Topographic Horizons

This set of functions, based on the paper in IEEE Geoscience and Remote Sensing Letters [1], computes the angles to the horizons from an elevation grid, measured in degrees upward from the horizontal. The one-dimensional problem uses an order N algorithm [2]. Horizons for arbitrary azimuths are derived by rotating the elevation grid, then calculating the horizons along the columns of the rotated grid and re-rotating the grid back to its original orientation. The code supports grids in either projected or geographic format; it also calculates the distances to the horizon.

The examples in the *demo* folder reproduce Figures 1 through 4 in the associated manuscript [1]. The demos require some functions from my Sun Position toolbox [3] that are included in the demo folder, but the horizon functions themselves do not. To reduce the size of the DEM (digital elevation model), run the demos using *MainDemo*, which uses just a 0.25°×0.25° section of that paper's topographic 1°×1° tile. Alternatively, run any of the Demo_...m functions in the demo folder with any geographic elevation grid Z and raster reference R.

A note on azimuth directions

Back in the late 1970s when I started working on topographic radiation problems, my favorite text about climate, Sellers' *Physical Climatology* [4], represented solar azimuths with zero South, positive East, negative West, consistent with a right-hand coordinate system (e.g., longitudes positive East). An alternative convention, zero North, positive clockwise around the circle to 360°, has turned out to be more common. For example, MATLAB's *gradientm* function uses it, although MATLAB's *atan2d* returns results in the ±180° range. My codes allow either convention. Considering that users will employ one or the other rather than switch back and forth, I decided against making the choice through an argument in many functions. Instead, users can set their preference in the *azimuthPreference* function; a bunch of my functions call that one to decide how to represent azimuths.

Main function: horizonAllDirections

The main function *horizonAllDirections* calculates horizons for elevation grid in all azimuths around the circle, specified either as ∓180° or 0 to 360° depending on settings in the *azimuthPreference* function. Available options for parallel processing comprise processing the different rotations in parallel or processing the columns in the rotated grid in parallel.

Other functions can be used: horizonRotatedLatLon, horizonRotatedProj, horizonAlongProfile, viewFactor

horizonAllDirections calls either *horizonRotatedLatLon* or *horizonRotatedProj* depending on whether the input elevation grid is in projected or geographic format. In turn, these routines call *horizonAlongProfile*, which computes the one-dimensional problem. These functions can be individually called depending on the application.

horizonAllDirections returns the azimuths (a vector), horizon angles (3D), and distances to horizons (3D). *viewFactor* uses the horizon data to calculate the view factors, the fraction of the sky open to a cell. The slope and aspect of the cell are also needed, so *topographicSlope* computes those.

Saving results: writeHorizon

Once the azimuths, horizons, and distances are computed, the storage function *saveHorizon* offers options to save in formats HDF 5, NetCDF 4, geotiff, or MATLAB. If HDF 5 or NetCDF 4 format is chosen, both the horizons and the distances can be saved in the same file. If geotiff is chosen, two files are output if both horizons and distances are selected. If MATLAB (.mat) is selected, useful interpolating functions are stored as the output (horizons or distances interpolated based on rows, columns, azimuths). These interpolating functions support models of radiation at the surface during a period when azimuths and solar angles vary spatially and temporally.

References

[1] J. Dozier, "Revisiting the topographic horizon problem in the era of big data and parallel computing," *IEEE Geosci. Remote Sens. Lett.,* 2021, doi: 10.1109/LGRS.2021.3125278.

[2] J. Dozier, J. Bruno, and P. Downey, "A faster solution to the horizon problem," *Comp. Geosci.,* vol. 7, pp. 145-151, 1981, doi: 10.1016/0098-3004(81)90026-1.

[3] J. Dozier. *Sun position: functions for declination, solar longitude, radius vector, equation of time, times of sunrise and sunset, sun angles and azimuths*, Version, MATLAB Central File Exchange, 2020 https://www.mathworks.com/matlabcentral/fileexchange/74939-sun-position.

[4] W. D. Sellers, *Physical Climatology*. Chicago: University of Chicago Press, 1965, 272 pp.