International Cartographic Association
Association Cartographique Internationale

A fast and simple algorithm for calculating flow accumulation matrices from raster digital elevation models

Guiyun Zhou*, Wenyan Dong, Hongqiang Wei

School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, P. R. China, Guiyun Zhou, zhouguiyun@uestc.edu.cn, Hongqiang Wei, 1135861564@qq.com

* Corresponding author

Keywords: Flow accumulation, Flow direction, DEM

Abstract:

Flow accumulation is an essential input for many hydrological and topographic analyses such as stream channel extraction, stream channel ordering and sub-watershed delineation. Flow accumulation matrices can be derived directly from DEMs and general have O(NlogN) time complexity (Arge, 2003; Bai et al., 2015). It is more common to derive the flow accumulation matrix from a flow direction matrix. This study focuses on calculating the flow accumulation matrix from the flow direction matrix that is derived using the single-flow D8 method (Barnes et al., 2014; Garbrecht & Martz, 1997; Nardi et al., 2008; O'Callaghan & Mark, 1984). In this study, we find give an overview of algorithms for flow accumulation calculation that have O(N) time complexity. These algorithms include algorithms are based on the concept of the number of input drainage paths (Wang et al.2011, Jiang et al. 2013), the algorithm based on the basin tree indices (Su et al. 2015), and the recursive algorithm (Choi, 2012; Freeman, 1991).

We propose a fast and simple algorithm to calculate the flow accumulation matrix. Compared with the existing algorithms that have O(N) time complexity, our algorithm runs faster and generally requires less memory. Our algorithm is also simple to implement. In our algorithm, we define three types of cells within a flow direction matrix: source cells, interior cells and intersection cells. A source cell does not have neighboring cells that drain to it and its NIDP value is zero. An interior cell has only one neighboring cell that drains to it and its NIDP value is one. An intersection cell has more than one neighboring cell that drains to it and its NIDP value is greater than one. The proposed algorithm initializes the flow accumulation matrix with the value of one. Our algorithm first calculates the NIDP matrix from the flow direction matrix. The algorithm then traverses each cell within the flow direction matrix row by row and column by column, similar to the traversal algorithm. When a source cell *c* is encountered, the algorithm traces all downstream cells of *c* until it encounters an intersection cell *i*. During the tracing, the accumulation value of a cell is added to the accumulation value of its immediate downstream cell. An interior cell has only one neighboring cell that drains to it and its final accumulation value is obtained when the tracing is done. The accumulation value of the intersection cell *i* is updated from this drainage path. However, cell *i* has other unvisited neighboring cells that drain to it and its final accumulation value cannot be obtained after this round of tracing. The algorithm decreases the NIDP value of *i* by one. Cell *i* is visited again when other drainage paths that pass through it are traced. When all of the drainage paths that pass through it are traced, cell *i* is treated as an interior cell and the final accumulation value of *i* is obtained correctly and the last tracing process can continue the tracing after cell *i* is treated as an interior cell. A worked example of the proposed algorithm is shown in Figure 1.

The five flow accumulation algorithms with O(N) time complexity, including Wang's algorithm, Jiang's algorithm, the BTI-based algorithm, the recursive algorithm and our proposed algorithm, are implemented in C++. The 3-m LiDAR-based DEMs of thirty counties in the state of Minnesota, USA, are downloaded from the FTP site operated by the Minnesota Geospatial Information Office. The first 30 counties in Minnesota in alphabetic order are chosen for the experiments to avoid selection bias. We use the algorithm proposed by Wang and Liu (2006) to fill the depressions and derive the flow direction matrices for all tested counties. The running times on the Windows system are listed in Figure 2. The average running times per 100 million cells are 14.42 seconds for Wang's algorithm, 15.90 seconds for Jiang's algorithm, 18.95 seconds for the BTI-based algorithm, 10.87 seconds for the recursive algorithm, and 5.26 seconds for our proposed algorithm. Our algorithm runs the fastest for all tested DEM. The speed-up ratios of our proposed algorithm over the second fastest algorithm is about 51%.

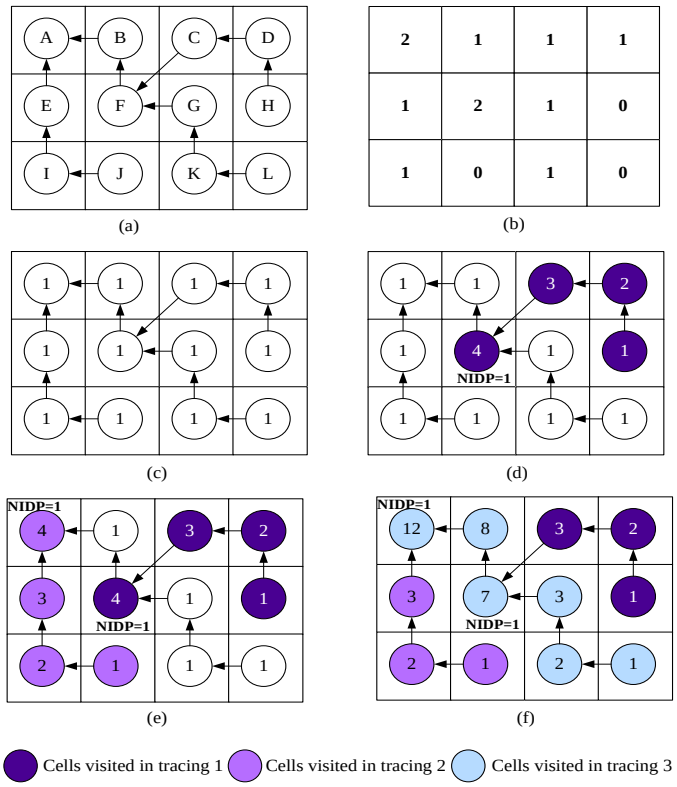

Figure 1. A worked example of the proposed algorithm. (a) A 3×4 DEM with flow directions. (b) Initial NIDP matrix. (c) The flow accumulation matrix is initialized with one. (d) Cells H, D, C and F are processed during the first round of tracing. The NIDP value of F is decreased by 1 and F is treated as an interior cell hereafter. (e) Cells J, I, E and A are processed during the second round of tracing. The NIDP value of A is decreased by 1 and A is treated as an interior cell hereafter. (f) Cells L, K, G, F, B and A are processed during the third round of tracing. The flow accumulation values of all cells are calculated after the tracing.

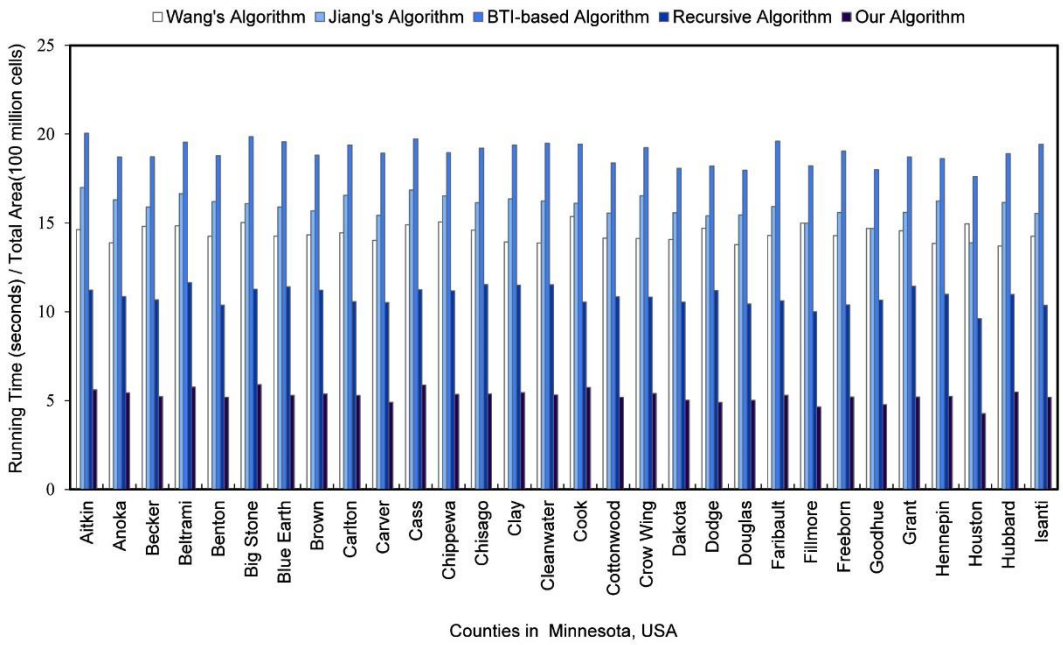

Figure 2. Running time (seconds) versus total area (100 million cells excluding NODATA cells) of five algorithms on the Windows system for 3-m LiDAR-based DEM data of 30 counties in Minnesota, USA.

Abstracts of the International Cartographic Association, 1, 2019.
29th International Cartographic Conference (ICC 2019), 15–20 July 2019, Tokyo, Japan | https://doi.org/10.5194/ica-abs-1-434-2019