Terrain Trees Library: a tool for efficient and scalable terrain mesh processing

Song Yunting[1], Fellegara Riccardo[2], Iuricich Federico[3], and De Floriani Leila[1]

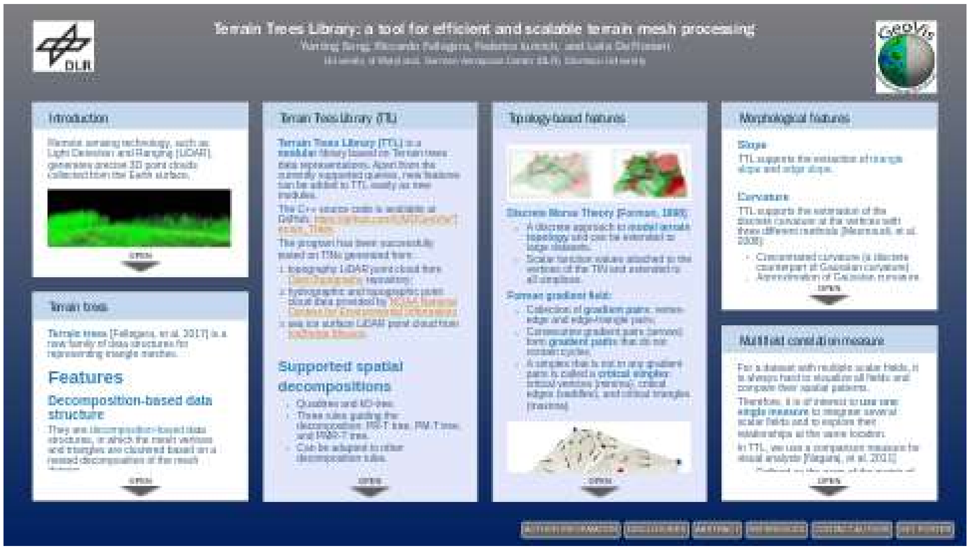
[1]University of Maryland College Park
[2]German Aerospace Center DLR Braunschweig
[3]Clemson University

November 16, 2022

Abstract

In light of the increased availability of massive point cloud data, acquired by advanced remote sensing techniques, software tools for their efficient representation and processing are needed. Triangulated Irregular Networks (TINs) are an effective way to represent point clouds without the need to interpolate them into raster-based terrain models. However, GISs tools have limited support for TINs due to large storage costs. For this reason, we present the Terrain Trees Library (TTL), a library for terrain analysis based on a new scalable data structure named Terrain trees. A Terrain tree relies on a hierarchical spatial index where each leaf block encodes the minimum amount of connectivity information for the TIN. Connectivity relations among the elements of the TIN are extracted locally within each leaf block at run-time and discarded when no longer needed. Moreover, the hierarchical domain decomposition makes the library well-suited for parallel processing. TTL contains a kernel for connectivity and spatial queries, and modules for extracting morphological features, including edge and triangle slopes, roughness, curvature. It also contains modules for extracting topological structures, like critical point, critical net, watershed segmentation, based on the discrete Morse gradient, and a technique for multivariate data visualization, which enables the analysis of multiple scalar fields defined on the same terrain. To evaluate the effectiveness and scalability of TTL, we compared it against the most compact state-of-the-art data structure for TINs, the IA data structure. When encoded by Terrain trees, a TIN requires 36% less storage than when encoded by the IA data structure. Beyond this storage reduction, Terrain trees also show better performance than the IA data structure in most terrain analysis operations. This speedup is obtained since Terrain trees enable 57% to 72% faster extraction of the triangles incident in a vertex. Extracting the triangles incident in vertices as well as the adjacent vertices on the mesh is a key task in most terrain feature extraction operations on a TIN. Using Terrain trees, we achieved 36% to 55% less time consumption computing morphological features and 20% less time consumption computing the discrete Morse gradient than using the IA data structure.

Terrain Trees Library: a tool for efficient and scalable terrain mesh processing



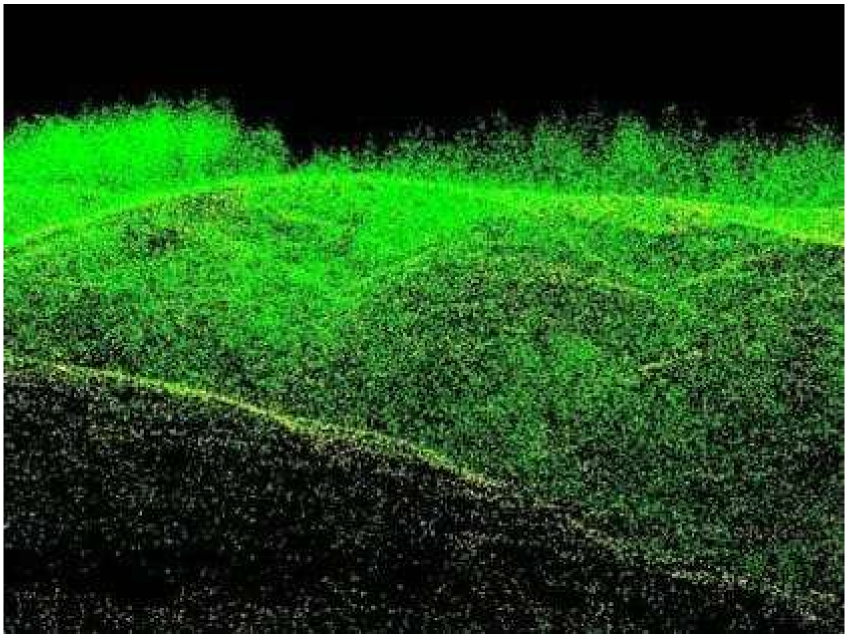
Yunting Song, Riccardo Fellegara, Federico Iuricich, and Leila De Floriani

University of Maryland, German Aerospace Center (DLR), Clemson University

PRESENTED AT:

INTRODUCTION

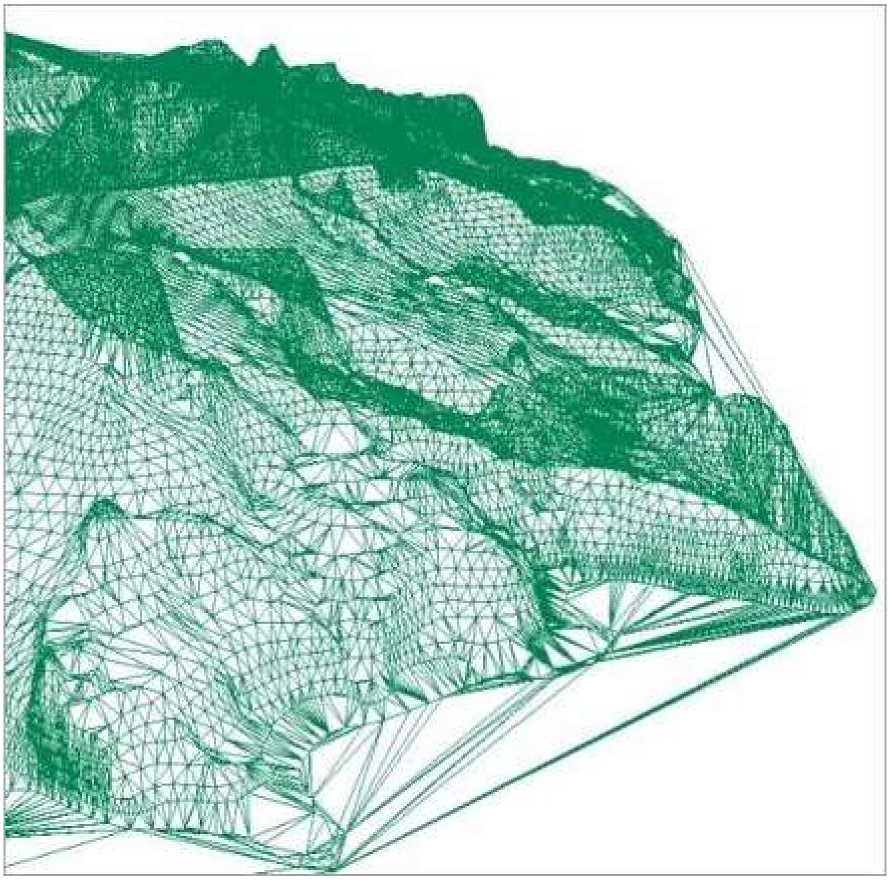
Remote sensing technology, such as Light Detection and Ranging (LiDAR), generates precise 3D point clouds collected from the Earth surface.



But most algorithms do not operate directly on point datasets:

- Raster-based Digital Elevation Models (DEMs) are generated from point clouds to lower the memory consumption
- The limitations of raster-based DEMs:
  - The conversion is computationally intensive;
  - Accuracy is affected by interpolation methods and terrain nature;
  - Line features like boundary or ridgelines are approximated.

An alternative method is to use Triangulated Irregular Networks (TINs) for connecting point clouds and modelling terrain surfaces. TINs are more suitable for irregularly distributed data and can better maintain boundary information.



However, the usage of TINs is limited by their large storage costs. Existing software tools have limited capabilities at handling TINs. And existing data structures for TINs also do not scale with the size of the point cloud.

To provide a solution to the limitations on TINs, we present the Terrain Trees Library (TTL), a library for terrain analysis based on a new scalable data structure named Terrain trees.

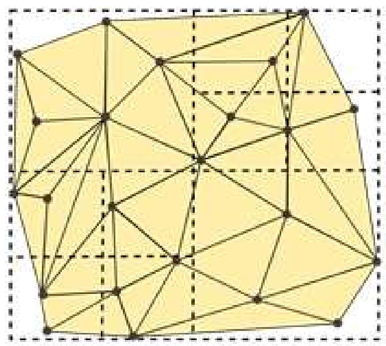
TERRAIN TREES

Terrain trees [Fellegara, et al. 2017] is a new family of data structures for representing triangle meshes.

Features

Decomposition-based data structure

They are decomposition-based data structures, in which the mesh vertices and triangles are clustered based on a nested decomposition of the mesh domain.

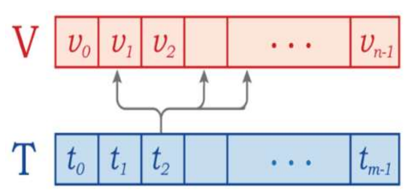


An example of domain decomposition

Minimal mesh connectivity

A Terrain tree for a given TIN Σ consists of:

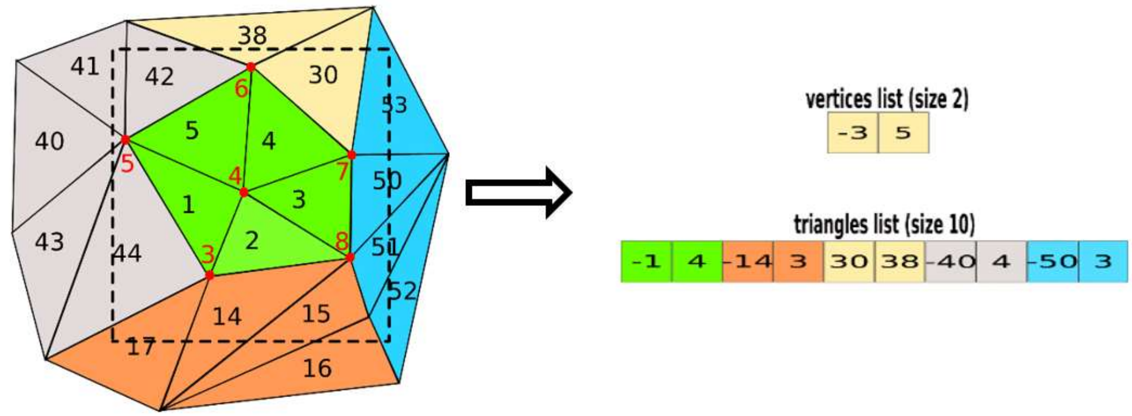
1. A global vertex array stores for each vertex its coordinates plus extra fields – stores the geometry of the mesh;
2. A global triangle array stores the indexes of three vertices for each triangle – stores the minimal mesh connectivity.



Compressed encoding

Each block encodes the indexes of all vertices located in it and triangles intersecting it.

Terrain trees use a **Sequential Run-Length (SRE)** encoding for representing the indexes of vertices and the triangles associated with a block.

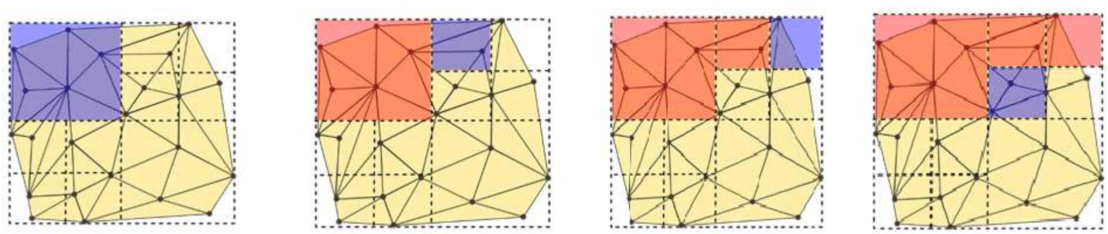


For a block ***b*** within the Terrain tree:

- The **vertex array** in ***b*** is represented with a single run.
- The **triangle array** in ***b*** is represented in general through more than one run, but the storage requirements of the compressed encoding for such array are highly reduced.

The basic paradigm of algorithms

- Locally process a subset of the mesh in a streaming manner by iterating over the blocks;
- For each block, local application-dependent data structures are generated and discarded after the processing.



TERRAIN TREES LIBRARY (TTL)

Terrain Trees Library (TTL) is a modular library based on Terrain trees data representations. Apart from the currently supported queries, new features can be added to TTL easily as new modules.

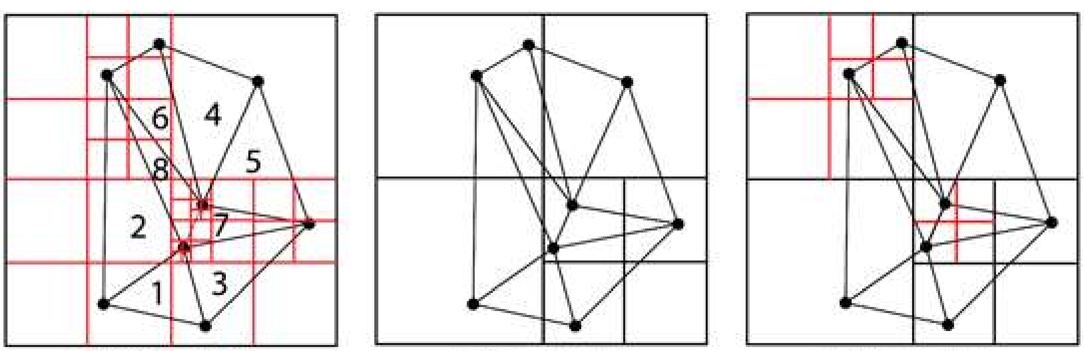
The C++ source code is available at GitHub: https://github.com/UMDGeoVis/Terrain_Trees (https://github.com/UMDGeoVis/Terrain_Trees)

The program has been successfully tested on TINs generated from:

1. topography LiDAR point cloud from OpenTopography (http://www.opentopography.org/) repository;
2. hydrographic and topographic point cloud data provided by NOAA National Centers for Environmental Information (https://www.fisheries.noaa.gov/inport/item/49753);
3. sea ice surface LiDAR point cloud from IceBridge Mission (https://nsidc.org/data/ilatm1b).

Supported spatial decompositions

- Quadtree and kD-tree.
- Three rules guiding the decomposition: PR-T tree, PM-T tree, and PMR-T tree.
- Can be adapted to other decomposition rules.



(a) PMR-T subdivision (b) PR-T subdivision (c) PM-T subdivision

Supported functions

Basic functionalities

- Spatial queries (point location, box query)

- Topological queries (neighbour finding, TIN traversal)

Application-specific queries

Topology-based features:

- Critical points extraction
- Critical net extraction

- Segmentation based on critical points (Influence regions of minima and maxima)

Morphological features:

- Slope, roughness and curvature estimation

Computation of a multifield correlation measure

Experimental evaluation on Terrain trees generation

Datasets

We have used a total of seven TINs with a number of vertices ranging from 34 million to 193 million. All of them are provided by the OpenTopography (https://www.opentopography.org/) repository.

	Terrain	
	Vertices	Triangles
Great Smokey Mount.	34.0M	68.0M
Canyon Lake	49M	98M
Sonoma County1	105M	210M
Sonoma County2	135.5M	271M
Big Creek	151M	303M
Sonoma County3	154M	309M
Sonoma County4	193M	386M

Hardware configuration

A dual Intel Xeon E5-2630 v4 CPU at 2.20Ghz, and 64GB of RAM.

Experiment results

The index of Terrain trees takes **0.5-1.7%** of the total storage cost (total storage cost = cost of vertex array + cost of triangle array + cost of the index).

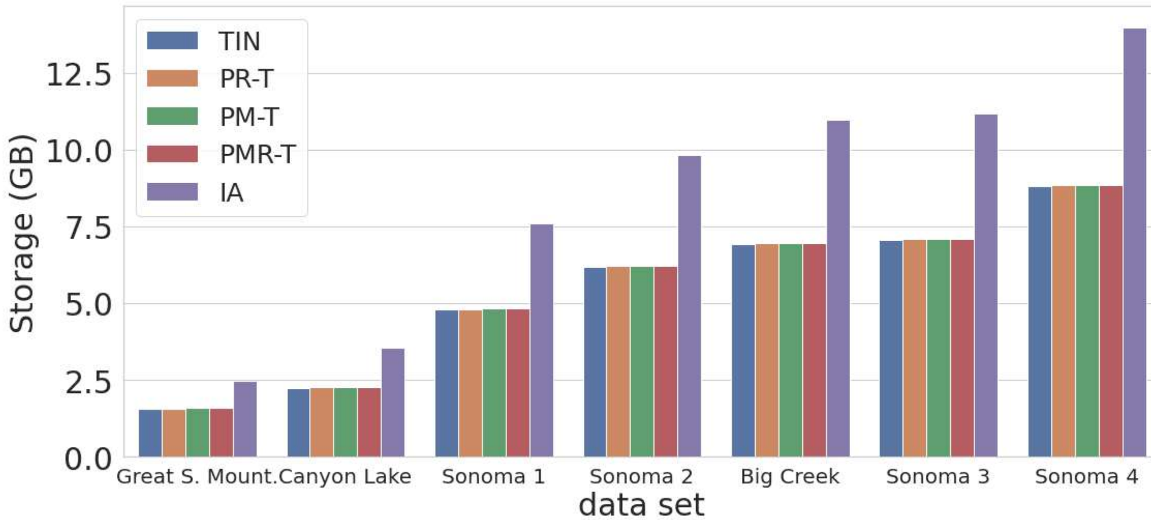
The generation of Terrain trees relies on two parameters (capacity of vertices/triangles per block).

- Our experiments show that the storage costs, generation time, and the time for extracting triangles incident in each vertex decrease as the capacities increase.
- But when using larger capacities, the time and storage costs of processing a single block increase. We choose capacities that can maintain a balance between the cost of generating the index and the cost of processing each block.

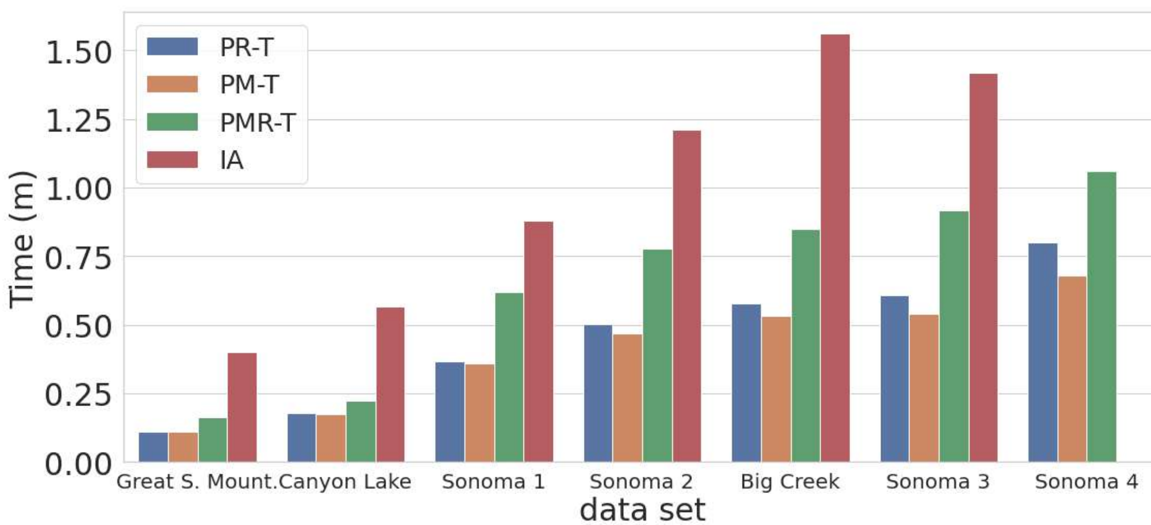
Comparison with the state-of-art data structure

In our experiments, we compare the performance of the TTL with the state-of-art data structure for triangle mesh, which is called the Indexed data structure with Adjacencies (**IA data structure**) [Paoluzzi, et al. 1993]. The IA data structure is the most compact data structure for triangle meshes.

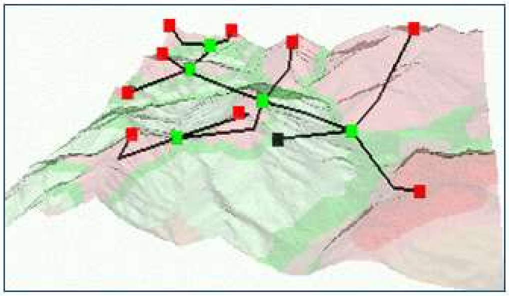
Storage costs: **36% less** cost for storing both the data structure and the TIN compared to the IA data structure.

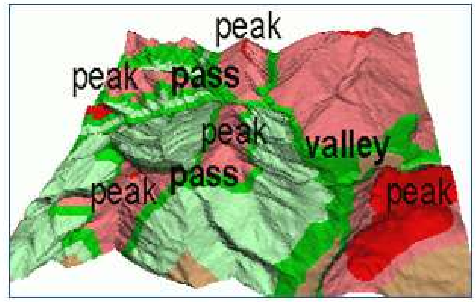


Time for extracting triangles incident in a vertex: **from 30% to 70% less** time than the IA data structure.



TOPOLOGY-BASED FEATURES



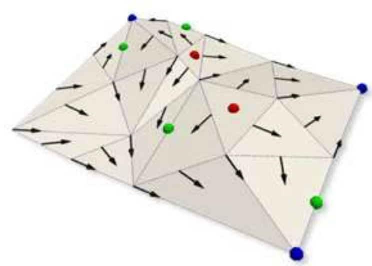


Discrete Morse Theory [Forman, 1998]

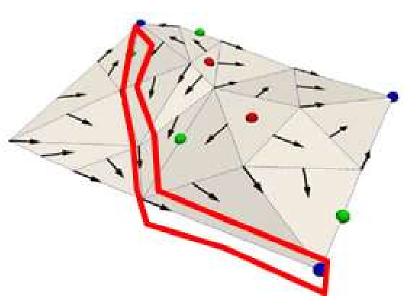
- A discrete approach to **model terrain topology** and can be extended to large datasets.
- Scalar function values attached to the vertices of the TIN and extended to all simplices.

Forman gradient field:

- Collection of **gradient pairs**: vertex–edge and edge–triangle pairs;
- Consecutive gradient pairs (arrows) form **gradient paths** that do not contain cycles;
- A simplex that is not in any gradient pairs is called a **critical simplex**: critical vertices (minima), critical edges (saddles), and critical triangles (maxima).



Forman gradient field



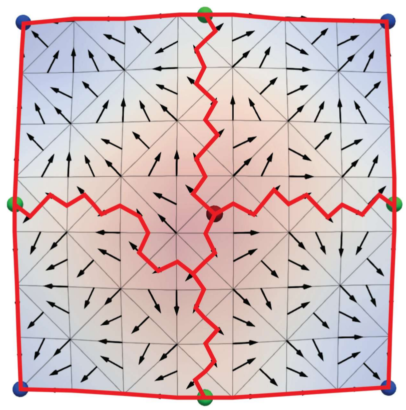
gradient path

Computation of Forman gradient field

TTL supports the computation of the Forman gradient field based on the elevation function defined on the terrain vertices. It also supports the extraction of critical simplices based on the Forman gradient.

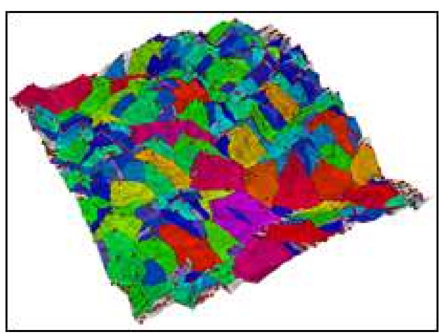
Critical net

TTL supports the extraction of the critical net of a computed Forman gradient field. The critical net consists of the critical simplices and the gradient paths connecting them.



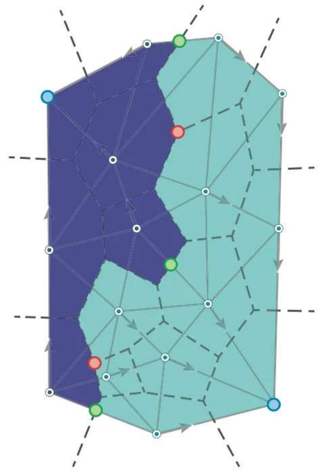
Influence regions

TTL supports the extraction of the influence regions of minima and maxima. This method can be used to segment the terrain based on the topology.

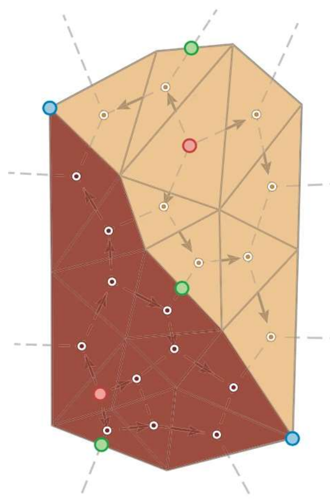


The influence regions of minima/maxima can be extracted based on Discrete Morse theory [Forman, 1998, Fellegara et al., 2014]

- Influence regions of minima: a collection of vertices on the mesh obtained following the (vertex, edge) arrows.



- Influence regions of maxima: a collection of triangles on the mesh obtained following the (edge, triangle) arrows.



Experimental evaluation

Forman gradient computation

- 20% less time than the IA data structure.
- From **21 to 23% less** memory cost than the IA data structure.

Critical net extraction

- PR-T tree and PM-T tree: up to 30% more time than the IA data structure.
- PMR-T tree: up to 8 times slower than the IA data structure.
- About **27% less** memory cost than the IA data structure.

MORPHOLOGICAL FEATURES

Slope

TTL supports the extraction of triangle slope and edge slope.

Curvature

TTL supports the estimation of the discrete curvature at the vertices with three different methods [Mesmoudi, et al. 2008]:

- Concentrated curvature (a discrete counterpart of Gaussian curvature)
- Approximation of Gaussian curvature
- Approximation of mean curvature

Roughness

TTL supports the computation of roughness at vertices. There are different definitions for roughness. In the library, we define it as the **standard deviation** of local elevations at each vertex.

Experimental evaluation

Edge slope

- From **37% to 45% less** time than the IA data structure
- About **56% less** memory cost than the IA data structure

Concentrated curvature

- From **25% to 30% less** time than the IA data structure
- From **27% to 39% less** memory cost than the IA data structure

Roughness

- From **36% to 55% less** time than the IA data structure
- From **27% to 39% less** memory cost than the IA data structure

MULTIFIELD CORRELATION MEASURE

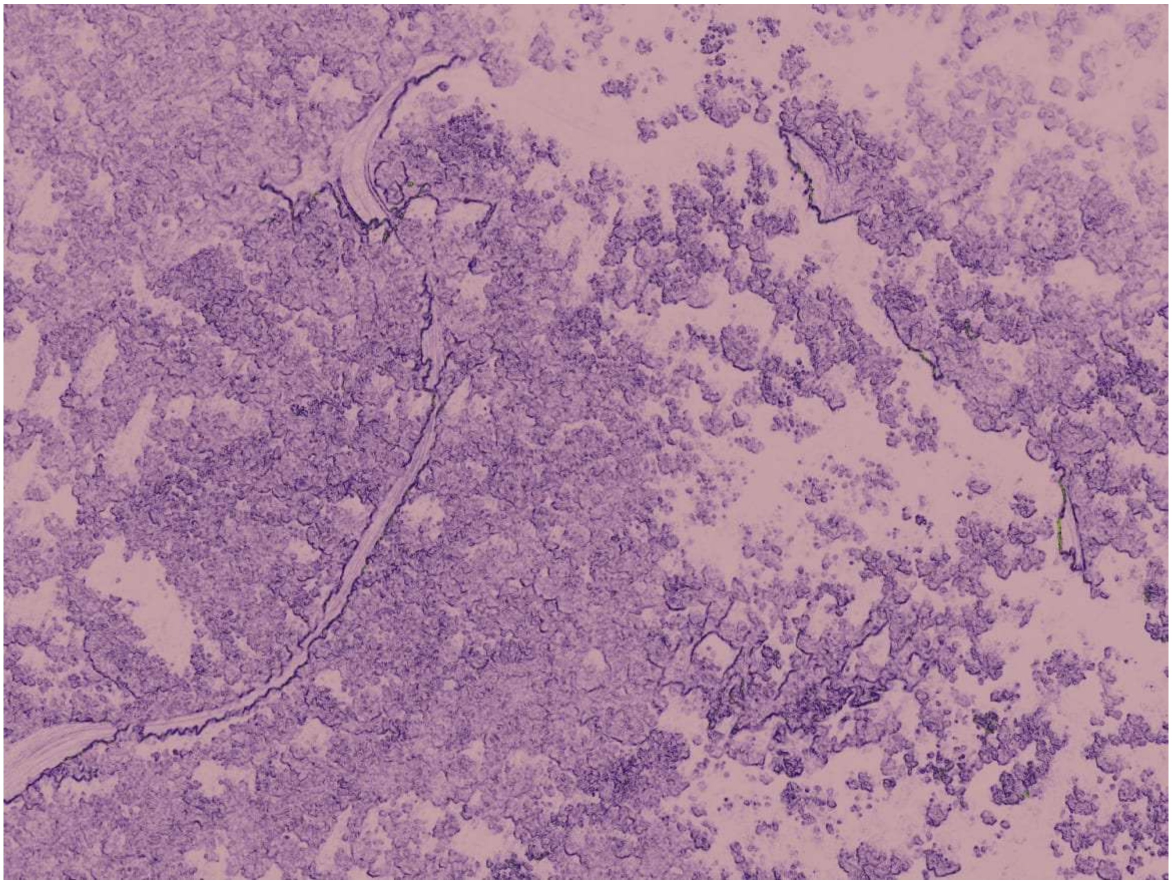
For a dataset with multiple scalar fields, it is always hard to visualize all fields and compare their spatial patterns.

Therefore, it is of interest to **use one single measure** to integrate several scalar fields and to explore their relationships at the same location.

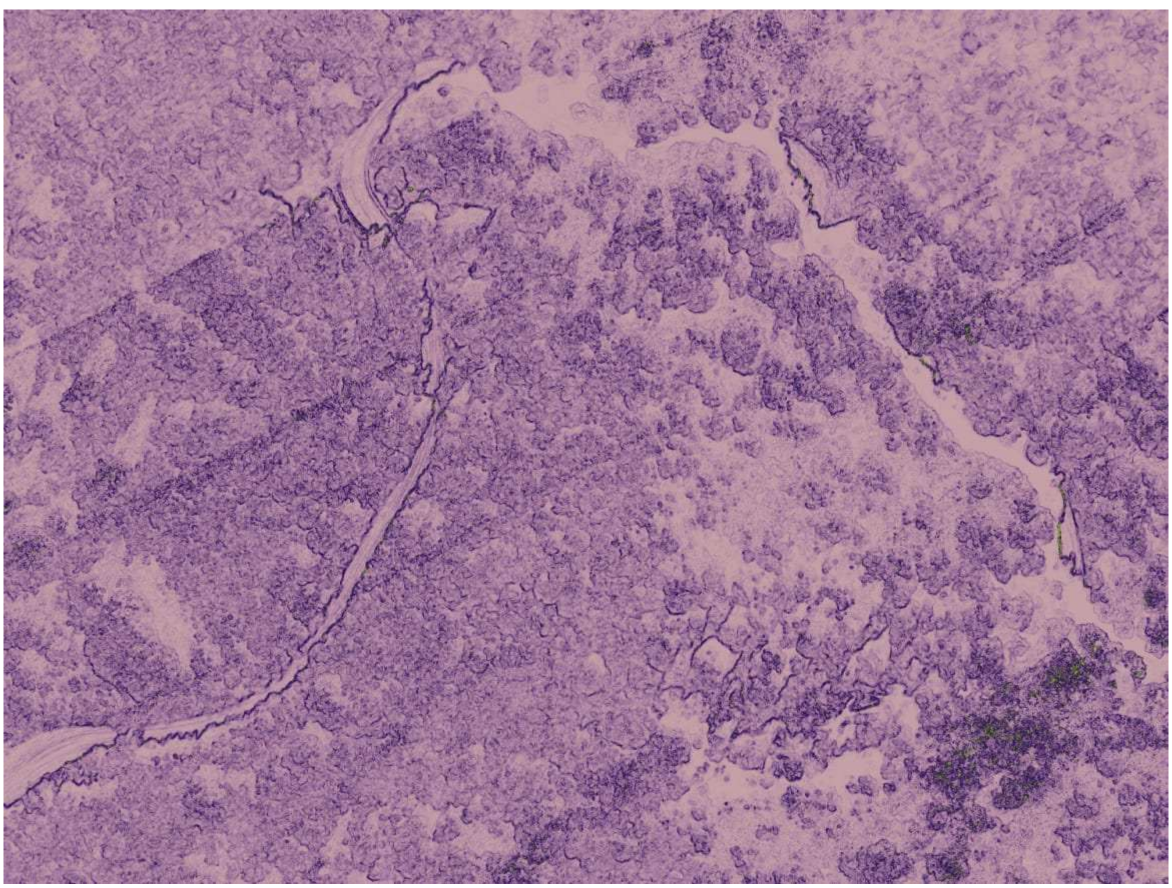
In TTL, we use a comparison measure for visual analysis [Nagaraj, et al. 2011]

- Defined as the norm of the matrix of partial derivatives of the scalar fields at a given vertex;
- Measures the similarity among the change rates of the different scalar fields at that vertex.

Satellite image (from Google Map)



Multifield measure calculated from the RGB values



Multifield measure calculated from the roughness and the RGB values

Experimental evaluation

- **5% less** time than the IA data structure
- **From 17% to 27% less** memory cost than the IA data structure

DISCLOSURES

This work has been supported by the US National Science Foundation (NSF) under grant number IIS-1910766. It has also been performed under the auspices of the German Aerospace Center (DLR) under Grant DLR-SC-2467209.

AUTHOR INFORMATION

Yunting Song, PhD student at the University of Maryland. Email: ytsong@umd.edu

Riccardo Fellegara, senior researcher at the German Aerospace Center (DLR), Institute for Software Technology, Germany. Email: riccardo.fellegara@dlr.de

Federico Iuricich, assistant professor at the School of Computing at Clemson University.
Email: fiurici@clemson.edu

Leila De Floriani, professor at the University of Maryland. Email: deflo@umd.edu

ABSTRACT

In light of the increased availability of massive point cloud data, acquired by advanced remote sensing techniques, software tools for their efficient representation and processing are needed. Triangulated Irregular Networks (TINs) are an effective way to represent point clouds without the need to interpolate them into raster-based terrain models. However, GISs tools have limited support for TINs due to large storage costs.

For this reason, we present the Terrain Trees Library (TTL), a library for terrain analysis based on a new scalable data structure named Terrain trees. A Terrain tree relies on a hierarchical spatial index where each leaf block encodes the minimum amount of connectivity information for the TIN. Connectivity relations among the elements of the TIN are extracted locally within each leaf block at run-time and discarded when no longer needed. Moreover, the hierarchical domain decomposition makes the library well-suited for parallel processing.

TTL contains a kernel for connectivity and spatial queries, and modules for extracting morphological features, including edge and triangle slopes, roughness, curvature. It also contains modules for extracting topological structures, like critical point, critical net, watershed segmentation, based on the discrete Morse gradient, and a technique for multivariate data visualization, which enables the analysis of multiple scalar fields defined on the same terrain.

To evaluate the effectiveness and scalability of TTL, we compared it against the most compact state-of-the-art data structure for TINs, the IA data structure. When encoded by Terrain trees, a TIN requires 36% less storage than when encoded by the IA data structure. Beyond this storage reduction, Terrain trees also show better performance than the IA data structure in most terrain analysis operations. This speedup is obtained since Terrain trees enable 57% to 72% faster extraction of the triangles incident in a vertex. Extracting the triangles incident in vertices as well as the adjacent vertices on the mesh is a key task in most terrain feature extraction operations on a TIN. Using Terrain trees, we achieved 36% to 55% less time consumption computing morphological features and 20% less time consumption computing the discrete Morse gradient than using the IA data structure.

REFERENCES

[1] Forman, R.: Morse theory for cell complexes. Advances in Mathematics 134, 90–145 (1998)

[2] Fellegara, R., Iuricich, F., De Floriani, L., Weiss, K.: Efficient computation and simplification of discrete Morse decompositions on triangulated terrains. In: Proceedings of the 22th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM (2014). URL: https://dl.acm.org/doi/10.1145/2666310.2666412 (https://dl.acm.org/doi/10.1145/2666310.2666412)

[3] Fellegara, R., Iuricich, F., De Floriani, L.: Efficient representation and analysis of triangulated terrains. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL'17, pp. 74:1–74:4. ACM, New York, NY, USA (2017). URL: http://doi.acm.org/10.1145/3139958.3140050 (http://doi.acm.org/10.1145/3139958.3140050)

[4] Iuricich, F., Scaramuccia, S., Landi, C., De Floriani, L.: A discrete morse-based approach to multivariate data analysis. In: SIGGRAPH ASIA 2016 Symposium on Visualization on - SA '16, SA '16, pp. 1–8. ACM, New York, NY, USA (2016). URL: http://dl.acm.org/citation.cfm?doid=3002151.3002166 (http://dl.acm.org/citation.cfm?doid=3002151.3002166)

[5] Nagaraj, S., Natarajan, V., Nanjundiah, R.S.: A Gradient-Based Comparison Measure for Visual analysis of Multifield Data 30(3), 1101–1110 (2011). URL: dl.acm.org/doi/10.1111/j.1467-8659.2011.01959.x (https://dl.acm.org/doi/10.1111/j.1467-8659.2011.01959.x)

[6] Paoluzzi, A., Bernardini, F., Cattani, C., Ferrucci, V.: Dimension-independent modeling with simplicial complexes. ACM Transactions on Graphics (TOG) 12(1), 56–102 (1993)

[7] Mesmoudi, M., De Floriani, L., Magillo, P.: Morphological analysis of terrains based on discrete curvature and distortion. In: W. Aref, M. Mokbel, H. Samet, M. Schneider, C. Shahabi, O. Wolfson (eds.) Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems, pp. 415–418. Irvine, CA, USA (2008). URL: https://dl.acm.org/doi/10.1145/1463434.1463498 (https://dl.acm.org/doi/10.1145/1463434.1463498)