	Cursor display
<D>	Display image
<E>	temporary Exit to DOS (DISPLAY still running in memory)
<f>	frequency histogram of screen image
<F>	Frequency histogram of image buffers
<G>	assign a Graphics plane
<H>	Help
<I>	store Image to a buffer
<K>	reset color palette
<M>	Make color image
<N>	set 16 bit Numeric range
<O>	mask Overlay
<P>	Plot spectra
<R>	Reset graphics planes to null
<S>	Subtract graphics plane
<V>	automatic stretch
<W>	display Windowed image
<X>	scatter plot
<Y>	spectra plot scaling
<Z>	Zoom image

SUBROUTINES

QUIT Program:

<esc> Terminate the DISPLAY program, return to DOS.

NOTE: <esc> means press the escape key.

Select and Display Image Buffer:

<1> Select and display image buffer 1. The currently displayed image is cleared from the screen (but not from any of the buffers). No parameters are used. DO NOT use <enter>. The screen will blank momentarily if the color palette changes from gray to color or vice versa.

<2> Select and display image buffer 2

<3> Select and display image buffer 3

<4> Select and display image buffer 4

<5> Select and display image buffer 5

<6> Select and display image buffer 6

Apply Graphics Plane:

<A> Apply a graphics plane to the displayed image. Enter a graphics plane number, 0 through 10, then <enter>. A graphics plane must be assigned using <G> above before it may be applied.

<number> <enter>

EXAMPLE: 10 <enter>

Related keystrokes: B,G,K,R,S

Reset black and white palette

<B> Resets the display palette to use the black and white (gray) palette, restores the image without graphics overlays, and resets the graphics overlays (they become undefined). No parameters are used.

Related keystrokes: A,K,R,S

Cursor:

<C> Select and display cursor, with scanline, pixel, DN, and selected image buffer (for DN value) shown in a window. The keyboard arrow keys along with <home>, <Pg Up>, <Pg Dn>, and <End> control the cursor movement direction. <Del> re-centers the cursor at the middle of the screen.

```
Home    ^    Pg Up
        |                   Image Buffer
                         <1, 2, 3, 4, 5, 6>
 <-    Del    ->
                               Screen
        |                       <0>
End     v    Pg Dn
```

Keys <0, 1, 2, 3, 4, 5, 6> select which image buffer is read for its DN value, with <0> being the screen, and <1> through <6> selecting image buffers 1 through 6, while the current screen image is being displayed. This is useful when one image is being displayed because of its spatial characteristics or color, while an image in a buffer is being analyzed. For example looking at a color ratio composite image while measuring the DNs from a hue image, or looking at a single gray band image while analyzing a gray ratio image. The cursor and displayed characters have DN values of 255 and may be colored using the graphics overlay subroutine (<G>).

The origin for the image is in the upper left corner, starting at scanline 1, pixel 1. Scanlines increase downward, while pixels increase to the right. Zoomed images retain their original coordinates, with the image origin now being in virtual space to the upper left. Also, pixel location and DN values are read from the center of the cursor.

DN values from 16 bits/pixel data are scaled to 8 bit data when the input image was read in with Display <D>, Window <W>, or Plot <P>; consequently, the display DN value is a scaled 8 bit representation of the 16 bit original value. To calculate the approximately true 16 bit DN value, multiply the displayed DN by 100 (DN 255 = 25,500). If the 16 bit numeric range has been redefined with <N>, the multiplier constant will be different. Note - for high contrast 16 bits/pixel images, either use the full numeric range (0-25,500), the <N> 16 bit rescale function, or scale the image to 8 bits/pixel data with HIST and STRETCH.

NOTE: <enter> returns the user to display mode

Display, <D> Image:

Display a black and white image from a disk file. The current displayed image is erased (save to image buffer if desired). Images are positioned starting in the upper left corner of the screen (screen coordinates 1,1). Use <E> Temporary Exit subroutine before displaying a new image to use any DOS command, such as the DOS DIR command, if needed (see <E> below).

Answer DISKIO for image name, and scanline and pixel subsets.

```
filename <enter>
scanline subset <enter>
pixel subset <enter>
```

```
EXAMPLE:     ANYWHERE.B3 <enter>
             1 200 0 0 <enter>
             1 320 0 <enter>
```

NOTE: DISKIO subsets may be larger (or smaller) than 200 scanlines by 320 pixels. The image subset will be truncated in the display to the first 200 scanlines by 320 pixels. The displayed image can be saved to an image buffer using the image buffer <I> subroutine described below.

see Appendix B DISKIO - User Interface for explanation of how image files are entered and used by the DISKIO subroutine

Related keystrokes: I,W,1-6

Temporary Exit:

<E> DISPLAY is temporarily halted and exited, with processing control being transferred to DOS (second COMMAND.COM child process). Save the screen image to an image buffer before temporarily exiting, if desired. Any command or program may be run which fits in the remaining system memory (~60 kilobytes - check with CHKDSK). The system environment, paths, files, and/or subdirectories may be changed before DISPLAY is re-entered. These changes are valid for the current child process, but will not affect the main (original) process when returned to with EXIT. At this point, all of the program and displayed images are in memory and do not need further access to any information stored on the hard-disk (except if Plot <P> is in use).

To return to DISPLAY, type EXIT <enter>. The program with all variables, including the image buffers, remain intact. The screen image is cleared, and the graphics planes are reset.

NOTE: <E> will not transfer control to DOS if there is insufficient memory available.

Frequency Histogram of Screen Image:

<f> Plot a pixel DN histogram of the screen image. The plot may be saved in any image buffer. The pixel DN interval axis (X-axis) ranges from DN = 0 at the left to DN = 255 on the right with interval steps of one DN. The frequency count axis (Y-axis) varies proportionately from 0 pixel counts per DN interval at the bottom to the maximum image pixel count per DN interval at the top. The axis and plot are assigned a DN of 255 and may be colored using the graphics overlay function <G> and <A>.

Use: display image on screen
press <f>

NOTE: use lower case only

Frequency Histograms of Image Buffers:

<F> Plot pixel DN histograms of every active image buffer. The plot may be saved in any image buffer. The pixel DN interval axis (X-axis) ranges from DN = 0 at the left to DN = 255 on the right, with interval steps of one DN. The frequency count axis (Y-axis) varies proportionately from 0 pixel counts per DN interval at the bottom to the maximum image pixel count per DN interval at the top. The axis is assigned a DN of 255. Each image buffer histogram is assigned a DN of 255 - image buffer number (image buffer 1 DN = 254, buffer 2 DN = 253, buffer 3 DN = 252, buffer 4 DN = 251, buffer 5 DN = 250, and buffer 6 DN = 249). These histogram plots may be colored with different (or the same) colors using the graphics overlay function <G> and <A>.

Use: store image or images into image buffers (using <I>)
press <F>

NOTE: use upper case only

Graphics, Plane Assign:

<G> Assign a graphics plane (0 through 10), a color, and a range of DNs.

1) enter graphics plane number and carriage return:
<desired number> <enter>

2) enter a color for the selected graphics plane
DO NOT use <enter>

<R>	Red
<Y>	Yellow
<G>	Green
<C>	Cyan
<B>	Blue
<M>	Magenta
<D>	Dark (black)
<W>	White

3) enter a DN range (0-255) to which the graphics plane will be applied, then press <enter>. The second DN number must be greater than or equal to the first DN number.

<1st number> <space> <2nd number> <enter>
EXAMPLE: 20 50 <enter>

Graphics planes are ordered with plane 0 being the first (bottom) plane, continuing to plane 10 as the last (top) plane. Graphics are applied from the first to last plane in order, so that later graphics may overwrite previous graphics with the same DNs.

Related keystrokes: A,B,K,R,S

Help:

<H> Display the help screens that list all of DISPLAY's major subroutine options.

Image, Store in Buffer:

<I> Store an image to an image buffer, 1 through 6. Enter a number between 1 and 6. press key:

<desired number> <enter>

NOTE: storing a displayed image into a buffer is extremely useful in case an option overwrites the screen image. The original image may be instantly recalled.

Reset Color Palette

<K> Resets the display palette to use the color palette, restores the image without graphics overlays, and resets the graphics overlays (they become undefined). No parameters are used.

Color Image:

<M> A color composite image is created and displayed. It can be saved to an image buffer. Image buffers 1, 2, and 3 must contain black and white images, else this option is ignored. Image buffer 1 contains the blue input image, buffer 2 contains the green image, and buffer 3 contains the red image. These three buffers can be loaded using the display <D> option. A maximum of 8 shades (intensities) of red and green are used, and a maximum of 4 shades of

blue are used, yielding a maximum of 256 colors. No parameters are used.

Numeric Range:

<N> Assign the maximum 16 bits/pixel numeric value. This defines the numeric DN range for scaling 16 bits/pixel image data into 8 bits/pixel space. Subsequent Display <D>, Window <W>, and Plot <P> operations will linearly rescale 16 bits/pixel DNs into 8 bits/pixel range using the new values. The startup default is to linearly scale the 16 bit data range of 0 to 25,500 to the 8 bit range of 0 to 255.

<number> <enter>
EXAMPLE: 10000 <enter> (scales 0-10,000 to 0-255)

Care must be used when plotting spectra (<P>) and assigning maximum 16 bit values. The internal representation of absolute reflectance values for 16 bit data are: Display DN of 0 = 0% reflectance (16 bit DN 0); and Display DN of 200 = 100% reflectance (16 bit DN 20,000). 16 bit DNs from 20,001 to 25,500 are truncated to 100% reflectance (DN=20,000). To set the maximum 16 bits/pixel numeric value DN (assigned by <N>) for use in a calibrated data file with 100% reflectance assigned something other than 20,000, the maximum numeric value DN may be calculated as:

(max 16 bit DN) = (100% reflectance 16 bit DN) * 1.275

so if your data has 100% reflectance set to a DN of 5000, the maximum numeric value set by <N> should be 6375. This way, within DISPLAY, any DN cursor values of 200 will equal 100% reflectance (and DN 0 = 0%).

Overlay, Mask:

<O> Uses background, overlay, and mask images stored in the image buffers, to generate a screen overlay image. A background image may be selected from image buffers 1 through 6, or as a defined DN value. The overlay image may be selected likewise. The mask image may only be selected from image buffers 1 through 6.

Once the background, overlay, and mask images have been selected or the DN value defined, and the mask DN interval is selected, the output image is generated and displayed on the screen. The output image is generated, pixel by pixel, by testing the mask image against the selected mask DN interval and then writing to the screen, either the background image or the overlay image pixel. If the mask pixel under test falls within the defined mask DN interval, then the overlay pixel DN (from the image or defined value) is written to the screen. If the mask pixel under test falls outside the defined mask DN interval, the background pixel DN is written to the screen.

Use:

1) press <O>
2) press the image buffer number for the background image (or <9> to define a DN value as the background DN)
3) press the image buffer number for the overlay image (or <9> to define a DN value as the overlay DN)

4) press the image buffer number for the mask image
5) enter the lower and upper DN values for the mask range
6) enter the DN value for the background or overlay (only if <9> was selected in step 2 or 3)

Example 1: image overlay
1) <O>
2) 4
3) 1
4) 2
5) 200 255 <enter>

displayed is an overlaid image of buffer 1 (overlay image) on top of an image of buffer 4 (background image) where pixel DN values of image buffer 2 (mask image) range from 200 to 255.

Example 2: DN defined overlay
1) <O>
2) 4
3) 9
4) 2
5) 200 255 <enter>
6) 255 <enter>

displayed is an image like example 1 above, except that the overlay DNs are all 255.

Spectra Plot:
<P> Plot pixel spectra from a band interleaved by line (BIL) file. Any pixel spectra within the BIL file may be plotted by positioning the cursor over the desired displayed image pixel and pressing <P> to generate the spectra plot. Image subsetting or windowing is available, with up to 224 channels of a spectra displayed.

The X-axis displays the selected 224 spectral channels (bands) while the Y-axis displays the channel values in units of % reflectance with starting values from 0 to 100%. 0% reflectance corresponds to a 16 bits/pixel DN value of 0, while 100% reflectance corresponds to a 16 bit DN value of 20,000. Likewise, with 8 bits/pixel data, 0% reflectance corresponds to a DN value of 0, while 100% reflectance corresponds to a DN value of 200. DN values higher than 100% reflectance are set to 100%. The Y-axis may be re-scaled using the spectra plot Y - axis scaling subroutine <Y> or the 16 bits/pixel numeric range scaling function <N>. It is more useful to use the <Y> scaling function to re-scale the Y axis in most cases because the absolute reflectance calibration is maintained. When data files are calibrated so that 100% reflectance is assigned to something other than DN 20,000, use <N> to reassign the 16 bit numeric range for 100% reflectance before plotting with <P>, then rescale with <Y> if desired.

NOTE: this is a very specialized remote sensing function for use with imaging spectrometry data, i.e.: AVIRIS, HIRIS, or GER-ASIS.

Use: once in the program (i.e.: displaying any image)

1) press <P>
2) enter the BIL filename (i.e.: IMAGE.BIL) <enter>
3) enter scanline subset (to position a reference image and point to the starting spectra band).

   NOTE: use the single band option to display the desired single band image, i.e..: 101 0 1 20, to display band 20, starting at image scanline 101 and skipping every other image scanline. The starting channel is automatically calculated to be band 1 of the starting image scanline. Band 20 would be displayed, and the spectra displayed would range from band 1 to the last band (or band 224, which ever is smaller).

   101 0 1 20 <enter> (skip every other image scanline)

4) enter pixel subset (normal REMAPP subset of image)

   i.e.: to start at the left edge of an image, read in the next 640 pixels skipping every other one for a 320 pixel display using the 0 defaults:

   0 0 1 <enter>

5) a raw image of the starting band is displayed (NOTE: it is usually very dark to black)

6) save the image to any image buffer by entering an image buffer number from 1 to 6

   i.e.: 1 (to store the image into image buffer 1)

7) a cursor is displayed

   move the cursor to the desired pixel to generate a spectra and press <enter>

8) a pixel spectra is displayed

ENHANCEMENTS:

After a spectra is displayed - display the original BIL image using the image buffer display keys <1> - <6>.

Stretch the original dark BIL image using <V> and store the stretched (brightened) image into a free image buffer using <I>.

Use the stretched BIL image for locating further pixel plots (display the stretched image and press <P> to generate additional spectra plots). Once the BIL file is opened, <P> repeats steps 7 and 8 above. Subroutines <D>, <E>, and <W> will close the BIL file.

Plots may be overlaid onto each other and/or saved into any image buffer. Plots have DNs of 255 for coloring with the graphics overlay subroutine <G>. The Y-axis may be scaled in % reflectance using <Y>.

NOTE: To display a spectra starting at anything other than channel 1, the BIL file would have to be defined as a single band using the band -1 option, and the record and skip position closely monitored (i.e.: 11 0 223 -1). This example starts at band (record) 11 of image scanline 1 (not image scanline 11, because of the -1). This method with the -1 is not recommended unless really needed.

Related keystrokes: G,I,Y

Reset graphics:

<R> Restore image without graphics. Graphics planes are removed from the display. All graphics planes are reset (undefined). No parameters are used.

Related keystrokes: A,G,S

Subtract Graphics Plane:

<S> Subtract a graphics plane from the displayed image. Enter a graphics plane number, 0 through 10, then <enter>. A graphics plane must be assigned (using <G>) and applied (using <A>) before it may be subtracted.

<number> <enter>
EXAMPLE: 10 <enter>

Related keystrokes: A,G,R

Automatic Stretch:

<V> The screen image is contrast stretched. The screen image is saved to image buffer 6, overwriting anything presently there. An approximate two percent, two piece linear stretch is then applied to the image in buffer 6, yielding an output image displayed on the screen. This subroutine is useful for brightening a dark raw data file. For a custom contrast stretch, run the stand-alone programs HIST and STRETCH. For 16 bits/pixel data, <N> may be used to reset the DN numeric range.

Windowing, Image:

<W> Display a new image from an image file read off the hard-disk. The currently displayed image is over-written only where the new image window is placed. The display palette is set to the black and white (gray) palette if the display presently is using the color palette.

The window is positioned using the cursor.

1) The upper left corner of the window is marked by positioning the cursor to the upper left corner location using the arrow (cursor) keys, then pressing <enter>.
2) The lower right corner is then marked by positioning the cursor at the lower right corner location and pressing <enter>.

3) Answer DISKIO for image name, and scanline and pixel subsets.

EXAMPLE: ANYWHERE.B3 <enter>
1 200 0 0 <enter>
1 320 0 <enter>

NOTE: DISKIO subsets may be larger (or smaller) than the size of the window. The image will be truncated to fit the display window starting from the upper left corner.

Scatter Plot:

<X> Plot pixel DNs of the first image against the pixel DNs of the second image. Image buffers 1 and 2 must contain images (usually two bands of the same image). The X-axis uses pixel DNs from image buffer 1, while the Y-axis uses pixel DNs from image buffer 2. Highly correlated images form a scatter plot with a narrow linear trend, usually sloping at 45 degrees.

Use: store images in image buffers 1 and 2 (using <I>)
press <X>

Spectra Plot Scaling

<Y> Scale the Y-axis of any spectra plotted using <P>. The Y-axis is in units of % reflectance with default values from 0 to 100%. 0% reflectance corresponds to a 16 bits/pixel DN value of 0, while 100% reflectance corresponds to a 16 bit DN value of 20,000. Likewise, with 8 bits/pixel data, 0% reflectance corresponds to a DN value of 0, while 100% reflectance corresponds to a DN value of 200. DN values higher than 100% reflectance are set to 100%.

Use: 1) press <Y>
2) enter the low and high % reflectance values
(i.e.: 10 80 <enter> - means display spectra values from 10% to 80% reflectance within the Y-axis).

Zooming, Image:

<Z> Enlarge the displayed image by a factor of two. The displayed image is automatically saved in buffer 6, overwriting any previous image there. The enlarged image is displayed and may be saved into any image buffer and/or further zoomed. Scanline and pixel coordinates, as well as pixel DNs, are saved and remain unchanged on the zoomed image. They correspond to the same pixels on the original image and may be used for close-up location and DN value determinations. Cursor position during the zoom operation determines how the zoomed area is selected. The pixel position under the cursor, of the original and enlarged images will be displayed at approximately the same point on the screen. Edges of the zoomed image are selected proportionately to how close the cursor is positioned to the edges of the original image. When the cursor is positioned at an edge of the original image, the enlarged image will begin at that edge.

The image is enlarged by positioning the cursor to the desired location using the cursor keys and pressing <enter>.

EDGES

EDGES calculates edge enhanced images to emphasize contrast differences of edge features. A central pixel's value is modified by a weighting factor applied to the central pixel and to the surrounding pixels adjacent to its four edges (Laplacian function). There is a one pixel unfiltered margin around the image.

<u>Filter</u>

```
  n
w c e
  s
```

Laplacian function:

(Enhanced DN of c) = c + weighting factor * (4c -n -s -w -e)

Example: weighting factor = 0.5

original data

4	4	4	2	2	2
4	4	4	2	2	2
4	4	4	2	2	2
4	4	4	2	2	2
4	4	4	2	2	2
4	4	4	2	2	2

enhanced data

4	4	4	2	2	2
4	4	5	1	2	2
4	4	5	1	2	2
4	4	5	1	2	2
4	4	5	1	2	2
4	4	4	2	2	2

Note: 1 pixel unfiltered edge

To run, start program:

1) answer DISKIO for input image filename, and scanline and pixel subsets.
2) enter an output filename
3) enter an enhancement (weighting) factor
   (usually 0.15 works well; the higher the number the grainier and higher the contrast)
4) enter 'Y' (Yes) to run the program again, 'N' (No) to exit.

ENLARGE

ENLARGE enlarges 'REMAPP' images by an integer amount through pixel and scanline duplication. This is useful for magnifying part of an image for hard copy output.

To run, start program:

1) enter the integer enlargement factor (i.e.: the multiplication expansion factor).

2) answer DISKIO for the input image filename, and scanline and pixel subsets.

3) enter output expanded image filename.

EXAMPLE - enlargement factor = 3
The output image will be 3 times larger in the pixel
and scanline directions than the input image DISKIO subset.

GEOMCORR

GEOMCORR geometrically corrects Landsat MSS data for pixel over-scan distortion and orbit skew through pixel deletion and scanline shifting. Geometric rectification for Landsat MSS images is performed on a line-by-line basis for over-sampling in the scanline direction (aspect ratio correction), and the non-polar orbit of the satellite and the rotation of the Earth under the satellite (skew correction). Skewing of the image is performed by offsetting the scanlines an integer amount determined from the image latitude. Aspect ratio is corrected by deleting pixels from the scanline.

To run, start program:

1) answer DISKIO for the input image filename, and scanline and pixel subsets.

2) enter output image (geometrically corrected) filename

3) enter latitude in degrees, minutes, and seconds (as integers)
   Input latitude is the latitude at the top of the image
   (scanline 1).

   EXAMPLE - latitude: 39 50 41
      (39 degrees, 50 minutes, 41 seconds)

GRDREM

GRDREM converts USGS standard GRID files to REMAPP image files. USGS GRID files are unformatted real sequential data files which start at the bottom. GRDREM copies and reformats the GRID data into the REMAPP image format, which is binary integer direct raster data that starts at the top of the image. Input is four byte real number data; output is 8 bit (one byte) integer data. GRDREM describes the grid file (title, size), and the minimum and maximum grid values. These minimum and maximum values are mapped to DNs of 1 and 255. Grid values between the minimum and maximum are linearly interpolated to DNs between 1 and 255. DVALs (deleted value points) in the grid file are mapped to DNs of 0. The remapping function is displayed as well as logged in REMAPP.LOG.

To run, start program:

1) enter the USGS GRID filename
   (GRID file header record is displayed)

2) answer DISKIO for the output image filename
   (min and max GRID and REMAPP values are displayed, as well as the conversion function)

HIST

HIST creates histograms and calculates basic statistics from 'REMAPP' image files. These statistics are useful for describing the data, such as histogram shapes, DN populations, and DN ranges. Input images may be either 8 or 16-bits per pixel data with maximum DNs of 32,767. Statistical output may be directed to either the screen, a printer, or an ASCII file on the harddisk.

The output contains all the statistics and histograms generated during a run of HIST. This output contains two parts for every image file, the header with numerical statistics, and the trailer with graphical intensities. Output may be routed to the screen console (CON), the default printer (PRN), or to a ASCII harddisk file.

The header lists the minimum and maximum sample DN selected by the user (which must be within the range of 0 to 255 for 8 bit data and 0 to 32,767 for 16 bit data), the minimum and maximum DNs within the image (which can be outside the sampled DN interval), the sample mean, the sample median, and sample mode and mode DN count. Also in the header are the number of pixels used within the sample interval, the number of pixels total within the image (DISKIO subset), the number of pixels counted below the sample interval, the number of pixels above the sample interval, the sample variance, and sample standard deviation. The sample 1%, 2%, 98%, and 99% DN values are also listed.

The trailer lists the DN class levels, frequency of occurrence, cumulative frequency, and a graphical representation of the histogram with relative intensity increasing to the right. The sample mode DN is set to 100% intensity.

For faster HIST statistics generation of larger images, the image may be subset using skips within DISKIO, i.e.:

```
DISKIO subset
    0 0 5 0
    0 0 5
```

For smaller images, all the pixels within the image might be used (DISKIO with 0 skips). A specific region of the image may be targeted using pixel subsetting within DISKIO to generate spot statistics.

The trailer graphics start with the first histogram class interval which includes the lowest pixel DNs within the sample, while the graphics end with the last histogram class interval which includes the highest pixel DNs within the sample. When the image subset is smaller than the full image, the full image may contain pixel DN values which are lower or higher than the tested image subset.

HIST may be run multiple times in a session, with succeeding statistics appended to the bottom of the previous runs. When generating ASCII text files, once the program is terminated, no more data may be appended (within HIST). New runs must use a new text file name. HIST prompts for a filename when text files are requested.

Image data DNs usually form various bell shaped curves (normal distribution) which are easy to enhance for viewing. Contrast stretching, using image histogram data to determine a bi-linear stretch (i.e. 2% 50% 98%) usually creates visually pleasing images (see example 1 of STRETCH).

To run, start program:

1) select a listing device for statistical output (CON, PRN, or diskfile)
   (type: 'CON' to select screen console output)
   (type: 'PRN' to select output to a default printer)
   (type any new [operating system legal] filename for an ASCII text file)
2) answer DISKIO for input image filename, and scanline and pixel subsets.
3) enter a title (up to 80 characters) to label the histogram data
   (may be left blank - i.e.. just press <enter>).
4) enter the minimum and maximum DN values to test - press <enter> for maximum DN range
   (8 bit maximum DN range is 0 to 255; and 16 bit maximum DN range is 0 to 32,767)

   EXAMPLE: 11 236
   Pixel DN values from 11 to 236 are used within the histogram. Pixel DN values from 0 to 10 are not used in the frequency statistics. DN values of from 237 to 255 and greater (> 255 only with 16 bit data) are likewise excluded from the statistics.

5) enter the histogram class interval) - press <enter> for default interval
   (8 bit images default to an interval of 1 DN while
   16 bit images default to an interval of 10 DN)

   (class interval = the number of DNs which forms an interval over which the frequency of occurrence of matching pixels is summed.)

   EXAMPLE: 1
   A class interval of one DN is usually used with 8-bit data (0 - 255), while an interval of 10 might be more appropriate for 16-bit data (0 - 32,767).

6) answer <Y> - yes or <N> - no to generate statistics on another image
   (press <N> or <n> to terminate program, or any other key
   to continue with new images)

HUEMAP

HUEMAP performs masking operations which blend an input image and a defined DN value through control by a second (mask) image. Masking copies the defined DN (overlay) value to the output image when the location pixels on the hue (mask) image fall within the specified DN masking interval, else pixels from the landsat (background) image are used. HUEMAP uses for input, a hue (mask) image, a landsat (background) image, a replacement user defined DN (overlay) value and a DN masking interval defined by the user.

Note: Images must be registered to each other for masking operations.

To run, start program:

1) answer DISKIO for the input hue (mask) image filename, scanline and pixel subsets.
2) answer DISKIO for the input landsat (background) image filename, scanline and pixel subsets.
3) enter output filename.
4) enter Min. & Max. hue limits (DN masking interval)
   EXAMPLE: 120 190
5) enter substitution hue code (defined DN overlay value)
   EXAMPLE: 0

A 0 DN pixel is substituted into the output image when the hue (mask) DN is between 120 - 190. If the DN is outside this range, a pixel from the landsat (background) image is used instead.

MASK

MASK combines two registered images into one output. This combination is controlled by a mask image. The user selected DN range defines how the mask image combines the two input images (background and overlay). Either, but not both, input image may be defined as a homogeneous single DN image (a solid image with only one DN value) that is contained in memory (not on the hard-disk). MASK can do all the masking operations found within the programs SHADRAT, PIXELSUB, HUEMAP, and SATSTR combined.

Three images, background, overlay, and mask, are used in combination to form a theme masked output image. The rule for combining the background and overlay images together to form the output image is simple. DN pixel values from the mask image control the selection of which image (background or overlay) will contribute it's pixel DN value to the output image. When a pixel DN is tested in the mask image and found to be within the DN interval specified by the user, the DN from the same pixel of the overlay image is written to the output image. When the tested mask pixel DN is outside this interval, the background pixel DN is written to the output image. The user is prompted for this mask DN interval, along with the image filenames.

MASK may also use only two images to generate an output theme masked image. It does this by defining the necessary third image in memory as a solid homogeneous image with a single DN. This defined solid image may take the place of either the background or overlay image. The mask image must always be supplied from a file. Also, the mask image may be used simultaneously as the background or overlay image. So in some cases, only one image is used, as a background or overlay image, and as a mask image, with defined single DN substitution.

MASK inputs a Background Image, Overlay Image, Mask Image, DN mask interval, and if selected, DN mask value.

To run, start program:

```
1)  select whether or not to display program instructions (y/n).
2)  select whether: <1> both background and overlay images
                 or <2> a single background or overlay image with a DN constant
                          homogeneous image
       will be used (press <1> or <2>).
    if <2>, a single background or overlay image with a DN constant
       homogeneous image was selected, select whether:
                   <1> a background image
               or  <2> an overlay image
       will be used with the DN constant image

3)  answer DISKIO for the input image filename, and scanline and pixel
       subsets (input image: background image if used, else overlay image).
4)  depending if a DN constant homogeneous image is used or not:
    a) when not using a DN constant image - answer DISKIO for the overlay
          input image filename, and scanline and pixel subsets.
    b) when using a DN constant image - enter the DN value to assign to
          the constant image (which then becomes the homogeneous DN
          background or overlay image - which ever was selected above).
5)  answer whether the background or overlay image will also be used as
       the mask image: (y/n)
    if <y> and not using a DN constant image, choose:
                   <1> use the background image as the mask image
               or  <2> use the overlay image as the mask image
       Note: when using a DN constant image, the program will automatically use the
       single disk based (background or overlay) image correctly without prompting.

    if <n> answer DISKIO for the mask image filename, and scanline and pixel subsets.
6)  enter output filename
7)  enter DN mask interval
```

Example 1:
use three images (background - 'ratio.31', overlay - 'base.bw', and mask -'hue') to generate an output image. In this example, ratio.31 could be an iron oxide classified Landsat ratioed image, base.bw could be a single gray band of the same scene, and hue could be a CIR hue file where vegetation would have a DN interval of 210 - 270 (red values). This masking process would generate an output file which would be the iron oxide theme image ('ratio.31'), with a simple gray image ('base.bw') overlaid where vegetation occurs (defined by mask image 'hue').

```
mask
do you want instructions (y/n): n

Choose -Mask Image used with:
         <1>  Background and Overlay Images
         <2>  Input Image (Background or Overlay) and DN constant

Press Key 1 or 2: 1
Open Background Image
Diskio: enter input filename
ratio.31
Diskio (2):ratio.31   [lines/band   480 pixels  640 byte size 16 bands 1]
enter 1st scanline, no. lines/band, skips, band:
0 0 0 0
enter 1st pixel, no. pixels, skips:
0 0 0
input file ratio.31        opened:
window                   band   1st  no. skip actual
lines                       1     1  480    0    480
pixels                            1  640    0    640

Open Overlay Image
Diskio: enter input filename
base.bw
Diskio (2):base.bw    [lines/band  480 pixels  640 byte size 8  bands 1]
enter 1st scanline, no. lines/band, skips, band:
0 0 0 0
enter 1st pixel, no. pixels, skips:
0 0 0
input file base.bw        opened:
window                   band   1st  no. skip actual
lines                       1     1  480    0    480
pixels                            1  640    0    640
Is Mask Image also the Background or Overlay Image? (Y/N): n
Open Mask Image
Diskio: enter input filename
hue
Diskio (2):hue      [lines/band  480 pixels  640 byte size 16 bands 1]
enter 1st scanline, no. lines/band, skips, band:
0 0 0 0
enter 1st pixel, no. pixels, skips:
0 0 0
input file hue              opened:
window                   band   1st  no. skip actual
lines                       1     1  480    0    480
pixels                            1  640    0    640
Open Output Image
Diskio: enter output filename
ironwo.veg
Enter DN Mask Interval
210 270
Diskio (2):ironwo.veg [lines/band   480 pixels  640 byte size 16 bands 1]
Stop - Program terminated.
```

Example 2:
use two images (background - 'base.bw' and mask - 'hue') to generate an output image. The overlay will be a generated DN constant image (DN = 255). In this example, base.bw could be a gray single band Landsat image, and hue could be a CRC (Color Ratio Composite) hue file where bedrock iron oxides would have a DN interval of 45 - 110 (mid cyan through mid yellow values) for this particular scene. This masking process would generate an output image of a gray single band with iron oxide anomalies superimposed on it with a DN value of 255.

```
mask
do you want instructions (y/n): n

Choose - Mask Image used with:
         <1>  Background and Overlay Images
         <2>  Input Image (Background or Overlay) and DN constant

Press Key 1 or 2: 2

Choose:
         <1>  Background Image and DN constant
         <2>  Overlay Image and DN constant

Press Key 1 or 2: 1
Open Background Image
Diskio: enter input filename
base.bw
Diskio (2):base.bw    [lines/band  480 pixels  640 byte size 8 bands  1]
enter 1st scanline, no. lines/band, skips, band:
0 0 0 0
enter 1st pixel, no. pixels, skips:
0 0 0
input file base.bw          opened:
window                     band   1st  no. skip actual
lines                         1     1  480    0    480
pixels                              1  640    0    640

Enter DN for overlay value: 255

Is Mask Image also the Background or Overlay Image? (Y/N): n
Open Mask Image
Diskio: enter input filename
hue
Diskio (2):hue       [lines/band   480 pixels  640 byte size 16 bands 1]

enter 1st scanline, no. lines/band, skips, band:
0 0 0 0
enter 1st pixel, no. pixels, skips:
0 0 0
input file hue              opened:
window                     band   1st  no. skip actual
lines                         1     1  480    0    480
pixels                              1  640    0    640
Open Output Image
Diskio: enter output filename
mask.fe
Enter DN Mask Interval
45 110
Diskio (2):mask.fe    [lines/band   480 pixels  640 byte size 8  bands  1]

Stop - Program terminated.
```

MUNSEL

MUNSEL converts images in the red, green, blue color coordinates to Munsell hue, saturation, and value (intensity) color coordinates (tristimulus values). Input is three 8 or 16-bit red, green, and blue image files (RGB), output is three 16-bit hue, saturation, and intensity (value) image files (HSI). Figure 1 displays these color coordinate systems.

Hue output:
'Colors' range from 0 - 360
Gray (black to white) = 400

Hue output 'pure primary color' values:
Blue = 0
Green = 120
Red = 240

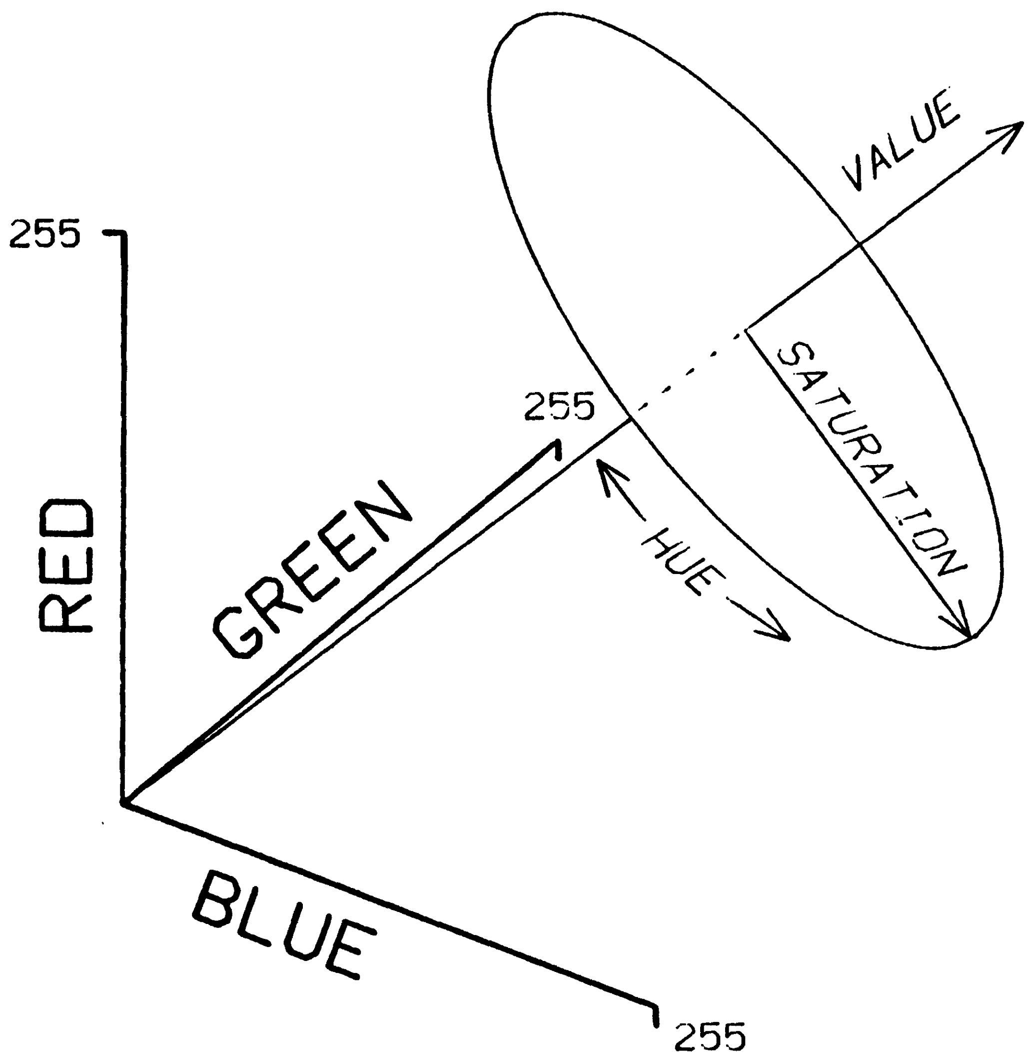
Output DNs:

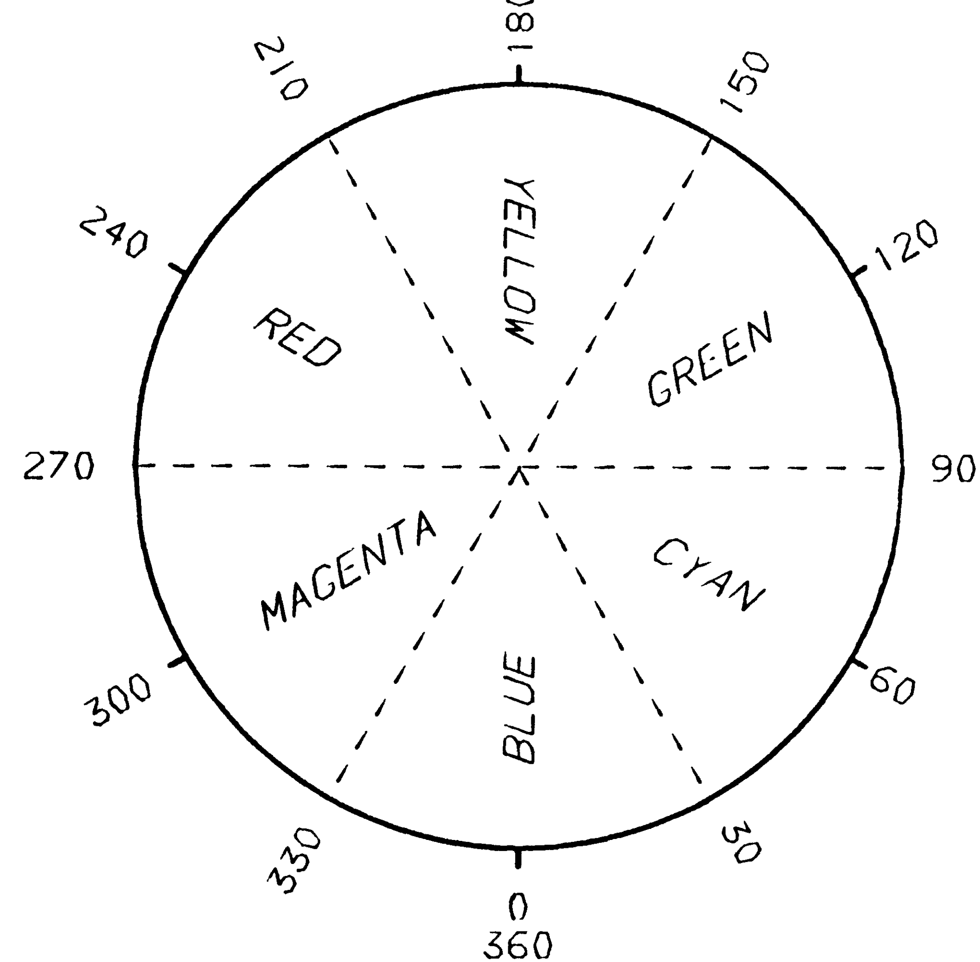
Hue DN range:	0 - 360, and 400
Saturation DN range:	0 - 212
Value (intensity) DN range:	0 - 442

To run, start program:

1) answer DISKIO for the red input image filename, and scanline and pixel subsets.
2) answer DISKIO for the green input image filename, and scanline and pixel subsets.
3) answer DISKIO for the blue input image filename, and scanline and pixel subsets.
4) enter a filename for the output hue image file.
5) enter a filename for the output saturation image file.
6) enter a filename for the output value (intensity) image file.

Note: TRISTM converts images to red, green, and blue coordinates.


A. GEOMETRY
255
RED
GREEN
255
BLUE
255
VALUE
SATURATION
HUE


180
210
150
240
120
YELLOW
RED
GREEN
270
90
MAGENTA
CYAN
BLUE
300
60
330
30
0
360

B. HUE

PIXELSUB

PIXELSUB performs masking operations which blend two input images through control by a third (mask) image. Masking copies pixels from the replacement (overlay) image to the output image when the same pixels on the template (mask) image fall within the specified DN masking interval, else pixels from the standard (background) image are used. PIXELSUB uses for input, a template (mask) image, a standard (background) image, and a replacement (overlay) image. A DN masking interval is defined by the user.

Note: Images must be registered to each other for masking operations.

To run, start program:

1) answer DISKIO for the input template image filename, scanline and pixel subsets.
2) answer DISKIO for the input standard image filename, scanline and pixel subsets.
3) answer DISKIO for the input replacement image filename, scanline and pixel subsets.
4) enter output (masked) image filename.
5) enter DN masking interval

PRIMG6

This program prints an image on an Epson MX compatible dot matrix printer. 17 shades of gray display a somewhat coarse image of limited utility. Image files must be in ASCII format, created with REMLST. Images can be unlimited in the number of scanlines, with individual panels up to 120 pixels wide. Several panels may be joined after printing to form one image. Shades of gray are generated by dithering dots within each pixel.

Black-and-white laser printer or color inkjet output from images 'grabbed' from the screen (being displayed with DISPLAY) with commercial paint programs such as Deluxe Paint II provide much better detail. Photographing the screen is another useful image output method. The most useful method is to transfer the image file to a professional filmwriter, electrostatic printer, or inkjet printer through hardwire (Ethernet, etc.), or through 9 track tape. This allows images of any size to be output.

To run, start program:
enter: the number of pixels per scanline, from 1 to 120
enter: image filename

RDEL

RDEL deletes single to multiple REMAPP image files from the disk. Up to nine filename arguments are accepted. RDEL tests for the existence of the data file within the image subdirectory, then deletes the header and data files and removes the directory. No wild cards are supported in the image name. When a non-existent name is encountered, RDEL terminates at that point. Separate image names with spaces.

To run:

1) enter: RDEL <filename arguments>

EXAMPLE 1: RDEL thisfile.b1

EXAMPLE 2: RDEL file1 file2 file3 file4 file5 file6 file7 file8 file9

EXAMPLE 3: RDEL filename ANYWHERE.IMG file4

REMBIL

REMBIL assembles single band images and/or single bands of band interleaved by line (BIL) images into a new band interleaved by line file. Both input and output files are in binary format. Each image used to assemble a BIL file is usually a single band image. See REMCOPY for single band image extraction and binary spectra extraction from a BIL image file. See Appendix C, File Formats, for a description of single band and band interleaved by line file descriptions.

To run, start program:

1) enter the number of bands the BIL output file will have (Total number of bands i.e.: 7 band BIL = 7)
2) answer DISKIO for input band 1 image name, and scanline and pixel subsets.
3) enter BIL output filename

after the input band 1 image file is copied:

4) enter the next input band image filename, and scanline and pixel subsets.

repeat step 4 until all input bands have been input.

When completed, statistics for the new BIL image file will be displayed. Compare this output with the number of lines and pixels desired to check that they match.

```
Diskio(2): filename.bil [lines/band  2100pixels  1900bytesize  8 bands  7]
```

i.e.: image filename.bil contains 2100 scanlines per band, 1900 pixels, at 8-bits/pixel format, and 7 bands.

REMCOPY

REMCOPY copies 'REMAPP' images or image subsets. A 'REMAPP' header record is generated, then an output image is copied from an input image or image subset. Both input and output images are in binary format. Single band images and spectra may be extracted from band interleaved by line (BIL) files. Image reduction can be accomplished through pixel skipping (subsampling). See REMBIL for BIL file assembly and EXPAND for enlargement. See Appendix B, DISKIO - User Interface for single band image and spectra extraction examples.

When copying whole image files, REMCOPY works just like the DOS COPY command, except a new 'REMAPP' header record is created for the output file.

To run, start program:

1) answer DISKIO for input image name, and scanline and pixel subsets.

2) enter an output filename

REMFILE

REMFILE generates a simple diagonal linearly increasing brightness test image file. Pixel DN values increase along the scanline from **0** to **# of last pixel - 1,** and down through the file from **0** to **# of last scanline - 1.**

Pixel DN = (pixel # -1) + (scanline # - 1)

The maximum pixel DN is 255 for 8 bits/pixel images or 32,767 for 16 bits/pixel images. The pixel DN then wraps back to 0. With band interleaved by line (BIL) images, every band in the scanline will contain the same values. This program is useful for generating known DN values for use in testing new programs and algorithms.

To run:

1) enter: the number of scanlines/file,
the number of pixels/scanline,
the number of bits/pixel, (either 8 or 16)
and number of bands (1 = single band; >1 = BIL)

2) enter output filename

EXAMPLE:

HOW MANY LINES,PIXELS,NBITS,NBANDS
200 320 8 1
Diskio: enter output filename
anywhere.img
Diskio(2):anywhere.img [lines/band 200 pixels 320 bytesize 8 bands 1]

Stop - Program terminated.

REMFLIP

REMFLIP flips an image top to bottom with no rotation. This program is very useful when converting data to be used with REMAPP, such as USGS GRID files, where the data file starts at the bottom of an image, and reads through the file to the top (REMAPP files start at the top and read down).

To run, start program:

1) answer DISKIO for the input image filename, and scanline and pixel subsets.

2) enter output flipped-image filename.

REMHDR

REMHDR displays the header information of a 'REMAPP' image file. This MS-DOS batch program is useful for listing the characteristics of an image file, such as the number of pixels/scanline, number of scanlines, number of bands, etc. These characteristics are stored as variables in the 'REMAPP' header file control block (FCB) found in the image HEADER file (**imagename\HEADER**). If the file control block is not correct, individual FCB's may be changed and the new values written to the 'REMAPP' header using any ASCII text editor. For FCB description, see U.S. Geological Survey Open-File Report 85-231 (Sawatzky, 1985).

Note - You must include the image name as the first argument on the command line

To run, start program (include image name):

1) **remhdr filename**

example:

remhdr cck.b3s

Results:

```
Image ProcSys:  REMAPP
Parameter:           0
No. pixels:        512
No. scanlines:     400
First pixel:         1
First scanline:      1
Pixel skip:          0
Scanline skip:       0
File name:      cck.b3s
reclen in bytes:   512
no. of bands:        1
current rec no.:   401
word no. 15          0
word no. 16          0
word no. 17          0
band 1st line:       1
band increment:      1
volume name
word no. 21          1
word no. 22          0
no. pixels/line:   512
no. recs/file:     400
bits/pixel:          8
no.last scl/win:   400
no.last pxl/win:   512
word no. 28:         1
word no. 29:         0
word no. 30:         0
```

REMLST

REMLST converts 'REMAPP' image files from binary form to ASCII form and sends the output to the screen, printer, or disk file. This program is useful for looking at the DN values of an image, and subsetting to examine image spectra of BIL files. Noise patterns in the scanline or pixel directions of a band can also be studied.

To run, start program:

1) enter CON — to select ASCII output to be displayed to the screen (console)
   enter PRN — to select ASCII output to be written to the default printer
   or

   enter a filename — to select ASCII output to be written to the default (current) subdirectory (any valid DOS filename may be used)

2) answer DISKIO for image name, and scanline and pixel subsets.

The start of the ASCII data is the first pixel of the first scanline selected in the 'REMAPP' subset. When displaying data to the screen, the <pause> key allows data to be examined, rather than just scrolling by.

REMROT

REMROT rotates 'REMAPP' images 90 degrees clockwise. If an image needs to be rotated 180 degrees, just run this program on the data twice.

To run, start program:

1) answer DISKIO for the input image filename, and scanline and pixel subsets.
2) enter output rotated image filename.

SATSTR

SATSTR performs masking operations which blend an input image and a DN value of zero through control by a second (mask) image. Masking copies pixels from the saturation (overlay) image to the output image when the same pixels on the hue (mask) image fall within the specified hue (DN masking) interval, else pixels with a DN value of zero (background) are used. SATSTR uses for input, a hue (mask) image, a saturation (overlay) image, a DN value of zero (background), and a hue interval (DN masking interval) defined by the user. Up to 10 hue (DN masking) intervals may be defined and applied.

Note: Images must be registered to each other for masking operations.

To run, start program:

1) answer DISKIO for the input hue (mask) image filename, scanline and pixel subsets.
2) answer DISKIO for the input standard (saturation) filename, scanline and pixel subsets.
3) enter output (classified saturation) image filename
4) enter the number of hue (DN masking) intervals (an integer number from 1 to 10)
5) enter the hue (DN masking) interval
   EXAMPLE: 25 40
   (the first integer must be less than the second)

Repeat step 5 until all hue (DN masking) intervals are specified as defined in step 4. Later hue intervals may be greater, lower, or overlap previously specified hue intervals.

This program was originally designed to retain selected saturation image pixels that contained a pixel hue within the designated ranges, and set the remaining parts of the saturation image to black.

SHADRAT

SHADRAT performs masking operations which blend two input images using the first input image also as the mask control. The band (overlay and mask) image controls the masking operation. Pixels are copied from the band (overlay and mask) image when DNs for these same pixels fall within the specified DN masking interval, else pixels from the ratio (background) image are used.

For example, shadow/water pixels are determined visually on a single band image (by their dark DN values); their DN range is selected. SHADRAT substitutes a pixel from the band (overlay and mask) image into the output file if the band image pixel falls within the selected DN interval, otherwise the ratio (background) image pixel is transferred to the output file.

Note: Images must be registered to each other for masking operations.

To run, start program:

1) answer DISKIO for the input band (overlay and mask) image filename, and scanline and pixel subsets
2) answer DISKIO for the input ratio (background) image filename, and scanline and pixel subsets.
3) enter output filename.
4) enter DN range interval (specified DN masking interval)

EXAMPLE: 0 45

When the pixel DN of the **band** image is between 0 and 45, that pixel is used in the output file. If the pixel DN is outside this interval (> 45), the **ratio** image pixel will be used in the output file.

SPECAVG

SPECAVG averages up to 32 ASCII band interleaved by line (BIL) spectral calibration files (i.e. calibration sites-files generated from CALFILE) into one output calibration file. Input and output files are composed of 2 columns of ASCII data, the first being the multiplication real number constant column and the second being the additive real number constant column. Band one calibration constants form the first line of the file, while the last band forms the last line of the file. Input file real numbers may be stored in any format. Output file real numbers are stored in scientific notation (FORTRAN E format).

To run, start program:

1) enter how many bands (channels) are contained in the input calibration file (each input file must contain the same number of bands).
2) enter ASCII input filename
3) respond to prompt for next ASCII input file ('Y' or 'N') a 'Y' response cycles back to step 2 for reading another input file; 32 input files maximum.
4) enter ASCII calibration output filename

NOTE: the output file may be applied by the program BILCAL to a raw imaging spectrometry BIL file to generate a calibrated imaging spectrometry BIL file.

SPECLST

SPECLST creates individual ASCII spectra files, with corresponding band (channel) information, from BIL 'REMAPP' images. The output is useful for immediate graphing using various commercial graphing programs. The DN (reflectance) band values of the selected pixel are matched with channel information, starting arbitrarily with band number 1 (i.e.: the first band read is labelled channel 1, the second is 2, etc. up to the total number of bands used within the pixel subset). Spectral band interval subsets may start and stop anywhere in the spectra, if only part of the spectra is of interest (see Appendix, DISKIO- User interface).

To run, start program:

1) answer DISKIO for BIL image name, and scanline and pixel subsets.
2) enter an output filename (for ASCII output)

Example:
Write a spectra for scanline 105, pixel 173, of a TM 7 band BIL image file, 400 scanlines/band (2800 records total in file) and 640 pixels/scanline. All bands will be used (band 1 - 7).

```
ANYWHERE.BIL    BIL image filename
729 7 0      729 = (scanline, channel 1) * bands per file + starting band
             7 bands (channels) to read          ((105-1)*7)+1
             0 skips
173 1 0      173 = starting pixel 173
             1 pixel to read
             0 skips
```

OUTPUT: (for example)

```
ANYWHERE.ASC     (filename)
1 56
2 107
3 89
4 95
5 160
6 198
7 136
```

STRETCH

STRETCH applies singular to multiple linear contrast stretches to an image file. DN values within the original image are proportionally set to new DN values within the output image. Various DN intervals of an image file may be shifted to new slopes and lengths. This program is very useful for bilinear stretches (i.e. 2%, 50%, 98% stretch for example) to contrast stretch images for visual display. **STRETCH** inputs 8 or 16-bit data and outputs 8-bit data.

To run, start program:

1) answer DISKIO for input image filename, and scanline and pixel subsets.
2) enter output (stretched) image filename
3) enter the number of linear stretch sets (mapping pairs)
   i.e.: 3 (two linear stretch intervals require 3 sets of pairs)

   NOTE: a 0 prompts the user for an ASCII text file containing the mapping pairs and step 4 below is skipped. Create this ASCII file with any word processor that saves text in ASCII (word only or unformatted) format.

4) enter the linear stretch pairs (mapping pairs), one pair per line, with the input image DN values (first column of the mapping pair) increasing in value with subsequent mapping pairs.
5) enter 'Y' (Yes) or 'N' (No) to continue stretching further images or to stop.

Example 1:
Two percent bi-linear stretch (set the two linear stretches and truncate any values outside the stretch range)

```
        step 3)          3
        step 4)          21 0
                         46 127
                         87 255
```

Result: data is stretched in two linear intervals

	Original Image		Stretched Image
DNs less than	21	are set to	0
DNs from	21 to 46	are proportionally set to	0 to 127
DNs from	46 to 87	are proportionally set to	127 to 255
DNs greater than	87	are set to	255

DNs of 0 to 21 comprise the lowest 2% (lower tail) of the original image and are set to 0 in the new image. DNs of 87 to 255 comprise the highest 2% (upper tail) of the original data and are set to 255 in the new image. The 50% DN level (data median) is 46 within the original image and is set to 127 (or 255/2) in the new image. DN percentages are calculated in HIST.

Example 2:

Four part linear stretch with mapping pairs called up from an ASCII text file (set the four linear stretches and no DN tail truncation)

```
        step 3)          0
        step 4)          CANON.ASC
```

Where CANON.ASC is:

```
0 0
21 50      NOTE:  Only mapping pairs are contained within text file.
40 60             Input image DNs [column 1] always increase downward.
80 14          (The third linear stretch demonstrates a negative slope)
255 255
```

Result: data is stretched in four linear intervals

	Original Image		Stretched Image	
DNs from	0 to 21	are proportionally set to	0 to 50	
DNs from	21 to 40	are proportionally set to	50 to 60	
DNs from	40 to 80	are proportionally set to	60 to 14	(negative slope)
DNs from	80 to 255	are proportionally set to	14 to 255	

Example 3:

Convert 16-bit data to 8-bit, with no change in pixel DN values, assuming all DNs are less than or equal to 255 (8-bit value maximum).

```
        step 3)          2

        step 4)          0    0
                         255  255
```

Result: data are converted to 8-bits per pixel format with no change in DN values.

STRETC16

STRETC16 is the same as STRETCH, but outputs 16-bit data. Image input may be either 8-bits or 16-bits per pixel.

NOTE: Using Example 3 in STRETCH with STRETC16 will convert 8-bits per pixel images to 16-bits per pixel format.

STRIPHDR

STRIPHDR strips any number of bytes from the beginning of a binary image file, then creates a REMAPP-PC format image. An image subdirectory is created, an ASCII header file is generated, the number of specified bytes is skipped from the beginning of an image file, and then the rest of the image file is copied into the new 'REMAPP' image data file. This program is useful for converting an image from another image processing system which uses embedded header records, into this system's format.

To run, start program:

1) answer prompt for input image filename

2) enter the number of pixels per scanline

3) enter the number of bits per pixel (either 8 or 16 bits/pixel)

4) enter the number of bytes to skip from the beginning of the file

5) enter the number of bands in the file

6) answer DISKIO for the output image filename

Note: compare the scanline and pixel sizes for proper dimensions

SUBBAND

SUBBAND subtracts individual bands (channels) from a band interleaved by line (BIL) input file. The output BIL file will be copied without the user defined band or bands. The total number of bands in the output file will be the number of bands in the input file minus the number of subtracted bands.

To run, start program:

1) answer DISKIO for image filename, and scanline and pixel subsets.

   NOTE: you must use -1 for band in scanline subset with BIL files

2) enter the input band to be deleted (not copied to the output file)

   NOTE: step 1 loops for more input until '0' is entered

3) enter output filename

TMCAL

TMCAL generates band multiplication and additive constants to calibrate Landsat TM images to absolute (at-satellite planetary) reflectance, with less than 10% variation. These constants are applied using ARITH.

The equations and look up tables were derived from Markham and Barker, 1986 (MSS calibration is also covered in this report).

$$Lw = LMINw + \left(\frac{LMAXw - LMINw}{QCALMAX}\right) * QCAL \qquad \text{(equation 1)}$$

where:

QCAL = the pixel DN value
QCALMAX = Max DN for system (255 for all TM, 127 for all MSS, except, 63 for early to mid MSS band 4)
Lw = Spectral radiance (milliwatts per square centimeter per steradian per micrometer)
LMINw = Spectral radiance at QCAL = 0 (DN = 0)
LMAXw = Spectral radiance at QCAL = QCALMAX (at DN = 255, 127, or 63)

and

$$Rp = \frac{pi * Lw * dsqr}{ESUNw * \cos(sza)} \qquad \text{(equation 2)}$$

where:

Rp = Unitless effective at-satellite planetary reflectance
pi = 3.14159
Lw = Spectral radiance at sensor aperture (equation 1 above)
dsqr = the square of the Earth-Sun distance in astronomical units (1 AU squared)
ESUNw = Mean solar exoatmospheric irradiances
sza = Solar zenith angle in degrees (note, program uses solar elevation angle above the horizon i.e.: 1 - sza)

To run, start program:

1) select scene date option: (press number <1>, <2>, or <3>)
   <1> Prior to August 1983
   <2> Prior to January 15, 1984 (but after 8/83)
   <3> After January 15, 1984
2) select whether the data was acquired by Landsat 4 or by Landsat 5 (press number <4> or <5>).
3) enter the solar elevation angle (sun angle above the horizon) in decimal degrees (example 33, or 33.85, or 33.716667)
4) copy reflectance calibration factors (constants) for application into ARITH (see ARITH). Alpha is the multiplication constant, beta is the additive constant. NOTE: the <Print Screen> key will dump the screen to the default printer

Result (something like below):

T H E M A T I C   M A P P E R

SATELLITE = LANDSAT 5
SOLAR ELEVATION = 54
ACQUIRED AFTER 15 JANUARY 1984

REFLECTANCE CALIBRATION FACTORS

BAND	**ALPHA**	**BETA**
1	1.1952	-2.9764
2	2.4945	-5.9448
3	2.0099	-2.9928
4	3.0209	-5.5633
5	1.9138	-6.5517
7	2.9692	-7.8165

TRISTM

TRISTM converts images in Munsell hue, saturation, and value color coordinates (tristimulus values), to red, green, and blue color coordinates (Munsell tristimulus values are inverted back to red, green, and blue color coordinates). Input is 8 or 16 bits/pixel, output is 8-bits/pixel. Also, see MUNSEL.

Output DNs:
Red color DN range: 0 - 255
Green color DN range: 0 - 255
Blue color DN range: 0 - 255

To run, start program:

1) answer DISKIO for the hue input image filename, and scanline and pixel subsets.

2) answer DISKIO for the saturation input image filename, and scanline and pixel subsets.

3) answer DISKIO for the value (intensity) input image filename, and scanline and pixel subsets.

4) enter a filename for the output red image file.

5) enter a filename for the output green image file.

6) enter a filename for the output blue image file.

TRISTM16

TRISTM16 is the same as TRISTM, but with 16-bit output. Also, see MUNSEL.

References

Livo, K. E., 1990, REMAPP-PC remote sensing processing software for MS - DOS personal computers, ver. 1.00: U.S. Geological Survey Open-File Report 90-88, 58 p., 4 - 5 1/4 inch H.D. diskettes.

Markham, B. L., and Barker, J. L., 1986, Landsat MSS and TM post-calibration dynamic ranges, exoatmospheric reflectances and at satellite temperatures: EOSAT Landsat Data User Notes, No.1, Aug. 1986, p. 3-8.

Raines, G. L., and Knepper, D. H., Jr., 1983, A hue-saturation-intensity transform to improve hydrothermal alteration mapping: IEEE Digest vol. II, Remote Sensing: Extending Man's Horizon, 1983 International Geoscience and Remote Sensing Symposium, Aug. 31 - Sept. 2, 1983, San Francisco, CA, p. 1.1 - 1.3.

Sawatzky, D. L., 1985, Programmer's Guide to REMAPP, REMote sensing Array Processing Procedures: U.S. Geological Survey Open-File Report 85-231, 21 p.

Silva, Dan, 1988, Deluxe Paint II Manual with program disk: Electronic Arts, 1820 Gateway Drive, San Mateo, CA 94404.

Appendix A

Suggested Processing Flow Charts

(images assumed to contain 'REMAPP' header)
(see ADDHDR, STRIPHDR)

1) Image Display
(raw images may be displayed as is, use <V> subroutine within DISPLAY for an automatic contrast stretch)

2) Image Display with hard contrast stretched file

HIST
STRETCH
(repeat above three times for color - Red, Green, & Blue bands)
DISPLAY

3) Vegetation Selection
(select vegetation using a hue range which is generated by the intense leaf structure reflections in the near IR)

MUNSEL (TM bands 4, 3, 2; MSS bands 7, 5, 4; RGB)
STRETCH (reduce 16 bit hue file to 8 bit with range of 0 - 180 from original 16 bit range of 0 - 360)
DISPLAY (select vegetation range of about 105 - 165 in hue file)
Mask with MASK, SHADRAT, PIXELSUB, HUEMAP, or SATSTR
(mask vegetation hue interval (105 - 165) as black or white on a single-band image, or as a color or gray on a color image)

4) Color Saturation Enhancement

MUNSEL
HIST (saturation band only)
STRETCH (saturation band only)
TRISTM

4) Mineral Alteration Map

TMCAL & ARITH (for TM data) calibrate TM band images (generate coefficients & apply)
ARITH (3 ratios: TM 3/4, 3/1, 5/7; MSS 4/5, 6/7, 4/6; RGB)
DISPLAY (measure DNs of shadows on a single-band gray image)
Mask with MASK, SHADRAT, PIXELSUB, HUEMAP, or SATSTR
(mask vegetation, shadow covered areas, and water)
Further processing can include hue, saturation, and value (intensity) masking intervals with MUNSEL transformed files.

5) Calibrate Image Spectroscopy File

CALFILE (generate site calibration ASCII files)
SPECAVG (combine site calibration ASCII files into one calibration ASCII file)
BILCAL (apply ASCII calibration file to raw image spectrometry file to generate calibrated image spectrometry output file)

Appendix B

DISKIO - User Interface

Introduction:

Successful operation of image processing is impossible within REMAPP without understanding the user interface. DISKIO (disk input/output) is the common user interface, used in the majority of REMAPP programs for user interaction with the imagery. The DISKIO subroutine prompts the user to define for the current program what image file or files to use, and how the program will use the image data. By enabling the user to describe what image, which part of the image to use, and how to use it, such as by subsampling and/or windowing, much greater latitude is given for accessing and combining image data.

An example query with DISKIO for an input image is shown below:

```
Diskio: enter input filename
can.b3
Diskio(2):can.b3       [lines/band   400 pixels   640 bytesize  8 bands  1]

enter 1st scanline, no. lines/band, skips, band:
0 0 0 0
enter 1st pixel, no. pixels, skips:
0 0 0
```

A DISKIO query for an output image prompts only for an image filename, as all other parameters are internally calculated when writing the output file.

As can be seen, the user is first prompted for an image name, in this case, can.b3. The original image dimensions are displayed once the image name is entered and a description of the image data is read from the REMAPP header. Here it shows that the selected image file, can.b3, contains 400 scanlines or records, with 640 pixels per scanline, that each pixel DN is 8 bits/pixel, and the file is a single band image (1 band). Next is the prompt for the scanline subset. In this case, the scanline subset includes all the scanlines within the file (defaults of 0 mean use every scanline). Likewise, the user is prompted for the pixel subset, in this example all pixels per scanline are used. This example uses the whole input file, as all scanlines and pixels are used with no subsampling.

The image name prompted by DISKIO must be a legal MS-DOS name of one to eight characters with an optional extension of up to three characters. Full or partial paths and/or drives may be prefixed to the image name. The maximum name length with all prefixes is truncated to the first 64 characters. MS-DOS filenames are not case sensitive.

The scanline and pixel subset information control which part of the image will be used by the current program. The subsets define the window or position within the full image where the data will be extracted by the program, such as the starting upper left corner of the subset window (the 1st scanline and 1st pixel to use), the ending lower right corner of the window (the starting position + number of scanlines and pixels used), and how much of the image subset will be used by subsampling (skipping scanlines and pixels). This combination allows the user to use all of the image, or extract just a part through windowing the data at full resolution (no skips), subsampling the whole image using skipped scanlines and pixels, or combining windowing and subsampling as a window with skipped scanlines and pixels.

The scanline subset also controls how the data is used when it is in a band interleaved by line (BIL) format. This format is less common than the single band image format, but is used extensively in spectroscopy. Band interleaved by line files group several spectral bands or records of data per image scanline. Single band data may be extracted from this type of file using the band argument, where band 1 = the first band in the file, band 2 = the second, and so forth. The band argument of -1 defines the usage of all bands of the BIL file within the processing program. The default band argument of 0 selects the extraction of band 1 in either a BIL or single band image file.

The scanline subset prompts for the four arguments are:

enter 1st scanline, no. lines/band, skips, band:

Argument 1 - 1st scanline (starting scanline)

DISKIO is prompting for where in the image the first scanline of the subset is to be read (starting scanline). The starting location of the scanline window may be anywhere inclusive between scanline 1 and the last scanline of the image. Scanline 1 is the top scanline within an image. The last scanline is the bottom scanline within an image. Argument values range from 1 to the last scanline number in the image. The default value of 0 is converted automatically to 1. A special case is when BIL files are used. The starting scanline for extracting a single band sub-image from within the BIL file is as above. No record numbers need to be calculated. If a BIL sub-image is being windowed from a BIL image, the 1st scanline is assigned as the file record within the BIL image containing band 1 of the starting scanline of the BIL sub-image. This record is calculated as the starting scanline minus 1, times the number of bands in the file, plus 1 record.

rec = (1st sl - 1) * # bands + 1   (BIL to BIL only!)

BIL to BIL addressing treats the original input BIL file as a single band image, so any band of any scanline within the input file must be calculated in absolute records.

Argument 2 - no. lines/band

DISKIO is prompting for how many scanlines to use, at and following the starting scanline (argument 1). This will define how many scanlines will be used by the program. All scanlines are counted within the area of the scanline window for calculating the image scanline height, even if skips are used. Argument values range from 1 to the number of scanlines from the starting scanline to the end of the file. The default value of 0 is converted automatically to the last scanline within the image. There is a special case when BIL files are used. When extracting a single band sub-image from a BIL image window, the number of image scanlines is used as above. When extracting a BIL sub-image from a BIL image window, the number of file records within the scanline subset is used. The number of records is calculated as the number of image scanlines used times the number of bands within the BIL file.

recs = # scanlines * # bands

Argument 3 - skips

DISKIO is prompting for how many scanlines will be dropped or skipped for subsetting the data. When skips are used, the starting scanline is read and used, the following number of scanlines are skipped (jumped), then the next scanline is read and used, and so forth, to the last scanline defined within the scanline window. Values start from 0 or no skips. When extracting a BIL sub-image from a BIL image, multiply the number of skips used for skipping scanlines by the number of bands within the file minus 1, to get the number of records needed to skip.

skips = # skipped scanlines * # bands -1   (BIL to BIL only!)

Argument 4 - band
DISKIO is prompting for which band or bands to use within the current program. This argument is normally used with band interleaved by line (BIL) image files. Single band data may be extracted from BIL files using the band argument, where band 1 = the first band in the file, band 2 = the second, and so forth. The band argument of -1 defines the usage of all bands of the BIL file within the processing program. The default band argument of 0 selects the extraction of band 1 in either a BIL or single band image file.

The pixel subset prompts for the three arguments are:

enter 1st pixel, no. pixels, skips:

Argument 1 - 1st pixel
DISKIO is prompting for where in the image the first pixel of the subset is to be read (starting pixel). The starting location of the pixel window may be anywhere inclusive between pixel 1 and the last pixel of the scanline. Pixel 1 is the left edge pixel within a scanline. The last pixel is the right edge pixel within a scanline. Argument values range from 1 to the last pixel number in the scanline. The default value of 0 is converted automatically to 1.

Argument 2 - no. pixels
DISKIO is prompting for how many pixels to use, at and following the starting pixel (argument 1). This will define how many pixels will be used by the program. All pixels are counted within the area of the pixel window for calculating the image scanline width, even if skips are used. Argument values range from 1 to the number of pixels from the starting pixel to the end of the scanline. The default value of 0 is converted automatically to the last pixel within the scanline.

Argument 3 skips
DISKIO is prompting for how many pixels will be dropped or skipped for subsetting the data. When skips are used, the starting pixel is read and used, the following number of pixels are skipped, then the next pixel is read and used, and so forth, to the last pixel defined within the pixel window. Values start from 0 or no skips.

Programming:

The user interface for most of the programs is the subroutine DISKIO. The various programs call DISKIO through function call statements. Open, read, write, scanline positioning, image windowing and subsetting, and close file functions are performed transparently to the user for a consistent user interface and compatible data structure. Programming with DISKIO is detailed in the USGS Open-File Report 85-231 (Sawatzky, 1985). Source code in the present image processing system may also be studied.

DISKIO object code within this package may be linked to user programs using the Large Memory model of FORTRAN. Program source code may be in either MS-FORTRAN ver. 5.0, MS-C ver. 6.0, or MS-QuickC ver. 2.0. C code must declare the DISKIO subroutine as PASCAL or FORTRAN.

DISKIO Examples:

Select image (single-band or BIL) with default values and input filename ANYWHERE.IMG. The image starts with scanline 1, reads all scanlines (and bands if BIL file), and does not skip past any scanline. The image starts with the first pixel, reads all pixels in the scanline, and does not skip past any pixels - i.e.: read entire file

```
Diskio: enter input filename
ANYWHERE.IMG
Diskio(2):ANYWHERE.IMG [lines/band   200 pixels   320 bytesize  8 bands  1]

enter 1st scanline, no. lines/band, skips, band:
0 0 0 -1
enter 1st pixel, no. pixels, skips:
0 0 0
```

Window a single-band image, from a larger single band image, starting at scanline 100, read 200 scanlines (end at scanline 299), no scanline skips, and starting at pixel 25, read 320 pixels (end at pixel 324), no pixel skips. Output size of the image is 200 scanlines by 320 pixels. Note - the output file will be truncated if the last scanline or last pixel per scanline is reached in the input file before the end of the entered window is reached, in which case, the truncated file will be smaller than expected, but is still useable. Here we can use the band default of 0 (or -1 or 1) because we are dealing with a single band input image.

```
ANYWHERE.IMG
100 200 0 0
25  320 0
```

Window a single-band image from a larger single-band image, starting at scanline 50, read 800 scanlines (end at scanline 849), with 3 skips (every fourth scanline is used), start at pixel 75, read 1280 pixels (end at pixel 1354), with 3 skips (every fourth pixel is used). Output image size is 200 scanlines by 320 pixels (because of the skips, i.e.: 200 scanlines * [3 skips + 1] and 320 pixels * [3 skips + 1]).

```
ANYWHERE.IMG
50  800 3 0
75 1280 3
```

Window out band 2 (a single-band image) from a 4 band BIL (band interleaved by scanline) image file, using all scanlines and pixels.

```
ANYWHERE.B2
0 0 0 2
0 0 0
```

Window out band 2 (a single-band) from a large 7 band BIL image file, starting at image-scanline 100 of band 2, wanting 200 scanlines, starting at pixel 250, and reading 320 pixels (ending pixel 569) with no skips. Output is band 2, 200 scanlines by 320 pixels.

```
ANYWHERE.B2
100 200 0 2
250  320 0
```

Subset out a 10 band BIL file from a larger 10 band BIL image file, with coordinates starting at scanline 250, reading 300 scanlines per band with no skips (read all bands), starting pixel 220, reading 512 pixels, no skips. Output is 10 bands, 300 scanlines, 512 pixels per band (or 3,000 records x 512 pixels).

```
  rec = ( 1st sl - 1 ) * # bands + 1    (BIL to BIL only!)
 2491 = (    250 - 1 ) * 10 + 1
```

```
ANYWHERE.BIL
2491 3000 0 -1
 220  512 0
```

Window a pixel spectra from a 61 band BIL image file, with coordinates at scanline 150, pixel 25, all 61 bands. (starting file record = 9090)

```
ANYWHERE.BIL
9090 61 0 -1
  25  1 0
```

Window a pixel spectra from a 61 band BIL image file, with a band interval from band 6 to band 41, and coordinates at scanline 150, pixel 25.

```
ANYWHERE.641
9095 36 0
  25 1 0
```

Appendix C

File Formats

'REMAPP' files are binary single-band or band interleaved by line (BIL) imagery (Microsoft FORTRAN BINARY format or byte stream C format). Data is simply a byte stream with no blocking or formatting bytes and may be 8 or 16-bits per pixel. Eight bits/pixel data use one byte per pixel, for a numeric range of 0 to 255 (all positive DNs). Sixteen bits/pixel uses two bytes per pixel, for a numeric range of 0 to 32,767 (do not use the negative number component). Eight bits equal one byte; one byte contains a range of 256 numbers, which is also 2 to the 8th power - 8 bits. Eight and 16-bit image data for MS-DOS and VAX computers store individual bytes of data in identical order (FORTRAN I*2).

A 'REMAPP' image is composed of a subdirectory, which is the image name, and within this subdirectory are two files named HEADER and DATA. The 'REMAPP' HEADER file is an ASCII file which describes the DATA raster file to the REMAPP system. This HEADER file, transparent to the user, is read by the subroutine DISKIO, and contains an array of file control blocks (FCB's) which describe the characteristics of the file, such as the size, name, data format and type, number of bands, etc. The FCB is in the form of a four-byte binary integer array (I*4). REMHDR displays the FCB of any 'REMAPP' image. Binary raster image data are stored in the DATA file. This data is in either 8 or 16 bits/pixel integer format, one scanline per record (single band data) or one scanline band per record (BIL data).

Single-band image data is stored as a single file, with each scanline being a record within the file. Scanlines progress downward through the scene, from top to bottom. They are stored similarly in the file, with consecutive scanline records from one (top) to the last record (bottom) in the file. Pixels in each scanline are stored likewise, with the first pixel in the scanline being the first one (8-bit) or two (16-bit) bytes in the record, continuing to the last pixel being stored as the last one or two bytes in the record.

Band interleaved by line images (BIL) are stored as a single file. They are stored similarly to single-band imagery, except that every scanline band set is stored such that each band follows one another for that particular image-scanline, then the next scanline band set follows the previous scanline band set.

```
image:    anywhere.img   (single band data)

(image size - say 600 pixels x 300 scanlines x 1 band, 16-bits/pixel)

file record 1:   scanline  1, band 1 (first data record)
 "     "   2:       "      2,  "   1 (second record)
 "     "   3:       "      3,  "   1 (third record)
                 .
                 .
                 .
 "     "  299:      "    299,  "   1
 "     "  300:      "    300,  "   1 (last record)

End of File

for a file size of 600 pixels x 300 records x 2 bytes/pixel = 360,000 bytes
```

```
image:    anywhere.bil    (7 band TM data - BIL)

(image size - say 320 pixels x 200 scanlines x 7 bands, 8-bits/pixel)

file record 1:   scanline   1, band 1 (first data record)
 "      "   2:      "       1,   "  2  (second record)
 "      "   3:      "       1,   "  3  (third record)
 "      "   4:      "       1,   "  4          .
 "      "   5:      "       1,   "  5          .
 "      "   6:      "       1,   "  6          .
 "      "   7:      "       1,   "  7
 "      "   8:      "       2,   "  1
 "      "   9:      "       2,   "  2
 "      "  10:      "       2,   "  3
 "      "  11:      "       2,   "  4
 "      "  12:      "       2,   "  5
 "      "  13:      "       2,   "  6
 "      "  14:      "       2,   "  7
 "      "  15:      "       3,   "  1
 "      "  16:      "       3,   "  2
                .
                .
                .
 "      " 1392:     "     199,   "  6
 "      " 1393:     "     199,   "  7
 "      " 1394:     "     200,   "  1
 "      " 1395:     "     200,   "  2
 "      " 1396:     "     200,   "  3
 "      " 1397:     "     200,   "  4
 "      " 1398:     "     200,   "  5
 "      " 1399:     "     200,   "  6
 "      " 1400:     "     200,   "  7  (last record)

End of File

for a file size of 320 pixels x 1400 records x 1 byte/pixel = 448,000 bytes
```

Appendix D
Compilation and Linkage Instructions

FORTRAN

All FORTRAN programs were compiled and linked with Microsoft FORTRAN version 5.00, large memory model, and math emulation. The large memory FORTRAN library module (LLIBFORE.LIB) is used with the math emulator package and without C graphics compatibility (except for DISKIONO.OBJ).

Eight different image processing subroutines exist as object files and are linked as needed into the program being created. They are:

DISKIO.OBJ	Directly interfaces 'REMAPP' programs and system memory with disk files
DISKIONO.OBJ	Modified version of DISKIO.FOR which does not log program operations into REMAPP.LOG, is modified for 40 column operation, and is compiled with C compatibility for linking with DISPLAY.C
PACK.OBJ	Packs 8 bit data for storage on disk. I*2 - 8 bits/pixel scanlines are packed as single bytes per pixel for storage in disk file (pixels are usually operated on as 2 byte words (16 bits) within system memory, whether the pixel is 8 or 16 bits/pixel data)
UNPACK.OBJ	Unpacks 8 bits/pixel - single byte pixel scanlines from disk file format into I*2 (16 bit) format for use in system memory
ILBYTE.OBJ	Returns the low byte of a two byte word

Object files below are appended within the DISKIO.OBJ file

REMLOG.OBJ	Logs program operation inputs into the file REMAPP.LOG
DATE.OBJ	Gets the CPU system clock date for REMLOG
TIME.OBJ	Gets the CPU system clock time for REMLOG
EXIT.OBJ	Error handling subroutine for 'REMAPP' programs

Compile and link:

Compile the 'REMAPP' program, and link with the Microsoft FORTRAN library LLIBFORE.LIB and any of the object files needed above.

EXAMPLE:

Compile and link ARITH

```
FL /FPc ARITH.FOR DISKIO UNPACK /link /E
```

NOTE: the program being compiled must have the .FOR extension. Separate linking object files with spaces. Have the object files in the same directory as the file being compiled and linked. The list of object files needed to be linked with the compiling program are listed near the top of each program's source code.

Refer to U.S. Geological Survey Open-File Report 85-231 (Sawatzky, 1985) for a guide to programming with 'REMAPP' and DISKIO. Note, REMAPP-PC uses I*2 (16 bit) pixels internally, while O.F. Report 85-231 describes an earlier I*4 (32 bit) pixel internal format.

QuickC

All QuickC programs are compiled with Microsoft QuickC version 2.0, using the large memory model and math emulation (LLIBCE.LIB). Compile the program within the QuickC environment using Make - Compile File. Link the program with the FORTRAN DISKIONO.OBJ subroutine by copying the program object file into the subdirectory containing the DISKIONO.OBJ file and starting the FORTRAN Linker. i.e.:

```
FL /FPc DISPLAY DISKIONO /link /E /NOE
```

NOTE: C FCB arrays start at 0, while FORTRAN FCB arrays start at 1 (with arrays passed by address)

Appendix E
Image Conversion from REMAPP-PC ver. 1.0

REMAPP-PC version 2.0 has a slightly different file format from REMAPP-PC version 1.0. These differences and how to convert version 1.0 images to version 2.0 file format are briefly explained here. For a more in-depth explanation of version 2.0 file formats, refer to Appendix C - File Formats.

REMAPP-PC versions 1.0 and 2.0 file formats have three major parts, an image name, a header image description, and the binary data. Version 2.0 places these three parts into separate units, while REMAPP-PC version 1.0 combines all three parts into one unit.

REMAPP-PC version 2.0 uses the subdirectory properties of MS-DOS to form a REMAPP-PC image. The image name becomes the subdirectory name, with the header image description as an ASCII text file named HEADER within this subdirectory. The binary data likewise is placed within this subdirectory as a file named DATA. The data file is solely an array of raw unformatted binary image data (8 or 16 bits/pixel) with no embedded ASCII information. Scanline one is the first record within the binary data file.

example: the image CANON.B3

```
CANON.B3\
         HEADER
         DATA
```

REMAPP-PC version 2.0 images should be used from either the subdirectory holding several image file subdirectories, or using full path names. The HEADER and DATA files are not used within image names (DISKIO takes care of appending \HEADER and \DATA where needed).

example: contrast stretch the image CANON.B3 held with several others in the directory C:\REMAPP\IMAGES.

```
stretch:  'C:\REMAPP\IMAGES\CANON.B3'
or        'CANON.B3'
```

(when the present working directory is C:\REMAPP\IMAGES)

REMAPP-PC version 1.0 combines all three major parts above within a single MS-DOS file. The image name becomes the filename, with the header image description as the first binary file record within this file. The image data follows this header record to form the body of the file, with scanline one (band one) being the second record of the file. Version 1.0 files were addressed like version 2.0 images above.

example: the image CANON.B3

```
stretch:  'C:\REMAPP\IMAGES\CANON.B3'
or        'CANON.B3'
```

To convert REMAPP-PC version 1.0 images to version 2.0, a subdirectory name is created for the version 2.0 image, the ASCII text HEADER file created, and the raw binary DATA file copied (minus the first binary header record of version 1.0). The easiest way for conversion is to use the STRIPHDR program and strip away the first record (header record). The number of bytes to strip are the number of pixels per scanline if 8 bit data, or two times the pixels per scanline if 16 bit data. STRIPHDR will then create an image subdirectory with HEADER and DATA files.

Note: use a disk cache program (such as smartdrv - found in the DOS 5.0 package) if the file to convert is large to increase copy speed and prevent disk thrashing.

Appendix F
About Supplied Images

Sample images are supplied in REMAPP format for use with the programs. They necessarily cover small areas in order to fit on only a few floppy disks for distribution. Much larger areas are routinely used. Various scanners and geographical areas are covered, demonstrating low to moderately high spatial and low to high spectral resolution; with various landforms, lithologies, and land use. Reflectance and emission (TM band 6) data show the variability response of surfical material with changes in wavelength.

Other raster data usable with this software have included geophysical - potential field data displayed spatially with changes in data intensity mapped as changes in brightness. SLAR, GER, and computer scanned paper and film products have also easily been inputted.

The images:

Four different scenes are supplied:

CANCITY - Canon City, Colorado
Landsat 1 (satellite) MSS Bands 4, 5, 6, & 7
400 scanlines x 512 pixels (8 bits/pixel)

80 meter resolution scene acquired in 1972. This raw scene, when compared with the geometrically rectified version (raw scene run through GEOMCORR), demonstrates the orbital skew and pixel over sampling of the MSS scanner. Band 4 is in the visible green wavelength, while band 5 is in the red, and bands 6 and 7 are in the near infra-red wavelengths.

CANON - Canon City, Colorado
Landsat 4 TM Bands 1, 2, 3, 4, 5, 6, & 7 (Path 33, Row 33)
400 scanlines x 640 pixels (8 bits/pixel)

Excellent scene, ground resolution 28.5 meters, visible blue to near infra-red wavelengths, 1984. Canon City, the Arkansas River, and irrigated fields (bright in band 4 IR) show well. The scene is centered on a southerly plunging syncline in Phanerozoic sedimentary rock with steeply dipping exposed limbs forming hog-backs. Cretaceous shales (syncline center) and sandstone (limbs), with Paleozoic hematite rich arkoses and dolomite caprock (upper right) cover a Precambrian schist and gneiss unit (exposed at the northeast and west). A southerly plunging anticline lies to the east of the syncline. Numerous faults and minor folds are also exposed. These structures form the southern end of the Southern Front Range Mountains, near the start of the Florence oil field (joint porosity in shales). Jurassic Morrison shales and sandstones on the north yielded dinosaur fossils.

CENTRAL - Central City, Colorado
Landsat 4 TM Bands 1, 2, 3, 4, 5, 6, & 7
275 scanlines x 320 pixels (8 bits/pixel)
(NOTE: limited dynamic range to data DNs, 1984 scrounge data)

Scene covers Central City - Black Hawk with North Clear Creek on the northeast and the main branch of Clear Creek with Idaho Springs and I-70 on the bottom edge. Numerous mines (epithermal gold), prospects, and tailings piles reflect brightly in Precambrian schists and gneisses. The Glory Hole open pit (large bright area with the open pit center in dark shadow) and oh-my-god-road (steep, with cliffs) lay between Central City and Idaho Springs. Gold was discovered by prospectors panning up Clear Creek/North Clear Creek to the gold's source in Russell Gulch (center of image) and surrounding region.

CRIPPLE.BIL - Cripple Creek, Colorado (Globe and Iron Clad Hills)
AVIRIS (NASA) Bands 1 - 208 (resampled from 224 channels)
53 scanlines x 53 pixels (16 bits/pixel)
(NOTE: data calibrated to absolute reflectance)

Scene covers Iron Clad and Globe Hill disseminated gold open pits with mine and mill waste dumps, dark areas are evergreens (Pikes Peak Mining). This imaging spectrometry data detects various hydrothermal clay and iron oxide mineralization, and was flown by NASA with a U-2 at 65,000 feet in the fall of 1989. Ground resolution is 20 meters while spectral resolution is 10 nanometers. Spectral wavelengths are from 0.4 microns to 2.4 microns (visible blue to near infra-red). The full scene (512 scanlines x 614 pixels x 224 bands) was acquired in approximately 40 seconds of flight and has a swath width of 11 kilometers.