L08 – Generic Mapping Tools (GMT) - Part 4

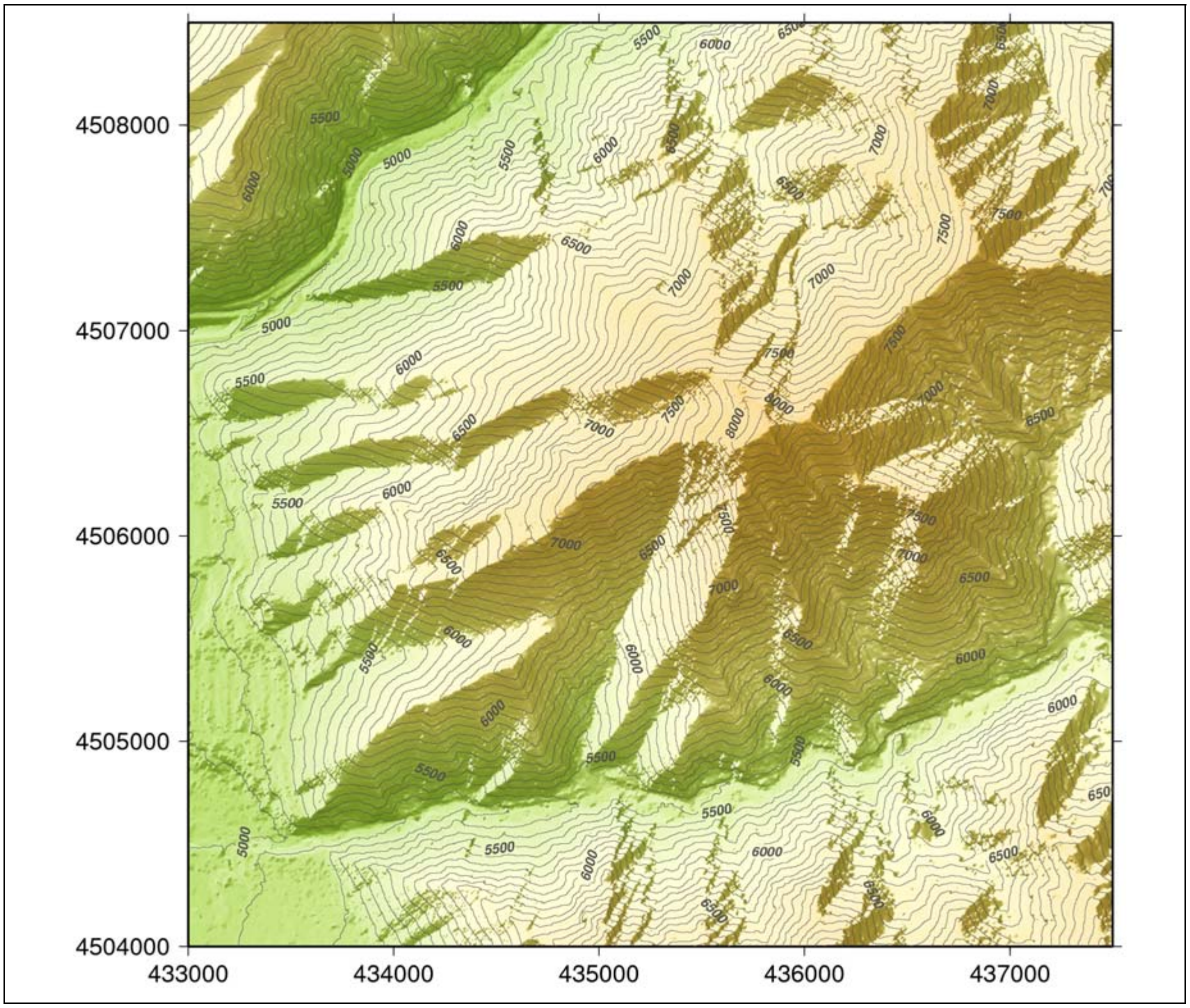
We've come a long way thus far in our GMT usage. We really have barely even touched on some of GMTs functionality. But, there are a few more useful tricks I would like to share with you before we move on to our next topic. We will continue to look at local scale applications in our examples, but note that the commands we talk about here are also applicable to global scale apps.

1. Contouring in GMT – Topo Maps

To put it simply contouring is a pain. Not that it's difficult to generate contour lines. But, it can be difficult to get everything to look nicely. One just needs patience and a lot of trial and error. So, how do we do it? Let's use our data set from the Mt. Olympus region DEM. If you don't have the file handy you can grab a new DEM file from the website. The next example shows a simple way to generate contour lines. Our primary command here is **grdcontour**.

```
#!/bin/csh

# Plot region zoomed around Grandeur Peak
set xmin = 433000
set xmax = 437500
set ymin = 4504000
set ymax = 4508500

# Make cpt
makecpt -CColor_DEM.cpt -Z -T3000/15000/1000 >! grandeur.cpt

# Make intensity file
grdgradient Mt_Olympus.grd -A0/270 -GMt_Olympus.gradients -Ne0.6 -V

# Grid image
grdimage Mt_Olympus.grd -R${xmin}/${xmax}/${ymin}/${ymax} -Jx1:24000 \
  -B1000g10000nSeW -Cgrandeur.cpt -V -P -K -IMt_Olympus.gradients \
  -Sb -Ei600 >! grandeur.ps

# plot 100 foot contour lines
grdcontour Mt_Olympus.grd -Jx -R -W1/80/80/80 -C100 -P -S4 -O \
 -A500+f3+k80/80/80+s8t -G2i/10 -Djunk -V >> grandeur.ps

rm junk_*.xyz   #get rid of temporary contour files

# remove excess files
rm grandeur.cpt
rm Mt_Olympus.gradients

gs -sDEVICE=x11 grandeur.ps
```

Executing this script will result in the following image:



Here, I went for less emphasis on the contour lines. But, now is your chance to play around. Try out the above script. There are a couple of important flags to keep in mind:

- **-C** – the easiest way to make contours is to supply the contour interval with this flag. In this example we set it at **–C100** which means make contours at 100 foot intervals. Try out 40 and 200 foot intervals.
- **-G** – The primary use of this flag is how far to space the contour labels. In this example I said put them every 2 inches. But, you should also try closer or further a part.
- **-A** – This flag has all kinds of options. I chose:

  500 - only label contours that are even multiples of 500 feet (i.e., draw labels on 5000, 5500, 6000, etc., feet).

  +f3 - use font number 3 (Helvetica-BoldOblique) to print contour labels

  +k100/100/100 – plot labels with color 100/100/100

  +s8 – use an 8 point font size for labels

As you can see there are many options with the **–A** flag. Try playing around with them now!

The previous example uses the simplest way to generate contour intervals. But, sometimes you may want to contour specific intervals that aren't evenly spaced. For example, perhaps I want to draw contours every 200 feet with heavy lines, but every 40 feet with lighter lines. Because 40 is a multiple of 200, I don't want to draw the 40 foot interval lines at every point. So, what I can do is generate a file that specifies which contour intervals to plot. The file would look as follows:

```
40    C
80    C
120   C
160   C
240   C
280   C
…     …
```

Where we note that we didn't supply a contour at the 200 foot level. The online material contains a file called **contours40.dat** that can be used as a test. Using this file one would specify the contour interval by the name of the file:

```
-Ccontours40.dat
```

2. Vector Data

If by chance you have been browsing the Utah GIS data resources you may have noticed that there are tons of interesting vector data resources. Unfortunately, all of these data are written in ESRI shapefile format (**.shp**). So, how do we deal with these data? That will be left as a future exercise for the reader.

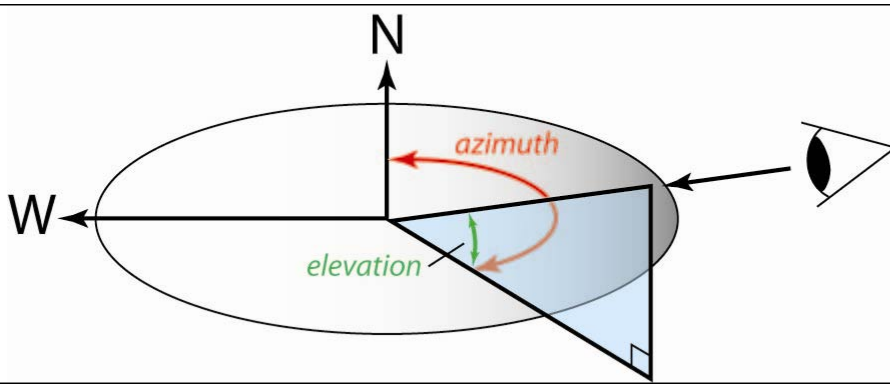
3. 3D Perspectives

GMT can also produce 3D perspective views using the **grdview** command. Revisiting our plot of Grandeur Peak:

```
#!/bin/csh

# Plot region zoomed around Grandeur Peak
set xmin = 433000
set xmax = 437500
set ymin = 4504000
set ymax = 4508500

# Make cpt
makecpt -CColor_DEM.cpt -Z -T3000/15000/1000 >! grandeur.cpt

# Make intensity file
grdgradient Mt_Olympus.grd -A0/90 -GMt_Olympus.gradients -Ne0.6 -V

# Gridview Azimuth/Elevation #1
grdview Mt_Olympus.grd -R${xmin}/${xmax}/${ymin}/${ymax} -Jx1:24000 \
  -JZ4i -B1000g10000nSeW -Cgrandeur.cpt -V -P \
  -IMt_Olympus.gradients -E180/30 -Qs >! grandeur.ps
```

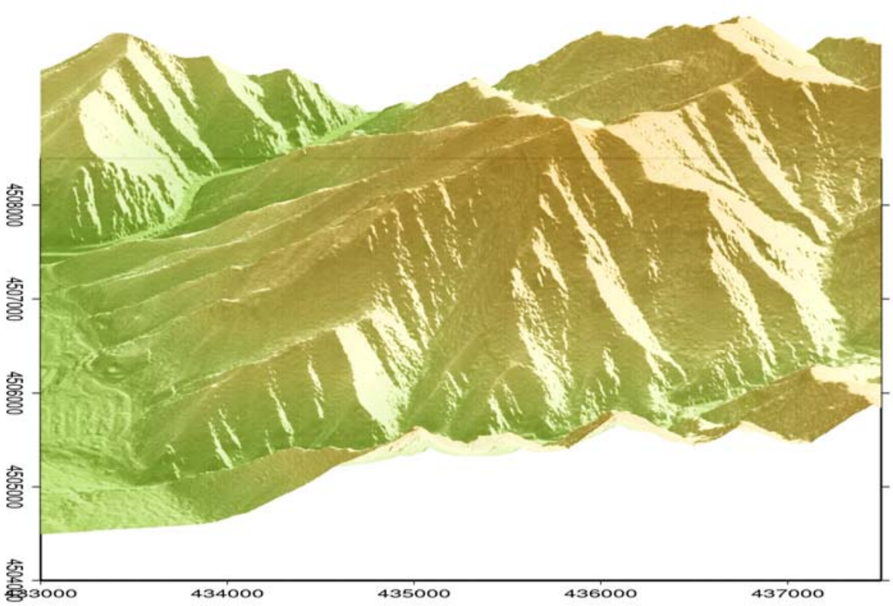
```
# Grdview Azimuth/Elevation #2
grdview Mt_Olympus.grd -R${xmin}/${xmax}/${ymin}/${ymax} -Jx1:24000 \
  -JZ4i -B1000g10000nSeW -Cgrandeur.cpt -V -P \
  -IMt_Olympus.gradients -E260/5 -Qs >! grandeur2.ps

# remove excess files
rm grandeur.cpt
rm Mt_Olympus.gradients

gs -sDEVICE=x11 grandeur*.ps
```

With grdview we need to specify both the Azimuth from which we are viewing the plot and the elevation as shown in the following diagram. As usual **azimuth** is measured clockwise from North and **elevation** is measured as the positive angle upwards from horizontal.



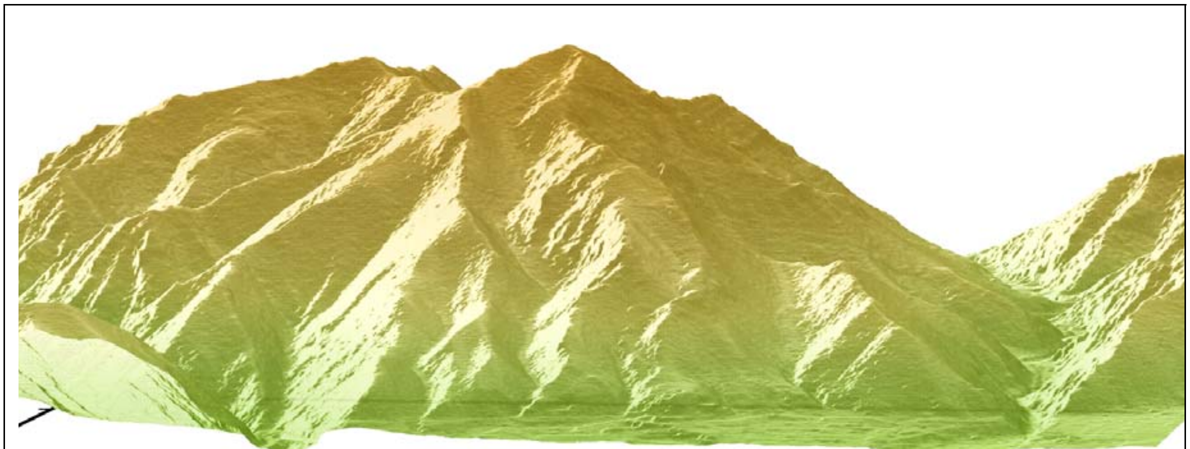
We specify these parameters with the **–E** flag, which has the syntax: **–Eazimuth/elevation**.

In the example script we first look at Grandeur Peak from an azimuth of 180° - or from due South. This is similar to our other views, where North is towards the top of our plot.



View of Grandeur Peak from an azimuth of 180° and an elevation of 30°.

The above plot is what Grandeur Peak might look like from the view of an airplane. However, a more familiar view to me is looking at it as I drive East up 3300 South. To mimic this view I set our azimuth as 260° and the elevation as 5°. Note that elevation must be positive in **grdview**.



View of Grandeur Peak from an azimuth of 260° and an elevation of 5°.

4. Raster Overlays (grdview)

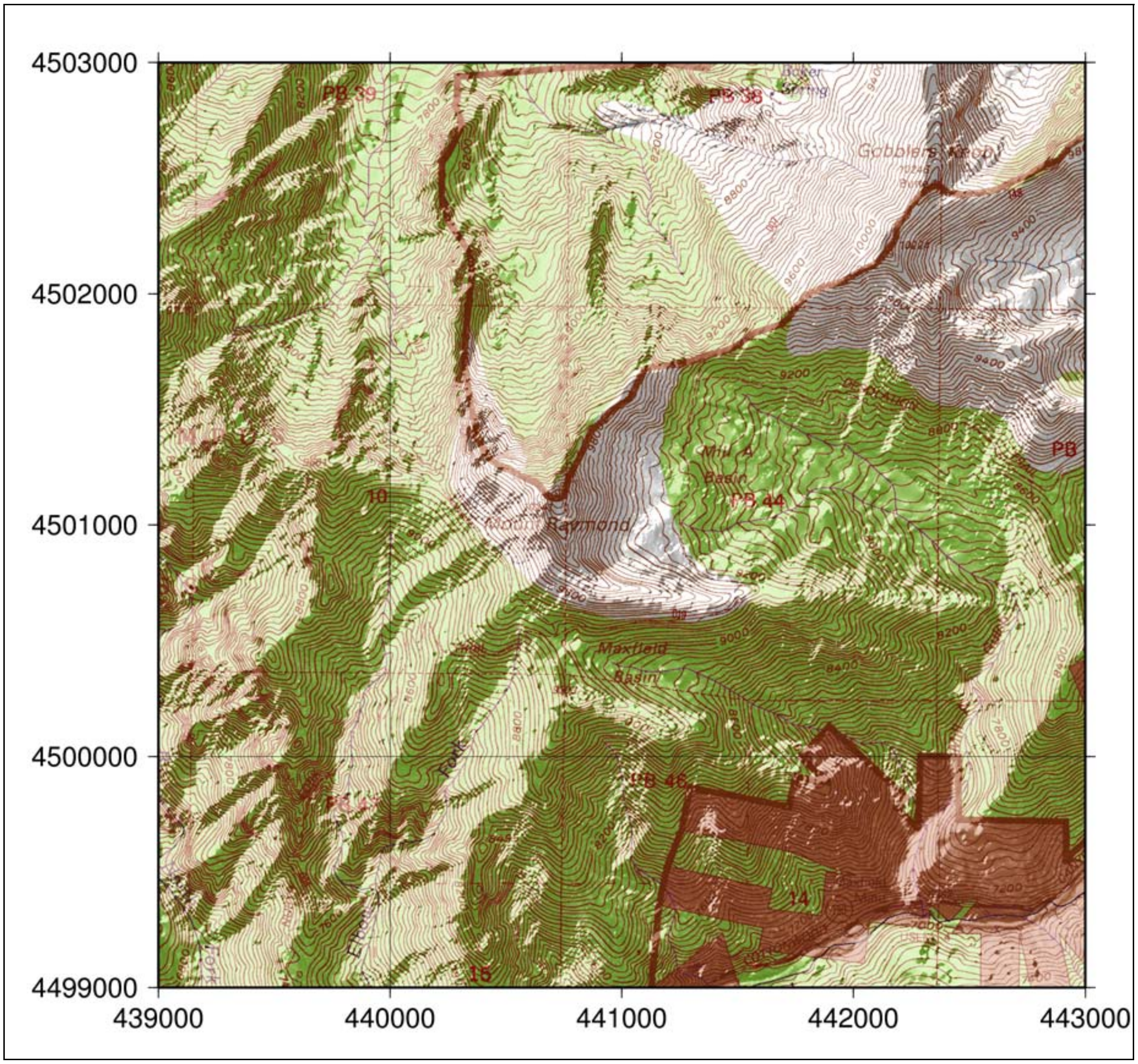
There is also another important feature supported by **grdview** in addition to being able to draw perspective views. Namely, it supports overlaying Raster images. However, it isn't always exceptionally the easiest thing to do, it is still useful when you want to script the plotting and/or generate several images.

To begin with, what kind of raster images can be draped? Pretty much anything you want. So, let's take as an example draping a topo quad. For the state of Utah we can get our topo quad images from http://gis.utah.gov.

Go to: **GIS Data & Resources** > **SGID, Utah GIS Data** > **Raster GIS Data** > **USGS Topographic Maps DRGs** > **By Quad Name, USGS Scanned Topographic Maps (1:24000, GeoTIFF)**

For the following example we use the **MOUNT AIRE** (**Q1321**) quad.

We will generate the following shaded relief image centered on Mount Raymond.



The following script demonstrates the process.

```
#!/bin/csh

# Which DRG files to use
set prefix = q1321_DRG24k-c   #leave off the extensions
set geotif = ${prefix}.tif    #actual .tif image
set reference = ${prefix}.tfw #reference positions


# Set bounds of geotif image
set lon1 = `awk 'NR == 5 {print $1}' $reference`
set lat2 = `awk 'NR == 6 {print $1}' $reference`
set x_inc = `awk 'NR == 1 {print $1}' $reference`
set y_inc = `awk 'NR == 4 {print $1}' $reference`

set lon2 = `awk 'NR == 1 {print ('$lon1'+'$x_inc'*8752)}' $reference`
set lat1 = `awk 'NR == 1 {print ('$lat2'+'$y_inc'*11447)}' $reference`

echo $lon1 $lon2 $lat1 $lat2


# Convert collarless tif to .ras using ImageMagick
convert $geotif topo.ras

# Convert .ras to .grd
gmt2rgb topo.ras -I1 -F -Gtopo_%c.grd

# Its easiest to set the bounds of the .grd files with grdedit
grdedit topo_r.grd -R${lon1}/${lon2}/${lat1}/${lat2}
grdedit topo_g.grd -R${lon1}/${lon2}/${lat1}/${lat2}
grdedit topo_b.grd -R${lon1}/${lon2}/${lat1}/${lat2}

# Make intensity file
grdgradient Mt_Olympus.grd -A0/270 -GMt_Olympus.gradients -Ne0.6 -V

# Make map bounds centered on Mt. Raymond
set xmin = 439000
set xmax = 443000
set ymin = 4499000
set ymax = 4503000

# Gridview it up
grdview Mt_Olympus.grd -R${xmin}/${xmax}/${ymin}/${ymax} -Jx1:24000
  -JZ4i -B1000g10000nSeW -V -P -IMt_Olympus.gradients \
  -E180/90 -Qi600 -Gtopo_r.grd,topo_g.grd,topo_b.grd >! mtraymond.ps

rm topo_r.grd topo_g.grd topo_b.grd
rm Mt_Olympus.gradients topo.ras

gs -sDEVICE=x11 mtraymond.ps
```

So what did we do? The following pointers may be helpful:

- Note that the file ***.tfw** contains the UTM positions of the upper-left corner of the image. This file also contains the increment between each pixel in the *x*- and *y*- direction.

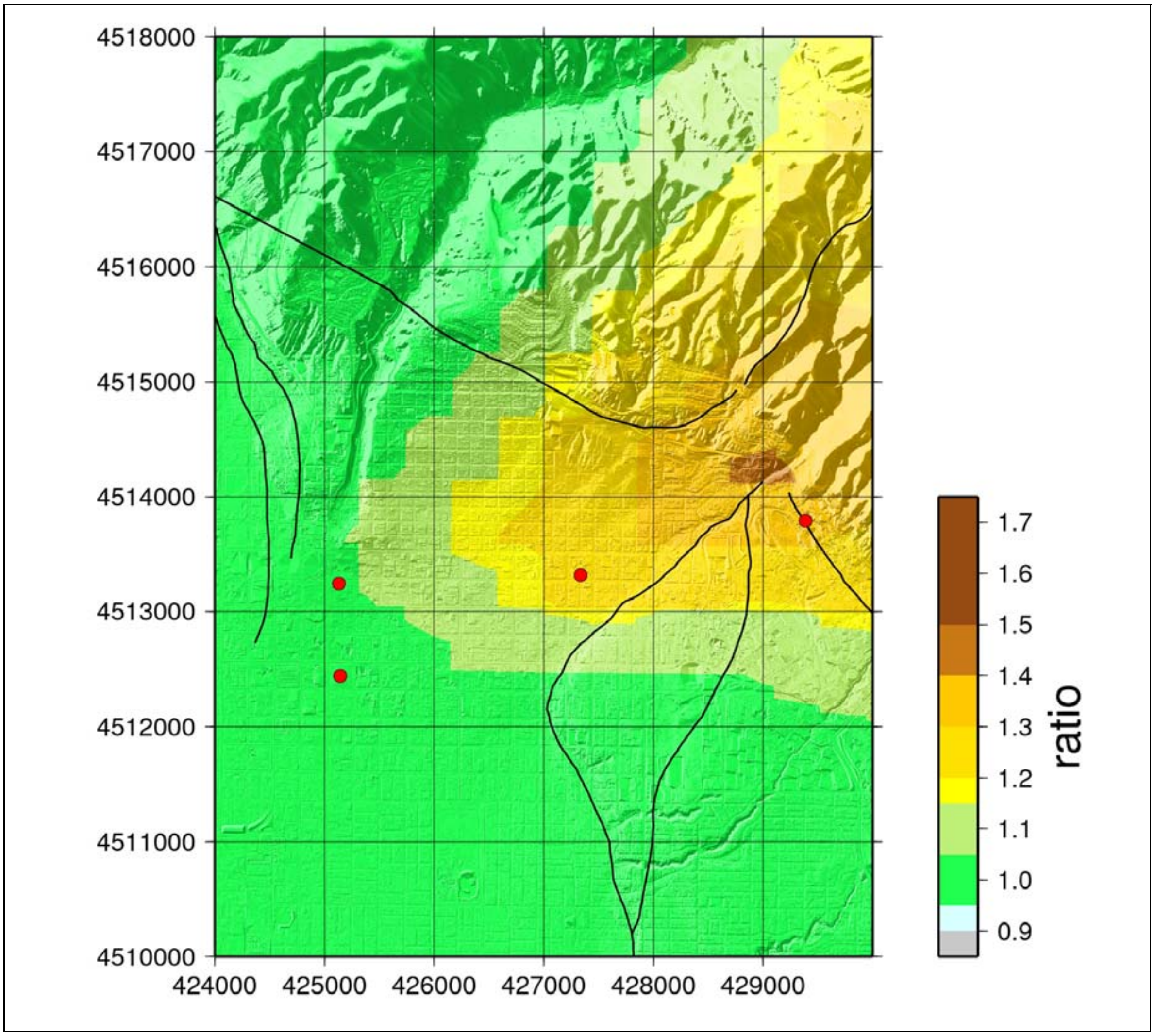
- The GMT codes can only read Sun raster image files (**.ras**). Fortunately it is exceptionally simple to convert from **.tif** to **.ras** with ImageMagick's convert utility.

- GMTs command **gmt2rgb** is used to convert the Raster image (**.ras**) into a series of three **.grd** files. The three files contain the Red, Green, and Blue color data from the raster image.

- I typically find it is easiest to change the boundaries of the **.grd** files using GMTs **grdedit** command.

- Lastly, we just drape the **.grd** files with the **grdview** command.

Once you've learned how to drape images you can add just about any type of information you want to a plot. As a final example consider the following plot I made that shows peak ground acceleration due to a hypothetical earthquake on the Wasatch Fault. What is important is not so much the peak acceleration values, but the point that anything you can color up can easily be draped in a GMT image.



5. Animations

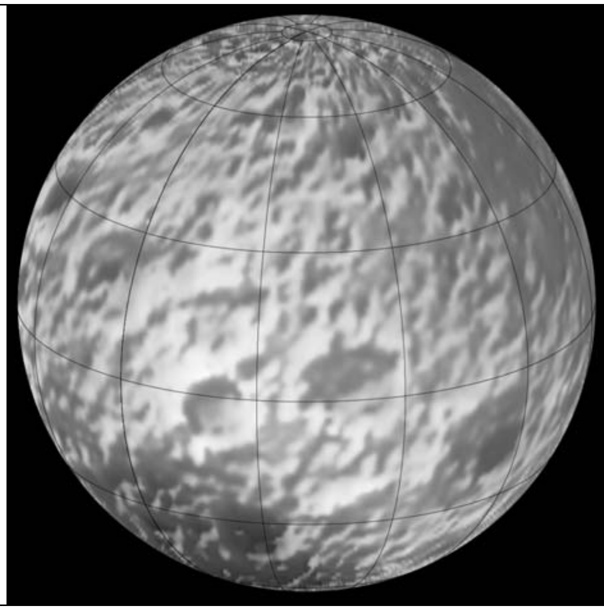
Generating Animations isn't exactly a GMT task. However, it is a natural extension since all plots made in GMT are scripted. Hence with some looping and variables we can quickly generate several different plots that can be strung together into an animation.

There are several ways to generate animations. Here we will discuss the primary technique and show two different ways to finalize the animation product: (1) Using ImageMagick to generate animated gifs (**.gif**), or (2) Using ImageReady to produce Quicktime (**.mov**) files.

All animations start with the same need – animation frames. Here we will generate animation frames for a very simple example: the rotating Lunar surface.

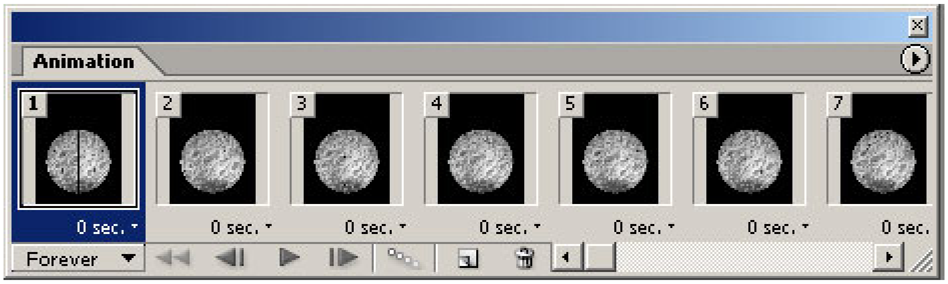
```
#!/bin/csh
# set up some initial values
#---------------------------------------------------------------------#
set gfile = lunar_topo.grd                                 # lunar topo
gmtset PAGE_COLOR 0/0/0                                    # color
makecpt -Cgray -Z -T-9000/8000/200 >! lunar.cpt            #.cpt file
grdgradient $gfile -A0/270 -Glunar.gradients -Ne0.6 –V     # gradients
#---------------------------------------------------------------------#

# start looping through longitudes
#---------------------------------------------------------------------#
@ lon = 0
while ($lon <= 360)

# create ordered numbering scheme for output files
if ($lon < 10) then
  set ofile = lunar_00${lon}.ps
  set jfile = lunar_00${lon}.jpg
else if ($lon < 100) then
  set ofile = lunar_0${lon}.ps
  set jfile = lunar_0${lon}.jpg
else
  set ofile = lunar_${lon}.ps
  set jfile = lunar_${lon}.jpg
endif

grdimage $gfile -R0/360/-90/90 -JG${lon}/20/6.5i -Bg30 \
  -Clunar.cpt -V -P -Sb -Ei300 -Ilunar.gradients >! $ofile

convert -compress Lossless -density 150x150 $ofile $jfile
rm $ofile

@ lon = $lon + 5
end
# end looping through longitudes
#---------------------------------------------------------------------#

# make an animated gif using ImageMagick
convert -adjoin -delay 10 -loop 0 *.jpg lunar.gif

rm lunar.cpt lunar.gradients
```

In the above example we use **grdimage** to generate images of the lunar surface using the supplied file: **lunar_topo.grd**. We actually make a loop through longitude and just change the projection center to a different longitude. Note, the most important thing here is our naming scheme of the output images. These need to named such that a **ls** command lists them in the correct order. Finally, each **.ps** image in converted to a **.jpg** image and all of these jpegs are combined into an animated **.gif** file using ImageMagick. One snapshot is shown below.



In running the above example we are left with a number of **.jpg** images. This is on purpose so we can see how to stitch them together into a **Quicktime** movie. Adobe Photoshop also comes with a program called **Adobe ImageReady**. You will have to be on a Windows or Mac OS to access this. To create a movie file in Image Ready do as follows:

- Copy a directory that contains all of the **.jpg** images over to the Windows or Mac machine.
- Launch Adobe ImageReady.
- In ImageReady do: **File** > **Import** > **Folder as Frames…** and select the folder containing your **.jpg** images.
- ImageReady opens up all the files and orders them by frame number. You will see the frame order in the Animation Window:



- Hit the triangle in the upper right corner of the Animation Window > **Select All Frames**
- Set a delay time between frames (e.g., 0.1 sec).
- Select: **File** > **Export** > **Original Document**
- Choose: **Quick Time Movie (.mov)**; Set the quality from **Medium** to **Best**.

Now, all you need to do is launch the QuickTime Player to watch your animation.

6. Some Details

As a final note on GMT, there are a ton of values that GMT has initially set as defaults. Don't believe me? Then type:

```
>> gmtdefaults –D
```

You may have noticed that in some of our example scripts we actually changed some of these. For example, in our script to plot the Lunar surface we made the default page background black with the following line:

```
gmtset PAGE_COLOR 0/0/0
```

Sometimes it is essential in GMT to change a setting. For example, when we used the **grdproject** command do you know what ellipsoid was being used for the projection? It was WGS-84. But, perhaps the map data were in the NAD-1927 datum. Then one needs to change the ellipsoid being used. In this case the Clarke-1866 ellipsoid would be needed and one would type:

```
gmtset ELLIPSOID Clarke-1866
```

To see what options you have in setting the defaults just type:

```
>> man gmtdefaults
```

There is one more useful item to discuss here that deals with page size. The following file exists on your system: **$GMTHOME/share/gmtmedia.d**. This file contains paper sizes that one may use. Of immediate importance is that it is customizable. So, if you noticed that when we made our animation the 8.5 × 11 page size wasn't exactly ideal. Instead we want something more akin to that of a screen size. A nice thing to do is add the following line to the file **gmtmedia.d**

```
snap        672         504
```

This specifies a paper type called **snap**. Now, if I want to use this paper size I can change the following GMT default:

```
gmtset PAPER_MEDIA snap
```

Alas, you have a page size that is much more suited for creating animations.

7. Homework

1) Generate a plot showing 3D perspective views of Antelope Island. The plot should contain two panels showing the island from two different viewpoints. As always be sure to include all relevant scale bars.

2) Pick an appropriate 3D perspective view of Antelope Island and generate an animation that mimics changing illumination from sunrise to sunset. That is, the illumination azimuth should start from the East and move to the West.