Review on algorithms of dealing with depressions in grid DEM

Yi-Jie Wang a,b, Cheng-Zhi Qin a,b,c and A-Xing Zhu a,b,c,d,e

© 2019 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group, on behalf of Nanjing Normal University.

CONTACT Cheng-Zhi Qin qincz@lreis.ac.cn State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing, 100101, China

ARTICLE HISTORY
Received 6 January 2019
Accepted 29 March 2019

KEYWORDS
Grid digital elevation model (DEM); depressions; flat areas; digital terrain analysis; flow direction

ABSTRACT
Depressions in grid digital elevation models (DEMs) need to be dealt with before the topographic attributes (such as specific catchment area) and terrain features (such as drainage networks) related to flow directions can be derived from DEMs in a hydrologically-correct manner. Many depression-processing algorithms, which adopt different strategies and take different information under consideration for determining correct flow directions in depressions, have been proposed. However, currently, there is still no one algorithm which can satisfactorily deal with depressions in grid DEMs under various application contexts. In this paper, we review existing depression-processing algorithms based on the adopted strategies (i.e. the DEM-revising strategy and the DEM-unchanging strategy). Algorithms with the DEM-revising strategy especially are discussed in detail according to their designs relating to the revision of DEM elevations, i.e. the smoothing filter, depression filling, depression breaching (or carving), using other qualified data, and applying different algorithms to depressions with different characteristics. Existing ways of improving the computation efficiency of depression-processing algorithms are also presented, i.e. serial algorithm optimization and parallel algorithms. Lastly, we discuss a possible design for an optimal depression-processing algorithm which may be developed in the future.

1. Introduction

Diverse topographic attributes and terrain features can be derived using digital terrain analysis on grid digital elevation models (DEMs). Many of them are widely used as important inputs for geographical modeling and applications, including those related to estimating overland flow, such as specific catchment area, topographic wetness index, and drainage networks. Their calculations have a general step of determining flow direction(s) for each cell of a grid DEM in the direction of the steepest downslope neighboring cell (as in single flow direction algorithms) or the direction of its multiple downslope neighboring cells (as in multiple flow direction algorithms) (Wilson 2012, 2018). However, this step often faces challenges under some problematic situations in grid DEMs.

It is common that some local elevation minima (pits or closed depressions, called ‘depressions’ hereinafter) without lower neighborhoods exist in a grid DEM. Especially in the earlier grid DEMs, most depressions are spurious features (artifacts) rather than real terrain features, which arise from errors during DEM generation (O’Callaghan and Mark 1984; Martz and Garbrecht 1993; Rieger 1998). The flow directions for cells inside these depressions will be incorrect if the normal method according to the downslope neighboring cells is adopted. Accordingly, topographic derivatives related to the flow directions and their hydrological applications would be impacted by this issue. Thus, it is necessary to deal with depressions in grid DEMs, which is one of the main functions of DEM pre-processing (Gruber and Peckham 2009; Wilson 2018).

Many algorithms dealing with depressions in grid DEMs (depression-processing algorithms for short) have been proposed (e.g. Jenson and Domingue 1988; Hutchinson 1989; Martz and Garbrecht 1998; Planchon and Darboux 2002). They adopt different strategies and also take different information (such as the qualified stream network data) under consideration for determining the flow directions in depressions. Meanwhile, because these algorithms are increasingly computing-intensive and time-consuming, some algorithms have also been designed to improve their computation...
efficiency (e.g. Wang and Liu 2006; Wallis et al. 2009; Barnes, Lehman, and Mulla 2014). Currently, there is still no one algorithm which can satisfactorily deal with depressions in grid DEMs under various application contexts (such as the application target of the processed DEM, the study area characteristics, and the characteristics of data).

For the future development of a satisfactory solution to this issue, in this paper, we review existing depression-processing algorithms. They are discussed according to the adopted strategies, which can be classified into two main types: 1) The algorithms with the DEM-revising strategy, which change the elevations of cells of depressions (Section 2); and 2) the algorithms with the DEM-unchanging strategy, which determine flow directions in depressions without revising DEM (Section 3). Section 4 discusses existing ways of improving the computation efficiency of the depression-processing algorithms. Discussions and conclusion are given in Section 5.

2. Depression-processing algorithms with the DEM-revising strategy

The basic idea of the DEM-revising strategy is that the depressions in a grid DEM normally are often spurious features rather than real terrain features, and thus elevations of depression cells should be revised to remove the depressions in the DEM. Thus, normally, a hydrologically-corrected DEM can be acquired by which the correct flow directions can be determined according to the down-slope neighboring cells. The differences among the algorithms with the DEM-revising strategy are mainly centered on how to revise DEM elevations to remove the depressions in DEMs (Table 1).

2.1 Smoothing filter

As a representative of the primary works on this issue, O’Callaghan and Mark (1984) used a smoothing filter to remove as many problematic features in a DEM as possible. A smoothing filter works mainly by removing shallow and small artificial depressions. For those with larger areas, a smoothing filter needs to be applied to a DEM with more iterations. Even so, this cannot ensure the removal of all depressions (especially larger and deeper depressions) in the DEM. Such indiscriminate application of the smoothing filter to the entire DEM often causes a problem of over-smoothing (Figure 1(a)), i.e. a loss of real information in non-problematic areas of the DEM (Band 1986; Tribe 1992; Soille, Vogt, and Colombo 2003).

2.2 Depression filling

To avoid over-smoothing with the smoothing filter, Jenson and Domingue (1988) proposed to fill the depressions in a DEM by raising the elevations of cells of a depression to the minimum elevation surrounding the depression. Compared with the former depression-filling algorithm with a similar idea (e.g. Marks, Dozier, and Frew 1984), Jenson and Domingue (1988)’s algorithm can deal well with looping depressions which are normal occurrences in DEMs with many depressions in a relatively flat area. Jenson and Domingue (1988)’s algorithm is well known and is the most widely used algorithm. It has been implemented in most GIS software and tool packages specified to digital terrain analysis and hydrological modeling, such as ArcInfo (ESRI 1999), GRASS (GRASS Development Team 2003), HEC GEO-HMS USACE (US Army Corps of Engineers) 2002, TauDEM (Tarboton 1997), TAPES-G (Gallant and Wilson 1996), RiverTools (Peckham 2009), and so on. Some other algorithms with similar ideas have also been proposed to delineate watersheds in grayscale images (e.g. DEMs with integer elevation values) (Vincent and Soille 1991; Beucher and Meyer 1992). Zhu, Tian, and Zhao (2006) noted that Jenson and Domingue (1988)’s algorithm cannot properly fill the compound depressions and thus designed a depression-filling algorithm to fill compound depressions.

Note that by this method the depressions in a DEM will be filled to be flat areas where the cells have the same elevation as their neighboring cell. Thus, all flats in the depressionless DEM will be processed using a flat-processing algorithm so that the drainage networks derived from the resulting flat-processed DEM will be hydrologically correct across the whole area (Mackay and Band 1998; Liang and Mackay 2000; Zhu, Tian, and Zhao 2006). The primary flat-processing algorithms adopted the DEM-unchanged strategy, which tries to assign flow direction on each cell in the flats without revising the elevations of the flat cells (Liang and Mackay 2000; Zhu, Tian, and Zhao 2006). For example, Jenson and Domingue (1988) proposed to assign flow directions of the cells in flats to their nearest cell with a determined flow direction using a repeated procedure. Jenson and Domingue (1988) flat-processing algorithm often results in an unrealistic parallel flow pattern in the flats (Tribe 1992; Martz and Garbrecht 1998). Tribe (1992) proposed a flat-processing algorithm to define the straight line between the inflow and the outlet (or the arbitrarily selected among multiple outlets) of a flat to be the main drainage through the flat, and then to assign the flow directions of other cells in the flat towards the nearest cells on the main
### Table 1. Classification of existing depression-processing algorithms with the DEM-revising strategy.

| Basic way          | Representative algorithm/ reference | Premises and characteristics                                                                 | Flat processing                             | Applicability                                                                                               | Limits                                                                                           |
|--------------------|------------------------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Smoothing filter   | O’Callaghan and Mark 1984          | All depressions are spurious features                                                          | Consistently process                        | Fit for calculating topographic attributes on slope surface                                                | Over-smoothing; cannot remove all depressions                                                    |
| Depression filling | Jenson and Domingue 1988           | All depressions are spurious features caused by underestimating DEM elevations                | Need further process                        | Available for looping depressions; widely implemented and used                                             | Separation of processing depressions and flats; might unrealistically raise elevations largely |
|                    | Planchon and Darboux 2002          | All depressions are spurious features caused by underestimating DEM elevations; 
revise all cells inside depressions and flats to be gentle slope | Consistently process                        | Flexible; Fit for calculating topographic attributes on slope surface                                     | Might unrealistically raise elevations largely                                                   |
| Depression breaching (or carving) | Depression breaching algorithm (Martz and Garbrecht 1998; Martz and Garbrecht 1999) | Depressions are mainly caused by spurious blockage due to overestimating elevations of narrow border of depressions; remove all depressions by lowering one or two cells at depression outlet and filling remaining depressions | Need further process                        | Hydrologically-corrected DEM for drainage network extraction                                           | Many depressions are not with spurious narrow blockage; cannot deal with nested depressions; similar limits as depression filling by Jenson and Domingue (1988) |
| Depression-carving algorithm (Soille, Vogt, and Colombo 2003; Soille 2004a) | Remove all depressions by carving a descending path from depression bottom to depression outlet and then to the nearest cell being lower than the depression bottom | Need further process                        | Hydrologically-corrected DEM for drainage network extraction                                            | Revising elevation outside depressions; separation of processing depressions and flats          |
| Filling and carving algorithm (Soille 2004b) | Minimize the total modifications of depressions                                              | Need further process                        | Hydrologically-corrected DEM for drainage network extraction                                            | Similar limits as depression filling by Jenson and Domingue (1988)                             |
| Using other qualified data | Stream-burning (Saunders and Maidment 1996) | The “known” stream network is available; lower all stream cells and/or raise all non-stream cells | Consistently process                        | Hydrologically-corrected DEM for drainage network extraction; consistency with the “known” stream network | Abrupt jumps in elevation on the resulting DEM; those depressions away from the known stream network could exist |

(Continued)
| Basic way | Representative algorithm/ reference | Premises and characteristics                                                                 | Flat processing                                                                 | Applicability                                                                 | Limits                                                                 |
|-----------|-------------------------------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------|------------------------------------------------------------------------|
|           | ANUDEM (Hutchinson 1989)            | The “known” stream network is available; drainage enforcement by applying the stream network as boundary condition to an iterative finite difference interpolation method | Consistently process Hydrologically-corrected DEM for drainage network extraction; smooth without abrupt jumps between the stream and non-stream cells; consistency with the “known” stream network; widely implemented and used | Those depressions away from the known stream network could exist            |
|           | Lindsay and Creed 2005              | Choose the depression-removing algorithm with the least impact among depression-filling and depression-breaching algorithms for removing each individual depression | Could consistently process by considering existing flat-processing algorithms | Flexible framework; deal with depressions with different characteristics in an adaptive manner | To be extended to build an optimal depression-processing workflow         |
drainage with a convergent pattern. Compared with Jenson and Domingue (1988) flat-processing algorithm, Tribe’s (1992) flat-processing algorithm can produce a realistic convergent drainage pattern in the flats. However, because the straight main drainage of a flat could run across the outside the flat when the flat has an irregular shape, Tribe’s algorithm treats such cells as cells in the flat, which is unrealistic. Tribe’s algorithm also cannot deal with flats without ascending border cells (i.e. plateau or flat summits) (Soille, Vogt, and Colombo 2003). Different orders of searching among cells in a flat will arbitrarily result in different and unreasonable patterns of flow directions in the flat (Liang and Mackay 2000).

Garbrecht and Martz (1997) and Martz and Garbrecht (1998) argued that existing flat-processing algorithms without DEM-revising arbitrarily assign flow directions of flat cells only based on their adjacent lower cells and thus they produce unrealistic drainage patterns in the flats. This shortcoming makes these flat-processing algorithms without DEM-revising methods only available for DEMs with small and infrequent flats (Garbrecht and Martz 1997). With the basic idea that natural drainages generally go away from higher terrain and towards lower terrain, Garbrecht and Martz (1997) proposed a flat-processing algorithm by revising elevations in flats to impose two very small gradients. One of the gradients makes the flow from the higher terrain around the flats could drain into the flats, while the other makes the flow inside flats towards the neighboring lower terrain. Both this flat-processing algorithm and a breaching algorithm for removing depressions (see Section 2.3) (Martz and Garbrecht 1998) have been implemented in TOPAZ (TOpographic PArameterization), a digital terrain analysis software for watershed modeling (Martz and Garbrecht 1998).

The separation of processing depressions and flats makes corresponding algorithms more complicated for

Figure 1. The schematic diagram of the depression-processing algorithms with the DEM-revising strategy. a) Smoothing filter. b) Depression Filling. c) Depression Breaching (or Carving). d) Drainage enforcement.
not only algorithm design but also implementation. A flat in a DEM can be viewed as a special case of depressions with a depth of zero or infinitely small (Martz and de Jone 1988), or a set of such looping depressions. Furthermore, flats in a DEM are often artifacts which arise from interpolation errors during DEM generation, truncation of interpolated values, and the limited spatial resolution of the grid DEM (O’Callaghan and Mark 1984; Martz and Garbrecht 1993; Rieger 1998). Cells inside flats, whose flow directions cannot be determined as in normal situations due to the missing downslope neighboring cells, should also be processed before calculating topographic derivatives related to flow directions and conducting their hydrological applications. Therefore, it would be more convenient to remove depressions and flats in a consistent manner (Planchon and Darboux 2002; Pan, Stieglitz, and Mckane 2012; Huang and Lee 2015; Su et al. 2015).

One idea for removing depressions and flats in a consistent manner is to raise elevations from near the outlet cell of each depression (or flat) to the distant cells inside the depression (or flat) iteratively with a given downslope gradient progressing towards the lower terrain (Pan, Stieglitz, and Mckane 2012; Huang and Lee 2015). Such a downslope gradient could be set with a very small elevation increment (Huang and Lee 2015), or through linear interpolation between the outlet cell with the lower elevation and the edge cells with higher elevation on the opposite side of the depression (or flat) (Pan, Stieglitz, and Mckane 2012).

The other idea of raising elevations to remove depressions and flats in a consistent manner is to simulate the flooding and draining process used for achieving the hydrological correction of a DEM (Planchon and Darboux 2002). Planchon and Darboux (2002) algorithm first inundates the surface (i.e. the grid DEM for a study area) with a thick enough layer of ‘water’ which results in an inundated DEM. The excess water on each of the cells of the inundated DEM is then iteratively removed (i.e. lowering the inundating elevations) until the elevation of every individual cell in the depressions and flats ensures that the cell has a downslope to neighboring cell(s) with a user-assigned, very small slope gradient, which makes the raising of the elevations as small as possible. Note that if the user-assigned slope gradient is zero, the algorithm will fill all depressions to be flats. This makes this algorithm flexible to combine with other flat-processing algorithms.

Compared with other depression-processing algorithms, algorithms which can revise all cells inside depressions and flats to have gentle slopes can prevent unreasonable flow direction results inside depressions and flats (Figure 1(b)). Therefore, such results for DEMs are suitable for calculating some topographic attributes on the slope surface (such as the topographic wetness index) using multiple flow direction algorithms instead of a single flow direction algorithm (Qin et al. 2011). Some algorithms based on this method have also been proposed to improve the computation efficiency (e.g. Wang and Liu 2006; Barnes, Lehman, and Mulla 2014), which will be discussed in Section 4.

Note that the depression-filling algorithms treat all depressions as spurious features caused exclusively by underestimating the DEM elevations and remove all of them (no matter how large or how deep) by raising the original DEM. This is not accurate because spurious depressions could also be caused by overestimating DEM elevations (Martz and Garbrecht 1998). Furthermore, some of the depressions in DEM are real terrain features (such as closed depressions in real terrain like ponds in wetlands). Only those spurious depressions should be filled (Tribe 1992). Thus, the depression-filling algorithms might unrealistically raise elevations largely for large areas and impact some relief-mattered analyses (such as calculating slope gradient, estimating earthwork volume, and simulating soil redistribution along slope) on the resulting depressionless DEM.

### 2.3 Depression breaching (or carving)

To relieve the above-mentioned problems in the depression-filling algorithms, depression breaching (or carving) algorithms were proposed to remove all depressions (treated as spurious features caused by under- or overestimation elevation errors) in a DEM to produce a hydrologically-corrected DEM (Martz and Garbrecht 1998, 1999). The basic idea of the depression-breaching algorithms is that the revision of DEM should be much smaller than that of the algorithms based solely on filling depressions. Martz and Garbrecht (1998, 1999)'s depression-breaching algorithm lowers the elevation of one or two cells at the outlet of each depression to eliminate or reduce the size of the depression. Then those remaining and size-reduced depressions were filled to become flats to be dealt with by the following flat-processing algorithm (Garbrecht and Martz 1997; Martz and Garbrecht 1998). Martz and Garbrecht (1999)'s depression-breaching algorithm actually assumes that depressions are mainly caused by spurious flow blockage due to overestimating the elevations of the cells of the narrow borders of the depressions. For depressions in a DEM which are far from being in accordance with this assumption, which is not rare, Martz and Garbrecht (1998, 1999)'s solution of first breaching and then filling
depressions has a similar problem as the depression-filling algorithms. Meanwhile, Martz and Garbrecht (1999)’s depression-breaching algorithm cannot deal with nested depressions (Reuter et al. 2009).

To avoid the problem of Martz and Garbrecht (1998, 1999)’s depression-breaching algorithm, Soille, Vogt, and Colombo (2003) and Soille (2004a) proposed a depression-carving algorithm. The basic idea is to create a descending path from the bottom of a depression to the proper outlet of the depression and then to carve the terrain (e.g. lowering the elevations) along this path through the outlet until reaching the nearest cell which is lower than the bottom of the depression (Figure 1(c)). A similar idea had also been proposed by Rieger (1992, 1998). To further reduce the total modifications on depression cells, Soille (2004b) improved Soille, Vogt, and Colombo (2003)’s depression-carving algorithm to be a filling and carving algorithm. With the filling and carving algorithm, depressions are filled to a certain level and then carved to that level by Soille, Vogt, and Colombo (2003)’s depression-carving algorithm. The level is set to minimize the total elevation modifications or the number of modified cells of the depressions. In such a way, the depression-carving process proposed by Soille, Vogt, and Colombo (2003) and Soille (2004b) can remove all depressions with much fewer modifications to the depression cells than depression-filling algorithms, and the resulting carved paths follow valley bottoms which means the resulting DEMs from this algorithm are appropriate for drainage network delineation (Poggio and Soille 2012). Schwanghart et al. (2013) further combined a constrained breaching process with Soille (2004a)’s carving algorithm to improve the reasonableness of the drainage network derived in low-relief area.

However, for those depressions which are spurious features along the slope surface with less local variance in the actual terrain, the carving process creates a sudden elevation salutation between the carved cells and the neighboring unchanged cells in the depressions. This means the DEMs resulting from the depression-carving algorithm might not be used to calculate reasonable topographic attributes (such as slope gradient, curvatures, specific contributing area, and topographic wetness index) on those unchanged cells (especially those neighboring the carved cells) in depressions.

2.4 Using other qualified data

All of the above-mentioned algorithms considered only the data of the grid DEM under processing. Note that drainage network extraction based on grid DEMs is highly sensitive to elevation errors in DEMs (Lindsay and Evans 2008). Both those elevation errors outside depressions and elevation modification on depression cells using the above-mentioned algorithms with the DEM-revising strategy might cause the drainage networks extracted based on the corresponding DEM to substantially differ from reality. Thus, when there are other qualified data (such as independently digitized stream networks) available for an area, data fusion could be considered to deal with depressions in a DEM so that the drainage networks extracted from the depression-removed DEM are in accordance with reality (Morris and Heerdegen 1988; Kenny, Matthews, and Todd 2008; Yamazaki et al. 2012; Schwanghart et al. 2013).

A natural way of using other qualified stream network data to remove depressions in a DEM is to lower the elevations of depression cells along the digitized river networks to provide consistency with the ‘known’ stream network (Figure 1(d)). This is called stream-burning (Saunders and Maidment 1996), reconditioning (Hellweger 1996), or drainage enforcement (Hutchinson 1989).

The simplest algorithm for this is to lower all stream cells dramatically and/or raise all non-stream cells by a given offset height (Saunders and Maidment 1996). The stream-burning algorithm causes abrupt jumps in elevation on the resulting DEM, which mainly aim to produce the fully-connected drainage network with single-cell width as the ‘known’ for some hydrological models, ignoring its applicability in other applications (Callow et al. 2007).

To avoid introducing the abrupt jumps in elevations between stream cells and non-stream cells, Hellweger (1996) designed a DEM reconditioning algorithm and implemented the corresponding tool named AGREE. AGREE applies a known stream (or ridge) network to lowering (or raising) the elevation of cells along the network for a user-assigned height offset. Then, the elevations of cells in a horizontal buffer of the network with a user-assigned distance are revised to ensure smooth elevation change and straight flow paths between the network cells and the buffer border cells. The arbitrary user-assigned distance in AGREE has a large influence on the resulting elevation modification. AGREE may cause significant elevation modification far beyond the scope of the depressions (Callow et al. 2007).

A more reasonable drainage enforcement algorithm is the ANUDEM algorithm (Hutchinson 1989; Hutchinson et al. 2013). ANUDEM applies a single-cell width stream network as a boundary condition to an
iterative finite difference interpolation method (i.e. thin-plate spline) for a DEM with depressions. By this means, all depressions intersected by the known stream network can be removed, and each stream cell has a downslope gradient. The resulting DEM is smooth and without abrupt jumps between the stream and non-stream cells. Note that potential elevations across the entire DEM could be revised by ANUDEM. However, depressions far away from the known stream network could still exist in the resulting DEM. The drainage network extracted from the resulting DEM is ensured to be very close to (but not necessary to be completely same as) the known stream network for some areas Kenny, Matthews, and Todd 2008. ANUDEM has been implemented in multiple separate tools (e.g. ANUDEM Version 5.3, and the Topo-to-Raster tool in ArcGIS Desktop Version 10.4) and widely used in real applications.

Note that the way of using other qualified data can deal with not only depressions in DEMs but also other areas in DEMs to result in a more reasonable (e.g. hydrologically-correct) DEM or flow-direction grid for potential applications. For example, Turcotte et al. (2001) developed an approach in which a digital river and lake network (DRLN) is used to direct the DEM-revision and flow direction determination for a reasonable drainage network result, especially around lake areas. ANUDEM allows users to adopt multiple qualified data (such as streamlines, lakes, irregularly spaced elevation points, contour lines, and cliff lines) to revise DEMs to be hydrologically-correct (Hutchinson et al. 2013). This goes beyond the processing of depressions and flats in a DEM.

Specifically considering the issue of depression-removing, Soille, Vogt, and Colombo (2003) proposed adaptive drainage enforcement to avoid unnecessary elevation modification outside depressions caused by the above-mentioned drainage enforcement algorithms. The basic idea of adaptive drainage enforcement is to apply drainage enforcement only to those places where the extracted drainage from a DEM deviates obviously from the known drainage network. This idea could be combined with other existing depression-processing algorithms.

Meanwhile, depressions away from the known stream network could exist in the DEM resulting from above-mentioned algorithms using other qualified data Kenny, Matthews, and Todd 2008. This phenomenon cannot be ignored when many depressions are not along the stream network. This situation could be normal especially when the resolution of the DEM for a low-relief area is pretty fine (e.g. 5-m resolution) but the independent stream network data for this area contains less detail at a smaller scale (e.g. 1:500,000) or contains a high generalization level.

### 2.5 Applying different algorithms to depressions with different characteristics

A combination of several algorithms with different characteristics, each corresponding to some depressions with specific characteristics in a DEM, could result in satisfactory depression-processing (Lindsay and Creed 2005; Kenny, Matthews, and Todd 2008). Such a method first judges the characteristics of each individual depression in a DEM and then applies the proper depression-processing algorithm to the depression. A typical example is Lindsay and Creed (2005a)’s impact reduction approach (IRA). For each depression in the DEMs, IRA chooses the depression-removing algorithm with the least impact among depression-filling and depression-breaching algorithms for the removal of this depression. The impact is quantified in terms of the count of elevation-revising cells of the depression and the corresponding mean absolute elevation difference before and after depression removal. The automatic IRA was implemented in the Terrain Analysis System (TAS) software (Lindsay 2005), currently known as the WhiteBox Geospatial Analysis Tools.

Applying different algorithms to depressions with different characteristics could deal with depressions with different characteristics in an adaptive manner, which would be more reasonable for real applications with varied application contexts. This will be further discussed in the last section.

### 3. Depression-processing algorithms with the DEM-unchanged strategy

The above-mentioned algorithms with the DEM-revising strategy mainly assume that all or most of the depressions in DEMs are spurious features rather than real terrain features. Such an assumption is increasingly unlikely to be true for the high quality, high precision, and fine spatial resolution DEMs (such as LiDAR DEMs) produced in recent years, which can well portray natural depressions (Lane and Chandler 2003; Arnold 2010). While some studies tried to distinguish between the spurious and natural depressions in a DEM (e.g. Hutchinson 1989; Lindsay and Creed 2006), every depression in the DEM needs to be further handled to determine the reasonable flow direction and the drainage structure across (and within) the depression.

For natural (or assumed to be natural) depressions in a DEM, algorithms with the DEM-unchanged strategy (e.g. Morris and Heerdegen 1988; Ehlschlaeger 1989;
Rodhe and Seibert (1999; Chou et al. 2004) have been proposed to keep the DEM unchanged and to determine reasonable flow directions for depression cells, which do not need to be in accordance with the downslopes shown by the elevations of the depression cells. The basic idea of the DEM-unchanged strategy is that natural depressions should be kept unchanged in the DEM for the precise description of the actual terrain. Meanwhile, it is thought that the continuity of the flow should not be interrupted by the depressions in the DEM.

In essence, every algorithm with the DEM-revising strategy could be treated as an algorithm with the DEM-unchanged strategy if the depressionless DEM resulting from the DEM-revising algorithm was discarded after the flow direction grid was calculated and kept. Thus, depression-removing ideas for algorithms using the DEM-revising strategy could be adopted for the development of algorithms with the DEM-unchanged strategy when the revision is conducted by adjusting the flow direction instead of the elevation for the depression cells. This means that the DEM-unchanged version of an algorithm using the DEM-revising strategy often requires extra data record (e.g. the adjusted flow direction) and potentially a more complicated algorithm design with a different output (e.g. the adjusted flow direction grid or the corresponding flow accumulation result, instead of a depressionless DEM). Also, the possible problems of a DEM-unchanging algorithm could also happen in its DEM-unchanged version.

For example, Morris and Heerdegen (1988)’s unblocking algorithm holds a similar idea to the depression-carving algorithm. However, Morris and Heerdegen’s algorithm only traces and revises the flow direction along the carved path recorded in an array of cells which are waiting for processing instead of changing the elevation. Morris and Heerdegen’s algorithm fails to deal with nested depressions. Based on a similar depression-carving idea, Byun and Seong (2015)’s algorithm can deal with nested depressions in DEMs and extract longitudinal stream profiles without changing the DEMs. Based on the depression-filling idea, Arnold (2010) proposed a ‘fill and overflow’ algorithm to calculate flow accumulation without changing the DEMs which are to be filled. Following the fill-and-overflow idea, Wu et al. (2015) proposed a localized contour tree algorithm to extract the hierarchical wetland depressions from high-resolution LiDAR DEM. For such hierarchical wetland depressions, Wu and Lane (2017) further proposed a semi-automated approach to identifying their catchment and wetland connectivity. Based on a drainage enforcement idea, Kenny, Matthews, and Todd (2008) proposed an approach to altering the flow direction grid by imposing the known hydrological networks for both depressions and flats.

From another perspective, some specific designs in algorithms using the DEM-unchanged strategy could also be combined with algorithms using the DEM-revising strategy for improvement of one of the specific problems regarding dealing with the depressions in DEMs. For example, the idea of determining the optimal outlet for individual depression in Chou et al. (2004)’s algorithm using the DEM-unchanged strategy could be combined with algorithms using the DEM-revising strategy.

4. Efforts on improving computation efficiency of algorithms
The above mentioned depression-processing algorithms are designed to be an iterative or even recursive process, and thus are computing-intensive and time-consuming, especially for DEMs with a large number of cells and many large depressions. For example, Jenson and Domingue (1988)’s depression-filling algorithm scans each depression in a DEM sequentially. Assuming that there are N cells in the DEM, both the count of depressions and the count of cells to be scanned for each depression is O(N). The time complexity of Jenson and Domingue’s algorithm is O(N^2). Planchon and Darboux (2002)’s algorithm is comparatively faster, with a time complexity which is theoretically O(N^{1.5}) for the worst case and can be further improved to be O(N^{1.25}) for natural surfaces by the implementation of the tree exploration process instead of iterative scanning.

There have been also many algorithms designed specifically to improve the computation efficiency of one existing depression-processing algorithm. There are mainly two ways of improving computation efficiency. The first is via serial algorithm optimization, which revises an existing algorithm to a faster serial version by using more efficient processing order and/or data structures. The second way is to design corresponding parallel algorithms by taking advantage of the power of advanced computing platforms (Wang and Armstrong 2009).

4.1 Serial algorithm optimization
For the serial algorithm optimization, the data structure of the queue was often adopted to change the time-consuming iterative scanning across the whole DEM to be more efficient at searching (e.g. Soille and Gratin...
Wang and Liu (2006) argued that Planchon and Darboux (2002)'s iterative scanning method (although faster among different depression-filling algorithms) across the whole DEM with eight alternating directions in a fixed order is still relatively time-consuming. Wang and Liu (2006) proposed an efficient depression-filling algorithm by using the least-cost search and priority queue. The least-cost search algorithm is adopted to minimize the search time of finding the optimal paths from the outlets on the margin of a DEM to the interior cells. Meanwhile, the priority queue data structure is adopted to keep those cells raised in an order. By this process, Wang and Liu (2006)'s algorithm avoids the blind iterative scanings in Planchon and Darboux (2002)'s algorithm and has a time complexity of $O(N \log_2 N)$. Their results showed that their algorithm was more than three times faster than Planchon and Darboux (2002)'s algorithm.

To further improve the efficiency of Wang and Liu (2006)'s algorithm, Liu, Zhang, and Xu (2009) also used the priority queue and adopted an additional two-dimensional Boolean array to indicate cells which are to be processed, preferring this instead of Wang and Liu (2006)'s judgement based on whether a cell presents itself in the priority queue. The time complexity of the algorithm implemented by Liu, Zhang, and Xu (2009) is $O(N \log_2 K)$, where $K$ is the count of unique elevation levels for the DEM and could be less than $N$ (Barnes, Lehman, and Mulla 2014).

Barnes, Lehman, and Mulla (2014) argued that the priority queue adopted in the above algorithms unnecessarily consumes execution time for the repetitive sorting process. Note that a cell which is being raised must have the same priority as the other cell which triggers its raise and has had the highest priority in the current priority queue. Therefore, it need not push the raised cell into the priority queue, which would incur a sorting cost. Barnes, Lehman, and Mulla (2014) adopted a first-in-and-first-out queue to receive the raised cell at cost $O(1)$ and thus designed an algorithm with a time complexity of $O(M \log_2 M)$, where $M$ is smaller than the cell count. According to their experimental results, the algorithm proposed by Barnes, Lehman, and Mulla (2014) achieved runtimes 16% faster than Wang and Liu (2006)'s algorithm using the priority queue.

Zhou, Sun, and Fu (2016) proposed a variant algorithm based on the algorithms proposed by Wang and Liu (2006) and Barnes, Lehman, and Mulla (2014). They adopted two plain queues to process cells outside depressions and flats in a DEM, which often comprise the majority of cells in the DEM. By this means, the count of cells which need to be processed by the priority queue could be reduced dramatically. Zhou, Sun, and Fu (2016)'s experimental results showed that their algorithm performed at a rate which was 44.6% faster than Barnes, Lehman, and Mulla (2014)'s algorithm.

Note that serial algorithms dealing with depressions and flats in DEMs always face the bottleneck of processing a massive DEM within the comparatively limited memory of personal computers. Although existing efforts towards serial algorithm optimization show the design art of serial algorithms, the normal way of optimizing serial algorithms for faster execution is often related to the adoption of additional data structures requiring more memory consumption. This puts more limits on the size of the processed DEM. The method of improving computation efficiency using serial algorithm optimization is difficult to suit to current application demands which normally include a large area under a fine resolution.

### 4.2 Parallel algorithms

Parallel computing based on various parallel computing platforms (e.g., graphics processing unit (GPU), symmetrical multiprocessor (SMP), and Beowulf cluster) has been widely adopted to not only speed up digital terrain analysis algorithms (e.g., Tesfa et al. 2011; Qin and Zhan 2012; Qin et al. 2017) but also to make the algorithms applicable to massive DEMs which will overflow the limited memory of a personal computer. For exerting the power of parallel computing for a specific parallel computing platform, parallel algorithms should be designed and implemented based on the specific parallel programming model which is available to the platform (e.g., Wallis et al. 2009; Qin and Zhan 2012; Zhou, Sun, and Fu 2016). Examples of parallel programming models include the Open Multi-Processing (OpenMP), which is a widely used multithreading programming model used for SMP parallel computing devices (such as multiprocessors in standard personal computers), the compute unified device architecture (CUDA) for GPU, and the message passing interface (MPI) for distributed memory parallel machines (such as the Beowulf cluster) (Qin et al. 2014a).

In general, the design and implementation of parallel algorithms is much more complicated than that of the serial algorithm (Qin et al. 2014a). This is not only because of the steeper learning curve of parallel programming models, but also the difficulties of parallel algorithm design due to the different parallelizability of serial algorithms. When the high parallelizability in algorithms using local and focal computation (such as
slope gradient calculation) is comparatively easy for parallel algorithm design, the iterative and even recursive process normally adopted in depression-processing algorithms makes the design of corresponding parallel algorithms more difficult. For example, Qin and Zhan (2012) proposed the strategy of designing the parallel algorithm based on CUDA for recursive multiple flow direction algorithms. This is based on first transferring the recursive algorithms to become serial iterative algorithms with high parallelizability.

Wallis et al. (2009) implemented Planchon and Darboux (2002)’s algorithm to an MPI-based parallel algorithm in TauDEM, a digital terrain analysis package. Barnes (2016) proposed an MPI-based parallel implementation for the serial depression-filling algorithm optimized by Barnes, Lehman, and Mulla (2014). The basic idea of designing parallel MPI-based algorithms for iterative depression-filling algorithms is to divide the DEM into strips, each of which is iteratively processed while the necessary parallel communication between it and its neighboring strips is conducted after each round of iterations. Based on OpenMP and MPI, Zhou et al. (2017) proposed a parallel algorithm for the serial depression-filling algorithm optimized by Zhou, Sun, and Fu (2016). Qin and Zhan (2012) transformed Planchon and Darboux (2002)’s algorithm to be a CUDA-based parallel algorithm on the GPU.

As far as we know, for large size DEMs, all parallel versions of depression-processing algorithms performed much faster than their corresponding serial algorithms. The CUDA-based parallel algorithms on the GPU in a personal computer can reach a higher performance level with a lower cost, while the size of the processed DEM is limited by the memory size of the personal computer (Qin and Zhan 2012). When adopting parallel input/output (Qin, Zhan, and Zhu 2014b; Barnes 2016), MPI-based or MPI/openMP-based parallel algorithms can take advantage of high-performance clusters with higher scalable performances and can handle massive DEM well.

Currently, the design and implementation of parallel algorithms with different parallel computing platforms are still much more complex than that of serial algorithms, especially for the digital terrain analysis algorithm researchers who are familiar only with serial programming and should not have to focus on the complex parallel computing details of the various parallel programming models. Some efforts towards providing more easy-to-use parallel programming libraries have been made to support users (i.e. the researchers of digital terrain analysis or other geocomputation algorithms) in coding parallel programs in the manner of serial programming (Guan and Clarke 2010; Qin et al. 2014a; Guan et al. 2014). The basic idea of these easy-to-use parallel programming libraries is to encapsulate the parallel programming details of a specific parallel computing platform (Guan and Clarke 2010; Guan et al. 2014) or even multiple platforms (Qin et al. 2014a) so as to hide them from users. These efforts provide a promising step toward the easier design of parallel algorithms used for dealing with depressions and flats in DEMs, as well as of other raster-based geo-computation algorithms.

5. Discussion and conclusions

Maximum reduction of spurious depressions in a DEM is valuable for not only lowering the workload required for depression-processing algorithms, but also for intrinsically improving the DEM quality for terrain analysis. This can be done by developing more accurate DEM productions using advanced terrain surveying technology and/or by fusing multiple DEMs as well as other data sources (e.g. Yue et al. 2017; Yamazaki et al. 2017). However, depressions which could be either spurious or real terrain features always exist in DEMs and must be dealt with.

Among the existing algorithms, there is no one single algorithm which can be called optimal for dealing with the depressions in DEMs. Different algorithms are based on different assumptions (such as what kind of depressions in a DEM should be considered as spurious or real terrain features), each of which is partly in accordance with the actual situation of the DEMs produced by various methods for different areas. The various designs of these algorithms are hardly proper for all situations. Similar to other digital terrain analysis tasks, the ‘optimal’ depression-processing algorithm should be appropriate for the specific application context (such as the application target of the processed DEM, the study area characteristics, and the characteristics of data) (Qin et al. 2016). For example, some algorithms aim at producing a hydrologically-corrected DEM for drainage network delineation, while other algorithms remove spurious depressions in a DEM for calculating reasonable topographic attributes on the slope surface (such as the topographic wetness index). While some algorithms fail to deal with low relief area, some other algorithms are specific to low-relief wetland area (e.g. Wu et al. 2015; Wu and Lane 2017). The data characteristics, such as the fine or coarse spatial resolution, and the different data source (e.g. LiDAR DEM, SRTM DEM, etc.), will impact not only the count of depressions in DEMs of same area, but also impact the ratio of real depressions in all depressions in DEM. Unfortunately, although the knowledge on applicability of individual algorithms under various application
contexts is very important, such knowledge is still non-systematic and vague, which needs new methods of domain knowledge formalization (Qin et al. 2016).

A possible way to achieve an optimal depression-processing algorithm is to combine different algorithms, each of which is appropriate for dealing with depressions with specific characteristics. A notable example is Lindsay and Creed (2005)’s work (see Section 2.5); however, it is still far from optimal. Such an optimal algorithm would actually be a complex workflow instead of a single algorithm applied indiscriminately to the whole DEM. With this workflow, each depression in a DEM to be processed should be evaluated first, such as its possibility of being spurious or real terrain, its size and depth, the terrain position type around it, and so on. According to the evaluated results as well as other application context information regarding the application target and data availability, a specific process will then be chosen to process it in an adaptive manner. For example, for a depression where a qualified river line is available which is intersected with it, the drainage enforcement process should accordingly be adopted. Such a process would be more reasonable for use in real applications with various application contexts.

In the implementation of the above-suggested workflow, the computing-intensive processes could be accelerated by parallel computing. Note that because some depressions might be processed without changing their elevations, the determined flow direction of each cell in these areas should be also recorded as output. This means such a workflow will result in at least two outputs, i.e. the depressionless DEM (with some elevation-changed areas), and the determined flow direction grid. An additional output which should be considered by the workflow is an uncertainty grid, which quantitatively indicates the uncertainty of the processed result for each cell. This is because the evaluation and the choice of processing in the workflow are intrinsically uncertain rather than deterministic. Besides these outputs, the detailed workflow for a DEM should be also recorded and made accessible for users, which could be valuable for the users’ application of the processed DEM and the subsequent analysis of the results.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This study was funded by the National Natural Science Foundation of China (No. 41431177, 41422109), PAPD, and Outstanding Innovation Team in Colleges and Universities in Jiangsu Province. Supports to A-Xing Zhu through the Vilas Associate Award, the Hammel Faculty Fellow Award, and the Manasse Chair Professorship from the University of Wisconsin-Madison are greatly appreciated.

ORCID

Yi-Jie Wang http://orcid.org/0000-0002-0779-1189
Cheng-Zhi Qin http://orcid.org/0000-0002-5910-9807
A-Xing Zhu http://orcid.org/0000-0002-5725-0460

References

Arnold, N. 2010. "A New Approach for Dealing with Depressions in Digital Elevation Models When Calculating Flow Accumulation Values." Progress in Physical Geography 34: 781–809. doi:10.1177/0309133310384542.
Band, L. E. 1986. “Topographic Partition of Watersheds with Digital Elevation Models.” Water Research 22: 15–24. doi:10.1016/WR022001p00015.
Barnes, R. 2016. “Parallel Priority-Flood Depression Filling for Trillion Cell Digital Elevation Models on Desktops or Clusters." Computers & Geosciences 96: 56–68. doi:10.1016/j.cageo.2016.07.001.
Barnes, R., C. Lehman, and D. Mulla. 2014. “Priority-Flood: An Optimal Depression-Filling and Watershed-Labeling Algorithm for Digital Elevation Models.” Computers & Geosciences 62: 117–127. doi:10.1016/j.cageo.2013.04.024.
Beucher, S., and F. Meyer. 1992. “The Morphological Approach to Segmentation: The Watershed Transformation.” In Mathematical Morphology in Image Processing, edited by E. R. Dougherty, 433–481. New York, NY: CRC Press.
Byun, J., and Y. B. Seong. 2015. "An Algorithm to Extract More Accurate Stream Longitudinal Profiles from Unfilled DEMs.” Geomorphology 242: 38–48. doi:10.1016/j.geomorph.2015.03.015.
Callow, J., K. Van Niel, and G. Boggis. 2007. “How Does Modifying a DEM to Reflect Known Hydrology Affect Subsequent Terrain Analysis?” Journal of Hydrology 332: 30–39. doi:10.1016/j.jhydrol.2006.06.020.
Chou, T. Y., W. T. Lin, C. Y. Lin, W. C. Chou, and P. H. Huang. 2004. “Application of the PROMETHEE Technique to Determine Depression Outlet Location and Flow Direction in DEM.” Journal of Hydrology 287: 49–61. doi:10.1016/j.jhydrol.2003.09.026.
Ehlschlaeger, C. R. 1989. "Using the A¹ Search Algorithm to Develop Hydrologic Models from Digital Elevation Data." In International Geographic Information Systems (IGIS) Symposium 89: 275–281.
ESRI. 1999. Technical Documentation of ArcInfo, Version 8.0.1. Redland, CA: Environmental System Research Institute.
Garbrecht, J., and L. W. Martz. 1997. "The Assignment of Drainage Direction over Flat Surfaces in Raster Digital Elevation Models." Journal of Hydrology 193: 204–213. doi:10.1016/S0022-1694(96)03138-1.
GRASS Development Team. 2003. "GRASS GIS 5.1 Reference Manual." Available online. http://grass.baylor.edu/grass51/manuals/html51_user/index.html
Gruber, S., and S. Peckham. 2009. “Land-Surface Parameters and Objects in Hydrology.” In Geomorphometry: Concepts, Software, Applications, edited by T. Hengl and H. I. Reuter, 171–194. Amsterdam, Netherlands: Elsevier.

Guan, Q. F., and K. C. Clarke. 2010. “A General-Purpose Parallel Raster Processing Programming Library Test Application Using A Geographic Cellular Automata Model.” International Journal of Geographical Information Science 24: 695–722. doi:10.1080/13658810902984228.

Guan, Q. F., W. Zeng, J. F. Gong, and S. Yun. 2014. “pRPL 2.0: Improving the Parallel Raster Processing Library.” Transactions in GIS 18: 25–52. doi:10.1111/tgis.12109.

Hellweger, R. 1996. “Agree.Aml.” Center for Research in Water Resources, The University of Texas at Austin.

Huang, P. C., and K. T. Lee. 2015. “A Simple Depression-Filling Method for Raster and Irregular Elevation Datasets.” Journal of Earth System Science 124: 1653–1665. doi:10.1007/s12040-015-0641-2.

Hutchinson, M. F. 1989. “A New Procedure for Gridding Elevation and Stream Line Data with Automatic Removal of Spurious Wells.” Journal of Hydrology 106: 211–232. doi:10.1016/0022-1694(89)90073-5.

Hutchinson, M. F., J. L. Stein, J. C. Gallant, and T. I. Dowling. 2013. “New Methods for Incorporating and Analyzing Drainage Structure in Digital Elevation Models.” In Proceedings of the 3rd International Conference on Geomorphometry. Nanjing, China. Available at http://geomorphometry.Hutchinson2013

Jenson, S. K., and J. O. Domingue. 1988. “Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis.” Photogrammetric Engineering and Remote Sensing 54: 1593–1600.

Kenny, F., B. Matthews, and K. Todd. 2008. “Routing Overland Flow through Sinks and Flats in Interpolated Raster Terrain Surfaces.” Computers & Geosciences 34: 1417–1430. doi:10.1016/j.cageo.2008.02.019.

Lane, S. N., and J. H. Chandler. 2003. “The Generation of High Quality Topographic Data for Hydrology and Geomorphology: New Data Sources, New Applications and New Problems.” Earth Surface Processes and Landforms 28: 229–230. doi:10.1002/esp.479.

Liang, C., and D. S. Mackay. 2000. “A General Model of Watershed Extraction and Representation Using Globally Optimal Path Flows and Up-Slope Contributing Areas.” International Journal of Geographical Information Science 14: 337–358. doi:10.1080/13658810050024278.

Lindsay, J., and I. F. Creed. 2005. “Removal of Artefact Depressions from Digital Elevation Models: Towards a Minimum Impact Approach.” Hydrological Processes 19: 3113–3126. doi:10.1002/hyp.5835.

Lindsay, J. B. 2005. “The Terrain Analysis System: A Tool for Hydro-Geomorphic Application.” Hydrological Processes 19: 1123–1130. doi:10.1002/hyp.5818.

Lindsay, J. B., and I. F. Creed. 2006. “Distinguishing Actual and Artefact Depressions in Digital Elevation Data.” Computers & Geosciences 32: 1192–1204. doi:10.1016/j.cageo.2005.11.002.

Lindsay, J. B., and Evans M. G. 2008. “The influence of elevation error on the morphometrics of channel networks extracted from DEMs and the implications for hydrological modelling.” Hydrological Processes, 22: 1588–1603. doi:10.1002/hyp.6728.

Liu, Y., W. Zhang, and J. Xu. 2009. “Another Fast and Simple DEM Depression-Filling Algorithm Based on Priority Queue Structure.” Atmospheric and Oceanic Science Letters 2: 214–219. doi:10.1080/16742834.2009.11446800.

Mackay, D. S., and L. E. Band. 1998. “Extraction and Representation of Nested Catchment Areas from Digital Elevation Models in Lake-Dominated Topography.” Water Resources Research 34: 897–901. doi:10.1029/98WR00094.

Magalhães, S. V., M. V. Andrade, W. R. Franklin, and G. C. Pena 2012. “A New Method for Computing the Drainage Network Based on Raising the Level of an Ocean Surrounding the Terrain.” In Bridging the Geographic Information Sciences: International AGILE2012 Conference, 391–407, Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-29063-3_21.

Marks, D., J. Dozier, and J. Frew. 1984. “Automated Basin Delineation from Digital Elevation Data.” Geo-Processing 2: 299–311.

Martz, L. W., and E. de Jone. 1988. “CATCH: A FORTRAN Program for Measuring Catchment Area from Digital Elevation Data.” Computers & Geosciences 14: 627–640. doi:10.1016/0098-3004(88)90018-0.

Martz, L. W., and J. Garbrecht. 1993. “Automated Extraction of Drainage Network and Watershed Data from Digital Elevation Models.” Water Resources Bulletin 29: 901–908. doi:10.1111/j.1752-1688.1993.tb03250.x.

Martz, L. W., and J. Garbrecht. 1998. “The Treatment of Flat Areas and Depressions in Automated Drainage Analysis of Raster Digital Elevation Models.” Hydrological Processes 12: 843–855. doi:10.1002/(ISSN)1099-1085.

Martz, L. W., and J. Garbrecht. 1999. “An Outlet Breaching Algorithm for the Treatment of Closed Depressions in a Raster DEM.” Computers & Geosciences 25: 835–844. doi:10.1016/S0098-3004(99)00018-7.

Metz, M., H. Mitasova, and R. S. Harmon. 2011. “Efficient Extraction of Drainage Networks from Massive, Radar-Based Elevation Models with Least Cost Path Search.” Hydrology and Earth System Sciences 15: 667. doi:10.5194/hess-15-667-2011.

Morris, D. G., and R. G. Heerdegen. 1988. “Automatically Derived Catchment Boundaries and Channel Networks and Their Hydrological Applications.” Geomorphology 1: 131–141. doi:10.1016/0169-555X(88)90011-6.

O’Callaghan, J. F., and D. M. Mark. 1984. “The Extraction of Drainage Networks from Digital Elevation Data.” Computer Vision, Graphics, and Image Processing 28: 323–344. doi:10.1016/0734-189X(84)90011-0.

Pan, F., M. Stiegglitz, and R. B. Mckane. 2012. “An Algorithm for Treating Flat Areas and Depressions in Digital Elevation Models Using Linear Interpolation.” Water Resources Research 48: 229–235. doi:10.1029/2011WR010735.

Peckham, S. 2009. “Geomorphometry in RiverTools.” In Geomorphometry: Concepts, Software, Applications, edited by T. Hengl and H. I. Reuter, 411–428. Amsterdam, Netherlands: Elsevier.

Planchon, O., and F. Darboux. 2002. “A Fast, Simple and Versatile Algorithm to Fill the Depressions of Digital Elevation Models.” Catena 46: 159–176. doi:10.1016/s0341-8162(01)00164-3.

Poggio, L., and P. Soille. 2012. “Influence of Pit Removal Methods on River Network Position.” Hydrological Processes 26: 1984–1990. doi:10.1002/hyp.8290.
Qin, C. Z., B. B. Ai, A. X. Zhu, and J. Z. Liu. 2017. “An Efficient Method for Applying a Differential Equation to Deriving the Spatial Distribution of Specific Catchment Area from Gridded Digital Elevation Models.” *Computers & Geosciences* 100: 94–102. doi:10.1016/j.cageo.2016.12.009.

Qin, C. Z., X. W. Wu, J. C. Jiang, and A. X. Zhu. 2016. “Case-Based Knowledge Formalization and Reasoning Method for Digital Terrain Analysis–Application to Extracting Drainage Networks.” *Hydrology and Earth System Sciences* 20: 3379–3392. doi:10.5194/hess-20-3379-2016.

Qin, C. Z., and L. J. Zhan. 2012. “Parallelizing Flow-Accumulation Calculations on Graphics Processing Units – From Iterative DEM Preprocessing Algorithm to Recursive Multiple-Flow-Direction Algorithm.” *Computers & Geosciences* 43: 7–16. doi:10.1016/j.cageo.2012.02.022.

Qin, C. Z., L. J. Zhan, and A. X. Zhu. 2014b. “How to Apply the Geospatial Data Abstraction Library (GDAL) Properly to Parallel Geospatial Raster I/O?” *Transactions in GIS* 18: 950–957. doi:10.1111/tgis.12068.

Qin, C. Z., L. J. Zhan, A. X. Zhu, and C. H. Zhou. 2014a. “A Strategy for Raster-Based Geocomputation under Different Parallel Computing Platforms.” *International Journal of Geographical Information Science* 28: 2127–2144. doi:10.1080/13658816.2014.911300.

Qin, C. Z., A. X. Zhu, T. Pei, B. L. Li, T. Scholten, T. Behrens, and C. H. Zhou. 2011. “An Approach to Computing Topographic Wetness Index Based on Maximum Downslope Gradient.” *Precision Agriculture* 12: 32–43. doi:10.1007/s11119-009-9152-y.

Reuter, H. I., I. Hengl, P. Gessler, and P. Soille. 2009. “Preparation of DEMs for Geomorphic Analysis.” In *Geomorphometry: Concepts, Software, Applications*, edited by T. Hengl and H. I. Reuter, 87–120. Amsterdam: Elsevier.

Rieger, W. 1992. “Automated River Line and Catchment Area Extraction from DEM Data.” In *Proceeding of 17th ISPRS Congress*, edited by L. W. Fritz and J. R. Lucas, 642–649. Vol. 28. Washington, D.C.: IAPRS.

Rieger, W. 1998. “A Phenomenon-Based Approach to Upslope Contributing Area and Depressions in DEMs.” *Hydrological Processes* 12: 857–872. doi:10.1002/(ISSN)1099-1085.

Rodhe, A., and J. Seibert. 1999. “Wetland Occurrence in Relation to Topography: A Test of Topographic Indices as Moisture Indicators.” *Agricultural and Forest Meteorology* 98–99: 325–340. doi:10.1016/S0168-1923(99)00104-5.

Saunders, W. K., and D. R. Maidment. 1996. “A GIS Assessment of Nonpoint Source Pollution in the San Antonio Nueces Coastal Basin.” *Center for Research in Water Resources, The University of Texas at Austin*.

Schwanghart, W., G. Groom, N. J. Kuhn, and G. Heckrath. 2013. “Flow Network Derivation from a High Resolution DEM in a Low Relief, Agrarian Landscape.” *Earth Surface Processes and Landforms* 38: 1576–1586. doi:10.1002/esp.3452.

Soille, P. 2004a. “Morphological Carving.” *Pattern Recognition Letters* 25: 543–550. doi:10.1016/j.patrec.2003.12.007.

Soille, P. 2004b. “Optimal Removal of Spurious Pits in Grid Digital Elevation Models.” *Water Resources Research* 40: 229–244. doi:10.1029/2004WR003060.

Soille, P., and C. Gratin. 1994. “An Efficient Algorithm for Drainage Network Extraction on DEMs.” *Journal of Visual Communication and Image Representation* 5: 181–189. doi:10.1006/jvci.1994.1017.

Soille, P., J. Vogt, and R. Colombo. 2003. “Carving and Adaptive Drainage Enforcement of Grid Digital Elevation Models.” *Water Resources Research* 39: 1366–1378. doi:10.1029/2002WR001879.

Su, C., X. Wang, C. Feng, Z. Huang, and X. Zhang. 2015. “An Integrated Algorithm for Depression Filling and Assignment of Drainage Directions over Flat Surfaces in Digital Elevation Models.” *Earth Science Informatics* 8: 895–905. doi:10.1007/s12145-015-0223-5.

Tarboton, D. G. 1997. “A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models.” *Water Resources Research* 33: 309–319. doi:10.1029/96WR03137.

Tesa, T. K., D. G. Tarboton, D. W. Watson, K. A. Schreuders, M. E. Baker, and R. M. Wallace. 2011. “Extraction of Hydrological Proximity Measures from DEMs Using Parallel Processing.” *Environmental Modelling & Software* 26: 1696–1709. doi:10.1016/j.envsoft.2011.07.018.

Tribe, A. 1992. “Automated Recognition of Valley Lines and Drainage Networks from Grid Digital Elevation Models: A Review and a New Method.” *Journal of Hydrology* 139: 263–293. doi:10.1016/0022-1694(92)90206-B.

Turcotte, R. J. P., Fortin, A. N. Rousseau, S. Massicotte, and J. P. Villeneuve. 2001. “Determination of the Drainage Structure of a Watershed Using a Digital Elevation Model and a Digital River and Lake Network.” *Journal of Hydrology* 240: 225–242. doi:10.1016/S0022-1694(00)00342-5.

USACE (US Army Corps of Engineers). 2002. *Geospatial Hydrologic Modeling Extension HEC-GeoHMS User’s Manual Version 1.0*. Davis, CA: USACE.

Vincent, L., and P. Soille. 1991. “Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations.” *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13: 583–598. doi:10.1109/34.87344.

Wallis, C., R. Wallace, D. G. Tarboton, D. W. Watson, K. A. T. Schreuders, and T. K. Tesa. 2009. “Hydrologic Terrain Processing Using Parallel Computing.” In 18th *World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation* 2540–2545. [http://www.mssanz.org.au/modsim09/F12/kragt.pdf](http://www.mssanz.org.au/modsim09/F12/kragt.pdf).

Wang, L., and H. Liu. 2006. “An Efficient Method for Identifying and Filling Surface Depressions in Digital Elevation Models for Hydrologic Analysis and Modelling.” *International Journal of Geographical Information Science* 20: 193–213. doi:10.1080/13658810500433453.

Wang, S., and M. P. Armstrong. 2009. “Atheoretical Approach to the Use of Cyberinfrastructure in Geographical Analysis.” *International Journal of Geographical Information Science* 23: 169–193. doi:10.1080/13658810801918509.

Wilson, J. P. 2012. “Digital Terrain Modeling.” *Geomorphology* 137: 107–121. doi:10.1016/j.geomorph.2011.03.012.

Wilson, J. P., eds. 2018. *Environmental Applications of Digital Terrain Modeling*. Hoboken, NJ: Wiley Blackwell.

Wu, Q., and C. R. Lane. 2017. “Delineating Wetland Catchments and Modeling Hydrologic Connectivity Using Lidar Data and Aerial Imagery.” *Hydrology and Earth System Sciences* 21: 3579–3595. doi:10.5194/hess-21-3579-2017.

Wu, Q., H. Liu, S. Wang, B. Yu, R. Beck, and K. Hinkel. 2015. “A Localized Contour Tree Method for Deriving
Geometric and Topological Properties of Complex Surface Depressions Based on High-Resolution Topographical Data.” *International Journal of Geographical Information Science* 29: 2041–2060. doi:10.1080/13658816.2015.1038719.

Yamazaki, D., C. A. Baugh, P. D. Bates, S. Kanae, D. E. Alsdorf, and T. Oki. 2012. “Adjustment of a Spaceborne DEM for Use in Floodplain Hydrodynamic Modeling.” *Journal of Hydrology* 436: 81–91. doi:10.1016/j.jhydrol.2012.02.045.

Yamazaki, D., D. Ikeshima, R. Tawatari, T. Yamaguchi, F. O’Loughlin, J. C. Neal, C. C. Sampson, S. Kanae, and P. D. Bates. 2017. “A High-Accuracy Map of Global Terrain Elevations.” *Geophysical Research Letters* 44: 5844–5853. doi:10.1002/2017GL072874.

Yue, L., H. Shen, L. Zhang, X. Zheng, F. Zhang, and Q. Yuan. 2017. “High-Quality Seamless DEM Generation Blending SRTM-1, ASTER GDEM V2 and ICESat/GLAS Observations.” *ISPRS Journal of Photogrammetry and Remote Sensing* 123: 20–34. doi:10.1016/j.isprsjprs.2016.11.002.

Zhou, G., X. Liu, S. Fu, and Z. Sun. 2017. “Parallel Identification and Filling of Depressions in Raster Digital Elevation Models.” *International Journal of Geographical Information Science* 31: 1061–1078. doi:10.1080/13658816.2016.1262954.

Zhou, G., Z. Sun, and S. Fu. 2016. “An Efficient Variant of the Priority-Flood Algorithm for Filling Depressions in Raster Digital Elevation Models.” *Computers & Geosciences* 90: 87–96. doi:10.1016/j.cageo.2016.02.021.

Zhu, Q., Y. Tian, and J. Zhao. 2006. “An Efficient Depression Processing Algorithm for Hydrologic Analysis.” *Computers & Geosciences* 32: 615–623. doi:10.1016/j.cageo.2005.09.001.