Computers & Geosciences 72 (2014) 156–166

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

A hierarchical network-based algorithm for multi-scale watershed delineation

Anthony M. Castronova [a,*], Jonathan L. Goodall [b,c]

[a] Department of Civil and Environmental Engineering, Utah State University, 8200 Old Main, Locan, UT 84322, USA
[b] Department of Civil and Environmental Engineering, University of Virginia, 351 McCormick Rd., Charlottesville, VA, USA
[c] Department of Civil and Environmental Engineering, University of South Carolina, 300 Main Street, Columbia, SC 29208, USA

ARTICLE INFO

Article history:
Received 30 January 2014
Received in revised form
2 June 2014
Accepted 18 July 2014
Available online 29 July 2014

Keywords:
Geographic information systems
Geographic information science
Terrain analysis
Hydrologic analysis
Spatial analysis

ABSTRACT

Watershed delineation is a process for defining a land area that contributes surface water flow to a single outlet point. It is a commonly used in water resources analysis to define the domain in which hydrologic process calculations are applied. There has been a growing effort over the past decade to improve surface elevation measurements in the U.S., which has had a significant impact on the accuracy of hydrologic calculations. Traditional watershed processing on these elevation rasters, however, becomes more burdensome as data resolution increases. As a result, processing of these datasets can be troublesome on standard desktop computers. This challenge has resulted in numerous works that aim to provide high performance computing solutions to large data, high resolution data, or both. This work proposes an efficient watershed delineation algorithm for use in desktop computing environments that leverages existing data, U.S. Geological Survey (USGS) National Hydrography Dataset Plus (NHD+), and open source software tools to construct watershed boundaries. This approach makes use of U.S. national-level hydrography data that has been precomputed using raster processing algorithms coupled with quality control routines. Our approach uses carefully arranged data and mathematical graph theory to traverse river networks and identify catchment boundaries. We demonstrate this new watershed delineation technique, compare its accuracy with traditional algorithms that derive watershed solely from digital elevation models, and then extend our approach to address subwatershed delineation. Our findings suggest that the open-source hierarchical network-based delineation procedure presented in the work is a promising approach to watershed delineation that can be used summarize publicly available datasets for hydrologic model input pre-processing. Through our analysis, we explore the benefits of reusing the NHD+ datasets for watershed delineation, and find that the our technique offers greater flexibility and extendability than traditional raster algorithms.

1. Introduction

A watershed boundary defines the land surface that contributes streamflow to a single outlet location (Chow et al., 1988). With advancements in geospatial software and readily available remotely sensed data, geographic information system (GIS) analysis has become widely used by hydrologists for determining a watershed boundary. Many research studies have investigated the various terrain processing components of GIS watershed delineation, such as methods for surface smoothing (Hutchinson, 1989), determination of flow direction (Douglas, 1986), slope and aspect calculations (Hodgson, 1998), depression filling (Jenson and Trautwein, 1987), and the extraction of drainage channels (O'Callaghan and Mark, 1984). These are only a few examples of the research that helped shape this domain; Moore et al. (2006) offer a more complete summary of the field.

The advent of high resolution digital terrain data and the need to analyze larger watersheds for environmental policy have resulted in efforts to advance the computational efficiency of terrain processing for hydrology applications. Recent studies have employed high performance computing (HPC) environments to overcome such computational limitations (Mineter, 2003; Wang and Armstrong, 2009; Huang et al., 2011). Through these studies it has been demonstrated that HPC solutions have the potential for large performance gains by uncovering the intrinsic parallelism in traditional geospatial algorithms (e.g. Wang and Armstrong, 2009). Parallel algorithms operate by sharing the computational burden

* Corresponding author.
E-mail addresses: tony.castronova@usu.edu (A.M. Castronova), goodall@virginia.edu (J.L. Goodall).

http://dx.doi.org/10.1016/j.cageo.2014.07.014
0098-3004/

of data processing with multiple resources, and communicating data among each other using protocols such as the Message Passing Interface (MPI) (Xie, 2012). These approaches use advanced computational algorithms for delineating watersheds from digital elevation models (DEMs), mostly using the divide and conquer approach (Hutchinson et al., 1996).

A similar, albeit fundamentally different approach for processing large datasets, is to leverage idle computing power by means of high throughput computing (HTC). HTC is a method for flexible distributed computing that takes advantage of relatively inexpensive collections of computing resources to achieve performance gains comparable to large HPCs (Thain et al., 2005). It is a convenient solution for processing large amounts of data that enables organizations to take advantage of existing network compute power without the need for special computer hardware. The goal is to achieve speedup over longer periods of time using computing grids rather than emphasizing computer architecture (Chaudhry et al., 2005). Recent studies have shown that this approach is effective in achieving significant computational speedup when processing large raster datasets (Gong and Xie, 2009; Huang and Yang, 2011).

While these approaches have been used extensively to process large datasets, they require access to advanced computing techniques and resources. For instance, a great deal of expertise is required to design and use parallel HPC software modules because of their inherently high "learning curve," which has a tendency to deter both commercial and academic developers (Mineter, 2000; Lu et al., 2010). An exception to this is software that have adapted their algorithms to distribute computational load among processor threads to incorporate some of the HPC advantages (i.s. distributed computing) on desktop computers. TauDEM is one software application that employs this tactic to provide users with the best of both worlds (Wallis et al., 2009). Similarly, HTC requires a large network of idle computers as well as specialized scheduling software to balance computing load across the network. Overall HPC and HTC solutions can be effective for data intensive computations, however they require specific computer hardware and a high level of sophistication. Moreover, many water resources professionals still rely on desktop computing environments as their main platform for watershed analysis. We lack a versatile approach of watershed delineation capable of efficiently resolving a wide range of spatial scales, without the use of HPC, HTC, or similar computing environments.

An alternative strategy for watershed delineation is to rely on pre-processed vector data. One example of this approach was presented by Djokic and Ye (1999), which aimed to overcome the computationally intensive nature of watershed delineation by separating static terrain-based properties from the delineation procedure. They proposed that since terrain measurements do not change often, they should not be linked directly to the delineation procedure. Rather, catchment geometries are processed prior to watershed delineation and later leveraged to construct a watershed boundaries. The major contribution of this work was their methodology, Fast Watershed Delineation (FWD), which is capable of rapidly yielding watershed boundaries using only desktop computing resources. Several additional efforts have been made to extend this technique for serving watershed delineations via web services. For example, the ArcGIS Watershed Delineation service provides a quick method for retrieving watershed delineations (Kopp, 2013). Both of these approaches, however, require that computationally intensive catchment pre-processing routines have been completed prior to usage. Similar web based efforts have been made by the Environmental Protection Agency (EPA) and United States Geologic Survey (USGS) to produce the Navigation Delineation Service and StreamStats, respectively. The EPA Navigation Delineation Service leverages the NHD+ dataset to determine watershed boundaries and has been implemented by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) HydroDesktop software, to delineate watershed boundaries which are then used to search for observation data within the Hydrologic Information System (Ames et al., 2012). Similarly, the USGS StreamStats application offers a delineation service that is built using the NHD+ dataset and ArcGIS tools, but it also requires significant pre-processing (Guthrie et al., 2009; Ries et al., 2009).

Since the work of Djokic and Ye (1999), new datasets have become available such as the USGS National Hydrography Dataset Plus (NHD+). The NHD+ is a dataset derived from measured elevation, digitized hydrography, and the USGS Watershed Boundary Dataset (WBD) to accurately match known surface hydrology. While the NHD+ contains elevation derived products such as flow direction and flow accumulation grids for the entire U.S., it also provides pre-processed hydrologic catchment boundaries and river flow networks. These can be leveraged to rapidly delineate watershed boundaries while eliminating data intensive pre-processing routines. Our approach is to leverage the concepts outlined by Djokic and Ye (1999), and the pre-processed NHD+ data to reconstruct watershed boundaries from pre-computed catchment geometries.. Using graphing algorithms, upstream flow direction cells and ultimately catchment boundaries are identified for a given outlet location. We demonstrate how our approach is capable of rapidly yielding watershed boundaries for large areas on a desktop computer, while also delineating small catchments in a timely manner. It is then applied to the delineation of subwatersheds to demonstrate how it can be adapted for other common hydrologic tasks. Overall we demonstrate how our approach is a versatile solution for performing multi-scale watershed delineation on a desktop computer.

2. Method

Our method for watershed delineation is a two-step approach that borrows from graph theory to transform river flow attributes and known watershed surface runoff patterns into relational networks. While hydraulic river flows are used to identify fluxes between catchments, surface runoff is used to establish flow paths between raster cells. Furthermore, the hydraulic river flow graph is used to determine the "upper" portion of the watershed, and in contrast the surface runoff graph is used to determine the "lower" portion of the watershed. These upper and lower geometries are later combined to form the complete watershed boundary. This delineation technique requires both reach network and catchment input shapefiles, and relies on the specific relationships that can be established between them. More specifically, each feature in the reach network must have defined start and end nodes which are associated with other reaches in the network, as well as attributes that support network traversal. In addition, each reach must be associated with a single catchment geometry. This section explains how the NHD+ dataset can be leveraged to rapidly delineate watersheds, however as long as the aforementioned constraints are met the same technique can be applied to other datasets.

Our method uses graph theory to form relationships between the hydraulic characteristics of reaches as well as gravity driven flows among terrain grid cells. The basic concept is that a graph consist of independent entities (i.e. vertices) that are related to one another through connections (i.e. edges) (Marcus, 2008). This basic concept is illustrated in Fig. 1. Once a graph is assembled, algorithms are used to navigate and traverse the nodes to solve a variety of problems, such as the famous traveling salesman problem (Hagberg and Schult, 2008). There are also many different types of graphs, such as undirected, directed and multigraphs, which can be applied to solve problems in nearly every discipline