

Procesamiento Digital de Imágenes en **GRASS**

Alejandro Vivas Perdomo
phi_software@users.sourceforge.net
Iván Herrera

21 de mayo de 2005

Índice general

| | |
|--|----------|
| 1. Introducción a GRASS | 1 |
| 1.1. Qué es GRASS | 1 |
| 1.2. Instalación | 1 |
| 1.2.1. Requerimientos | 1 |
| 1.2.2. Instalación a Partir de un Binario | 1 |
| 1.2.3. Instalación a partir del Código Fuente | 2 |
| 1.2.4. Ventajas y Desventajas de Cada Una de las Instalaciones | 3 |
| 1.3. Bases de Datos | 4 |
| 1.3.1. Región | 4 |
| 2. Comenzando en GRASS | 7 |
| 2.1. Creación Base de Datos | 7 |
| 2.2. Obteniendo Ayuda Sobre los Comandos de GRASS | 10 |
| 2.2.1. El comando <code>g.manual</code> | 10 |
| 2.3. Importando y Exportando | 10 |
| 2.3.1. Importando imágenes en formato TIFF | 10 |
| 2.3.2. Importando Imágenes en Formato PNG | 10 |
| 2.3.3. Importando Imágenes en Formato ERDAS <code>.lan</code> | 11 |
| 2.3.4. Importando Imágenes con el comando <code>r.in.gdal</code> | 11 |
| 2.3.5. Ejemplo de Como Importar una Imagen | 12 |
| 2.4. Exportando | 13 |
| 2.4.1. Formato TIFF | 13 |
| 2.4.2. Formato PNG | 13 |
| 2.4.3. Sin formato (binario) | 13 |
| 2.4.4. <code>r.out.bil</code> | 13 |
| 2.4.5. <code>r.out.elas</code> | 13 |
| 2.4.6. <code>r.out.ppm3</code> | 13 |
| 2.4.7. <code>r.out.ascii</code> | 13 |
| 2.4.8. <code>r.out.ppm</code> | 13 |
| 2.4.9. Erdas Lan | 13 |
| 2.4.10. <code>r.out.gdal</code> | 13 |
| 2.5. Después de Importar | 13 |
| 2.5.1. Actualización de la Region | 13 |
| 2.5.2. Uso del Comando <code>r.support</code> | 15 |
| 2.5.3. El Comando <code>g.list</code> | 17 |
| 2.5.4. Grupos y Subgrupos de imagenes en GRASS | 17 |
| 2.6. Visualización de Imágenes | 19 |
| 2.6.1. Lanzar monitores con <code>d.mon</code> | 19 |
| 2.6.2. Visualización de rasters con <code>d.rast</code> | 20 |
| 2.6.3. Visualización de Composiciones a Color con <code>d.rgb</code> y <code>i.colors</code> | 20 |
| 2.6.4. Manejo de marcos en un monitor <code>d.frame</code> | 21 |
| 2.6.5. Creación de Composiciones a Color | 23 |

| | |
|---|-----------|
| 2.7. Mascaras | 24 |
| 3. Procesamiento de Imágenes | 25 |
| 3.1. Georreferenciación | 25 |
| 3.1.1. <code>i.target</code> | 25 |
| 3.1.2. <code>i.points</code> | 25 |
| 3.1.3. <code>i.rectify</code> | 26 |
| 3.1.4. Ejemplo | 26 |
| 3.2. Transformaciones | 36 |
| 3.2.1. Transformada RGB-HIS y HIS-RGB | 36 |
| 3.2.2. Transformada RGB-HSV y HSV-RGB | 37 |
| 3.2.3. Transformada de Fourier | 39 |
| 3.2.4. Componentes Principales | 40 |
| 3.2.5. Transformación TassledCap | 40 |
| 3.3. Clasificación de Imágenes | 40 |
| 3.3.1. Clasificación No Supervisada | 40 |
| 3.3.2. Clasificación Supervisada | 43 |

Índice de cuadros

| | |
|--|----|
| 2.1. Formatos raster soportados por GRASS | 11 |
|--|----|

Índice de figuras

| | |
|--|----|
| 1.1. Jerarquía de Directorios de las Bases de Datos en GRASS | 4 |
| 1.2. Ejemplo Gráfico de una Región | 5 |
| 2.1. Monitor | 20 |
| 2.2. Monitor desplegando una imagen con el comando <code>d.rast map=pan</code> | 21 |
| 2.3. Monitor desplegando una imagen con el comando <code>d.rgb</code> | 21 |
| 2.4. Definiendo la esquina izquierda abajo de un marco | 22 |
| 2.5. Definiendo la esquina derecha arriba de un marco | 23 |
| 2.6. Imagen pancromatica desplegada en un marco | 23 |
| 2.7. Imagen pancromatica y multiespectral desplegadas en un marco | 24 |
| 3.1. Monitor permitiendo digitar un punto de control | 33 |
| 3.2. Monitor mostrando resultados del calculo del polinomio | 34 |
| 3.3. Transformada HIS | 37 |
| 3.4. Transformada HSV | 38 |
| 3.5. Parte Real e Imaginaria de la Transformada de Fourier de la Imagen Pancromatica | 39 |

Agradecimientos

Alejandro Vivas quiere agradecer especialmente al Profesor de la Universidad Distrital “Francisco José De Caldas” Iván Lizarazo quien motivo la realización de este documento. También deseo agradecer especialmente a Constanza Diaz quien reviso varias veces el documento, sugirió importantes modificaciones y ayudo en el proceso de edición. Samuel Mesa, quien colaboro en la publicación de este documento en Internet.

Sobre el documento

Este documento se asume que el usuario esta trabajando con una versión de **GRASS** 5.0 ó superior, sin embargo si se esta trabajando con versiones inferiores este documento puede ser aún de utilidad. Este documento esta aún en desarrollo por lo cual puede contener algunas imprecisiones y capítulos y secciones sin terminar.

Convenciones Tipográficas

- El texto que encontremos así:

```
$ digite
```

es texto sobre una consola, parte de este texto debe digitarse en una consola **UN*X**.

```
GRASS: >digite
```

es texto sobre una consola en **GRASS** .

- El texto que encontremos con letra **nombre_de_un_archivo** indican archivos.
- Las teclas que deben oprimirse se indican con `<TECLA>` .
- Si el texto aparece de esta forma www.phi.sourceforge.net No estamos refiriendo a una pagina web.
- El texto que aparece precedido por el signo de dolar \$VARIABLE corresponde a una variable de entorno o a una variable que puede tomar cualquier valor.
- El texto que este escrito con letra itálica *importante* indica alguna palabra clave.

Como Leer Este Documento

Este documento esta diseñado para que la primera y la segunda parte sen leídas linealmente, la tercera parte puede ser leída saltado de sección en sección según el usuario lo desee.

Historia del Manual

Desde que inicio la redacción de este manual a su estado actual han surgido algunos cambios, la primera redacción de este documento se baso principalmente en la version 5 de **GRASS** , con la liberación de la version 6 de **GRASS** grandes cambios se han realizado sobre el software, por lo cual este documento ha tenido que ser reescrito para ser compatible con esta nueva version, es por esto que algunos comandos que se explican en este Manual o han desaparecido o se han modificado en su forma de ejecutar. En la actualidad se esta reescribiendo este documento con el fin de ser compatible tanto con la version 5 como con la version 6 de **GRASS** .

El autor agradece enormemente recomendaciones y correcciones que sirvan para obtener un documento de mejor calidad.

Este documento puede ser copiado, distribuido y/o modificado bajo los términos de la GNU Free Documentation License, Versión 1.2 o cualquiera de las versiones posteriores publicadas por la Free Software Foundation (véase <http://www.gnu.org/licenses/fdl.html>).

Capítulo 1

Introducción a GRASS

En este capítulo se explica los conceptos básicos para poder usar **GRASS** .

1.1. Qué es GRASS

GRASS es un software para Sistemas de Información Geográfica, que fue desarrollado hasta la versión 4 en U.S. Army Construction Engineering Research Laboratory, siendo hoy en día mantenido por la Universidad de Baylor en Waco, Texas y la Universidad de Hannover en Hannover, Alemania, es por esto que no se puede pensar en **GRASS** como software especializado en procesamiento de imágenes. Esto no quiere decir que **GRASS** no posea muchas y muy buenas herramientas para procesar imágenes, sino mas bien es un software con otras perspectivas sobre el procesamiento de imágenes. Ya que **GRASS** ha sido y sigue siendo un herramienta desarrollada por científicos, encontramos siempre a **GRASS** sobre plataformas **UN*X** (puede compilarse en Win32 sobre el CygWin), por lo cual recomiendo antes de comenzar a trabajar en **GRASS** aprender las bases de **UN*X**. **GRASS** además es un herramienta de Software Libre, es decir no esta bajo el control absoluto de ninguna empresa, lo cual la hace mucho mas flexible que otras herramientas.

1.2. Instalación

1.2.1. Requerimientos

Antes de poder instalar **GRASS** debemos tener acceso a un ordenador que ejecuta **UN*X**, preferiblemente **LINUX**, si no es así, pero dispone de una máquina en **Win32** deberá instalar **CygWin** ¹ con anterioridad.

1.2.2. Instalación a Partir de un Binario

Para instalar **GRASS** a partir de un binario si esta ejecutando **LINUX** debemos descargar los siguientes archivos:

- `grassVERSION_i686-pc-linux-gnu_install.sh`
- `grassVERSION_i686-pc-linux-gnu_bin.tar.gz`

Donde VERSION corresponde al numero de la versión de **GRASS** . Una vez tengamos los archivos debemos ejecutar en un shell lo siguiente:

```
$grassVERSION_i686-pc-linux-gnu_install.sh grassVERSION_i686-pc-linux-gnu_bin.tar.gz
```

¹**CygWin** es un entorno **UN*X** que permite que programas diseñado para **UN*X** se ejecuten en **Win32**, esta disponible www.cygwin.com.

1.2.3. Instalación a partir del Código Fuente

Para instalar **GRASS** a partir del código fuente necesitamos antes descargar las siguientes librerías:

Programas Requeridos

- Compilador de C (cc, gcc, egcs, ...)
gcc: <http://www.gnu.org/software/gcc/gcc.html>
- GNU make , recomendado
gmake : <http://www.gnu.org/software/make/make.html>
- flex, analizador lexicográfico (lex no es totalmente soportado, mejor usar flex)
flex: <http://www.gnu.org/software/flex/flex.html>
- yacc ó bison
bison: <http://www.gnu.org/software/bison/bison.html>
- X11 window, se requieren las herramientas de desarrollo
xfree86: <http://www.xfree.org>

Librerías Requeridas

- zlib, usada por **GRASS** internamente para comprimir archivos
libz: <http://www.gzip.org/zlib>
- libncurses4.x/5.x
ncurses: <http://www.gnu.org/software/ncurses/ncurses.html>
- PROJ4
<http://www.remotesensing.org/proj>

Librerías Opcionales

- fftw Fast Fourier Transform in the West
www.fftw.org
- gdal Geospatial Data Abstraction Library
<http://www.remotesensing.org/gdal/index.html>
- Postgresql Manejador de base de datos SQL
<http://www.postgresql.org/>
- ODBC
<http://www.unixodbc.org>
- Tcl/Tk 8.x
<http://tcl.sourceforge.net>
- Mesa3d
<http://mesa3d.sourceforge.net>
- Compilador de C++
gcc : <http://www.gnu.org/software/gcc/gcc.html>
- libpng
<http://www.libpng.org/pub/png/libpng.html>
- libjpeg
<http://www.ijg.org>

- libtiff
<http://www.libtiff.org>
- mySQL
<http://www.mysql.org>
- Motif ó Lesstif
<http://www.lesstif.org>
- R lenguaje
<http://cran.r-project.org>
- FreeType2
<http://www.freetype.org>

La mayoría de las librerías requeridas se encuentran en cualquier distribución de **LINUX**, si no es así o esta trabajando en otro sabor de **UN*X**, deberá descargarlas de Internet. Para la instalación de cada una de las librerías ver la documentación suministrada por cada uno de los paquetes.

Una vez instaladas las librerías necesarias, debemos descargar el código fuente de **GRASS** , el cual esta disponible en <http://www.baylor.edu/grass/> o en <http://grass.itc.it> , el archivo debe tener un nombre similar `grass-VERSION_src.tar.gz` , una vez hecho esto debemos descomprimirlo así:

```
$ tar xzf grass-VERSION_src.tar.gz
```

Donde nos quedara un directorio llamado `grass-VERSION`, a continuación nos ubicamos en el directorio así:

```
$cd grass-VERSION
```

A continuación debemos configurar **GRASS** , esto lo hacemos así:

```
$/configure
```

El siguiente paso es compilar:

```
$make
```

Esto puede tardar de minutos a horas según la velocidad de la máquina.
Para instalar debemos hacer:

```
$su  
#make install
```

1.2.4. Ventajas y Desventajas de Cada Una de las Instalaciones

Una de las desventajas de realizar una instalación binaria es que los binarios no están optimizados para la máquina, lo cual si podemos realizar en una instalación a partir del código fuente. Otra desventaja de una instalación binaria es que en los sitios oficiales de **GRASS** existen binarios para muy pocas plataformas.

Una desventaja de una instalación a partir del código fuente es el tiempo de compilación, ya que esto puede tardar bastante tiempo, según la máquina que tengamos, además al realizar la instalación a partir del código fuente hay que instalar todas las dependencias, lo cual lleva bastante tiempo.

1.3. Bases de Datos en GRASS

Como dije anteriormente, **GRASS** es un software para Sistemas de Información Geográfica, y como tal las bases de datos juegan un papel muy importante. Es por esto que debemos conocer los conceptos básicos de las bases de datos en **GRASS** para entender el funcionamiento del modulo de procesamiento de imágenes. **GRASS** organiza las bases de datos en una estructura jerárquica de directorios, como la que se muestra en el siguiente gráfico:

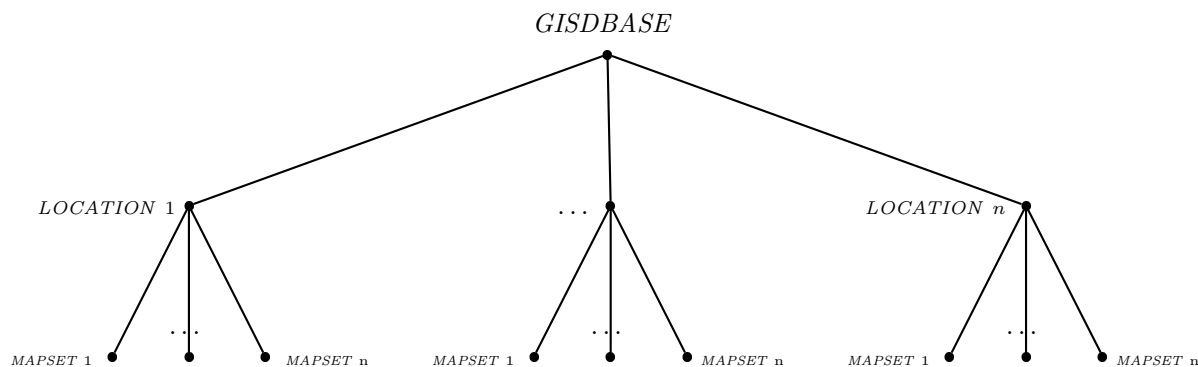


Figura 1.1: Jerarquía de Directorios de las Bases de Datos en **GRASS**

GISDBASE

El directorio en el nivel mas alto es conocido *GISDBASE* , este directorio es el encargado de contener las distintas bases de datos y es necesario especificar este directorio en el momento de ingresar a **GRASS** .

LOCATION

Los directorios que se encuentran debajo de *GISDBASE* son conocidos como *LOCATION* , los *LOCATION* son bases de datos independientes y son modificadas en el momento de trabajar e **GRASS** , el usuario debe seleccionar un *LOCATION* en el momento del ingreso a **GRASS** . La principal característica que tiene un *LOCATION* es que la información almacenada en ellos esta referida a un sistema de referencia fijo.

MAPSET

Los directorios debajo de de los *LOCATIONS* son conocidos como *MAPSETS*, el usuario también debe seleccionar un *MAPSET* en el momento de entrar a **GRASS** , las modificaciones a la base de datos son únicamente realizadas al *MAPSET* actual.

Una explicación mas detallada de las bases de datos en **GRASS** se puede leer en [1]

1.3.1. Región

El concepto de *región* es muy importante en **GRASS** , una región es una zona rectangular sobre la cual actúan todos los comandos de **GRASS** . Esta capacidad nos permite limitar la zona de interés sobre nuestros datos. La región es propia de un *LOCATION* y puede ser cambiada por el usuario cuando lo estime mediante el comando `g.region` el cual veremos mas adelante. Una *región* se define a partir de cuatro coordenadas:

1. Norte : Máximo valor que puede tomar la coordenada en y ó ϕ
2. Sur : Mínimo valor que puede tomar la coordenada en y ó ϕ
3. Este : Máximo valor que puede tomar la coordenada en x ó λ
4. Oeste : Mínimo valor que puede tomar la coordenada en x ó λ

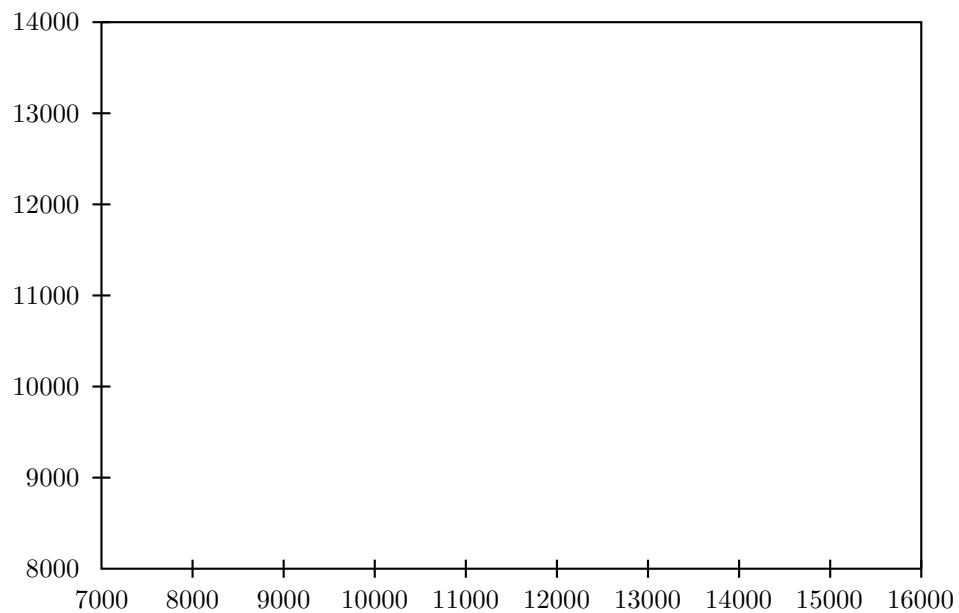


Figura 1.2: Ejemplo Gráfico de una Región

Gráficamente lo podemos ver con este ejemplo :

En este ejemplo Sur = 8000 , Norte = 1400, Este = 16000, Oeste = 7000

Capítulo 2

Comenzando en GRASS

En este capítulo se explican como crear una base de datos en **GRASS**, como importar imagenes a la base de datos, así como también la configuración que debe realizarse posterior a importar las imagenes. Una vez que el lector termine de leer este capítulo deberá estar en capacidad de crear una base de datos, estar familiarizado con el entorno de **GRASS**, además deberá ser capaz de desplegar imagenes.

La sección 2.1 explica como crear una base de datos en **GRASS**, para realizar esto el lector debe tener claro conceptos *GISDBASE*, *LOCATION*, *MAPSET*. También el lector deberá tener claro conceptos como proyección cartográfica, Latitud y Longitud.

2.1. Como Crear una Base de Datos en GRASS

El primer paso para crear una base de datos en **GRASS** es crear el directorio *GISDBASE*, el cual almacenara la base de datos, esto se realiza con el comando `mkdir`.

```
$ mkdir $MYDATABASE
```

Donde `$MYDATABASE` es la ruta completa al directorio que se creara. A continuación se debe ingresar a **GRASS** con el comando `grassX` donde X indica la versión de **GRASS**.

```
$grassX
```

Seguido de esto **GRASS** desplegara una pantalla similar a esta.

```
GRASS 5.0.0pre1

LOCATION: This is the name of an available geographic location. -spearfish-
         is the sample data base for which all tutorials are written.

MAPSET:  Every GRASS session runs under the name of a MAPSET.  Associated
         with each MAPSET is a rectangular COORDINATE REGION and a list
         of any new maps created.

DATABASE: This is the unix directory containing the geographic databases

         The REGION defaults to the entire area of the chosen LOCATION.
         You may change it later with the command: g.region
-----
LOCATION:  MYLOCATION____ (enter list for a list of locations)
MAPSET:  usuario_______ (or mapsets within a location)

DATABASE: /home/usuario/MYDATABASE_____
AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
         (OR <Ctrl-C> TO CANCEL)
```

En esta pantalla se debe ingresar en DATABASE la ruta completa al directorio que acabamos de crear es decir \$MYDATABASE, para este ejemplo se ha supuesto que MYDATABASE=/home/usuario/MYDATABASE , en LOCATION y en MAPSET el nombre de los correspondientes directorios que GRASS creará. Seguido de esto se debe oprimir < ESC > < ENTER >, donde a continuación GRASS lanzara un mensaje de error diciendo que el LOCATION no existe y preguntara si desea crearlo.

```
LOCATION <MYLOCATION> - doesn't exist

Available locations:
%-----
%-----
Would you like to create location <MYLOCATION> ? (y/n)
```

Para crear el LOCATION se debe digitar < y >. Hecho esto GRASS crea el nuevo directorio y preguntara si se conoce información sobre el sistema de coordenadas del LOCATION.

```
To create a new LOCATION, you will need the following information:

1. The coordinate system for the database
   x,y (for imagery and other unreferenced data)
   Latitude-Longitude
   UTM
   Other Projection
2. The zone for the UTM database
   and all the necessary parameters for projections other than
   Latitude-Longitude, x,y, and UTM
3. The coordinates of the area to become the default region
   and the grid resolution of this region
4. A short, one-line description or title for the location

Do you have all this information for location <MYLOCATION> ? (y/n)
```

Aceptamos digitando < y >.

```
Please specify the coordinate system for location <MYLOCATION>

A  x,y
B  Latitude-Longitude
C  UTM
D  Other Projection
RETURN to cancel

>A
```

GRASS a continuación solicita el sistema de coordenadas, como se desea trabajar con imágenes, se debe escoger la primera opción digitando A, y confirmamos la opción deseada al oprimir < ENTER >.

```
x,y coordinate system? (y/n) [y]
```

Paso siguiente GRASS solicita información que describa el location, por ejemplo este ejemplo podemos ingresar Location con una Imagen Spot Bogotá

```
Please enter a one line description for location <MYLOCATION>

> Location con una Imagen Spot Bogot\'a
```

Oprimir < ENTER > para aceptar.

```

Please enter a one line description for location <MYLOCATION>

> Location con una Imagen Spot Bogot\'a
=====
Location con una Imagen Spot Bogota
=====
ok? (y/n) [y]

```

A continuación **GRASS** solicita la información sobre la región, (la región es la zona en la cual trabaja **GRASS** ver sección 1.3.1). En este punto se pueden tomar dos opciones, la primera indicar el norte como 0 y sur como $N - 1$, donde N es el número de filas de la imagen y oeste como 0 y este como $M - 1$, siendo M el número de columnas de la imagen. La segunda opción es tomar las dejadas por defecto y actualizar la región después de importar las imágenes. La segunda opción es la más práctica. La grilla de resolución debe dejarse en uno, esto quiere decir que se crearan $N - 1$ filas, y $M - 1$ columnas.

```

DEFINE THE DEFAULT REGION

===== DEFAULT REGION =====
| NORTH EDGE:1_____ |
|                       |
WEST EDGE |                       | EAST EDGE
0_____ |                       | 1_____
| SOUTH EDGE:0_____ |
|                       |
=====

PROJECTION: 0 (x,y)                ZONE: 0

GRID RESOLUTION
    East-West:    1_____
    North-South:  1_____

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

```

Culminado este paso se debe oprimir $\langle ESC \rangle$ $\langle ENTER \rangle$. **GRASS** nos mostrará a continuación la información sobre la región y confirmara la región, para aceptar $\langle ENTER \rangle$.

```

projection:  0 (x,y)
zone:        0
north:       1
south:       0
east:        1
west:        0

e-w res:     1
n-s res:     1

total rows:          1
total cols:          1
total cells:         1

Do you accept this region? (y/n) [y] >

```

Habiendo terminado este paso **GRASS** nos muestra que habrá creado el *LOCATION*

```

LOCATION <MYLOCATION> created!
Hit RETURN -->

```

Se debe a continuación oprimir $\langle ENTER \rangle$, en donde se puede ver los valores del *DATABASE*, *LOCATION* y *MAPSET* trabajados anteriormente. Solo queda oprimir nuevamente $\langle EXC \rangle$ $\langle ENTER \rangle$ para terminar e ingresar inmediatamente a **GRASS**.

```

Welcome to GRASS 5.0.0pre1 (May 2001)

Geographic Resources Analysis Support System (GRASS) is Copyright,
1999-2001 by the GRASS Development Team, and licensed under terms of the
GNU General Public License (GPL).

This GRASS 5.0.0pre1 release is coordinated and produced by the
GRASS Development Team headquartered at University of Hannover with
worldwide support and further development sites located at Baylor
University and the University of Illinois.
This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

This version running thru the Bash Shell (/bin/bash)
Help is available with the command:          g.help
See the licence terms with:                 g.version -c
Start the graphical user interface with: tcltkgrass&
When ready to quit enter:                   exit
GRASS:~ >

```

Arriba se puede ver que **GRASS** esta corriendo sobre Bash, ¿Que quiere decir esto?, **GRASS** es una colección de ejecutables que deben ser llamadas sobre un interprete de comandos, en este caso Bash, cuando inicia **GRASS** lo único que el hace es alterar unas variables de entorno que permite acceder a estos ejecutables. Habiendo terminado felizmente estos pasos se puede proceder a importar las imágenes como veremos en el siguiente capitulo.

2.2. Obteniendo Ayuda Sobre los Comandos de GRASS

2.2.1. El comando g.manual

2.3. Importando y Exportando Imágenes

El primer paso al trabajar imágenes en **GRASS** es importar las imágenes, **GRASS** es capaz de importar imágenes en los siguientes formatos:

2.3.1. Importando imágenes en formato TIFF

Para importar una imagen **TIFF** es necesario utilizar el comando `r.in.tiff` el cual tiene la siguiente sintaxis :

```
r.in.tiff [-v] input=PATH output=PREFIX
```

Donde `PATH` indica la ruta en donde se encuentra la imagen y `PREFIX` el prefijo del nombre con el cual serán almacenadas las imágenes y lo que encontramos en `[]` son las opciones.

- `-v` Modo verboso.

2.3.2. Importando Imágenes en Formato PNG

Para importar una imagen **PNG** es necesario utilizar el comando `r.in.png` el cual tiene la siguiente sintaxis :

```
r.in.png input=PATH output=PREFIX
```

Donde `PATH` y `PREFIX` tienen el mismo significado visto anteriormente.

| Formato | Georreferenciación | Comando |
|---------------------------------|--------------------|------------------------------|
| Tiff/GeoTiff | Si | r.in.gdal / r.in.tiff |
| Erdas Imagine .hfa (.img) | No | r.in.gdal |
| Erdas .lan | | '' |
| Arc/Info Binary Grid | Si | '' |
| CEOS (Spot for instance) | No | '' |
| First Generation USGS DOQ | Si | '' |
| New Labelled USGS DOQ | Si | '' |
| Military Elevation Data (DTED) | Si | '' |
| Eosat Fast Format | No | '' |
| ESRI .hdr Labelled | Si | '' |
| TIFF / GeoTIFF | Si | '' |
| Erdas Imagine .hfa | No | '' |
| JPEG JFIF | No | '' |
| PCI .aux Labelled | Si | '' |
| Portable Network Graphics (PNG) | No | '' |
| USGS SDTS DEM | Si | '' |
| SAR CEOS | No | '' |
| JPEG/ JFIF | No | '' |
| Portable Network Graphics (PNG) | Si | r.in.gdal / r.in.png |
| PCI .aux Labelled | Si | i.in.erdas |
| PPM | | r.in.ppm |

Cuadro 2.1: Formatos raster soportados por **GRASS**

2.3.3. Importando Imágenes en Formato ERDAS .lan

Para importar una imagen Erdas .lan es necesario utilizar el comando `r.in.erdas` el cual tiene la siguiente sintaxis :

```
i.in.erdas input=PATH output=PREFIX trailer=PATH2
```

Donde `PATH` y `PREFIX` tienen el mismo significado visto anteriormente , `PATH2` es la ruta del archivo tráiler.

2.3.4. Importando Imágenes con el comando `r.in.gdal`

Este comando es uno de los mas útiles par importar imágenes, ya que soporta gran cantidad de formatos como se puede ver en la tabla (2.1). La sintaxis del comando es la siguiente:

```
r.in.gdal [-o] [-e] input=PATH [band=number] output=PREFIX
          [target=name] [TITLE=Title] [location=name]
```

- `-o` Sobrescribe la proyección (usa la proyección del *LOCATION*)
- `-e` Extiende la region actual (`DEFAULT_WIND`) en le mapset *PERMANENT* para incluir al nuevo raster. La resolución anterior es preservada, pero la region, y el numero de filas y columnas son actualizadas. Esto puede fallar si el usuario no tiene permiso de escritura en el mapset *PERMANENT*.
- `input=PATH` Ruta a un raster soportado por GDAL.
- `band=number` Numero de de la banda a extraer, por defecto extrae todas las bandas, el numero debe estar entre 1 el numero de bandas del raster.
- `output=PREFIX` Nombre del raster dentro de **GRASS** , si se importa mas de una banda, se creara un grupo.

- **target=name** Nombre de un *LOCATION* del cual se lee la información de la proyección , para la transformación GCP.
- **TITLE=Title** Título asignado al raster resultante.
- **location=name** Nombre de un *LOCATION* a ser creado basado sobre el raster, si el usuario lo desea.

2.3.5. Ejemplo de Como Importar una Imagen

Para importar una imagen en formato HFA (Erdas) de tres bandas la cual se encuentra en la ruta `/usr/local/imagenes/imagen.img`, se puede usar el comando `r.in.gdal` así:

```
GRASS: >r.in.gdal input=/usr/local/imagenes/imagen.img output=xs
```

GRASS deberá mostrar algo similar:

```
PROJ_INFO file not found for location MYLOCATION
PROJ_UNITS file not found for location MYLOCATION
100 %
CREATING SUPPORT FILES FOR xs.1
100 %
CREATING SUPPORT FILES FOR xs.2
100 %
CREATING SUPPORT FILES FOR xs.3
```

Dentro de la base de datos de **GRASS** quedara almacenado los siguientes raster:

- xs.1
- xs.2
- xs.3

También se puede importar una imagen pancromática así:

```
GRASS: >r.in.gdal input=/usr/local/imagenes/pancromatica.img output=pan
```

GRASS deberá mostrar algo similar:

```
PROJ_INFO file not found for location MYLOCATION
PROJ_UNITS file not found for location MYLOCATION
100%
CREATING SUPPORT FILES FOR pan
```


2.4. Exportando

2.4.1. Formato TIFF

2.4.2. Formato PNG

2.4.3. Sin formato (binario)

2.4.4. `r.out.bil`

2.4.5. `r.out.elas`

2.4.6. `r.out.ppm3`

2.4.7. `r.out.ascii`

2.4.8. `r.out.ppm`

2.4.9. Erdas Lan

2.4.10. `r.out.gdal`

2.5. Pasos después de Importar las Imágenes

Una vez que importadas las imágenes se deben hacer la región igual tamaño de la imagen importada, para lo cual debemos correr el comando `g.region`, y a continuación el comando `r.support`¹ con el cual cambiamos algunas propiedades de los rasters importados. A continuación se ve esto mas detalle.

2.5.1. Actualización de la Region

El comando `g.region` permite modificar la región con la que trabaja **GRASS**, la región es la localización geográfica sobre la que actúan los comandos de **GRASS**², inicialmente lo mas adecuado es hacer la región tan grande como la imagen que se desea trabajar, para realizar esto debemos es ejecutar el comando `g.region`.

```
GRASS: >g.region
```

A continuación **GRASS** muestra:

¹Este comando no es necesario si esta trabajando en con una version de **GRASS** mayor a 6

²Ver sección 1.3.1

```

REGION FACILITY
LOCATION: MYLOCATION                                MAPSET: usuario
CURRENT REGION: N=1 S=0 RES=1 ROWS=1
                  E=1 W=0 RES=1 COLS=1
PROJECTION: 0 (x,y)
ZONE:      0

Please select one of the following options:

    Current Region                                Region Database
1 Modify current region directly                  8 Save current region in the database
2 Set from default region                        9 Create a new region
3 Set from a database region                     0 Modify an existing region
4 Set from a raster map
5 Set from a vector map
6 Set from a sites list map
7 Set from 3d.view file

RETURN to quit
>4

```

Para hacer que la region sea tan grande como un raster se debe ingresar el número < 4 > < ENTER >, a continuación g.region muestra:

```

Enter the name of an existing raster file
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
>pan

```

Aquí se debe ingresar el nombre del raster del cual se va a basar g.region para actualizar el tamaño de la región, si se ingresa list el desplegara un listado de los raster disponibles, en este ejemplo se ingresa el nombre del raster pan y se oprime < ENTER >, a lo cual se muestra algo similar, esto puede variar según la imagen que se trabaje.

```

IDENTIFY REGION

===== DEFAULT REGION =====
|           Default North:1           |
|                                     |
|           ===== YOUR REGION =====           |
|           | NORTH EDGE:0_____ |           |
|           |                                     |           |
Def. West |WEST EDGE |           |EAST EDGE | Def. East
-1001    |-1001_____ |           |0_____ | 1
|           | SOUTH EDGE:-1001_____ |           |
|           |                                     |           |
|           Default South:-1001           |
|                                     |
=====
PROJECTION: 0 (x,y)                                ZONE: 0

Default  GRID RESOLUTION  Region
1        --- East-West ---  1_____
1        -- North-South --  1_____

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

```

A continuación se debe oprimir < ESC > < ENTER >, GRASS pregunta si se acepta o no la region

```

projection:  0 (x,y)
zone:        0
north:       0
south:      -1001
east:        0
west:       -1001

e-w res:     1
n-s res:     1

total rows:          1001
total cols:          1001
total cells:        1,002,001

Do you accept this region? (y/n) [y] >

```

Para aceptar se debe oprimir < *ENTER* > y **GRASS** despliega lo siguiente:

```

                                REGION FACILITY
LOCATION: MYLOCATION                                MAPSET: usuario

CURRENT REGION: N=0  S=-1001  RES=1  ROWS=1001
                  E=0  W=-1001  RES=1  COLS=1001
PROJECTION: 0 (x,y)
ZONE:        0

Please select one of the following options:

    Current Region                                Region Database
1  Modify current region directly                8  Save current region in the database
2  Set from default region                      9  Create a new region
3  Set from a database region                   0  Modify an existing region
4  Set from a raster map
5  Set from a vector map
6  Set from a sites list map
7  Set from 3d.view file

RETURN to quit
>

```

En donde se puede volver a dar < *ENTER* >.

2.5.2. Uso del Comando `r.support`

El comando `r.support`³ permite cambiar archivos de soporte para un raster, están son las tareas que nos permite realizar:

- Editar el header del raster.
- Actualizar las estadísticas de la imagen (histograma,media).
- Editar el archivo de categoría.
- Crear una tabla de color.
- Editar la historia del raster.

Una vez importadas las imágenes y actualizada la región de trabajo, se debe actualizar las estadísticas de la imágenes y crearles una tabla de color (lo cual nos permite realizar una adecuada visualización). Esto se realiza ejecutando el comando `r.support`:

³Este comando desapareció en **GRASS 6**

```
GRASS: >r.support
```

Una vez hecho esto `r.support` solicita el nombre del raster sobre el cual se va a trabajar, por ejemplo `pan`:

```
{Enter name of raster file for which you will create/modify support files
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
>pan
```

Después `r.support` solicita si deseamos modificar el header, para este ejemplo se debe decir que no, dando `< ENTER >`, a continuación se solicita si se desea actualizar las estadísticas, a lo cual se debe decir que si oprimiendo `< y >` y después `< ENTER >`.

```
{Edit the header for [pan]? (y/n) [n]
Update the stats (histogram,range) for [imagen.1]? (y/n) [n] y
Updating the stats for [pan]
```

A continuación nos solicita `r.support` si deseamos editar el archivo de categoría, debemos oprimir `< ENTER >` para decirle que no y después se solicita si desea cambiar la tabla de color, debemos oprimir `< y >` y `< ENTER >` para aceptar:

```
Edit the category file for [pan]? (y/n) [n]
Create/Update the color table for [pan]? (y/n) [n]y
```

El comando `.support` presenta las siguientes opciones:

```
Color table needed for file [pan] in mapset [usuario].

Please identify the type desired:
  1: Random colors
  2: Red, green, and blue color ramps
  3: Color wave
  4: Gray scale
  5: Aspect
  6: Rainbow colors
  7: Red through yellow to green
  8: Green through yellow to red
RETURN quit
>4
```

Una opción razonable es 4, la cual permite ver la imagen en escala de grises. Para las siguientes opciones se debe oprimir solo `< ENTER >`:

```
Edit the history file for [pan]? (y/n) [n]

The null file for [pan] might indicate that some cells contain
no data. If null file for [pan] doesn't exist all zero cells in [pan]
are treated by GRASS application programs as no data.

Do you want to create/reset null file for [pan] so that
all cell values are considered valid data? (y/n) [n]

Do you want to delete null file for [pan]
(all zero cells will then be considered no data)? (y/n) [n]
```

2.5.3. El Comando g.list

El comando `g.list` es un útil comando que permite ver los archivos que estén en un *LOCATION* incluyendo los raster, para ver la lista de archivos que estén en un *LOCATION* solo se debe ejecutar `g.list` así:

```
GRASS: >g.list
```

El pedirá que seleccionemos el tipo de archivo que se desea ver, para este fin se debe dar 1 y `< ENTER >`:

```

                                LIST FACILITY

This program allows you to list files from mapsets in your search path

Please select the type of file to be listed

    1 raster files
    2 binary vector files
    3 paint icon files
    4 paint label files
    5 site list files
    6 region definition files
    7 imagery group files
    8 3D view parameters

RETURN to exit

>1

```

Después de oprimir `< ENTER >` se mostrara algo similar:

```

raster files available in mapset usuario:
xs.1 xs.2 xs.3 pan
-----
hit RETURN to continue -->

```

Para este ejemplo el comando `g.list` dice que se encuentran los raster `xs.1 xs.2 xs.3 pan`. Para terminar simple se debe ejecutar dos veces `< ENTER >`.

2.5.4. Grupos y Subgrupos de imagenes en GRASS

Un grupo y un subgrupo en **GRASS** corresponden a un conjunto de imágenes, el propósito de esto es identificar las bandas utilizadas para tareas de clasificación y georreferenciación. Los grupos son necesarios para la tarea de georreferenciación, y los subgrupos son necesarios para llevar acabo una clasificación. Para crear un grupo se utiliza el comando `i.group` así:

```
GRASS: >i.group
```

A continuación se muestra lo siguiente:

```

LOCATION: MYLOCATION                i.group                MAPSET: usuario

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

Please enter the group to be created/modified

GROUP:  xs_____ (list will show available groups)

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

```

En GROUP se debe ingresar el nombre del grupo que se desea crear, para este ejemplo se ingreso `xs`, < ENTER >, y a continuación

```
xs - does not exist, do you wish to create a new group? (y/n) [n]
```

En donde debemos dar < y > < ENTER > para indicarle a **GRASS** que cree el nuevo grupo, a continuación se muestra:

```
LOCATION: MYLOCATION                GROUP: xs                MAPSET: usuario
Please mark an 'x' by the files to be added in group [nuevo]
MAPSET: usuario
x_ xs.1
x_ xs.2
x_ xs.3
-- pan
Enter 'end' to end the add query mode: ___

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

Aquí se debe marcar con x las bandas que deseamos para nuestro grupo, una vez echo esto se debe dar < ESC > < ENTER > . A continuación se nos muestra:

```
Group [xs] references the following raster files
-----
      xs.1 in usuario
      xs.2 in usuario
      xs.3 in usuario
-----
Look ok? (y/n)
```

Para aceptar oprimir < y >

```
LOCATION: MYLOCATION                GROUP: xs                MAPSET: usuario

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

Para crear un subgroup debemos dar < 5 >, en donde se nos muestra:

```
LOCATION: MYLOCATION                MAPSET: usuario
Please enter the subgroup to be created

GROUP:   xs
SUBGROUP: subxs_----- (list will show available subgroups)

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

En SUBGROUP se debe ingresar el nombre del subgrupo que se desea crear, para este ejemplo `sub_xs` y `< ESC >` `< ENTER >`, en donde se mostrará lo siguiente

```

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

x_ xs.1 in usuario
x_ xs.2 in usuario
x_ xs.3 in usuario

Enter 'end' to end the add query mode: ___

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

```

A continuación se debe marcar con x las bandas que se desean agregar al subgrupo y después se debe teclear `< ESC >` `< ENTER >`

```

Subgroup [subxs] references the following raster files
-----
-----
                xs.1 in usuario
                xs.2 in usuario
                xs.3 in usuario
Look ok? (y/n)

```

GRASS pide que confirmemos, en donde se debe dar `< y >` para aceptar, hecho esto **GRASS** muestra:

```

LOCATION: MYLOCATION                GROUP: xs                MAPSET: usuario

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

RETURN  exit

>

```

Aquí se puede dar simplemente `< ENTER >` y se crea el grupo y subgrupo.

2.6. Visualización de Imágenes

Para visualizar imágenes es necesario lanzar una ventana en la cual se puedan dibujar las imágenes, estas ventanas en **GRASS** son llamadas monitores, una vez se tenga un monitor se pueden utilizar los comandos para visualizar imágenes.

2.6.1. Lanzar monitores con `d.mon`

La forma mas rápida de utilizar `d.mon` es así:

```
GRASS: >d.mon x0
```

El argumento del comando (`x0`) es el nombre del monitor.



Figura 2.1: Monitor

2.6.2. Visualización de rasters con d.rast

Para superponer una capa raster en el monitor activo, utilizamos el comando `d.rast` cuya sintaxis es:

```
d.rast [-oi] map=raster [catlist=clist] [vallist=vlist] [bg=color]
```

El raster que se indica con el parámetro `map`, se despliega en el monitor activo, de lo contrario el programa preguntará por el raster que se desea desplegar.

Opciones:

- `-o` Superpone la capa *name* a la capa que se encuentra en el monitor activo, de forma que los valores nulos de esta capa no se visualizan permitiendo ver la capa inferior
- `-i` Invierte la lista de categorías

Parámetros:

- `clist` Lista de categorías que a ser desplegadas, en mapas con valores enteros.
- `vlist` Lista de valores a ser desplegados, en mapas con valores flotantes.
- `color` Color de fondo para los valores nulos de la capa, el color por defecto es blanco.
- `raster` Nombre del raster que se desea desplegar.

2.6.3. Visualización de Composiciones a Color con d.rgb y i.colors

El comando `d.rgb` permite combinar tres capas raster para formar una composición a color. su sintaxis es:

```
d.rgb [-o] red=RED green=GREEN blue=BLUE
```

Debe indicarse el nombre de cada una de las capas que serán utilizadas en los cañones de la composición, de lo contrario el programa preguntará por las imágenes que deben ser desplegadas.

Opciones:

- `-o` Superpone la capa actual a la capa que se encuentra en el monitor activo, de forma que los valores nulos de esta capa no se visualizan permitiendo ver la capa inferior



Figura 2.2: Monitor desplegando una imagen con el comando `d.rast map=pan`

Parámetros:

- RED Banda que se desplegara en el cañón del rojo.
- GREEN Banda que se desplegara en el cañón del verde.
- BLUE Banda que se desplegara en el cañón del azul.

La siguiente imagen se desplegó con el comando:

```
GRASS: >d.rgb red=xs.3 green=xs.2 blue=xs.1
```



Figura 2.3: Monitor desplegando una imagen con el comando `d.rgb`

El comando `i.colors`⁴ crea composiciones a color con las capas ubicadas dentro de un grupo. Solo funciona mediante modo gráfico, requiere de un monitor activo y el usuario debe elegir previamente el grupo de imágenes con el que desea trabajar con el comando `i.group`.

2.6.4. Manejo de marcos en un monitor `d.frame`

Este comando maneja marcos de visualización dentro del monitor activo, su sintaxis es:

```
d.frame [-c,-e,-p,-s,-l,-D] [frame=name] [at=bottom,top,left,right]
```

⁴Este comando desapareció en la versión 6 de GRASS

Opciones:

- **-c** Crea un nuevo marco de visualización en el monitor activo
- **-e** Elimina los marcos de visualización existentes.
- **-p** Imprime el nombre del marco activo
- **-s** Selecciona un marco del monitor para convertirlo en el marco activo
- **-l** Lista el nombre de los mapas visualizados en el monitor
- **-D** Imprime el estado de los monitores usados

Parámetros:

- **name** El nombre del marco de visualización creado o seleccionado.
- **ottom,top,left,right** Dónde se colocará el marco. Las coordenadas del marco seleccionadas: esquina inferior, superior, derecha e izquierda. La esquina inferior derecha del monitor tendrá la coordenada de localización (0,0) y la esquina superior izquierda la coordenada (100, 100)

Este comando es útil si se desea visualizar varias imágenes en un solo monitor, para realizar esto se deben realizar los siguientes pasos:

1. Crear un marco para desplegar una imagen.
2. Desplegar la imagen sobre el marco.
3. Repetir 1 si se desea visualizar otra imagen.

Ejemplo: Si deseamos visualizar tres bandas `xs.1,xs.2,xs.3` podemos ejecutar los siguientes comandos:

```
GRASS: >d.frame -c
```

A continuación se debe crear el marco en el monitor, para esto con el botón izquierdo del mouse se debe marcar con la esquina inferior izquierda del marco

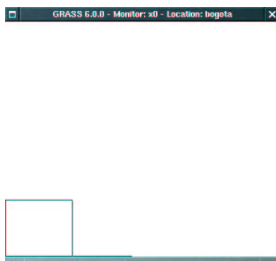


Figura 2.4: Definiendo la esquina izquierda abajo de un marco

Después se debe definir la esquina superior derecha del marco, esto se debe realizar haciendo click con el botón derecho del mouse.

Ahora se debe visualizar la primera imagen, por ejemplo se puede desear la imagen `pan` con el siguiente comando:

```
GRASS: >d.rast map=pan
```

Para desplegar otra imagen debemos repetir los pasos anteriores, con la diferencia que cuando se el nuevo marco este debe ubicarse en lugar diferente en el que se definió el anterior o los anteriores marcos. La siguiente imagen muestra dos imágenes desplegadas en un mismo monitor con la creación de marcos.

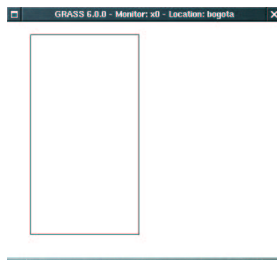


Figura 2.5: Definiendo la esquina derecha arriba de un marco



Figura 2.6: Imagen pancromatica desplegada en un marco

2.6.5. Creación de Composiciones a Color

`i.composite`⁵ crea una imagen por medio de la composición de tres raster especificadas por el usuario, una para el rojo, otra para el verde y otra para el azul. El resultado es una imagen raster que puede ser manipulada de la misma forma que si se tratara de una de los raster originales. Antes de usar `i.composite` se debe ejecutar el comando `r.support` y actualizar las estadísticas, esto se puede realizar así `r.support -r map=raster`, también se debe ejecutar `i.group` para incluir las imagenes con las cuales se creara la composición color. Una vez se actualice las estadísticas se debe ejecutar el comando `i.composite`:

```
GRASS: >i.composite
```

Al ejecutar el comando debe verse un menú así:

```
Select an imagery group file
Enter 'list' for a list of existing imagery groups
Enter 'list -f' for a verbose listing
Hit RETURN to cancel request
>xs
```

Aquí debemos ingresar el nombre del grupo en el que se encuentra las imágenes con las cuales vamos crear nuestra composición a color, a continuación damos < ENTER > y se nos muestra algo similar:

⁵Este comando desapareció en la version 6 de GRASS

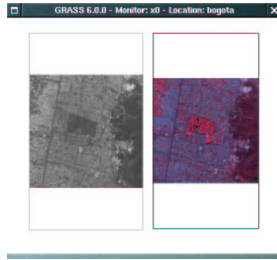


Figura 2.7: Imagen pancromatica y multiespectral desplegadas en un marco

```

LOCATION: MYLOCATION          GROUP: xs          MAPSET: usuario

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___ xs.1 in usuario
      g___ xs.2 in usuario
      r___ xs.3 in usuario

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

```

Con **r** indica la banda dedicada al rojo, con **g** la banda correspondiente al verde y con **b** la del azul, una vez que escoja la bandas presione **< ESC >** **< ENTER >** en donde se nos mostrara lo siguiente:

```

Colors assigned as follows:

RED:    xs.3 in usuario
GREEN:  xs.2 in usuario
BLUE:   xs.1 in usuario

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

```

damos **< y >** y **< ENTER >** para aceptar, **i.colors** nos muestra lo siguiente:

```

How many color levels (number of colors = levels^3)? 12

```

Aquí debemos indicar el número de colores a desplegar, se recomienda un número entre 10 y 15, 12 es una opción aceptable, damos **< ENTER >** y se nos pide que confirmemos.

```

This will result in a color table with 1728 entries. Ok? (y/n) [n]

```

Oprimimos para aceptar y se muestra:

```

Enter a new raster file name
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
>rgb

```

Ahora se debe escribir el nombre del raster nuevo y **< ENTER >**.

2.7. Mascaras

Crear las areas con **r.digit** correr **r.mask** con el raster obtenido con **r.digit**

Capítulo 3

Procesamiento Digital de Imágenes

3.1. Georreferenciación de una Imagen

Para georreferenciar una imagen es necesario tener imágenes ya importadas con los cuales trabajar, una vez hecho esto es necesario seguir los siguientes pasos:

1. Crear un *LOCATION* con la proyección y el elipsoide en que quedaran las imágenes georreferenciadas.
2. Entrar al *LOCATION* donde se encuentran las imágenes originales.
3. Crear un nuevo grupo que contenga las imágenes a georreferenciar.
4. Correr `i.target` para especificar el destino de las imágenes georreferenciadas.
5. Ingresar los puntos de control con `i.points` con los que se georreferenciarán las imágenes.
6. Georreferenciar las imagenes con el comando `i.rectify`.

Antes de iniciar el proceso se va a explicar los comandos que son necesarios para realizar la georreferenciación.

3.1.1. `i.target`

El proceso de georreferenciación de imágenes de satélite o fotografías aéreas implica el paso de un sistema de coordenadas arbitrario (filas y columnas) a un sistema de coordenadas estándar. Este cambio de sistemas de coordenadas implica que las imágenes no corregidas deben almacenarse en un *LOCATION* específico para todos las imágenes que contengan coordenadas arbitrarias y después del proceso de georreferenciación pasarse a un *LOCATION* con coordenadas reales. El comando `i.target` permite establecer cual será el *LOCATION* y *MAPSET* en el que se almacenarán las imágenes a georreferenciar, las imágenes que se desean georreferenciar deben estar contenidas en un determinado grupo de imágenes (creado con el comando `i.group`) en el momento que sean georreferenciadas con el comando `i.rectify`. Tanto el comando `i.rectify` como el comando `i.points` necesitan que el destino (target) haya sido previamente establecido.

3.1.2. `i.points`

`i.points` Permite establecer puntos de control en un raster no georreferenciado (una imagen de satélite por ejemplo) e introducir sus coordenadas reales, para generar la matriz de transformación de coordenadas necesaria para georreferenciarla con el comando `i.rectify`.

3.1.3. `i.rectify`

`i.rectify` Procede a la georreferenciación de una imagen mediante una transformación de coordenadas para cada celdilla de la imagen utilizando una matriz de coeficientes de transformación creada por el módulo `i.points`. `i.rectify` solo transformará aquellas celdillas situadas en la región activa, por tanto es importante comprobar que este bien seleccionada antes de proceder a la georreferenciación. El orden del polinomio que es utilizado para la georreferenciación puede tener un grado entre 1 y 3.

3.1.4. Ejemplo

En este ejemplo se va a georreferenciar una imagen Spot de la ciudad de Bogotá, la cual tiene una resolución espacial de 10 metros. La proyección cartográfica que vamos a usar es Transversa de Mercator y el elipsoide utilizado es el internacional.

Crear un Nuevo *LOCATION*

Lo primero que vamos a realizar es crear un nuevo *LOCATION* llamado GEO, para hacer esto primero debemos iniciar **GRASS** así:

```
$ grassX
```

A continuación **GRASS** nos mostrara una algo similar:

```

                                GRASS 5.0.0

LOCATION: This is the name of an available geographic location.  -spearfish-
         is the sample data base for which all tutorials are written.

MAPSET:  Every GRASS session runs under the name of a MAPSET.  Associated
         with each MAPSET is a rectangular COORDINATE REGION and a list
         of any new maps created.

DATABASE: This is the unix directory containing the geographic databases

         The REGION defaults to the entire area of the chosen LOCATION.
         You may change it later with the command: g.region
-----

LOCATION:  GEO_____          (enter list for a list of locations)
MAPSET:   usuario_____      (or mapsets within a location)

DATABASE: /home/usuario/MYDATABASE_____

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

```

Oprimimos `< EXC >` `< ENTER >` y **GRASS** nos muestra lo siguiente:

```

LOCATION <GEO> - doesn't exist

Available locations:
-----
MYLOCATION
-----

Would you like to create location <GEO> ? (y/n)y

```

En donde se debe ingresar `< y >` `< ENTER >` para crear el *LOCATION*, a continuación **GRASS** muestra:

```

To create a new LOCATION, you will need the following information:

1. The coordinate system for the database
   x,y (for imagery and other unreferenced data)
   Latitude-Longitude
   UTM
   Other Projection
2. The zone for the UTM database
   and all the necessary parameters for projections other than
   Latitude-Longitude, x,y, and UTM
3. The coordinates of the area to become the default region
   and the grid resolution of this region
4. A short, one-line description or title for the location

Do you have all this information for location <GEO> ? (y/n)y

```

En este paso se debe oprimir `< y >` `< ENTER >` para aceptar, ya que se conoce la información solicitada. Una vez realizado esto **GRASS** solicita información sobre el sistema de coordenadas, en donde se debe oprimir `< D >` `< ENTER >`

```

Please specify the coordinate system for location <GEO>

A  x,y
B  Latitude-Longitude
C  UTM
D  Other Projection
RETURN to cancel

>D

```

Después de esto **GRASS** solicita confirmación sobre la opción, esto se hace tecleando `< y >` `< ENTER >`

```

Other Projection coordinate system? (y/n) [y]

```

GRASS también **GRASS** solicita una descripción sobre el location, para este ejemplo simplemente bogota

```

Please enter a one line description for location <GEO>

>bogota

```

Aceptar oprimiendo `< ENTER >`

```

=====
bogota
=====
ok? (y/n) [y]

```

A continuación se debe definir la proyección a usar, en este ejemplo se selecciona `tmerc` para indicarle a **GRASS** que se desea usar la proyección Transversa de Mercator.

```

Please specify projection name
Enter 'list' for the list of available projections
Hit RETURN to cancel request

>tmerc

```

Después se debe ingresar `international` para que **GRASS** use el elipsoide internacional.

```

Please specify ellipsoid name
Enter 'list' for the list of available ellipsoids
Hit RETURN to cancel request

>international

```

En este paso **GRASS** solicita el meridiano y el paralelo central, para el caso de Bogotá se tiene $\varphi = 4^{\circ}35'56,57''$ Norte y $\lambda = 74^{\circ}04'51,30''$ Oeste, además el factor de escala es 1 y las coordenadas de origen son (1000000,1000000), es importante ver que los ángulos son ingresados a **GRASS** como una sola cifra decimal.

```

Do you want to specify a map datum for this location?(y/n) [n] n
Enter Central Parallel [lat_0] (23N) :4.599047222

Enter Central Meridian [lon_0] (96W) :-74.08091667

Enter Scale Factor at the Central Meridian [k_0] [1.0000000000]: 1

Enter False Easting [x_0] [0.0000000000]: 1000000

Enter False Northing [y_0] [0.0000000000]: 1000000
Enter plural form of units [meters]:meters

```

Después **GRASS** solicita el tamaño de la región, la cual se puede dejar con los valores por defecto y después se actualiza con las coordenadas de la imagen georreferenciada, para aceptar simplemente oprimir `< ESC >` `< ENTER >`.

```

DEFINE THE DEFAULT REGION

===== DEFAULT REGION =====
| NORTH EDGE:1_____ |
|                       |
WEST EDGE |                       | EAST EDGE
0_____ |                       | 1_____
| SOUTH EDGE:0_____ |
|=====|

PROJECTION: 99 (Other Projection)      ZONE: 0

GRID RESOLUTION
East-West:      1_____
North-South:    1_____

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

```

Aquí también se debe oprimir `< y >` `< ENTER >`

```

projection: 99 (Other Projection)
zone:      0
north:     1
south:     0
east:      1
west:      0

e-w res:   1
n-s res:   1

total rows:      1
total cols:      1
total cells:     1

Do you accept this region? (y/n) [y] > y

```

Hecho esto **GRASS** muestra que la nueva *LOCATION* se ha creado, aquí se debe oprimir `< ENTER >`


```
LOCATION <GEO> created!
Hit RETURN -->
```

Después **GRASS** solicita el nombre del nuevo *LOCATION* que se ha creado, para aceptar se debe digitar *< ESC >* *< ENTER >*

```

GRASS 5.0.0

LOCATION: This is the name of an available geographic location. -spearfish-
is the sample data base for which all tutorials are written.

MAPSET: Every GRASS session runs under the name of a MAPSET. Associated
with each MAPSET is a rectangular COORDINATE REGION and a list
of any new maps created.

DATABASE: This is the unix directory containing the geographic databases

The REGION defaults to the entire area of the chosen LOCATION.
You may change it later with the command: g.region
-----

LOCATION:  GEO_____          (enter list for a list of locations)
MAPSET:   PERMANENT_____    (or mapsets within a location)

DATABASE: /home/usuario/MYDATABASE_____

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

GRASS da la bienvenida y se debe oprimir *exit* para salir.

```

Welcome to GRASS 5.0.0 (August 2002)

Geographic Resources Analysis Support System (GRASS) is Copyright,
1999-2002 by the GRASS Development Team, and licensed under terms of the
GNU General Public License (GPL).

This GRASS 5.0.0 release is coordinated and produced by the
GRASS Development Team headquartered at ITC-irst (Trento, Italy) with
worldwide support and further development sites located at Baylor
University and the University of Illinois (U.S.A.).

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

This version running thru the Bash Shell (/bin/bash)
Help is available with the command:      g.help
See the licence terms with:              g.version -c
Start the graphical user interface with: tcltkgrass&
When ready to quit enter:                exit
GRASS:~/MYDATABASE >exit
```

Crear un nuevo grupo

Para realizar este paso lo primero que se debe hacer es ingresar a **GRASS** en el *LOCATION* que contiene la imagen que se desea georreferenciar, y a continuación se debe ejecutar el comando *i.group* así:

```
GRASS: >i.group
```

GRASS muestra lo siguiente, como se desea crear un grupo nuevo, se debe ingresar *gpan* el cual sera el nombre del nuevo nuevo grupo, y digitar *< ESC >* *< ENTER >*.

```

LOCATION: MYLOCATION                i.group                MAPSET: usuario

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

Please enter the group to be created/modified

GROUP:  gpan_____ (list will show available groups)

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

```

Aquí se pregunta si se desea crear el nuevo grupo, a lo cual se debe digitar si ingresando `< y >` y después `< ENTER >`.

```

gpan - does not exist, do you wish to create a new group? (y/n) [n]y

```

A continuación se debe ingresar con `x` las imágenes que se desean georreferenciar, para este ejemplo se selecciona únicamente **pan**.

```

LOCATION: MYLOCATION                GROUP: gpan                MAPSET: usuario

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

MAPSET: usuario

x_ pan
-- xs.1
-- xs.2
-- xs.3

Enter 'end' to end the add query mode: ---

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

```

GRASS confirma, para aceptar se digita `< y >` `< ENTER >`

```

Group [gpan] references the following raster file
-----
                pan in usuario
-----
Look ok? (y/n)y
group gpan created !

```

El comando `i.group` muestra el siguiente menú, se debe digitar `< ENTER >` para salir.

```

LOCATION: MYLOCATION                GROUP: gpan                MAPSET: usuario

1.    Select a different group

2.    Edit group title

3.    Include new raster files in the group
      or remove raster files from the group

4.    Assign colors to the group

5.    Create a new subgroup within the group

RETURN  exit

>

```


En el monitor es desplegado la imagen dos veces como se ve a continuación, para hacer zoom se debe hacer click sobre el botón **zoom** que se encuentra en la parte inferior izquierda del monitor, una vez hecho esto se puede dibujar un cuadro haciendo click con el botón izquierdo sobre la imagen que se encuentra en la parte superior. Habiendo realizado el cuadro se muestra en la imagen de abajo el zoom correspondiente al cuadro que anteriormente se realizo.

Hecho esto se muestran las coordenadas del punto que se ingresaron.

```

+-----+
|Point 1 marked on image at                                     |
|                                                             |
| East:   -4482.79                                           |
|                                                             |
| North:  -2735.92                                           |
|                                                             |
| Point located at                                           |
| East:   994780.00                                          |
| North:  1010680.00                                          |
|                                                             |
+-----+
|Use mouse now                                               |
|                                                             |
+-----+

```

Se realiza este procedimiento para todos los puntos, después se debe hacer click en **ANALIZE** y se muestra :

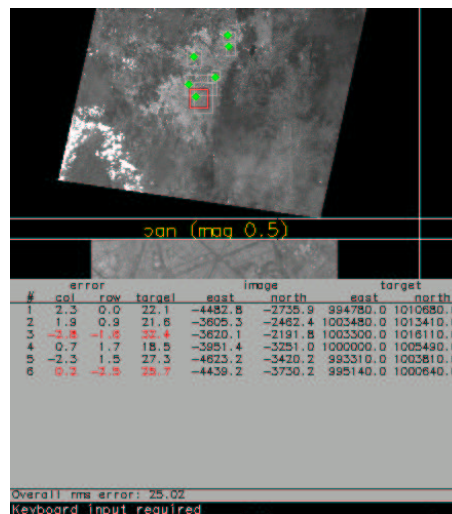


Figura 3.2: Monitor mostrando resultados del calculo del polinomio

Se hace click en **DONE**, **QUIT** y **YES**.

Georreferenciación de la Imagen

Para georreferenciar la imagen se debe ejecutar el comando `i.rectify`, este comando únicamente se ejecuta interactivamente por lo cual los datos se deben ingresar uno a uno.

```
GRASS: >i.rectify
```

A continuación `i.rectify` solicita el nombre del grupo de imagenes que se desea georreferenciar, se escribe `gpan` para este ejemplo.


```

north:      1038358.41
south:     969246.45
east:      1039283.97
west:      960569.52

e-w res:    9.95
n-s res:    10.11
total rows: 6836
total cols: 7911
total cells: 54079596

Do you accept this region? (y/n) [n] y
Would you like this region saved as the region in the target location?
(y/n) y

```

Realizado esto el comando `i.rectify` empieza a georreferenciar la imagen o el grupo de imágenes. Es importante tener en cuenta que el comando `i.rectify` se ejecuta en segundo plano, ya que el proceso puede llevar según el tamaño de la imagen y la velocidad del computador desde minutos hasta horas.

3.2. Transformaciones

3.2.1. Transformada RGB-HIS y HIS-RGB

El comando `i.rgb.his`

El comando `i.rgb.his` nos permite realizar la transformada RGB (Red Green Blue) a HIS (Hue Intensity Saturation), la sintaxis del comando es sencilla:

```
i.rgb.his red_input=raster_red green_input=raster_green blue_input=raster_blue
          hue_output=raster_h intensity_output=raster_i saturation_output=raster_s
```

- `raster_red` Raster que contiene la banda del rojo
- `raster_green` Raster que contiene la banda del verde
- `raster_blue` Raster que contiene la banda del azul
- `raster_h` Raster que almacenara el matiz
- `raster_i` Raster que almacenara la intensidad
- `raster_s` Raster que almacenara la saturación

Si se tiene almacenado en **GRASS** los raster `xs.1`, `xs.2`, `xs.3` correspondientes al verde, azul y rojo respectivamente, para obtener la intensidad, el matiz y la saturación se puede ejecutar el comando `i.rgb.his` de la siguiente manera:

```
GRASS: >i.rgb.his red_input=xs.3 green_input=xs.2 blue_input=xs.1 hue_output=h
intensity_output=i saturation_output=s
```

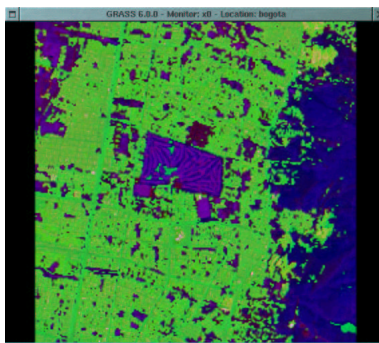



Figura 3.3: Transformada HIS

El comando `i.his.rgb`

Este comando permite realizar la transformada HIS a RGB y su sintaxis es la siguiente:

```
i.his.rgb hue_input=raster_h intensity_input=raster_i saturation_input=raster_s
        red_output=raster_red green_output=raster_green blue_output=raster_blue
```

`raster_h` : Raster con el matiz

`raster_i` : Raster con la intensidad

`raster_s` : Raster con la saturación

`raster_red` : Raster que almacenara el rojo

`raster_green` : Raster que almacenara el verde

`raster_blue` : Raster que almacenara el azul

Una forma de utilizar el comando seria la siguiente:

```
GRASS: >i.his.rgb hue_input=h intensity_input=i saturation_input=s
red_output=rojo green_output=verde blue_output=azul
```

3.2.2. Transformada RGB-HSV y HSV-RGB

Los comandos que nos permiten realizar estas transformadas son dos scripts `rgb.hsv.sh` y `hsv.rgb.sh` que utilizan el comando `r.mapcalc` el cual se vera mas adelante, por ser estos comandos scripts el tiempo de ejecución es mas largo de lo normal.

El Comando `rgb.hsv.sh`

Este comando efectúa la transformada RGB (Red Green Blue) a HSV (Hue Saturation Value)², la sintaxis del comando es la siguiente:

```
rgb.hsv raster_red raster_green raster_blue
```

`raster_red` : Raster correspondiente al rojo

`raster_green` : Raster correspondiente al verde

`raster_blue` : Raster correspondiente al azul

²Este comando no esta disponible en la version 6 de **GRASS**

Para realizar la transformada se puede ejecutar el siguiente comando:

```
GRASS: >rgb.hsv.sh xs.3 xs.2 xs.1
```

Los raster obtenidos tienen los nombres H, S y V. Al ejecutar este comando se tiene una salida similar a esta:

```
Computing Hue (H), Saturation (S), and Value (V)
From Red (xs.3), Green (xs.2), and Blue (xs.1)
V = (max(xs.3[0,0],xs.2[0,0],xs.1[0,0]))
S = (div(mul(255.000000,double(sub(V,min(xs.3[0,0],xs.2[0,0],xs.1[0,0])))),
double(V)))
H = (if(S,eval(m = (float(min(xs.3[0,0],xs.2[0,0],xs.1[0,0]))),
r = (div(float(sub(V,xs.3[0,0])),sub(float(V),m))),
g = (div(float(sub(V,xs.2[0,0])),sub(float(V),m))),
b = (div(float(sub(V,xs.1[0,0])),sub(float(V),m))),
h = (add(add(if(double(eq(xs.3[0,0],V)),sub(b,g)),
if(double(eq(xs.2[0,0],V)),sub(add(float(2),r),b))),
if(double(eq(xs.1[0,0],V)),sub(add(float(4),g),r))),
mul(if(double(le(h,float(0))),add(h,float(6)),h),float(60))))))
```

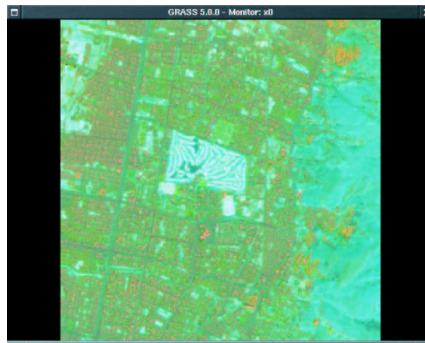


Figura 3.4: Transformada HSV

El Comando `hsv.rgb.sh`

El comando `hsv.rgb.sh` lleva a cabo la transformada HSV a RGB, la sintaxis del comando es la siguiente:

```
hsv.rgb.sh H S V
```

H : Raster con la matiz

S : Raster con la saturación

V : Raster con el valor

Se puede ejecutar este comando así:

```
GRASS: >hsv.rgb.sh H S V
```

La salida de la ejecución del comando debe ser similar a la siguiente:

```

Computing Red (R), Green (G), and Blue (B)
From Hue (H), Saturation (S), and Value (V)
R = (eval(s = (div(S[0,0],255.000000)),h = (div(double(if(double(ge(H[0,0],
float(360))),sub(H[0,0],float(360)),H[0,0])),60.000000)),i = (int(h)),
f = (sub(h,double(i))),p = (mul(double(V[0,0]),sub(double(1),s))),q =(mul(
double(V[0,0]),sub(double(1),mul(s,f))),t = (mul(double(V[0,0]),sub(double
(1),mul(s,sub(double(1),f))))),add(add(add(add(add(double(if(double(eq(i,0))
,V[0,0])),if(double(eq(i,1)),q)),if(double(eq(i,2)),p)),if(double(eq(i,3)),
p)),if(double(eq(i,4)),t)),double(if(double(eq(i,5)),V[0,0]))))
G = (eval(s = (div(S[0,0],255.000000)),h = (div(double(if(double(ge(H[0,0],
float(360))),sub(H[0,0],float(360)),H[0,0])),60.000000)),i = (int(h)),
f = (sub(h,double(i))),p = (mul(double(V[0,0]),sub(double(1),s))),q =(mul(
double(V[0,0]),sub(double(1),mul(s,f))),t =(mul(double(V[0,0]),sub(double(1)
,mul(s,sub(double(1),f))))),add(add(add(add(add(if(double(eq(i,0)),t),double
(if(double(eq(i,1)),V[0,0])),double(if(double(eq(i,2)),V[0,0])),if(double(
eq(i,3)),q)),if(double(eq(i,4)),p)),if(double(eq(i,5)),p))))
B = (eval(s=(div(S[0,0],255.000000)),h =(div(double(if(double(ge(H[0,0],float
(360))),sub(H[0,0],float(360)),H[0,0])),60.000000)),i = (int(h)),f = (sub(h,
double(i))),p =(mul(double(V[0,0]),sub(double(1),s))),q = (mul(double(V[0,0])
,sub(double(1),mul(s,f))),t = (mul(double(V[0,0]),sub(double(1),mul(s,sub(
double(1),f))))),add(add(add(add(add(if(double(eq(i,0)),p),if(double(eq(i,1))
,p)),if(double(eq(i,2)),t)),double(if(double(eq(i,3)),V[0,0])),double(if(
double(eq(i,4)),V[0,0])),if(double(eq(i,5)),q))))

```

3.2.3. Transformada de Fourier

Transformada Directa de Fourier

Para efectuar la transformada de fourier en **GRASS** se utiliza el comando `i.fft`, este comando realiza la transformada rápida de fourier **FFT**, debido ha esto el raster de entrada debe tener dimensiones de la forma 2^n , si no es así **GRASS** completa el raster con ceros hasta que se complete el tamaño mas cercano a 2^n . La sintaxis del comando es la siguiente:

```
i.fft input_image=raster real_image=parte_real imaginary_image=parte_imaginaria [range=value]
```

`raster` : Raster al cual se le calculan la transformada

`parte_real` : Raster que almacena la parte real

`parte_imaginaria` : Raster que almacena la parte imaginaria

`value` : Rango de valores al visualizar `parte_real` y `parte_imaginaria`

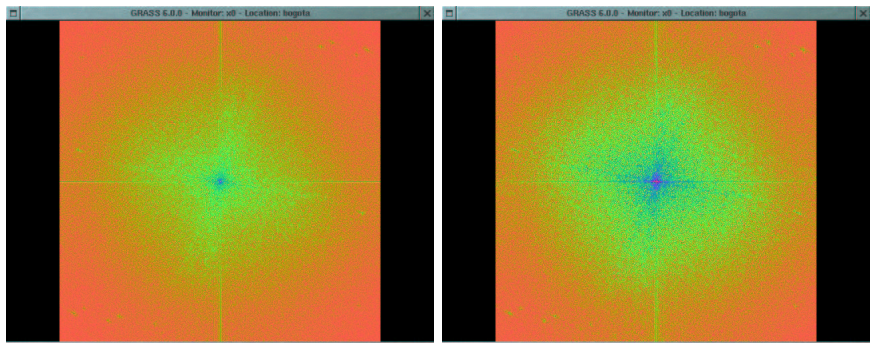


Figura 3.5: Parte Real e Imaginaria de la Transformada de Fourier de la Imagen Pancromatica

Transformada Inversa de Fourier

La transformada inversa de fourier en **GRASS** se realiza con el comando `i.fft` el cual tiene la siguiente sintaxis:

```
i.ifft real_image=parte_real imaginary_image=parte_imaginaria output_image=raster
```

`parte_real` : Raster con la parte real

`parte_imaginaria` : Raster con la parte imaginaria

`raster` : Raster generado

Antes de ejecutar este comando normalmente se debe definir una mascara con el comando `r.mask`

3.2.4. Componentes Principales

- i.pca

3.2.5. Transformación TassledCap

Para realizar la transformada TassledCap en **GRASS** están disponibles dos scripts `i.tasscap.tm4`, `i.tasscap.tm5`, el primero como su nombre lo indica para es para una imagen procedente de **LANSAT 4**

3.3. Clasificación de Imágenes

3.3.1. Clasificación No Supervisada

La clasificación de imágenes consta de dos pasos, en el primer paso se obtienen las firmas espectrales y en el segundo se obtiene la imagen ya clasificada. Como prerrequisito para realizar una clasificación se deben crear un grupo y un subgrupo en donde se encuentren las imágenes a clasificar. El primer paso se realiza ejecutando el comando `i.cluster`

```
GRASS: >i.cluster
```

Aquí se pide el nombre del grupo en el que se encuentran las imágenes

```
You may run i.cluster to generate signature file only,
or accompanied by a report file in a specified directory
(report-filename-only results in current directory).

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

A continuación `i.cluster` pide el nombre del subgrupo

```
Enter the subgroup filename
Enter 'list' for a list of subgroups in group [class]
Enter 'list -f' for a verbose listing
Hit RETURN to cancel request
> subclass
```

Después se solicita el nombre del archivo en el se almacenaran las firmas espectrales.

```
RESULT SIGNATURE
Enter name for the resulting signature file
Enter 'list' for a list of existing subgroup [subclass] signature files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
>firmas
```

En el siguiente paso se solicita un archivo que contenga firmas espectrales obtenidas en una clasificación anterior, las cuales serán utilizadas como semillas. Como no se dispone para este caso de firmas para ser utilizadas como semillas no se proporciona el nombre de un archivo.

```
SEED SIGNATURES
Select the signature file to use for the initial class means
Enter 'list' for a list of existing subgroup [subclass] signature files
Enter 'list -f' for a list with titles
Hit RETURN to use DEFAULT means
>
```

Después `i.cluster` pregunta si se desea un archivo de reporte en el cual quedara almacenado la siguiente información:

- Location actual
- Mapset actual
- Coordenadas de la Region
- Nombre del grupo de las imagenes
- Nombre del subgrupo de las imagenes
- Información sobre la mascara
- Parámetros de agrupamiento
- Media y desviación estándar de cada banda
- Media y desviación estándar media de cada clase inicial para cada banda incluida.
- Parámetros indicados al comando `i.cluster`

Si se desea obtener esta información se digita `< y >< ENTER >` para que se cree el archivo

```
The report file contains current location, mapset, region
coordinates, imagery group name, subgroup name and mask information.
Initial parameters of clustering, means and standard deviations of
each band, and initial class means and standard deviations for each
band are included.

At each step of iteration, number of classes, percentage
of stable points, and points of each class are recorded.
Finally, results including number of classes, class
separability matrix, and class means and standard deviations
for each band are at the end of the report file.

Would you like to have a report file? (y/n) [y] y
```

Despues se ingresa el nombre del archivo, por ejemplo `reporte`

```
Enter report filename: reporte
```

Ya que este comando puede tardar mucho tiempo, se puede ejecutar en segundo plano, si se desea de esta forma se debe digitar `< y >< ENTER >`, de lo contrario podemos digitar `< ENTER >`

```
Would you like to run in the background? (y/n) [n]
```

En el siguiente paso **GRASS** nos solicita los siguientes parámetros:

1. Número inicial de clases

2. Mínimo tamaño de una clase
3. Separación de la clase
4. Porcentaje de convergencia
5. Máximo número de iteraciones
6. Tamaño en filas y columnas de la muestra

Los valores escogidos para este ejemplo aparecen a continuación:

```

Please set the following information

Number of initial classes      5_____
Minimum class size            17_____
Class separation               0_____
Percent convergence           98_____
Maximum number of iterations  30_____

Your current region contains 6671 rows and 8208 cols (54755568 cells)
Please set the sampling intervals

      Row interval            66_____
      Col interval           82_____

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

```

Para aceptar se debe digitar < *ESC* >< *ENTER* >

```

Reading image ... 100%
Iteration 1: %Convergence 78.33 (0s elapsed, 0s left)
Iteration 2: %Convergence 83.89 (0s elapsed, 0s left)
Iteration 3: %Convergence 90.00 (0s elapsed, 0s left)
Iteration 4: %Convergence 90.00 (0s elapsed, 0s left)
Iteration 5: %Convergence 90.56 (0s elapsed, 0s left)
Iteration 6: %Convergence 95.00 (0s elapsed, 0s left)
Iteration 7: %Convergence 95.00 (0s elapsed, 0s left)
Iteration 8: %Convergence 97.78 (0s elapsed, 0s left)
Iteration 9: %Convergence 98.89 (0s elapsed, 0s left)

```

Despues de este paso se debe ejecutar el comando `i.maxlik`, el cual muestra algo similar

```

LOCATION: MYLOCATION                MAXIMUM LIKELIHOOD                MAPSET: usuario

Please select the group/subgroup containing the signatures
to be used in the classification

GROUP:   class_____ (list will show available groups)
SUBGROUP: subclass_____ (list will show available subgroups)

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

```

A continuacion se solicita el archivo que contiene las firmas espectrales

```
SIGNATURE
Enter the signature file to be used for classification
Enter 'list' for a list of existing subgroup [subclass] signature files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
>firmas
```

Ingresar el nombre del raster que contiene la información clasificada

```
Please name the CLASSIFIED map layer to be generated
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
>clasificada
```

Ingresar el nombre del raster que contiene los píxeles que se rechazarán

```
Please name the REJECT THRESHOLD map layer to be generated
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN if you don't want this layer
> rechazo
```

SEGUNDO METODO `i.smap`
 correr el comando `r.digit` Antes

3.3.2. Clasificación Supervisada

Para llevar a cabo una clasificación supervisada se debe crear una composición a color, esto se lleva a cabo con el comando `i.composite`

Bibliografía

- [1] *GRASS 5.0 Programmer's Manual*, Grass Development Team.
- [2] *Grass Tutorial*, Moritz Lennert