JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 113, F04001, doi:10.1029/2007JF000964, 2008

Global search algorithm for nondispersive flow path extraction

Kyungrock Paik[1]

Received 8 December 2007; revised 12 May 2008; accepted 17 June 2008; published 7 October 2008.

[1] A new algorithm is developed to extract flow paths from digital elevation data without planar dispersion on the basis of the concept of global search. Widely used nondispersive algorithms, such as deterministic eight-neighbor flow direction retrieval algorithms, suffer serious uncertainty in their determined flow paths because of the lack of variability, i.e., only eight allowed flow directions. Although uncertainty at the local level is an inherent problem residing in the domain discretization, this study shows that more reasonable flow paths at the global scale can be obtained by maximizing the use of all information stored in the given digital elevation data. By utilizing information stored in cells other than those in the direct vicinity, this alternative approach can reduce the uncertainty residing in the extraction of flow paths. The proposed algorithm makes a significant improvement in flow path variability on both theoretical and real landscapes, while it is still simple.

Citation: Paik, K. (2008), Global search algorithm for nondispersive flow path extraction, *J. Geophys. Res.*, *113*, F04001, doi:10.1029/2007JF000964.

1. Introduction

[2] The use of digital elevation models (DEMs) has significantly changed the way of studying Earth's surface processes. One of the advantages that digital maps offer is the convenient estimate of surface water flow paths, such as rivers and streams, through automatic processing. Once surface water flow paths are determined, important hydrogeomorphological parameters, such as the drainage area and slope, can be estimated. This convenience, together with the increasing availability of DEMs over the world, makes DEMs and their applications widely and frequently utilized in scientific investigations and various engineering practices in a number of disciplines such as hydrology, geomorphology, geology, and ecology.

[3] Inappropriate flow path retrieval algorithms or erroneous values in original DEMs can result in the faulty estimation of flow paths [e.g., *Callow et al.*, 2007]. The determination of correct drainage directions is crucial, as an incorrect estimate of flow direction at a single point will result in a wrong estimate of hydrogeomorphological parameters (e.g., the drainage area) of all downstream points and a completely erroneous watershed delineation. As a result, improving the performance of algorithms that retrieve flow paths from DEMs and reducing uncertainties inherent in the DEMs are fundamental requirements for any quantitative use of DEM-based parameters. In this paper, the author addresses the issue of the performance of flow path retrieval algorithms.

[1]School of Environmental Systems Engineering, University of Western Australia, Crawley, Western Australia, Australia.

0148-0227/08/2007JF000964$09.00

[4] With a general viewpoint from physics, extracting surface water flow directions from a geographic map can be regarded as retrieving directions of matter or energy fluxes which should be perpendicular to equipotential lines. For example, electric field lines are always perpendicular to equipotential lines, where the equipotent means the equal electric voltage. Groundwater flow directions are also perpendicular to equipotential lines, which form the flow net. In this case, the equipotential line is the line that traces points of an equal hydraulic head. Similarly, if forces other than gravity are negligible in driving surface water flow, surface water flow directions should be perpendicular to geographic contour lines, the lines that connect points of equal potential energy.

[5] However, difficulty arises when it comes to the digitized domain (DEMs), where assigning the flow direction exactly perpendicular to equipotential lines becomes ambiguous on such a discontinuous domain. Various algorithms have been proposed for the extraction of flow paths from DEMs in the raster grid format. The first of these is often referred to as the deterministic eight-neighbor flow direction retrieval approach (D8) [*O'Callaghan and Mark*, 1984], where the flow follows the steepest of eight possible flow directions toward surrounding cells (Figure 1). Although D8 has been widely used [e.g., *Tribe*, 1992], the algorithm is limited because only eight allowed flow directions constrain the representation of flow path variability [e.g., *Fairfield and Leymarie*, 1991]. Efforts to alleviate this drawback have focused on (1) introducing more than eight flow directions and (2) adding supplementary algorithms within the limit of eight flow directions. These are briefly reviewed below.

[6] Several algorithms allow flow in directions other than the eight discrete directions [e.g., *Freeman*, 1991; *Quinn et*

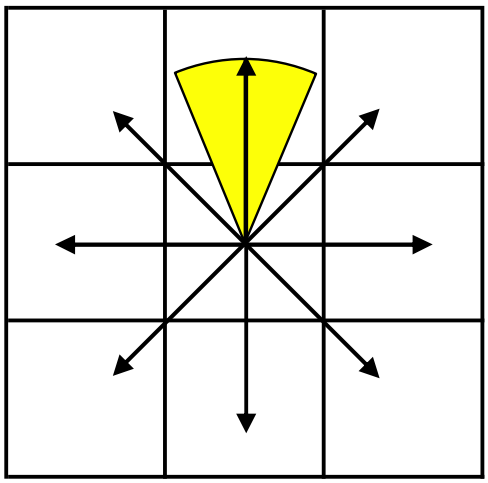

Figure 1. Raster grids and eight possible flow directions toward eight surrounding cells. The determined flow direction has the uncertainty in the angle of specified flow direction, as wide as $\pi/4$ rad (colored area) when square grids are used.

al., 1991; *Costa-Cabral and Burges*, 1994; *Tarboton*, 1997; *Seibert and McGlynn*, 2007]. For example, *Tarboton* [1997] assigned the flow direction angle which is continuously defined between 0 and 2π rad, allowing an infinite number of possible flow directions. In such algorithms, the water is assumed to flow toward more than one neighbor cell (multiple flow directions). This results in the spread of the water over the domain, called planar dispersion, which does not necessarily have physical meaning [*Tarboton*, 1997]. Since these algorithms are inherently accompanied by planar dispersion, they are called dispersive algorithms. In contrast, algorithms that assign a single flow direction to each cell, such as D8, are called nondispersive algorithms [*Orlandini et al.*, 2003].

[7] Some dispersive algorithms can improve the estimate of certain variables, such as the drainage area. However, dispersive algorithms cannot define specific flow paths, and, hence, they are not suitable for investigating the transport of biomass, carbon, nutrient, pollutant, sediment, water, etc., through channel corridors. Specific definition of adequate flow paths is also important in studying river geomorphology, where morphodynamic mechanisms along specifically defined channels eventually trigger the evolution of the whole landscape [e.g., *Paik and Kumar*, 2008]. Therefore, when channel processes are important, a nondispersive algorithm, which assigns a single flow direction (instead of multiple flow directions) to one of eight neighboring cells, is preferred [*Orlandini et al.*, 2003].

[8] Various schemes have been proposed to improve the decision of flow directions without introducing planar dispersion. *Fairfield and Leymarie* [1991] added a random term in the selection of the steepest direction in D8 and obtained more natural looking river networks. However, this algorithm has been criticized [e.g., *Costa-Cabral and Burges*, 1994] in that the reasoning behind the introduction of randomness is weakly supportive; the randomness generates different flow paths at every run for the same DEM, and extracted flow paths over a plane are not parallel, which is unreasonable. Therefore, this algorithm is advantageous only when statistical characteristics of overall patterns are of interest. *Lea* [1992] proposed a nondispersive algorithm that assigns flow direction continuously (between 0 and 2π rad), the algorithm later employed by *Costa-Cabral and Burges* [1994]. However, Lea's algorithm makes faulty decisions even in a simple example, as illustrated by *Tarboton* [1997]. *Orlandini et al.* [2003] proposed nondispersive algorithms on the basis of two measures of the angular and transversal deviation from the theoretical drainage direction and recommended the algorithm developed for the least transversal deviation (D8-LTD). Their complicated algorithms require users to specify the value of the model parameter, called the "dampening factor." They recommended the use of a dampening factor of 1 on the basis of several empirical tests. However, there is no analytical background to support the decision of the value of the dampening factor.

[9] In this paper, the author demonstrates that significant improvement over the limitation of D8 and D8-LTD can be achieved using a new and simple idea without introducing any model parameter. It is important to realize that the problems associated with D8 are inherent in the nature of digitization; hence, no absolute solution exists. Therefore, all we can do is "improvement" instead of resolving the issues. Within the fundamental limit of the discretization, the best improvement may be obtainable by maximizing the use of all information stored in the digital elevation data. On the basis of this general concept, a new nondispersive algorithm based on a global search scheme is introduced in this paper. The proposed algorithm systematically orders all cells in the domain and considers not only directly neighboring (first-order) cells but also cells of other orders to make a better flow direction decision. The new algorithm is still a D8 method because flow is assigned to one of the eight neighbors, but because a global search is used, the new algorithm is referred to as global D8 (GD8), while the original D8 approach, which uses only local slopes [*O'Callaghan and Mark*, 1984], is referred to as local D8 (LD8).

[10] The philosophical background of the proposed concept is similar to that of global search algorithms for optimization problems. The importance of finding global optima, instead of ending up with local optima, has already gained significant attention in various optimization problems [e.g., *Goldberg*, 1989]. Successful finding of global optima may be possible by considering all available variables (instead of a few local variables) over the entire search range (instead of narrowing down the search extent) at each search, as shown by several heuristic algorithms [e.g., *Holland*, 1975]. In this sense, the proposed algorithm for flow path search shares the same philosophy with global search optimization algorithms but deals with a different context.

[11] The rest of this paper is organized in the following order. In section 2, the proposed algorithm is explained with a simple example. Then, the developed algorithm is applied to theoretical terrains and a natural landscape in section 3. Summary and discussion are given in section 4.

2. Nondispersive Algorithm Based on Global Search

[12] Figure 1 illustrates the problem raised in this study. This study deals with raster (or regular) DEMs, although the

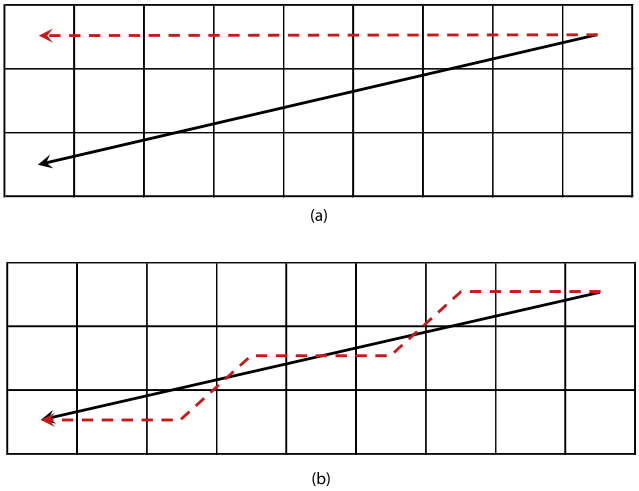

Figure 2. Illustration showing the theoretical flow path (solid line) over a plane and the estimated flow path (dashed line) by (a) LD8 and (b) an ideal nondispersive algorithm.

proposed algorithm can also be adapted to other discretizations, such as triangular grids [e.g., *Mark*, 1975; *Takayasu and Inaoka*, 1992; *Braun and Sambridge*, 1997]. Raster DEMs have a simple matrix format, which requires a relatively small amount of memory for data storage and is compatible with various computational applications. Currently, DEMs provided by various public sectors are mostly in the raster format. In this discretization, each cell is surrounded by eight neighboring cells. LD8 compares slopes to these eight cells and chooses the steepest direction, i.e., the local steepest direction (LSD), as the flow direction. However, as shown in Figure 1, this causes substantial uncertainty in the angle of flow direction. In a domain of square grids, this angular uncertainty is as wide as $\pi/4$ rad. This means that whether the real flow direction is 22° or −22°, LD8 approximates the direction as 0°.

[13] If this uncertainty is accumulated along a path, the consequence is significant (Figure 2). In Figure 2, the solid line indicates the theoretical flow path, which is perpendicular to geographic contour lines. Although the theory to draw the ideal flow path on a given geographic map is clear, once the path is projected on the digitized domain, the theoretical flow path is not consistent with raster discretization; that is, it is not clear which cells this path passes. Dispersive algorithms essentially treat this issue by artificially partitioning the flow into multiple cells, introducing planar dispersion. To avoid planar dispersion, one may use LD8. However, LD8 extracts a very wrong flow path if the uncertainty at each local search is accumulated along the path (Figure 2a). For cases in which planar dispersion should be avoided, the most ideally estimated flow path will be the path that defines a specific flow direction (one of eight discrete directions) at the local level but follows the correct trend of the theoretical flow path at the global level (Figure 2b). For this reason, the local flow path will become zigzag, which is an inevitable cost of the benefit of both using DEMs and avoiding planar dispersion.

[14] To achieve such an ideal nondispersive flow path (Figure 2b), the proposed algorithm uses elevation data stored in cells other than the eight neighboring cells as supplementary information. To do so, as a first step, all cells within the domain are systematically ordered on the basis of their topological closeness to a specific cell, called the "start cell." Initially, the start cell is designated as the highest-altitude cell in the domain. Flow direction assignment starts from the start cell and proceeds along the downstream direction. The start cell can be updated with a downstream cell as the flow direction assignment proceeds downstream. The eight neighboring cells of the start cell are designated as the first-order neighbors. Continuing outward, the cells that surround the nth-order neighbors are designated as the $(n+1)$th-order neighbors. For example, cells that surround the first-order neighbors are called the second-order neighbors (Figure 3). This systematically orders all cells in the domain on the basis of their closeness to the start cell (cell A in Figure 3). Following this ordering scheme, the number of nth-order neighbor cells is $8n$, unless some of the nth-order cells under the ideal expansion are interrupted by the domain boundary or are ineffective cells (such as lake and ocean). In other words, there are 8 cells of first-order neighbor and 16 cells of second-order neighbor.

[15] Note that LD8 uses only information stored in the first-order neighbors. The idea of the proposed algorithm is to utilize information stored in higher-order neighboring cells to reduce the uncertainty caused by the local search. One may consider only the first- and the second-order neighbors, which is the minimal extension of LD8. If cells up to the highest-order neighbors in the domain are considered, this becomes a global search (GD8).

[16] In a local search, the deviation between the real flow path and the determined direction indicates the error of the local search, as shown in Figure 1. Note that the real flow

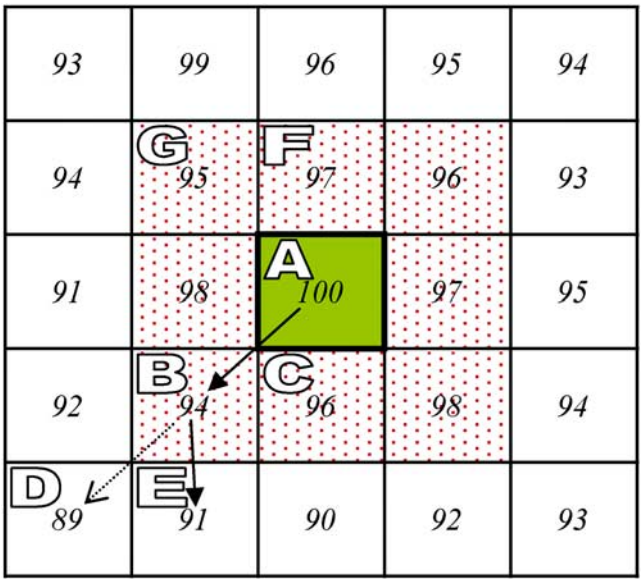

Figure 3. Illustration showing the proposed algorithm. Cells that surround the start cell (A) are the first-order neighbors (dotted cells). Cells that surround the first-order neighbors are the second-order neighbors (uncolored cells without dots). The number in each cell represents the cell's elevation. In this example, local search (LD8) gives the flow path as A to B to D. However, the proposed algorithm chooses the path as A to B to E.

direction can be on either the clockwise or the counterclockwise side of the determined flow direction. However, if the deviation is repeatedly found to be on one side of the determined flow direction along a flow path, the deviation between the true and the determined flow paths may become wide enough, at a certain stage, that it is appropriate to correct the flow direction by overriding the local search on the basis of the information accrued since the selection of the flow direction for the start cell. This is the key feature of the proposed algorithm.

[17] To support this correction procedure, the proposed algorithm finds two possible flow directions for each cell (instead of one direction of steepest descent), i.e., the LSD and the secondary direction. The secondary direction is defined as the steeper direction between clockwise and counterclockwise directions adjacent to the LSD. Given this information, it is reasonable to assume that the real flow direction is located somewhere between the LSD and the secondary direction.

[18] The LSD is chosen as a priori flow direction. However, if the deviation between the real direction and the LSD is unidirectional and wide enough to result in a steeper global slope, the secondary direction is selected as the flow direction. Specifically, the secondary direction is chosen if all of the following three conditions are satisfied.

[19] 1. The LSD is the same as the determined flow direction of the upstream cell (unidirectionality).

[20] 2. The secondary direction is the same as that of the upstream cell (accrued deviation).

[21] 3. The gradient between the downstream cell of the secondary direction and the start cell is steeper than the gradient between the downstream cell of the LSD and the start cell (steeper global slope).

[22] Here the proposed algorithm is explained with a simple example, illustrated in Figure 3. The algorithm begins from the start cell, which contains the highest altitude in the domain (cell A). For the start cell, the algorithm finds the LSD (toward cell B) and the secondary direction (toward cell C). The flow direction of the start cell is chosen as the LSD since no accrued deviation is available yet.

[23] Once the flow direction of the start cell is determined, the next step is to determine the flow direction of its downstream cell (cell B). The start cell remains as cell A. A local search is made at cell B; that is, the LSD (toward cell D) and the secondary direction (toward cell E) are estimated. In this example, the LSD of cell B is the same as the determined flow direction of the upstream cell (cell A) (condition 1) and the secondary direction of cell B also agrees with that of the upstream cell (condition 2). If the LSD is selected as the flow direction of cell B, the chosen path based on two consecutive searches (since cell A) has accumulated bias on the same side. The proposed algorithm recognizes this circumstance and accordingly corrects this flow direction bias. Between two candidates (cells D and E) for downstream cells, the flow path is determined as toward the cell that yields the steeper gradient with respect to the start cell (condition 3). Therefore, in this case, the slope between the downstream cell of the LSD (cell D) and the start cell (cell A) is compared to the slope between the downstream cell of the secondary direction (cell E) and the start cell. Note that in obtaining slopes, the proposed algorithm considers difference in diagonal and cardinal distances: in this illustration, whereas distances between centroids of adjacent cells both on the abscissa (e.g., between cells B and C) and on the ordinate (e.g., between cells A and C) are unity, the distance between centroids is $\sqrt{2}$ for cells A and B and $\sqrt{5}$ for cells A and E. The slope from cell A to cell E ($(100 - 91)/\sqrt{5} = 4.02$) is greater than that from cell A to cell D ($(100 - 89)/2\sqrt{2} = 3.89$), so in this example the secondary direction to cell E is chosen as the flow direction for cell B.

[24] The above procedure is repeated for the newly determined downstream cell until the determined flow path reaches the boundary of the domain, an ineffective cell, or a cell whose flow direction is already specified. Then, the algorithm designates a new start cell as the cell of the highest altitude among remaining cells, that is, cells whose flow directions are not assigned yet. The entire procedure is repeated until the flow direction of all effective cells (i.e., excluding cells corresponding to lake or ocean) in the domain is determined.

[25] On some occasions, the start cell is updated while the flow direction assignment is progressing downstream. This is illustrated in the case of cell F in Figure 3. The LSD of cell F is toward cell G. Since both cells on the clockwise and counterclockwise sides adjacent to the LSD have elevations higher than cell F, there is no secondary direction having positive downward slope. This is the situation in which the flow passes through a clearly defined corridor with steep lateral slopes. In this case, the LSD is chosen as the flow direction, and its downstream cell (cell G) becomes a new start cell because the deviation accumulated since the previous start cell is assumed to be resolved.

[26] The above procedure explains the proposed global search algorithm (GD8). If necessary, we can limit the extent of the search algorithm by forcing the start cell, which acts as a reference, to move downstream regularly. For example, we can consider only the first- and the second-order neighbors by continuously updating the start cell with every downstream cell, 1 order apart from the cell of consideration (extended D8 up to second-order neighbors (ED8 2nd)). We can also consider up to the third-order neighboring cells only (ED8 3rd) by keeping the start cell moving downstream, 2 orders apart from the cell of consideration. Alternately, we can take the start cell as the same as the current cell at every search. For this special case, the proposed algorithm will be equivalent to LD8. As presented above, the proposed algorithm is composed of simple rules, making it easy to program or implement. However, this simple algorithm brings a great improvement in flow path variability and achieves a more reasonable global flow pattern, as will be shown in section 3.

3. Applications

[27] The proposed algorithm was tested on theoretical terrains (outward and inward cones and a plane) and a real landscape. In all examples, raster discretization with square grids was adopted. Theoretical terrains such as cone-like terrains and plains serve as excellent yardsticks to evaluate retrieval algorithms since ideal flow paths on these terrains are well known. Theoretical terrains chosen here are those frequently used for similar tests in earlier studies [*Fairfield*

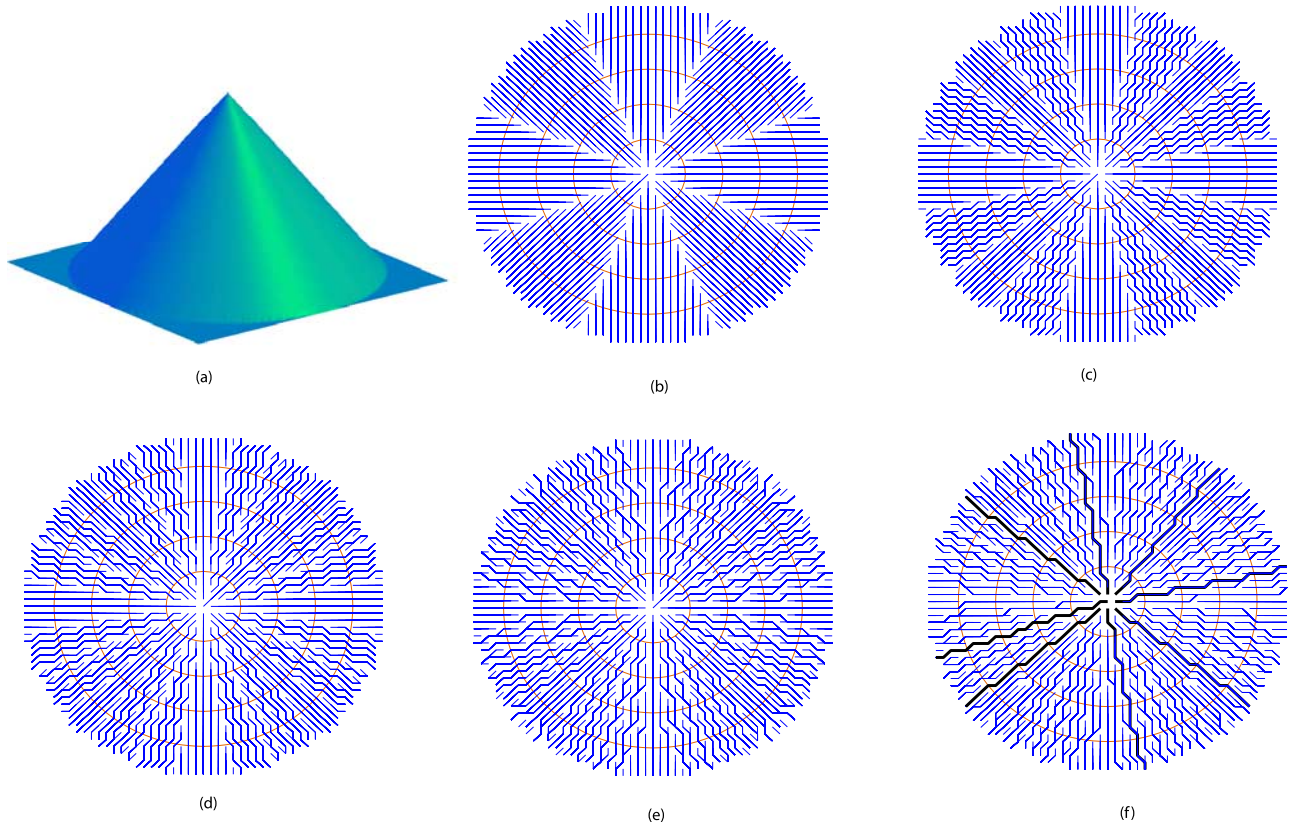

Figure 4. (a) A theoretical cone-like terrain and retrieved flow paths using (b) LD8 (local search), (c) ED8 2nd (considering the first- and the second-order neighboring cells only), (d) ED8 3rd (considering up to the third-order neighboring cells), (e) GD8 (considering all cells in the domain), and (f) D8-LTD. Blue lines indicate determined flow paths, and orange circles indicate the elevation contour drawn at uniform interval. For D8-LTD, flow paths initiated from the nine highest-altitude cells are deviated from the ideal flow paths (highlighted with thick black lines). Note that for square grids, where distances between centroids of adjacent cells both on the abscissa and on the ordinate are identical, the distance between diagonally parallel flow paths is cos (π/4) ($\approx$0.7) times (i.e., about 30% narrower than) the distance between cordially parallel paths. This difference is particularly well shown in Figure 4b.

and Leymarie, 1991; *Freeman*, 1991; *Costa-Cabral and Burges*, 1994; *Tarboton*, 1997; *Orlandini et al.*, 2003; *Seibert and McGlynn*, 2007]. The proposed algorithm with various search extents is compared to other nondispersive algorithms such as LD8 and D8-LTD, using both visual and quantitative measures. In the implementation of D8-LTD, the dampening factor is consistently set as 1, following the recommendation of *Orlandini et al.* [2003].

[28] Figure 4a shows a theoretical terrain of a circular cone-like setting (domain size 51 $\times$ 51). Ideally, the flow paths over this terrain should show a radially symmetric explosive pattern, with straight and unparallel individual paths heading in an infinite number of directions. In this example, LD8 clearly shows its limitation in the variability of flow paths (Figure 4b). There are many erroneous paths in which streams initiated from two nearby upstream cells arrive at outlets which are either too far or too close (as a result of faulty parallel paths especially). This problem is caused by the raster discretization itself, and use of finer-resolution DEMs will make no difference. To alleviate this issue without abandoning the discretization itself, we need a more sophisticated algorithm.

[29] In contrast to LD8, the proposed algorithm successfully retrieves a flow pattern closer to the theoretical pattern, i.e., explosive to infinite directions. Adding only the second-order neighboring cells into consideration already makes a great improvement (Figure 4c). At each cell (local level), the flow direction is still assigned to one of eight directions (no planar dispersion), the same as in LD8. However, the global pattern shows that the flow direction over the domain is composed of 16 directions, two times as variable as LD8. Accordingly, the associated uncertainty in the angle of the flow path has been reduced to half that of LD8, i.e., π/8 rad. In this case, the algorithm finds a best direction among 16 (8n, where n = 2) directions instead of 8. This is analogous to drawing a straight line between the start cell and one of its 16 second-order cells, ignoring first-order cells located in between. However, it is impossible to perfectly represent this straight line on a digitized domain, as discussed in sections 1 and 2. The algorithm approximates the most plausible first-order cell that this straight line passes by local search. This local direction is corrected in the determination of the flow direction from the first-order cell to a second-order cell, resulting in an unavoidable zigzagging path.

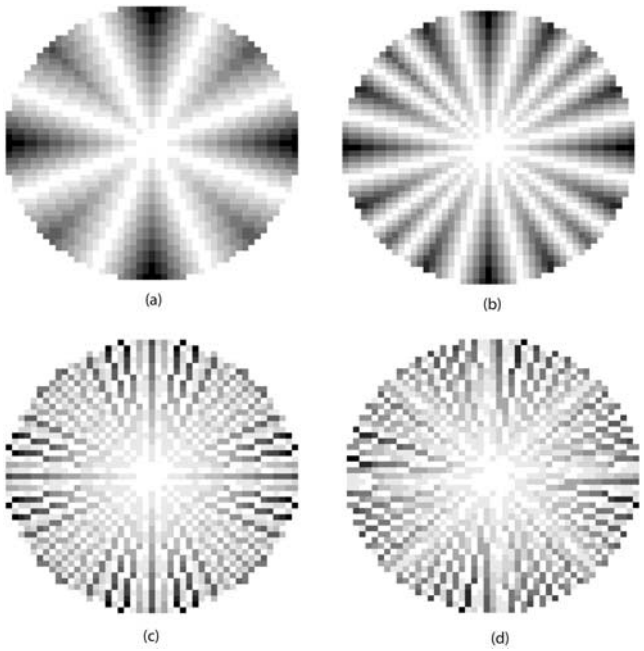

Figure 5. The spatial distribution of contributing area for the theoretical cone-like terrain (Figure 4a) by (a) LD8, (b) ED8 2nd, (c) GD8, and (d) D8-LTD. The gray tone of each cell is scaled to represent the contributing area with white as zero and black as the greatest contributing area in the domain. The asymmetric flow pattern of D8-LTD is also reflected in the drainage area distribution.

[30] Adding higher-order neighbors into consideration further increases the variability of global flow directions. In the case in which up to the third-order neighboring cells are considered, the extracted flow paths head in 24 (8×3) directions at the global scale (Figure 4d). Eventually, this number of global flow directions becomes unlimited as the search extent grows infinitely (GD8) (Figure 4e). However, as shown in Figures 4d and 4e, increased flow path variability may result in confluences of streamlines over the domain, which theoretically, should not appear on this terrain. This is due to the conflict of decisions made at different (local and wider) scales, i.e., consequences of local collision of paths which are inevitably zigzag to avoid planar dispersion. This example shows that there is a trade-off between increased variability and conflicts of scales in considering higher-order neighbors.

[31] The result of D8-LTD (Figure 4f) also exhibits increased flow path variability. However, the overall flow pattern obtained by D8-LTD is radially asymmetric. This is mainly caused by several paths distorted from ideal paths. As highlighted in Figure 4f, this problem occurs even on paths initiating from the nine highest-altitude cells (the tip and the eight cells around the tip), for which flow paths should clearly follow eight straight directions (north, east, west, south, and four diagonal directions). Note that even the simplest LD8 (and of course the proposed algorithm) correctly retrieves these straight lines for these obvious cases (Figures 4b–4e). Like GD8, D8-LTD suffers the occurrence of unrealistic confluences, the unavoidable problem as a trade-off of increased variability without planar dispersion.

[32] Difference among these algorithms is also clearly shown in the comparison of the spatial distribution of estimated contributing area (Figure 5). Theoretically, cells the same distance apart from the tip should have the same contributing area, resulting in a series of concentric circles. Although GD8 and D8-LTD show the most well mixed drainage area distribution on concentric circles among nondispersive algorithms compared here, none of these nondispersive algorithms gives the ideal contributing area distribution. There should be only one channel initiation cell (i.e., the tip) in the cone, and the flow from this cell should be dispersed over the entire domain. This is physical dispersion, distinguished from artificial planar dispersion. If a nondispersive algorithm is used, the flow from the tip follows a specific path instead of spreading out. As a result, there are many channel initiation cells over the domain, which makes the resulting drainage area distribution unrealistic. Dispersive algorithms are known to give a better estimate of drainage area for this example [e.g., *Quinn et al.*, 1991; *Tarboton*, 1997]. However, dispersive algorithms achieve these results via artificial planar dispersion instead of correctly accommodating physical dispersion. In an ideal flow path extraction, planar dispersion should be minimized, and, if necessary, physical dispersion should be modeled separately [*Tarboton*, 1997].

[33] Performance of these algorithms is also compared on other theoretical terrains such as an inward cone (the reverse of the terrain shown in Figure 4a) and a plane where the ratio of lateral and longitudinal gradients is 1:4 (Figure 6). The results for the inward cone are basically the same as those of the outward cone: LD8 shows limited variability; GD8 overcomes the limited variability of LD8 and achieves the appropriate global pattern; and D8-LTD also exhibits increased variability, but its retrieved global pattern is asymmetric. For the plane, LD8 again exhibits its limitation while the result of D8-LTD is equally good as that of GD8.

[34] For quantitative comparison of the above algorithms, the author measured the cumulative lateral deviation between determined paths and true paths. For outward and inward cones, the true path of the most upstream cell of a determined path is an extension of the straight line that connects the tip of a cone and the cell. For the plane, true paths are parallel lines with arctan (1/4) rad inclined from the longitudinal line.

[35] Beginning from a specific cell through its determined flow path, the point lateral deviation is computed at each of its downstream cells as the shortest distance between the coordinate of the downstream cell and the true path. The sum of the point lateral deviations of all downstream cells along the retrieved flow path of a specific cell is designated as the path lateral deviation. Then, the cumulative lateral deviation is obtained as the sum of the path lateral deviation of all paths, starting from all effective cells in the domain, allowing duplicate portions of different paths.

[36] The results show that the cumulative lateral deviation decreases as higher-order neighbors are considered, and eventually, GD8 yields the least cumulative lateral deviation among algorithms compared here (Table 1). In the plane example, performance of the proposed algorithm with constraint on the search extent (e.g., ED8 2nd and ED8 3rd)

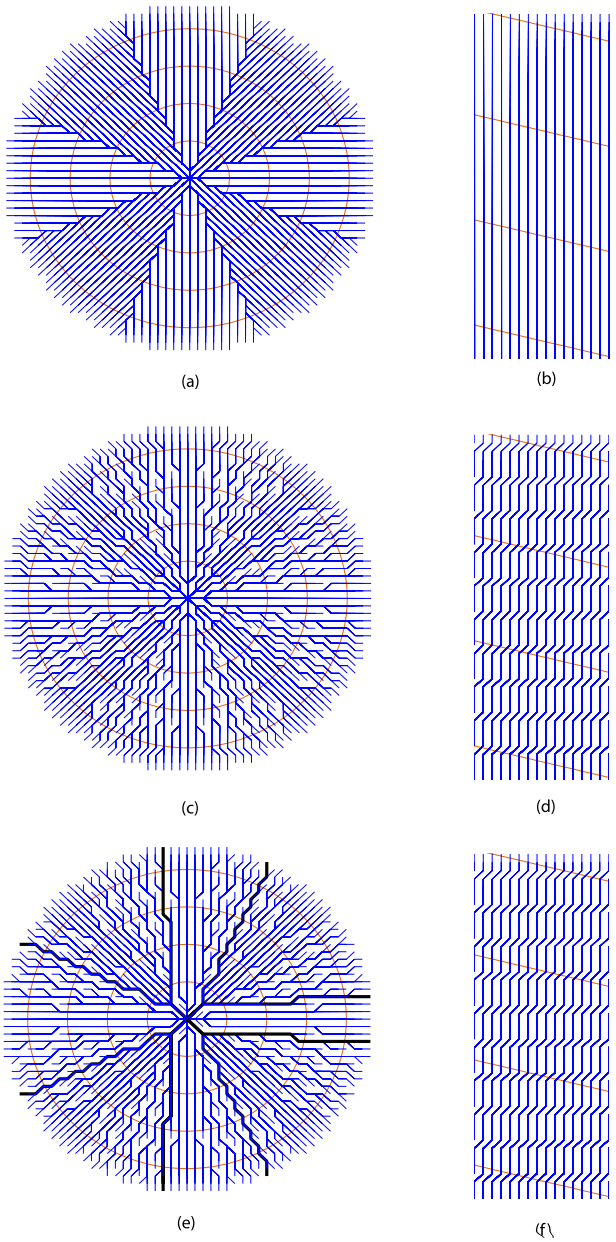

Figure 6. Comparison of nondispersive algorithms on (a, c, e) an inward cone-like terrain and (b, d, f) a plane. Flow direction retrieval results of LD8 (Figures 6a and 6b), GD8 (Figures 6c and 6d), and D8-LTD (Figures 6e and 6f) are compared. Blue lines indicate determined flow paths, and orange background circles (for the inward cone) or lines (for the plane) are the elevation contour drawn at uniform interval. The global pattern obtained by D8-LTD in the inward cone is deviated from the ideal symmetric pattern, as some paths highlighted with thick black lines in Figure 6e show.

depends on the ratio between lateral and longitudinal gradients of the plane. However, the performances of both GD8 and D8-LTD are robust regardless of this ratio. D8-LTD performs as well as GD8 for the plane example but generates a larger deviation for cones, where the terrains are more complex than the plane. This reflects the inadequate global pattern obtained by D8-LTD explained earlier.

[37] To measure the applicability of the proposed algorithm to real landscapes, the algorithm was tested on the Stirling Range, located in the southwestern region of Western Australia. The author used Shuttle Radar Topographic Mission 90 m digital elevation data provided by the Consortium for Spatial Information of the Consultative Group on International Agricultural Research (Jarvis et al., Hole-filled SRTM for the globe version 3, 2006, available at http://srtm.csi.cgiar.org). This data, originally produced by NASA, has horizontal resolution of 3 arc sec (often called 90 m resolution) and vertical resolution of 1 m. In this example, the author intended to test how close retrieved flow paths are to normal to the background contour. To test this, the background contour should be based on a DEM with a higher resolution than the DEM used for flow path extraction. Therefore, the original 3 arc sec DEM was used to construct contour lines, while coarser resolution digital data, i.e., 9 arc sec horizontal resolution, were produced via spatial averaging and were used for flow path extraction. The domain was composed of 144 $\times$ 35 cells at this coarser resolution, covering an area of approximately 367 km^2.

[38] Figure 7 shows that general flow patterns retrieved from GD8 are more perpendicular to elevation contour lines than those from LD8. Flow paths obtained by GD8 are more variable than those from LD8, where many paths of LD8 exhibit unrealistic parallel lines, repeating the same issue found in the cone examples. In comparing results from LD8 and GD8, differences in flow directions are found at 421 cells out of 5040 cells (8.35%). Differences in parameters such as drainage area and flow discharge, which are estimated on the basis of the extracted stream network, are much more significant than this difference in flow directions.

4. Summary and Discussion

[39] Defining adequate flow paths in landscapes is the first step in the simulation of hydrological and geomorphological processes. To model vital hydrogeomorphological processes through channel corridors, flow directions have to be determined specifically without introducing planar dispersion. However, existing nondispersive flow path retrieval algorithms suffer limited variability or show unrealistic tendencies in flow paths, as shown in Figures 4b, 4f, 6a, 6b, and 6e. In this study, the author developed an alternative nondispersive algorithm based on the concept of global search. This algorithm incorporates all available information

Table 1. Comparison of the Cumulative Lateral Deviation for Different Algorithms Applied to Various Theoretical Terrains[a]

	LD8	ED8 2nd	ED8 3rd	GD8	D8-LTD
Cone	100	64	55	51	60
Inward cone	100	70	61	59	63
Plane[b]	100	29	16	4	4

[a]Values are normalized by the results of LD8 being 100. The calculated cumulative lateral deviation varies with the domain size, but GD8 exhibits the least cumulative lateral deviation regardless of the domain size. The values here are from the domain sizes 51 $\times$ 51 for cones and 101 $\times$ 34 for the plane.

[b]The ratio of lateral and longitudinal gradients is 1:4.

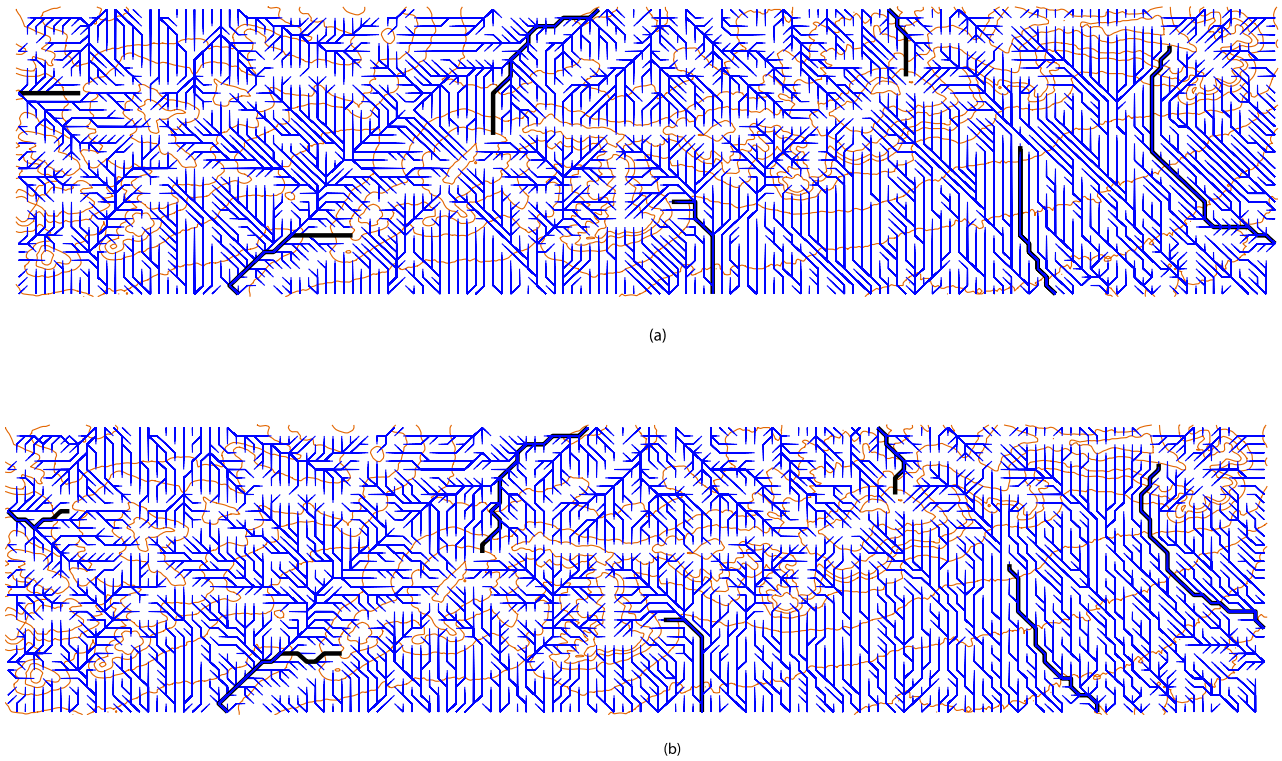

Figure 7. Retrieved flow paths over the Stirling Range using (a) LD8 and (b) GD8. Flow paths (blue lines) for every cell are shown without pruning. For reference, elevation contour lines (orange) are drawn in the background. The contour lines are based on the higher-resolution (3 arc sec) DEM and at altitudes of 160, 180, 210, 250, 300, 350, 400, 600, 800, and 1000 m. For easy comparison of two results, several flow paths initiated from the same cells are highlighted using thick black lines.

stored in the DEM in the decision making. The extent of cells for consideration is adjustable depending on the purpose. If all cells in the domain are considered in the decision making, the proposed algorithm is regarded as a global search algorithm (GD8). On the other hand, if only cells in the direct vicinity are considered, the search algorithm is regarded as a local search (LD8), fundamentally the same as the original D8. Clear improvement of GD8 in retrieving flow paths is demonstrated for theoretical and real landscapes using both visual and quantitative measures.

[40] Different flow path extraction algorithms yield different flow lengths, which has direct hydrological implication. Flow length estimation is affected by both the DEM resolution and the flow path extraction algorithm. The impact of the DEM resolution occurs at the local scale. Owing to the fractal characteristics of channel length, as finer resolution is used, the more intricate features (meandering) are captured, resulting in greater length [*Saco*, 2003]. For example, *Saco* [2003] showed about 38% increase in the estimated main channel length of Vermilion River from a 30 m DEM, compared to the results from a 90 m DEM. Since DEMs always have limited resolution, there are always missing subgrid features, resulting in local-scale underestimation of flow length.

[41] On the other hand, the effect of the retrieval algorithm on flow length estimation for a given DEM occurs over the global scale. As shown in the plane and Stirling Range examples, LD8 tends to make flow paths straight (Figures 2a, 6b, and 7a), resulting in underestimation of the path length. This implies that LD8 causes further underestimation of flow length in addition to the inherent local-scale underestimation. In contrast, GD8 extracts longer flow paths compared to LD8 for the same DEMs because of the zigzagging effect (Figures 2b, 6d, and 7b). The author postulates that this global-scale overestimation of GD8 can compensate for the inherent local-scale underestimation of flow length to some extent, which can provide a better estimation of overall flow path length compared to LD8. This possibility warrants further study. The cone examples are noticeable cases where not only LD8 but also GD8 underestimates path lengths despite the zigzagging effect of GD8. This is due to a large number of channel initiation cells, which results in many paths being shorter than ideal. Any nondispersive algorithm approximates multiple channel initiation cells over the cone, whereas there is only one channel initiation cell in theory.

[42] Overall, the increase of computational burden of GD8 over LD8 is insignificant. Compared to LD8, the only considerable extra computational cost of GD8 is in the assignment of start cells. In the present study, the assignment is made in the order of altitude, which requires sorting of cells based on their elevation. This creates a certain computational load, especially as domain size grows. Nevertheless, a moderate PC (Intel Pentium D 950 system) took 0.016 s to solve the Stirling Range example (Figure 7) with GD8, which is regarded as computationally affordable. Note that the altitude-based start cell assignment is not the only

option for GD8 and that there is room for developing alternative methodologies in the future, which will change the computational load as well.

[43] In summary, the proposed algorithm successfully reduces the uncertainty residing in local searches and produces more natural flow patterns, while it is still simple, computationally efficient, and easy to use (no model parameter). Software of GD8 used for examples given in this paper is available upon request from the author.

[44] **Acknowledgments.** This research is supported by the University of Western Australia Research Grants Scheme 2007. Comments from anonymous reviewers are greatly appreciated. The author also acknowledges E. J. Holden and Nikolaus Callow for their helpful comments on the early manuscript from the perspectives of a computer scientist and a geomorphologist, respectively.

References

Braun, J., and M. Sambridge (1997), Modelling landscape evolution on geological time scales: A new method based on irregular spatial discretization, *Basin Res.*, *9*, 27–52.

Callow, J. N., K. P. Van Niel, and G. S. Boggs (2007), How does modifying a DEM to reflect known hydrology affect subsequent terrain analysis?, *J. Hydrol.*, *332*, 30–39.

Costa-Cabral, M. C., and S. J. Burges (1994), Digital elevation model networks (DEMON): A model of flow over hillslopes for computation of contributing and dispersal areas, *Water Resour. Res.*, *30*(6), 1681–1692, doi:10.1029/93WR03512.

Fairfield, J., and P. Leymarie (1991), Drainage networks from grid digital elevation models, *Water Resour. Res.*, *27*(5), 709–717, (Correction, Water Resour. Res., 27(10), 2809, (1991)).

Freeman, T. G. (1991), Calculating catchment area with divergent flow based on a regular grid, *Comput. Geosci.*, *17*(3), 413–422.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass.

Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*, 183 pp., Univ. of Mich. Press, Ann Arbor, Mich.

Lea, N. L. (1992), An aspect driven kinematic routing algorithm, in *Overland Flow: Hydraulics and Erosion Mechanics*, edited by A. J. Parsons and A. D. Abrahams, pp. 393–407, Chapman and Hall, New York.

Mark, D. M. (1975), Computer analysis of topography: A comparison of terrain storage methods, *Geogr. Ann., Ser. A*, *57*(3–4), 179–188.

O'Callaghan, J. F., and D. M. Mark (1984), The extraction of drainage networks from digital elevation data, *Comput. Vision Graphics Image Process.*, *28*, 323–344.

Orlandini, S., G. Moretti, M. Franchini, B. Aldighieri, and B. Testa (2003), Path-based methods for the determination of nondispersive drainage directions in grid-based digital elevation models, *Water Resour. Res.*, *39*(6), 1144, doi:10.1029/2002WR001639.

Paik, K., and P. Kumar (2008), Emergence of self-similar tree network organization, *Complexity*, *13*(4), 30–37, doi:10.1002/cplx.20214.

Quinn, P., K. Beven, P. Chevallier, and O. Planchon (1991), The prediction of hillslope flow paths for distributed hydrological modeling using digital terrain models, *Hydrol. Processes*, *5*, 59–80.

Saco, P. (2003), Flow dynamics in large river basins: Self-similar network structure and scale effects, Ph.D. dissertation, Univ. of Ill. at Urbana-Champaign, Urbana.

Seibert, J., and B. L. McGlynn (2007), A new triangular multiple flow direction algorithm for computing upslope areas from gridded digital elevation models, *Water Resour. Res.*, *43*, W04501, doi:10.1029/2006WR005128.

Takayasu, H., and H. Inaoka (1992), New type of self-organized criticality in a model of erosion, *Phys. Rev. Lett.*, *68*, 966–969.

Tarboton, D. G. (1997), A new method for the determination of flow directions and upslope areas in grid digital elevation models, *Water Resour. Res.*, *33*(2), 309–319.

Tribe, A. (1992), Automated recognition of valley lines and drainage networks from grid digital elevation models: A review and a new method, *J. Hydrol.*, *139*, 263–293, doi:10.1016/0022-1694(92)90206-B.

K. Paik, School of Environmental Systems Engineering, University of Western Australia, M015, 35 Stirling Highway, Crawley, WA 6009, Australia. (kyungrock.paik@uwa.edu.au)