Flow on Noisy Terrains: An Experimental Evaluation

Mark de Berg Herman Haverkort Constantinos P. Tsirogiannis*

Abstract

We investigate experimentally the changes of drainage structures on triangulated terrains when noise is applied. Given real world data sets, we examine two techniques for identifying watersheds on terrains created using the same data set yet under multiple noise instances. For the needs of our experiments we have developed a robust implementation of an algorithm that computes drainage structures on triangulated terrains under the assumption that water always follows the direction of steepest descent. We provide here also a description of our implementation which is based on CGAL.

1 Introduction

Analysing the flow of water on drainage basins is important for flood prediction, understanding landscape erosion, and other forms of hydrological analysis. To be able to automate such analyses, one needs a computer model of the terrain forming the drainage basin under study and a model for the way in which water flows on the terrain. There are two main terrain models: the digital elevation model (DEM), which is based on a regular grid, and the triangulated irregular network (TIN), which is a 2D triangulation where every vertex is associated with a height value.

The most natural model for water flow is that water follows the direction of steepest descent (DSD) on a surface. As water flows on the surface of a terrain $\mathcal{T}$ following the DSD it accumulates on the local minima of $\mathcal{T}$. For a local minimum p on $\mathcal{T}$, the *watershed* of p is the set of all points on $\mathcal{T}$ from which water flows down to p as it follows the DSD. Since a DEM is a discrete, non-continuous surface, it is not completely clear how to translate the DSD model to DEMs. For TINs, on the other hand, the DSD model can be directly applied (although here one also has to decide how to define flow on flat areas). Nevertheless, existing software for flow modeling often makes use of DEMs, because of the ease of implementing algorithms on DEMs. Implementing the DSD model on a TIN is non-trivial because of robustness problems; indeed, existing software packages for flow computation on TINs do not follow the exact DSD model, but discretise the flow (e.g. by only allowing water to move between a fixed set of points like the vertices of the TIN or the barycenters of the triangles). This poses the question of how reliable the output of such software packages is since they approximate water flow in a coarse way. The only exception is the implementation of Liu *et al.* [8]. However, their software uses fixed precision arithmetic which leads to round-off errors during computation. As they conclude, this eventually produces inconsistencies in the output.

Moreover, an important factor that affects the output of a flow model is the noise that appears in the coordinates of the point set used by a terrain model. Noise comes as a result of sampling with limited accuracy equipment, data conversion between different terrain representation models and calculations under fixed precision arithmetic. For a given TIN, noise can be practically represented at each vertex as an interval of possible height values while the xy-coordinates of the vertex are considered to be fixed. It is therefore interesting to evaluate to what extent the structure of the watersheds of a TIN change if we perturb the height values of its vertices.

Our results. In the current paper, we present a robust C++ software package to compute drainage structures on TINs. Our implementation follows the formal flow model described by de Berg *et al.* [2] and Yu *et al.* [10] , which is based on the exact DSD model. Our package is based on the Computational Geometry Algorithms Library (CGAL), an open source software library for geometric computations that supports the use of exact arithmetic [4]. Thus it is the first ever implementation of a flow model on TINs that follows strictly the DSD assumption and at the same time uses exact arithmetic.

We have used our implementation on TINs that represent real world geographical areas to perform experiments involving perturbations on the height values of their vertex set. We assess which percentage of the area of such a TIN is covered by watersheds of shallow local minima and we test how this percentage changes as the size of the perturbation interval grows. We also investigate how much of the area of indvidual watersheds is preserved between different noise instances. For this reason, we examine two techniques for matching watersheds between TINs that are created when different noise instances are applied to the same terrain data set. Such techniques can be as well

*Dept. of Mathematics and Computer Science, Eindhoven University of Technology, mdberg@win.tue.nl, cs.herman@haverkort.net, ctsirogi@win.tue.nl

useful for applications of *data conflation*; that is for identifying the same watershed(s) between TINs that were generated from different data sources yet representing the same geographical area.

2 Experiments

2.1 Software Implementation

To conduct our experiments we have developed a software package that computes drainage structures on triangulated terrains. In particular, we have implemented algorithms for computing paths of steepest descent/ascent, river networks [2], watershed maps and strip maps [10] on TINs. Our software was implemented in C++ using CGAL which provided basic geometric objects, predicates and number types as building blocks for our needs.

The implemented algorithms that compute watershed maps and strip maps take as an input argument an object that is a model of the `CGAL::Triangulation_euclidean_traits_xy_3`; this is the appropriate triangulation type in CGAL to represent a terrain. The triangulation object is then converted to an object of an augmented version of the `CGAL::Polyhedron_3` type. This is because boundaries between watersheds and strips may not always coincide with the triangulation edges and can as well extend through triangle interiors. The output of the watershed map algorithm is an object of the augmented `CGAL::Polyhedron_3` type where each facet, edge and vertex on the surface is tagged with the watershed that it belongs to. Also each of the watersheds in the output is provided as a model of the graph concept of the BOOST Graph Library [6].

We have also designed a traits class with the name `Drainage_traits_2` that provides all the predicates and geometric object types that are essential for the main algorithms. This traits class is a template class with two parameters; a version of the `CGAL::Triangulation_euclidean_traits_xy_3` model and a model of a CGAL linear geometric kernel [3]. The linear kernel type parameter may be instatiated with any number type, yet in practice the algorithms of this package will not execute properly unless an arbitrary precision number type is used such as `gmpq`. This is due to the fact that fixed precision arithmetic is not enough to represent the coordinates of the intersection points between a path of steepest descent/ascent and terrain edge interiors [8]. Because of this, we have run our experiments using exclusively as a kernel parameter the `CGAL::Exact_predicates_exact_constructions_kernel`. We intend to show in our future work how the bit-size of the computed structures affects in practice the applicability of the described flow model.

We have ran our implementation in a Linux Ubuntu operating system version 9.10 using the GNU g++ version 4.4 compiler and CGAL version 3.6.1.

2.2 Input Data Specifications and Noise Model

The TINs that we have used for our experiments are constructed from DEM data sets that are publicly available from the U.S. Geological Survey (USGS) server [9]. The data files hosted in this server are of the USGS *DEM* digital format. Each file stores a grid terrain of 1201 × 1201 cells where the dimension of each cell is 90 meters and the elevation values of the cells are integers.

We have created three TINs, each derived from a different DEM data set. Each of these TINs consists of 50000 vertices and roughly 100000 triangles. The vertex set of each TIN was created by sampling a set of points on the surface of one of the DEMs. A Delaunay triangulation was then constructed on the *xy*-projection of the selected point set. Thus the final TIN can be seen as the result of lifting the vertices of this triangulation up to their original elevation values. The names of the TIN data sets and the range of elevations among the vertices of each TIN are listed in Table 1.

Name	Elevation Range (in meters)
`aztec`	[1704, 3523]
`marion`	[213, 457]
`carlsbad`	[861,2256]

Table 1: The names and the elevations ranges of the TIN data sets that we use in our experiments.

Perturbations on the elevations of the TIN vertices were conducted in the following way. To the elevation of each TIN vertex we added a value picked uniformly at random from an interval of the form $[-\eta, \eta]$. The value of η used in each experiment is explicitly stated in the description of the experiment.

3 Experiments

3.1 Area covered by small watersheds

In the first experiment we examine how large is the part of the terrain covered by watersheds of spurious minima and how this changes as noise is added to the elevations of the vertices. There exist several criteria in the GIS literature to define whether a local minimum is a spurious minimum or not e.g. topological persistence [5]. In our experiment we consider a local minimum to be spurious if the xy-projection of its watershed covers less than one thousandth of the xy-projection of the total terrain area. We consider all watersheds that are larger than this threshold as *sufficiently large*. Deciding which threshold value is appropriate to distinguish which watersheds are sufficiently large is a debatable issue and depends on the scale that is used in the analysis.

We have conducted the experiment in the following way; for each of the original TIN data sets we have constructed six new TINs by perturbing the elevation values of their vertices. We use a perturbation interval $[-\eta, \eta]$ of different size for each new TIN instance by choosing the value of η from the range $\{0.5m, 1.0m, \ldots, 3.0m\}$. We compute the watershed map for each new TIN and we measure:

- the number of watersheds that belong to spurious minima.
- the percentage of the total area that is covered by those watersheds on the xy-projection of the terrain.

The results of this experiment are summarized in Table 2. For the `aztec` data set we see that roughly 50% of the terrain is covered by relatively small watersheds and this percentage reaches approximately 55% as the max size of the absolute perturbation increases to 3 meters. For the `carlsbad` data set this percentage reaches up to 72% . Worse than that, the area covered by small watersheds on the `marion` data set rises dramatically even when the max absolute noise is half a meter. This is due to the fact that the elevation range of this data set is quite small, about 200 meters. Thus the original TIN consists of large flat areas that break into small watersheds even with small perturbations. The number of small watersheds rises substantially for all of the data sets with respect to the maximum noise and in the `marion` more than doubles when the max absolute noise reaches 3 meters.

3.2 Watershed area maintained among different noise instances

Next we investigate to what extent large watersheds on a TIN occupy the same area after perturbations of the terrain vertices. We proceed with the following definition.

Consider a TIN $\mathcal{T}_1$ and let $v_1 \in \mathcal{T}_1$ be a vertex that is also a local minimum. Let $\mathbb{T} = \{\mathcal{T}_2, \mathcal{T}_3, \ldots, \mathcal{T}_m\}$ be a set of TINs such that each $\mathcal{T}_i \in \mathbb{T}$ can be induced by modifying the height values of the vertices of $\mathcal{T}_1$. Assume that for every $\mathcal{T}_i \in \mathbb{T}$ the vertex v_i that corresponds to (has the same xy-coordinates as) v_1 is also a local minimum of $\mathcal{T}_i$. Let w^i_{xy} be the xy-projection of the watershed of v_i for every $1 \leq i \leq m$. We call the *core watershed* of v_1 the intersection $\cap_{1 \leq i \leq m} w^i_{xy}$.

Based on this definition of core watersheds we perform our next experiment as follows. Given one of our TIN data sets we construct three new instances by perturbing the vertices of the original TIN on the z-coordinate. The noise interval that we use to derive each of the three new instances is $[-1.0m, 1.0m]$. We compute the watershed map on the three new TINs as well as on the original TIN. We then distinguish on the original TIN all local minima whose watersheds cover individually more than one thousandth of the terrain xy-area. We then compute the core watersheds of these minima and calculate the percentage of the terrain xy-area that these core watersheds cover.

The results of this experiment (not displayed in this version of the paper) show that only roughly 5% of the total terrain xy-area is indeed covered by the core watersheds. This is partly due to the fact that not all vertices that constitute local minima on the original terrain appear as local minima also on the new terrain instances. Consider a vertex v of locally minimum elevation that has a very small height difference from its neighbours on the triangulation. Then even a slight perturbation among all terrain vertices may cause another nearby vertex to become a local minimum instead of v while no large changes may be induced in the structure of the surrounding drainage area.

Being aware of this issue we revise our experiment by adopting the following approach. Let $\mathbb{T} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4\}$ be the four TINs that we have created, that is the original TIN data set and the three new instances acquired by applying noise on the original TIN. Let $\mathcal{W}_i$ be the set of all sufficiently large watersheds on the surface of $\mathcal{T}_i \in \mathbb{T}$. Let w be a watershed in $\mathcal{W}_i$. Our intention is to match w with exactly one large watershed from *each* other instance $\mathcal{W}_j, j \neq i$ such that the matched watersheds have the largest possible intersection on the xy-plane.

More formally, we consider a hypergraph where each graph-node corresponds to a watershed in $\cup_{1 \leq i \leq 4} \mathcal{W}_i$. Each hyperedge is incident to four nodes, each node corresponding to a watershed from a different TIN instance. Also every hyperedge has a positive weight, equal to the area of the intersection of (the xy-projections of) the watersheds corresponding to the incident nodes. Our goal is to compute the maximum weight 4D-matching on the described hypergraph. This graph problem is known to be NP-hard, yet there exists a 3-approximation algorithm based on local search [1]. We have implemented this local search algorithm for the needs of our experiments and tested it on the described TINs. The results obtained from the local search matching algorithm are listed on Table 3.

Data set	# of Matched Watersheds	Area %
`aztec`	203	22.11
`marion`	77	5.26
`carlsbad`	174	15.24

Table 3: Results of matching watersheds using the 4D matching approximation algorithm.

Data set: Aztec			
Noise Interval (in meters)	# of Small Watersheds	# of All Watersheds	Total Area of Small Watersheds %
$[0, 0]$	2032	2348	48.02
$[-0.5, 0.5]$	1735	2061	48.67
$[-1.0, 1.0]$	1797	2119	49.20
$[-1.5, 1.5]$	1919	2230	52.85
$[-2.0, 2.0]$	1984	2292	53.54
$[-2.5, 2.5]$	2071	2372	56.45
$[-3.0, 3.0]$	2089	2395	55.62

Data set: Marion			
Noise Interval (in meters)	# of Small Watersheds	# of All Watersheds	Total Area of Small Watersheds %
$[0, 0]$	2041	2233	38.80
$[-0.5, 0.5]$	2972	3176	68.23
$[-1.0, 1.0]$	3301	3509	71.60
$[-1.5, 1.5]$	3570	3755	75.82
$[-2.0, 2.0]$	3805	3972	79.34
$[-2.5, 2.5]$	4069	4206	83.37
$[-3.0, 3.0]$	4350	4458	86.95

Data set: Carlsbad			
Noise Interval (in meters)	# of Small Watersheds	# of All Watersheds	Total Area of Small Watersheds %
$[0, 0]$	2032	2787	59.38
$[-0.5, 0.5]$	1915	2212	52.72
$[-1.0, 1.0]$	2085	2374	56.43
$[-1.5, 1.5]$	2315	2594	59.86
$[-2.0, 2.0]$	2444	2843	64.21
$[-2.5, 2.5]$	2862	3096	68.17
$[-3.0, 3.0]$	3058	3267	72.83

Table 2: The number of small watersheds and percentage of xy-area they cover on the TIN data sets

4 Conclusions and Future work

As expected, a large part of the area in the acquired data sets is covered by small watersheds. We further show that the application of small perturbations magnifies substantially this phenomenon especially on terrains with relatively small elevation range. Sufficiently large watersheds maintain only a small part of their total area when applying small perturbations on the TIN vertices. We intend to evaluate how much these results can improve with the use of a technique that floods spurious local minima. We also opt to conduct experiments with data sets of different resolution and compare the output with the current results.

References

[1] E.M. Arkin and R. Hassin On Local Search for Weighted k-Set Packing. *Mathematics of Operations Research*, 23:640–648, 1998.

[2] M. de Berg, P. Bose, K. Dobrint, M. van Kreveld, M. Overmars, M. de Groot, T. Roos, J. Snoeyink and S. Yu. The Complexity of Rivers in Triangulated Terrains. In *Proc. 8th Canadian Conference on Computational Geometry*, pages 325–330, 1996.

[3] H. Brönnimann, A. Fabri, G.J. Giezeman, S. Hert, M. Hoffmann, L. Kettner, S. Pion and S. Schirra. 2D and 3D Geometry Kernel. In *CGAL User and Reference Manual*, CGAL Editorial Board, 3.7 edition, 2010.

[4] CGAL, Computational Geometry Algorithms Library. `http://www.cgal.org`.

[5] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological Persistence and Simplification. In *Proc. 41st IEEE Symposium on Foundations of Computer Science*, pages 45–463, 2000.

[6] A. Fabri, F. Cacciola and R. Wein. CGAL and the Boost Graph Library. In *CGAL User and Reference Manual*, CGAL Editorial Board, 3.7 edition, 2010.

[7] M. McAllister. A Watershed Algorithm for Triangulated Terrains. In *Proc. 11th Canadian Conference on Computational Geometry*, 103–106, 1999.

[8] Y. Liu and J.Snoeyink. Flooding Triangulated Terrain. In *Proc. 11th International Symposium on Spatial Data Handling*, pages 137–148, 2005.

[9] United States Geological Survey file server. `http://dds.cr.usgs.gov/pub/data/DEM/250/`.

[10] S. Yu, M. van Kreveld and J. Snoeyink. Drainage Queries in TINs: From Local to Global and Back Again. In *Proc. 7th International Symposium on Spatial Data Handling*, pages 13–1, 1996.