Tectonics Research Scott Wolnum
David Fraser
Brent Henderson 12/28/95

Contents

Page

Project: Regional Tectonics Research 20-5704-163
PI: David Ferrill x6082

Objective: Data processing and visualization to support investigations of tectonics processes and relationships in regional area.

Area of investigation includes 120°W to 114°W longitude and 32°N to 40°N latitude.

Utilize various digital elevation model (DEM) databases from USGS including 3-arcsecond and 30-meter resolution DEMs.

Also use Landsat Thematic Mapper (TM) imagery as well as data from various mapping and modeling packages such as ARC/INFO and DGI EarthVision.

Current research includes investigation of maximum horizontal stress, relationships of faults and seismic events, and relationships of faults and topography.

Rand Hunter
8-5-94

Shading elevation (Lake Meade) and combining with postscript outline.

1. Work ___.ect file
2. run DemToWater (file=dma0351.dem)
   (out=water.rgb)

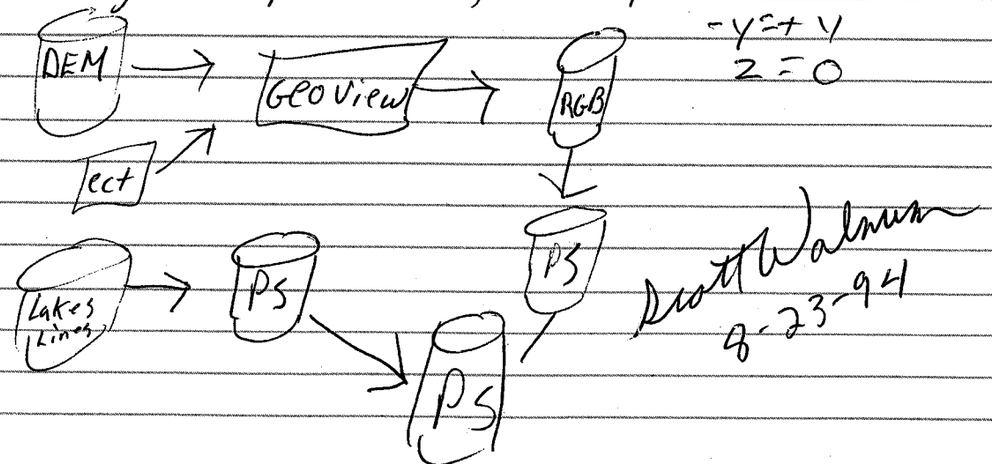
3. Make sure dma0351.rgb is a properly shaded terrain file.

4. run Assemble Lakes
   uses: dma0351.rgb = back
   water.rgb = over
   test.water.rgb = out

5. TOPS test.water.ps -rgb > water.ps
   * Edit water.ps appropriately * - (scale, showpage)
6. run 30MkMap - combines water.ps and lakes.ps → 00map

* To edit rgb use clr paint

printWidth printHeight scale
use gsave up to scale, [x00 -y00 z] : make -y=+y z=0



Scott Walnum
8-23-94

FTP site for DEM's

FTP edcftp.cr.usgs.gov : anonymous
userid
/pub/data/DEM/250 (* binary mode)

When downloading dems, or anything you can make a vi file using ftp commands and then using F4 option get the ftp going in the background.

get F/File.gz file.gz

Taking all image files (.rgb) and assembling them into one complete image.

look at runAssemble 30

place all .rgb's and rgb.water files in the specified directory.

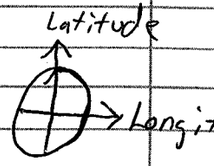
make sure the files are all the same size (istat). They should be 360x 360y. If not there is a zoom option under runAssemble.

Then run A. and the complete image will be in the current directory.

Scott Walnum
8-24-94

TO convert to UTM from geographic.

— reference for map projections
UTM - universal Transverse Mercator



- Earthvision notebook ② pg CT-9
- Arc/info - Map Projections & Coordinate Man.
- Map Projections poster

To combine the map.ps with the faults, etc.

1. Get to the img1_landers.dir.
In the ~~Parts Utm~~ parts Utm dir, have the image file as part024---- for a color image.
- run RgbProject

In ~~img~~ img1_landers, run Assemble. places the new postscript file in the current directory.

Trick: if the image file is too large to edit in vi, some options:

1. izoom the large image before conversion to ps.
2. Use nawk to edit

editing instructions on pg 2 #5

Scott Walrum
9-25-94

Processes for making black and white

1. In the rgbDems dir / bin
run010DemTORgb
Just make sure you uncomment the BW section and that you have all of the data directories correct
*Errors? - make sure you are not running an old geoview, and you have enough disk space.

Tar * tar -cvf endfile dir To compress
tar -xvf

There are hundreds of dem files, and so to convert them all into BW rgb images, I set up script files (run010DemToRgb) on performer, Atlantis, and Bemore. They will run overnight and convert all the images.

I ran the script files over night and every dem was converted except Twin Falls (my homeland). oh well, I went to the ftp site got the DEM and am running it again. After this, all the images will be converted!

Scott Walrum
9-26-94

Memory problem with circles.
looking over the code Brent wrote for circles and reviewing a reference manual, it appears that if we use the save/restore functions in the ps file, it should eliminate the VM problems. To change these files I think that a script file using nawk may be necessary.

reference save/restore pg 60-

After further study, we have determined that the memory problem is simply when the program is loading into memory. There are no memory allocations running rampant as the ps executes. So, to reduce it as much as possible, any way, we worked on methods of putting more information into the data files, and ~~less~~ take some of the processes out of the code. These experiments are described on page 9.

another interesting item we found about VM, is that it seems to be allocated in blocks, and not just as needed.

Scott Walnum
4-30-94

After converting all the DEMs on Thurs 8/26, now I am going to try and make a complete image with all of the files.

~~fiout. To~~

To begin, I put a copy of all the .rgb files using the comand ls -1 or ls -1 *.rgb | nawk '{print $NF}' > list.txt Then I edited this file to arrange the rgbs in order starting in bottom left corner going right. After this I used the command

cat list.txt | nawk '{printf "%s \\\n", $NF}' > t

to put a '\' at the end of each file name. Then at the beginning of this ~~t~~ file I placed assemble 20 12 \ out \ so I could view the whole compiled image. This works great, but there are some down loaded DEM files which have some problems and we are going to try and fix them.

Scott Walnum
4-26-94

Our problem reading in the dem files is that when there is ocean water, then in the file there is a max elevation and min elevation, which are not used, but we must read them in any way. When they are zero, they are represented as 0.0, but when they are not zero, they are in scientific notation. eg. 0.0010+13, which will read in differently than 0.0, putting the file pointer at the wrong position.

I am currently proposing 2 methods to resolve this problem. The location of the file that is needin modification is in /usr3/brent/geoview/src/dem_io.c It is cataloged using rcs - man this.

1. My first idea is to search the file until I reach the beginning of the number, then search until I find a blank. Do this 3 times for each number, and we should be home free.

2. Since these numbers all begin in the same relative positions, we could simply move the file pointer forward to the beginning of the elevation data.

The position in dem_io.c where we are working is line 420 or so.

Scott Walrum
4-26-94

We determined that one acceptable method for reducing the VM required to ~~de~~run the circle rutine would be to remove the color and mgnitude table from Epicenter header file and pass those values into the function. So, to find these values and put them into the function variables, I built a nawk procedure.

My nawk procedures begins by converting the original raw data into the format which was already being used. Then it takes this file, and the .lut (look up tables) files to make the new format. It reads the lut's into memory, then finds the proper radius for that magnitude, and finds the right RGB for the date. I writes all the data out to file, and presto, we have the new data file. Some minor modifications to the header file so that it reads in the data properly, and we're done.

This method seems to save around 2000K of VM.

The file for this conversion is run EpicentersToPs, and should be in the partsUTM dir.

rawdata UTM

Scott Walrum
4-30-94

Getting info back from tape.
put in tape.

cd /usr3/brent/backups/tape

vi -pscr0 or whatever

cd /pscr0/?

bru -cvf /dev/rmt/tps0d3 →
→ cat → ./scott/— file

I have prepared a nawk script that will return any sub map out of the DEM's. It works from a pre sorted list of all the DEM names, sorted from E to W, and S to N. Out of this list, it will ~~return~~ build an assemble command with the correct data. To do this, I use essentially two counters. If ~~when~~ the desired subset is reached during the count, it will print out the name for that coordinate

So, using it is simple. just type

runMkMap — with "all", "30", "48"
or eventually the desired coordin.

and the return image is
map.rgb
map.Zoom (and a zoom)

Scott Walnum
6-21-94

Coloring the oceans:

I plan to follow the basic pattern as for the lake, except I would like to draw the state and coast lines on an image, and then using an image editor, flood fill the ocean area, missing the mainland and the islands.

For now, I must get a blank, totally blue square for the complete ocean squares. then do an elevation fill.

~~con image~~ con img = BW or RGB

- Shading ocean

In *rgbDems/bin
run 010—

- in ../dem place the unzipped dem with no extensions (ie .gz, .rgb)
- in ../bin runDemTOWater will convert dem into the water part and place in ../wate
- into ../nowater copy the accompanying rgb file. Be sure to unzip and have the .rgb extension
- in ../Bin runMerge will merge and place combined image in ../merged

Scott Walnum
9-13-94

To transfer files to be printed on ilford or fiery

ftp 129.162.26.3

guest

Anonymous

cd h:
cd div20
cd brent
cd ilford19, fiery19

frank T. X 5850

Image editors

Img works
* enhance
clr paint

Blue color 0x9b 0xd3 0xFd

To change the color out of an imgge use rgb_Filter then -in c.e -out

Scott Walnum
10-10-94

Today I began automating the production of maps. The following is a tentative plan for what the program should accomplish and the order in which they must be done.

assemble RGB image

Convert to UTM - get zone

Convert to post-script

Build state-lines

Build extra-data
- epicenters
-
-

Build faults

Build tic marks

get legend(s)

add Box around Yukka mts

Combine with clipping

Scott Walnum
10-13-94

For most practical purposes, the map automation is complete.

The process is centered around a 'C' proram which builds an executable script file with all the info needed.

First, the coordinates, scale and zone are entered. Then the main.temp file built. In this file, the runMx RGB is called which assembles the image. Next it uses RGB_filter to change the color of the ocean. Then it projects into UTM. Then converts into ps post script and prepares it to be included in the final image. Then the appropriate state lines are found and converted into UTM. Finally, it is all put together with the buildParts script.

Another feature included is the makeDemo ~~Demo~~ which makes a complete ps file, but without the image. This helps making edits quicker.

Scott Walnum
10-25-94

To Shade a DEM.

You need to use run010DemToRGB

To get this file, copy the directory ~swalnum/#rgbDems.

In this directory, place all gzipped DEM's into the ./dem directory.

Then, in the ./bin dir, edit run010DemTORGB appropriately according to whether you want color or BW. Then exicute it.

The completed RGB files will be gzipped in the rgbDems/RGB dir.

11-19-94

Scott Walnum

Flt Slip View

For several weeks I have been working on the slip tendency and dilation viewer.

Brent has worked on this, and I have worked on adding faults and coloring them. Also have gotten some stats.

I will now summarize the process developed for the viewer dealing with faults.

When lineClass is constructed for the lineObj, the file name for the input data is set and the Initialize routine called. This routine sets the size and location of the viewing box, it then reads the file into a linked list of line nodes. The line nodes each contain a point of data and whether or not to start a new line segment. Also stored is info about length of segment, angle and quadrant of line direction.

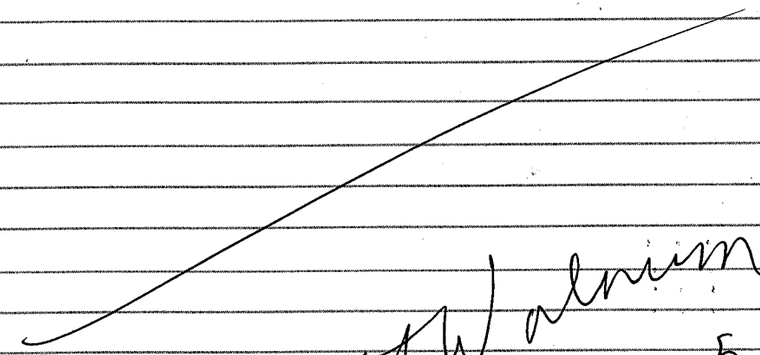
When the info is to be drawn to the screen, the display() function is called. This draws the box and displays information about the dimensions of the box.

Scott Walrum
2-14-95

Next, it goes through the list of line points, using the info such as angle to call a function which returns the proper color according to the slip/dilation viewer. It draws the colored line into the box at the proper scaled position.

During this, stats are being kept which store total fault length in and longest fault length in each azimuth bin.

The function (draws Rose diagram) is called to use this information. It builds the diagram, displaying a circular ~~data~~ object with each azimuth bin containing a relative solid fill of the total amount of fault length and a dot representing the relative length of the longest fault in the bin.



Scott Walrum
2-14-95

Shadows

The functions I designed were to implement shadows in the geoview application package.

Most of the functions are located in the shade_util.c program, and can be easily located by searching for "Scott".

The shadow functions can be turned on by setting the shadow function in the sun file to 5.

When set to 5, the shading util will either shade a vertex or not depending on whether there is something between it and the sun, Line of Sight (LOS).

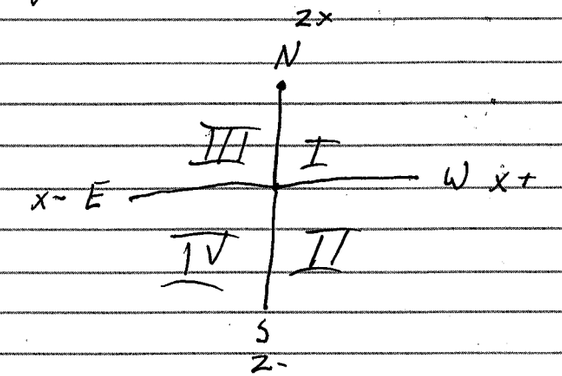
My functions, check the LOS of a vertex given a sun direction vector.

To understand what I did, we must know how the data is represented. It is organized in a grid like the one below, with each point having its own elevation data.

⋮ ⋮
⋮ ⋮

Scott Walpum
2-14-95

The first thing I do is determine what direction the sun is in the XZ plane (Y is elevation). The value of the quadrant is determined by the following diagram

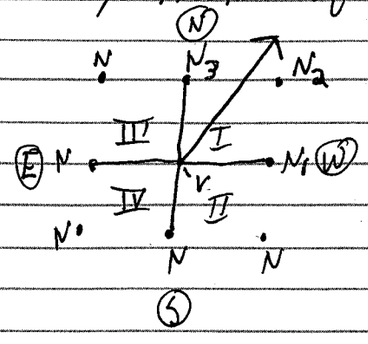


given the quadrant that the sun lies in, it now proceeds to trace the ~~line~~ sun_line (line between vertex and sun) to the edge of the information. It does this by getting neighboring points, then it finds the location of the intersection of a line between neighboring points and the sun_line in the XZ plane. Then it finds the Y value at that point on both lines. If the Y value ~~is less~~ of the sun_line is less than the other, then the sun_line has been obscured.

I shall try to illustrate this on the proceeding page

Scott Walpum
2-14-95

all vertices except those on a boarder have the following arrangement of neighbors



Supposing the red line to be the sun line, the the quadrant is the first, so neighbors N_1, N_2, N_3 would be used. It checks to see if the sun line intersects between points $N_1 : N_2$ or $N_2 : N_3$. This case is $N_2 : N_3$, so it then check the Y values and returns the LOS.

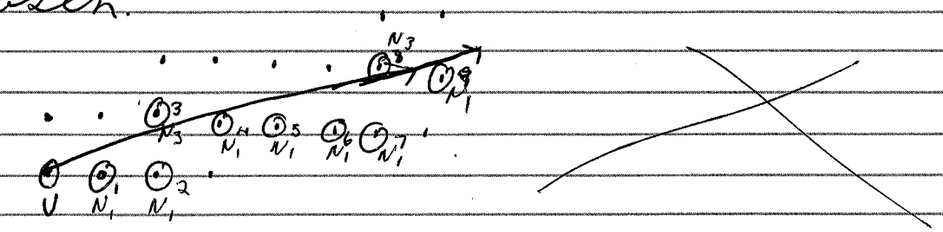
Now, if the sun line were not obscured, then the process would need to continue until either the line is obscured, or it reaches the edge (not obscured).

To do this it now uses one of the neighbors as the center vertex. Note: the sun line remains "attached" to the original vertex

Scott Walrum
2-14-95

In our example, the sun line intersected between $N_2 : N_3$, therefore the next vertex we would use would be N3. If the intersection had been between $N_1 : N_2$, then we would use N_1 as the new vertex.

To illustrate this, below is a vertex and sun line, with all of the neighbors which would be used to trace the ray to the boundary. the ones which are circled are the points which were used as the center for checking, and their position when chosen.



This method, however, is slow since it must repeat the process for every point. If I get the chance to refine it, I would like to make the process quicker by adding the ability to test more than one point. If after determining which point will be used as the next center, it could be placed on a queue. after which every point on the queue could be checked for LOS. if in LOS, then the vert data could be marked as being check and LOS = True. Then when it is time to check that vert, it would already have its status.

Scott Walrum
2-14-95

Obtaining batholithic data from cd-rom.

The data in USGS is 30'' info.

Copy it, then use:

dd conv=swab if=file of=out

The function dd can be used to view the file.

Building Posters

The files are located in the makeMaps directory, but the most important files are in ~~scripts~~ makeMaps/scripts.

- To build a poster from scratch which is a subset of the southwest US ~~data~~ DEM set: Edit or input the coordinates into poster.c. Make sure all the DEM files are in makeMaps/unzipped and they are not gzipped. Run the compiled executable from cc poster.c -o out. After OUT, view main.temp. This is the script which will build the poster. Make any changes desired or leave alone. Execute ~~main.temp~~ main.temp
  Main.temp will build the DEM (Note: edit out if dem is already built), then it converts it to utm, to post script, edits it and stores it in the ../parts dir. It also selects the state lines. ~~and builds a dem a~~ continued.

Scott Wilson
3-29-95

cont.

Once Main.tmp has completed, all the data will be in the ./parts dir. To combine this data into posterg, use the build files. A demo file can be built which leaves out the image so edits are ~~can~~ viewed more easily. The order of the build files is important so be careful with putting things in them.

some important files in ./parts

part0107_page.ps - the image size, positioning and PageSelect.

All ___Hdr.ps files

All _legend_ files

Note: For epicenters, probably use

~~___ contactHdr~~

___ = CenterHdr.ps.new > should be used

___ sortedepi___.ps

Neil L. Frueh
for
Scott Walmore
3/29/95

Build Earthquake Data

Simply edit ~~makeMaps/scripts/parts/Eq.data~~

makeMaps/scripts/parts/Eq.data/Eq.build/run

Change the coordinates entered into the awk. Then ~~run~~ execute run. Watch the comments, you will ~~have~~ be asked to exit from xpsview ~~manually~~. The xpsviewer will appear, and turn red when done. The finished ~~set~~ data will be stored in part053___ in that dir.

Neil L. Frueh
for
Scott Walmore
3/29/95.

Objective: To enable a student, unfamiliar with shading DEM's, to know where to find data and how to produce shaded DEM images.

(1) FTP site for raw DEM data: (note: use binary mode)
ftp edcftp.cr.usgs.gov :anonymous cd /pub/data/DEM/250

(2) Use run010DemToRgb found in rgbDems/bin.

rgbDems/bin/run010DemToRgb

```
#!/bin/csh

set demDir = "../dem"
set rgbDir = "../rgb"

foreach demFile ($demDir/*.gz)
  set noPath = ${demFile:t}
  set noExt  = ${noPath:r}
  echo $demFile $noPath $noExt

  echo Remove existing RGB file $noExt.rgb...
  rm ${rgbDir}/${noExt}

  echo Unzip dem file $demFile...
  gunzip $demFile

  echo Shade dem file $noExt...
  geoview \
    -intype 4 \
    -in ${demDir}/${noExt} \
    -outtype 3 \
    -out ${rgbDir}/${noExt}.rgb \
    -proj 0 \
    -nodraw \
    -scale 30.0 6.0 -30.0 \
#    FOR COLOR: \
#     -elev grn.ect \
#     -sunfile sun.txt \
#    FOR BW: \
    -c 0x00b0b0b0 0x00b0b0b0 \
    -sunfile sunbw.txt

  echo Show image file $noExt...
# ipaste ${rgbDir}/${noExt}.rgb

  echo Zoom image file $noExt...
#  izoom ${rgbDir}/${noExt}.rgb ${rgbDir}/${noExt}.rgb.zoom .3 .3
#  rm ${rgbDir}/${noExt}.rgb
#  mv ${rgbDir}/${noExt}.rgb.zoom ${rgbDir}/${noExt}.rgb
#  rm ${demDir}/dd_temps

  echo Compress dem file $noExt...
  gzip ${demDir}/${noExt}
  gzip ${rgbDir}/${noExt}.rgb

end

echo Finished "\007"
```

Place gzipped DEM's in demDir

Completed RGB's go here

Specify file names for loop

GEOVIEW: (/usr3/brent/geoview/bin)
intype 4 = DEM
outtype 3 = RGB
standard scale used

(3)

For color images, uncomment these and comment the two lines after BW.

(4)

Black and white images:
Background color
Sunfile (current directory)

Further options for completed RGB

Brent Hohn
for Scott Wahrman
12-20-95

Continued:

(3) Contents and description of an ect file.

rgbDems/bin/zgrn2Sc2.ect

```
# Color table zgrn2.ect scaled by 2
#
-20000.0000000000   ff000000
-200.0000000000     ff007cc0
-50.0000000000      ff008bc7
100.0000000000      ff009bcf
250.0000000000      ff00acd6
400.0000000000      ff00bede
550.0000000000      ff00d0e5
700.0000000000      ff00e3ec
850.0000000000      ff00f4f0
1000.0000000000     ff00fbea
1150.0000000000     ff00f8e3
1300.0000000000     ff00f5dc
.
.
.
7150.0000000000     ff008919
7300.0000000000     ff008616
7450.0000000000     ff008313
7600.0000000000     ff008010
7750.0000000000     ff007e0e
7900.0000000000     ff007b0b
8050.0000000000     ff007809
8200.0000000000     ff007506
8350.0000000000     ff007204
8500.0000000000     ff007002
8650.0000000000     ff006d00
8800.0000000000     ff026a00
```

Lower bound on elevations.

Color scale.

All points which fall between two elevations will be colored according to the lower elevation.

Upper bound on elevations.

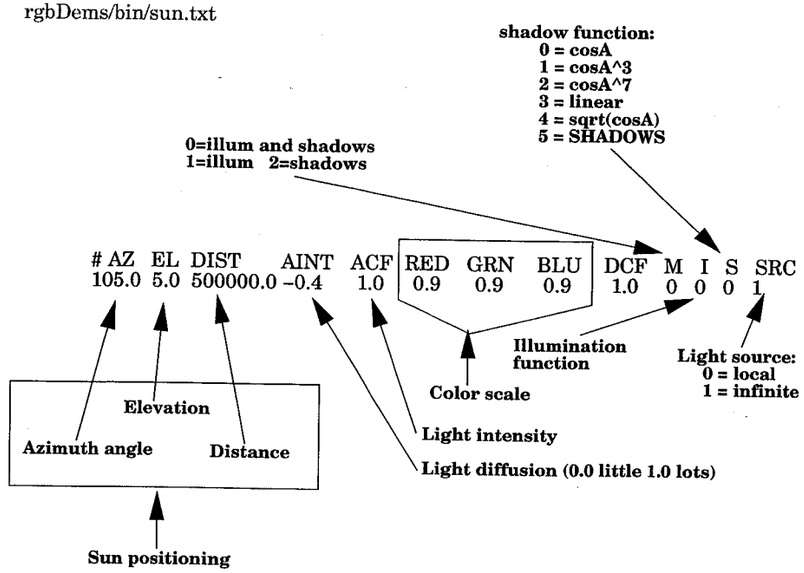
Elevations.

Brent Hohn
for Scott Wahrman
12-20-95

Continued:

(4) Contents and description of a sun illumination file.

rgbDems/bin/sun.txt

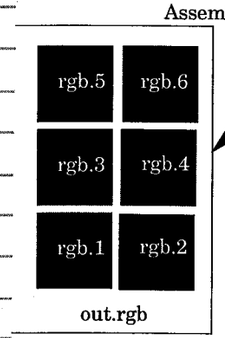


Bert Hahn
for
Scott Wahrman
12-20-95

Continued:

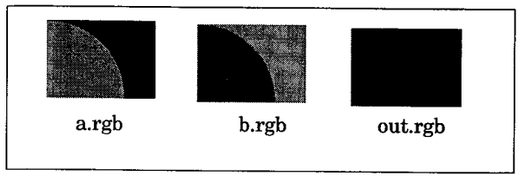
(5) Examples of how to use various image commands:

Assemble:



assemble 2 3 out.rgb rgb.1 rgb.2 rgb.3 rgb.4 rgb.5 rgb.6

Rgb images are arranged here. Also,
be sure to have all images the same size.



Rgb_merge:

```
rgb_merge \
    -bckgrnd a.rgb \
    -overlay b.rgb \
    -out out.rgb \
    -red 0x00 \
    -grn 0xff \
    -blu 0x00 \
    -blend 1.0
```

This is the color to be swapped out.

This takes a.rgb and gives the green pixels the pixel color
of the corresponding pixel in b.rgb.

Bert Hahn
for
Scott Wahrman 12-20-95

Continued:

Examples of how to use various image commands:

rgbProject:

```
rgbProject
  -inFile     map.rgb \
  -inLngMin   $1 \
  -inLngMax   $2 \
  -inLatMin   $3 \
  -inLatMax   $4 \
  -zone       $5 \
  -outFile    map.utm \
  -outPixX    $6 \
```

UTM coordinates → (-inLngMin, -inLngMax, -inLatMin, -inLatMax)

UTM zone (11) → -zone

Output size → -outPixX

Takes the lat/lng rgb image and projects it into UTM

rgb_filter:

Example: How I combined the bathometric data with the DEM data to get a complete image.

I built the bath. data image to cover the same area as the DEM using the 5 minute data set. In the ect script I had all elevation points above 0 be green. After building the image I scaled it so that it had the same dimensions as the DEM. Then using the rgb_merge function, I combined the two images, having the DEM replace all of the green in the bath image. This produced a combined image with elevation and bath. data.

Brit Hutchinson
for
Scott Wolman
12-20-95

Objective: Enable student, unfamiliar with the process of building a poster, to understand the process using the automated system thus far devised.

The scripts necessary to build a poster are located in makeMaps/scripts.

To build a map which is a subset of the south west data set (–126W to –106 W, 31 N to 43 N) then the file makeMaps/scripts/poster should be used. All of the rgb images for the DEMs should be in the makeMaps/unzipped/ directory and be ungzipped.

After makeMaps/scripts/poster has been executed, shell scripts will appear in the directory. View the script makeMaps/scripts/main.temp and make any changes desired. Then execute that script. The process will then build the dem, project it, convert it into the proper post–script file, select the state lines, build the coordinates for the map, and other various tasks. All of the completed files will be placed in makeMaps/scripts/parts directory. These files can be combined using the makeMaps/scripts/build* scripts. Two types of files can be built, the first is the complete post–script file, the other is a demo file which does not contain the background image and can therefore be edited more quickly. To use these different files, edit the makeMaps/scripts/build* scripts.

Brit Hutchinson
for Scott Wolman
12-20-95

Objective: Enable student, unfamiliar with the process of building earthquake data, to understand the process using the automated system thus far devised.

The scripts necessary to build a poster are located in makeMaps/scripts/parts/Eq.data/eq.build/.

To build an earthquake data set, use makeMaps/scripts/parts/Eq.data/eq.build/run. This file can be edited by adding the desired coordinates to the nawk command. Then execute the run script and observe the comments which are presented. At one point, the script will call xpsview. The xpsviewer screen will appear and when the script is through using it, the window will turn red. You must exit the xpsview manually when this happens. The completed file will appear in the directory as part053___. This should be place in the makeMaps/scripts/parts directory.

The epicenters will be sorted by depth, using the sort command which is located in the run script. If different sorting is desired, the call can be modified in the script.

The colors of the epicenters is determined by the tables in the file makeMaps/scripts/parts/Eq.data/eq.build/table.ps. If different colors are desired, these tables can be modified or replaced.

The raw earthquake data is located in /u6/gisdb/d20lib/seismic/rawdata.

for
Scott Wolman 12-20-95

Projection Script file:

```
cnt = 0;
}
{
    if(cnt == 0){
    a = $1;
    b = $2;
    }
    else
    {
    c = $1;
    d = $2;
    }
    if(cnt == 1){

        theta = atan2((c-a),(d-b)) * 180.0 / 3.1415927;

        printf("%f %f\n",a,b);
        printf("%f %f\n",c,d);
        printf("%f\n",theta);

        cnt = 0;
        }
    else    cnt = 1;
}'    c > d
::::::::::::::
runHawk3
::::::::::::::

nawk '{    a = $1;
           b = $2;
           printf("%f %f\n",a,b);
}' part062_horzStress.ps > tmp
::::::::::::::
runHawk4
::::::::::::::

nawk '{
           a = $1;
           b = $2;
           c = $3;
           d = $4;
           e = $5;
           f = $6;
           g = $7;
           h = $8;
           printf("%f %f %.2f %.1f %s %s %s\n",b,d+a,e,f,g,h);
}' c > f
```

```
::::::::::::::
runHawk3
::::::::::::::

runBH

runHawk

runBH2

runHawk2

paste d part062_horzStress.ps > e

runHawk4

::::::::::::::
runBH
::::::::::::::
#!/bin/csh

setenv PROJ_LIB /usr2/brent/geoview/lib_proj/PROJ.4/nad
setenv PROJ_BIN /usr2/brent/geoview/lib_proj/PROJ.4/bin

# $PROJ_BIN/proj +proj=utm +zone=11 -E file.txt

#_____________________
# From UTM Zone 11 To Lng/Lat
#_____________________
$PROJ_BIN/invproj +proj=utm +zone=11 -E <<EOF
0.0    0.0
EOF

$PROJ_BIN/invproj +proj=utm +zone=11 -E tmp > a
$PROJ_BIN/invproj +proj=utm +zone=11 -f "%20.10f "  tmp > a

::::::::::::::
runBH2
::::::::::::::
#!/bin/csh

setenv PROJ_LIB /usr2/brent/geoview/lib_proj/PROJ.4/nad
setenv PROJ_BIN /usr2/brent/geoview/lib_proj/PROJ.4/bin

$PROJ_BIN/proj +proj=utm +zone=11 b > c

#_____________________
# From UTM Zone 11 To Lng/Lat
#_____________________
$PROJ_BIN/invproj +proj=utm +zone=11 -E <<EOF
0.0    0.0
EOF

$PROJ_BIN/invproj +proj=utm +zone=11 -E tmp > a
$PROJ_BIN/invproj +proj=utm +zone=11 -f "%20.10f "  tmp > a
::::::::::::::
runHawk
::::::::::::::

nawk '{    a = $1;
           b = $2;
           c = b + 1;
                printf("%f %f\n",a,b);
                printf("%f %f\n",a,c);
}' a > b
::::::::::::::
runHawk2
::::::::::::::

nawk '$SIZE<
```

for
Scott Wolman 12-20-95

End of notebook

Brent Hulme

12-20-95