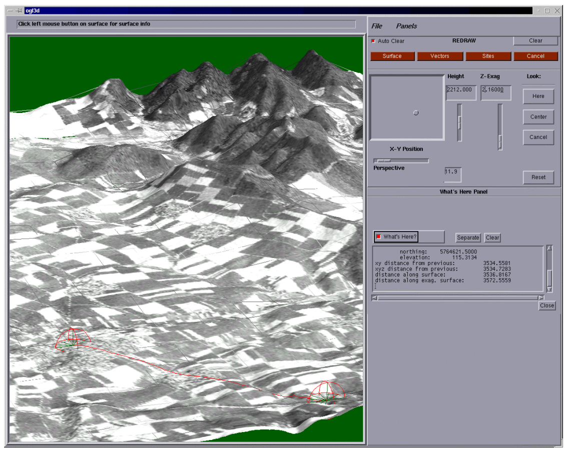
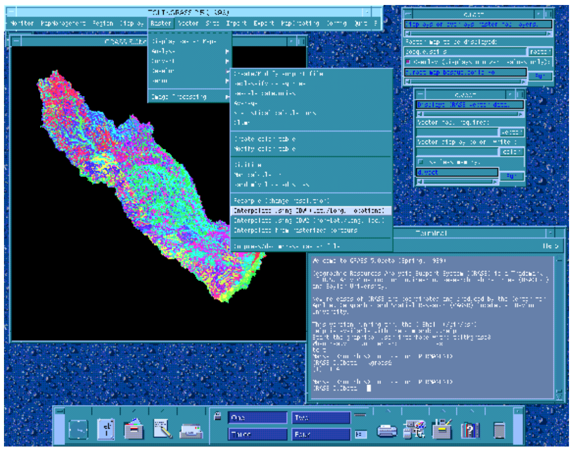
GRASS Reference Manual

Imagery Commands





GRASS Development Team

USA Headquarters
Center for Applied Geographic & Spatial Research
Baylor University
P.O. Box 97351
Waco, Texas 76798-7351
USA

European Headquarters
Institute of Physical Geography-Landscape Ecology
University of Hannover
Schneiderberg 50
30167 Hannover
Germany

grass@baylor.edu

http://www.baylor.edu/~grass
http://www.geog.uni-hannover.de/grass/

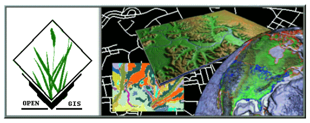


Table of Contents

GRASS Introduction

GRASS (Geographic Resources Analysis Support System) is a raster based GIS, vector GIS, image processing system, and graphics production system. GRASS contains over 200 programs and tools to render maps and images on monitor and paper; manipulate raster, vector, and sites data; process multi-spectral image data; and create, manage, and store spatial data. GRASS uses both an intuitive windows interface as well as command line syntax for ease of operations. GRASS can interface with commercial printers, plotters, digitizers, and databases to develop new data as well as manage existing data.

GRASS is ideal for use in engineering and land planning applications. Like other GIS packages, GRASS can display and manipulate vector data for roads, streams, boundaries, and other features. GRASS can also be used to keep maps updated with its integral digitizing functions. Another feature of GRASS is its ability to use raster, or cell, data. This is particularly important in spatial analysis and design. GRASS functions can convert between vector data to raster data for seamless integration.

GRASS' strengths lie in several fields. The simple user interface makes it an ideal platform for those learning about GIS for the first time. GRASS is capable of reading and writing maps and data to many popular commercial GIS packages including ARC/Info and Idrisi. Users wishing to write their own code can do so by examining existing source code, interfacing with the documented GIS libraries, and using the GRASS Programmers Manual. This allows more sophisticated functionality to be integrated in GRASS.

The ability to work with raster data gives GRASS the unique ability to function as a surface modeling system. GRASS contains more than 100 multi-function raster analysis and manipulation commands. Surface processes such as rainfall-runoff modeling, flowline construction (as shown), slope stability analysis, and spatial data analysis are just a few of the many applications of GRASS to engineering and land planning. Since many of the raster tools are multi-functional, users can create their own maps from GRASS data analysis.

In addition to standard two-dimensional analysis, GRASS allows users to view data in three-dimensions. Raster maps, vector maps, and sites data can be used for visualization. Example applications of such capabilities include airspace analysis for airport planning (as shown), terrain analysis and “flybys”, and spatial trends. Tools in GRASS allow the user to animate any spatial data available with options to switch between data layers “on-the-fly”. Data used in 3-D visualization may also be saved as still pictures, or as mpeg movie files for later replay and analysis.

Accompanying its land planning and engineering applications, GRASS contains a suite of tools to aid in hydrologic modeling and analysis. Currently, tools are also available for performing such functions as watershed analysis, curve number generation, flood analysis, and stream channel characteristics for comprehensive watershed modeling. Other GRASS programs can generate graphs, statistics, and charts of modeled and calibrated data. Additionally, GRASS can use field data for model input or simulate parameters based on numerical data.

In addition to the traditional command line version of GRASS, a new user interface, based on Tcl/Tk has been written. This puts the power of spatial analysis and modeling into an easy to use Graphical User Interface that is platform-independent. This intuitive user interface lets users quickly and easily view, manipulate, and use data. Nearly all of the programs available in GRASS are available in the new GUI, with the standard command-line still available, giving users all of the functionality of GRASS.

This manual is part of a comprehensive set of documentation written to support GRASS. This Users Guide consists of a complete set of command references for all current GRASS functions and tools, including examples. An installation guide and fact sheet guides users through the installation process. For those wishing to write their own spatial analysis and modeling applications for GRASS, a Programmers Guide is also available. GRASS runs on a variety of UNIX and Linux platforms including SUN SPARCstations and Ultras, HP, Silicon Graphics, and PC’s running Windows 95 and Windows NT.

The GRASS Development Team is currently working to further upgrade and enhance the capabilities of GRASS. Future developments include tools that give the user the ability to work completely in 3-D, a capability that does not exist in any other GIS package. Users will be able to work with raster elevation data as well as vector and sites data in the 3-D environment, adding to the visualization capabilities of GRASS. Enhancements in the numerical processing functions of GRASS also now allow for floating-point operations to be performed on data.

For the latest information on GRASS contact the GRASS Development Team at grass@baylor.edu or visit our web sites at:

http://www.baylor.edu/~grass if you're in the U.S.

http://www.geog.uni-hannover.de/grass if you're in Europe

Look for our worldwide mirrors!

The GRASS Development Team is:

Bruce Byars and Markus Neteler are the development team leaders and coordinators.

Helena Mitasova and Bill Brown of the GMS Lab at UIUC have made significant contributions with the development of GRASS 5.

Additional authors include:
Lisa Zygo, Edward Zarecky, Jacques Bouchard, Steve Clamons, Brent Duncan, Jason Cipriano, Jim Westervelt, Michael Shapiro, Darrell McCauley, Dave Gerdes, Bill Hughes, Bernhard Reiter, Brook Milligan, Eliot Cline, Jaro Hofierka, Clay Cockrell, and Bob Lozar. See the web pages for author affiliations.

Note:

Many other people have contributed to the GRASS GIS. Without any one of them, GRASS would not exist in its current form. The authors of the individual programs are listed at the end of their manual page in the GRASS users manual, however, numerous authors of bug fixes and enhancements as well as people who have been working on coordination, integration, documentation and testing are not mentioned.

Please allow us to extend our most cordial thanks to all of you. If you contributed to GRASS at any point during its existence, let us know your name and e-mail address so we can add your name to the comprehensive on-line list.

To reference GRASS:

GRASS Development Team, 1999, Geographic Resources Analysis and Support System - GRASS: Baylor University, Waco, Texas.

GRASS Development Team
Center for Applied Geographic and Spatial Research
Baylor University
P.O. Box 97351
Waco, Texas, U.S.A. 76798-7351

i.cca

NAME
i.cca **-** Canonical components analysis (cca) program for image processing.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.cca
i.cca help
i.cca group=name subgroup=name signature=name output=name

DESCRIPTION
i.cca is an image processing program that takes from two to eight (raster) band files and a signature file, and outputs the same number of raster band files transformed to provide maximum separability of the categories indicated by the signatures. This implementation of the canonical components transformation is based on the algorithm contained in the LAS image processing system.

Typically the user will use the *i.class* program to collect a set of signatures and then pass those signatures along with the raster band files to *i.cca*. The raster band file names are specified on the command line by giving the group and subgroup that were used to collect the signatures.

The output raster map names are built by appending a ".1", ".2", etc. to the output raster map name specified on the command line.

Parameters:

group=name	Name of the imagery group to which the 2 to 8 raster band files used belong.
subgroup=name	Name of the imagery subgroup to which the 2 to 8 raster band files used belong.
signature=name	Name of an ASCII file containing spectral signatures.
output=name	Output raster file prefix name. The output raster map layer names are built by appending a ".1", ".2", etc. onto the output name specified by the user.

NOTES
i.cca respects the current geographic region definition and the current mask setting while performing the transformation.

SEE ALSO
Schowengerdt, Robert, A., Techniques for Image Processing and GRASS 4.1 U.S. Army CERL1 Classification in Remote Sensing, Academic Press, 1983.

i.class, i.pca, r.covar, r.mapcalc

AUTHORS
David Satnik, GIS Laboratory, Central Washington University
Ali R. Vali, Space Research Center, University of Texas, Austin

i.class

NAME
i.class **-** An imagery function that generates spectral signatures for an image by allowing the user to outline regions of interest. The resulting signature file can be used as input for *i.maxlik* or as a seed signature file for *i.cluster*.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.class

DESCRIPTION
i.class performs the first pass in the GRASS two-pass supervised image classification process; the GRASS program *i.maxlik* executes the second pass. Both programs must be run to generate a classified map in GRASS raster format.

i.class is an interactive program that allows the user to outline a region on the screen and calculate the spectral signature based on the cells that are within that region. During this process the user will be shown a histogram of the region for each image band. The user can also display the cells of the image bands that fall within a user-specified number of standard deviations from the means in the spectral signature. By doing this, the user can see how much of the image is likely to be put into the class associated with the current signature.

The spectral signatures that result are composed of region means and covariance matrices. These region means and covariance matrices are used in the second pass (*i.maxlik*) to classify the image.

Alternatively, the spectral signatures generated by *i.class* can be used for seed means for the clusters in the *i.cluster* program.

USER INPUTS
The first screen in the program *i.class* asks the user for the imagery group and subgroup to be analyzed:

```
LOCATION: location  SUPERVISED CLASSIFIER  MAPSET: demo

    Please select the group and subgroup to be analyzed

    GROUP: spot_______   (list will show available groups)
    SUBGROUP: 123________   (list will show available subgroups)


    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
   (OR <Ctrl-C> TO CANCEL)
```

The group should contain the imagery bands that the user wishes to classify. The subgroup is a subset of this group. The user must create a group and a subgroup by running the GRASS program *i.group* before running *i.class*. The subgroup should contain only the image bands that the user wishes to classify. Note that this subgroup must contain more than one band.

After the first screen, the program asks the user for the name of the resulting signature file. The signature file is both the output file for *i.class* and the required input file for the GRASS program *i.maxlik*. It contains the region means and covariance matrices that are used to classify an image in *i.maxlik*.

After entering the resulting signature file name, the user is asked to enter the name of a seed signature file. This is optional. A "seed" signature file is a previously created signature file. Such a seed signature file may be the result of an earlier run of *i.class*. The seed signature file is copied into the new resulting signature file before any new signatures are added by *i.class*. In this way, you can collect the work from several sessions with *i.class* into one signature file.

Next, the user is asked to enter the name of the raster map to be displayed during the process of outlining regions. Typically, the user will want to enter the name of a color composite previously created by *i.composite*. However, the user can enter the name of any existing raster map. This leaves the potential for using a raster map not directly derived from the image as a backdrop on which the user can outline the classes of interest.

At this point the *i.class* graphics screen will be drawn on the graphics monitor and the user will be directed to use the mouse. From this point on the user will primarily work with the mouse, selecting options from the menus and outlining regions on the screen. The only time that the user will need to return to the text terminal is to enter names for the signatures created.

THE DISPLAY FRAMES
The display frame layout that *i.class* uses is represented below for reference.

```
+-----------------------------+-----------------------------+
|   |    |
|   |    Map Display Frame   |
|   |    |
|   |    |
|   Histogram Display   |    |
|   Frame|    |
|   |    |
|   |    |
|   |    |
|   +-----------------------------+
|   |    |
|   |    Zoom Display Frame  |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
+-----------------------------+-----------------------------+
|   Menu Frame|
+-----------------------------------------------------------+
```

THE MENUS
All of the menus in the *i.class* program are displayed across the bottom of the graphics monitor in the Menu Frame. To select an option from one of these menus, simply place the cursor over your selection and press any button on the mouse. Each of the menus is discussed in the following paragraphs.

The Command Menu
The Command Menu includes the following selections:

Zoom **-** This command allows the user to outline a rectangular region in either the Map or Zoom Display Frames and the region is displayed, magnified, to fit in the Zoom Display Frame. A red rectangle is drawn in the Map Display Frame, indicating what area the Zoom Display Frame shows.

To outline the rectangular region simply use any mouse button to anchor the first corner of the border and then use any button to choose the other corner.

Define region **-** This selection takes the user to the Region Menu. This menu includes the options that allow the user to outline a region of interest on the displayed raster map.

Redisplay map - This selection takes the user to the Redisplay Menu. The Redisplay Menu allows the user to redraw map display frames.

Analyze region - This selection starts the process of analyzing the currently defined region. A histogram of the defined region will be displayed for each band. On the histogram for each band, the mean, standard deviation, minimum cell value and maximum cell value are marked. The histograms are automatically scaled in an attempt to fit the data into the space available, but it is possible that all of the data will not fit. In this case, as much of the data as possible, centered around the mean, will be displayed. After the histograms are displayed, the user will be given the Signature Menu.

Quit - The user should make this selection to end the session with *i.class*.

The Region Menu
The Region Menu contains the following selections:

Erase region - This selection erases any currently defined region.

Draw region - This selection allows the user to use the mouse to draw a region on either the Map or Zoom

Display Frame. An explanation of which mouse buttons to use is displayed in the Menu Frame. The user does not need to try to complete the region boundary. The last line of the region will be added when the user selects the Complete region option on the Region Menu.

Restore last region - This selection restores the last region that was drawn. After a region is completed, it will be saved to be restored later. Only one previous region is saved.

Complete region - This selection completes the region that is currently being drawn. As noted above, it saves the complete region to be restored later, if needed. Once the user has made a complete region, it can be analyzed with the Analyze Region selection on the Command Menu.

Done - Use this selection to return to the Command Menu.

The Redisplay Map Menu
The Redisplay Map Menu has the following selections, which are useful to redraw the raster maps displayed in the Map and Zoom Display Frames.

Map geographic region - This selection causes the raster map in the Map Display Frame to be redrawn.

Zoom region - This selection causes the Zoom Display Frame to be redrawn.

Both - This selection causes both the Map and Zoom Display Frames to be redrawn.

Cancel - Use this selection if you do not want to redisplay either of the above regions. The user will be returned to the Command Menu.

The Analyze Region Menu
The Analyze Region Menu contains the Signature Menu, which allows the user to set the number of standard deviations and the display color, and then to display (as an overlay) the cells that match the signature within the number of standard deviations specified. Note that once the matching cells are displayed, the Map Display Frame must be redisplayed to see only the original raster map again. The following selections are available on the Signature Menu:

Set std dev's **-** This selection allows the user to set the number of standard deviations from the mean for the maximum and minimum range. The maximum and minimum range is used when finding the cells that "match" the signature. The user is presented with a menu of typical choices and an "Other" option. If the "Other" option is selected, enter the number of standard deviations from the keyboard on the text terminal. Otherwise, the selected option will be used. When the number of standard deviations is set, the histograms for each band will be redrawn with the maximum and minimum range marked.

Note that the number in parentheses on this selection is the current number of standard deviations.

Set color **-** This selection allows the user to set the color for the display of cells that "match" the current signature. The user is presented with a menu of color choices. The color selected will be used when the Display Matches Menu selection is made.

Note that the color in parentheses on this selection is the current color for display.

Display matches **-** This selection displays the cells that "match" the current signature in the current color. A cell "matches" the current signature if the cell value in each band is between the minimum range and maximum range for that band defined by the number of standard deviations currently set.

Done **-** When this selection is chosen, the user will be asked whether or not he/she would like to save the current signature. If the user answers with the "Yes" selection, he/she will be asked to enter a description for the resultant signature file on the text terminal keyboard. The saved signature file description will be used by *i.maxlik* to name the category that is created from the current signature. After either a "No" answer or the signature description is entered, the user is returned to the Command Menu.

NOTES
i.class uses the current MASK to generate the overlay for cells that match a signature. As a result, if a MASK already exists it will be removed during the execution of this program.

The cell values in the image bands cannot fall outside of the range of 0 to 255. *i.class* will report an error if they do.

i.class, like some of the other imagery programs, does not use the standard GRASS display frames. After running *i.class,* you will need to create a display frame (e.g., using *d.frame* or *d.erase*) before you can use most of the GRASS display (d.) commands.

i.group must be run before *i.class* to create an imagery group and a subgroup containing the image bands to be classified.

The user can perform a supervised image classification by running *i.class* followed by *i.maxlik*. The user can perform an unsupervised classification by running *i.cluster* followed by *i.maxlik*.

i.class is interactive and requires no command line arguments. The user must be running a graphics display monitor (see *d.mon*) to run this program.

SEE ALSO
GRASS Tutorial Image Processing
d.frame, d.mon, g.region, i.cca, i.cluster, i.composite, i.group, i.maxlik, r.mapcalc, r.mask

AUTHOR
David Satnik, Central Washington University GIS Laboratory

i.cluster

NAME
i.cluster **-** An imagery function that generates spectral signatures for land cover types in an image using a clustering algorithm. The resulting signature file is used as input for *i.maxlik*, to generate an unsupervised image classification.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.cluster
i.cluster help
i.cluster [-q] group=name subgroup=name sigfile=name classes=value [seed=name] [sample=row_interval,col_interval] [iterations=value] [convergence=value] [separation=value] [min_size=value] [reportfile=name]

DESCRIPTION
i.cluster performs the first pass in the GRASS two-pass unsupervised classification of imagery, while the GRASS program *i.maxlik* executes the second pass. Both programs must be run to complete the unsupervised classification.

i.cluster is a clustering algorithm that reads through the (raster) imagery data and builds pixel clusters based on the spectral reflectance of the pixels. The pixel clusters are imagery categories that can be related to land cover types on the ground. The spectral distributions of the clusters (which will be the land cover spectral signatures) are influenced by six parameters set by the user. The first parameter set by the user is the initial number of clusters to be discriminated. *i.cluster* starts by generating spectral signatures for this number of clusters and "attempts" to end up with this number of clusters during the clustering process. The resulting number of clusters and their spectral distributions, however, are also influenced by the range of the spectral values (category values) in the image files and the other parameters set by the user. These parameters are: the minimum cluster size, minimum cluster separation, the percent convergence, the maximum number of iterations, and the row and column sampling intervals.

The cluster spectral signatures that result are composed of cluster means and covariance matrices. These cluster means and covariance matrices are used in the second pass (*i.maxlik*) to classify the image. The clusters or spectral classes result can be related to land cover types on the ground.

OPTIONS
The program can be run either non-interactively or interactively. It will be run non-interactively if the user specifies the name of group file, the name of subgroup file, the name of a file to contain result signatures, the initial number of clusters to be discriminated, and optionally other parameters (see below) on the command line using the form:

i.cluster [-q] group=name subgroup=name sigfile=name classes=value [seed=name]
[sample=row_interval,col_interval] [iterations=value] [convergence=value] [separation=value]
[min_size=value] [reportfile=name]

where the group should contain the imagery files that the user wishes to classify. The subgroup is a subset of this group. The user must create a group and subgroup by running the GRASS program *i.group* before running *i.cluster*. The subgroup should contain only the imagery band files that the user wishes to classify. Note that this subgroup must contain more than one band file. The purpose of the group and subgroup is to collect map layers for classification or analysis. The sigfile is the file to contain result signatures, which can be used as input for *i.maxlik*. The classes value is the initial number of clusters to be discriminated; any parameter values left unspecified are set to their default values. Alternatively, the program will be run interactively if the user types only *i.cluster*; in this case the program will prompt the user for parameter values using the standard GRASS *parser* interface described in the manual entry for *parser*.

Flags:
-*q* Run quietly. Suppresses output of program percent-complete messages and the time elapsed from the beginning of the program. If this flag is not used, these messages are printed out.

Parameters:
group=name The name of the group file which contains the imagery files that the user wishes to classify.

subgroup=name The name of the subset of the group specified in group option, which must contain only imagery band files and more than one band file. The user must create a group and a subgroup by running the GRASS program *i.group* before running *i.cluster*.

sigfile=name The name assigned to output signature file which contains signatures of classes and can be used as the input file for the GRASS program *i.maxlik* for an unsupervised classification.

classes=value The number of clusters that will initially be identified in the clustering process before the iterations begin.

seed=name The name of a seed signature file is optional. The seed signatures are signatures that contain cluster means and covariance matrices, which were calculated prior to the current run of *i.cluster*. They may be acquired from a previously run of *i.cluster* or from a supervised classification signature training site section (e.g., using the signature file output by *i.class*). The purpose of seed signatures is to optimize the cluster decision boundaries (means) for the number of clusters specified.

sample=row_interval,col_interval These numbers are optional with default values based on the size of the data set such that the total pixels to be processed is approximately 10,000 (consider round up).

iterations=value This parameter determines the maximum number of iterations which is greater than the number of iterations predicted to achieve the optimum percent convergence. The default value is 30. If the number of iterations reaches the maximum designated by the user; the user may want to rerun *i.cluster* with a higher number of iterations (see reportfile).
Default: 30

convergence=value A high percent convergence is the point at which cluster means become stable during the iteration process. The default value is 98.0 percent. When clusters are being created, their means constantly change as pixels are assigned to them and the means are recalculated to include the new pixel. After all clusters have been created, *i.cluster* begins iterations that change cluster means by maximizing the distances between them. As these means shift, a higher and higher convergence is approached. Because means will never become totally static, a percent convergence and a maximum number of iterations are supplied to stop the iterative process. The percent convergence should be reached before the maximum number of iterations. If the maximum number of iterations is reached, it is probable that the desired percent convergence was not reached. The number of iterations is reported in the cluster statistics in the report file (see reportfile).
Default: 98.0

separation=value This is the minimum separation below which clusters will be merged in the iteration process. The default value is 0.0. This is an image-specific number (a "magic" number) that depends on the image data being classified and the number of final clusters that are acceptable. Its

determination requires experimentation. Note that as the minimum class (or cluster) separation is increased, the maximum number of iterations should also be increased to achieve this separation with a high percentage of convergence (see convergence).
Default: 0.0

min_size=value This is the minimum number of pixels that will be used to define a cluster, and is therefore the minimum number of pixels for which means and covariance matrices will be calculated.
Default: 17

reportfile=name The reportfile is an optional parameter which contains the result, i.e., the statistics for each cluster. Also included are the resulting percent convergence for the clusters, the number of iterations that was required to achieve the convergence, and the separability matrix.

NOTES
Running in command line mode, *i.cluster* will overwrite the output signature file and reportfile (if required by the user) without prompting if the files existed.

SEE ALSO
GRASS Tutorial: Image Processing
i.class, i.group, i.gensig, i.maxlik

AUTHORS
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory
Tao Wen, University of Illinois at Urbana-Champaign, Illinois

i.colors

NAME
i.colors **-** An imagery function that creates colors for imagery groups.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.colors

DESCRIPTION
i.colors allows the user to interactively assign red, green, and blue colors to the band files in an imagery group while viewing the display of the combined bands on the graphics monitor.

The user is asked to select an imagery group. Band files in the group that are currently assigned to the red, green, and blue color bands are displayed along the left edge of the user's graphics display frame, and a composite image of these bands is displayed in the right portion of the display frame. A small matrix appears in the lower left portion of the display frame, and reflects current red, green, and blue band file color assignments. Red, green, and blue color boxes appear next to the names of the image band files to which they are currently assigned. The user can change current color assignments by clicking the mouse cursor beside desired color assignments.

A menu along the bottom of the graphics display frame indicates that the user can also replot and enlarge the composite image on display, or quit the program, by clicking the mouse cursor on the desired option.

The GRASS program *i.group* can also be used to assign red, green, and blue colors to imagery group band files.

NOTES
This program is not yet complete. To assign colors to an imagery group use the GRASS program *i.group*.

This program is interactive and requires no command line arguments. The user must be running a graphics monitor to use this program.

SEE ALSO
d.mon, hsv.rgb.sh, i.composite, i.grey.scale, i.group, i.his.rgb, i.rgb.his, r.mapcalc, rgb.hsv.sh

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.composite

NAME
i.composite **-** An imagery function that creates a color composite image from three imagery band files specified by the user.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.composite

DESCRIPTION
i.composite creates a color composite image from three band files specified by the user. The user specifies the bands to be used by assigning a red, blue, and/or green color to each band. The resulting image is a raster map layer of raw spectral data composed of the three bands chosen by the user. The color composite can then be displayed, painted, or manipulated as would any raster map layer in GRASS.

The first prompt asks the user for the imagery group whose files are to be used.

The following menu is then displayed:

Please indicate which files to use for red, green, and blue colors. You may leave any color out. You may specify more than one color per file. However, each color may only be specified once. For example, to get a full color image, specify r, g, b for 3 different files. To get a grey scale image, specify rgb for a single file.

```
b__    spot.1
g__    spot.2
r__    spot.3
___    spotclass
___    spotreject

   AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
  (OR <Ctrl-C> TO CANCEL)
```

The user is then allowed to check the choice of bands:

Colors assigned as follows:

```
 RED:spot.3@mapsetname
 GREEN:   spot.2@mapsetname
 BLUE:    spot.1@mapsetname

Look ok? (y/n) [y]
```

The color table that is created has 1000 colors (10 saturation levels (or shades) per primary color (blue, green, red)). The number of colors that can be displayed at one time on a color graphics monitor will depend on the graphics monitor being used. For example, if the graphics monitor can only display 512 colors at one time, then the user must run the GRASS command *d.colormode* mode=fixed before displaying the raster map layer. The colors that cannot be displayed will be assigned to the nearest displayable color, and the raster map layer will retain its relative color accuracy. If the user is in float color mode, however, the raster map layer displayed on the graphics monitor will not accurately reflect the map's real color assignments.

The user is then asked to name the composite image raster map layer. The percentage completed is echoed to the screen and *r.support* files are created automatically.

NOTES
The user should always check the geographic region settings before running most imagery commands. It is very easy for the boundaries of the geographic region to be completely off the image. Before running *i.composite*, or other imagery commands, the user should probably set the geographic region to match that of the raster map layers to be read. This can be accomplished using option 4 of the *g.region* command.

This program is interactive and requires no command line arguments.

SEE ALSO
d.colormode, d.his, d.rast, g.region, i.colors, i.grey.scale, i.group, r.support
GRASS Tutorial: Image Processing

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.fft

NAME
i.fft **-** Fast Fourier Transform (FFT) for image processing.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.fft
i.fft help
i.fft input_image=name real_image=name imaginary_image=name [range=value]

DESCRIPTION
i.fft is an image processing program based on the algorithm given by Press (1988), with enhancements by Vali (1990), that processes a single input raster map layer (input_image) and constructs the real and imaginary Fourier components in frequency space.

Parameters:

input_image=name	Input raster map layer on which the fast Fourier transform is run.
real_image=name	Output real part arrays stored as raster map layer.
imaginary_image=name	Output imaginary part arrays stored as raster map layer.
range=value	Range of values used during fast Fourier transformation.

The real and imaginary components are stored as arrays of doubles in the cell_misc directory (for use in the inverse transform program, *i.ifft*), and are also scaled and formatted into the real_image and imaginary_image raster map layers for inspection, masking, etc. In these raster map layers the low frequency components are in the center and the high frequency components are toward the edges. The input_image need not be square; before processing, the X and Y dimensions of the input_image are padded with zeroes to the next highest power of two in extent (i.e., 256 x 256 is processed at that size, but 200 x 400 is padded to 256 x 512). The cell category values for viewing, etc., are calculated by taking the natural log of the actual values then rescaling to 255, or whatever optional range is given on the command line, as suggested by Richards (1986). A color table is assigned to the resultant map layer.

The current geographic region and mask settings are respected when reading the input file. The presence of a mask will, in general, make the resulting fast Fourier transform invalid, or at least difficult to interpret.

SEE ALSO
i.cca, i.class, i.ifft, i.pca

Numerical Recipes in C , by William H. Press ,Cambridge University ,Press ,1988.
Remote Sensing Digital Image Analysis, by John A. Richards, Springer-Verlag, 1986.

Personal communication, between program author and Ali R. Vali, Space Research Center, University of Texas, Austin, 1990.

AUTHOR
David Satnik, GIS Laboratory, Central Washington University

i.gensig

NAME
i.gensig **-** Generates statistics for *i.maxlik* from raster map layer.
(GRASS Imagery Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.gensig
i.gensig help
i.gensig trainingmap=name group=name subgroup=name signaturefile=name

DESCRIPTION
i.gensig is a non-interactive method for generating input into *i.maxlik*. It can be used as the first pass in the GRASS two-pass classification process (instead of *i.cluster* or *i.class*). It reads a raster map layer, called the training map, which has some of the pixels or regions already classified. *i.gensig* will then extract spectral signatures from an image based on the classification of the pixels in the training map and make these signatures available to *i.maxlik.*

The user would then execute the GRASS program *i.maxlik* to actually create the final classified map.

COMMAND LINE OPTIONS
Parameters:
trainingmap Ground truth training map

This map must be prepared by the user in advance. Programs like *v.digit* or *r.digit* can be used to define representative areas of the classes the user defines to be in the image. Of course other methods could be devised by the user for creating this training map **-** *i.gensig* makes no assumption about the origin of this map layer. It simply creates signatures for the classes defined in the training map for the image to be classified (the image is specified in other options **-** see below).

group Imagery group

This is the name of the group that contains the band files that comprise the image to be analyzed. The *i.group* command is used to construct groups of raster layers, which comprise an image.

subgroup Subgroup containing image files

This names the subgroup within the group that selects a subset of the bands to be analyzed. The *i.group* command is also used to prepare this subgroup. The subgroup mechanism allows the user to select a subset of all the band files that form an image.

signaturefile Resultant signature file

This is the resultant signature file (containing the means and covariance matrices) for each class in the training map that is associated with the band files in the subgroup select (see above).

INTERACTIVE MODE
If none of the arguments are specified on the command line, *i.gensig* will interactively prompt for the names of these maps and files.

It should be noted that interactive mode here only means interactive prompting for maps and files. It does not mean visualization of the signatures that result from the process.

SEE ALSO
i.group, v.digit, r.digit, i.cluster, i.class

AUTHOR
Michael Shapiro, U.S. Construction Engineering Research Laboratory

i.gensigset

NAME
i.gensigset **-** generate statistics for *i.smap* from raster map layer.
(GRASS Imagery Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.gensigset
i.gensigset help
i.gensigset trainingmap=name group=name subgroup=name signaturefile=name [maxsig=value]

DESCRIPTION
i.gensigset is a non-interactive method for generating input into *i.smap*. It is used as the first pass in the two-pass classification process. It reads a raster map layer, called the training map, which has some of the pixels or regions already classified. *i.gensigset* will then extract spectral signatures from an image based on the classification of the pixels in the training map and make these signatures available to *i.smap*. The user would then execute the GRASS program *i.smap* to create the final classified map.

OPTIONS
Parameters:
trainingmap Ground truth training map

This raster layer, supplied as input by the user, has some of its pixels already classified, and the rest (probably most) of the pixels unclassified. Classified means that the pixel has a non-zero value and unclassified means that the pixel has a zero value.

This map must be prepared by the user in advance. The user must use *r.digit*, a combination of *v.digit* and *v.to.rast*, or some other import/development process (e.g., *v.in.transects*) to define the areas representative of the classes in the image.

At present, there is no fully interactive tool specifically designed for producing this layer.

group Imagery group

This is the name of the group that contains the band files, which comprise the image to be analyzed. The *i.group* command is used to construct groups of raster layers, which comprise an image.

subgroup Subgroup containing image files

This names the subgroup within the group that selects a subset of the bands to be analyzed. The *i.group* command is also used to prepare this subgroup. The subgroup mechanism allows the user to select a subset of all the band files that form an image.

signaturefile Resultant signature file

This is the resultant signature file (containing the means and covariance matrices) for each class in the training map that is associated with the band files in the subgroup selected.

maxsig Maximum number of sub-signatures in any class
Default: 10

The spectral signatures that are produced by this program are "mixed" signatures (see **NOTES**). Each signature contains one or more subsignatures (representing subclasses). The algorithm in this program starts with a maximum number of subclasses and reduces this number to a minimal number of subclasses that are spectrally distinct. The user has the option to set this starting value with this option.

INTERACTIVE MODE

If none of the arguments are specified on the command line, *i.gensigset* will interactively prompt for the names of these maps and files.

It should be noted that interactive mode here only means interactive prompting for maps and files. It does not mean visualization of the signatures that result from the process.

NOTES

The algorithm in *i.gensigset* determines the parameters of a spectral class model known as a Gaussian mixture distribution. The parameters are estimated using multispectral image data and a training map which labels the class of a subset of the image pixels. The mixture class parameters are stored as a class signature, which can be used for subsequent segmentation (i.e., classification) of the multispectral image.

The Gaussian mixture class is a useful model because it can be used to describe the behavior of an information class, which contains pixels with a variety of distinct spectral characteristics. For example, forest, grasslands or urban areas are examples of information classes that a user may wish to separate in an image. However, each of these information classes may contain subclasses each with its own distinctive spectral characteristic. For example, a forest may contain a variety of different tree species each with its own spectral behavior.

The objective of mixture classes is to improve segmentation performance by modeling each information class as a probabilistic mixture with a variety of subclasses. The mixture class model also removes the need to perform an initial unsupervised segmentation for the purposes of identifying these subclasses. However, if misclassified samples are used in the training process, these erroneous samples may be grouped as a separate undesired subclass. Therefore, care should be taken to provide accurate training data.

This clustering algorithm estimates both the number of distinct subclasses in each class, and the spectral mean and covariance for each subclass. The number of subclasses is estimated using Rissanen's minimum description length (MDL) criteria [1]. This criteria attempts to determine the number of subclasses which "best" describe the data. The approximate maximum likelihood estimates of the mean and covariance of the subclasses are computed using the expectation maximization (EM) algorithm [2,3].

REFERENCES

[1] J. Rissanen, "A Universal Prior for Integers and Estimation by Minimum Description Length," *Annals of Statistics*, vol. 11, no. 2, pp. 417-431, 1983.

[2] A. Dempster, N. Laird and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Roy. Statist. Soc*. B, vol. 39, no. 1, pp. 1-38, 1977.

[3] E. Redner and H. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Review*, vol. 26, no. 2, April 1984.

SEE ALSO

i.group, v.digit, r.digit, i.smap

AUTHOR

Charles Bouman, School of Electrical Engineering, Purdue University
Michael Shapiro, U.S. Construction Engineering Research Laboratory

i.grey.scale

NAME
i.grey.scale **-** An interactive imagery function that assigns a histogram contrast stretch grey scale color table to a raster map layer.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.grey.scale

DESCRIPTION
i.grey.scale is an interactive imagery function that assigns a histogram contrast stretch grey scale color table to a raster map layer. The histogram contrast stretch expands the original range of digital (category) values to utilize the full range of the user's graphics monitor.

The user is asked for the name of the raster map layer that needs a grey scale. When the raster map layer is displayed it will be displayed with a grey scale color scheme.

NOTES
This program is interactive and requires no command line arguments.

SEE ALSO
d.colormode, d.colors, d.colortable, i.colors, i.composite
GRASS Tutorial: Image Processing

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.group

NAME
i.group **-** An imagery function that creates and edits groups and subgroups of (raster) imagery files. (GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.group

DESCRIPTION
i.group allows the user to collect raster map layers in an imagery group by assigning them to user-named subgroups or other groups. This enables the user to run analyses on any combination of the raster map layers in a group. The user creates the groups and subgroups and selects the raster map layers that are to reside in them. Imagery analysis programs like *i.points*, *i.rectify*, *i.ortho.rectify* and others ask the user for the name of an imagery group whose data are to be analyzed. Imagery analysis programs like *i.cluster* and *i.maxlik* ask the user for the imagery group and imagery subgroup whose data are to be analyzed.

The first menu in the *i.group* program asks the user to select a group. If the group does not exist, the user will be asked if he or she would like to create a new group.

This program edits imagery groups. You may add raster map layers to, or remove such layers from, an imagery group. You may also create new groups.

```
Please enter the group to be created/modified

GROUP: _____________ (enter 'list' for a list of groups)


AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO EXIT)
```

If the word list is entered, groups that have already been created in the user's current LOCATION_name and MAPSET(S) will be listed. The second menu in *i.group* provides the user with the following options:

```
1.    Select a different group
2.    Edit group title
3.    Include new raster (cell) files in the group
 or remove raster (cell) files from the group
4.    Assign colors to the group
5.    Create a new subgroup within the group

RETURN to exit
```

The options are described as follows:

Select a different group
If option number 1 is chosen, the following menu is displayed:

```
Please enter the group to be created/modified

GROUP: _____________ (enter 'list' for a list of groups)


  AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
  (OR <Ctrl-C> TO EXIT)
```

If the word list is entered, groups that have already been created in the current LOCATION_name and MAPSET(S) will be displayed.

Edit group title
If option number 2 is selected, an entry space is provided to type in the group title. This title is useful in identifying each group:

```
TITLE______________________________________
```

This option offers an opportunity to go back and change the entry if it is not correct by asking:

```
Look ok? (y/n).
```

(cell) files from the group
Include new raster (cell) files in the group or remove raster

When choosing option number 3, the following menu is displayed:

```
LOCATION: location   GROUP: spot   MAPSET: demo

If you wish to delete a file from group [spot], remove the 'x' from in front of
the file name.

x_    spot.1 in demo
x_    spot.2 in demo
x_    spot.3 in demo


AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR Ctrl-C> TO CANCEL)
```

Next, a menu listing all the other raster map layers present in the current MAPSET(S) will be displayed:

```
LOCATION: location GROUP: spot MAPSET: demo

Please mark an 'x' by the files to be added in group [spot]

MAPSET: demo

x_   composite1
x_   spotclass1
__   spotclass2

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR Ctrl-C> TO CANCEL)
```

If more than one MAPSET is selected, menus for those mapsets will also be displayed. All raster map layers selected with an 'x' will be included in the group being updated.

The user will then have the opportunity to check the contents of the group that was just modified:

```
Group [spot] references the following raster files

__________________
spot.1  in demo
spot.2  in demo
spot.3  in demo
composite1   in demo
spotclass1   in demo
__________________
Look ok? (y/n)
```

If the user responds with the letter y then the following sentence is displayed on the screen:

```
Group [spot] updated!
```

And the main menu for *i.group* returns.

If the user responds n, the menu containing the group files after it was modified will be displayed and the user will be asked to place an x in front of those raster map layers that are to be removed from the group. Then, a menu listing all of the other raster map layers in the current MAPSET will be displayed again, and the user will be again asked to place an x in front of raster map layers to be included in the group. This gives the user the opportunity to correct mistakes or make changes in the choice of raster map layers to be selected in a group without exiting *i.group*.

Assign colors to the group
Option number 4 provides the following menu:

Please indicate which files to use for red, green, and blue colors. You may leave any color out. You may specify more than one color per file. However, each color may only be specified once. For example, to get a full color image, specify r, g, b for 3 different files. To get a grey scale image, specify rgb for a single file.

```
b__    spot.1
g__    spot.2
r__    spot.3
___    composite1
___    spotclass1

<<< r,g,b can only be specified once >>>

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)
```

This menu allows you to select a color for each imagery band or for each file for display. Note, however, that composite images and classified images are already assigned colors during their creation.

An opportunity to change the choice of colors is offered after escaping the menu by:

```
Look ok? (y/n)
```

Create a new subgroup within the group. The following menu enables the user to create a subgroup out of any combination of raster map layers in the group. Any number of subgroups may be created by repeating the option.

```
LOCATION: location MAPSET: spot

GROUP: spot1
SUBGROUP: ___________ ('list' will show available subgroups)


AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
```

```
        (OR <Ctrl-C> TO CANCEL)
```

After selecting or creating a subgroup, this menu is displayed:

```
        Mark an 'x' by the files to form subgroup [123]

        x_    spot.1
        x_    spot.2
        x_    spot.3
        __    composite1
        __    spotclass1

        AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
        (OR <Ctrl-C> TO CANCEL)
```

The user is then given the opportunity to check the contents of the subgroup:

```
        Subgroup [123] references the following raster (cell) files

        __________________
        spot.1   in demo
        spot.2   in demo
        spot.3   in demo
        __________________
        Look ok? (y/n)
```

If the user responds with the letter n, the group menu will appear again enabling the user to select raster map layers to form the subgroup.

NOTES
The *i.group* options are only available for imagery map layers in the current LOCATION_name.

Subgroup names may not contain more than 12 characters.

SEE ALSO
GRASS Tutorial: Image Processing
i.cluster, i.maxlik, i.points, i.rectify

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.his.rgb

NAME
i.his.rgb **-** Hue-intensity-saturation (his) to red-green-blue (rgb) raster map color transformation function. (GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.his.rgb
i.his.rgb help
i.his.rgb hue_input=name intensity_input=name saturation_input=name red_output=name green_output=name blue_output=name

DESCRIPTION
i.his.rgb is an image processing program that processes three input raster map layers as hue, intensity and saturation components and produces three output raster map layers representing the red, green and blue components of this data. The output raster map layers are created by a standard hue-intensity-saturation (his) to red-green-blue (rgb) color transformation. Each output raster map layer is given a linear gray scale color table. The current geographic region and mask settings are respected.

OPTIONS
Parameters:

hue_input=name	Name of input raster map layer representing hue.
intensity_input=name	Name of input raster map layer representing intensity.
saturation_input=name	Name of input raster map layer representing saturation.
red_output=name	Output raster map layer representing the red component in the data.
green_output=name	Output raster map layer representing the green component in the data.
blue_output=name	Output raster map layer representing the blue component in the data.

NOTES
It is not possible to process three bands with *i.his.rgb* and then exactly recover the original bands with *i.rgb.his*. This is due to loss of precision because of integer computations and rounding. Tests have shown that more than 70% of the original cell values will be reproduced exactly after transformation in both directions and that 99% will be within plus or minus 1. A few cell values may differ significantly from their original values.

SEE ALSO
hsv.rgb.sh, i.colors, i.grey.scale, i.rgb.his, rgb.hsv.sh

AUTHOR
David Satnik, GIS Laboratory, Central Washington University
with acknowledgements to Ali Vali, Univ. of Texas Space Research Center, for the core routine.

i.ifft

NAME
i.ifft **-** Inverse Fast Fourier Transform (ifft) for image processing.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.ifft
i.ifft help
i.ifft real_image=name imaginary_image=name output_image=name

DESCRIPTION
i.ifft is an image processing program based on the algorithm given by Press (1988), and modified by Vali (1990), that converts real and imaginary frequency space images (produced by *i.fft*) into a normal image.

OPTIONS
Parameters:

real_image=name	Input raster map layer for inversion fast Fourier transform, real part.
imaginary_image=name	Input raster map layer for inversion fast Fourier transform, imaginary.
output_image=name	Output inversion raster map layer after fast Fourier transformation.

The current mask is respected when reading the real and imaginary component files; thus, *r.mask* become a primary program for selecting the portion of the frequency space data to be included in the inverse transform. The GRASS program *d.digit* can be used to create masks while viewing the real or imaginary component image. When *i.ifft* is executed, it (automatically) uses the same GRASS region definition setting that was used during the original transformation done with *i.fft*.

The real and imaginary components are read from arrays of doubles in the cell_misc directory (produced by the forward transform program, *i.fft*), and the reconstructed image will preserve the cell value scaling of the original image processed by *i.fft*. No color table is assigned to the output map; one should be created before viewing the output_image.

SEE ALSO
i.cca, i.class, i.fft, i.pca
Numerical Recipes in C, by William H. Press, Cambridge University Press, 1988.
Remote Sensing Digital Image Analysis, by John A. Richards, Springer-Verlag, 1986.
Personal communication, between program author and Ali R. Vali, Space Research Center, University of Texas, Austin, 1990.

AUTHOR
David Satnik, GIS Laboratory, Central Washington University

i.in.erdas

NAME
i.in.erdas -Creates raster files from ERDAS files.
(GRASS Raster Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.in.erdas
i.in.erdas help
i.in.erdas [-l] input=name output=name [trailer=name] [bands=value[,value,...]] [srow=value] [scol=value][rows=value] [cols=value]

DESCRIPTION
This program creates raster map files from ERDAS files. It creates one raster file for each selected band, up to a maximum of seven bands. GRASS color and category support files are created if an ERDAS trailer file is specified.

Only ERDAS version 7.4 or later files in 4-bit, 8-bit, or 16-bit formats are supported. GRASS raster files will be named prefix.band

Note:
Remember that it is necessary to run *r.support* to create the histogram or change the color table. *i.group* to associate the individual raster files as an image group.

OPTIONS
Flag:

-*l*	List the ERDAS header only (no raster files created)

Parameters:

input	ERDAS input file name
output	Output prefix of the GRASS raster files to be created.
trailer	ERDAS trailer input file name
bands	Selected bands to extract. (defaults to all bands)
srow	Starting row. (defaults to first row in file)
scol	Starting column. (defaults to first column in file)
rows	Number of rows to extract. (defaults to all rows)
cols	Number of columns to extract. (defaults to all columns)

SEE ALSO
r.support, i.group

AUTHOR

M. L. Holko, USDA, SCS, NHQ-CGIS
Paul H. Fukuhara, USDA, SCS, NCG-GSS

i.maxlik

NAME
i.maxlik **-** An imagery function that classifies the cell spectral reflectances in imagery data based on the spectral signature information generated by either *i.cluster, i.class, or i.gensig*.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.maxlik
i.maxlik help
i.maxlik [-q] group=name subgroup=name sigfile=name class=name [reject=name]

DESCRIPTION
i.maxlik is a maximum-likelihood discriminant analysis classifier. It can be used to perform the second step in either an unsupervised or a supervised image classification. Either image classification methods are performed in two steps. The first step in an unsupervised image classification is performed by *i.cluster*; the first step in a supervised classification is executed by the GRASS program *i.class*. In both cases, the second step in the image classification procedure is performed by *i.maxlik*.

In an unsupervised classification, the maximum-likelihood classifier uses the cluster means and covariance matrices from the *i.cluster* signature file to determine to which category (spectral class) each cell in the image has the highest probability of belonging. In a supervised image classification, the maximum-likelihood classifier uses the region means and covariance matrices from the spectral signature file generated by *i.class*, based on regions (groups of image pixels) chosen by the user, to determine to which category each cell in the image has the highest probability of belonging.

In either case, the raster map layer output by *i.maxlik* is a classified image in which each cell has been assigned to a spectral class (i.e., a category). The spectral classes (categories) can be related to specific land cover types on the ground.

The program will run non-interactively if the user specifies the names of raster map layers, i.e., group and subgroup names, seed signature file name, result classification file name, and any combination of non-required options in the command line, using the form

i.maxlik[-q] group=name subgroup=name sigfile=name class=name [reject=name]

where each flag and options have the meanings stated below.

Alternatively, the user can simply type *i.maxlik* in the command line without program arguments. In this case the user will be prompted for the program parameter settings; the program will run foreground.

OPTIONS
Flags:
-*q* Run quietly, without printing program messages to standard output.

Parameters:
group=name The imagery group contains the subgroup to be classified.

subgroup=name The subgroup contains image files, which were used to create the signature file in the program *i.cluster, i.class*, or *i.gensig* to be classified.

sigfile=name The name of the signatures to be used for the classification. The signature file contains the cluster and covariance matrices that were calculated by the GRASS program *i.cluster* (or the region means and covariance matrices generated by *i.class*, if the user runs a supervised classification). These spectral signatures are what determine the categories (classes) to which image pixels will be assigned during the classification process.

class=name The name of a raster map holds the classification results. This new raster map layer will contain categories that can be related to land cover categories on the ground.

reject=name The optional name of a raster map holds the reject threshold results. This is the result of a chi square test on each discriminant result at various threshold levels of confidence to determine at what confidence level each cell classified (categorized). It is the reject threshold map layer, and contains one calculated confidence level for each classified cell in the classified image. One of the possible uses for this map layer is as a mask, to identify cells in the classified image that have the lowest probability of being assigned to the correct class.

NOTES
The maximum-likelihood classifier assumes that the spectral signatures for each class (category) in each band file are normally distributed (i.e., Gaussian in nature). Algorithms, such as *i.cluster*, *i.class*, or *i.gensig*, however, can create signatures that are not valid distributed (more likely with *i.class*). If this occurs, *i.maxlik* will reject them and display a warning message.

This program runs interactively if the user types *i.maxlik* only. If the user types *i.maxlik* along with all required options, it will overwrite the classified raster map without prompting if this map existed.

SEE ALSO
Grass Tutorial: Image Processing
i.class, i.cluster, i.gensig i.group, r.mask

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory
Tao Wen, University of Illinois at Urbana-Champaign, Illinois

i.ortho.photo

NAME
i.ortho.photo **-** An interactive imagery function for the ortho-rectification of imagery group files.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.ortho.photo

DESCRIPTION
i.ortho.photo allows the user to ortho-rectify imagery group files. An imagery group consists of several scanned aerial photographs (raster files) of a common area. Imagery groups can be created or modified using the GRASS Image Processing Program *i.group*, or using the first menu option described below. *i.ortho.photo* guides the user through the steps required to ortho-rectify the raster files in a single imagery group.

The first menu in *i.ortho.photo* provides the user with the following options:

```
Initialization Options
1) Select/Modify imagery group
2) Select/Modify imagery group target
3) Select/Modify target elevation model
4) Select/Modify imagery group camera

Transformation Parameter Computation
5) Compute image-to-photo transformation parameters
6) Compute photo-to-target transformation parameters
7) Initialize exposure station parameters

Ortho-rectification Option
8) Ortho-rectify imagery group raster files

RETURN to exit
>
```

The options are described as follows:

Select/Modify imagery group
The current imagery group is display at the top of the previous menu. You may select another (new or existing) imagery group for the ortho-rectification program using option (1). After choosing option (1) you will be prompted for the name of a new or existing imagery group. Option (1) using the GRASS Image Processing Program *i.group*, for creation or modification of imagery groups. For more information on imagery group creation or modification please the GRASS manual page for *i.group*.

Select/Modify imagery group target
The target location and mapset may be selected or modified using option (2). After choosing option (2) you will be prompted for the names of the target location and mapset, where the ortho-rectified raster files will reside. The target location is also the location from which the elevation model (raster file) will be selected **--** see option (3). Option (2) uses the GRASS Image Processing Program *i.target*, for selection or modification of the imagery group target location and mapset. For more information on imagery group selection or modification please the GRASS manual page for *i.target*.

Select/Modify imagery group
Option (3) allows you to select the raster file from the target location to be used as the elevation model. The elevation model is required for both the computation of photo-to-target parameters (option 6) and for the ortho-rectification of the imagery group files (option 8). The raster file select for the elevation model should cover the entire area of the image group to be ortho-rectified. Currently, the elevation model raster file is expected to be in units of meters. DTED and DEM files are suitable for use as the elevation model in the ortho-rectification program. After selection option (3) you will be prompted for the name of the raster file in the target location that you want to use as the elevation model.

Select/Modify imagery group camera
Using option (4) you may select or create an camera reference file that will be used with the current imagery group. A camera reference file contains information on the internal characteristics of the aerial camera, as well as the geometry of the fiducial or reseau marks. The most important characteristic of the camera is its focal length. Fiducial or reseau marks locations are required to compute the scanned image to photo-coordinate transformation parameter (option 5). For a more detailed description of option (4) please see the GRASS manual page for *photo.camera*.

Compute image-to-photo transformation parameters
The scanned image to photo-coordinate transformation parameters are computed using option (5). In this interactive option you associate scanned reference points (fiducials, reseau marks, etc.) with their known photo coordinates from the camera reference file.

Complete documentation for this option is available under the manual entry *photo.2image.*

Compute photo-to-target transformation parameters
The photo to target transformation parameters are computed using option(6). Here control points are marked on one or more imagery group files and associated with their known UTM and elevation coordinates. Complete documentation for this option is available under the manual entry *photo.2target*.

Select initial exposure station parameters
If option (7) is selected, initial camera exposure station parameters and initial variances may be selected or modified. Complete documentation for this option is available under the manual entry *photo.init*.

Ortho-photo imagery group files
Option (8) is used to perform the actual image ortho-rectification after all of the transformation parameters have been computed. Ortho-rectified raster files will be created in the target location for each selected imagery group file. You may select either the current window in the target location or the minimal bounding window for the ortho-rectified image. Complete documentation for this option is available under the manual entry *photo.rectify*.

NOTES
i.ortho.photo currently requires the elevation model to be in meters, and the target location to be a UTM coordinate system.

SEE ALSO
photo.camera[2], photo.2image[2], photo.2target[2], photo.init[2], photo.rectify[2]

AUTHOR
Mike Baba, DBA Systems, Inc.

i.pca

NAME
i.pca **-** Principal components analysis (pca) program for image processing.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.pca
i.pca help
i.pca input=name,name[,name,name,...] output=name rescale=min,max

DESCRIPTION
i.pca is an image processing program based on the algorithm provided by Vali (1990), that processes n (2 >= n) input raster map layers and produces n output raster map layers containing the principal components of the input data in decreasing order of variance ("contrast"). The output raster map layers are assigned names with .1, .2, ...n suffixes. The current geographic region definition and mask settings are respected when reading the input raster map layers. When the rescale option is used, the output files are rescaled to fit the min,max range.

OPTIONS
Parameters:
input=name,name[,name,name,...] Name of two or more input raster map layers.

output=name The output raster map layer name to which suffixes are added. Each output raster map layer is assigned this user-specified name with a numerical (.1, .2, ...n) suffix.

rescale=min,max The optional output category range. (Default: 1,255) If rescale=0,0, no rescaling is performed on output files.

SEE ALSO
i.cca, i.class, i.fft, i.ifft, m.eigensystem, r.covar, r.mapcalc

Richards, John A., Remote Sensing Digital Image Analysis, Springer-Verlag, 1986.
Richards (l986) gives a good example of the application of principal components analysis (pca) to a time series of LANDSAT images of a burned region in Australia.

AUTHOR
David Satnik, GIS Laboratory, Central Washington University
Major modifications for GRASS 4.1 were made by Olga Waupotitsch and Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.points

NAME
i.points **-** An imagery function that enables the user to mark coordinate system points on an image to be rectified and then input the coordinates of each point for creation of a coordinate transformation matrix. (GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.points

DESCRIPTION
i.points is an imagery function that enables the user to mark points on a (raster) image to be rectified and then input the geographic coordinates of each point for calculation of a coordinate transformation matrix. *i.points* must be followed by use of the GRASS program, which rectifies the image using the transformation matrix coefficients calculated by *i.points*.

Rectification is the mapping (transformation) of an image from one coordinate system to another. The geometry of an image extracted into a GRASS LOCATION having an x,y coordinate system is not planimetric. To create a planimetric image, that is, to convert the x,y coordinate system into a standard coordinate system (for example, the UTM coordinate system or the State Plane coordinate system), points from a map having the standard coordinates must be associated with the same points on the image to be rectified. *i.points* enables the user to mark points on an image and input the standard coordinates for that point. *i.points* then calculates a least squares regression using the two coordinate systems (x,y and standard) for the marked points. A matrix containing transformation coefficients is the output file for *i.points*.

During the process of marking points and entering map coordinates, the user can compute the RMS (root mean square) error for each point entered. *i.points* does this by calculating the transformation equation (the same one that is calculated in the GRASS program *i.rectify*), and then plugging these results into an equation for RMS error.

i.points offers a zoom option to locate precisely the point to be marked on an image. This program also offers the user the option of acquiring standard coordinates for a marked point from a map layer in the target database.

i.target must be run before running *i.points* to enable the PLOT RASTER option to be used and to identify a target data base LOCATION_name and MAPSET for the rectified image. To run *i.points*, a graphics monitor is required.

The procedure for marking points, entering coordinates, and calculating RMS error is described below.

The first prompt in the program asks the user for the imagery group to be registered. Note that if *i.target* is not run before *i.points*, the *i.points* program will display the following error message:

```
ERROR: Target information for group [spot] missing
Please run i.target for group [spot]
```

After entering the group to be registered the terminal screen displays the message:

```
Use mouse now…
```

The graphics monitor displays the following screen:

```
 ____________________________________________________________________
| imagery   filename   (mag)  |  target   filename   (mag)  |
|_____________________________|_____________________________|
|                             |                             |
|                             |                             |
|                             |                             |
|                             |                             |
|                             |                             |
|                             |                             |
|_____________________________|_____________________________|
|                             |                             |
|                             |                             |
|                             |                             |
|                             |                             |
|                             |                             |
|                             |                             |
|_____________________________|_____________________________|
|QUIT ZOOM PLOT RASTER ANALYZE|                             |
|_____________________________|_____________________________|
```

A pop-down menu like that shown below will be superimposed on the left half of the screen:

```
 _________________________________
| Double click on raster map layer|
| to be plotted                   |
|   Double click here to cancel   |
|_________________________________|

 ____________________
|    Mapset demo     |
|____________________|
| spotclass|  spot.1|
|__________|________|
| composite|  spot.2|
|__________|________|
| spot.3   |        |
|__________|________|
```

Any single raster map layer in the imagery group may be used on which to mark points, and the user can mark points on more than one raster map layer in the imagery group to accumulate the suggested minimum number of 12 points. Any raster map layer in the imagery group can be rectified (using *i.rectify*) based on the transformation matrix computed from these points.

The imagery file chosen by the user is displayed in the upper left quadrant of the screen.

ZOOM
To magnify the displayed file, the user must place the mouse cross hairs on the word ZOOM. The following menu will then be displayed at the bottom of the screen:

```
 ___________________________________________
| Cancel|  Box|  Point|  Select type of ZOOM|
|_______|_____|_______|_____________________|
```

The user has the option of identifying the zoom region either by using the mouse to make a box, or by using the mouse to mark the two diagonal points of the desired region. The terminal screen will display a mouse button menu to guide the user in identifying the corner points of the region.

MARKING POINTS
To mark the points on the image that correspond to the points on a standard coordinate system map, the user must place the mouse cross hairs on the corresponding location on the image to be marked and press the left hand button on the mouse. A diamond shaped symbol will be marked on the image.

The user's terminal will display the following menu:

```
 ____________________________________________________________
| Point 1 marked on the image at  |                          |
| East: 1023.77                   |                          |
| North: -164.41                  |                          |
|                                 |                          |
|                                 |                          |
|                                 |                          |
|                                 |                          |
|                                 |                          |
|____________________________________________________________|
| Enter coordinates as east north:                           |
|____________________________________________________________|
```

The user then enters the easting and northing (separated by a space) for the point marked on the image. If the user wishes not to enter a coordinate, he or she may simply hit RETURN to return control to the mouse; the marked point then disappears.

PLOT RASTER
In addition to acquiring reference points from a standard map, the user has the option of acquiring the reference points from a raster map layer in the target database LOCATION_name. The data base raster map layer is displayed by placing the mouse cross hairs on the words PLOT RASTER. The following line is then displayed at the bottom of the graphics monitor:

```
 ________________________________________________
| Cancel |  Indicate which side should be plotted|
|_______ |_______________________________________|
```

Which side of the graphics monitor is to be plotted is indicated by placing the mouse cross hairs on the half of the graphics monitor screen that the user would like to use, and pressing the left mouse button. The following pop-down menu will be superimposed on the half of the screen that was chosen:

```
 _________________________________________
| Double click on raster (cell) map layer |
| to be plotted                           |
| Double click here to cancel             |
|_________________________________________|

 ______________________________
|         Mapset demo          |
|______________________________|
| tm.rectified  |              |
|_______________|______________|
| tm.classified |              |
|______________________________|
|  Mapset PERMANENT            |
|______________________________|
| elevation    |  geology      |
|______________|_______________|
| slope        |       soils   |
|______________|_______________|
| aspect        |              |
|_______________|______________|
| roads         |              |
|_______________|______________|
| streams       |              |
|_______________|______________|
| airfields     |              |
|______________________________|
```

After the raster map layer is displayed the following message appears at the bottom of the graphics monitor:

```
 _____________________________________
| input method -->|  keyboard|  screen|
|_________________|__________|________|
```

If the user wishes to use the plotted raster map layer only as a comparative reference, then the keyboard can be chosen as the means to input coordinates corresponding to the marked points on the image. This is done by placing the mouse cross hairs on the word KEYBOARD and pressing the left button on the mouse.

If the user selects the SCREEN option, then points marked on the image will automatically be associated with the coordinates from the corresponding points on the target data base map layer. In this option, when the user marks a point on the image, the following menu is displayed at the terminal:

```
 ___________________________________________________________
| Point 5 marked on the image at |                          |
| East: 1023.77                  |                          |
| North: -164.41                 |                          |
|                                |                          |
|                                |                          |
| Point located at               |                          |
| East: 679132.57                |                          |
| North: 4351080.67              |                          |
|                                |                          |
|                                |                          |
|                                |                          |
|                                |                          |
|___________________________________________________________|
| use mouse now...                                          |
|___________________________________________________________|
```

The user then uses the mouse to mark a corresponding point on the displayed image from the target database. The coordinates for the target data base map layer are automatically saved as the coordinates corresponding to the marked point on the image.

ANALYZE

After a number of points have been marked (4 to 7), the user can check the RMS error of the points marked on the image. This is done by placing the mouse cross hairs on the word ANALYZE at the bottom of the graphics monitor. An error report resembling that shown below is superimposed on the graphics monitor:

```
 ______________________________________________________________________
|    error  image  target            |
| #rowcol    target    east    north east  north  |
|______________________________________________________________________|
|10.0-0.9    1.01048.5   -144.8   679132.5   4351080.6|
|20.41.01.32153.1   -567.2   684314.7   4399001.4|
|3    -1.2-0.5.61452.8   -476.5   567841.4   3457682.8|
|41.10.51.31034.0   -109.2   677573.8   4352626.4|
|5    -2.714.0    14.2    1048.6   -144.9   679132.6   4351080.7|
|                                                                    |
|____________________________________________________________________|
|    overall   rms    error:   4.46                                   |
|____________________________________________________________________|
```

The following menu then appears at the bottom of the graphics monitor:

```
 ___________________________________________________________________
| DONE|  PRINT FILE|   Double click on point to be included/excluded|
|_____|____________|________________________________________________|
```

The RMS error for the image is given under the column titled "error" and subtitled "row" and "col". In the above report, point number 1 is 0.0 rows and -0.9 columns from the predicted location calculated from the transformation equation. The RMS error for the target raster map layer is listed under the heading "target". This is the RMS error for the east and north coordinates of the target map layer, but it is presented in the table using one general value. The overall RMS error is displayed at the bottom of the screen in meters. Points that create high RMS error are displayed in red on the graphics monitor (represented here in italics).

The location of the point marked on the imagery group file is given under the heading "image" and the subheadings "east" and "north". The location of the point in the target database is given under the heading "target" and the subheadings "east" and "north". If the user would like to exclude or include a point, this can be accomplished by placing the mouse cross hairs on the point number to be included (if the point is absent) or excluded (if the point is displayed) and pressing the left button on the mouse twice. When a point is excluded, it is not afterwards included in the calculation of the RMS error, or included in the final transformation matrix. However, it can be retrieved within *i.points* at any time by double clicking with the mouse as described above.

QUIT
To end the *i.points* program place the mouse cross hairs on the word QUIT; the marked points (including coordinates) will be saved.

NOTES
A good rule of thumb is to mark at least 12 to 15 points that are evenly distributed over the entire imagery group file in order to obtain an accurate transformation equation for the rectification process. The RMS error may increase with more points added, but the transformation equation will be more accurate.

An RMS error of less than or equal to approximately one resolution unit (pixel or cell) for the image being rectified is generally considered acceptable.

In order to use a digitizer with *i.points*, at least one digitizer driver besides "none" (the on-screen digitizer) must be available in the digitcap file.

This program is interactive and requires no command line arguments.

SEE ALSO
g.mapsets, i.group, i.rectify, i.target

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.quantize

NAME
i.quantize **-** An interactive imagery function that creates a raster map layer whose color table is based on the red, green, and blue color values present in existing, user-specified imagery group files.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.quantize

DESCRIPTION
i.quantize is an interactive imagery function that allows the user to create a new raster map layer whose color table values are based on the red, green, and blue color values present in (as many as) three raster map layers in an imagery group.

The user is first asked to enter the name of the imagery group from whose map layers red, green, and blue color values are to be extracted. The user is then shown a listing of imagery files in the specified group, and is asked to indicate which files in the group are to be used for red, green, and blue colors in the new map layer. Each color (red, green, and blue) must be specified once and only once.

The user is also asked to assign a name to the new raster map layer output.

NOTES
This program is interactive and requires no command line arguments.
The input files must ALL be within the range of 0-255.

SEE ALSO
d.colormode, d.colors, d.colortable, hsv.rgb.sh, i.colors, i.composite, i.grey.scale, i.group, i.his.rgb, i.rgb.his, r.mapcalc, rgb.hsb.sh, r.colors

Paul Heckbert "Image Quantization" Siggraph Proceedings 1982

AUTHOR
David Gerdes , U.S. Army Construction Engineering Research Laboratory

i.rectify

NAME
i.rectify **-** An imagery function that rectifies an image by computing a coordinate transformation for each cell (pixel) in the image using the transformation coefficient matrix created by the GRASS program *i.points*.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.rectify

DESCRIPTION
i.rectify rectifies an image by using the transformation coefficient matrix created by *i.points*. Rectification is the process by which the geometry of an image is made planimetric. This is accomplished by mapping (transforming) an image from one coordinate system to another. In *i.rectify*, the coefficients computed by *i.points* are used in an equation to convert x,y coordinates to standard map coordinates for each cell in the image. The result is an image with a standard map coordinate system. Upon completion of the program the rectified image is deposited in a previously targeted GRASS LOCATION_name and MAPSET.

The first prompt in the program asks the user for the name of the group containing the raster map layers to be rectified.

The user is then asked to select the map layer(s) within the group to be rectified:

```
Please select the file(s) to rectify by naming an output file

   spot.1 in demo  .........
   spot.2 in demo  .........
   spot.3 in demo  .........
   spotclass in demo    spotrectify..
   spotreject in demo   .........

(Enter list by any name to get a list of existing raster files)

    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL)
```

More than one raster map layer may be rectified at a time. Each raster map layer should have a unique output file name.

Next the user is asked to select one of two geographic region settings:

```
Please select one of the following options
  1.   Use the current region in the target location
  2.   Determine the smallest region which covers the image
```

i.rectify will only rectify that portion of the image that occurs within the chosen geographic region setting. Only that portion will be relocated in the target database. It is therefore important to check the current geographic region settings in the target location if choice number one is selected.

NOTES
i.rectify uses nearest neighbor resampling during the transformation.

i.rectify uses a linear affine transformation:

$x' = ax + by + c$

$y' = Ax + Bt + C$

The a,b,c,A,B,C are determined by least squares regression based on the control points entered. This transformation applies scaling, translation and rotation. It is NOT a general purpose rubber-sheeting, nor is it ortho-photo rectification using a DEM, not second order polynomial, etc. It can be used if (1) you have geometrically correct images, and (2) the terrain or camera distortion effect can be ignored.

Use *i.ortho.photo* for photos for which you have camera information and a DEM, or *i.rectify2* if you want a 2nd or 3rd order polynomial transformation.

i.rectify will run in the background and notify the user by mail when it is finished. The process may take an hour or more depending on the size of the image, the number of files, and the size of the geographic region definition.

The rectified (raster) image will be located in the target LOCATION when the program is completed. The original unrectified raster map layers are not modified or removed.

This program is interactive and requires no command line arguments.

SEE ALSO

i.group, i.points, i.target

AUTHOR

Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.rectify2

NAME
i.rectify2 **-** This routine rectifies an image by computing a coordinate transformation for each pixel in the image based on the control points created by the GRASS program *i.points* or *i.vpoints*.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.rectify2

DESCRIPTION
The *i.rectify2* program uses the control points identified in *i.points* or *i.vpoints* to calculate a transformation matrix based on a first, second , or third order polynomial and then converts x,y cell coordinates to standard map coordinates for each pixel in the image. The result is a planimetric image with a transformed coordinate system (i.e., a different coordinate system than before it was rectified). The *i.vpoints* or *i.points* program must be run before *i.rectify2*, and both programs are required to rectify an image. An image must be rectified before it can reside in a standard coordinate LOCATION, and therefore be analyzed with the other map layers in the standard coordinate LOCATION. Upon completion of *i.rectify2*, the rectified image is deposited in the target standard coordinate LOCATION. This LOCATION is selected using the *i.target* program of GRASS.

Program Prompts

The first prompt in the program asks for the name of the group containing the files to be rectified.

```
Enter the group containing files to be rectified
Enter 'list' for a list of existing imagery groups
Enter 'list -f' for a verbose listing
Hit RETURN to cancel request
>
```

This is the same imagery group that was selected in *i.points* or *i.vpoints* and the group that contains the cell files with the marked points and their associated map coordinates. You are then asked to select the cell file(s) within the group to be rectified:

```
Please select the file(s) to rectify by naming an output file

spot1.1 in mapsetname    ............
spot1.2 in mapsetname   ............
spot1.3 in mapsetname    ............
spotclass1 in mapsetname
spotrectify1..

spotreject1 in mapsetname ............

(enter list by any name to get a list of existing cell files)

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL
```

More than one cell file may be rectified at a time. Each cell file should be given a unique output file name.

Next, you are asked to select one of two windows regions:

```
Please select one of the following options
1.  Use the current window in the target location
2.  Determine the smallest window which covers the image
>
```

The *i.rectify2* program will only rectify that portion of the image or cell file that occurs within the chosen window region, and only that portion of the cell file will be relocated in the target database. It is important therefore, to check the current mapset window in the target LOCATION if choice number one is selected. If you are rectifying a file with plans to patch it to another file using the GRASS program patch[1] or patch[2], choose option number one, the current window in the target location. This window, however, must be the default window for the target LOCATION. When a file being rectified is smaller than the default window in which it is being rectified, zeros are added to the rectified file. Patching files of the same size that contain 0/non-zero data, eliminates the possibility of a no-data line the patched result. This is because, when the images are patched, the zeros in the image are "covered" with non-zero pixel values. When rectifying files that are going to be patched, rectify all of the files using the same default window.

By selecting the 1st ORDER option, the user may select the order of transformation desired:

```
Select order of transformation -->   1st Order   2nd Order   3rd Order
```

The program will immediately recalculate the RMSE and the
number of points required.

Polynomial Transformation Matrix

The ANALYZE function has been changed to support calculating the registration coefficients using a first, second, or third order transformation matrix. The number of control points required for a selected order of transformation (represented by ~n~) is ((n + 1) * (n + 2) / 2) or 3, 6, and 10 respectively. It is strongly recommended that one or more additional points be identified to allow for an overly- determined transformation calculation which will generate the Root Mean Square (RMS) error values for each included point. The RMS error values for all the included control points are immediately recalculated when the user selects a different transformation order from the menu bar. The polynomial equations in the new *i.rectify2* program are performed using a modified Gaussian elimination method instead of the Cramer's rule method used in the *i.rectify* routine.

Program Execution

i.rectify2 will run in the background and notify you by mail when it is finished.

Note: The rectified image or rectified cell files will be located in the target LOCATION when the program is completed. The original unrectified files are not modified or removed.

This program is interactive and requires no command line arguments.

SEE ALSO
i.group, i.points, i.vpoints, i.rectify, i.target

AUTHOR
William R. Enslin, Michigan State University Center for Remote Sensing

i.rgb.his

NAME
i.rgb.his **-** Red-green-blue (rgb) to hue-intensity-saturation (his) function for image processing.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.rgb.his
i.rgb.his help
i.rgb.his red_input=name green_input=name blue_input=name hue_output=name intensity_output=name saturation_output=name

DESCRIPTION
i.rgb.his is an image processing program that processes three input raster map layers as red, green, and blue components and produces three output raster map layers representing the hue, intensity, and saturation of the data. The output raster map layers are created by a standard red- green-blue (rgb) to hue-intensity-saturation (his) color transformation. Each output raster map layer is given a linear gray scale color table. The current geographic region definition and mask settings are respected.

Parameters:

red_input=name	Input raster map layer representing the red component.
green_input=name	Input raster map layer representing the green component.
blue_input=name	Input raster map layer representing the blue component.
hue_output=name	Output raster map layer representing hue.
intensity_output=name	Output raster map layer representing intensity.
saturation_output=name	Output raster map layer representing saturation.

SEE ALSO
hsv.rgb.sh, i.his.rgb, rgb.hsv.sh

AUTHOR
David Satnik, GIS Laboratory, Central Washington University,
with acknowledgements to Ali Vali, Univ. of Texas Space Research Center, for the core routine.

i.smap

NAME
i.smap **-** An imagery function that performs contextual image classification using sequential maximum a posteriori (SMAP) estimation.
(GRASS Imagery Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.smap
i.smap help
i.smap [-mq] group=name subgroup=name signaturefile=name [blocksize=value] output=name

DESCRIPTION
The *i.smap* program is used to segment multispectral images using a spectral class model known as a Gaussian mixture distribution. Since Gaussian mixture distributions include conventional multivariate Gaussian distributions, this program may also be used to segment multispectral images based on simple spectral mean and covariance parameters.

i.smap has two modes of operation. The first mode is the sequential maximum a posteriori (SMAP) mode [1,2]. The SMAP segmentation algorithm attempts to improve segmentation accuracy by segmenting the image into regions rather than segmenting each pixel separately (see **NOTES**).

The second mode is the more conventional maximum likelihood (ML) classification, which classifies each pixel separately, but requires somewhat less computation. This mode is selected with the -*m* flag (see below).

OPTIONS
Flags:
-*m* Use maximum likelihood estimation (instead of smap). Normal operation is to use SMAP estimation (see **NOTES**).

-*q* Run quietly, without printing messages about program progress. Without this flag, messages will be printed (to stderr) as the program progresses.

Parameters:
group The imagery group that defines the image to be classified.

subgroup The subgroup within the group specified that specifies the subset of the band files that are to be used as image data to be classified.

signaturefile The signature file that contains the spectral signatures (i.e., the statistics) for the classes to be identified in the image. This signature file is produced by the program *i.gensigset* (see **NOTES**).

blocksize Size of submatrix to process at one time default: 128. This option specifies the size of the "window" to be used when reading the image data.

This program was written to be nice about memory usage without influencing the resultant classification. This option allows the user to control how much memory is used. More memory may mean faster (or slower) operation depending on how much real memory your machine has and how much virtual memory the program uses.

The size of the submatrix used in segmenting the image has a principle function of controlling memory usage; however, it also can have a subtle effect on the quality of the segmentation in the smap mode. The smoothing parameters for the smap segmentation are estimated separately for each submatrix. Therefore, if the image has regions with qualitatively different behavior, (e.g., natural woodlands and man-made agricultural fields) it may be useful to use a submatrix small enough so that different smoothing parameters may be used for each distinctive region of the image.

The submatrix size has no effect on the performance of the ML segmentation method.

output The name of a raster map that will contain the classification results. This new raster map layer will contain categories that can be related to landcover categories on the ground.

INTERACTIVE MODE
If none of the arguments are specified on the command line, *i.smap* will interactively prompt for the names of the maps and files.

NOTES
The SMAP algorithm exploits the fact that nearby pixels in an image are likely to have the same class. It works by segmenting the image at various scales or resolutions and using the course scale segmentations to guide the finer scale segmentations. In addition to reducing the number of misclassifications, the SMAP algorithm generally produces segmentations with larger connected regions of a fixed class, which may be useful in some applications.

The amount of smoothing that is performed in the segmentation is dependent of the behavior of the data in the image. If the data suggests that the nearby pixels often change class, then the algorithm will adaptively reduce the amount of smoothing. This ensures that excessively large regions are not formed.

REFERENCES
[1] C. Bouman and M. Shapiro, "Multispectral Image Segmentation using a Multiscale Image Model," Proc. of IEEE Int'l Conf. on Acoust., Speech and Signal Proc., pp. III-565 - III-568, San Francisco, California, March 23-26, 1992.

[2] C. Bouman and M. Shapiro, "A Multiscale Random Field Model for Bayesian Image Segmentation," submitted to the IEEE Trans. on Image Processing.

SEE ALSO
i.group, i.gensigset

AUTHOR
Charles Bouman, School of Electrical Engineering, Purdue University
Michael Shapiro, U.S. Construction Engineering Research Laboratory

i.tape.mss

NAME
i.tape.mss **-** An imagery function that extracts Multispectral Scanner (MSS) imagery data from half-inch tape.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.tape.mss

DESCRIPTION
i.tape.mss is a program that extracts Multispectral Scanner (MSS) imagery data from tape.

This program must be run in a LOCATION_name with a x,y coordinate system (i.e., a coordinate system with projection 0). For further information regarding this LOCATION_name refer to the manual entry for imagery.
The first prompt in *i.tape.mss* asks the user for the tape device name. This is sometimes /dev/rmt0 (for a half-inch tape with a tape density of 1600), but this varies with each machine.

The next prompt is:

```
Please mount and load tape, then hit RETURN -->
```

IMAGE IDENTIFICATION MENU
The first menu in the program asks the user for information about the data.

```
Please enter the following information

    Tape Identification:    __

    Image Description: __

    Title for the Extracted Raster (Cell) Files:__

    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL)
```

This program automatically enters the scene ID number and the date of the image into the field for Tape Identification. The sun angles are automatically entered into the field for Image Description.

The second menu is:

```
MSS TAPE EXTRACTION
please select the desired tape window (geographic region definition) to extract

  first row: _______(1-2984)
  last row: _______(1-2984)

  first col: _______(1-3548)
  last col: _______(1-3548)


    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL)
```

The numbers in parentheses are the total number of rows and columns on the tape including filler (zeros). This information and additional information can also be obtained by running the GRASS program *i.tape.mss.h*, which reads the header information on an MSS tape. Any subset of the image on tape may be extracted. For a discussion of row and column extraction see the subheading titled ROW AND COLUMN
EXTRACTION below.

The next menu is:

```
Please make an x by the bands you want extracted

_____ 1
_____ 2
_____ 3
_____ 4

    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL)
```

MSS imagery has 4 bands, but the user may want to extract only a subset of these bands. See the subheading in this entry titled ROW AND COLUMN EXTRACTION.

The user then is asked to enter the prefix/group for the band files to be created. This name will precede each band file extracted into GRASS. For example, if three bands are extracted the following three (raster) band files will result:

```
prefixname.1
prefixname.2
prefixname.3
```

Whatever prefix name is specified will also automatically become the name for the imagery group file being created. Each image (i.e., each run of *i.tape.mss*) should be given a unique prefix/group name.

The extraction process will begin by first skipping the number of specified files, advancing to the starting row, and then reading the tape. The percent completion of the extraction is displayed on the screen. If more than one tape is required to store the image, the program will pause and inform the user to mount the next tape.

The extracted (raster) band files will be listed as raster map layers available in the current MAPSET and may be displayed using the GRASS commands *d.display, d.rast* or *i.points*.

NOTES
After extracting an image from tape, the geographic region definition in the x,y coordinate LOCATION_name will be set based upon the extracted rows and columns from the tape. The relationship between the image rows and columns and the geographic coordinates of the region is discussed in the manual entry for imagery.

This program is interactive and requires no command line arguments.

ROW AND COLUMN EXTRACTION
The display options in GRASS allow the user to locate rows and columns on the digital image. If enough disk space is available, one band of an entire image, or one band of a portion of an image known to contain the area of interest, can be extracted and displayed. The measurements option in *d.display*, or *d.where* (following the use of *d.rast*) will echo x and y coordinates to the screen. (These coordinates will display negative numbers in the north-south direction, but ignoring the negative sign will yield the row number.) See the imagery manual entry for further explanation.

If a photograph of the digital image is available, the rows and columns to be extracted can be determined from it by associating inches with the total number of known rows and columns in the scene. For example, if the total length of the photograph is 12 inches, the total number of rows on the tape is 2000, and the northwest corner of the area of interest begins 2 inches from the top of the photo, then:

```
12" / 2000 rows = 2" / x rows
    x = 333.333
```

The northwest corner of the area of interest starts at row 333. The starting row, ending row, starting column, and ending column can be calculated in this manner.

SEE ALSO
d.display, d.rast, d.where, i.group, i.points, i.tape.mss.h, i.tape.other, i.tape.tm

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.tape.mss.h

NAME
i.tape.mss.h **-** An imagery function that extracts header information from LANDSAT Multispectral Scanner (MSS) imagery data stored on half-inch tape.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.tape.mss.h
i.tape.mss.h help
i.tape.mss.h tape_drive_name

DESCRIPTION
i.tape.mss.h reads the header information on a Multispectral Scanner (MSS) tape. This program reads the specified input file (the computer-compatible tape tape_drive_name), and by default displays the output to the user's terminal. The user may redirect output to a file by using the UNIX redirection mechanism. For example:

i.tape.mss.h /dev/rmt0 > h.out

The name of the tape drive depends on the computer being used.

This program can be run either non-interactively or interactively. The user can run the program by specifying program arguments on the command line. Alternately, the user can simply type *i.tape.mss.h* on the command line, without program arguments. In this event, the program will prompt the user to enter a tape device name using the standard user interface described in the manual entry for *parser*.

SEE ALSO
i.tape.mss

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.tape.other

NAME
i.tape.other **-** An imagery function that extracts scanned aerial imagery (NHAP, etc.) and satellite imagery (TM, SPOT, etc) from half-inch or 8mm tape.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.tape.other

DESCRIPTION
i.tape.other is a generic program that extracts imagery from tape using the tape description that is input by the user.

This program must be run in a LOCATION_name with a x,y coordinate system (i.e., a coordinate system with projection 0). For further information regarding this LOCATION_name refer to the manual entry for imagery.

The first prompt in *i.tape.other* asks the user for the tape device name. This is sometimes /dev/rmt0 (for a density of 1600), but this varies with each machine.

The next prompt is:

```
Please mount and load tape, then hit RETURN -->
```

IMAGE IDENTIFICATION MENU
The first menu in the program asks the user for information about the data.

```
Please enter the following information

    Tape Identification:    __

    Image Description: __

    Title for the Extracted Raster (Cell) Files:__

    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL)
```

TAPE LAYOUT MENU
The next menu asks for the physical layout of the tape.

```
  GENERIC TAPE EXTRACTION

tape layout
    _0_  number of tape files to be skipped
    _0_  number of records in the remaining files to be skipped
band files
    _0_  number of bands on the tape
data format
    ___  band sequential (BSQ)  |  mark one with an x
    ___  band interleaved (BIL) |
    _0_  if you select BSQ format and all the bands are in a single file,
    enter the total number of records in the file. Otherwise enter 0
    _0_  length (in bytes) of the longest record on the tape
    _1_  blocking factor of data in the file

  AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
  (OR <Ctrl-C> TO CANCEL)
```

number of tape files to be skipped
If there are files at the beginning of the tape which are not image data, they can be skipped. Sometimes information that comes with a tape will indicate the number of header files or records on the tape. The GRASS utility *m.examine.tape* will also provide this information. The record length is the number of columns in the image, while the number of records is the number of rows in the image (not always correct, see blocking factor of data in the file below). NHAP imagery and usually most scanned aerial imagery do not have tape header files, but this should be checked. TM imagery has one header file that contains imagery format of data files and parameters of data acquisition. SPOT imagery has two files that should be skipped on the first tape, and one file to be skipped on the second tape (of a two-tape set).

number of records in the remaining files to be skipped
If the files which contain the image begin with non- image data, these records can also be skipped. This is usually 0 for most data types. SPOT imagery stored in 1600bpi has one header record in the image file on each tape that should be skipped.

number of bands on the tape
Most aerial imagery have 3 bands, but satellite simulator data may have more. TM data has seven bands and SPOT has three bands as a standard, respectively. The total number of bands on the tape should be specified here, not just the number that will be extracted.

data format
The two formats that imagery data are most commonly stored in are called band interleaved format (BIL) and band sequential format (BSQ). In BIL format, each record on the tape contains one line for one band of data. If the data contains three bands, then the first five records will look like this:

```
band 1, line 1
band 2, line 1
band 3, line 1
band 1, line 2
band 2, line 2
```

In BSQ format, all lines of one band are stored together on a tape, followed by all lines of another band, followed by all lines of the next band, etc. These data are stored as if they were in a one band BIL format:

```
band 1, line 1
band 1, line 2
band 1, line 3
    .
    .
    .
band 2, line 1
band 2, line 2
    .
    .
band 2, line 156
band 2, line 157
```

Each pixel contains one byte and there is one line per record. BSQ format is the format that is usually created by optical scanning devices when they scan photographs, but not all digitized aerial imagery are stored in this format. The format of the data is usually written on the exterior of the tape; this should be checked.

length (in bytes) of the longest record on the tape

This must be set to the number of bytes in the longest data record. It is used to determine how large a buffer to use for reading the tape. This value can be obtained using *m.examine.tape*.

blocking factor of data in the file
This is the number of lines combining into one physical record on tape. It is usually one for most of imageries, i.e., one line per physical record. However, considering on data compressing and tape memory saving, sometimes more than one lines are combined into one physical record on the tape. This number may be written on the exterior of the tape, otherwise the user need to experiment on this number by running of *i.tape.other*.

BAND EXTRACTION MENU
The user is then asked to mark an x beside the bands to be extracted. See the subheading in this entry entitled ROW AND COLUMN EXTRACTION.

```
Please mark an x by the bands you want extracted

  _____1
  _____2
  _____3
  _____4


    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL)
```

PREFIX/GROUP name
The user is asked to enter the prefix/group for the (raster) band files to be created. This name will precede each band file extracted into GRASS. For example, if three bands are extracted the following three band files will result:

```
prefixname.1
prefixname.2
prefixname.3
```

The specified prefix name will also automatically become the name for the imagery group file being created. Each image (i.e., each run of *i.tape.other*) should be given a unique prefix/group name.

ROW AND COLUMN MENU
Finally, the starting row, ending row, starting column and ending column are required. This allows the user to extract any subset of the image from the tape.

EXTRACT

```
Please select desired tape window (geographic region definition) to extract

start row:_0_
end row:_0_
start col:_0_
end col:_0_


    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL)
```

The extraction process will begin by first skipping the number of specified files, advancing to the starting row, and then reading the tape. The percent completion of the extraction is displayed on the screen.

Following the extraction, the extracted band files will be listed as raster map layers available in the current MAPSET. These raster map layers may be displayed individually using the GRASS commands *d.display*, *d.rast*, or *i.points*.

NOTES
This program can be used for extraction of TM, SPOT and other types of data from tape; however, the user must supply information to the program on how the image data is laid out on the tape. For example, the image data may be padded with surrounding extra rows and/or columns; the user may wish to skip over these rows and columns and extract only the actual image data from the tape. *i.tape.other* does not know where image data actually begins on the tape; the user must tell the program what portion of the tape data to extract. Often, information on the orientation and layout of the image data on tape will be included on a printout accompanying any tape data received by the user; however, this may not always be the case. The user may need to experiment with various runs of *i.tape.other* before extracting the portions of an image actually desired.

After extracting an image from tape, the geographic region in the x,y coordinate LOCATION_name will be set, based upon the extracted rows and columns from the tape. The relationship between the image rows and columns and the coordinates bounding the geographic region is discussed in the imagery manual entry.

This program is interactive and requires no command line arguments.

ROW AND COLUMN EXTRACTION
The display options in GRASS allow the user to locate rows and columns on the digital image. If enough disk space is available, one band of an entire image, or one band of a portion of an image known to contain the area of interest, can be extracted and displayed. The measurements option in *d.display*, or *d.where* (following a run of *d.rast*) will echo x and y coordinates to the screen. (These coordinates will display negative numbers in the north-south direction but ignoring the negative sign will yield the row number. See the imagery manual entry for further explanation.)

If a photograph of the digital image is available, the rows and columns to be extracted can be determined from it by associating inches with the total number of known rows and columns in the scene. For example, if the total length of the photograph is 12 inches, the total number of rows on the tape is 2000, and the northwest corner of the area of interest begins 2 inches from the top of the photo, then:

```
12" / 2000 rows = 2" / x rows
x = 333.333
```

The northwest corner of the area of interest starts at row 333. The starting row, ending row, starting column, and ending column can be calculated in this manner.

SEE ALSO
d.display, d.rast, d.where, i.group, i.points, i.tape.mss, i.tape.mss.h, i.tape.tm, i.tape.tm.fast, imagery, m.examine.tape

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory
Tao Wen, University of Illinois at Urbana-Champaign, Illinois

i.tape.spot

NAME
i.tape.spot **-** An imagery function that extracts SPOT imagery from half-inch tape.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.tape.spot

DESCRIPTION
i.tape.spot is a program that extracts SPOT imagery from 9-track, half-inch tape.

This program must be run in a LOCATION_name with an x,y coordinate system (i.e., a coordinate system with projection 0). For further information regarding this LOCATION_name type refer to the imagery manual entry.

The first prompt in *i.tape.spot* asks the user for the tape device name. This is sometimes /dev/rmt0 (for a half-inch tape having a density of 1600 bpi), but this varies with each machine.

The next prompt is:

```
Mount SPOT tape and  hit RETURN -->
```

IMAGE IDENTIFICATION MENU
The first menu in the program asks the user for information about the data.

Please enter the following information

```
TAPE IDENTIFICATION:    __

IMAGE DESCRIPTION: __

TITLE FOR THE EXTRACTED CELL (RASTER) FILES:__

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)
```

This program automatically reads the satellite name, tape product code, instrument name, interleaving indicator, spectral mode, preprocessing level and work order number into the field for TAPE IDENTIFICATION. The mission, path, row, scene shift, scene center date and time, orientation, incidence, azimuth, elevation angle, and absolute calibration coefficients and offsets are automatically entered into the field for IMAGE DESCRIPTION. User can type in any other messages into the two sections and a description as the title of raster map layer. The second menu is:

```
SPOT IMAGE EXTRACT
Please select region of the image to extract

start row: 0______(1-3002)
end row: 0______(1-3002)

start col: 0______(1-3166)
end col: 0______(1-3166)
```

```
AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)
```

The numbers in parentheses are the total number of rows and columns on the tape including zeros (filler). This information and additional information can also be obtained by running the program *m.examine.tape*. *m.examine.tape* will read any tape and provide the user with the number of files on a tape, the number of records on a tape, and the record lengths. Any subset of the image on the tape may be extracted. For a discussion of row and column extraction see the subheading entitled ROW AND COLUMN EXTRACTION below.

The next menu is:

```
Please mark an x by the bands you want extracted

_____ 1
_____ 2
_____ 3

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)
```

SPOT imagery has three bands, but the user may want to extract a subset of these bands. See the subheading in this entry entitled ROW AND COLUMN EXTRACTION.

The user then is asked to enter the prefix/group name for the raster band files to be created. This name will precede each band file extracted into GRASS. For example, if two bands are extracted the following two band files will result:

```
prefixname.1
prefixname.2
```

The specified prefix name will also automatically become the name for the imagery group file being created. Each image or subset (i.e., each run of *i.tape.spot*) should be given a unique prefix/group name.

The extraction process will begin by first skipping a number of files which are not data or not requested, advancing to the first band requested, forwarding to the requested column, and then reading the data. After extracting the requested rows and columns from each band, the program creates support files for the raster band map layer. The percent completion of the extraction is displayed on the screen. Because sometimes SPOT imagery is very large and is stored in multiple tape sets, the program is designed to read image by pausing when the tape need to be changed and inform the user to mount and load next tape. The number of tapes required to store one scene depends on the number of bytes per inch (bpi) in which the data are stored.

The extracted band files will be listed as raster map layers available in the current MAPSET and may be displayed using the GRASS commands *d.display*, *d.rast*, or *i.points*.

ROW AND COLUMN EXTRACTION
The display options in GRASS allow the user to locate rows and columns on the digital image. If enough disk space is available, one band of an entire image or one band of a portion of an image known to contain the area of interest, can be extracted and displayed. The measurements option in *d.display*, or *d.where* (following use of *d.rast*) will echo x and y coordinates to the screen. (These coordinates will display negative numbers in the north-south direction, but ignoring the negative sign will yield the row number.) See the imagery manual entry for further explanation.

Each scene of a SPOT image contains filler on both the left and right sides of the quad. The user may want to identify the row and column numbers in order to exclude the filler. This filler will otherwise be extracted with the image and take up unnecessary disk space.

If a photograph of the digital image is available, the rows and columns to be extracted can be determined from it by associating inches with the total number of known rows and columns in the scene. For example, if the total length of the photograph is 12 inches, the total number of rows on the tape is 2000, and the northwest corner of the area of interest begins 2 inches from the top of the photo, then:

```
12" / 2000 rows = 2" / x rows
x = 333.333
```

The northwest corner of the area of interest starts at row 333. The starting row, ending row, starting column, and ending column can be calculated in this manner.

NOTES
After extracting an image from tape the geographic region definition in the x,y coordinate LOCATION_name will be set based upon the extracted rows and columns from the tape. The relationship between the image rows and columns and the coordinates of the geographic region is discussed in the imagery format manual entry.

This program is interactive and requires no command line arguments.

SEE ALSO
d.display, d.rast, d.where, i.group, i.points, i.tape.mss, i.tape.mss.h, i.tape.other, i.tape.tm, imagery, m.examine.tape

AUTHOR
Tao Wen, University of Illinois at Urbana-Champaign, Illinois

i.tape.tm

NAME
i.tape.tm **-** An imagery function that extracts LANDSAT Thematic Mapper (TM) imagery from half-inch tape.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.tape.tm

DESCRIPTION
i.tape.tm is a program that extracts LANDSAT Thematic Mapper (TM) imagery from half-inch tape.

This program must be run in a LOCATION_name with a x,y coordinate system (i.e., a coordinate system with projection 0). For further information regarding this LOCATION_name refer to the imagery manual entry.

The first prompt in *i.tape.tm* asks the user for the tape device name. This is sometimes /dev/rmt0 (for a half-inch tape having a density of 1600 bpi), but this varies with each machine.

The next prompt is:

```
Please mount and load tape, then hit RETURN -->
```

IMAGE IDENTIFICATION MENU
The first menu in the program asks the user for information about the data.

```
Please enter the following information

    Tape Identification:    __

    Image Description: __

    Title for the Extracted Raster (Cell) Files:__

    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL)
```

This program automatically enters the scene ID number into the field for Tape Identification. The mission, path, row, quadrant, date, and whether the image is corrected is automatically entered into the field for Image Description.

The second menu is:

```
THEMATIC MAPPER EXTRACT
Please select the desired tape window (geographic region definition) to extract

  first row: _______(1-2984)
  last row: _______(1-2984)

  first col: _______(1-4220)
  last col: _______(1-4220)
```

```
            AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
            (OR <Ctrl-C> TO CANCEL)
```

The numbers in parentheses are the total number of rows and columns on the tape including zeros (filler). This information and additional information can also be obtained by running the program *m.examine.tape*. *m.examine.tape* will read any tape and provide the user with the number of files on a tape, the number of records on a tape, and the record lengths. Any subset of the image on the tape may be extracted. For a discussion of row and column extraction see the subheading entitled ROW AND COLUMN EXTRACTION below.

The next menu is:

```
        Please make an x by the bands you want extracted

        _____ 1
        _____ 2
        _____ 3
        _____ 4
        _____ 5
        _____ 6
        _____ 7

            AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
            (OR <Ctrl-C> TO CANCEL)
```

TM imagery has 7 bands, but the user may want to extract only a subset of these bands. See the subheading in this entry entitled ROW AND COLUMN EXTRACTION.

The user then is asked to enter the prefix/group for the raster band files to be created. This name will precede each band file extracted into GRASS. For example, if three bands are extracted the following three band files will result:

```
         prefixname.1
         prefixname.2
         prefixname.3
```

The specified prefix name will also automatically become the name for the imagery group file being created. Each image or quad (i.e., each run of *i.tape.tm*) should be given a unique prefix/group name.

The extraction process will begin by first skipping the number of specified files, advancing to the first band requested, and then reading the tape. After extracting the requested rows and columns for each band, the program creates support files for the raster band map layer. The percent completion of the extraction is displayed on the screen. Because TM imagery is divided into four quads and is stored in multiple tape sets, the program is designed to read one quad at a time. The number of tapes required to store one quad depends on the number of bytes per inch in which the data is stored. If more than one tape is required to store one quad, the program will pause and inform the user to mount the next tape.

The extracted band files will be listed as raster map layers available in the current MAPSET and may be displayed using the GRASS commands *d.display*, *d.rast* or *i.points*.

NOTES

After extracting an image from tape the geographic region definition in the x,y coordinate LOCATION_name will be set based upon the extracted rows and columns from the tape. The relationship between the image rows and columns and the coordinates of the geographic region is discussed in the imagery manual entry.

This program is interactive and requires no command line arguments.

ROW AND COLUMN EXTRACTION

The display options in GRASS allow the user to locate rows and columns on the digital image. If enough disk space is available, one band of an entire image or, one band of a portion of an image known to contain the area of interest, can be extracted and displayed. The measurements option in *d.display*, or *d.where* (following use of *d.rast*) will echo x and y coordinates to the screen. (These coordinates will display negative numbers in the north-south direction, but ignoring the negative sign will yield the row number.) See the imagery manual entry for further explanation.

Each quad of a TM image contains filler on both the left and right sides of the quad. The user may want to identify the row and column numbers in order to exclude the filler. This filler will otherwise be extracted with the image and take up unnecessary disk space.

If a photograph of the digital image is available, the rows and columns to be extracted can be determined from it by associating inches with the total number of known rows and columns in the scene. For example, if the total length of the photograph is 12 inches, the total number of rows on the tape is 2000, and the northwest corner of the area of interest begins 2 inches from the top of the photo, then:

```
12" / 2000 rows = 2" / x rows
x = 333.333
```

The northwest corner of the area of interest starts at row 333. The starting row, ending row, starting column, and ending column can be calculated in this manner.

SEE ALSO

d.display, d.rast, d.where, i.group, i.points, i.tape.mss, i.tape.mss.h, i.tape.other, imagery, m.examine.tape

AUTHOR

Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.tape.tm.fast

NAME
i.tape.tm.fast **-** An imagery function that extracts Thematic Mapper (TM) imagery from tape media (GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.tape.tm.fast
i.tape.tm.fast help
i.tape.tm.fast [-q] input=name group=name bands=value[,value,...] [rows=firstrow-lastrow] [cols=firstcol-lastcol] [title=name]

DESCRIPTION
i.tape.tm.fast is a program that extracts TM imagery from tape media with different blocking factors (its value indicates how many rows are combined into one physical record on the tape).

i.tape.tm.fast must be run in a LOCATION_name with a (x,y) coordinate system (i.e., a coordinate system with projection 0). For further information regarding the LOCATION_name type, please refer to the imagery manual entry.

i.tape.tm.fast reads the blocking factor from the header file as well as other parameters, such as gains and offsets for each band, map projection, sun elevation and azimuth, etc., and writes into history file (depending upon the contents of the header file).

OPTIONS
This program can be run either non-interactively or interactively. It will be run non-interactively if the user specifies the name of input device, the name of output group file, bands to be extracted, and optionally other parameters (see below) on the command line using the form:

i.tape.tm.fast [-q] input=name group=name bands=value[,value,...] [rows=firstrow-lastrow] [cols=firstcol-lastcol] [title=name]

where the input should be the device name on which the tape media are mounted. The group is an imagery group that will store the extracted TM imagery. The bands value in a list separated by commas is the bands the user wants to extract from the imagery. The rows and cols represent the region which the user wishes to extract, where the default is whole imagery. At last an optional title is for information only. Alternatively, the program will be run interactively if the user types only *i.tape.tm.fast*; in this case the program will prompt the user for parameter values using the standard GRASS *parser* interface described in the manual entry for *parser*.

Flag:
-*q* Run quietly. Suppresses output of program percent-complete messages. If this flag is not used, these messages are printed out.

Parameters:

input=name	The name of the device on which the tape media containing the imagery files mounted.
group=name	The name of the group, which will store the imagery extracted from tape media.

bands=value[,value,...] The bands the user wishes to extracted from the tape media.
Options: 1-7

rows=firstrow-lastrow The values of first and last row of the extracting region.
Default: full imagery

cols=firstcol-lastcol The values of first and last column of the extracting region.
Default: full imagery

title=name The title of the extracting imagery.
Default: TM Imagery File Extracted from Tape

NOTES
Running in command line mode, *i.tape.tm.fast* will overwrite the group file and support files without prompting if the files existed.

SEE ALSO
i.group, i.tape.mss, i.tape.mss.h, i.tape.other, i.tape.tm, i.tape.spot, imagery m.examine.tape

AUTHOR
Tao Wen, University of Illinois at Urbana-Champaign, Illinois

i.target

NAME
i.target **-** An interactive imagery function that establishes a GRASS target location and mapset for an imagery group.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.target

DESCRIPTION
i.target targets an imagery group to a GRASS database location name and mapset. During the imagery program *i.rectify*, a location name and mapset are required into which to transfer the rectified file just prior to completion of the program; *i.target* enables the user to specify this location. *i.target* must be run before *i.points* and *i.rectify*.

The first prompt in the program asks the user for the name of the imagery group that needs a target. The imagery group must be present in the user's current mapset.

The following menu asking for the target LOCATION name and MAPSET is displayed:

```
Please select the target LOCATION and MAPSET for group <spot>

CURRENT LOCATION:  location______
CURRENT MAPSET:  demo___________

TARGET LOCATION:_________
TARGET MAPSET:___________

(Enter list for a list of location names or mapsets within a location)


    AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
    (OR <Ctrl-C> TO CANCEL
```

An imagery group may be targeted to any GRASS location.

NOTES
This program is interactive and requires no command line arguments.

SEE ALSO
i.group, i.points, i.rectify

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

i.texture

NAME
i.texture **-** calculate textural features on a raster file
(GRASS Image Processing Program)

GRASS VERSION
4.x,5.x

SYNOPSIS
i.texture
i.texture help
i.texture rast=name

DESCRIPTION
Reads a GRASS raster map as input. Calculates textural features based on spatial dependence matrices for north-south, east-west, northwest, and southwest directions using a 3x3 neighborhood (i.e., a distance of 1). Writes to standard output. Be sure to carefully set your resolution (using *g.region*) before running this program, or else your computer could run out of memory. Also, make sure that your raster map has no more than 255 categories.

OPTIONS
Parameter:

rast=name	Raster map name.

NOTES
Textural features include:

1. Angular Second Moment,
2. Contrast,
3. Correlation,
4. Variance,
5. Inverse Difference Moment,
6. Sum Average,
7. Sum Variance,
8. Sum Entropy,
9. Entropy,
10. Difference Variance,
11. Difference Entropy,
12. Information Measure of Correlation,
13. Another Information Measure of Correlation, and
14. Maximal Correlation Coefficient.

Algorithm taken from:
Haralick, R.M., K. Shanmugam, and I. Dinstein. 1973. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybertinetics, SMC-3(6):610-621.

The code was taken by permission from pgmtexture, part of PBMPLUS (Copyright 1991, Jef Poskanser and Texas Agricultural Experiment Station, employer for hire of James Darrell McCauley).

BUGS
The program can run incredibly slow for large raster files (larger than 64 x 64) and command line options are limited.

The method for finding (14) the maximal correlation coefficient, which requires finding the second largest eigenvalue of a matrix Q, does not always converge.

It would be interesting to write raster files to map features for neighborhoods, with some sort of quantization to record category values. This may be useful for image classification schemes, but this exercise is left to the reader (the changes would be fairly trivial).

REFERENCES

IEEE Transactions on Systems, Man, and Cybertinetics, SMC-3(6):610-621.

SEE ALSO

g.region, r.reclass

AUTHOR

James Darrell McCauley, Agricultural Engineering, Purdue University

i.vpoints

NAME
i.vpoints **-** Identifies coordinate pairs of points from a vector map or keyboard entry and corresponding points in an image.

GRASS VERSION
4.x, 5.x

DESCRIPTION
This program enables the user to identify coordinate pairs of points from a vector map or keyboard entry and corresponding points in an image to be rectified. The map coordinate values of each point are used to calculate a transformation matrix. The operator may then use the *i.rectify* or *i.rectify2* program to rectify the image using the transformation matrix coefficients calculated from the control point file created in *i.vpoints*. The *i.rectify* program performs only a first order transformation of the image whereas *i.rectify2* can rectify an image using either a first, second, or third order polynomial equation.

The first step is to display the unrectified image and corresponding vector map data. The operator would then mark corresponding control point locations on the image and map. To identify the precise location of a point to be marked, *i.vpoints* has a zoom option. In addition to marking control points on an image to be rectified and inputting their world coordinate values using the keyboard, *i.vpoints* has the option to simultaneously display vector map data available in the targeted database, and identify on the vector map the location of the corresponding marked points. When this option is chosen, the coordinate values are input automatically. Any GRASS map layer or vector map in the targeted database LOCATION can be displayed using *i.vpoints*. The *i.vpoints* program also has the capability of overlaying (i.e., warping) the vector data onto the raster image to visually check the accuracy of the registration based on the current set of active control points. During the process of marking points and entering map coordinates, the user can compute the RMS (root mean square) error for each point entered. The *i.vpoints* program does this by calculating a transformation equation (the same one that is calculated in the GRASS program *i.rectify2*). Coefficients are computed for the equation. The coefficients are then used in the equation along with the x,y coordinates of the marked points. The results are plugged into an equation for RMS error. The interpretation of RMS error is described in the ANALYZE subsection.

The procedures for marking control points (registration points), displaying vector map layers, overlaying vector maps onto the raster image, and calculating RMS error are described in the following sections.

To enter the program (the *i.vpoints* program requires the use of a graphics monitor) type *i.vpoints* . . .

GRASS **-** *GRID > i.vpoints*

the first prompt in the program asks for the imagery group to be registered . . .

```
Enter imagery group to be registered
Enter 'list' for a list of existing imagery groups
Enter 'list -f' for a verbose listing
Hit RETURN to cancel request
>
```

for example, >listmight produce the following response

```
<list>
Available groups
- - - - - - - - - - - - - - - - - - - - - - - - - - -
test
- - - - - - - - - - - - - - - - - - - - - - - - - - -
hit RETURN to continue -->
```

whereas, >list -f might produce the following response

```
<list -f>
Available groups
- - - - - - - - - - - - - - - - - - - - - - - - - - -
test
photo in PERMANENT
res2 in tifftest
- - - - - - - - - - - - - - - - - - - - - - - - - - -
hit RETURN to continue -->
```

The imagery group entered above should contain the files that you wish to rectify. After entering the group to be registered, the terminal screen displays the message:

```
>test
<test>
Use mouse now . . .
```

And the color graphics monitor displays the following screen:

Any single file in the imagery group may be used to mark points, and points can be marked on more than one file in the imagery group to accumulate the suggested minimum number of points (3 for a 1st order transformation, 6 for a 2nd order, and 10 for a 3rd order). Any file in the imagery group can be subsequently rectified (using *i.rectify2*) based on the transformation matrix computed from these points. The chosen file is displayed in the upper left quadrant of the monitor at a default magnification based on the extent of the current active window.

RASTER IMAGE

The raster image option on the menu at the bottom of the window allows the user to display any single file in the imagery group in the upper right quadrant of the window screen. The option provides the same file selection pick list as is presented when you first enter the *i.vpoints* program. When you select this option, the program will erase the data contained in all of the four quadrant windows and will reinitialize all program values.

VECTOR MAPS
The vector maps option on the menu at the bottom of the screen allows the user to display vector map data in the upper right quadrant of the screen. After selecting the vector map layer to display, a menu selection bar appears along the bottom on the screen. This pick list is used to select the line color (blue, gray, green, red, white, or yellow) for the selected vector data layer.

Refresh

The refresh option on the main menu allows the user to "refresh" or re-draw the displayed vector data. This function will erase all outlines showing the limits of previously zoomed areas. A "yes/no" prompt will appear:

```
Refresh Map ?  NO   YES
```

```
Zoom
```

To enlarge a raster or vector image, place the mouse cross hairs on the word zoom on the main menu and press the left button. The following menu will be displayed at the bottom. of the screen:

```
       CANCEL   BOX   POINT   Select type of zoom
```

You have the option to identify the map extent of the zoom window using either the mouse to define a box, or the mouse to mark a center point from which to enlarge the image. The box option first prompts you to identify a starting corner for the zoom region and then allows you to define the area to be zoomed using a rubber band box. The prompts appear as follows...

```
       CANCEL    Mark the first corner of region

       CANCEL    Define the region
```

After marking the first corner of the region to be enlarged, hold down the left button and move the mouse to change the size and shape of the rubber band box. After defining the area to be enlarged, press the right button to accept it.

The point method for enlarging an image will display a mouse menu to guide you in selecting the appropriate enlargement. To enlarge or reduce the magnification factor, place the cursor on the "+" or "-" box and press the left button on the mouse. You may zoom either the raster or the vector display.

Upon accepting the new region limits, the raster or vector data are redisplayed in either the lower left (raster) or lower right (vector) windows.

The extent of the zoomed area is outlined on the unzoomed image in the main window area.

While the main menu is displayed, you can mark corresponding control points on the raster and vector images or enter map coordinates from the keyboard. If you are using coordinates taken from a reference map, circle these points and then use whatever means you have available to identify as precisely as possible the coordinate values for these points. Digitizing software is recommended, especially GRASS 3,0 program digit/1/. Once you have determined the standard coordinates (for example, UTM's) of each circled point, you are ready to mark the points on the displayed image. To mark the points on the image, that correspond to the points on the standard coordinate map, place the mouse cross hairs on the point on the image to be marked (you will probably have to ZOOM to find the exact spot) and press the left hand button on the mouse. A diamond shaped symbol will be marked on the image. The text monitor will display the following screen:

```
       Point 1 marked on the image at
       East:  1023.77
       North:  -164.41

       Enter coordinates as east north:

       Analyze
```

After a number of points have been marked (a minimum of 4 for a 1st order transformation, 7 for a 2nd order, and 11 for a 3rd order), the RMS error of the points marked on the image can be checked. This is done by placing the cross hairs on the word ANALYZE on the main menu at the bottom of the monitor. The following error report is superimposed on the monitor:

```
       errorimage
       target
       #    col  row  target    east  north
       east
       north

       1    -0.9 0.0  1.0  1048.5    -144.8 679132.5
       4351080.6
       2    1.0  0.4  1.3  2153.1    -567.2 684314.7
```

```
4399001.4
.
.
.

Overall rms error  76.85
```

The RMS error for the image being rectified is recorded under the column "error" and subtitled "row' and "col". In the above report, the marked point number 1 is 0.0 rows and -0.9 columns from the predicted location calculated by the transformation equation. The RMS error for the target database map is recorded under the heading "error" and the subheading "target". This is the RMS error for the east and the north coordinate values of the target map, but it is represented in the table using one general value. The overall RMS error for the image is displayed at the bottom of the screen in meters. Points that generate a high RMS error are displayed in red on the monitor. The x,y coordinate values of the point marked on the image being rectified are recorded under the heading "image" and the subheadings "east" and "north". The standard coordinate values of the point in the target database are recorded under the heading "target" and the subheadings "east" and "north". If the user would like to exclude or include a point, this can be accomplished by placing the mouse cross hairs on the point number to be included (if the point is absent) or excluded (if the point is displayed) and then pressing the left button on the mouse twice. When a point is excluded, it is not included in the calculation of the RMS error, or included in the final transformation matrix. However, it can be retrieved within *i.vpoints* at any time by double clicking with the mouse as described above.

The following menu appears at the bottom of the monitor:

```
   DONE   PRINT   FILE   OVERLAY   DELETE ON
Transformation - ->   1st ORDER   Double click on point to be DELETED
```

Selecting DELETE ON will toggle the option to DELETE OFF, the toggle option is used to allow the user to physically remove a control point from the POINTS file instead of just flagging it as an non-active reference point.

Overlay allows the user to overlay the vector map(s) onto the raster image. Overlay can be used to warp (register) and display the selected vector file data on top of the raster image contained in the upper left window of the color screen. An inverse coordinate transformation is performed using the currently active order of transformation (i.e., first, second, or third).

```
Overlay vectors on raster image   NO   YES
```

By selecting the 1st ORDER option, the user may select the order of transformation desired:

```
Select order of transformation -->   1st Order   2nd Order   3rd Order
```

The program will immediately recalculate the RMSE and the number of points required.

To exit the *i.vpoints* program, place the mouse cross hairs on the word QUIT at the bottom of the monitor and all of the marked points (including coordinates) will be saved.

SEE ALSO
g.mapsets, i.group, i.points, i.rectify, i.rectify2, i.target

AUTHOR
William R. Enslin, Michigan State University Center for Remote Sensing

i.zc

NAME
i.zc - Zero-crossing "edge detection" raster function for image processing.
(GRASS Image Processing Program)

GRASS VERSION
4.x, 5.x

SYNOPSIS
i.zc
i.zc help
i.zc input_map=name zc_map=name [width=value] [threshold=value] [orientations=value]

DESCRIPTION
i.zc is an image processing program used for edge detection. The raster map produced shows the location of "boundaries" on the input map. Boundaries tend to be found in regions of changing cell values and tend to run perpendicular to the direction of the slope. The algorithm used for edge detection is one of the "zero-crossing" algorithms and is discussed briefly below.

This program will be run interactively if the user types *i.zc* without program arguments on the command line. In this event, the program will prompt the user for parameter values using the standard interface described in the manual entry for *parser*. Alternately, the user can run the program non- interactively by specifying program parameter values on the command line.

OPTIONS
Parameters:
input_map=name Name of input raster map layer.

zc_map=name Name of raster map layer to be used for zero-crossing values.

width=value This parameter determines the x-y extent of the Gaussian filter. The default value is 9; higher and lower values can be tested by the user. Increasing the width will result in finding "edges" representing more gradual changes in cell values.
Default: 9

threshold=value This parameter determines the "sensitivity" of the Gaussian filter. The default value is 10; higher and lower values can be tested by the user. Increasing the threshold value will result in fewer edges being found.
Default: 10

orientations=value This value is the number of azimuth directions the cells on the output raster map layer are categorized into (similar to the aspect raster map layer produced by the *r.slope.aspect* program). For example, a value of 16 would result in detected edges being categorized into one of 16 bins depending on the direction of the edge at that point.
Default: 1

The current region definition and mask settings are respected when reading the input map.

NOTES
The procedure to find the "edges" in the image is as follows: 1) The Fourier transform of the image is taken, 2) The Fourier transform of the Laplacian of a two-dimensional Gaussian function is used to filter the transformed image, 3) The result is run through an inverse Fourier transform, 4) The resulting image

is traversed in search of places where the image changes from positive to negative or from negative to positive, 5) Each cell in the map where the value crosses zero (with a change in value greater than the threshold value) is marked as an edge and an orientation is assigned to it. The resulting raster map layer is output.

SEE ALSO
i.fft, i.ifft, r.mapcalc, r.mfilter, r.slope.aspect, parser

AUTHOR
David Satnik, GIS Laboratory, Central Washington University

imagery

NAME
imagery **-** Description of GRASS image processing functions.

IMAGE PROCESSING IN GRASS
The following discussion is intended to provide a quick overview of image processing in GRASS. Some concepts and some hints are provided. For a more complete discussion and description of image processing in GRASS see GRASS Tutorial: Image Processing.

EXTRACTING IMAGERY DATA INTO A GRASS DATABASE
Remotely sensed images are captured for computer processing by filtering radiation emanating from the image into various electromagnetic wavelength bands, converting the overall intensity for each band to digital format, and storing the values on computer compatible media such as magnetic tape.

The GRASS programs which extract image data from magnetic tape can read LANDSAT multi-spectral scanner (MSS) data (*i.tape.mss*), LANDSAT thematic mapper (TM) data, (*i.tape.tm*), and other formats, such as scanned aerial photography or SPOT satellite data (*i.tape.other*). They extract the band data into raster files in a GRASS database. Each band becomes a separate raster file, with standard GRASS map layer support, and can be displayed and analyzed just like any other raster file.

UNREGISTERED DATA
The band data extracted from tapes are assumed to be unregistered data. This means that the GRASS software does not know the earth coordinates for pixels in the image. The only coordinates known at the time of extraction are the columns and the rows relative to the way the data was stored on the tape.

Data can only be extracted into a database, which has an x,y coordinate system, and not into a projected database (e.g., a UTM database). This is to prevent users from mixing the unregistered data with registered data in the same database. The GRASS system comes with the database imagery that is an x,y database. New databases can be created by users during GRASS startup. See the *g.help* section on "Setting Up a GRASS Database" for instructions on creating a new database.

CELL HEADERS
The cell headers for the band files in these x,y databases are set to reflect the rows and columns of the extracted data. The north-south values represent the rows, and the east-west values represent the columns. The resolution of the unregistered data is set to 1.

Note, however, that while the row numbers increase from 1 to n from north to south, GRASS requires that the values of the user's current geographic region decrease from north to south. The solution adopted was to represent the rows with negative values (i.e., -1 to -n). This allows them to decrease from north to south and, if the minus sign is ignored, to reflect the row numbers.

The cell headers for the layers in x,y databases are set so that the coordinates at the center of each pixel exactly reflect the row and column for that pixel. The northern edge is set to 0.5 less than the first row, the southern edge 0.5 larger than the last row, the west to 0.5 less than the first column, and the east to 0.5 larger than the last column. When the image is displayed on the graphics monitor, the *d.where* command can be used to report row and column values.

For example, suppose rows 100-500 and columns 200-800 are extracted. Then the cell headers for the extracted data will be given the following values:

```
north: -99.5
south: -500.5
west: 199.5
```

```
        east: 800.5
        ns res: 1.0
        ew res: 1.0
```

REGION AND MASK

Since the data layers are given essentially contrived cell headers, users must exercise extra care when analyzing or displaying unregistered images. It is very easy for the user's GRASS region to have absolutely no relationship to the data he is trying to display. This could happen when the region is set for data extracted from one tape, but the analysis is attempted on data extracted from another tape. A good habit to develop is to set the region to exactly match one of the band files. This can be done using the GRASS *g.region* command.

Another pitfall is to have a mask set to a band file from one data set while trying to read another. Even if the region is set properly, the data will appear to be all no- data since the mask will effectively knock out any data. Be sure that the mask is either set to a related data layer or not set at all. See *r.mask* for information on setting and unsetting the mask.

Please note that the tape extraction routines set your database region to match the rows and columns of the data that is extracted.

GROUPS

Since the band files are individual raster files, it is necessary to have a mechanism to maintain a relationship between band files from the same image as well as raster files derived from the band files. The GRASS group data structure accomplishes this goal. The group is essentially a list of names of raster files that belong in the group. For user convenience, groups are also created (and updated) by the tape extraction routines. The tape extraction programs ask the user to supply a group name as well as to specify the bands to be extracted. Suppose that the user extracts bands 1, 2, and 3 into a group called nhap. Then the band files will become the raster files nhap.1, nhap.2, and nhap.3 and the group nhap will list these 3 raster files as members of the group.

Groups can also be created and modified by the user using the GRASS command *i.group*.

IMAGE REGISTRATION AND RECTIFICATION

Image registration and image rectification is the process of associating earth coordinates with pixels on the image and then converting the unregistered raster files to raster files in a projected database.

Image registration (*i.points*) is applied to a group, rather than to individual raster files. The control points are stored in the group, allowing all related band files to be registered in one step rather than individually.

Image rectification (*i.rectify*) is applied to individual raster files, with the control points for the group used to control the rectification and the group target (*i.target*) used to specify the database where the rectified layer will live.

IMAGE CLASSIFICATION

Image classification methods process all or a subset of the band files as a unit. Spectral signatures for the image are generated (*i.cluster*) and then used to produce a landcover map (*i.maxlik*).

The signatures must be associated only with the raster files actually used in the analysis. This means that with a group subgroups must be created (*i.group*) which list the band files to be grouped for classification purposes. The signatures are stored with the subgroup.

Note that multiple subgroups can be created within a group. This allows different classifications to be run with different combinations of band files. Also note that raster files produced by the classification process (*i.maxlik*) are automatically listed as part of the group.

RECTIFIED VS. UNRECTIFIED ANALYSIS

There are two possible routes for processing image data. The first is to register the group (*i.points*), perform the analyses on the unrectified band data (*i.maxlik*), and then rectify the results (*i.rectify*). The second is to register the group (*i.points*), rectify the band data (*i.rectify*), then run analyses on the rectified band data in the target location (*i.rectify*). Both routes are permissible in GRASS. Users will most likely prefer the first. The second route requires leaving GRASS and re-running GRASS under the target location. It also will require that a group be created to hold the rectified band files since *i.rectify* does not create or modify groups. Also, spatial filtering may not be as effective on rectified data since the rectification of the data requires resampling the original data.

SEE ALSO

GRASS Tutorial: Image Processing, d.where, g.region, i.cluster, i.group, i.maxlik, i.points, i.rectify, i.tape.other, *i.tape.mss*, *i.tape.tm*, *i.target*, *r.mask*

AUTHOR

Michael Shapiro, U.S. Army Construction Engineering Research Laboratory