This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LGRS.2021.3125278, IEEE Geoscience and Remote Sensing Letters

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 are now extensive globally at 10 to 30 m spatial resolution and locally at spatial resolutions down to a few cm. Either 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 employ 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*—Digital elevation models, surface topography, big data applications, parallel processing**

I. Introduction

Simulation of 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.

Many reasons for knowledge about surface elevations have led to data acquisition over continental-scale areas. Widely used and freely available global DEMs (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 to 90 m spatial resolution are 8 to 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 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 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. The 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 one-dimensional problem consists of a set of profiles through the elevation grid. For a profile, define an elevation function Z on a set of points $j = 1,2, \dots N$ 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 point j, i.e., $H_f(j) = k$.

Define a slope function, which converts negatives to zero:

Manuscript submitted June 26, 2021, revision submitted October 24, 2021. This work was supported in part by University of California Award LFR-18-548316 and NASA Award 80NSSC21K0620. The AWS Cloud Credit for Research program provided computing support.

J. Dozier is with the Bren School of Environmental Science & Management, University of California, Santa Barbara, CA 93106-5131, USA (e-mail: dozier@ucsb.edu).

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

If $Z(j) > Z(k)$ for all $k \le N$, 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 to every other point.

The better algorithm (Fig. 2) obtains its efficiency by noting that if $\mathrm{slope}(j,k) > \mathrm{slope}\left(j, H_f(k)\right)$, then all points forward of k need not be checked and $H_f(j) = k$. Alternatively, if $\mathrm{slope}(j,k) \le \mathrm{slope}\left(j, H_f(k)\right)$, then the next candidate to check is $H_f(k)$, ignoring points between k and $H_f(k)$. 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 one-dimensional algorithm along the columns. Horizons in both forward and backward directions are computed for any rotation, so rotations need cover only a range of 180° (e.g., ±90°) to cover the full circle.

The analysis must account for three artifacts:

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

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 azimuth direction ϕ, along with 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 $\cos(A-\phi) \ge 0$ [21]:

$$\begin{aligned} V_\Omega = \frac{1}{2\pi}\int_{\phi_1}^{\phi_2} & [\cos S \cos^2\psi_\phi + \sin S \cos(A-\phi) \\ & \times \left(\frac{\pi}{2} - \psi_\phi - \sin\psi_\phi \cos\psi_\phi\right)] d\phi \end{aligned} \tag{2}$$

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 to 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].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LGRS.2021.3125278, IEEE Geoscience and Remote Sensing Letters

A. Parallel Processing of Columns

For a rotated or non-rotated topographic grid, 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 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

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 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 a 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 to 15 km would eliminate significant edge effects in the topographic grid in Fig. 1.

D. Parallel Processing of View Factors

Eq. (2) applies to every grid cell independently, so that calculation can run in parallel. The 3D array of 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 to 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 II.B. Therefore, two processors finish their tasks rapidly, hence trivial difference between speeds for 14 vs 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 two seconds. 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 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 through 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.

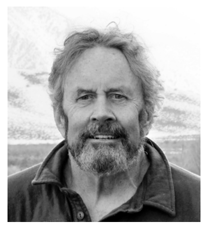
References

[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, 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," *Geog. Anal.*, vol. 11, pp. 65-85, 1979, 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, pp. 145-151, 1981, doi: 10.1016/0098-3004(81)90026-1.

[4] R. O. Dubayah and P. Rich, "Topographic solar radiation models for GIS," *Intl. J. Geog. Info. Sci.,* vol. 9, 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, p. 2728, 2019, doi: 10.3390/rs11232728.
[6] M. Lamare, M. Dumont, G. Picard, F. Larue, F. Tuzet, C. Delcourt, and L. Arnaud, "Simulating optical top-of-atmosphere radiance satellite images over snow-covered rugged terrain," *Cryosphere,* vol. 14, pp. 3995-4020, 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," *Bound.-Lay. Meteorol.,* vol. 114, pp. 595-614, 2005, doi: 10.1007/s10546-004-1670-6.
[8] F. W. Davis, N. W. Synes, G. A. Fricker, I. M. McCullough, J. M. Serra-Diaz, J. Franklin, and A. L. Flint, "LiDAR-derived topography and forest structure predict fine-scale variation in daily surface temperatures in oak savanna and conifer forest landscapes," *Ag. Forest. Meteorol.,* vol. 269, pp. 192-202, 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, 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," *Intl. J. Aero. Space Sci.,* vol. 21, pp. 1105-1113, 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," *Comp. Geosci.,* vol. 107, pp. 20-27, 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, pp. 175-190, 2004, doi: 10.1111/j.1467-9671.2004.00174.x.
[13] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, D. Seal, S. Shaffer, J. Shimada, J. Umland, M. Werner, M. Oskin, D. Burbank, and D. Alsdorf, "The Shuttle Radar Topography Mission," *Rev. Geophys.,* vol. 45, p. RG2004, 2007, 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, pp. 2707-2714, 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, 2020, doi: 10.3390/rs12213482.
[16] D. E. Shean, O. Alexandrov, Z. M. Moratto, B. E. Smith, I. R. Joughin, C. Porter, and P. Morin, "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, 2016, doi: 10.1016/j.isprsjprs.2016.03.012.
[17] A. Dehecq, A. S. Gardner, O. Alexandrov, S. McMichael, R. Hugonnet, D. Shean, and M. Marty, "Automated processing of declassified KH-9 Hexagon satellite images for global elevation change analysis since the 1970s," *Front. Earth Sci.,* vol. 8, p. 516, 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, pp. 2203-2215, 2016, doi: 10.5194/tc-10-2203-2016.
[19] M. E. Hodgson and P. Bresnahan, "Accuracy of airborne lidar-derived elevation: Empirical assessment and error budget," *Photogramm. Engr. Remote Sens.,* vol. 70, pp. 331-339, 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, pp. 6596-6605, 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, pp. 963-969, 1990, doi: 10.1109/36.58986.
[22] J. Van doninck. *horizon: Horizon search algorithm*, Version 1.2, R Package Documentation, 2018 https://rdrr.io/cran/horizon/.
[23] S. Havens. *topocalc*, Version 0.5.0, Python Package Index, 2021 https://pypi.org/project/topocalc/.
[24] M. Lamare. *REDRESS GitHub repository*, Version 0.1-2020.10.29, GitHub, 2020 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, pp. 160-181, 2009, doi: 10.1016/j.rse.2008.09.008.
[26] J. Dozier. *Topographic horizons: angles to the horizons from an elevation grid with options for parallelism*, Version 4.2, MATLAB Central File Exchange, 2021 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, pp. 259-278, 2007, doi: 10.1007/s10851-007-0017-6.
[28] W. Pomp. *parfor*, Version 2021.7.1, Python Package Index, 2021 https://pypi.org/project/parfor/.
[29] USGS. *EarthExplorer, Digital Elevation*, Version SRTM 1 Arc-Second Global, U.S. Geological Survey, https://earthexplorer.usgs.gov/
[30] "Predicting and measuring parallel performance." Intel. https://www.intel.com/content/dam/develop/external/us/en/documents/1-1-appthr-predicting-and-measuring-parallel-performance-165587.pdf.



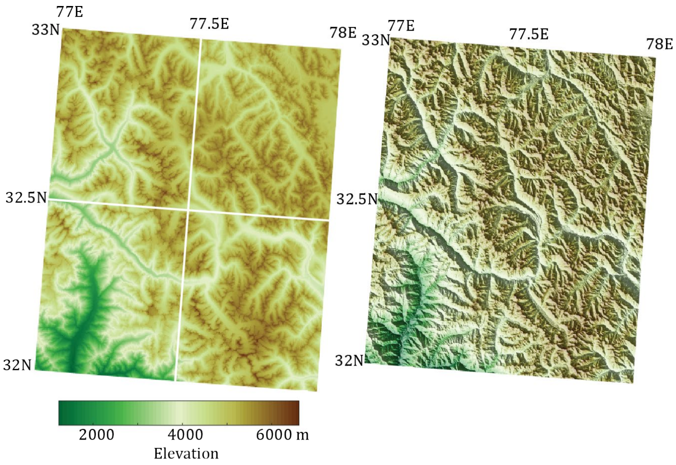
Jeff Dozier (Life Senior Member, IEEE) received the B.A. degree in geography from California State University East Bay, Hayward, CA, USA, in 1968, and the Ph.D. degree in geography from the University of Michigan, Ann Arbor, MI, USA, in 1973.

He is a Distinguished Professor Emeritus with the University of California, Santa Barbara, CA, USA, having retired in 2018. He had been a Faculty Member at UC Santa Barbara since 1974, and he founded the Bren School of Environmental Science and Management and served as its first dean for 6 years. From 1987 to 1990, he worked part-time with Jet Propulsion Laboratory as a Senior Member of the Technical Staff, and from 1990 to 1992, he was the Senior Project Scientist for NASA's Earth Observing System at the NASA Goddard Spaceflight Center.

Prof. Dozier received the IEEE GRSS *Transactions* Prize Paper Award in 1988. He is a Fellow of the American Geophysical Union and the American Association for the Advancement of Science and a Distinguished Scientist in the Chinese Academy of Sciences. He received the NASA/Department of Interior William T. Pecora Award in 2005 and the 2009 Jim Gray Award from Microsoft for his achievements in data-intensive science. In 2016, he was the Burges Distinguished Visiting Professor at the University of Washington.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LGRS.2021.3125278, IEEE Geoscience and Remote Sensing Letters

FIGURES



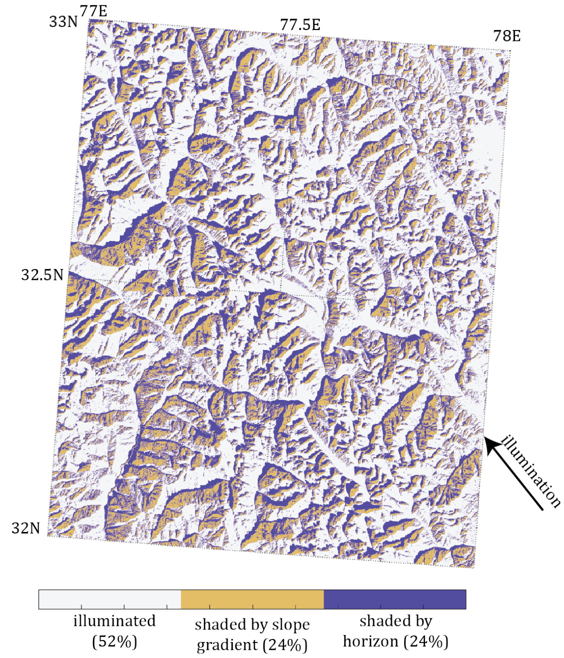


Fig. 1 that the Sun illuminates, those where the slope shades itself from direct sunlight (24%), and those that would otherwise be directly illuminated but are shaded by neighboring horizons (additional 24%). Solar illumination geometry is on the winter solstice at 09:00 Indian Time Zone. Solar zenith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

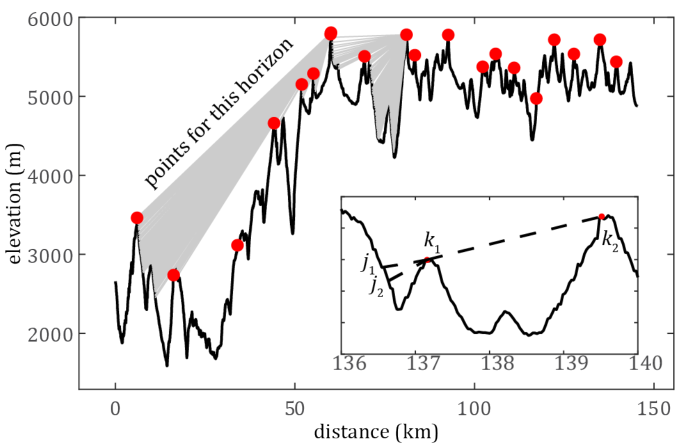


Fig. 1. Of the 5000 points in the profile, the algorithm identifies 675 as horizons, of which 20 are shown. For two horizon locations, the light gray lines identify the associated profile points. The inset illustrates the algorithm's behavior: $\text{slope}(j_1, k_1) < \text{slope}(k_1, k_2)$ so k_2 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.

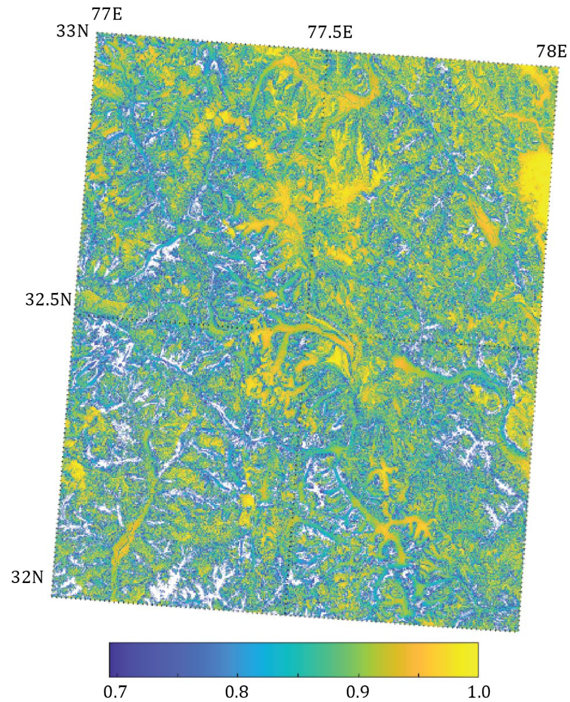


Fig. 1. 95% of the values lie between 0.7 and 1.0. Values below this range are in white, especially near the southwest corner.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LGRS.2021.3125278, IEEE Geoscience and Remote Sensing Letters