Efficient Constrained Delaunay Triangulation implementation in Java for spatial hydrological analysis

Thomas Leduc[1], Erwan Bocher[2], Fernando González Cortés[3]

[1]CNRS, IRSTV, CERMA, ENSA Nantes, France, thomas.leduc@cerma.archi.fr
[2]CNRS, IRSTV, École Centrale de Nantes, France, erwan.bocher@ec-nantes.fr
[3]CNRS, IRSTV, France, fergonco@gmail.com

Abstract

To quantify the impact of increased urbanization on the peri-urban hydro-systems, it is necessary to design adapted models. Currently, most of the hydrological models, based on classical raster approach, suffer from a lack of efficiency because they essentially do not take into account the urban artifacts and do not match complex geometries such as urban ones (heterogeneous objects, strong space's partition, roads, pits, buildings...).

That is the reason why, we have decided to choose a Triangulated Irregular Network (TIN) model. This approach requires a Constrained Delaunay Triangulation (CDT) preprocessing phase. Also CDT algorithms are numerous, we must admit that there is a real lack of free, efficient and robust implementations adapted to our technological choices. Indeed, all the hydrological simulations we have developed, had been made in the context of the OrbisGIS platform (an open source GIS developed in Java in the French IRSTV Institute). The main objective, with OrbisGIS, is to remain coherent and give to the community a consistent set of simulations tools and methods for the spatial analysis of the urban environment in the context of a single and cross-operating systems GIS.

In this paper, we will first present the OrbisGIS platform. The remainder of this paper is dedicated to the CDT algorithm we choose, its implementation and some comparisons with other implementations. The last part of this paper is dedicated to the hydrological process and the first results we have obtained.

1. Introduction

Due to the increase of urbanization, 60% of the worldwide population is expected to live in towns in 2030. The most important land-use modifications occur in peri-urban zones. A higher concentration of built infrastructures has an impact on hydrology and ecosystems functions. It is therefore necessary to design models to quantify the impact of increased urbanization on the peri-urban hydro-systems.

The main objective of the AVuPUR project (*Assessing the Vulnerability of Peri-Urban Rivers*) is to handle properly the heterogeneity of peri-urban surfaces. Existing hydrological models will be adapted to include the discontinuities induced by natural features such as the hydrographic network and anthropogenic elements such as roads… using a combination of an object-based description developed for urbanised area and a TIN-based approach, used traditionally for rural areas.

In the hydrology community, there has been a substantial amount of research and development using the classical raster approach. The two basics concepts in terrain analysis for hydrology processing are flow routing (model globally the water flow through the terrain) and flow accumulation (quantify how much water flows through each point). Spatial models are often realized with 2D regular orthogonal lattices, using an uniform grid with an elevation given for each cell. Based on this raster DEM, a flow direction grid can be created with several different methods. The simplest and most effective method is the D8 one (O'Callaghan and Mark, 1984). Several methods have then been derived (Tarboton, 1997). This grid approach is the most popular one in hydrology because of its intrinsic simplicity. But, as written in (Kenny, 2005), "a major limitation […] is that watershed boundaries extracted from these data often have significant errors". That the reason why "stream burning offset", "surface reconditioning process" and similar techniques have emerged.

Currently, most of the models suffer from a lack of efficiency because they essentially do not match complex geometries such as urban ones (heterogeneous objects, strong space's partition...). Moreover, most of the grid approaches introduce, in the input data themselves, a great spatial approximation (due to the artificial rasterization process) and also too few degrees of freedom in term of flow directions.

For all those reasons, we have decided to choose a TIN model. Validation will me made using topological and morphological characterization as described in (Gironás et al., 2007). To take benefit from current software development in the French IRSTV institute, we have also decided to develop the model in the context of our *OrbisGIS* platform.

2. The *OrbisGIS* platform and its *GDMS* layer

2.1 Overall view of *GDMS*

GDMS (Bocher et al., 2007) is related with a similar layer used in the gvSIG project to manage the alphanumeric data access. This layer is called GDBMS (Anguix et al., 2005) and one of its main limitation is that it can only be used for alphanumeric purposes.

As noticed in (Bocher et al., 2007), the heterogeneity of source types makes difficult the reuse of algorithms that are tightly coupled with specific format. With an intermediate layer, the work developed by the former will not be coupled with the specificities of each format but with the intermediate layer itself, letting the work to be reused in a much more wide set of scenarios and of course simplify the learning curve for new developers. With GDMS, the idea is to postpone all problems of interoperability across data repositories, but also about the SQL semantics by developing a highly flexible, portable and standards compliant tool to build SQL queries.

With the purpose of simplifying both access and manipulation of data sources, *GDMS* provides a SQL processor that lets the execution of the common Data Manipulation Language statements against any source mapped by a driver. To avoid introducing a new grammar, *GDMS* fully preserves the SQL-92 one and adds to this standard, geometric concepts and spatial functions as in OGC simple features SQL specification.

2.2 OrbisGIS in action

All the hydrological simulations we have developed, have been written in the context of the *OrbisGIS* platform. The main objective, with *OrbisGIS*, is to remain coherent and give to the research community a consistent set of simulations tools and methods for the spatial analysis of the urban environment in the context of a single and cross-operating systems GIS.

3. Preprocessing phase: the Constrained Delaunay Triangulation (CDT)

3.1 Preliminaries

As mentioned in the introduction, the methodological choice we made, lead us first to develop a CDT preprocessing phase. There are a wide amount of algorithms available to build a triangulation for a set of points. To obtain an optimal set of triangles, without adding *Steiner points* (that is to say vertices of the mesh that are not vertices of the input), the most frequently applied method is the one of Delaunay. A quite exhaustive review of sequential DT algorithms has been made in (Su, 1997). Given a finite set of points in the plane, three points contribute a triangle to the Delaunay Triangulation (DT) if the circumcircle through those three points contains no other point in its strict interior. This is the well-known Delaunay Property (DP). Such a definition produces an optimal set of well-shaped triangles. Moreover, the produced triangulation is unique (assuming careful handling of the degeneracies: 4 or more co-circular points) and so repeatable.

In addition to point, linear constraints may also be given as input datas (as in a set of iso-height lines). The objective is to be able to mesh any planar straight-line graph (PSLG). PSLG includes polygons with or without hole(s), dangling edges and isolated vertices. This generalization of the original DT is called the CDT: constraining input segments must become edges of the generated CDT. The DP itself has to evolve into a weaker version: if the circumcircle of a triangle contains a fourth point, this one must be located behind a constraining edge.

The algorithms for building CDT can be classified in two groups:

- the first one considers separately points and line-segments. First process is to mesh given set of points with a classical DT (choosing a strategy: divide and conquer, sweep-line, incremental, high dimensional embedding…) and then add the linear constraints respecting the weaker DP;
- the second one processes points and edges simultaneously. (Domiter et al., 2008) notices that "such algorithms are faster, consume less memory and are less sensitive to data distribution".This is the strategy that is chosen in the algorithm described in (Domiter et al., 2008).

3.2 The Sweep-line CDT algorithm

The CDT algorithm we have implemented uses a sweep-line paradigm combined with Lawson's legalization. As written in (Domiter, 2008), the proposed algorithm consists of three parts:

- an initialization phase:
  - input data points are first sorted. Each point is also associated with the information whether or not it is the upper ending point of one or more edges;
  - then, two sensible artificial points, an initial triangle and an initial sweep-line (SL) are created,
- a sweeping phase: following the sweep-line paradigm, it stops at every point and check its status
  - in case of a point event (isolated or lower edge point):
    - the point's vertical projection on the SL is computed. Based on this projection point, one or two triangles are generated and legalized. The SL is updated;
    - to avoid undulation of the advancing front, basin are filled by triangles,
  - in case of an edge event (upper edge point):
    - first, each triangle that intersects a constraining edge is removed (taking into account that the SL may be intersected at one or more points) and the SL is updated,
    - then, what still remains is to re-triangulate the empty area.
- a finalization phase: when all points have been swept, two ending tasks still remain
  - removing all the triangles defined by at least one artificial point,
  - adding the bordering triangles (the union of them all must form the convex hull of input data).

3.3 Our own implementation

For coherency reason, we have implemented this CDT algorithm in the *OrbisGIS* platform, that is based on the robust JTS (Java Topology Suite) library. This library provides spatial indexes that we have included: a MX-CIF quadtree for the triangles and a query-only R-tree created using the STR algorithm for the immutable set of input data. However, the implementation we made is loosely coupled to the *GDMS* layer. Indeed, the interface part has just to do with inputs (reading the PSLG) and outputs data (storing the resulting triangles).

The asset of the platform is its ability to make the development of spatial tools easier. Thus, it is already possible to verify the validity of the DP for each resulting triangle with a single (*GDMS*) SQL query: *select CheckDelaunayProperty() from mySetOfTriangles*. We have made it also possible to compare two sets of triangles (*CheckSpatialEquivalence*), measure the quality of the produced triangles (*QualityMeasuresOfTIN*)…

Before implementing the sweep-line algorithm, we have also tried to interface some existing meshing tools. Thus, we have tested, in the platform context, the famous *Triangle* program developed in C language by Shewchuk, but also a Java implementation made by M. Davis (WaterBuG project), another one written by P. Austin and, at least, a really efficient one written by M. Michaud. As an example we may focus on the Shewchuk's implementation. It is the only one we tested that give the possibility to refine a mesh. As shown in figure 1, the *Steiner points* that are added increase the minimum angle value (from 11.4 to 21.4 degrees) and therefore produce better shaped triangles.

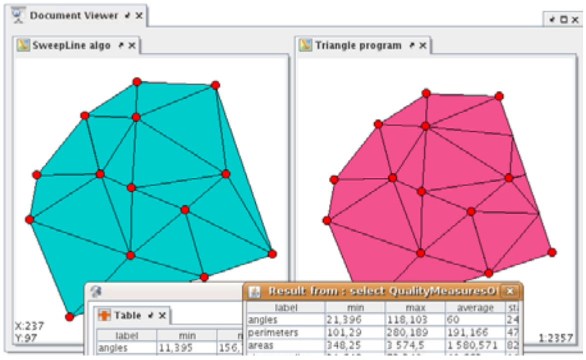

Figure 1. The influence of the Delaunay refinement algorithm on the triangulation. Shewchuk's implementation (on the right side) introduces some Steiner points to obtain a greater minimum angle.

4. Hydrological simulations

As written in (McAllister, 1999), every sample point of a Triangulated Irregular Networks (TINs) must store not only its coordinates and elevation values but also face adjacency information. Starting from a set of triangles, we have to build a topographic graph. The data structure we adopt, is the one described in (Kreveld, 1996). The next step is to build a hydrological graph using this topographic one. The idea is to determine the contributing area for a river[1]. It presupposes that for each graph node a steepest descent direction is computed (we assume that water always follows the path of steepest descent). The hydrological graph is produced from the topographic one according to the drainage characteristics. All the topographic vertices that contribute to a same common local minimum in the terrain are grouped. These groups are the watersheds.

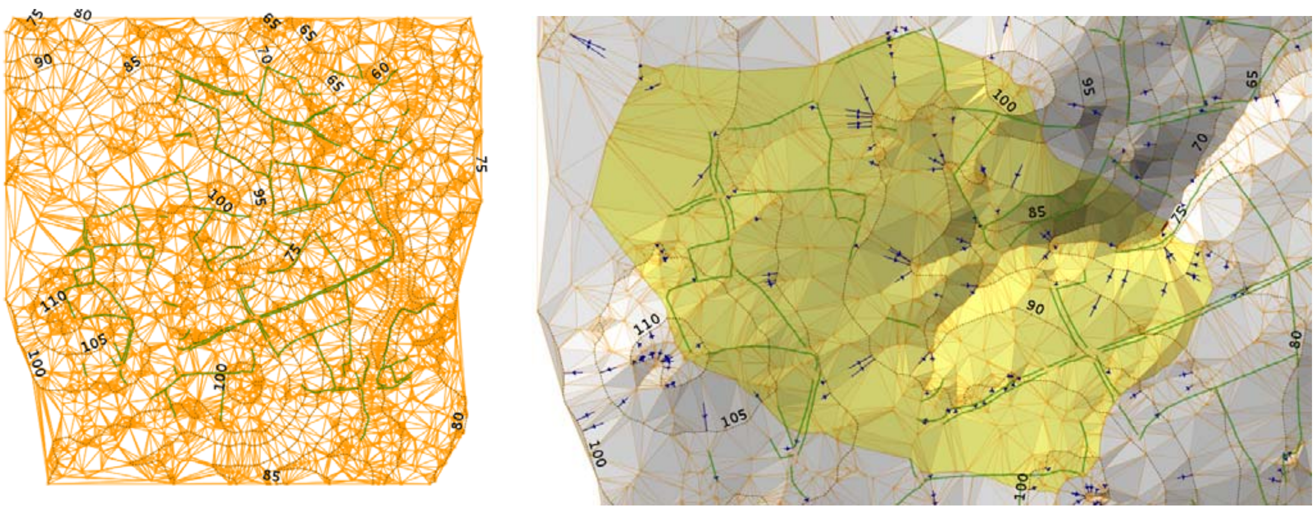

Figure 2. Starting from a set of set of iso-height lines and a set of linear constraints, a CDT is build (left side). The watershed is derived from the TIN, through a hydrological graph (right side).

1 The watershed of a river is the set of terrain points that drain into this river.

5. Conclusion

We have presented an implementation of a new CDT algorithm and its first application in a TIN-based hydrological model. There are several directions for further work. The first one has to do with a CDT refinement algorithm, in order to obtain better shaped triangles. We have also planned to develop a *GDMS* native format to store both the topographic graph and the hydrological graph.

Acknowledgments

The AVuPUR project was funded by the French *Agence Nationale de la Recherche* (ANR) under contract n°ANR-07-VULN-01.

References

A. Anguix and G. Carrion, 2005. 'gvSIG: Open Source Solutions in spatial technologies', *GIS Planet*, Estoril, Portugal.

Bocher E., T. Leduc, G. Moreau, and F. González Cortés, 2008. 'GDMS: an abstraction layer to enhance Spatial Data Infrastructures usability', *11th AGILE International Conference on Geographic Information Science - AGILE'2008*, Girona, Spain.

Domiter V. and B. Žalik, 2008. 'Sweep-line algorithm for constrained Delaunay triangulation'. *International Journal of Geographical Information Science*, 22(4) :449–462.

Gironás, J., L. A. Roesner and H Andrieu, 2007. 'Morphologic approach in studying developing urban watersheds', *Geophysical Research Abstracts*, Vol. 9.

Kenny F. and B. Matthews, 2005. 'A methodology for aligning raster flow direction data with photogrammetrically mapped hydrology'. *Computers & Geosciences*, Elsevier ltd., 31(6) :768–779.

McAllister M, 1999. 'The Computational Geometry of Hydrology Data in Geographic Information Systems'. PhD thesis, The University of British Columbia.

O'Callaghan, J. F. and D. M. Mark, 1984, 'The Extraction of Drainage Networks From Digital Elevation Data', *Computer Vision, Graphics and Image Processing*, 28: 328-344.

Shewchuk J. R., 2000. 'Sweep algorithms for constructing higher-dimensional Constrained Delaunay Triangulations'. In *Proceedings of the 16th annual symposium on Computational geometry*, Kowloon, Hong Kong, pages 350–359. ACM.

Su P. and R. L. S. Drysdale, 1997. 'A comparison of sequential Delaunay triangulation algorithms'. *Computational Geometry Theory Appl.*, 7(5-6) :361–385.

Tarboton, D. G., 1997, 'A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models', *Water Resources Research*, 33(2): 309-319.

van Kreveld M. J., 1996. Digital elevation models and tin algorithms. In Algorithmic Foundations of Geographic Information Systems, pages 37–78.