

MakeHillshade.nb

Written July 6, 2014. Last revised August 27, 2022.

This code is copyright © 2010-2022 by Vincent S. Cronin.

```
In[1]:= startTime = AbsoluteTime[];
```

Introduction

The purpose of this notebook is to create hillshade images (from an ASCII DEM) that are illumined in a user-specified direction. The user can also specify the elevation of the illumination, between 0 degrees (at the horizon) and 90 degrees (at the zenith, directly overhead).

Input Data

Import the Earthquake Data File

This code reads an external data file that is in the form of an ASCII DEM file (e.g., and Arc ASCII-GRID file), with the .dat file-type designator. In the example below, a file named **DogValleyFullNAD83.-dat** is located on the desktop of Vince's computer.

To modify this code so that it can be used for your dataset, do the following.

1. Start with the following as an input line (in the gray box, below):

```
initialDataFile = Import[];
```

2. Put your cursor between the square brackets in the **Import[]** statement and click to establish the insertion point we will need in the next step.

3. Go to the **Insert** menu, select **File Path**, navigate to the correct input data file and choose it, and the correct file path will be inserted at the cursor. In this example, the Excel file is located on the desktop of Vince's computer, and so *Mathematica* will insert the path **"/Users/vince/Desktop/DogValleyFullNAD83.-dat"** between the brackets. The path will be different for every different file, and on every different computer. The result should look something like the following:

```
initialDataFile = Import["/Users/vince/Desktop/DogValleyFullNAD83.-dat"];
```

You can make sure it is an input line by clicking on that line's bracket on the far right edge of this

window, going to the **Format** menu, selecting the **Style** submenu, and then choosing **Input**

```
In[2]:= initialDataFile = Import["/Users/vince/Desktop/DogValleyFullNAD83.dat"];
```

This won't work unless you follow directions and add the appropriate input line above this line.

Your input line should look something like the example:

```
initialDataFile = Import["/Users/vince/Desktop/DogValleyFullNAD83.-
dat"];
```

Input Other Information About the Map

1. What is the geometric center of the data file, expressed in latitude and longitude?

The reason for this input is that the orientation of the vertical edges of a UTM grid (UTM north - south) are only aligned with true north - south lines in the geographic coordinate system along the central meridian of a UTM zone. We want to know the divergence between the UTM north - south line at the center of the dataset and the geographic north - south line, a quantity known as the grid declination or grid divergence.

North latitudes are positive values, south are negative. East longitudes are positive values, west are negative. Latitudes and longitudes are expressed as decimal degrees.

```
In[3]:= midLatitude = 39.405;
```

```
In[4]:= midLongitude = -120.175;
```

2. What is the central meridian (centralMeridian) of the UTM zone in which the map area is located?

The central meridian is expressed as a longitude in decimal degrees. East longitudes are positive values and west are negative.

UTM zones are 6 ° of longitude wide, and the UTM central meridians in the western United States all have negative (west) longitudes. Appropriate values include -129 °, -123 ° (for Tahoe - area studies in California), -117 °, -111 °, *et cetera*.

```
In[5]:= centralMeridian = -123;
```

3. From what direction and elevation angle do you want the hillshade image illuminated?

The **sunAzimuth** is the compass direction or azimuth from which the light source is shining on the topography, expressed in degrees. So, for example, a sunAzimuth of 90 ° means that the sun is shining from the east. Compass azimuth is measured from north (0 ° azimuth) clockwise, so east is 90 °, south is 180 °, and west is 270 °. The input data are expressed in the conventional geographic coordinate system, in which Earth's spin axis defines true north (toward the Arctic Ocean) and true south (toward Antarctica).

```
In[6]:= sunAzimuth1 = 130;
```

The **sunElevation** is the vertical angle measured up from horizontal to the light source, **expressed in degrees**. So, for example, a sunAzimuth of 0° means that the light is shining from the horizontally.

```
In[7]:= sunElevation = 25;
```

Other relevant input tasks

1. Change the file path and output file name in the blue boxes near the end of this code.

Primary Computation

```
In[8]:= ncols = initialDataFile[[1, 2]];
```

```
In[9]:= nrows = initialDataFile[[2, 2]];
```

```
In[10]:= xllcorner = initialDataFile[[3, 2]];
```

```
In[11]:= yllcorner = initialDataFile[[4, 2]];
```

```
In[12]:= nodeSpacing = initialDataFile[[5, 2]];
```

```
In[13]:= xMax = xllcorner + (ncols * nodeSpacing);
```

```
In[14]:= yMax = yllcorner + (nrows * nodeSpacing);
```

```
In[15]:= upperLeftCorner = {Round[xllcorner], DecimalForm[yMax]};
```

```
In[16]:= lowerLeftCorner = {Round[xllcorner], IntegerPart[yllcorner]};
```

```
In[17]:= lowerRightCorner = {DecimalForm[xMax], IntegerPart[yllcorner]};
```

```
In[18]:= upperRightCorner = {DecimalForm[xMax], DecimalForm[yMax]};
```

Adjustment for input geographic coordinates based on true north to the relevant UTM grid north

The **gridNorthAdjustment** is the angle between true north and grid north at/near the center of the map area. If grid north is found by a clockwise rotation from true north, the sign of the **gridNorthAdjustment** is negative; otherwise, it is positive. The **gridNorthAdjustment** is approximately equal to $(\text{zoneMeridian} - (\text{longitude at center of map area})) * \text{Sin}[\text{latitude at center of map area}]$.

```
In[19]:= gridNorthAdjustment = ArcTan[
```

```
    Tan[(midLongitude - centralMeridian) Degree] * Sin[midLatitude Degree]] * (180 / π);
```

Convert the light direction from compass direction to a standard geometric reference frame, with the angle expressed in radians.

```
In[20]:= geomDir = If[(90 - (sunAzimuth1 + gridNorthAdjustment)) < 0,
  (360 + (90 - (sunAzimuth1 + gridNorthAdjustment))),
  ((90 - sunAzimuth1 + gridNorthAdjustment))];
```

```
In[21]:= lightDir1 = geomDir * (π / 180);
```

Convert the light elevation from degrees to radians.

```
In[22]:= sunElev = (sunElevation * π) / 180;
```

Make the first hillshade image

```
In[23]:= demImgFile = Table[initialDataFile[[i + 6, j]], {i, nrows}, {j, ncols}];
```

```
In[24]:= Clear[initialDataFile];
```

```
In[25]:= elev1 = Table[demImgFile[[i, j]], {i, nrows, 1, -1}, {j, ncols}];
```

```
In[26]:= hillshadeMap1 =
  ReliefPlot[elev1, ColorFunction -> None, LightingAngle -> {lightDir1, sunElev}];
```

Bill Haneberg contributed the "ColorFunction -> None" portion of the input line above, which changes the relief plot into a hillshade.

Make the second hillshade image

```
In[27]:= lightDir2 = lightDir1 + π;
```

```
In[28]:= sunAzimuth2 = If[(sunAzimuth1 > 180), (sunAzimuth1 - 180), (sunAzimuth1 + 180)];
```

```
In[29]:= hillshadeMap2 =
  ReliefPlot[elev1, ColorFunction -> None, LightingAngle -> {lightDir2, sunElev}];
```

Output

```
In[30]:= outFile1 = Show[hillshadeMap1];
```

```
In[31]:= Show[%]  
Out[31]=
```



```
In[32]:= outFile2 = Show[hillshadeMap2];
```

```
In[33]:= Show[%]
```

```
Out[33]=
```



```
In[34]:= metadata = {"upper left corner", upperLeftCorner}, {"upper right corner",
    upperRightCorner}, {"lower left corner", lowerLeftCorner},
    {"lower right corner", lowerRightCorner}, {"sunAzimuth1", sunAzimuth1},
    {"sunAzimuth2", sunAzimuth2}, {"sunElevation", sunElevation},
    {"grid divergence in degrees", gridNorthAdjustment}};
```

```
In[35]:= textList = MatrixForm[metadata]
```

```
Out[35]//MatrixForm=
```

| | |
|----------------------------|----------------------|
| upper left corner | {734 105, 4385486.} |
| upper right corner | {759475., 4385486.} |
| lower left corner | {734 105, 4 357 769} |
| lower right corner | {759475., 4 357 769} |
| sunAzimuth1 | 130 |
| sunAzimuth2 | 310 |
| sunElevation | 25 |
| grid divergence in degrees | 1.79417 |

```
In[36]:= outFile3 = %;
```

Export data and image files

The file created in the next line is JPEG image that contains the graphic showing the hillshade image developed from the input DTM.

IMPORTANT NOTE: The user must supply a name for the output file that is different from existing file

names, or else the existing files may be over-written.

```
In[37]:= Export["/Users/vince/Desktop/DogValleyFullNAD83-1.png", outFile1];
```

```
In[38]:= Export["/Users/vince/Desktop/DogValleyFullNAD83-2.png", outFile2];
```

```
In[39]:= Export["/Users/vince/Desktop/DogValleyFullNAD83-metadata.xlsx", outFile3];
```

```
In[40]:= ClearAll[outFile1, outFile2, outFile3];
```

How long did this program take to run, in minutes?

```
In[41]:= minutesForProcessing = (AbsoluteTime[] - startTime) / 60
```

```
Out[41]=  
0.68781730
```