

## Modeling Terrain Elevation Data in MATLAB

Costăchioiu Teodor<sup>1</sup>, Vasile Lăzărescu<sup>1</sup>, Rodica Constantinescu<sup>1</sup>

**Abstract** - Topography is basic to many earth surface processes. It is used in analyses in ecology, hydrology, agriculture, climatology, geology, pedology, geomorphology, and many others, as a means both of explaining processes and of predicting them through modeling. Through this paper we consider the implementation of an algorithm for representation of real elevation data in MATLAB. This algorithm takes the DEM (Digital Elevation Model) data and constructs a 1201x1201 elevation matrix corresponding to a tile of 1 degree latitude and degree longitude. The DEM data was obtained by NASA's SRTM mission in 2000 and was made publicly available in June 2004. This offers an almost worldwide coverage of elevation data that can be further analyzed in MATLAB, for example for finding the pattern for water flow accumulation.

**Keywords:** elevation data, topography

### I. INTRODUCTION

When analyzing elevation data from the world, one may find data that is outdated, inaccurate or nonexistent. Another problem that comes when researching worldwide elevation data is the fact that each country is using different representation systems for topographic data so many times this data are very difficult to process due to lack of standardization.

In February 2000, by a joint effort of National Geospatial-Intelligence Agency (NGA), the National Aeronautics and Space Administration (NASA), the German Space Agency (DLR) and the Italian Space Agency (ASI), the SRTM (Shuttle Radar Topography Mission) was launched. During an 11 day period a special radar system onboard of Space Shuttle Endeavour recorded elevation data for almost 80% of the Earth surface, from 56S to 60N, using radar interferometry. The SRTM instrument consisted of the Spaceborne Imaging Radar-C (SIR-C) hardware set modified with a Space Station - derived mast and additional antennae to form an interferometer with a 60 meter long baseline (see figure 1). Synthetic aperture radars are side-looking instruments and acquire data along continuous swaths.

The SRTM swaths extended from about 30 degrees off-nadir to about 58 degrees off-nadir from an altitude of 233 km, and thus were about 225 km wide.

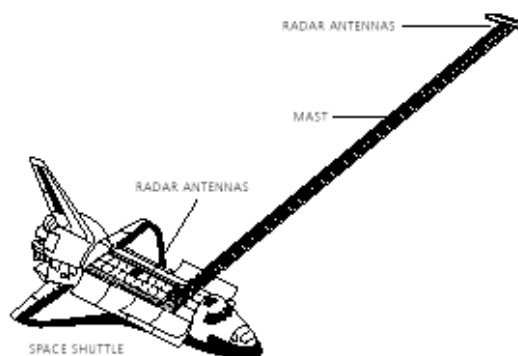


Fig.1 The NASA Shuttle taking SRTM data

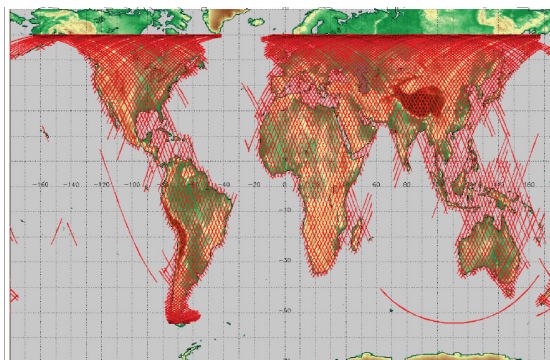


Fig.2 The SRTM Mission coverage

During the data flight the instrument was operated at all times the orbiter was over land and about 1000 individual swaths were acquired over the ten days of mapping operations (see figure 2).

Length of the acquired swaths range from a few hundred to several thousand km. Systematic interferometric data were collected for 222.4 consecutive hours. The instrument operated almost flawlessly and imaged 99.96% of the targeted landmass at least one time, 94.59% at least twice and about 50% at least three or more times. The goal was to image each terrain segment

<sup>1</sup> Chair of Applied Electronics and Informatics Engineering, University POLITEHNICA of Bucharest, No. 1-3 Iuliu Maniu, 061071, Bucharest, Romania, email: c\_tedor@yahoo.com

at least twice from different angles (on ascending, or north-going, and descending orbit passes) to fill in areas shadowed from the radar beam by terrain. All data was processed in the same way, thus offering a global, standardized elevation map. Worldwide DEM data is available for download by anonymous ftp to: <ftp://e0srp01u.ecs.nasa.gov>.

## II. BACKGROUND

SRTM data are organized into individual rasterized cells, or tiles, each covering one degree by one degree in latitude and longitude. Sample spacing for individual data is 3 arc-seconds, referred to as SRTM3 or "90 meter" data, 1 arcsec is 1/3600 of a degree; on the Earth at medium geographical latitudes 1 arcsec is about 25m.

The names of individual data tiles refer to the longitude and latitude of the lower-left (southwest) corner of the tile. For example the coordinates of the lower-left corner of tile N45E025 are 45 degrees north latitude and 25 degrees east longitude. To be more exact, these coordinates refer to the geometric center of the lower left sample, which in the case of SRTM3 data will be about 90 meters in extent.

SRTM3 data are sampled at three arc-seconds and contain 1201 lines and 1201 samples with the rows at the north and south edges as well as the columns at the east and west edges of each cell overlap and are identical to the edge rows and columns in the adjacent cell. The DEM file (extension ".hgt") is provided as 16-bit signed integer data in a simple binary raster. There are no header or trailer bytes embedded in the file.

The data is stored in row major order (all the data for row 1, followed by all the data for row 2, etc.). Byte order is Motorola ("big-endian") standard with the most significant byte first. Since they are signed integers elevations can range from -32767 to 32767 meters, encompassing the range of elevation to be found on the Earth.

These data also contain occasional voids from a number of causes such as shadowing, phase unwrapping anomalies, or other radar-specific causes. Voids are flagged with the value -32768.

All elevations are in meters referenced to the WGS84/EGM96 geoid as documented at <http://www.NGA.mil/GandG/wgsegm/>.

Relatively little work on SRTM accuracy has been published to date. In most cases accuracy is better than the 10-16 m objective of the SRTM project (NASA 2002).

## III. METHOD

Data is read into Matlab using the following commands:

```
fid=fopen(filename,hgt,'r','b')
A=fread(fid,[1201, 1201],'*int16');
fclose(fid);
```

This opens the file *filename.hgt* for reading in binary format and fills a 1201 x 1201 matrix with the readed data. Keeping count that the **fread** command fills the matrix by columns (column 1 first, then column 2, etc) but the data in the original file is organized by row order, we need a 90 degrees rotation of this matrix (see figure 3).

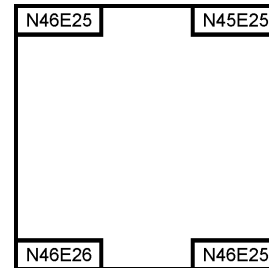


Fig.3. Corner coordinates after reading the DEM file

As a result of this rotation, the matrix is correct from the longitude point of view, but the latitudes are reversed (see figure 4).

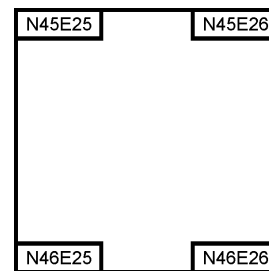


Fig.4 Corner coordinates after rotation of the matrix

A further vertical flip of the matrix is required to put the data into geographically correct form (the first data that is read corresponds to the point N46 E025 – see figure 5).

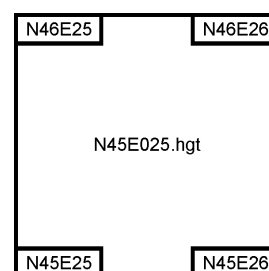


Fig. 5 The final form of the elevation matrix

We can easily verify if the import of the elevation data is correct by comparing the last row/column of our matrix with the first row/column in the adjacent matrix (see figure 6 and table 1).

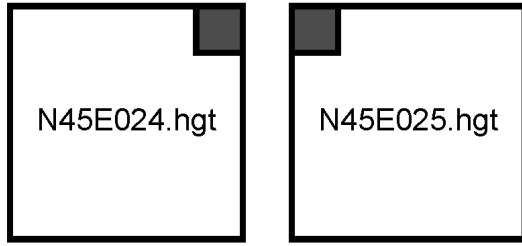


Fig.6 Comparison of the corners of two adjacent DEM files

For example, comparing a 3x4 area of the upper right corner of N45E024.hgt with the upper left corner of N45E025.hgt shows the following values:

Table 1 Comparison of the corners of two adjacent DEM files

N45E024 .hgt			N45E025 .hgt		
574	578	<b>576</b>	<b>576</b>	570	565
571	576	<b>578</b>	<b>578</b>	576	569
568	572	<b>577</b>	<b>577</b>	580	576
567	572	<b>581</b>	<b>581</b>	586	585

We can see the overlapping of data between these two data sets.

This feature is extremely important when working with data that spans over two or more DEM files, because we need to eliminate duplicate rows or columns

#### IV. RESULTS

1. Data voids in the file are flagged by the value -32768 so we can count the number of errors in the elevation file. For our example, in the file N45E025.hgt there are 951 errors, corresponding to 0,6593 % of the total data set.

2. Determining the maximum elevation in file N45E025 shows the value  $H_{max}=2498m$ , corresponding to position  $[xpos,ypos]=[666,549]$  in our data set. From this we can calculate the latitude and longitude for the highest peak, keeping count that there are 1200 samples in a degree (the 1201-th overlapping with the next degree), using equations (1) and (3):

$$Lat_m = \frac{1200 - xpos - 1}{1200} + 45 \quad (1)$$

$$Lat_m = 45.4450^\circ N \quad (2)$$

where  $Lat_m$  is the latitude of the point with the maximum elevation and  $xpos$  is the position of that point in the elevation file.

$$Lon_m = \frac{ypos - 1}{1200} + 25 \quad (3)$$

$$Lon_m = 25.4567^\circ E \quad (4)$$

where  $Lon_m$  is the longitude of the point with the maximum elevation and  $ypos$  is the actual position of that point in the elevation file.

In both formulas we know that the coordinates of the lower left corner of the elevation matrix are  $Lat = 45^\circ N, Lon = 25^\circ E$ . Since in Matlab the first element of the matrix is defined by position [1,1] we need to subtract 1 from the actual position of a point to calculate the position of that point.

The highest peak existing in the area is Omu Peak, with an elevation of 2207m, situated at 45.4455N, 25.4563E (source: [www.alpinet.org](http://www.alpinet.org)).

3. We choose a set of data corresponding to a square of 380 x 380 data points which on the field covers the area of the Bucegi Mountains. A 3D surface plot shows the following (see figure 6):

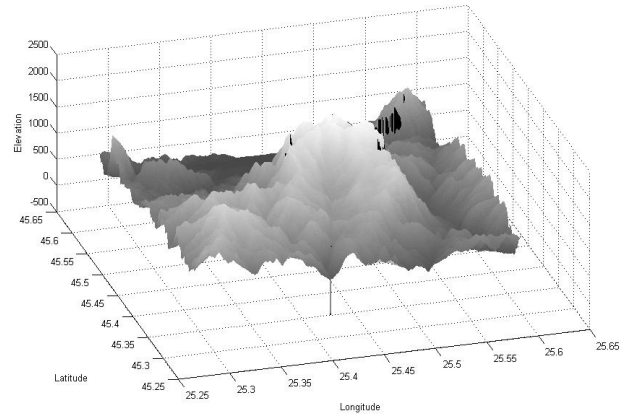


Fig.6 A surface plot of the elevation of Bucegi Mountains

In this graphic the errors (void data) appear to have a negative elevation. Such errors are more common in mountain area with steep slopes where the coverage of the radar is reduced,

As a comparison, the same area as it is viewed in Google Earth (figure 7).



Fig. 7 Bucegi Mountains viewed in Google Earth

## V. CONCLUSIONS

While we have used Google Earth as a source for images for our comparison, this program allows only viewing the data and performing some basic measurements. For example, creating altitude profiles of specific areas is not possible in Google Earth.

Elevation data represented in Matlab can be analyzed in many various ways, such as determination of area of specific landmarks, based on geographical coordinates and elevation profile. This is especially important in hill and mountain area where the surface has an irregular shape and where other methods can lead to false results.

For the future is intended to find an algorithm for filling the voids in the data sets, using interpolation and data from other sources, which can be either data measured on the field, or from other sources of elevation data, such as the lower precision GTOPO 30, which contains 30 arcseconds elevation data.

Also we want to expand this application for detection of ridges and valleys in a specific area, and for calculating the surface of a valley based on the elevation model, with applications in modeling of flow of rainwater in a specific area.

## REFERENCES

- [1] <http://www2.jpl.nasa.gov/srtm/>
- [2] Rodriguez, E., C.S. Morris, J.E. Belz, E.C. Chapin, J.M. Martin, W. Daffer, S. Hensley, 2005, An assessment of the SRTM topographic products, Technical Report JPL D-31639, Jet Propulsion Laboratory, Pasadena, California
- [3] <ftp://e0srp01u.ecs.nasa.gov/>
- [4] <http://srtm.usgs.gov/>
- [5] <http://www.alpinet.org>
- [6] <http://earth.google.com>
- [7] Amos Gilat – Matlab: An Introduction with Applications