Highlights

Fast, accurate watershed delineation with a hybrid of raster and vector methods

Matthew Heberger

- Open source software in Python enables delineation of watersheds of any size.
- Hybrid delineation using vector and raster data is faster than conventional methods.
- The Global Watersheds web app brings watershed delineation to the general public.

Fast, accurate watershed delineation with a hybrid of raster and vector methods

Matthew Heberger[a]

[a]*San Francisco Estuary Institute, 4911 Central Ave, Richmond, 94804, CA, United States*

Abstract

We describe a method of watershed delineation that combines vector-based methods (fast but approximate) and raster-based methods (more accurate but computationally expensive). This hybrid method is faster and uses less memory than existing software, and can be run on a normal laptop computer. This allows the delineation of very large watersheds, such as the Nile or the Mississippi, using high-resolution 90 m terrain data, in under 2 minutes on a laptop computer. While the hybrid concept is established, this work provides a rigorous validation, along with a modern, open-source implementation. We introduce a Python software package for performing watershed delineation almost anywhere on Earth using this method and the MERIT-Hydro and MERIT-Basins datasets. Benchmarking shows that the software is 10–100 times faster than conventional methods of watershed delineation, with equivalent accuracy. Compared with industry-standard software from ESRI, our software's watersheds were a better match with USGS gage watersheds, with a coefficient of areal correspondence of 0.93. Finally, we describe Global Watersheds, a free and easy-to-use web application that allows anyone to find watersheds and downstream flow paths over almost all of the Earth's land area (excluding Antarctica and some small islands). The app democratizes access to state-of-the-art hydrological analysis tools that were previously limited to specialists, making it a powerful tool for education, policy, and research.

Keywords: watersheds, rivers, Python, network analysis, open-source software, Digital Elevation Models

Email address: `matth@sfei.org` (Matthew Heberger)

1. Introduction

Watersheds, also referred to as *catchments* or *drainage basins*, are fundamental geographic units in hydrology. Watershed *delineation* is the process of finding a watershed's boundaries. It is vital to many fields of science and engineering, such as the planning and analysis of water supply, flooding, aquatic habitat, or the fate and transport of pollutants. Historically, analysts delineated watersheds manually using paper topographic maps, using the elevation contour lines to determine the direction of steepest slope and the direction of water flow. Computerized methods for watershed delineation were introduced in the 1980s and have been widely used since. The basic methods of computerized watershed delineation were described by O'Callaghan and Mark (1984), and important contributions in data and software were made over the next two decades. Thorough overviews of watershed delineation using modern software is given by Maidment and Djokic (2000) and Johnson (2009).

Automated watershed delineation can sometimes be more accurate than manual delineation, especially where high-resolution terrain data is available. In any case, it is more repeatable and removes analysts' subjectivity. Today, watershed delineation tools are a part of several Geographic Information System (GIS) software packages (e.g., ArcGIS, QGIS, GRASS, Whitebox, Global Mapper). In addition, there are standalone programs such as TauDEM (Tarboton, 2016), and watershed delineation functions in specialty hydrology software such as HEC-GeoHMS and SWAT+. Libraries are also available for Python (e.g., pysheds, PyFlwDir, PCRaster) and Matlab (Schwanghart and Scherler, 2014, TopoToolbox).

Most computer methods for watershed delineation use gridded, or *raster*, datasets that represent the elevation of the land surface and the direction and accumulation of water flow. Another class of methods uses *vector* data, with polylines representing rivers and flow pathways, and polygons representing subdivisions of drainage basins. Figure 1 illustrates the difference between vector and raster data models. Vector methods can be much faster, but the spatial resolution of the output watershed is limited by the size of the input data's smallest polygons or unit catchments. Raster-based methods may be more accurate, but for large watersheds or for high-resolution input data, the analysis can be slow and require more computer memory than is available to

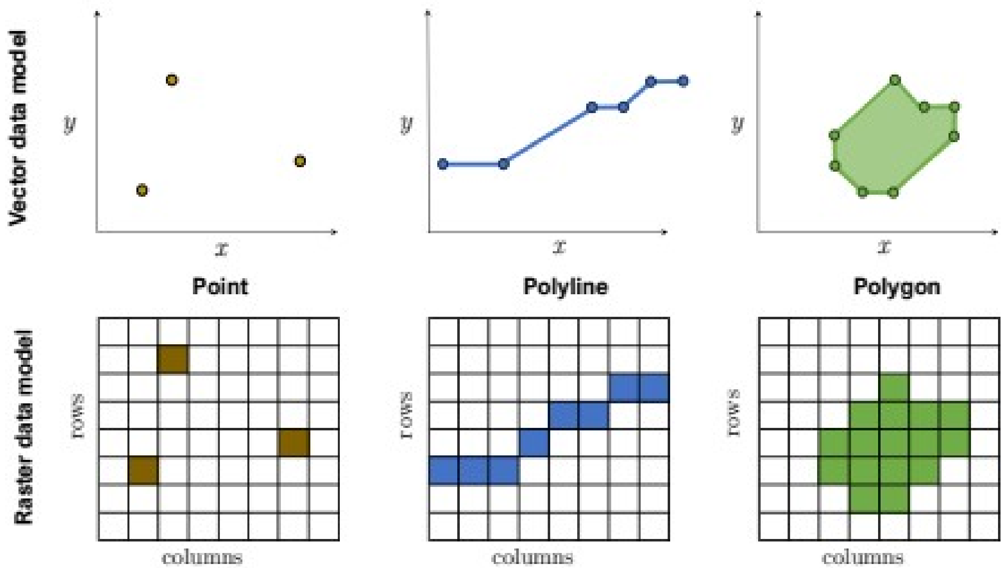


Figure 1: Raster and vector data models for geographic data.

some analysts.

The development of software described here was motivated by the author's need to delineate a large number of watersheds for studies in large-scale hydrology (LSH). This subdiscipline of hydrology investigates the movement and storage of water at spatial scales that range from multiple river basins to the entire globe (Cloke and Hannah, 2011; Addor et al., 2020). One of several challenges in large-scale investigations is the difficulty in accurately delineating river basin boundaries (Kauffeldt et al., 2013). The ability to create large numbers of watersheds, quickly and accurately, will facilitate the next wave of innovative studies in the hydrologic sciences. [1]

1.1. Raster methods of watershed delineation

Automated watershed delineation uses raster elevation data from a Digital Elevation Model (DEM). Raw DEM data generally produces inaccurate

[1] Examples of notable applications that depended on large collections of catchments include: Prediction in Ungaged Basins (Hrachowitz et al., 2013); the Panta Rhei Initiative of the International Association of Hydrological Sciences (IAHS) focused on change in hydrology and society (Blöschl et al., 2025); and hydrologic forecasting with machine learning (e.g., Nearing et al., 2024).

flow networks, so a number of processing steps create a "hydrologically conditioned" DEM, typically including: fill pits, resolve flats, burn streams, and fence ridgelines. From the conditioned DEM, analysts create a flow-direction grid based on the steepest slope for each cell. The D8 algorithm, where water in each grid cell flows in 1 of 8 directions into an adjacent cell, remains the most widely used despite more complex alternatives like $D\infty$ (Tarboton, 1997).

For the most accurate results, analysts use the highest-resolution data available. The current standard for global terrain data in hydrology is 3 arcsecond resolution (approximately 90 m near the equator), as seen in HydroSHEDS (Lehner, 2022) and MERIT-Hydro (Yamazaki et al., 2019). This is 10 times better than datasets available 20 years ago, and this trend continues with the forthcoming HydroSHEDS version 2 based on 0.4 arcsecond (12 m) TanDEM-X data (Lehner et al., 2022).

Higher resolution brings challenges. File size and memory requirements increase quadratically with resolution. Large-scale hydrology studies often require high-performance computing facilities (e.g., Amatulli et al., 2022; Do et al., 2018), which are out of reach for many researchers, limiting the potential of high-resolution datasets.

One workaround for large watersheds is using lower-resolution data, common in large-sample hydrology (e.g., Pan et al., 2012) and global land surface modeling (e.g., Harrigan et al., 2020). However, flow-direction rasters from lower-resolution elevation data have historically had many errors requiring manual correction (Lehner, 2022), and watersheds automatically derived from coarse grids often have area errors exceeding 70% for basins over 5,000 km^2 (Kauffeldt et al., 2013).

Another approach involves *upscaling* raster datasets while preserving drainage network fidelity. The idea is to leverage high-resolution DEM accuracy to create accurate flow-direction grids, then upscale to more computationally lightweight resolutions. Watersheds delineated with lower-resolution data will have blocky boundaries, but if the polygon is accurate in size, extent, and location, it suffices for many applications. Eilander et al. (2021) developed an iterative algorithm for creating accurate upscaled flow-direction grids, as well as "scale invariant" estimates of river length and slope.

To speed up watershed delineation, recent approaches include enabling parallel processing for high-performance clusters or multi-core computers (e.g., Tarboton, 2016), leveraging GPUs (Kotyra, 2023; Kumar et al., 2025), and tiling strategies that reduce memory usage for continental-scale datasets

(Zhao et al., 2025).

The "watershed marching algorithm" by Haag et al. (2020) finds a watershed's *perimeter* rather than every interior pixel, achieving 100–1,000 times faster performance than ESRI software. However, this requires precomputing a special data structure, and no source code is available.

Im algorithms and GPU capabilities can speed up watershed delineation compared to existing software. In this paper, we take a different approach: rather than optimizing raster processing, we seek to *minimize* the necessary raster processing and make use of more efficient vector-based geographic operations.

1.2. Vector methods of watershed delineation

For analysts not concerned with pixel-perfect accuracy, or where quality raster data are unavailable, vector-based methods offer a different approach to delineating watersheds. We are aware of three different approaches based on vector data. The first method can be applied where terrain data is unavailable (or unusable due to low quality), but a vector representation of the river network map is available. Karimipour et al. (2013) developed a Voronoi polygon-based medial axis extraction algorithm that was shown to be "reasonably comparable" to DEM-based methods. A second method is based on Triangulated Irregular Networks (TINs), alternatives to raster data that efficiently describe terrain. Several delineation methods using TINs have been proposed (Jones et al., 1990; Mee-Jeong et al., 2006; Freitas et al., 2016).

The third vector-based method of watershed delineation makes use of vector hydrography datasets that divide the land surface *unit catchments* or *subwatersheds*. Typically, the unit catchments and river reaches have a 1:1 relationship and share a common identification number. Reaches (and unit catchments) are related to one another via a network topology, an abstraction of the real-world flow network. Using network theory, a river network can be modeled as an *acyclic directed graph*.

Analysts may create their own, custom vector datasets by digitizing topographic data, or by processing digital terrain data using GIS software (e.g., ESRI, GRASS, SAGA plugins for QGIS), or specialized software for hydrologic data processing (e.g., TauDEM, WhiteBox, HEC-GeoHMS). The size of unit catchments depends on the resolution of the terrain data and the user's selected *drainage density*—the upstream area or number of pixels considered necessary for stream formation.

One well-known vector hydrographic dataset with coverage of the United States is the Watershed Boundary Dataset published by the U.S. Geological Survey (USGS, 2022).[2] Datasets with near-global coverage include HydroSHEDS (Lehner, 2022), HDMA (Verdin, 2017), MERIT-Basins (Lin et al., 2019), and TDX-Hydro (Carlson et al., 2024).

To delineate a watershed using vector hydrography data, you first find the terminal (most downstream) or "home" unit catchment—the one in which the watershed outlet is located. This is done via a straightforward geospatial overlay analysis. Using techniques from network analysis, you next find all of the upstream unit catchments. This can be done efficiently in Python using the NetworkX library. There are also tools or plugins for GIS to perform such analyses (e.g., Network Tools in ArcGIS or the Flow Trace plugin for QGIS). The result from the network analysis is the set of river reaches (and unit catchments) that contribute flow to your chosen outlet. For display, the set of unit catchments can be merged and dissolved to create the final watershed polygon.

Watershed creation with vector data is less computationally demanding than raster methods, as it uses larger building blocks; unit catchments are typically hundreds or thousands of times bigger than pixels.[3] However, watershed boundaries created with vector unit catchments will usually be downstream of the requested outlet, and the watershed will be larger than it should be. As a rule, the watershed boundary should pass through the outlet point. So with vector methods of watershed delineation, we have a tradeoff for increased speed in terms of lower accuracy and an area bias.

Area errors of a few percent will have minimal impact on hydrologic analyses like calculating water balances. Regardless, it just *looks wrong* when you zoom in and the watershed boundary does not coincide with the outlet point. Such errors may be unacceptable to cartographers, picky clients, or persnickety advisors. Greater precision may also be necessary for studying water pollution, where it is important to pinpoint the location of contaminant sources. When the delineated watershed is too large, we may falsely identify locations as part of the contributing area when they are, in fact, downstream

[2] The USGS uses the term Hydrologic Unit (HU) for individual catchments, and each is assigned a unique Hydrologic Unit Code (HUC).

[3] With the datasets used in this paper, MERIT-Basins unit catchments average 45 km^2, over 5000 times bigger than the 0.008 km^2 pixels in MERIT-Hydro from which they are derived.

of the outlet.

1.3. Hybrid method of watershed delineation

The "hybrid" method of watershed delineation leverages the strengths of both vector and raster data. First, we use vector data and network analysis to find the set of unit catchments upstream of the outlet. Then, to find the watershed boundary near the outlet, we use more computationally expensive raster methods to "split" the downstream unit catchment. The results of the two analyses are then merged. With this method, we can achieve pixel-perfect accuracy with much greater speed.

The hybrid method for watershed delineation is not new. It was described in a 1999 conference paper by Djokic and Ye. The authors called their method Fast Watershed Delineation, or FWD. They wrote that it was first created at ESRI in 1997 as a set of scripts called Watershed Delineator, "developed for the Texas Natural Resource Conservation Commission, with the sole purpose of efficiently delineating watersheds." Instead of using an existing vector dataset, their scripts create a new set of unit catchments at runtime using the Arc/Info `subwatershed` command. While these scripts were in the public domain, they required proprietary ESRI software to run them. The scripts were written in Avenue, the defunct scripting language for ESRI's ArcView, which the company stopped supporting in 2002.

Since then, the hybrid method has been used by the U.S. Geological Survey (USGS). Examples include the NHD Watershed Tool for ArcView (Steeves, 2003) and NHDPlus Tools for ArcMap (Horizon Systems, 2010). These scripts also require proprietary ESRI software, and they are limited to watersheds in the continental United States. The USGS also uses the hybrid method of watershed delineation in the web application StreamStats (USGS, 2021) and for its application programming interface (API) called NLDI (Blodgett, 2022). Finally, Castronova and Goodall (2014) describe a similar method of watershed delineation using both raster and vector data, although the code is no longer available.

Despite the fact that these methods have been in use for over two decades, many researchers (particularly those who work outside the US or do not use ArcGIS) may be unaware of them. In this paper, we introduce an implementation of the hybrid method of watershed delineation that is open source and based on free software and data.

2. Software Description

The software described here, **delineator.py**, is a set of Python scripts that can be run on any computer. The scripts described here have been created specifically to work with two related datasets with global coverage, MERIT-Hydro (raster) and MERIT-Basins (vector), and could be readily adapted to use other datasets. The required data files are:

- flow direction (raster)
- flow accumulation (raster)
- unit catchments (vector polygons)
- river reaches (vector polylines)

The data files are organized into 61 continental-scale basins (Pfafstetter level 2 basins), referred to here as *megabasins* (see Figure 2). This project includes sample data for Iceland (megabasin 27) and links to download data files for all other regions.

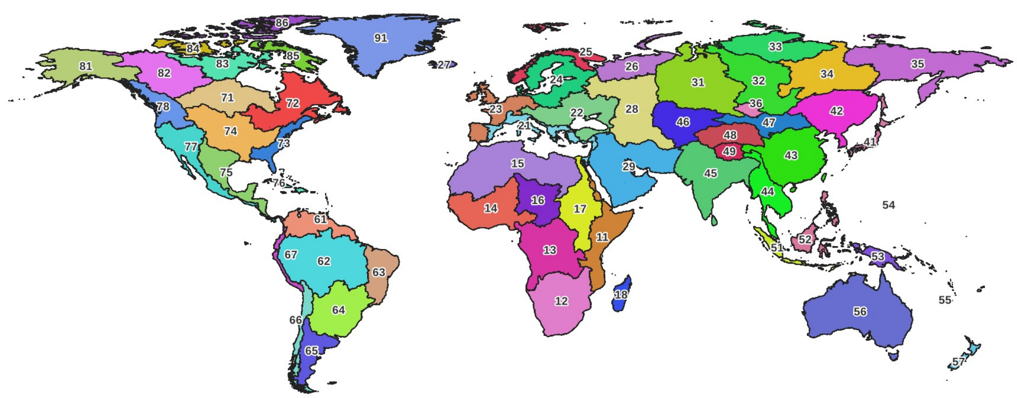
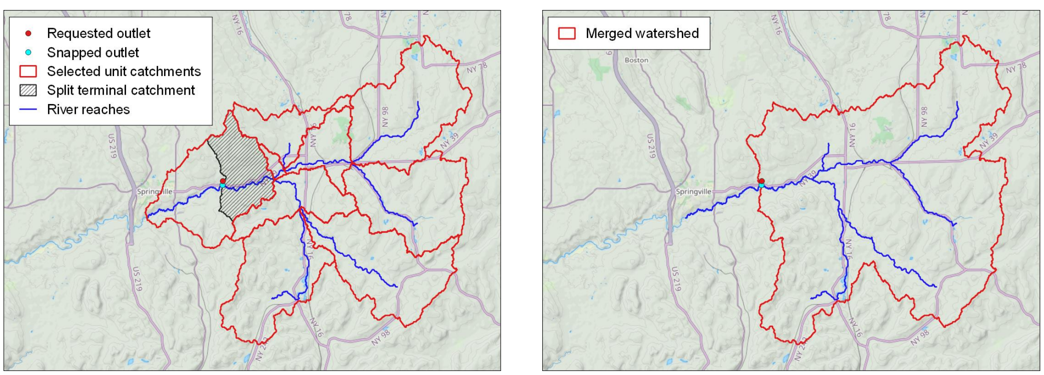


Figure 2: Data files for use with this project are organized into 61 continental-scale megabasins.

An overview of the main steps in the hybrid method of watershed delineation follows, with a more detailed explanation below.

1. Find the continental-scale megabasin in which the outlet falls (see Figure 2. This tells the software which data files to use in the analysis.
2. Find the unit catchment corresponding to the watershed outlet. We refer to this as the *terminal* unit catchment or the *home* unit catchment.
3. Identify all of the unit catchments upstream of the home unit catchment.
4. Split the home unit catchment to find the exact boundary around the outlet.
5. Merge and dissolve the upstream unit catchments found in step 2 with the split unit catchment found in step 3. This is the final watershed.

These steps are illustrated in Figure 3. This example is for a relatively small (500 km^2) watershed, on Cattaraugus Creek in western New York, USA. For step 1, the outlet point is Basin 72, which covers parts of northeastern United States and Canada (see Figure 2. This tells the script which data files to read in the following steps.



(a) Unit catchments

(b) Final watershed

Figure 3: Illustration of vector polygon unit catchments which are merged to create the watershed.

For step 2, the script performs a spatial join on the outlet point and the unit catchments geodata layer. Here, the outlet falls in a unit catchment with a COMID (for common ID) of 72057856.

For step 3, the script uses network analysis to find the home unit catchment's upstream neighbors. In this case, the script finds 8 upstream unit catchments. The network data for step 3 comes from the MERIT-Basins rivers shapefile. Reading shapefiles and creating a GeoDataFrame is a fairly slow process. Once you have done this once, you can optionally save time in the future by storing the GeoDataFrame as a Python pickle (.pkl) file. Note that these files are not any smaller than the original shapefile, so they do not save disk space; they are just much faster to load so future watershed delineation will be faster.

For step 4, the script uses raster methods to *split* the home unit catchment at the outlet point. Step 4 is illustrated in Figure 4. In brief, first, the pour point is snapped to the stream flow line, or the grid cell with a sufficiently high flow accumulation. (For pour point snapping options, see Section 2.1.4.) Next, we use the flow-direction raster to find the set of grid cells that are upstream of this point. The result is a raster mask, which we then *polygonize* or convert to a vector polygon.

For step 4, the script uses the Python library `pysheds` (Bartos et al., 2023). The app reads a slice or "window" from raster datasets (flow direction and flow accumulation) in a bounding box around the unit catchment. Next, we mask out pixels that do not overlap the unit catchment by setting them to NaN (not a number). This step to be the key to getting reliable results, which we discovered through trial and error. Without this step, there is a risk of the pour point being snapped to another nearby stream.

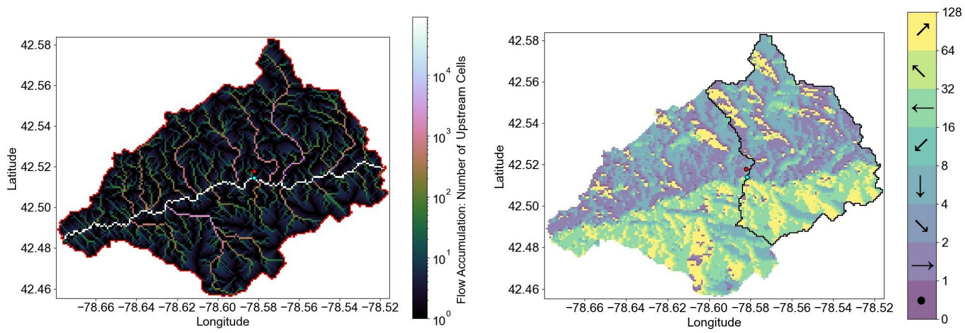
Finally, in step 5, we merge and dissolve the individual unit catchments into the final watershed polygon. For large watersheds with thousands of unit catchments, dissolving is still the slowest step in the delineation pipeline. The result of the dissolve operation is the final watershed polygon, which may then be saved to one of several possible formats for display or analysis.

2.1. Options and configuration

The Python scripts in delineator.py are configurable and offer several options to users. Some options govern what outputs will be produced, while other options affect the watershed's appearance.

2.1.1. Higher-resolution threshold

In some cases, high precision is not necessary. Configuration options include a "lower resolution threshold", in square kilometers. Lower-resolution mode has two main differences. First, the program will use simplified unit catchment boundaries. Because these polygons have fewer vertices, processing them is faster. Second, the script will not perform the detailed raster-based calculations near the outlet. As a result, the watershed will contain



(a) Flow accumulation (b) Flow direction

Figure 4: Illustration of detailed raster-based methods used to split the terminal unit catchment, or draw an accurate boundary at the outlet. At left, the outlet is snapped or relocated to a stream, or a grid cell with a high flow accumulation. At right, the set of cells upstream of the outlet is determined based on flow direction.

some extra area downstream of the requested outlet. On the author's laptop, it took 93 seconds to delineate the 6 million km² Amazon watershed using high-resolution data, versus 27 seconds in lower-resolution mode. We recommend setting the lower-resolution threshold at around 50,000 km² for general hydrology applications, or higher if you intend to publish maps or figures where appearances matter. The loss in precision is barely noticeable in large watersheds and at low zoom levels.

2.1.2. Simplify shapes

The geodata output by the script may contain more detail than you need for display or analysis, and unnecessarily large output files. To remove some vertices from the watershed boundary and river flowlines, set `SIMPLIFY = True`. You will also need to set `SIMPLIFY_TOLERANCE` to a value in decimal degrees, corresponding to the distance parameter in the Douglas-Peucker algorithm for polyline simplification. Note that the MERIT-Basins vector data is based on MERIT-Hydro raster data with a resolution of 3 arcseconds, or $\approx 0.000833°$. Setting the simplification tolerance to half this value removes the jagged "stair step" appearance typical of polygonized raster data without much loss of detail, and reduces file sizes by over 50%.

2.1.3. Interactive map

The script can output a browser-based interactive map, a fast and convenient way to review the results. An HTML map page is output when you set

MAKE_MAP = True. Interactivity is provided by JavaScript and the DataTables and Leaflet plugins. For large watersheds, the river network may be too dense, and contain too many polylines to display, causing the browser to lag or crash. Therefore, the app will *prune* the river network, removing small tributary streams, based on Strahler stream order (Strahler, 1957). We recommend setting NUM_STREAM_ORDERS to 4 or less.

2.1.4. Pour point snapping

A common challenge in watershed delineation is "snapping the pour point" to coincide with a flowline in the digital dataset. While various methods have been implemented, no clear consensus exists on the best approach (Lindsay et al., 2008; Xie et al., 2022). Our script includes two methods. The conventional approach snaps the outlet to the closest point on the river reach within the home unit catchment. Sometimes the requested outlet does not fall inside a unit catchment, especially if you are searching for watershed outlets near the coast. The SEARCH_DISTANCE parameter (in decimal degrees) controls the search radius when the point does not overlap a catchment. Setting SEARCH_DIST = 0 requires the point to fall inside a unit catchment. We recommend at least 0.005° for reliable results, with larger values accommodating greater coordinate uncertainty.

A second, experimental method applies when you have an a priori area estimate of the watershed area. Setting MATCH_AREAS = True searches the neighborhood around the outlet for a river reach whose upstream area is the closest match. This method is experimental and works best for watersheds over 1,000 km^2. Manual adjustment of the pour point toward the correct river centerline often yields better results.

2.1.5. Fill internal holes

Oftentimes, watersheds created by computer will contain internal gaps or "donut holes." Many small gaps and slivers are data artifacts, which we recommend filling. To do so, set FILL = True. Larger holes may represent sinks, where water flows in but not out. As an example, consider North America's Rio Grande watershed, shown in Figure 5. Between the two main branches, the Rio Grande in the west and the Pecos River in the east, there is an endorheic basin that runs north-south for 560 km from New Mexico to Texas. Within this basin, there are several alkaline lakes or *playas*. These may be important for surface water flow and overland runoff, and less so for studies of groundwater or basin water budgets. To keep larger holes, and enter a non-zero value for the FILL_THRESHOLD parameter.

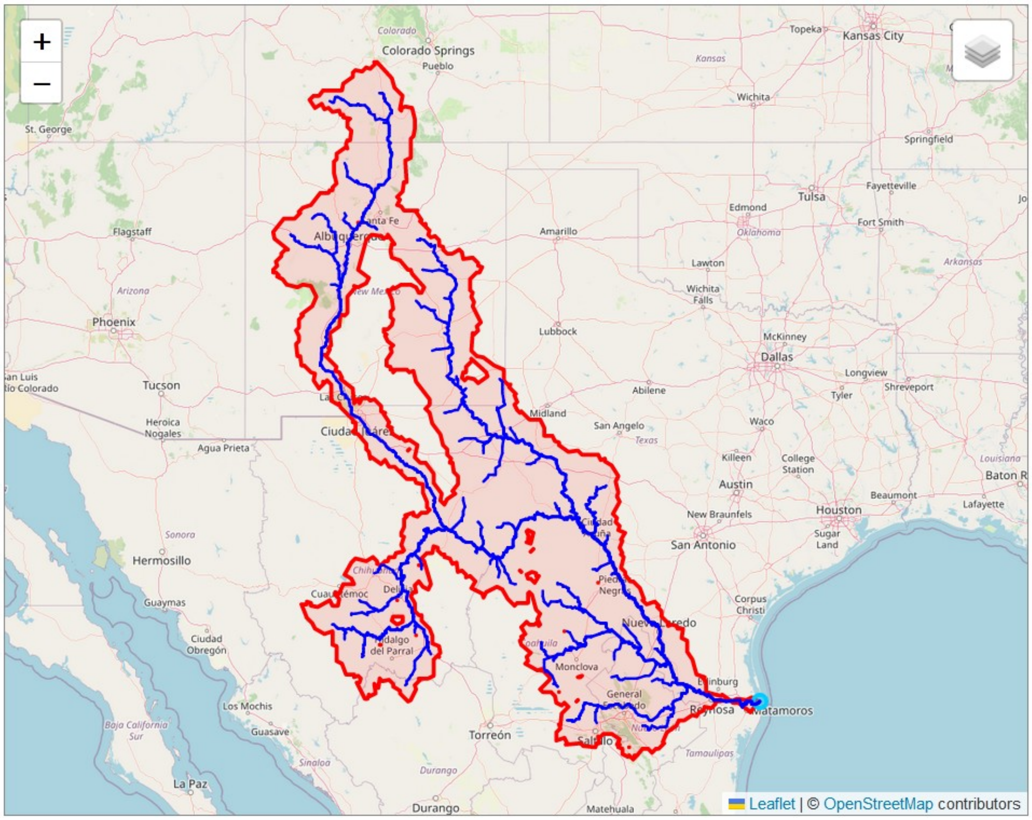


Figure 5: Rio Grande watershed showing endorheic basin between main branches.

2.2. Web application for watershed delineation

The web app Global Watersheds is an online implementation of the software described here, at `https://mghydro.com/watersheds`. The app lets users delineate watersheds or find downstream flow paths for nearly any point on Earth. The app is very easy to use; one simply clicks a point on a map and then clicks a button. More advanced features are available under an Options menu. For example, the user can choose from among three data sources: MERIT-Hydro, HydroSHEDS, or the USGS NLDI. There is also an option to delineate watersheds for a line or a polygonal area, rather than a single point. In addition to upstream watersheds, users can also trace downstream flow paths.

The web version includes certain features that make it faster than the desktop software, including a PostGIS-enabled PostgreSQL database and memoization of expensive function calls. In essence, the app gets faster over time, as it draws on an expanding library of saved results. Further, it uses robust web infrastructure (nginx web server, caching with Memcached, gzip compression) that is optimized for speed and efficiency. For example, users request the Amazon and the Nile watersheds almost every day, and the response time is under three seconds.[4]

The app allows users to download geodata of watershed boundaries and rivers in several formats (shapefile, geopackage, GeoJSON, and KML). There is also an option to create a watershed data report with information on land cover, population, hydrology, and climatology. These data are extracted from state-of-the-art global databases from a variety of sources, and summarized over the watershed area.

In November 2025, users delineated around 99,000 watersheds, and downloaded geodata for around 12,000 of them. Based on an informal poll conducted in the spring of 2023, over half the respondents are scientists and engineers, but there were also many students, educators, and members of the general public. Our main goal in building the app was to democratize access to hydrological analysis and to help raise "watershed consciousness" (Parsons, 1985). Or, in other words, the understanding that we all live in a watershed and that water connects us all, sometimes in surprising and unexpected ways.

2.3. Application Programming Interface (API)

An API for the web app allows one to get watersheds and rivers without using the map interface. In general, the API is faster and easier to use than the Python scripts; it does not require downloading large datasets, and takes advantage of the web optimizations described above. Full documentation of the API is provided at mghydro.com, along with example Python code in a Jupyter notebook.

[4] To put into perspective how transformational this is, in a 2014 forum post on Stack Exchange, a leading authority on terrain analysis wrote that, after downloading and processing the terrain data, "extracting the watershed for a point on the Nile delta took 2 hours and 9 minutes" (Lindsay, 2014).

3. Validation and Benchmarking

We claimed above that a hybrid of raster and vector methods should be faster and more accurate than using either method on its own. Here, we describe a set of experiments designed to test this assertion. The scripts for the benchmarking experiments are available in this paper's code repository. Benchmarking scripts were all run in the same virtual environment using Python v3.11 on the author's Dell laptop (Intel i7 processor, 32 GB of RAM) running Windows 10.

3.1. Validating the accuracy of watershed boundaries

Watersheds delineation with a flow-direction raster and the D8 algorithm are strictly deterministic. For a given point on the flow-direction grid, the set of upstream grid cells should always be the same, regardless of the software or algorithm used. So the first, most basic validation was to compare waterheds output from our software to those from established raster-based software packages. We compared output for several watersheds created with delineator.py to those created with TauDEM and pysheds. TauDEM is an industry standard software package that is distributed as a Windows executable (Tarboton, 2016). Pysheds is a Python library for watershed delineation (Bartos et al., 2023). (Note that delineator.py uses pysheds for raster processing. Here, for benchmarking, we are using pysheds by itself to do watershed delineation using purely raster methods.) In every instance, the watersheds produced by the three software packages were 100%identical down to the individual pixel scale. This confirmed that our implementation of the hybrid method of watershed delineation creates the same results as widely-used and tested software packages.

We performed a second validation test to evaluate the accuracy of watershed boundaries. This second validation experiment tested the software's capability when outlet points may not be precisely aligned with the raster datasets. In other words, it tests the entire workflow, including snapping the pour point and finding the upstream contributing area. We considered a watershed correct if it closely matched authoritative, published geodata for watershed boundaries. For this experiment, we found the watersheds corresponding to river flow gages in the United States operated by the U.S. Geological Survey (USGS). The USGS has high standards for review and verification of published data, requiring all watershed delineations to be checked by experienced hydrographers (Johnston et al., 2009; Jones et al., 2022).

Therefore, we were confident that these watershed boundaries are mostly correct and use them as the basis for validating our software's outputs.

To compare the similarity of two polygons, we used the Coefficient of Areal Correspondence (CAC), defined as the ratio of the intersecting area and the union area (Taylor, 1977). The CAC[5] has been used by researchers to compare the areas of two watersheds delineated with competing methods (e.g., Johnston et al., 2009). If the test watershed polygon is A, and the true polygon is B, the CAC can be stated as:

$$\text{CAC} = \frac{A \cap B}{A \cup B} \tag{1}$$

Some researchers have compared watersheds by comparing their areas, for example by creating a scatterplot and calculating R^2. This method has an important flaw however. Two watersheds may have a similar surface area, and yet be completely non-overlapping. In such cases, R^2 will be misleading and exaggerate the quality of the result.

We sampled 120 gages actively operated by the USGS in the continental United States, as shown in Figure 6.[6] To ensure we sampled a range of watershed areas, we stratified the population into quintiles and sampled 24 gages from each. We downloaded USGS watersheds using their web API, the Hydro Network-Linked Data Index (NLDI), and carefully reviewed each in QGIS. Then, we delineated watersheds for each gage using the latitude and longitude coordinates of the gage with our software delineator.py (in higher-resolution mode) and with the "Create Watersheds" function in ESRI's ArcGIS Online (ESRI).

Figure 7 shows the results of the validation experiment. On average, our software's watersheds are more accurate than those created with ESRI software (average $CAC = 0.93$, vs. ESRI's 0.83). However, where ESRI

[5] This metric appears to have been re-invented several times, and is also called the Jaccard Index or the Tanimoto Index (Wikipedia contributors, 2024). In computer vision, it is called Intersection over Union (IoU Rosebrock, 2016).

[6] We also tested our software for delineating watersheds globally, outside of the United States, and the results often appeared better than those returned by ESRI's software. However, we could not find a definitive source for the true watershed boundaries, so we do not report any results here. We hypothesize that the reason our software produces better results is related to data sources; it appears that ESRI's watershed tool uses HydroSHEDS terrain data outside of the United States, while our software is using newer, more accurate MERIT-Hydro data.

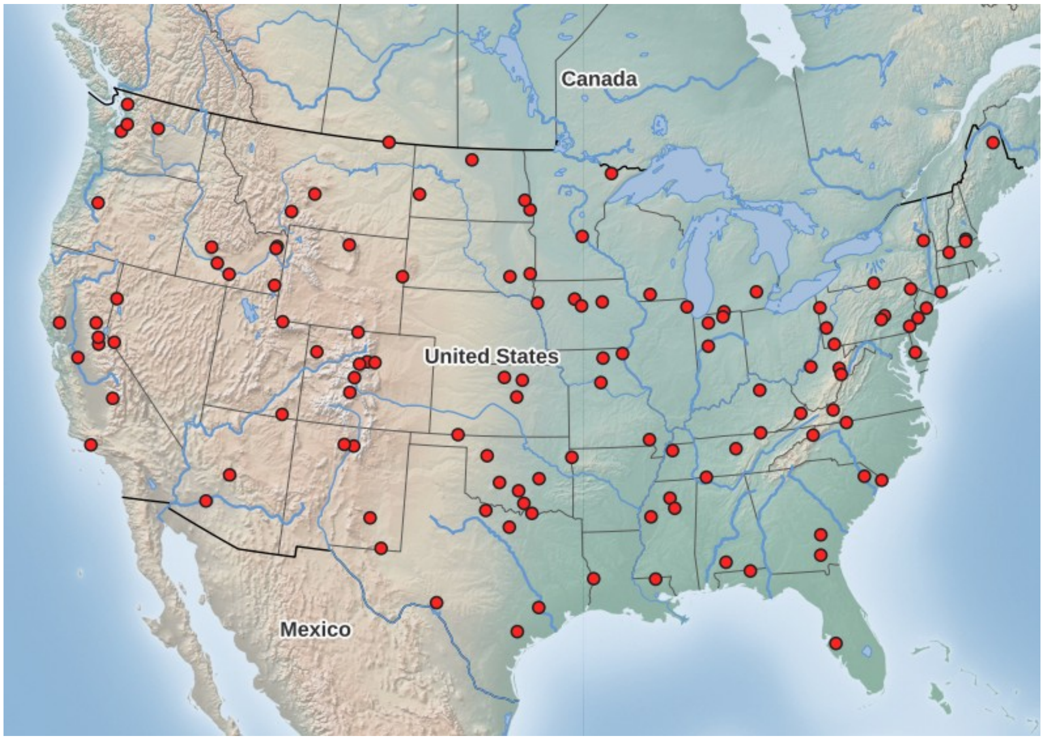


Figure 6: Sample of 120 USGS gages used as watershed outlets to test the accuracy of the watershed delineation software.

returns poor results, it is usually due to incorrect pour point snapping. For 14 of the 100 requested watersheds, ESRI's routine snapped the watershed outlet to the incorrect location, and returned a spurious watershed, often on a nearby tributary. This pour-point snapping error does not appear to be related to geography or watershed size. Nevertheless, when ESRI's software does snap the pour point to the correct river reach, the watershed boundaries are accurate: 81% of the ESRI watersheds had a $CAC > 0.9$. With our software, 83% of watersheds had a $CAC > 0.9$. This small experiment shows that delineator.py returns watershed boundaries that are accurate, rivaling those of industry-standard software for watersheds in the USA.

3.2. Benchmarking delineation speed

To evaluate the speed of the software, two benchmarking experiments were conducted. For these experiments, we compared the performance of delineator.py to two other widely-used raster-based methods of watershed

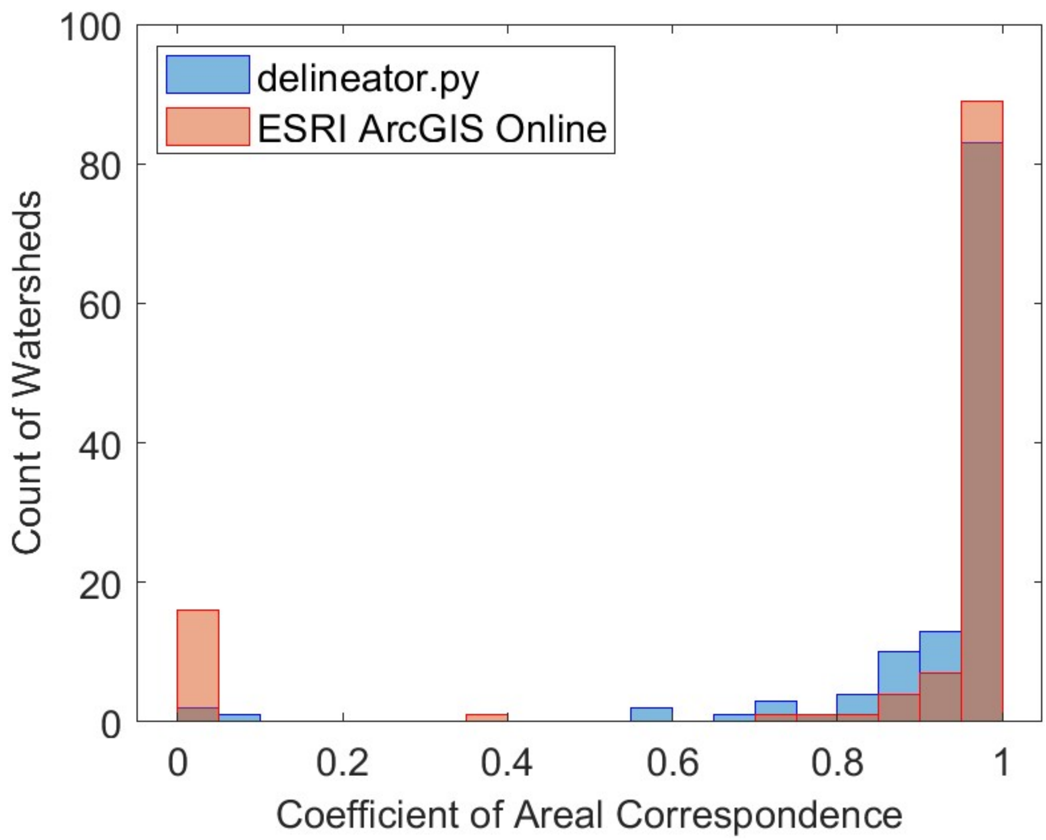


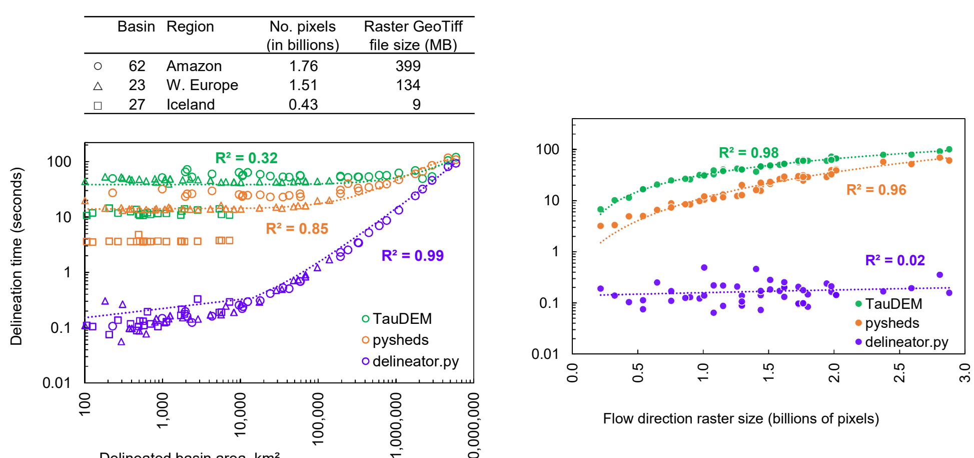
Figure 7: Accuracy of watershed delineation for 120 sampled locations in the continental United States.

delineation: TauDEM and pysheds, described briefly in the Section 3.1 Delineation was controlled by a set of scripts in Python 3.11, including timing of the processing.

For the first experiment, we sampled river locations from the gage catalog of the Global Runoff Data Center (GRDC), which contained 10,360 stations in 2023. We used a stratified sampling design, first eliminating gages with small watersheds (< 100 km^2) or unreported area. We divided the remaining gages into deciles by area and drew equal samples from each, ensuring watersheds ranging from 100 km^2 to 4 million km^2. We added the Amazon River outlet to test performance on the world's largest watershed (6 million km^2). Because delineation appears to take longer with larger input rasters, we also stratified by drawing samples from three megabasins of varying size: Iceland (small), Western Europe (medium), and the Amazon (large).

The results of the first benchmarking experiments are shown in Figure 8a. Overall, we see that our software, delineator.py, outperforms TauDEM and pysheds in terms of speed. For small- to mid-size watersheds (< 100,000 km²), delineator.py returned results in less than one second (mean response time of 0.26 seconds), while pysheds averaged 14 s and TauDEM 37 s. As the size of watersheds increases, the performance gap closes. For the world's largest watershed, the Amazon (6 million km²), the response times were: delineator.py: 93 s; pysheds: 108 s, TauDEM 120 s.

With each software package, we see a relationship between delineation time and the size of the watershed. In Figure 8a, a second-order polynomial trendline has been fitted to each data series. With delineator.py, the speed is highly dependent on the basin size ($R^2 = 0.99$). This has to do with the time required to merge and dissolve the vector unit catchments, which is a function of the number of polygons and the number of vertices to be processed. The results in Figure 8a also show that the size of the raster data files has an effect on speed. With TauDEM and pysheds, the processing times are shorter in the Iceland megabasin (squares on the plot), where the flow-direction raster is relatively small. Processing time is longer over the Amazon megabasin, where the raster data files are larger.



(a) Experiment 1: influence of watershed area

(b) Experiment 2: Influence of flow-direction raster data file size

Figure 8: Results of benchmarking experiments: delineation time for three watershed delineation software packages, including the one presented in this paper.

For a second benchmarking experiment, we tested the effect of input raster size more methodically, and holding the watershed size roughly equal (around 10 million pixels). To vary the input size, we used MERIT-Hydro raster data files for 52 of the 61 continental-scale megabasins shown in Figure 2 (those with flow accumulation $> 10^7$). In each megabasin, we searched the flow accumulation raster for a pixel with about 10^7 upstream cells. The location of this pixel becomes our test outlet for benchmarking. The results of this experiment are shown in Figure 8b. With both TauDEM and pysheds, we see a strong relationship between the size of the input raster and processing time. On the other hand, the delineation time with delineator.py does *not* vary with the input raster size. This is because the software only reads a small slice of the raster data, in a window the size of a single unit catchment, with an average area of 45 km^2. This greatly reduces the time to read raster data into memory. In contrast, the other software packages read the entire flow-direction raster file, which is a costly operation.

4. Discussion and future direction

4.1. Known limitations

In many areas, automated watershed delineation simply does not produce reasonable results. Flow paths may be ambiguous or indeterminate, or the digital terrain data may not accurately model real-world flow paths. In some instances, the deterministic 8-direction flow algorithm cannot always capture the complexity of real-world flow paths. This is another reason that we insist on the necessity of checking all results, or what some refer to as human-in-the-loop processing. Typical problem areas include:

- **Flat land** where the algorithm has trouble determining which direction water flows. Examples: Florida, the Netherlands, the Ganges-Brahmaputra Delta.
- **Deserts and arid areas** where there are fewer channels because there is little rainfall or runoff. Examples: North Africa, Central China, much of the American Southwest, the Orange River on the border of South Africa and Namibia.
- **Glaciers, ice, and tundra** environments where the surface is frozen all or part of the time. Such environments can be found across Arctic and subarctic regions. Examples: Iceland, Greenland, northern Canada, and northern Russia.

- **Karst, sand, or other permeable land** where water infiltrates into the ground. Here, it is not clear whether surface watersheds are a meaningful concept. Examples: the Yucatan Peninsula in Mexico, parts of the Deschutes River basin in Oregon, the Karst Plateau along the Italy-Slovenia border.
- **Urban areas** with many impervious surfaces and where flow paths are affected by curbs, sewers, or drains.
- Areas where **hydraulic infrastructure** such as irrigation canals or pipelines can reroute flow in unexpected ways.

In addition, there are some land areas where our scripts cannot delineate watersheds because MERIT-Hydro data is not available. This affects some islands, such as Hawaii and the Azores. However, there is data for the Canary Islands, Fiji, Tuvalu, the Galapagos, and many others.

4.2. The myth of "fully automated" delineation

No automated software for watershed delineation can entirely replace human judgment. Automated watershed delineation routines often make mistakes. The good news is that some of these mistakes can be readily corrected through manual intervention. Errors having to do with incorrect pour point snapping, for instance, can often be resolved by "nudging" or slightly moving the watershed outlet point. An experienced analyst can quickly identify and fix such issues. Software with a user-friendly display lets you review results and iterate quickly. To this end, the software described here creates an interactive map viewable in a web browser that lets the user easily review the results via a simple, familiar interface. Where the results are unsatisfactory, analysts can modify the outlet coordinates and re-run the delineation.

The Global Watersheds web app makes reviewing delineation results even faster and more intuitive. One particularly useful feature for improving consistency is the option to overlay MERIT-Basins river centerlines on the map. By placing outlet points directly on these centerlines, analysts can achieve more predictable and reliable delineation results.

4.3. Possible future enhancements

There are possible enhancements of this software that could make it even faster and more accurate. Accuracy is a function of the terrain data used as input. The hybrid watershed delineation algorithm described in this paper has been implemented using global datasets MERIT-Hydro (raster data of flow direction and flow accumulation) and MERIT-Basins (vector data for river reaches and unit catchments). The hybrid approach could be adapted for use with other datasets, for example, higher-resolution regional or national datasets, or other global datasets such as the forthcoming HydroSHEDS version 2 Lehner et al. (2022). The important thing is to have raster and vector data derived from the same source and consistent with one another. In our testing, mixing and matching different datasets gives poor results.

To increase the *speed* of watershed delineation with this method, the focus should be on the vector data processing. The slow step is merging and dissolving the upstream unit catchments. One promising approach is to preprocess the catchment data at various geographic scales, and to create a nested set of larger catchments, using an encoding system like Pfafstetter or Hydrologic Unit Codes. This is the approach we have taken with the Global Watersheds web app, where we use memoization to save the results of merging and dissolving unit catchments. As this is an expensive operation, the app does it once and subsequently reuses the result.

5. Conclusion

We described a method to delineate watersheds using a hybrid approach that combines the speed of vector-based methods with the accuracy of raster-based methods. While the hybrid method is not new, to the best of our knowledge, our implementation in Python is the first that is open source and based on free software. Our implementation uses state-of-the-art global datasets at 90 m resolution: MERIT-Hydro (Yamazaki et al., 2019) and MERIT-Basins (Lin et al., 2019). The software can be run on an ordinary desktop or laptop computer and can create watersheds of any size, including those of the largest river basins on Earth such as the Amazon, Mississippi, or Ob Rivers.

The methods described here are general and can be used with other terrain datasets aside from the ones described above. For example, it could be used with high-resolution local terrain data derived from LIDAR. Datasets require some preprocessing for use with this method, which can be time-consuming and require some expertise. It may not be worth doing this preprocessing if you are only looking to create a few watersheds. However, it is worthwhile when you want to create hundreds or thousands of watersheds, or to set up a watershed delineation service.

There are already a number of software packages and libraries for delineating watersheds. The software described here satisfies the needs of analysts performing large scale hydrologic investigations, as it can be used nearly anywhere (over land) on the globe and can readily delineate hundreds or thousands of watersheds. It is faster than previous methods, and can be run with modest hardware, including ordinary laptop computers. Nevertheless, as with any delineation software, results need to be carefully reviewed. Automated watershed delineation is often wrong, and there is a strong need for expert judgment.

An online demo of the software, the Global Watersheds web app (at https://mghydro.com/watersheds), allows you to perform watershed delineation without downloading or installing any software or data. The web app is the first of its kind, allowing users to delineate watersheds for nearly anywhere on Earth with a click of a mouse button. It is being used by a wide range of users in science, engineering, and education. It is our sincere hope that it will contribute to increased environmental awareness and help to raise watershed consciousness.

6. Software and data availability

The Python code described in this paper for performing watershed delineation is free and open source. The software is released under the MIT License. Additional code for the benchmarking experiments described in this paper can be found in the project's GitHub archive.

- Name of the software: **delineator.py**
- Developer: Matthew Heberger
- Contact information: matt@mghydro.com
- Hardware required: Laptop or desktop computer
- Program language: Python, HTML, JavaScript
- Year first available: 2022
- Cost: Free

- License: MIT License, https://spdx.org/licenses/MIT.html
- DOI: 10.5281/zenodo.7314287
- Software available from: https://github.com/mheberger/delineator
- Online Demo: https://mghydro.com/watersheds
- Program size: 69 MB

The software repository comes bundled with a sample of the input data covering Iceland so that users can get started quickly and test the software. Additional data is needed to perform watershed delineation in different regions (see Figure 2). As described in the article, these data are free; however, their licenses only allow for non-commercial use. For complete global coverage, the datasets (MERIT-Hydro and MERIT-Basins) have a total size of 73 GB.

7. Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

8. Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the author used Claude.ai to suggest revisions to an overly verbose draft and Notebook LM in order to help create the required graphical abstract. After using these services, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

9. Acknowledgments

The author wishes to thank Filipe Aires and Victor Pellet for fruitful conversations that led to the development of this software. My gratitude is also due to the many developers of free, open-source software used for this project.

This research was conducted independently prior to the author's affiliation with the San Francisco Estuary Institute. This work was performed as a part of the author's PhD dissertation research at Sorbonne University and the Paris Observatory and funded by ESTELLUS and a contract with the European Space Agency (contract #4000136793/21/I-DT-lr). We would like to thank Espen Volden for managing this project at ESA.

References

Addor, N., Do, H.X., Alvarez-Garreton, C., Coxon, G., Fowler, K., Mendoza, P.A., 2020. Large-sample hydrology: recent progress, guidelines for new datasets and grand challenges. Hydrological Sciences Journal 65, 712–725. URL: https://doi.org/10.1080/02626667.2019.1683182, doi:10.1080/02626667.2019.1683182.

Amatulli, G., Garcia Marquez, J., Sethi, T., Kiesel, J., Grigoropoulou, A., Üblacker, M.M., Shen, L.Q., Domisch, S., 2022. Hydrography90m: a new high-resolution global hydrographic dataset. Earth System Science Data 14, 4525–4550. URL: https://essd.copernicus.org/articles/14/4525/2022/, doi:10.5194/essd-14-4525-2022.

Bartos, M., Smith, T.J., Itati01, Debbout, R., Kraft, P., Huard, D., 2023. pysheds: 0.3.5. URL: https://zenodo.org/record/3822494, doi:10.5281/ZENODO.3822494.

Blodgett, D., 2022. The network linked data index geoprocessing with OGC API processes. USGS Water Data for the Nation blog. URL: https://waterdata.usgs.gov/blog/nldi-processes/.

Blöschl, G., Murugesu Sivapalan, Marty Anderies, José Carlos de Araújo, Berit Arheimer, Roy Brouwer, Ximing Cai, Christophe Cudennec, Alexia Fürnkranz-Prskawetz, Patricia Gober, Joyeeta Gupta, Anthony Jakeman, Shabbaz Khan, Wolfgang Kinzelbach, Ching Leong, Xin Li, Heng Liu, Daniel Peter Loucks, Aditi Mukherji, Alberto Montanari, John Ndiritu, Taikan Oki, Claudia Pahl-Wostl, Stefan Uhlenbrook, James L. Wescoat Jr., Jun Xia, Margreet Zwarteveen (Eds.), 2025. Coevolution and Prediction of Coupled Human-Water Systems. Elsevier, S.l. URL: https://www.sciencedirect.com/science/chapter/edited-volume/pii/B9780443417368200018, doi:10.1016/C2024-0-03476-5.

Carlson, K.A., Levin, H.K., Morris, A.L., Candela, S.G., Rivera, A.M.M., Huening, V.G., Fredericks, J.G., 2024. TDX-Hydro: Global High-Resolution Hydrography Derived from TanDEM-X. ESS Open Archive URL: https://www.authorea.com/users/783331/articles/939736-tdx-hydro-global-high-resolution-hydrography-derived-from-tandem-x.

Castronova, A.M., Goodall, J.L., 2014. A hierarchical network-based algorithm for multi-scale watershed delineation. Computers & Geosciences 72, 156–166. URL: https://linkinghub.elsevier.com/retrieve/pii/S0098300414001800, doi:10.1016/j.cageo.2014.07.014.

Cloke, H.L., Hannah, D.M., 2011. Large-scale hydrology-advances in understanding processes, dynamics and models from beyond river basin to global scale. Hydrological Processes 25, 991–1200. URL: https://www.cabdirect.org/cabdirect/abstract/20113099402, doi:10.1002/hyp.8059.

Djokic, D., Ye, Z., 1999. DEM preprocessing for efficient watershed delineation, in: Proceedings of the 1999 ESRI User Conference, ESRI, Inc.. p. Paper 676. URL: https://proceedings.esri.com/library/userconf/proc99/proceed/papers/pap676/p676.htm.

Do, H.X., Gudmundsson, L., Leonard, M., Westra, S., 2018. The Global Streamflow Indices and Metadata Archive (GSIM) – Part 1: The production of a daily streamflow archive and metadata. Earth System Science Data 10, 765–785. URL: https://essd.copernicus.org/articles/10/765/2018/, doi:10.5194/essd-10-765-2018.

Eilander, D., van Verseveld, W., Yamazaki, D., Weerts, A., Winsemius, H.C., Ward, P.J., 2021. A hydrography upscaling method for scale-invariant parametrization of distributed hydrological models. Hydrology and Earth System Sciences 25, 5287–5313. URL: https://hess.copernicus.org/articles/25/5287/2021/, doi:10.5194/hess-25-5287-2021.

ESRI, . Create Watersheds. URL: https://doc.arcgis.com/en/arcgis-online/analyze/create-watersheds-mv.htm.

Freitas, H.R.d.A., Freitas, C.d.C., Rosim, S., Oliveira, J.R.d.F., 2016. Drainage networks and watersheds delineation derived from TINbased digital elevation models. Computers & Geosciences 92, 21–37. URL: https://www.sciencedirect.com/science/article/pii/S0098300416300954, doi:10.1016/j.cageo.2016.04.003.

Haag, S., Schwartz, D., Shakibajahromi, B., Campagna, M., Shokoufandeh, A., 2020. A fast algorithm to delineate watershed boundaries for simple geometries. Environmental Modelling & Software 134, 104842. URL: https://www.sciencedirect.com/science/article/pii/S1364815220308999, doi:10.1016/j.envsoft.2020.104842.

Harrigan, S., Zsoter, E., Alfieri, L., Prudhomme, C., Salamon, P., Wetterhall, F., Barnard, C., Cloke, H., Pappenberger, F., 2020. GloFAS-ERA5 operational global river discharge reanalysis 1979–present. Earth System Science Data 12, 2043–2060. URL: https://essd.copernicus.org/articles/12/2043/2020/, doi:10.5194/essd-12-2043-2020.

Horizon Systems, 2010. NHDPlus basin delineation tool. National Hydrography Dataset Plus. URL: https://nhdplus.com/NHDPlus/NHDPlusV1_tools.php.

Hrachowitz, M., Savenije, H., Blöschl, G., McDonnell, J., Sivapalan, M., Pomeroy, J., Arheimer, B., Blume, T., Clark, M., Ehret, U., Fenicia, F., Freer, J., Gelfan, A., Gupta, H., Hughes, D., Hut, R., Montanari, A., Pande, S., Tetzlaff, D., Troch, P., Uhlenbrook, S., Wagener, T., Winsemius, H., Woods, R., Zehe, E., Cudennec, C., 2013. A decade of Predictions in Ungauged Basins (PUB)—a review. Hydrological Sciences Journal 58, 1198–1255. URL: https://www.tandfonline.com/doi/full/10.1080/02626667.2013.803183, doi:10.1080/02626667.2013.803183.

Johnson, L.E., 2009. Geographic Information Systems in Water Resources Engineering. CRC Press.

Johnston, C., T.G. Dewald, T.R. Bondelid, B.B. Worstell, L.D. McKay, Alan Rea, R.B. Moore, J.L. Goodall, 2009. Evaluation of catchment delineation methods for the medium-resolution national hydrography dataset. URL: http://pubs.usgs.gov/sir/2009/5233/.

Jones, K., Niknami, L., Buto, S., Decker, D., 2022. Federal standards and procedures for the national watershed boundary dataset (WBD) (5 ed.). URL: https://pubs.usgs.gov/tm/11/a3/.

Jones, N.L., Wright, S.G., Maidment, D.R., 1990. Watershed Delineation with Triangle-Based Terrain Models. Journal of Hydraulic Engineering 116, 1232–1251. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9429%281990%29116%3A10%281232%29, doi:10.1061/(ASCE)0733-9429(1990)116:10(1232).

Karimipour, F., Ghandehari, M., Ledoux, H., 2013. Watershed delineation from the medial axis of river networks. Computers & Geosciences 59, 132–147. URL: https://www.sciencedirect.com/science/article/pii/S0098300413001684, doi:10.1016/j.cageo.2013.06.004.

Kauffeldt, A., Halldin, S., Rodhe, A., Xu, C.Y., Westerberg, I.K., 2013. Disinformative data in large-scale hydrological modelling. Hydrology and Earth System Sciences 17, 2845–2857. URL: https://hess.copernicus.org/articles/17/2845/2013/, doi:10.5194/hess-17-2845-2013.

Kotyra, B., 2023. High-performance watershed delineation algorithm for GPU using CUDA and OpenMP. Environmental Modelling & Software 160, 105613. URL: https://www.sciencedirect.com/science/article/pii/S1364815222003139, doi:10.1016/j.envsoft.2022.105613.

Kumar, R., Jingar, V., Jindal, A., Seth, A., 2025. GPU-Accelerated Hydrology Algorithms for On-Prem Computation: Flow Accumulation, Drainage Lines, Watershed Delineation, Runoff Simulation, in: Proceedings of the 2nd ACM SIGPLAN International Workshop on Programming for the Planet, Association for Computing Machinery, New York, NY, USA. pp. 31–46. URL: https://dl.acm.org/doi/10.1145/3759536.3763805, doi:10.1145/3759536.3763805.

Lehner, B., 2022. HydroSHEDS technical documentation. URL: https://data.hydrosheds.org/file/technical-documentation/HydroSHEDS_TechDoc_v1_4.pdf.

Lehner, B., A. Roth, M. Huber, M. Anand, M. Thieme, 2022. A sharper look at the world's rivers and catchments. Eos, Transactions American Geophysical Union 103. URL: http://eos.org/science-updates/a-sharper-look-at-the-worlds-rivers-and-catchments, doi:10.1029/2022EO220167.

Lin, P., Pan, M., Beck, H.E., Yang, Y., Yamazaki, D., Frasson, R., David, C.H., Durand, M., Pavelsky, T.M., Allen, G.H., Gleason, C.J., Wood, E.F., 2019. Global reconstruction of naturalized river flows at 2.94 million reaches. Water Resources Research 55, 6499–6516. URL: https://onlinelibrary.wiley.com/doi/abs/10.1029/2019WR025287, doi:10.1029/2019WR025287.

Lindsay, J., 2014. Answer to "delineating large scale watershed?". Geographic Information Systems Stack Exchange. URL: https://gis.stackexchange.com/a/116214/18465.

Lindsay, J.B., Rothwell, J.J., Davies, H., 2008. Mapping outlet points used for watershed delineation onto DEM-derived stream networks. Water Resources Research 44. URL: http://onlinelibrary.wiley.com/doi/abs/10.1029/2007WR006507, doi:10.1029/2007WR006507.

Maidment, D.R., Djokic, D., 2000. Hydrologic and Hydraulic Modeling Support: With Geographic Information Systems. ESRI, Inc.

Mee-Jeong, P., Hyun, H., Tae-Gon, K., Kyo, S., Jeong-Jae, L., 2006. Watershed Delineation Algorithm Using Kruskal's Algorithm and Triangulated Irregular Network. Journal of The Korean Society of Agricultural Engineers 48, 3–12. URL: https://koreascience.or.kr/article/JAKO200633242474085.page, doi:10.5389/KSAE.2006.48.4.003. publisher: The Korean Society of Agricultural Engineers.

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, S., Hassidim, A., Klotz, D., Kratzert, F., Metzger, A., Nevo, S., Pappenberger, F., Prudhomme, C., Shalev, G., Shenzis, S., Tekalign, T.Y., Weitzner, D., Matias, Y., 2024. Global prediction of extreme floods in ungauged watersheds. Nature 627, 559–563. URL: https://www.nature.com/articles/s41586-024-07145-1, doi:10.1038/s41586-024-07145-1. publisher: Nature Publishing Group.

O'Callaghan, J.F., Mark, D.M., 1984. The extraction of drainage networks from digital elevation data. Computer Vision, Graphics, and Image Processing 28, 323–344. doi:10.1016/S0734-189X(84)80011-0.

Pan, M., Sahoo, A.K., Troy, T.J., Vinukollu, R.K., Sheffield, J., Wood, E.F., 2012. Multisource estimation of long-term terrestrial water budget for major global river basins. Journal of Climate 25, 3191–3206. doi:10.1175/JCLI-D-11-00300.1.

Parsons, J.J., 1985. On "bioregionalism"and "watershed consciousness"*. The Professional Geographer 37, 1–6. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0033-0124.1985.00001.x, doi:10.1111/j.0033-0124.1985.00001.x.

Rosebrock, A., 2016. Intersection over union (IoU) for object detection. PyImageSearch. URL: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

Schwanghart, W., Scherler, D., 2014. Short Communication: TopoToolbox 2 – MATLAB-based software for topographic analysis and modeling in Earth surface sciences. Earth Surface Dynamics 2, 1–7. URL: https://esurf.copernicus.org/articles/2/1/2014/esurf-2-1-2014.html, doi:10.5194/esurf-2-1-2014. publisher: Copernicus GmbH.

Steeves, P., 2003. NHD watershed: Tools and applications. URL: https://proceedings.esri.com/library/userconf/proc03/abstracts/a0910.pdf. ESRI User Conference.

Strahler, A.N., 1957. Quantitative analysis of watershed geomorphology. Eos, Transactions American Geophysical Union 38, 913–920. URL: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/TR038i006p00913, doi:10.1029/TR038i006p00913.

Tarboton, D., 2016. Terrain analysis using digital elevation models (TauDEM). URL: https://hydrology.usu.edu/taudem/taudem5/.

Tarboton, D.G., 1997. A new method for the determination of flow directions and upslope areas in grid digital elevation models. Water Resources Research 33, 309–319. doi:10.1029/96WR03137.

Taylor, P.J., 1977. Quantitative Methods in Geography: An Introduction to Spatial Analysis. Houghton Mifflin. URL: https://archive.org/details/quantitativemeth0000tayl.

USGS, 2021. StreamStats. U.S. Geological Survey. URL: https://streamstats.usgs.gov/ss/.

USGS, 2022. Watershed Boundary Dataset. URL: https://www.usgs.gov/national-hydrography/watershed-boundary-dataset.

Verdin, K.L., 2017. Dataset: Hydrologic derivatives for modeling and applications (HDMA). URL: https://www.sciencebase.gov/catalog/item/5910def6e4b0e541a03ac98c, doi:10.5066/F7S180ZP.

Wikipedia contributors, 2024. Jaccard index. Wikipedia. URL: https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=1254621122. page Version ID: 1254621122.

Xie, J., Liu, X., Bai, P., Liu, C., 2022. Rapid Watershed Delineation Using an Automatic Outlet Relocation Algorithm. Water Resources Research 58. URL: http://onlinelibrary.wiley.com/doi/abs/10.1029/2021WR031129, doi:10.1029/2021WR031129.

Yamazaki, D., Ikeshima, D., Sosa, J., Bates, P.D., Allen, G.H., Pavelsky, T.M., 2019. MERIT hydro: A high-resolution global hydrography map based on latest topography dataset. Water Resources Research 55, 5053–5073. URL: https://onlinelibrary.wiley.com/doi/10.1029/2019WR024873, doi:10.1029/2019WR024873.

Zhao, F., Xiong, L., Wang, H., Luo, L., Na, J., Li, S., Tang, G., 2025. Parallel algorithm of watershed delineation by considering the topological relationship into information organisation. Environmental Modelling & Software 194, 106706. URL: https://www.sciencedirect.com/science/article/pii/S1364815225003901, doi:10.1016/j.envsoft.2025.106706.