RichDEM: High-Performance Terrain Analysis

Richard Barnes
Energy & Resources Group
UC Berkeley
Berkeley, USA
richard.barnes@berkeley.edu

Abstract—**To answer geomorphological questions at unprecedented spatial and temporal scales, we need to (a) parse terabyte-scale datasets (DEMs), (b) perform millions of model realizations to pinpoint the parameters which govern landscape evolution, and (c) do so with statistical rigor, which may require thousands of additional realizations. A core set of operations underpin many geomorphic models. These include determination of terrain attributes such as slope and curvature; flow routing; depression flooding and breaching; flat resolution; and flow accumulation. Here, I present RichDEM, a high-performance C++ library and set of wrappers for performing these operations. The library incorporates a number of options for performing each operation and makes full use of modern high-performance capabilities. The library can scale to process DEMs of over one trillion cells and operates effectively on laptops or supercomputers.**

I. Introduction

Raster digital elevation models (DEMs) express the elevation of terrain above or below a given baseline and store data in a rectangular array of floating-point or integer values. They are widely used in geospatial analysis for estimating a region’s hydrologic and geomorphic properties, including soil moisture, terrain stability, erosive potential, rainfall retention, and stream power. In geomorphological modeling, DEMs are often evolved in time, which means performance is critical.

Both analysis and modeling often require that the DEM be depression-free. A depression-free DEM is one in which every cell has a monotonically-descending flow path to the edge of the DEM; this guarantees that every cell’s flow is able to reach the edge of the DEM. Analyses also often require that each cell have an associated flow accumulation value (otherwise known as upslope area, contributing area, and upslope contributing area). Informally, if there were a rain storm, flow accumulation is directly proportional to the total amount of water which would pass through a cell as it flowed downhill from higher elevations.

DEMs have increased in resolution from 30–90m in the recent past to sub-meter resolutions becoming available today. This has led to increased data sizes: current DEMs are on the order of gigabytes and increasing, with billions of cells. Even low-resolution DEMs may cover large areas: 30m Shuttle Radar Topography Mission (SRTM) elevation data has been released for 80% of Earth's landmass. [5] While computer processing and memory performance have increased appreciably, development of algorithms suited to efficiently manipulating large, continent-scale DEMs is on-going.

This paper presents RichDEM, a library designed to help fill this gap. RichDEM is available online at *https://github.com/r-barnes/richdem*.

II. Design

RichDEM is designed to accelerate any form of raster-based analysis or modeling, regardless of the language in which it is written. It is built as an open-source C++ header-only library with no hard dependencies. If your data is a row-major array, it can be passed to RichDEM in one line of code.

A Python wrapper for RichDEM is already available at *https://pypi.org/project/richdem*. This includes a set of command-line tools. Python's distribution and build system makes installation easy on any OS and allows integration with GIS software such as ArcGIS, QGIS, and GRASS.

RichDEM follows industry-standard best practices. Its code is modular, well-commented, includes unit tests, and has OpenMP and OpenACC directives for accessing many-core and GPU accelerators. It provides machine-parsable output and progress bars. RichDEM uses Travis-CI and ReadTheDocs for continuous integration testing and documentation generation. No other landscape evolution package, including FastScape, Whitebox GIS, and the widely-used TauDEM, offers all these benefits.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27099v1 | CC BY 4.0 Open Access | rec: 8 Aug 2018, publ: 8 Aug 2018

RichDEM's library-plus-wrappers design means that RichDEM can easily be incorporated into almost any language offering the possibility of a single codebase that serves many users. Similarly, the internal simplicity and extensibility of the library means that it is easy for others to alter and contribute to it. I believe that designs such as the foregoing are needed to drive progress in geomorphometry.

An overview of RichDEM's performance is given below. The references cited describe the algorithms textually, in pseudocode, with step-by-step figures, and via associated reference source code. They also include further performance comparisons.

III. Serial Performance

The slowest operations in geomorphological modeling are depression-filling and flat resolution. RichDEM uses improved algorithms to overcome this.

Depression-filling in RichDEM leverages the Priority-Flood [4] algorithm to operate in $O(N)$ time for integer data and $O(N \log N)$ time for floating-point data, lower than TauDEM's $O(N^{1.5})$ and ESRI's $O(N^2)$ algorithms. In testing, RichDEM gave identical results to TauDEM, but ran 6x faster on one processor than TauDEM did with 11 processors, a 67x reduction in compute time. [4]

Similarly, RichDEM's flat resolution algorithm operates in $O(N)$ time versus TauDEM's $O(N^2)$ time. In testing, RichDEM gave identical results to TauDEM, but took 30s on one processor instead of 53 minutes on 16 processors: a 107x reduction in wall-time and 1,763x reduction in compute time. [1]

IV. Parallel Performance

RichDEM also works in parallel with MPI. Previous algorithms either swapped data continuously to and from memory or used continuous communication between many nodes to simulate a single, large DEM loaded fully into RAM. In contrast, RichDEM can fully decompose its calculations across a tiled DEM, allowing for a divide and conquer approach. [2,3]

The approach ensures that, regardless of a DEM's size each DEM cell is accessed a only fixed number of times and only a fixed number of low-cost communication events are required. These properties ensure that the algorithms can run effectively on laptops or supercomputers. The algorithms are optimal in terms of time, space, and communication complexity.

Depression-filling on the 120 billion cell 10m NED dataset took 48 minutes using 48 cores [2] and flow accumulation took 15 minutes [3]. In comparison against TauDEM on a 1.6 billion cell dataset (near the limits of TauDEM's capabilities), RichDEM's depression-filling algorithm ran 6.3x faster, used 19x less bandwidth, 70x less communication time, and 7x less RAM. [2]

Wall-times for RichDEM's parallel algorithms scaled linearly with the size of the tested datasets. The largest DEM tested was a 3.34 TB downscaling of the SRTM global dataset consisting of 14,297 tiles containing two trillion (10^{12}) cells. This is equivalent in size to the entire conterminous United States at 3m resolution and three orders of magnitude larger than any DEM previously tested in the literature. [2]

With 48 cores, depression-filling on this dataset took 4.8 hours. With 14,000 cores, as might be available on a supercomputer, processing would have taken about three minutes. On a laptop, processing would have taken 9.3 days, but would have required only 13.5 GB of RAM. [2] This ability to scale both up and down makes the library “algorithmically democratic”: users of any resource level can benefit from using it.

V. Conclusions

The library described here, RichDEM, offers several design advantages over existing terrain analysis and geomorphological modeling software, as well as performance benefits. The library can be obtained from *https://github.com/r-barnes/richdem*. Documentation is available at *https://richdem.readthedocs.io/*. A Python package is available at *https://pypi.org/project/richdem/*.

Acknowledgment

This work was supported by the National Science Foundation's Graduate Research Fellowship and by the Department of Energy's Computational Science Graduate Fellowship (Grant No. DE-FG02-97ER25308). Early-stage development utilized supercomputing time and data storage provided by the University of Minnesota Supercomputing Institute. Empirical tests were performed on XSEDE's Comet supercomputer which is supported by the National Science Foundation (Grant No. ACI-1053575) [6].

References

[1] Barnes, R., Lehman, C., Mulla, D., 2014. An efficient assignment of drainage direction over flat surfaces in raster digital elevation models. Comput. Geosci. 62, 128–135. doi: 10.1016/j.cageo.2013.01.009.

[2] Barnes, R., 2016. Parallel priority-flood depression filling for trillion cell digital elevation models on desktops or clusters. Comput. Geosci. 96, 56--68. doi: 10.1016/j.cageo.2016.07.001.

[3] Barnes, R., 2017. Parallel non-divergent flow accumulation for trillion cell digital elevation models on desktops or clusters. Environemtanl Modelling & Software. 92, 202–212. doi: 10.1016/j.envsoft.2017.02.022.

[4] Barnes, R., Lehman, C., Mulla, D., 2014. Priority-flood: an optimal depression-filling and watershed-labeling algorithm for digital elevation models. Comput. Geosci. 62, 117–127. doi: 10.1016/j.cageo.2013.04.024.

[5] Farr, T.G., Rosen, P.A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L., Seal, D., Shaffer, S., Shimada, J., Umland, J., Werner, M., Oskin, M., Burbank, D., Alsdorf, D., 2007. The shuttle radar topography mission. Rev. Geophys. 45 (2). doi: 10.1029/2005RG000183.

[6] Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G.D., et al., 2014. XSEDE: accelerating scientific discovery. Comput. Sci. Eng. 16 (5), 62--74.