Revisiting Topographic Horizons in the Era of Big Data and Parallel Computing

Jeff Dozier, *Life Senior Member, IEEE*

***Abstract*— Widely used to calculate illumination geometry for estimates of solar and emitted longwave radiation, and for correcting remotely sensed data for topographic effects, digital elevation models (DEMs) are now extensive globally at 10–30-m spatial resolution and locally at spatial resolutions down to a few centimeters. Globally, regionally, or locally, elevation datasets have many grid points. Many software packages calculate gradients over every grid cell or point, but in the mountains, shading by nearby terrain must also be assessed. Terrain may obscure a slope that would otherwise face the Sun. Four decades ago, a fast method to calculate topographic horizons at every point in an elevation grid required computations related only linearly to the size of the grid, but grids now have so many points that parallel computing still provides an advantage. Exploiting parallelism over terrain grids can use alternative strategies: among columns of a rotated grid, or simultaneously at multiple rotation angles, or on different tiles of a grid. On a multi-processor machine, the improvement in computing time approaches 2/3 the number of processors deployed.**

***Index Terms*— Big data applications, digital elevation models (DEMs), parallel processing, surface topography.**

I. Introduction

SIMULATION of the incoming or reflected solar radiation or incoming or emitted longwave radiation in the mountains requires knowledge of the angles to the horizon around the circle of azimuths. The topographic gradients at each grid cell in a digital elevation model (DEM) affect the magnitude of the incoming radiation, but terrain might shade a slope that otherwise would be directly illuminated by the Sun. The earliest efforts to incorporate the horizon in modeling radiation in mountainous terrain [1], [2] used an inefficient method: for each cell, the slope to every other cell was calculated, and the maximum in each direction was chosen. That method's computational complexity is of order N^2: the number of calculations and the computing time increase with the square of the number of cells in the terrain model. A method subsequently developed for calculating horizons in order N time [3] made computation over larger terrain models feasible. Many, if not most, radiation calculations over mountains now use that method [4]–[7]; it has been extended to include trees as terrain elements [8] and applied to snow transport by wind [9] and the availability of solar energy to rovers on other planetary bodies [10]. An existing parallel model for solar radiation calculations over terrain [11], [12] could be even more efficient by incorporating the horizon strategies.

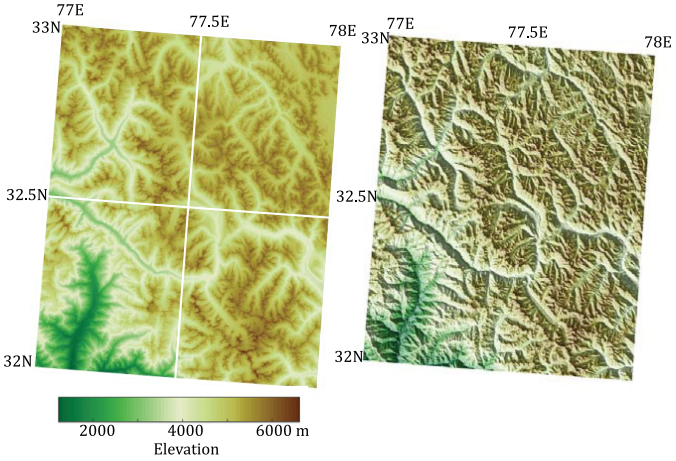

Fig. 1. Example of a DEM from Himachal Pradesh in the Indian Himalaya. Left image shows a 1° × 1° quadrangle from SRTM [13] at 1 arcsec resolution (∼30 m). Right image shows the same grid as a shaded relief. Images are portrayed on an equal-area Albers conic projection centered at 36 °N, 85 °E, with standard parallels at 25 °N and 47 °N.

Many reasons for knowledge about surface elevations have led to data acquisition over continental-scale areas. Widely used and freely available global DEMs (see Fig. 1), derived either from interferometric radar or stereo-photogrammetry, include those from the Shuttle Radar Topography Mission [SRTM: 13] and the Advanced Spaceborne Thermal Emission and Reflection Radiometer on NASA's Terra satellite [ASTER: 14]. Root-mean-square errors in these and four other available DEMs at 30–90-m spatial resolution are 8–10 m in the global products [15]. The global products are distributed as latitude–longitude grids, typically at 1° × 1° tiles. Airbus provides commercially available products that are available through Environmental Systems Research Institute (ESRI).

At regional scales, stereo-photogrammetry from aircraft or commercial fine-resolution satellite sensors yields DEMs with spatial resolutions of a few meters [16], and the same technique applied to declassified spy satellite imagery produces DEMs going back 50 years [17], [18]. Even finer scale DEMs

Manuscript received June 26, 2021; revised September 27, 2021 and October 24, 2021; accepted November 1, 2021. Date of publication November 4, 2021; date of current version January 11, 2022. This work was supported in part by the University of California under Award LFR-18-548316 and in part by National Aeronautics and Space Administration (NASA) under Award 80NSSC21K0620. The Amazon Web Services (AWS) Cloud Credit for Research Program provided computing support.

The author is with the Bren School of Environmental Science & Management, University of California at Santa Barbara, Santa Barbara, CA 93106 USA (e-mail: dozier@ucsb.edu).

Digital Object Identifier 10.1109/LGRS.2021.3125278

can be acquired at local scales by lidar [19] or structure from motion [20] on aircraft, drone, or terrestrial scanners. These products covering smaller area are typically distributed in projected coordinates.

II. Algorithm

The horizon calculation has order N computational complexity. Even so, identifying the horizons around the range of azimuths for large DEMs involves significant computation, hence the desire to exploit parallel computing. Three potential strategies for parallelization of the horizon problem include: across columns of a rotated grid, or simultaneously at multiple rotation angles, or on overlapping tiles of a large grid.

A. One-Dimension

For a given azimuth, the 1-D problem consists of a set of profiles through the elevation grid. For a profile, define an elevation function Z on a set of points $j = 1, 2, \ldots, N$ each at monotonically increasing or decreasing distance D from an arbitrary origin. The points need not be equally spaced. In the forward direction, the objective is to identify the point k that forms the horizon for the point j, i.e., $H_f(j) = k$.

Define a slope function, which converts negatives to zero

$$\text{slope}(j, k) = \max\left[0, \frac{Z(k) - Z(j)}{|D(k) - D(j)|}\right]. \tag{1}$$

If $Z(j) > Z(k)$ for all $k \leq N$, then j is its own horizon, i.e., $H_f(j) = j$ and the slope to the horizon is zero. The naïve approach [2] would be to calculate the slopes for all $k > j$ and select the maximum, but that computational complexity is of order N^2 because every point is compared with every other point.

The better algorithm (see Fig. 2) obtains its efficiency by noting that if $\text{slope}(j, k) > \text{slope}(j, H_f(k))$, then all the points forward of k need not be checked and $H_f(j) = k$. Alternatively, if $\text{slope}(j, k) \leq \text{slope}(j, H_f(k))$, then the next candidate to check is $H_f(k)$, ignoring points between k and $H_f(k)$. The code for this order N algorithm for a whole profile, originally published in Pascal [3], has been translated to C [21], R [22], Python [23], [24], and MATLAB [25], [26]. Some MATLAB code [26] accounts for earth's (or another planet's) curvature, but seldom does that calculation change the identification of H_f.

Identifying the horizons H_b in the backward direction is accomplished by flipping the Z and D vectors. Once the points defining H_f and H_b are identified, the horizon angles ψ are the arctangents of the slopes. Note that some authors express the horizon angles as downward from zenith [6], [21] instead of upward from horizontal.

B. Applied to a Topographic Grid

By rotating the grid—elevations and their coordinates—the forward and backward horizon angles and distances for the resulting azimuths can be calculated by running the 1-D algorithm along the columns. The horizons in both the forward and backward directions are computed for any rotation, so rotations need to cover only a range of 180° (e.g., ±90°) to cover the full circle.

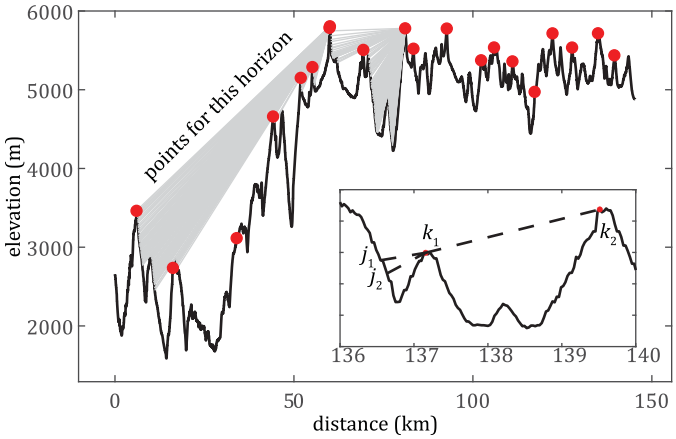

Fig. 2. Example of a horizon profile in the forward direction, extending from the southwest to northeast corners of the 1°× 1° quadrangle in Fig. 1. Of the 5000 points in the profile, the algorithm identifies 675 as horizons, of which 20 are shown. For two horizon locations, the light gray lines identify the associated profile points. The inset illustrates the algorithm's behavior: $\text{slope}(j_1, k_1) < \text{slope}(k_1, k_2)$ so k_2 is the next candidate for $H_f(j_1)$, but $\text{slope}(j_2, k_1) > \text{slope}(k_1, k_2)$ so $H_f(j_2) = k_1$. In neither case does the algorithm check the points between k_1 and k_2.

The analysis must account for three artifacts as follows.

1) Except for the rotation of a multiple of 90°, preserving the full grid when rotating creates a larger grid with unused locations outside the original and missing locations inside. Nearest-neighbor interpolation in the rotation avoids problems encountered with bilinear or cubic interpolation, which can introduce spurious values near the edges of the original in the rotated grid. With nearest neighbors, the intrinsic coordinates (row, column) are also rotated, so the resulting calculations can be directly translated to the coordinates of the original un-rotated grid.
2) Nearest-neighbor rotation leaves out some cells, up to 18% of them for a 45° rotation. Gaps are inpainted [27] from the surrounding cells.
3) The rotation angle needed for a specific azimuth is not easily analytically predictable; the great-circle direction from the southwest to northeast corner of a latitude–longitude grid varies with latitude. If the grid is in projected rather than geographic coordinates, sometimes map projections are rotated from their conventional orientation, requiring an affine transformation from northing–easting coordinates to latitude–longitude. One can numerically solve for the rotation needed for a particular azimuth, or for horizon angles around the full circle, one can rotate through a regular increment of angles, calculate the resulting azimuth(s) along the columns, and interpolate from the results in calculating topographic views.

C. Horizon Angles and Topographic Views

At a given azimuth ϕ, the topography shades a cell from direct solar illumination at solar zenith angle θ_0 if the slope itself faces steeply enough away from the Sun or by the

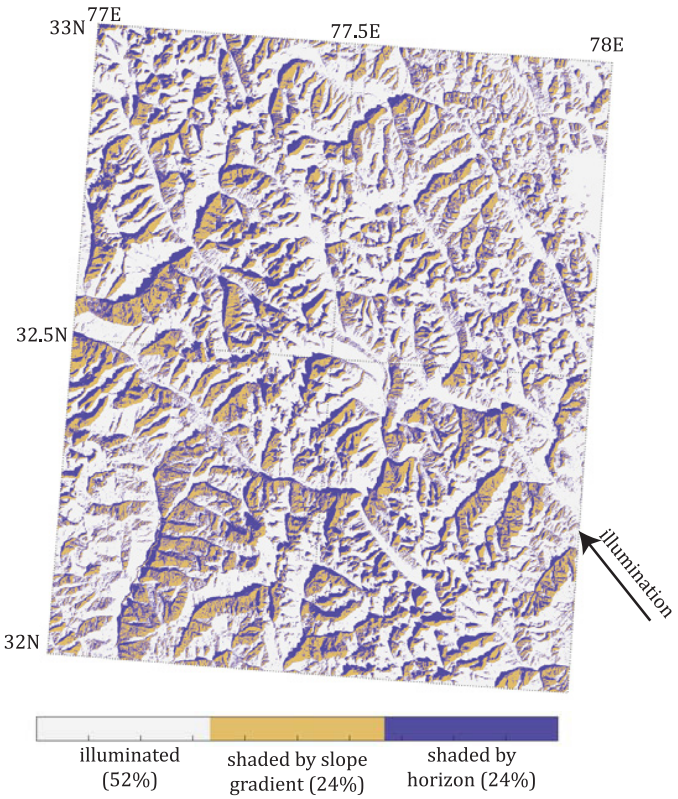

Fig. 3. Example of areas in Fig. 1 that the Sun illuminates, those where the slope shades itself from direct sunlight (24%), and those that would otherwise be directly illuminated but are shaded by neighboring horizons (additional 24%). Solar illumination geometry is on the winter solstice at 09:00 Indian Time Zone. Solar zenith angle in the center of the grid is 73°.

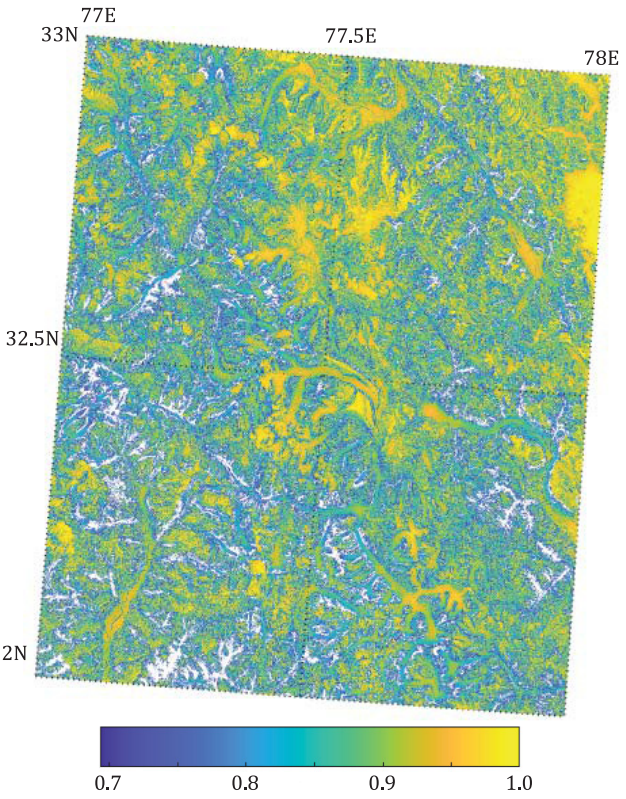

Fig. 4. Sky view factor V_Ω for the topographic grid in Fig. 1. 95% of the values lie between 0.7 and 1.0. Values below this range are in white, especially near the southwest corner.

horizon if $\sin\psi > \cos\theta_0$. In mountainous terrain, shading by horizon can affect a significant fraction of an area, as shown in Fig. 3.

The horizons also determine the view factor V_Ω, the fraction of a location's overlying hemisphere visible to the sky. This quantity enables calculation of diffuse solar radiation or atmospheric thermal radiation over a topographic grid. Define ψ_ϕ as the horizon angle in radians in the azimuth direction ϕ, along with the slope angle S and azimuth A of a cell in a topographic grid. ψ and S are upward from horizontal. The origin and direction of azimuths can be arbitrary (e.g., zero can reference any origin, directions can be clockwise or counterclockwise) but ϕ and A must follow the same convention. For slopes that face toward the horizon, the limits of integration $[\phi_1, \phi_2]$ lie where $\cos(A-\phi) \geq 0$ [21]

$$V_\Omega = \frac{1}{2\pi}\int_{\phi_1}^{\phi_2}\Big[\cos S\cos^2\psi_\phi + \sin S\cos(A-\phi) \times\Big(\frac{\pi}{2} - \psi_\phi - \sin\psi_\phi\cos\psi_\phi\Big)\Big]d\phi. \quad (2)$$

For the azimuths where $\cos(A-\phi) < 0$, the slope itself might obscure the horizon. To account for those cases, the limits of integration cover the azimuths outside the range covered in (2) and the values of ψ_ϕ are set to

$$\max\left[\psi_\phi, \sin^{-1}\left(\sqrt{1-\frac{1}{1+\cos^2(A-\phi)\tan^2 S}}\right)\right].$$

For flat grid cells, $S = 0$, so the integrand in (2) reduces to $\cos^2\psi_\phi$ and the limits of integration cover the full circle $[-\pi, \pi]$. Investigators find that 32–64 directions provide enough information to integrate over the full 360° range to calculate the view factors [6], [21]. Fig. 4 shows the view factors for the topographic grid in Fig. 1, computed from 64 horizon azimuths.

III. Strategies for Parallelism

Parallel computing divides a problem into segments that are processed independently and therefore simultaneously in nondeterministic order, either on multi-core computers that can access the same memory or on a cluster of separate computers that share nothing except access to storage. Where separate computers in a cluster also have multiple cores, nested parallelism is possible. On a multi-core computer, parallelism is most conveniently achieved if the language implements a parallel loop, like the parfor loop in MATLAB's Parallel Computing Toolbox or its copy in Python [28].

A. Parallel Processing of Columns

For a rotated or non-rotated topographic grid, the forward and backward horizons for each column are independent of the other columns, so they can be calculated in parallel. Moreover, most topographic grids have more columns than processors in a multi-core computer, so processors deploy efficiently as they access different columns in turn.

Languages like MATLAB or Fortran store matrices in column-major order, so processing by column targets elevations contiguous in memory. In languages that store matrices

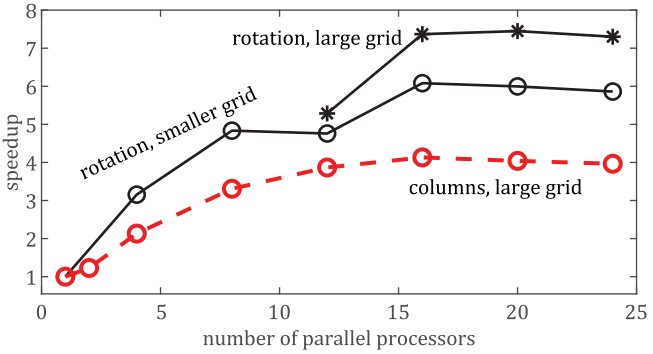

Fig. 5. Increase in performance ("speedup") for parallel processing of horizon angles, by parallelizing either the rotation angles or the calculations of the horizon angles along each column. The large grid size is 3601 × 3601; the smaller size is 901 × 901, i.e., 1/16th the larger size.

in row-major order, like C/C++ or Python, one might modify the code to process the rows in parallel or transpose the grid matrix first.

B. Parallel Processing of Rotations

The rotations to provide forward and backward horizons, i.e., for pairs of azimuths, are independent of one another, so they can be computed in parallel. Because the rotation involves other computation than just the horizons along a profile, this choice provides a greater speed improvement than parallel processing of the columns. However, the maximum number of processors that can deploy is just half the number of horizon azimuths, because calculation of horizon angles in the forward and backward directions happens in one rotation.

C. Processing by Tiles and Recombining

With a cluster of computers, an option for horizons over a larger area is to assign separate topographic tiles to different machines, each parallelizing either the profiles or the rotations. For example, the EarthExplorer data center [29] distributes the SRTM and ASTER DEMs in 1° × 1° tiles. Near tile's edges, the algorithm will underestimate horizon angles in the directions away from the grid because points in the adjacent tile might form the true horizon. To calculate horizons more correctly near the edges of a grid, the tiles should overlap. The horizon function returns both the horizon angles and the distances to the horizons, providing information about the amount of overlap needed. In the grid in Fig. 1, the 95th-percentile distance to all horizons is 14.7 km; the 95th-percentile distance to horizon angles exceeding 10° is 4.8 km, the maximum distance being 20 km. Therefore, overlapping the tiles by 10–15 km would eliminate significant edge effects in the topographic grid in Fig. 1.

D. Parallel Processing of View Factors

Equation (2) applies to every grid cell independently, so that calculation can run in parallel. The 3-D array of the horizon angles can be stored in band-sequential (BSQ) format, where the third dimension contains each azimuth, or in band-interleaved-by-pixel (BIP) format, where the first dimension contains the azimuths. Permuting the data cube to the format that places the horizons for each grid cell contiguous in memory (BIP in MATLAB or Fortran, BSQ in C/C++ or Python) enhances computational efficiency in computing the view factors.

E. Efficiencies Achieved With Parallel Computing

Fig. 5 shows "speedup," the ratio of serial execution time to parallel execution time [30] for the topographic grid in Fig. 1 for horizons in 32 azimuth directions. Parallelizing by rotation provides greater speedup, based on tests starting with one processor and going to 24, using in turn parallelizing by rotation and by columns. Processing of rotations peaks in this case at 14–16 processors, half the number of azimuth directions. Because each rotation calculates horizons for two directions, additional processors beyond 16 remain unused. Moreover, rotations of 0° or 90° are calculated much more rapidly because post-processing the rotations back to the original grid does not involve the steps described in Section II-B. Therefore, two processors finish their tasks rapidly, hence trivial difference between speeds for 14 versus 16 processors. In processing by columns, the speedup becomes asymptotic around 16 processors, where calculating horizons for a single direction for the whole 3601 × 3601 grid takes only 2 s. At that point, the overhead of adding more processors negates further improvement.

F. Storage

Once the azimuths, horizons, and distances are computed, formatting options for storage include HDF 5, NetCDF 4, geotiff, and MATLAB. With HDF 5 or NetCDF 4, both the horizons and the distances can be saved in the same file. With geotiff, two files are needed if both the horizons and distances are selected. In a MATLAB file, the output can comprise useful interpolating functions (horizons or distances interpolated based on rows, columns, azimuths). These interpolating functions support models of radiation at the surface for situations where solar geometry varies spatially and temporally.

IV. Code and Data Availability

The MATLAB code and data that reproduce versions of Figs. 1–5 (with a dataset cropped to 901 × 901 size to reduce download volume) are available from the MATLAB Central File Exchange [26]. The SRTM and other digital elevation data are available in EarthExplorer [29] among other sources.

V. Conclusion

As digital elevation data covering large areas or at fine spatial resolution become widely available, computing topographic horizons and view factors efficiently using parallel computing will enable incorporating these variables into analyses more easily. On a multi-processor machine, the improvement in computing time approaches two-thirds the number of processors deployed, to a point where the computation is "fast enough" that the overhead of adding more processors does not improve performance.

References

[1] L. D. Williams, R. G. Barry, and J. T. Andrews, "Application of computed global radiation for areas of high relief," *J. Appl. Meteorol. Climatol.*, vol. 11, pp. 526–533, Apr. 1972, doi: 10.1175/1520-0450(1972)011<0526:AOCGRF>2.0.CO;2.

[2] J. Dozier and S. I. Outcalt, "An approach toward energy balance simulation over rugged terrain," *Geograph. Anal.*, vol. 11, no. 1, pp. 65–85, Sep. 2010, doi: 10.1111/j.1538-4632.1979.tb00673.x.

[3] J. Dozier, J. Bruno, and P. Downey, "A faster solution to the horizon problem," *Comp. Geosci.*, vol. 7, no. 2, pp. 145–151, 1981, doi: 10.1016/0098-3004(81)90026-1.

[4] R. Dubayah and P. M. Rich, "Topographic solar radiation models for GIS," *Int. J. Geograph. Inf. Syst.*, vol. 9, no. 4, pp. 405–419, 1995, doi: 10.1080/02693799508902046.

[5] M. P. Bishop, B. W. Young, J. D. Colby, R. Furfaro, E. Schiassi, and Z. Chi, "Theoretical evaluation of anisotropic reflectance correction approaches for addressing multi-scale topographic effects on the radiation-transfer cascade in mountain environments," *Remote Sens.*, vol. 11, no. 23, p. 2728, Nov. 2019, doi: 10.3390/rs11232728.

[6] M. Lamare *et al.*, "Simulating optical top-of-atmosphere radiance satellite images over snow-covered rugged terrain," *Cryosphere*, vol. 14, no. 11, pp. 3995–4020, Nov. 2020, doi: 10.5194/tc-14-3995-2020.

[7] R. Bellasio, G. Maffeis, J. S. Scire, M. G. Longoni, R. Bianconi, and N. Quaranta, "Algorithms to account for topographic shading effects and surface temperature dependence on terrain elevation in diagnostic meteorological models," *Boundary-Layer Meteorol.*, vol. 114, no. 3, pp. 595–614, Mar. 2005, doi: 10.1007/s10546-004-1670-6.

[8] F. W. Davis *et al.*, "LiDAR-derived topography and forest structure predict fine-scale variation in daily surface temperatures in oak savanna and conifer forest landscapes," *Agricult. Forest Meteorol.*, vols. 269–270, pp. 192–202, May 2019, doi: 10.1016/j.agrformet.2019.02.015.

[9] D. Marks, A. Winstral, and M. Seyfried, "Simulation of terrain and forest shelter effects on patterns of snow deposition, snowmelt and runoff over a semi-arid mountain catchment," *Hydrol. Process.*, vol. 16, pp. 3605–3626, Dec. 2002, doi: 10.1002/hyp.1237.

[10] J. H. Bai and Y.-J. Oh, "Global path planning of lunar rover under static and dynamic constraints," *Int. J. Aeronaut. Space Sci.*, vol. 21, no. 4, pp. 1105–1113, Dec. 2020, doi: 10.1007/s42405-020-00262-x.

[11] J. Hofierka, M. Lacko, and S. Zubal, "Parallelization of interpolation, solar radiation and water flow simulation modules in GRASS GIS using OpenMP," *Comput. Geosci.*, vol. 107, pp. 20–27, Oct. 2017, doi: 10.1016/j.cageo.2017.07.007.

[12] M. Šúri and J. Hofierka, "A new GIS-based solar radiation model and its application to photovoltaic assessments," *Trans. GIS*, vol. 8, no. 2, pp. 175–190, 2004, doi: 10.1111/j.1467-9671.2004.00174.x.

[13] T. G. Farr *et al.*, "The shuttle radar topography mission," *Rev. Geophys.*, vol. 45, no. 2, 2007, Art. no. RG2004, doi: 10.1029/2005RG000183.

[14] H. Fujisada, G. B. Bailey, G. G. Kelly, S. Hara, and M. J. Abrams, "ASTER DEM performance," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 12, pp. 2707–2714, Dec. 2005, doi: 10.1109/TGRS.2005.847924.

[15] E. Uuemaa, S. Ahi, B. Montibeller, M. Muru, and A. Kmoch, "Vertical accuracy of freely available global digital elevation models (ASTER, AW3D30, MERIT, TanDEM-X, SRTM, and NASADEM)," *Remote Sens.*, vol. 12, p. 3482, Jan. 2020, doi: 10.3390/rs12213482.

[16] D. E. Shean *et al.*, "An automated, open-source pipeline for mass production of digital elevation models (DEMs) from very-high-resolution commercial stereo satellite imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 116, pp. 101–117, Jun. 2016, doi: 10.1016/j.isprsjprs.2016.03.012.

[17] A. Dehecq *et al.*, "Automated processing of declassified KH-9 Hexagon satellite images for global elevation change analysis since the 1970s," *Frontiers Earth Sci.*, vol. 8, p. 516, Nov. 2020, doi: 10.3389/feart.2020.566802.

[18] J. M. Maurer, S. B. Rupper, and J. M. Schaefer, "Quantifying ice loss in the eastern Himalayas since 1974 using declassified spy satellite imagery," *Cryosphere*, vol. 10, no. 5, pp. 2203–2215, Sep. 2016, doi: 10.5194/tc-10-2203-2016.

[19] M. E. Hodgson and P. Bresnahan, "Accuracy of airborne LiDAR-derived elevation," *Photogramm. Engr. Remote Sens.*, vol. 70, pp. 331–339, Mar. 2004, doi: 10.14358/PERS.70.3.331.

[20] J. Meyer and S. M. Skiles, "Assessing the ability of structure from motion to map high-resolution snow surface elevations in complex terrain: A case study from senator Beck Basin, CO," *Water Resour. Res.*, vol. 55, no. 8, pp. 6596–6605, Aug. 2019, doi: 10.1029/2018WR024518.

[21] J. Dozier and J. Frew, "Rapid calculation of terrain parameters for radiation modeling from digital elevation data," *IEEE Trans. Geosci. Remote Sens.*, vol. 28, no. 5, pp. 963–969, Sep. 1990, doi: 10.1109/36.58986.

[22] J. Van doninck. (2018). *Horizon: Horizon Search Algorithm, Version 1.2, R Package Documentation*. [Online]. Available: https://rdrr.io/cran/horizon/

[23] S. Havens. (2021). *Topocalc, Version 0.5.0, Python Package Index*. [Online]. Available: https://pypi.org/project/topocalc/

[24] M. Lamare. (2020). *REDRESS GitHub Repository, Version 0.1-2020.10.29, GitHub*. [Online]. Available: https://github.com/maximlamare/REDRESS

[25] P. Sirguey, R. Mathieu, and Y. Arnaud, "Subpixel monitoring of the seasonal snow cover with MODIS at 250 m spatial resolution in the Southern Alps of New Zealand: Methodology and accuracy assessment," *Remote Sens. Environ.*, vol. 113, no. 1, pp. 160–181, 2009, doi: 10.1016/j.rse.2008.09.008.

[26] J. Dozier. (2021). *Topographic Horizons: Angles to the Horizons From an Elevation Grid With Options for Parallelism, Version 4.3, MATLAB Central File Exchange*. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/94800-topographic-horizons

[27] F. Bornemann and T. März, "Fast image inpainting based on coherence transport," *J. Math. Imag. Vis.*, vol. 28, no. 3, pp. 259–278, 2007, doi: 10.1007/s10851-007-0017-6.

[28] W. Pomp. (2021). *Parfor, Version 2021.7.1, Python Package Index*. [Online]. Available: https://pypi.org/project/parfor/

[29] USGS. *EarthExplorer, Digital Elevation, Version SRTM 1 Arc-Second Global*. Accessed: Nov. 1, 2021. [Online]. Available: https://earthexplorer.usgs.gov/

[30] *Predicting and Measuring Parallel Performance*, Intel, Santa Clara, CA, USA, 2012. [Online]. Available: https://www.intel.com/content/dam/develop/external/us/en/documents/1-1-appthr-predicting-and-measuring-parallel-performance-165587.pdf