This is an Accepted Manuscript of an article published in the Journal of Computers & Geosciences in 2014, available online: doi:10.1016/j.cageo.2014.07.014

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

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

[a]*Research Assistant Professor, Department of Civil and Environmental Engineering, Utah State University, 8200 Old Main, Locan, Utah 84322*

[b]*Associate Professor, Department of Civil and Environmental Engineering, University of Virginia, 351 McCormick Rd., Charlottesville, VA USA*

[c]*Adjunct Professor, Department of Civil and Environmental Engineering, University of South Carolina, 300 Main Street, Columbia, SC, 29208*

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.

Keywords: Geographic Information Systems; Geographic Information Science; Terrain Analysis; Hydrologic Analysis; Spatial Analysis

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 have 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) offers 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 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 processes large datasets, they require access to advanced computing techniques and resources. For instance, a great deal of expertise is required to design

*Corresponding author
Email addresses: tony.castronova@usu.edu (Anthony M. Castronova), goodall@virginia.edu (Jonathan L. Goodall)

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 *et al.*, 2000; Lu *et al.*, 2010). An exception to this is are 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 (2000), 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 *et al.*, 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 (2000), 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 (2000), 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 Figure 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 *et al.*, 2008). There are also many different types of graphs, such as undirected, directed, multigraphs, etc., which can be applied to solve problems in nearly every discipline (Rosen, 2003). In our work we use the NHD+ catchments to define graph vertices and the NHD+ reaches as graph edges between these nodes. The terrain is also represented as a graph, where the centroid of each cell in a DEM is a vertex and the flow

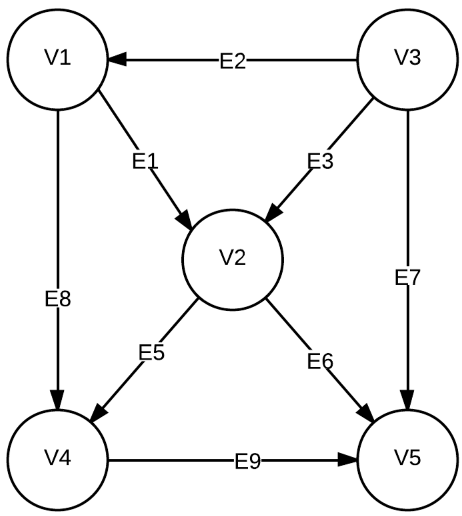

Figure 1: A basic directional graph consists of vertices that are connected by edges to describe the relationship among them.

direction at the node is used to establish an edge with its downstream neighbor. These two directed graphs enable us to traverse the NHD+ dataset in a hydraulically upstream/downstream manner. Both of these concepts are further explained in the following paragraphs.

First, consider that any given watershed may consist of one or more NHD+ catchments. Figure (2, i) shows the NHD+ river network overlaying a small watershed consisting of pre-delineated catchments. These catchments are related to one another by river flow attributes, for instance, each catchment drains into exactly one of its neighbors. Since, each river in the NHD+ dataset is associated with an upstream and downstream reach, this information can be used to create the graph illustrated in Figure (2, ii). This graph network defines the water flow paths among catchments. Using our approach, we assume that a watershed is encapsulated within this network and moreover consists of one or more catchments that can be identified by hydraulic river flow attributes. In this manner, all catchments upstream of a given outlet location can be quickly determined (Figure 2, iii), and subsequently merged together to resolve the geometry for the upper portion of the watershed (Figure 2, iv).

A watershed outlet can be located anywhere within a catchment, not necessarily coinciding with the drainage point used in the NHD+. Therefore, we must consider an alternative approach for resolving the remaining, lower portion, of the watershed. This is accomplished by leveraging watershed surface flow attributes. First, a mathematical graph is created using flow direction raster cell values. In a single-flow-direction raster grid, each pixel contains a numeric value that defines the direction water flows from the

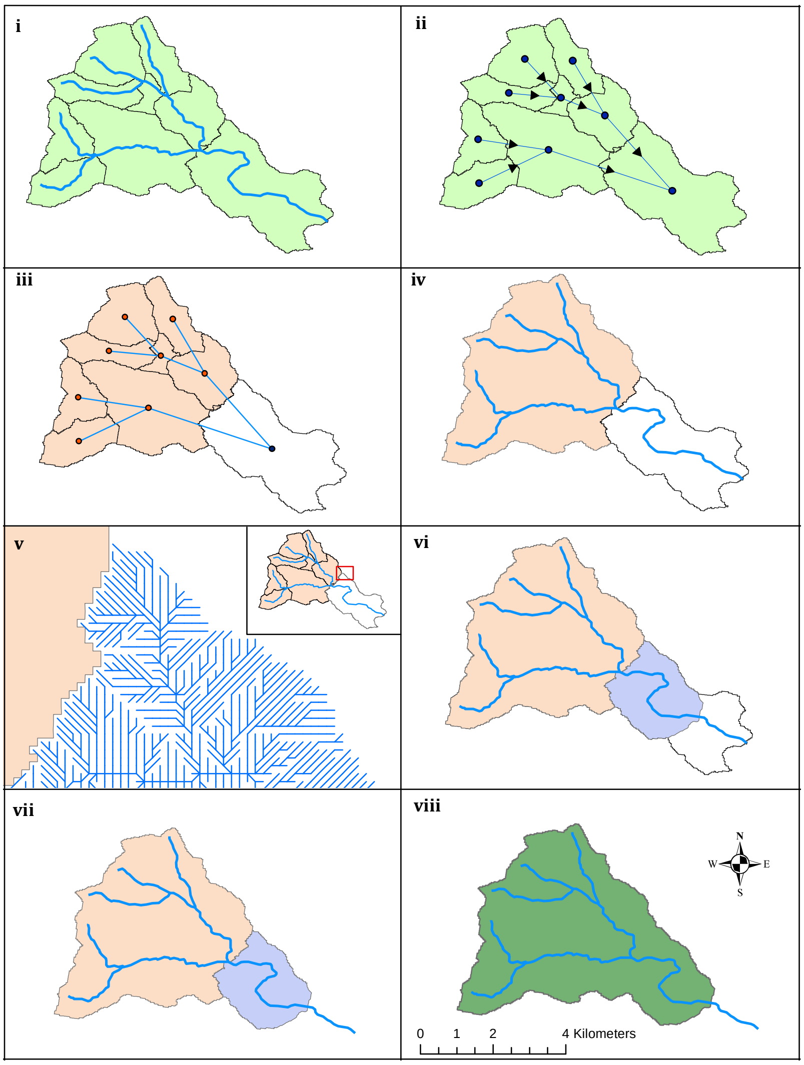

Figure 2: A graphical representation of the watershed delineation procedure.

surface in flooded conditions. By iterating through these cell values, a graph is constructed whereby each node represents the centroid of a cell and each cell is connected to its "downstream" neighbor. Figure 2, v) illustrates how an NHD+ catchment is transformed in to a dense graph network of cell-level flow paths. Using graph theory, all of the raster cells contributing to a given location, i.e. watershed outlet, can be determined by tracing the flow network in the upstream direction. This task is made trivial by leveraging well-established software libraries that employ optimized graph traversing algorithms. Once the upstream graph elements are known, the boundary for the lower geometry can be constructed by eliminating all interior graph nodes (Figure 2, vi). Once the upper and lower portions of the watershed have been resolved (Figure 2, vii), they are spatially merged together using GIS software to produce the complete watershed boundary (Figure 2, viii).

To construct the catchment and flow direction graphs, existing software libraries can be leveraged such as the NetworkX Python library (Hagberg *et al.*, 2008). Using the aforementioned approach for constructing flow direction graphs, an edge is created between each cell and its "downstream" neighbor. However, applying this methodology for large areas results in extremely large graphs, and is therefore infeasible. For instance, creating a graph network consisting of 1 arcsecond data covering South Carolina, results in approximately 8.88×10^7 graph elements. A graph of this magnitude is impractical because its memory footprint is too large for most desktop computers. Since this graph is only used to resolve the lower portion of the watershed, it must only cover the area of an NHD+ catchment. Given this, single flow direction rasters can be extracted individually for each NHD+ catchment and graphs can be subsequently created. Moreover, these catchment level graphs can be serialized and stored for later use to further eliminate redundant operations. This technique results in one graph for each NHD+ catchment boundary. In contrast, the catchment graph is much less dense and as a result a single graph will suffice for an area covering South Carolina. By using this graphical approach, (1) all upstream catchments are identified, (2) the raster cells contributing flow to the outlet can be determined, and finally (3) the results of these operations can be merged to form a complete watershed.

This method can also be extended to support the delineation of subwatersheds, which is an important feature of GIS software for hydrology applications. Subwatersheds are generally derived from outlets that correspond with available observation data, and are often used to pre-process or summarize watershed related data for model inputs. A similar approach is taken to determine these subwatershed areas, albeit with a small modification. First all upstream catchments are determined. Next, the catchments corresponding to each individual outlet are isolated, such that each catchment is associated with only one outlet location. For each outlet, the upper catchments and lower catchment are combined in a manner consistent with Figure 2.

3. Implementation

The NHD+ provides many GIS data products to the public for free. The watershed delineation technique presented in this work uses several of these data products, as well as supplementary database files used to enhance their geospatial representations. While a newer version of the NHD+ dataset (version 2) is currently available, this work was initiated and completed using the NHD+ version 1. These data provide additional feature-based values and attributes to support the NHD+ vector data. This supplemental information can be leveraged to establish relationships between features of multiple NHD+ vector files. Moreover, the delineation algorithm relies on the NHD+ vector products and these additional datasets to establish relationships between digitized rivers and catchment boundaries to rapidly delineate watershed boundaries.

There exists a many-to-one relationship between the NHD+ river and catchment features. This means that there is at least one reach for every catchment defined in the dataset. As a result, the NHD+ reaches

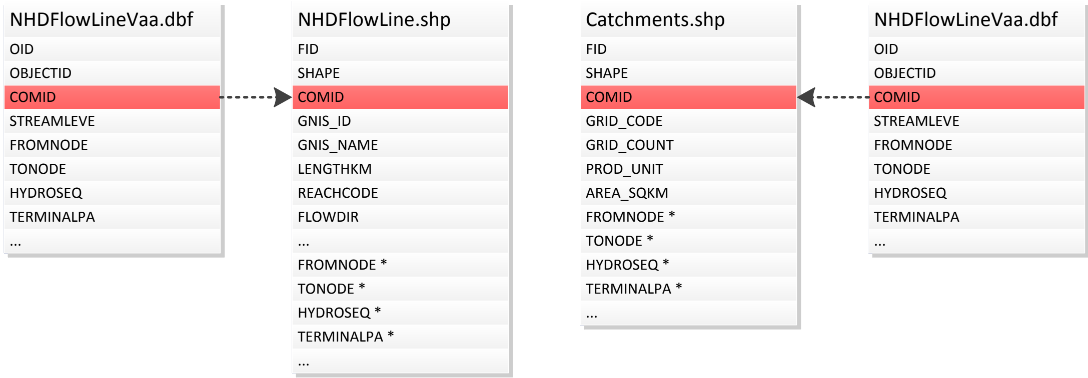

Figure 3: *Joining the NHDFlowLineVaa dataset to both the NHDFlowLines and Catchment shapefiles provides the necessary attributes to form a connection between river reaches and catchments.*

can be used to identify specific catchments, but to do this additional data attributes must be appended to the reach dataset. Figure 3 illustrates how the NHDFlowlineVAA.dbf (i.e. Value Added Attributes, VAA) data product can be used to enhance the nhdflowline and catchment vector files. By spatially joining the VAA attributes to both of these datasets, additional information is appended to each feature such as *fromnode*, *tonode*, *hydroseq*, and *terminalpa*. The *fromnode* and *tonode* attributes are unique identification numbers that denote the start and end nodes for every reach in the dataset. The *hydroseq* parameter is a unique hydrologic sequence number assigned to each reach in the dataset. These sequence identifiers are assigned such that upstream reaches have larger values than downstream reaches. Finally, the *terminalpa* parameter defines the hydrologic sequence number of the terminal feature of the reach network. This data is used to create a directed graph which can then be used to identify upstream or downstream reaches from any location within the network.

Using this upstream and downstream reach information, a graphical network is created to represent relationships between digitized reaches, and ultimately the catchment features. First, NHD+ reach and attribute data is transformed into a graphical network which later provides a mechanism for tracing flow path's and identifying upstream reaches id's using graph tracing algorithms. Figure 4 illustrates how this graph network is created using the Python programming language and the NetworkX library (Hagberg *et al.*, 2008). For each feature in the NHD+ reach network, a graph edge is established between its *fromnode* and *tonode* identifiers. In addition, *hydroseq* and *terminalpa* values are stored as attributes on the graph object. Once this process is complete for every reach feature, the graph is serialized for later use. Serialization is an important part of this procedure because it enables the graph data object to be reloaded when needed, rather than reconstructing it from scratch, which would be timely and inefficient. Using this graph, data for

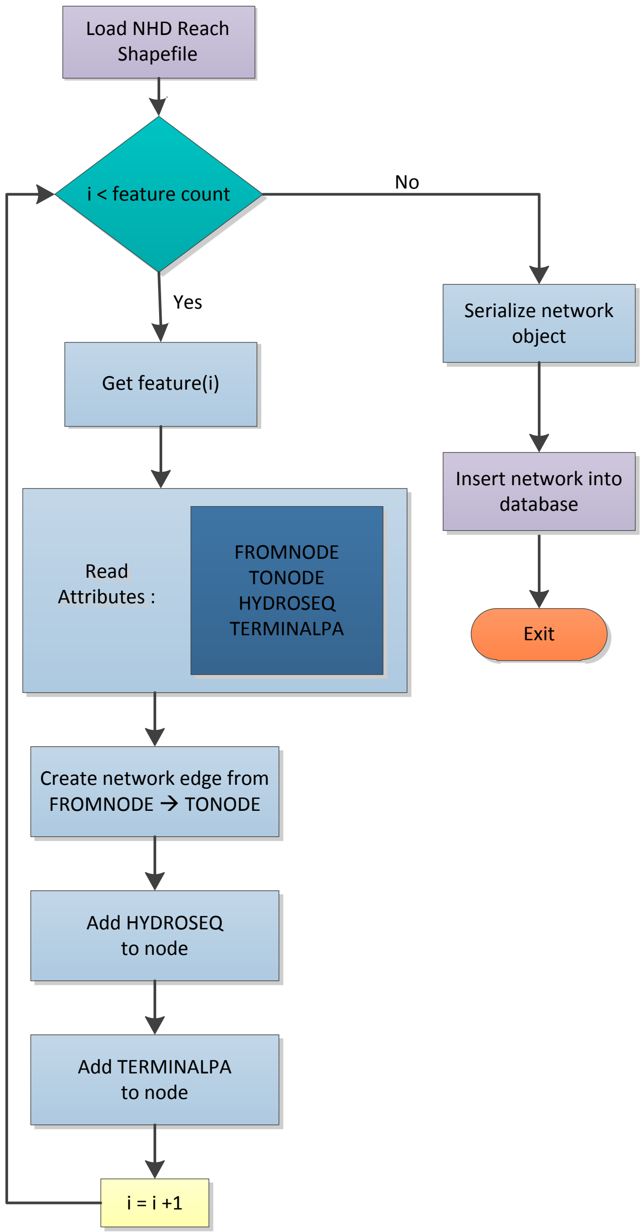

Figure 4: *Flow chart illustrating how NHD+ digitized reaches are transformed into a mathematical graph network that is then serialized for later use.*

all elements preceding a given node can be identified using NetworkX graph tracing functions. This provides an effective means for selecting the NHD+ reach elements that contribute flow to a common downstream location. For example, given a watershed outlet, all upstream reach attributes can be quickly identified. These data are then used to select the corresponding NHD+ catchment geometries.

Databases offer a mechanism for archiving large amounts of data in an easily accessible manner. Because of this, a database was chosen for storing the NHD+ catchment geometries and feature attributes. For this implementation the open source PostgreSQL database was chosen because it can easily be extended to support spatial data queries using PostGIS. This setup enables spatial data to be archived in an easily accessible format as well as retrieved using spatial data querying using standard SQL statements. Moreover, the PostGIS extension is equipped with numerous spatial operations that can be performed on-the-fly when querying data. Therefore, the NHD+ catchment geometry data was loaded into a PostgreSQL database along with the appended NHD+ value added attributes. In addition, an empty *network* database field was defined to store cell-level NetworkX graphs which are constructed on-the-fly using flow direction values within the selected catchment boundary. For instance, when an outlet is chosen and the corresponding catchment is identified, we must first check to see if a cell-level graph network exists in the database for this catchment. If not, it is created on-the-fly and saved in the database for later use. This design consideration aims to reduce unnecessary pre-processing steps, however this calculation can be performed ahead of time if maximum speed is a priority (e.g. web deployment). Once these catchment attributes are loaded into the database, spatial SQL queries are used to quickly extract catchment geometries when needed.

With the pre-processed NHD+ catchment data stored in a spatial database, features are queried using standard and spatial SQL statements. For instance, all catchments "upstream" of a graph location have a *hydroseq* identifier greater than that of the current location yet no greater than the largest *hydroseq* in the graph tree. The upper limit of this range is quantified as *maxseq*, and is calculated by simply iterating through all "upstream" nodes in the graph. Furthermore, all features must also belong to the same *terminalpa*. Therefore, given an outlet location as *outlet_node*, all upstream catchments can be identified in a manner consistent with Figure 5. This results in the extraction of all NHD+ catchment geometries upstream of the outlet. These geometries are deserialized and subsequently merged together by performing a spatial union, which results in the upper portion of the watershed boundary. Since the watershed outlet can be located anywhere within the lowest catchment of the watershed, an alternative approach must be used to resolve the lower portion of the watershed boundary.

Since NHD+ catchment areas are relatively small in scale, mathematical graphs can be created to represent the flow paths between the interior flow direction values for each catchment. To create one of these networks, the NHD+ flow direction grid is first extracted over the area of a single catchment geometry. For each cell within this smaller grid, graph edges are created between each cell and its "downstream" neighbor. The "downstream" neighbor is easily identified for each cell using the single flow direction notation (i.e. D8

```sql
SELECT Geometry
FROM catchments_datatable
WHERE HYDROSEQ > outlet_node.HYDROSEQ
   AND HYDROSEQ <= MaxSEQ
   AND TERMINALPA == outlet_node.TERMINALPA
```

Figure 5: *SQL query for extracting catchment geometries stored in a PostreSQL database using HYDROSEQ and TERMINALPA attributes.*

grid values). The final product is a graphical network consisting of edges that define the flow paths between raster cells, which are confined to a single NHD+ catchment boundary (i.e. catchment flow path graph). As mentioned earlier, this operation is performed on-the-fly when needed, and stored in the spatial database within the *network* field as a serialized NetworkX object.

Using this flow direction graph, all nodes (i.e. cells) upstream of the outlet can be quickly identified using graph traversing algorithms. The result is a set of coordinates that represent all cell locations "upstream" of the outlet, but within the "most downstream" NHD+ catchment. By tracing the edge of this delineation upstream from the outlet, the border locations can be identified. Once this boundary is identified, a polygon object is created that represents the lower portion of the watershed. Finally, the upper and lower watershed polygons are combined to form a single watershed boundary. Once complete, the overall catchment boundary is saved as Well Known Text (WKT) and are later converted into Esri shapefile format for visualization.

4. Application

Two studies were conducted to evaluate the application of the provided watershed delineation technique. First it is evaluated in its ability to delineate watersheds at various spatial scales, then it is applied to the delineation of subwatersheds. While similar Three community accepted software applications are used to provide context for the general accuracy of the hierarchical algorithm. The first benchmark software, Esri's ArcGIS, is a widely used commercial-grade GIS suite. It consists of many tools for GIS analysis, including a hydrology toolbox which is capable of performing a wide range of hydrology-related data processing routines. In addition, ArcGIS contains a built-in Python interpreter, which enables these tools to be executed programmatically. This functionality was leveraged to automate the ArcGIS watershed delineation procedure, which consisted of executing several tools in series.

The second benchmark software, Terrain Analysis Using Digital Elevation Models (TauDEM), is an open source terrain analysis project (Tarboton *et al.*, 1997). It employs parallel computing concepts to divide large datasets into smaller subsets. Terrain processing is performed on each of these subsets simultaneously, and messages are passed between computational processes when necessary. Because of this design consideration, it can theoretically perform terrain processing on very large datasets at a much faster rate than other software. It has been used in a number of academic studies from basic watershed analysis (Tarboton *et al.*, 1997) to parallel terrain computations (Wallis *et al.*, 2009). Furthermore, the latest release of TauDEM (version 5.1) contains a toolbox plugin for ArcGIS (versions 9.3.1 – 10.0), therefore in a similar manner to ArcGIS, TauDEM tools can be executed autonomously through Python scripting. Our motivation for using these specific GIS software tools for comparison to our approach is to first provide context with regards to a closed-source commercially developed product, and then to an open-source academic tool.

The third benchmark software is ArcHydro Tools, which is a spatial processing tool pack that automates ArcGIS tools to perform advanced hydrology-related functions. In summary, it is an advanced information system that integrates the spatial and temporal aspects of hydrology to provide a complete GIS modeling and analysis suite (Maidment *ed.*, 2002). ArcHydro Tools is used in this work for its capacity to evaluate the accuracy and general efficiency of the provided delineation technique when applied to subwatershed delineation.

The testing all watershed delineations was performed using 32-bit Python interpreter on a desktop computer with a quad core 2.80 GHz processor having 4 GB of memory. The execution of the provided hierarchical delineation procedure is done by simply invoking the package from the commandline and passing it either one or multiple outlet point coordinates. The user must be careful to supply outlets in the spatial reference system that is used by the NHD+ catchment and river reach data. The algorithm uses several scientific libraries such as NumPY, GDAL, and NetworkX which must be installed separately. In addition, the NHD+ data must be downloaded, specifically the *catchments*, *river reaches*, and *supplementary* data files. Lastly, for this work data is stored in a PostgreSQL database using the PostGIS extension for handling spatial attributes.

4.1. Multi-Scale Watershed Delineation

To evaluate our approach for various watershed scales, five basins were delineated along the South Carolina, Georgia border (i.e. the Savannah River basin) as well as one watershed that includes part of North Carolina (i.e. the Cooper River basin), shown in Figure 6. All watershed delineations were performed using input data provided by the NHD+ (i.e. flow accumulation and D8 flow direction), and the size of the datasets used in each scenario are summarized in Table 1. While the input data for each delineation scenario were provided at the same resolution, the grid sizes increased ascendingly in order to evaluate a wide range of computational scales. The experiments began with smaller headwaters and progressed to the larger downstream areas, concluding with the Savannah and Cooper River basins.

Table 1: *General properties of the input raster datasets used for each watershed delineation scenario.*

Dataset	Resolution	Dimensions	Grid Size	Disk Size
Scenario 1	30 m	926 × 891	8.3×10^5	3.15 MB
Scenario 2	30 m	1835 × 2153	39.5×10^5	15.07 MB
Scenario 3	30 m	5822 × 5848	340.5×10^5	129.88 MB
Scenario 4	30 m	8287 × 7792	645.7×10^5	246.32 MB
Scenario 5	30 m	10165 × 11011	1119.3×10^5	426.97 MB

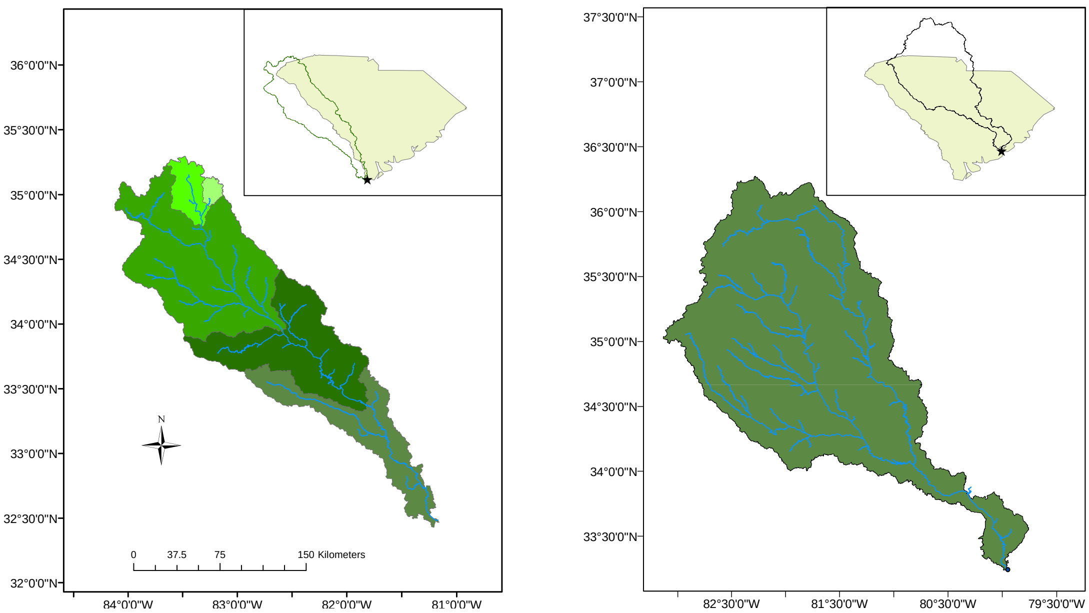

Figure 6: *Six watersheds were used to evaluate the performance of the hierarchical delineation approach. The Savannah River basin (Left) which was divided into five separate subwatersheds, and the Cooper River watershed (Right).*

The watershed delineation procedure using the ArcGIS software suite is illustrated in Figure 7. First, the processing environment is prepared for executing ArcGIS tools. This consists of loading Python modules as well as registering ArcGIS extensions. Once the environment has been prepared, the input raster grids (i.e. flow direction and flow accumulation) are reduced to the extent of the known watershed boundary using the ArcGIS Clip function. In practice this extent is often unknown, but since this experiment is evaluating the speed of ArcGIS watershed delineation, we assume that ideal input information is available. Next, a new point shapefile is created that contains a single feature, the watershed outlet location. This outlet point is then relocated to the neighboring raster pixel that has the highest flow accumulation value, using the Snap Pour Point tool. This is done to ensure that the outlet resides at a location of high flow accumulation, as determined by the terrain topography. Once these steps are complete, the Watershed tool is executed. Finally, the raster output from the watershed delineation is converted to a Esri polygon shapefile.

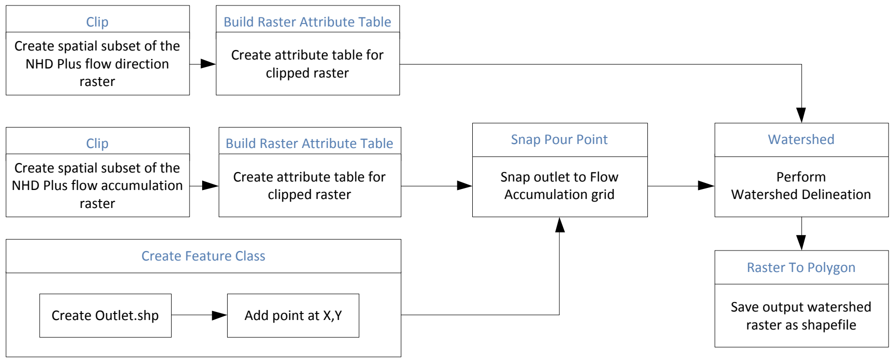

Figure 7: *The procedure used for watershed delineation using ArcGIS tools, automated using the Python programming language.*

TauDEM watershed delineation requires similar pre-processing routines to the ArcGIS approach. Many of the data pre-processing steps use standard ArcGIS tools, for instance *Clip* and *Reclassify*, whereas the actual watershed delineation uses only TauDEM tools which are indicated by the blue bold headings in Figure 8. While ArcMap was used to prepare the input data for this watershed delineation, alternative software could also be used for this task without affecting the results. Figure 8 follows the delineation procedure that is suggested by Tarboton (2010). First, a new point feature is created containing the desired outlet location of the watershed. Next, the NHD+ flow accumulation, elevation, and flow direction grids are clipped to the approximate extent of the watershed. While this step is optional, it can have a significant effect on the overall speed of the delineation by reducing the size of the input data. Map algebra is used to select flow accumulation cells based on a user specified threshold which creates a new raster grid in which rivers have a value of 1 and all other cells have a value of 0. This new river grid is then used to snap the outlet onto the river network using the TauDEM *Move Outlets To Streams* tool. This aligns the desired outlet with the watershed outlet as defined by the terrain. Next, the clipped flow direction grid is converted from the tradition single-flow-direction notation, into the single-flow-direction values used by TauDEM (i.e. $\{1,2,3,4,5,6,7,8\}$). The *Peuker Douglas Stream Definition* tool is executed using the snapped outlet, and the clipped flow accumulation, flow direction, and elevation grids as input. The tool creates several new output datasets that summarize various reach properties. One in particular, the stream raster grid, identifies all of the reaches upstream of the outlet location. This grid is used as input to the *Stream Reach And Watershed* tool, along with the snapped outlet, and clipped flow direction, flow accumulation, and elevation grids. Execution of this tool results in several more raster grids such as stream order, stream connectivity, stream coordinates, and a watershed grid. The watershed grid is supplied as input into the *Watershed Grid To shapefile* tool which converts it into a vector file, thus completing the delineation procedure.

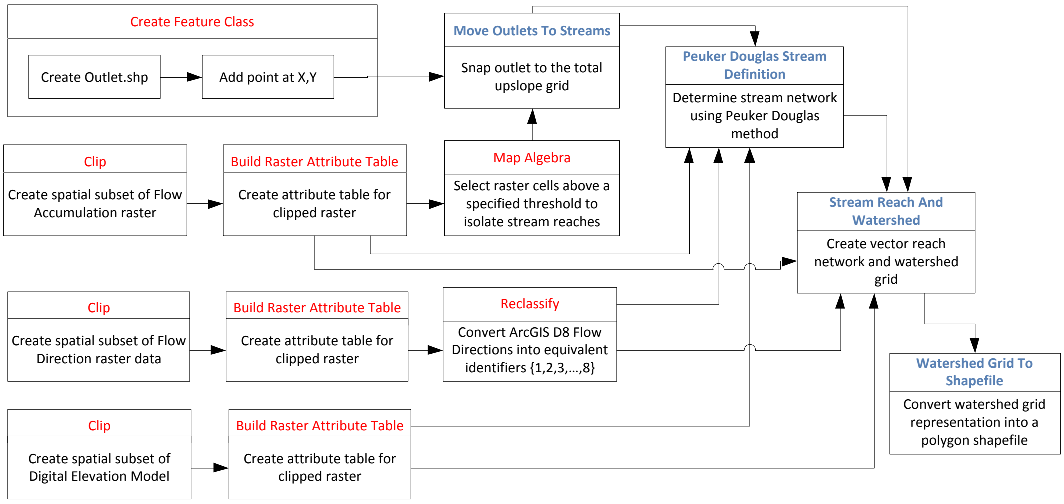

Figure 8: *The procedure for watershed delineation using TauDEM tools, implemented using the Python programming language. The blue bold titles indicate operations performed by ArcGIS and bold titles indicate operations performed by TauDEM.*

The provided approach and the two watershed processing techniques described above, were used to delineate six different watersheds. The objective of this study is to first evaluate the accuracy of hierarchical watershed delineation approach, and second to quantify its performance. Using the ArcGIS and TauDEM processing routines as benchmarks, we evaluate how well our approach performs with respect to commercial product and large scale terrain processing software. This experiment provides insight into the general application of our watershed delineation approach compared with two widely used software suites. It revealed that there exist differences in the watershed boundaries computed by each software suite. Table 2 illustrates these discrepancies for each scenario as the percent difference of the area taken with respect to the average computed watershed area. This serves as a basis of reference for comparing the variations in the watersheds computed by each algorithm. Minor boundary differences exist between the TauDEM and ArcGIS calculations, however these were found to be a result of polygon simplification. In contrast, the provided delineation exhibits larger variations with respect to the average computed watershed areas. These discrepancies may be explained by the manner in which the NHD+ dataset is created. For instance, the NHD+ is constructed using modified DEMs that closely match the known digitized hydrography as well as the WBD. The catchment features used in our approach were modified to match river streamflow and velocity estimates (Johnston *et al.*, 2009). In contrast, ArcGIS and TauDEM used surface elevation measurements which were not modified in the same manner. We suggest that the boundary differences outlined in Table 2 are a byproduct of the NHD+ design and are therefore inherent to our technique.

Dataset	Average Area km^2	ArcGIS % difference	TauDEM % difference	Hierarchical % difference
Scenario 1	336.893	0.543	0.538	1.081
Scenario 2	1716.599	0.124	0.123	0.248
Scenario 3	13841.155	0.029	0.029	0.058
Scenario 4	21581.296	0.018	0.019	0.037
Scenario 5	26774.317	0.018	0.018	0.037

Table 2: *Differences in the watershed areas computed by each software suite. Variations are recorded relative to the mean watershed size for each scenario.*

To provide context for the efficiency of our approach we compared the overall computation time for each of the delineation approaches. In this analysis we only interested in the time it takes for a user to perform a delineation from start to finish. We found that ArcGIS performs exceptionally well at small scales, however it follows a non-linear trend as the size of the dataset continues to grow. This is expected because the ArcGIS tools that are used in this study are designed for general purpose desktop computing and are not designed for processing very large data sets . In contrast, TauDEM which has the capability of processing large raster data, executed the fastest for all watersheds delineations except the largest. Similar to ArcGIS, it also scales in a non-linear fashion as dataset size increases. However, this work was all performed on a computer using 4 processing threads and 4GB of memory. We expect that TauDEM will perform more favorably on a high performance computer. Our technique is slower when delineating watersheds at small scales, however it follows a linear trend as area increases, and completes the largest delineations significantly faster than the other software systems. For instance, it finished approximately 2.1 times faster than ArcGIS and 1.7 times faster than TauDEM for the Savannah River basin, and approximately 2.5 times faster than ArcGIS and 2.7 times faster than TauDEM for the Cooper River basin. We believe that this speedup is because our approach is able to leverage pre-processed catchments rather than directly processing DEMs. This supports our argument that using pre-processed national datasets, such as the NHD+, has some distinct advantages over DEM processing approaches. However, as the number of catchments increases (i.e. dataset size increases), the amount of time dedicated to geometry extraction from the database and geometry merging, becomes more pronounced. This insight suggests that future work should focus on the internal algorithms used in our approach. These basic benchmarks are an average of several simulation runs in which raster pre-processing routines have been omitted. Therefore, we can expect the end-to-end execution times to increase with inclusion of raster data preparation, whereas this will not effect the our approach.

4.2. Subwatershed Delineation

Another important feature of GIS software for hydrology applications is the ability to delineate subwatershed areas. Relatively little work is required to extend our approach to subwatershed delineation, which further demonstrates its flexibility. The same general methodology is applied to determine watershed areas, therefore only a small extension required to provide the additional functionality of deriving subwatershed areas from NHD+ catchments.

The extension is largely made to perform the procedure outlined in Section 3 over a list of outlet points rather than a single location. This is illustrated by the loop mechanism in Figure 9, in which the geometry and graph networks are queried for each outlet provided by the user. The upper and lower watershed areas are determined for each outlet by following the methodology presented in Section 3. However, once all the upper catchments have been identified for all outlets, they are isolated such that each catchment geometry can only belong to one subwatershed, which in turn, is associated with one outlet. This is done to eliminate redundant merging of catchment geometries. This can also be done in a less efficient manner by first merging the upper and lower geometries for each outlet and then subtracting them from one and other to derive the subwatersheds. The subwatershed boundaries are used to create a polygon output file, and the outlet points are used to create a point output file. Together, these new files summarize the subwatershed topographic characteristics of a particular region.

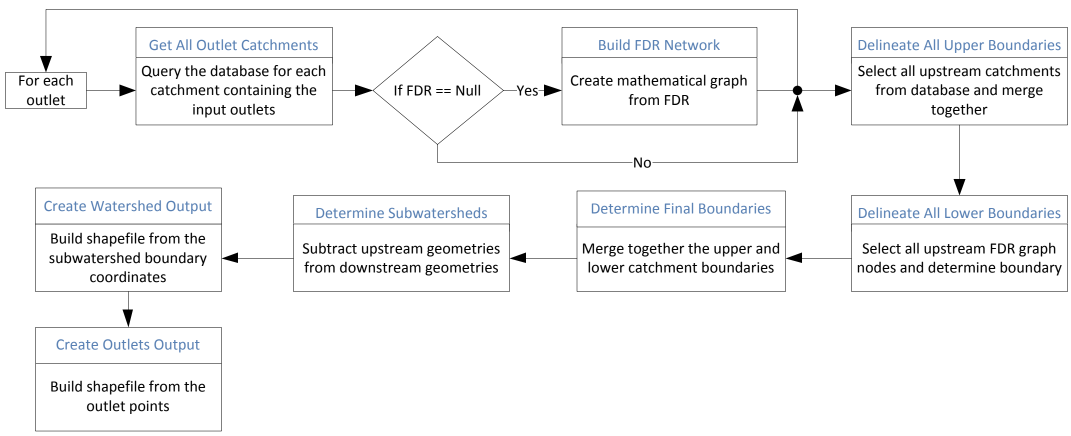

Figure 9: *The procedure for subwatershed delineation using the hierarchical technique.*

For comparison purposes, a widely used software suite is used to calculate the subwatersheds over the same area: Arc Hydro Tools. While Arc Hydro Tools is capable of performing a myriad of advanced hydrology-related processing, we only leverage its subwatershed delineation functionality, outlined in Figure 10. First, a point shapefile is created using a list NHD+ output locations. Five additional attribute fields are created to match the format of the Arc Hydro Tools "Batch Point" file that is required as input to the *Batch Subwatershed Delineation* tool. Next, values for these attribute fields must be assigned, specifically, *BatchDone* is set to 0 and *SnapOn* is set to 1. These values indicate that (1) batch processing has not been completed and (2) that each outlet must be snapped onto the river network. Once complete, these points are imported into the Arc Hydro Tools geodatabase using ArcCatalog. Next, the river network is defined using the *Stream Definition* tool. This tool creates a raster product derived from the NHD+ flow accumulation grid that consists of cells having an accumulation greater than a user defined value. Finally, the *Batch SubWatershed Delineation* tool is executed to delineate basins for each of the outlets provided.

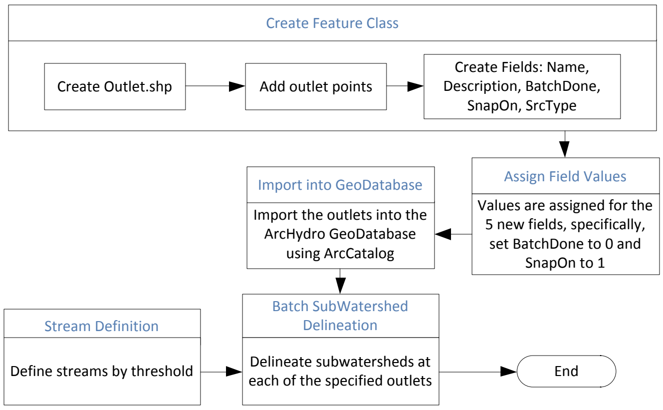

Figure 10: *The procedure for subwatershed delineation within ArcGIS, using Arc Hydro Tools.*

Both of these approaches were applied to delineate watersheds at 49 locations, corresponding to USGS streamflow monitoring gages. Figure 11 shows the boundaries that were delineated using each of the aforementioned GIS approaches. The hierarchal technique finishes this operation in 69 seconds, whereas using Arc Hydro this took 2 minutes. Again, discrepancies exist between the boundary calculated by Arc Hydro Tools using raster computations and the boundary that was assembled from NHD+ catchments using the hierarchical approach. Insets (i) and (ii) of Figure 11 illustrate the nature of the boundary differences we found. In both cases, small catchment areas are left out of our calculation which contradicts the boundary calculated directly from the terrain elevation using raster calculations. As described in Section 4.1, we believe that this is a direct result of how the NHD+ dataset was created and the flow attributes therein. In the first inconsistency, NHD+ derived boundaries don't exactly match the those calculated using raster computations. Upon further inspection it appears that this has been corrected in version 2 of the NHD+ dataset. The second case, however, is due to the river flow attributes that we used to trace upstream reaches and subsequently catchments. This may be due to corrections that were made to the natural terrain to account for the actual hydrography of the surface, or they could mistakes within the NHD+ dataset that must be corrected.

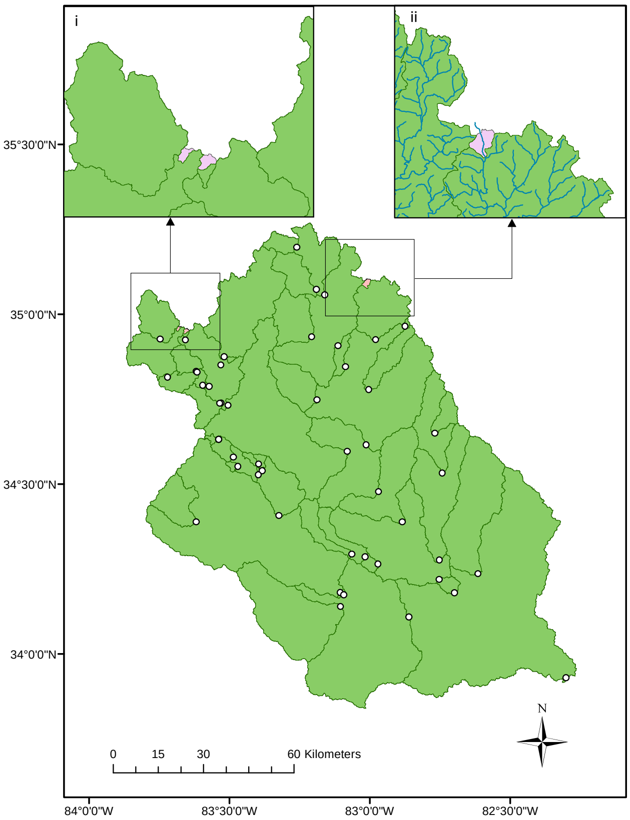

Figure 11: *Subbasins were delineated at USGS streamflow stations using Arc Hydro Tools and the hierarchical approach. Boundary differences were found when compared with raster-based delineation and are illustrated by the lighter catchments in insets (i) and (ii).*

5. Summary and Discussion

A watershed delineation technique was presented that uses existing GIS vector and raster data to resolve watershed boundaries for a wide range of spatial scales. It leverages freely available input data and open-source software which makes it easily accessible to a wide range of hydrologic scientists. Traditional watershed delineation approaches perform raster computations directly on DEM's, which inadvertently results in redundant computations (Djokic and Ye, 2000). Our approach is advantageous when large pre-delineated watershed datasets are available. When this is not the case, traditional DEM processing may be the preferred option. While these datasets can be created manually (Djokic and Ye, 2000), the encouragement of programs similar to the USGS NHD+ by international agencies, would enable our algorithm to be easily adopted for a wide range of hydrologic science applications. This approach will not replace traditional raster processing algorithms which, among numerous other applications, is instrumental to deriving raster data products required for model simulation (e.g. Quinn*et al.*, 1995) . However, its versatility lends it useful as a hydrologic data processing tool that can be used to spatially summarize data attributes on a catchment or subcatchment level. It can also be used as a boundary for search, collection, and/or extraction of simulation input data (e.g. observation and spatial data) in a web based environment.

By leveraging pre-processed watershed catchment vectors, our technique offers an efficient solution to watershed delineation. Furthermore, the input data used to construct watershed boundaries is pre-processed by the USGS (i.e. NHD+), thus eliminating time intensive processing routines which have been necessary in past works (e.g. Djokic and Ye, 2000; Arge *et al.*, 2006; Danner *et al.*, 2007). In addition, the NHD+ dataset has been checked for accuracy by an interdisciplinary team of USGS and U.S. EPA scientists (Bondelid *et al.*, 2010). Moreover, the quality of watershed delineations should continue to improve with each release of the NHD+ dataset. For example implementing our method on the newest version of the NHD+ (version 2) will automatically correct any errors that were detected in the previous version of the dataset. This is significant because it streamlines the process of upgrading surface data by eliminating the need to download numerous individual elevation raster grids which not only saves time, but also storage space.

Our approach for watershed delineation uses NHD+ data products to rapidly assemble watershed boundaries. First, a small portion of the watershed is determined by tracing the grid cells upstream of the outlet location. This upstream trace identifies all cells that contribute flow to the outlet, but that are limited to the NHD+ catchment in which the outlet resides. This was made possible by borrowing from mathematical graph theory and leveraging the NetworkX graphing library (Hagberg *et al.*, 2008). The upper portion of the watershed is determined using the flow relationships between the NHD+ digitized river reaches to identify all upstream rivers and their respective catchments. The geometries for these catchments are then merged and later combined with the lower portion of the watershed to complete the delineation. Because this technique does not require grid processing, it can rapidly resolve a watershed boundary with a little computational overhead. Furthermore, it was shown that our algorithm is advantageous for delineating watersheds at a wide range of scales as well as delineating subwatersheds.

The application of our delineation approach demonstrates a method for delineating watersheds and subwatersheds at various scales in a time efficient manner. However, the results of Section 4 show that boundary differences exist between the watersheds delineated by our approach and those derived directly from rasters (i.e. ArcGIS, TauDEM, Arc Hydro Tools). The variations in watershed boundaries are a result of the datasets used to derive the NHD+ catchment boundaries and flow relationships, i.e. modified DEM and the WBD rasters used in combination with a watershed delineation algorithm designed to produce the best agreement of available data (Johnston *et al.*, 2009). This method allows experts to modify remotely sensed terrain data to ensure that it is consistent with known field measurements, unlike traditional raster-based watershed delineation. However, the NHD+ dataset may contain processing errors such as those outlined in Figure 11. In situations such as this, a manual correction may have been made to the dataset to account for the actual hydrography of the surface (e.g. an obstruction to river flow), however it could also be a mistake. We must consider these watershed boundary differences inherent to the dataset used by our hierarchical algorithm, which in this case is the NHD+. If desired, a custom or alternate watershed boundary dataset can be used in place of the NHD+ for greater quality assurance. For example, this work uses the NHD+ version 1.0 dataset, however, a newer version of the dataset is now available that consists of the most accurate and up-to-date data. In fact, it appears that some of these boundary differences have been corrected in the NHD+ version 2. A significant advantage of our approach is its ability to easily adapt to newer, more accurate, datasets without having to process large datasets.

Overall, our technique consists of a light-weight software algorithm that is implemented in the Python programming language and mathematical graph theory to process watershed boundaries. NHD+ network relationships are stored in serialized graphs, while spatial data are stored in a PostgreSQL spatial database that leverages PostGIS functionality. However, this algorithm has also been adapted to leverage SQLite database storage. This versatility demonstrates the portability of this technique, and as a result, it may be a good candidate for remote hosting via web services or deployment in cloud environments. Future work will investigate how this technique can be deployed as a web service to provide on-demand watershed delineations by leveraging emerging cloud computing environments such as Microsoft Azure or Amazon Elastic Compute Cloud (EC2). A service such as this could then be used to retrieve, process, and summarize input data for models. In addition, this technique can be expanded upon to supply users with other NHD+ data products or attributes consisting of new or summarized data. This is particularly useful when preprocessing data to create model input files.

6. Acknowledgments

This work was supported by the National Science Foundation under the grant NSF EAR 1048125 “Collaborative Research: CiC (SEA): Using the Cloud to Model and Manage Large Watershed Systems.”

7. Software Availability

The delineation software presented in this paper is available for download under the GNU General Public Licence V3 at `https://bitbucket.org/Castronova/hierarchical-watershed-delineation`.

References

Ames, Daniel P., Jeffery S. Horsburgh, Yang Cao, Jiri Kadlec, Timothy Whiteaker, and David Valentine., 2012. HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis. *Environmental Modelling & Software, 37, 146-156.*

Arge, L., et al., 2003. Efficient flow computation on massive grid terrain datasets. GeoInformatica, 7 (4), 283–313.

Arge, L., et al., 2006. I/O-efficient hierarchical watershed decomposition of grid terrain models. Progress in Spatial Data Handling, 825–844.

Bondelid, T., et al., 2010. NHDPlus User Guide Version 1. Technical report, United States Geological Survey.

Chaudhry, S., et al., 2005. High-performance throughput computing. Micro, IEEE, 25 (3), 32–45.

Chow, V., et al., 1988. Applied hydrology. Mc-Graw Hill, New York.

Danner, A., et al., 2007. TerraStream: from elevation data to watershed hierarchies. In: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems, p. 28.

Djokic, D. and Ye, Z., 2000. DEM Preprocessing for Efficient Watershed Delineation. In: D. Maidment and D. Djokic, eds. Hydrologic and Hydraulic Modeling Support with Geographic Information Systems. Esri Press, 65–84.

Douglas, D.H., 1986. Experiments to locate ridges and channels to create a new type of digital elevation model. Cartographica, 23 (4), 29–61.

Gillies, S., 2013. The Shapely User Manual. [online] `http://toblerity.github.io/shapely/manual.html` *[Accessed: 1/28/2013].*

Gong, J. and Xie, J., 2009. Extraction of drainage networks from large terrain datasets using high throughput computing. Computers & Geosciences, 35 (2), 337–346.

Guthrie, J.D. Dartiguenave, Christine, and Ries, K.G., III, 2009. Web Services in the U.S. Geological Survey StreamState Web Application. In: The International Conference on Advanced Geographic Information Systems & Web Services (GEOWS), IEEE, pp. 60-63.

Hagberg, A.A., Schult, D.A., and Swart, P.J., 2008. Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference (SciPy2008), Aug.., Pasadena, CA USA, 11–15.

Hodgson, M., 1998. Comparison of angles from surface slope/aspect algorithms. Cartography and Geographic Information Science, 25 (3), 173–185.

Huang, F., et al., 2011. Explorations of the implementation of a parallel IDW interpolation algorithm in a Linux cluster-based parallel GIS. Computers & Geosciences, 37 (4), 426–434.

Huang, Q. and Yang, C., 2011. Optimizing grid computing configuration and scheduling for geospatial analysis: An example with interpolating DEM. Computers & Geosciences, 37 (2), 165–176.

Hutchinson, D., et al., 1996. Parallel neighbourhood modelling. In: Proceedings of the 4th ACM international workshop on Advances in geographic information systems, 25–34.

Hutchinson, M., 1989. A new procedure for gridding elevation and stream line data with automatic removal of spurious pits. Journal of Hydrology, 106 (3), 211–232.

Jenson, S. and Trautwein, C., 1987. Methods and applications in surface depression analysis. In: Proc. Auto-Carto, Vol. 8, 137–144.

Johnston, C.M., et al., 2009. Evaluation of Catchment Delineation Methods for the Medium-Resolution National Hydrography Dataset. Technical report, United States Geological Survey.

Kopp, S., 2013. Custom Watersheds at the Click of a Button: Watershed Delineation in ArcGIS Online. ArcGIS Resources, ESRI. August 13, 2013. `http://blogs.esri.com/esri/arcgis/2013/08/13/custom-watersheds-at-the-click-of-a-button-watershed-delineation-in-arcgis-online`

Lu, W., Jackson, J., Ekanayake, J., Barga, R.S., and Araujo, N., 2010. Performing large science experiments on Azure: Pitfalls and solutions. In: IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), 209–217

Maidment, D.R., ed. ArcHydro: GIS for water resources. Esri, Inc., 2002.

Marcus, D. (2008). Graph theory: A problem oriented approach. Mathematical Association of America.

Mineter, M., 2003. A software framework to create vector-topology in parallel GIS operations. International Journal of Geographical Information Science, 17 (3), 203–222.

Mineter, M., Dowers, S., and Gittings, B., 2000. Towards a HPC framework for integrated processing of geographical data: encapsulating the complexity of parallel algorithms. Transactions in GIS, 4 (3), 245–261.

Moore, I., Grayson, R., and Ladson, A., 2006. Digital terrain modelling: a review of hydrological, geomorphological, and biological applications. Hydrological processes, 5 (1), 3–30.

O'Callaghan, J. and Mark, D., 1984. The extraction of drainage networks from digital elevation data. Computer vision, graphics, and image processing, 28 (3), 323–344.

Quinn, P. F., Beven, K. J., and Lamb, R., (1995). The $\ln(a/\tan\beta)$ *index: How to calculate it and how to use it within the Topmodel framework. Hydrological processes, 9(2), 161-182.*

Ries, K.G., III, Steeves, P.A., Guthrie, J.D., Rea, A.H., and Stewart, D.W., 2009. Stream network Navigation in the U.S. Geological Survey StreamStats Web Application. In: The International Conference on Advanced Geographic Information Systems & Web Services (GEOWS), IEEE, pp. 80-84.

Rosen, K., 2003. Discrete Mathematics and Its Applications 5th edition. McGraw-Hill Science.

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

Tarboton, D., 2010. Terrain Analysis Using Digital Elevation Models (Taudem version 5.0), Utah Water Research Laboratory, Utah State University, `http://hydrology.usu.edu/taudem/taudem5/documentation.html`

Tesfa, T., et al., 2011. Extraction of hydrological proximity measures from DEMs using parallel processing. Environmental Modelling & Software, 26 (12), 1696–1709.

Thain, D., Tannenbaum, T., and Livny, M., 2005. Distributed computing in practice: The Condor experience. Concurrency and Computation: Practice and Experience, 17 (2-4), 323–356.

Wallis, C., et al., 2009. Hydrologic Terrain Processing Using Parallel Computing. In: R. Anderssen, R. Braddock and L. Newham, eds. 8th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation. Modeling and Simulation Society of Australia and New Zealand Incorporated, 2540–2545.

Wang, S. and Armstrong, M., 2009. A theoretical approach to the use of cyberinfrastructure in geographical analysis. International Journal of Geographical Information Science, 23 (2), 169–193.

Xie, J., 2012. Implementation and performance optimization of a parallel contour line generation algorithm. Computers & Geosciences, 49, 21–28.