infrastructures

MDPI

Technical Note

Comparative Analysis of Triangulation Libraries for Modeling Large Point Clouds from Land and Their Infrastructures

Luis Lopez-Fernandez [1], Pablo Rodriguez-Gonzalvez [1], David Hernandez-Lopez [2], Damian Ortega-Terol [2] and Diego Gonzalez-Aguilera [1,*]

[1] High Polytechnic School of Engineering, University of Salamanca, Ávila. Av. de los Hornos Caleros, 50, 05003 Ávila, Spain; luisloez82@usal.es (L.L.-F.); pablorgsf@usal.es (P.R.-G.)

[2] Institute for Regional Development (IDR), Albacete, University of Castilla-La Mancha, Campus Universitario s/n, 02071 Albacete, Spain; david.hernandez@uclm.es (D.H.-L.); dortegat@gmail.com (D.O.-T.)

* Correspondence: daguilera@usal.es; Tel.: +34-920-353-500

Academic Editors: Higinio González Jorge and Pedro Arias-Sánchez
Received: 9 November 2016; Accepted: 27 December 2016; Published: 3 January 2017

Abstract: Although the generation of large points clouds from geomatic techniques allows us to realize the topography and appearance of the terrain and its infrastructures (e.g., roads, bridges, buildings, etc.), all these 3D point clouds require an unavoidable step to be conveniently treated: the definition of the surface that connects these points in space through digital surface models (DSM). In addition, these point clouds sometimes have associated attributes and geometric constraints such as breaklines and/or exclusion areas, which require the implementation of efficient triangulation techniques that can cope with a high volume of information. This article aims to make a comparative analysis of different Delaunay triangulation libraries, open or with academic versions available for the scientific community, so that we can assess their suitability for the modeling of the territory and its infrastructures. The comparison was carried out from a two-fold perspective: (i) to analyze and compare the computational cost of the triangulation; (ii) to assess the geometric quality of the resulting meshes. The different techniques and libraries have been tested based on three different study cases and the corresponding large points clouds generated. The study has been useful to identify the limitations of the existing large point clouds triangulation libraries and to propose statistical variables that assess the geometric quality of the resulting DSM.

Keywords: large point clouds; Delaunay triangulation; land; infrastructures; surveying engineering; software development

1. Introduction

The beginning of this century has been characterized by the advancement of geomatics techniques, from remote data collection to processing and visualization of geoinformation [1]. Technologies such as Light Detection and Ranging (LiDAR) [2] and the emergence of new sensors in photogrammetry (e.g., multispectral and hyperspectral cameras) [3] and remote sensing (e.g., Sentinel) [4] have led to a remarkable growth in the quality of geomatic products, in particular (i) the generation of 2D cartographic products in the form of maps and orthoimages [5]; (ii) the generation of 3D cartographic products in the form of digital surface models (DSM) [6], canopy height models (CHM) [7] and even digital city models (DCM) [8]; and (iii) monitoring and simulation based on the temporal analysis of these products [9]. Overall, the fundamental role of these geomatics technologies along with their products is to extract useful information, offering efficient support tools for problem-solving and decision-making at different scales.

All land-related disciplines have benefited, to a greater or lesser extent, from these developments, but especially civil and surveying engineering, since they allow for land characterization along with their associated infrastructures. However, one of the biggest challenges that this new trend of geoinformation faces is to obtain efficient processing and analysis of so much information, called big data analytics, so that heterogeneous data can be integrated, yielding useful and relevant information [10].

Although LiDAR technology is being widely used [11] to capture 3D information in the form of dense point clouds in a quick and easy way, its spatial resolution is limited to 6–10 points/m^2. In addition, LiDAR technology encloses a number of significant technical limitations such as the effective measurement acquisition range, acquisition rate (density), precision, and high cost. On the contrary, modern photogrammetry and dense matching techniques [12] provide better resolution than LiDAR (one depth or one elevation is generated per pixel), although they require a high photogrammetric computational cost. In any case, both LiDAR and photogrammetric point clouds require the extraction of relevant and useful information beyond their own point cloud. So, the reconstruction and analysis of the surface defined by these points is desirable. This surface defines the geometric and topological relationships among 3D points in space. The most common methods to represent surfaces from vector data are triangular irregular network (TIN) [13] structures, which have to face difficulties of triangulating millions of points, incorporating geometric constraints such as breaklines and/or exclusion areas. Therefore, the triangulation of point clouds in the TIN format becomes an unavoidable step that requires a large computational cost, especially with large point clouds [14].

Several strategies have been developed for the triangulation of large point clouds: (i) *Divide-and-conquer strategies*, which allow us to divide the problem into smaller sub-problems that can be solved independently [15]. More specifically, this strategy splits the original point cloud into small datasets, then recursively merges their triangulations to complete the Delaunay triangulation. The merge of subsets and the modification of overlap areas is the most crucial step in the process [16]; (ii) *Cache-efficient strategies*, which act directly in the hardware memory (caches and virtual memory) [17]. These strategies can respond to optimized software for a particular cache architecture or software designed to cooperate well with any cache or virtual memory, regardless of the details of its architecture [18]; (iii) *External memory strategies*, which outsource the calculation through data structures stored on a disk [19]. These strategies explicitly manage the contents of each level of the memory hierarchy directly in the triangulation algorithm, passing through the virtual memory system [20]; (iv) *Streaming strategies*, which sequentially read a stream of data and retain only a small portion of the information in the memory [21]. The common basic idea of streaming geometric algorithms is to exploit the spatial coherence of the data stream, which is the proximity of the points in the space in relation with their proximity in the stream [22]. There are also several authors who have developed techniques for optimizing the Delaunay triangulation of large point clouds: Blandford et al. [14] developed structures of compressed data to dynamically maintain triangulations in two and three dimensions. Isenburg et al. [18] developed algorithms to accelerate large Delaunay triangulations in 2D and 3D. Agarwal et al. [23] designed and implemented an external memory algorithm to build Delaunay triangulations constrained in a plane.

Looking for efficient and open solutions to address this challenge, in this paper a comparative analysis in terms of computational cost (machine time) and geometric quality (accuracy assessment), using different strategies and libraries for Delaunay triangulations, has been carried out. Delaunay triangulation (or tetrahedralization) of a set of points has the property that the circle which circumscribes each triangle or the sphere which circumscribes each tetrahedron does not enclose any point; also, each one of the edges or sides of the triangles generated does not cut any breakline or geometric constraint that clearly defines a change in the terrain or associated infrastructures.

The paper has been structured as follows: after this introduction, Section 2 describes the employed libraries and the developed tool. Section 3 analyzes and compares the results of the three study cases in terms of computational cost and geometric quality. Finally, the most significant conclusions of the article are outlined.

2. Materials and Methods

In this section, the five triangulation libraries used and the in-house software developed for the comparative analysis are described. The triangulation libraries were chosen on the basis of being open or, at least, having an accessible licensing for the scientific community, so that the study can be replicated by anyone interested in the triangulation of large points clouds.

2.1. Evaluated Libraries

2.1.1. LASTools

LASTools is a collection of command line tools that allow multi-core and batch processing of points clouds [24]. The distribution includes tools for classification, conversion, filtering, rasterize and triangulating point clouds among many other utilities. LASTools is a solution to process LiDAR or photogrammetric points clouds (in .las, .laz. or ascii format) supporting up billions of points datasets with the BLAST extension. The triangulation approach used is based on streaming strategy [21]. The input stream for the sequential Delaunay triangulation is the point, as well as the information about regions of space for which the stream is free of future points [18]. This modification allows to retain in memory only the active parts of the triangulations under generation. Although LASTools has the option of a commercial licensing, focused on working with billions of points (BLAST), its open licensing for the Scientific Community allows to integrate available versions with a limitation in the number of points.

2.1.2. Fade2.5D

Fade2.5D is a library developed in C ++ and focused on a 2.5D Delaunay triangulation [25]. The 2.5D term implies that the terrain models are monotonic in X and Y, namely, there is only one possible Z for each XY location. This does not hinder its use for 3D dataset (e.g., urban models), although just one Z value, of all the points with common planar coordinates, is used for the computation of the DSM. The library allows the inclusion of breaklines and the option to force the Delaunay triangulation vertices to be interpolated. Fade2.5D is a tool open for scientific research and students.

2.1.3. Triangle

Triangle is a C library implemented to generate high quality two-dimensional meshes of Delaunay triangulations [26–28]. The software allows the generation of strict Delaunay triangulations, constrained Delaunay triangulations by conditional elements as breaklines, Voronoi diagrams and high quality triangular meshes using geometric constraints (angular and edge distance). Thus, the result is suitable for finite element analysis. Two-dimensional mesh generation and construction of Delaunay triangulation is carried out by means of a Ruppert's Delaunay refinement algorithm [29] plus a refinement algorithm for concavities and holes removal. When they are removed, the algorithm refines the mesh inserting additional vertices into it according to Lawson's algorithm [30] to maintain the Delaunay property, until two constraints (minimum angle and maximum triangle area) are accomplished. Triangle library is open and can be used freely for scientific purposes [28].

2.1.4. CGAL

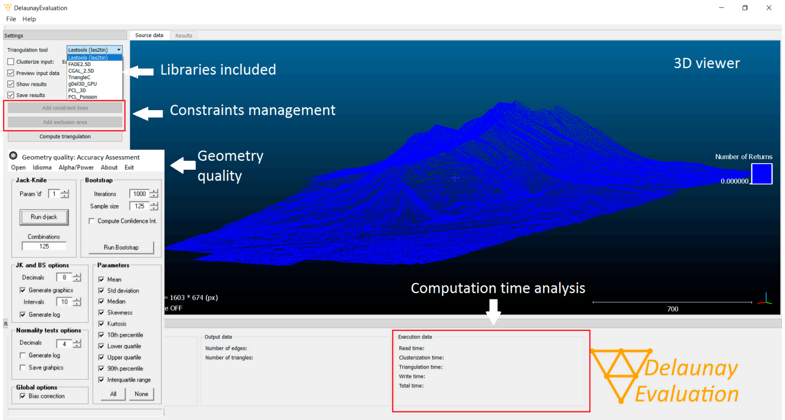
Computational Geometry Algorithm Library (CGAL) is a library for geometric calculations that provides easy access to efficient and reliable geometric algorithms written in C++ [31,32]. The library allows the triangulations of point clouds in three dimensions, supporting different triangulations options: basic Delaunay, generalized Delaunay and the Constrained Delaunay, which was used in this paper. Please note that a constrained Delaunay is a triangulation whose faces do not necessarily fulfill the empty circle property but any triangle satisfies the constrained empty circle propriety: its circumscribing circle does not enclose vertex visible from the interior of the triangle [33]. In the process, the 3D points are projected onto a reference plane (*XY* in the terrain case), adding the vertices incrementally and iteratively while performing successive refinements [34]. Optionally, algorithms and regular triangulation support "multi-core" and "shared-memory" architectures to benefit from available parallelization strategies. CGAL library allows an open use for scientific research, but incorporates a part of commercial licensing.

2.1.5. gDel3D

gDel3D algorithm is a hybrid GPU-CPU algorithm to perform a Delaunay triangulation carrying out an incremental strategy for point insertion followed by a parallelization algorithm [35–37]. The first step is carried out in the GPU to obtain a triangulation with very few locally non-Delaunay facets. More detail, at each iteration, each tetrahedron with points inside picks one of them to insert. After all the point insertions, the facets in the triangulation are checked and locally non-Delaunay facets are flipped to create facets that are locally Delaunay by a bilateral flip in 3D [30]. Then, the algorithm refines the result using a conservative star splaying strategy [38] sequentially applied on the CPU for the final 3D Delaunay triangulation. The use of heterogeneous parallel programming by gDel3D provides a run times optimization up to 10 times compared with 3D Delaunay triangulation implemented by CGAL [36]. gDel3D algorithm is licensed for commercial use, while allowing its use in demo mode for scientific purposes.

2.2. *In-House Software*

In order to carry out a comparative analysis in an optimal way, an in-house software, "DE-Delaunay Evaluation", has been developed in C++/QT as a framework to test the different large point clouds triangulations libraries and to perform their statistical analysis (Figure 1). The software allows loading any point cloud in LAS or ASCII format, displaying the point cloud or resulting mesh in a 3D viewer. However, in the executed tests, the LAS format was chosen due to the considerable reduction of reading times compared to ASCII format. Whereas, the output format chosen was the *wavefront* (.obj) for its simplicity of writing, interpretation and analysis by the geometric quality module of the software. Besides, the software allows the triangulation through any of the triangulation libraries described above and the incorporation of geometric constraints (e.g., breaklines, exclusion areas), which are a key factor to model land and their infrastructures. Last but not least, the software includes a clustering strategy to provide a greater flexibility in point clouds viewing and management. Finally, the software incorporates two assessment modules: (i) one related to the computation time along its different phases (reading, triangulation and writing) and (ii) other regarding the geometric quality (accuracy assessment) of the obtained results. For this last check, a comparison with a "ground truth" was performed using a certified and contrasted commercial software for DSM generation "INPHO-DTMaster" [39].



DelaunayEvaluation

Settings

Triangulation tool: Lastools (las2tin)

- Lastools (las2tin)
- FADE2.5D
- CGAL_2.5D
- TriangleC
- gDel3D_GPU
- PCL_3D
- PCL_Poisson

☐ Clusterize input:
☑ Preview input data
☑ Show results
☑ Save results

Geometry quality: Accuracy Assessment

Jack-Knife: Param 'd' 1; Combinations 125

Bootstrap: Iterations 1000; Sample size 125; ☐ Compute Confidence Int.

JK and BS options: Decimals 8; ☑ Generate graphics; Intervals 10; ☑ Generate log

Normality tests options: Decimals 4; ☐ Generate log; ☐ Save grahpics

Global options: ☑ Bias correction

Parameters: ☑ Mean ☑ Std deviation ☑ Median ☑ Skewness ☑ Kurtosis ☑ 10th percentile ☑ Lower quartile ☑ Upper quartile ☑ 90th percentile ☑ Interquartile range

Output data: Number of edges: Number of triangles:

Execution data: Read time: Clusterization time: Triangulation time: Write time: Total time:

Figure 1. Interface of the software developed, "DE-Delaunay Evaluation", for the comparative analysis of triangulation libraries for modeling large point clouds.

3. Results

3.1. Study Cases Description

The comparative analysis of triangulation libraries has been performed using three study cases which represent different needs for terrain modeling and its infrastructures (Figure 2): (i) Zone 1—Archaeology (Albacete, Spain); (ii) Zone 2—Quarry (Alicante, Spain); and (iii) Zone 3—Road (Alicante, Spain).

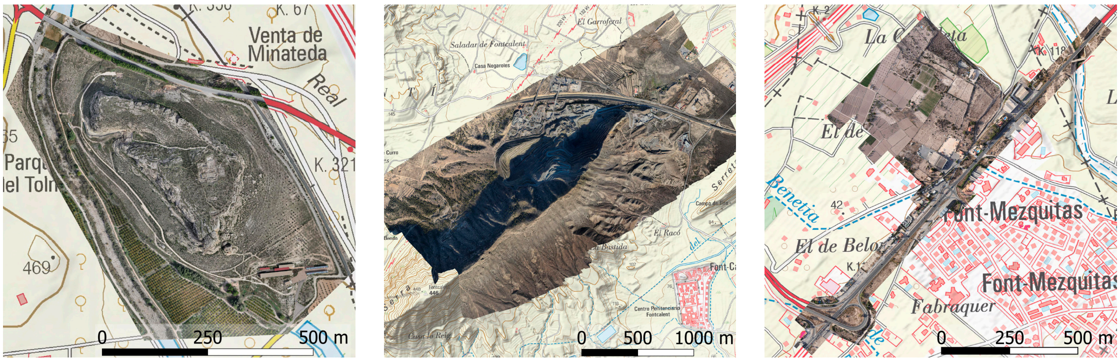


Figure 2. Study cases employed for the large point clouds triangulation: Zone 1—Archaeology (**Left**); Zone 2—Quarry (**Center**); and Zone 3—Road (**Right**).

Each study case tries to highlight the need for a high quality DSM generation that allows us to model the topography and its infrastructures.

In the case of *Zone 1—Archaeology*, a steep part of the archaeological site of Tolmo de Minateda (Albacete, Spain) was chosen with the aim of generating the true orthoimage of the existing archaeological remains. Generating an orthoimage requires the prior and mandatory DSM calculation of the area and hence the need to undertake an efficient and high quality triangulation of the resulting point cloud.

In the case of *Zone 2—Quarry*, an existing open-pit gravel quarry was chosen to highlight the applicability of DSM through time to monitor mining operations. More specifically, there was an interest for the monitoring of quarries (monthly cubage processes of the extracted material) in difficult-to-access areas. Providing an accurate and reliable volumetric certification of the extracted material in the quarry will depend on generating a high quality DSM that can be compared over time.

Finally, in the case of *Zone 3—Road*, the aim was to generate a DSM that allows us to extract the road alignment, both the plan and elevation, as well as the topography of parcels and adjacent buildings. Therefore, generating a high quality DSM was desirable.

The resulting point clouds were obtained from the photogrammetric processing based on dense matching and using GRAPHOS software [12]. The results are outlined in Figure 3.

3.2. Computational Performance Analysis

Algorithms have been tested on a Mountain computer equipped 32 GB of RAM DDR3-1600, a dedicated Nvidia Quadro 2000 graphics card and an Intel Core i5-3570K processor at 3.40 GHz. In order to limit hardware dependencies that could affect execution times due to the resource needs of each tool, hardware usage was monitored during execution, ensuring that no volatile memory overflows occurred. This control ensured that no persistent memory was used as an exchange area, affecting the execution time due to differences in reading/writing processes. The different performance tests were performed using different DSMs with different resolutions. Thus, tests were carried out for subsets of 10,000, 25,000, 50,000, 100,000, 500,000, one million, two million, five million, 10 million, 25 million, 50 million, 100 million and 250 million points. The purpose of analyzing different subsets of points was two-fold: on one hand, to analyze the computational efficiency of the libraries with different sets of points, observing the trend (i.e., lineal, logarithmic); on the other hand, to adapt those libraries which, in spite of being open to the scientific community, restrict the maximum number of points for triangulation.

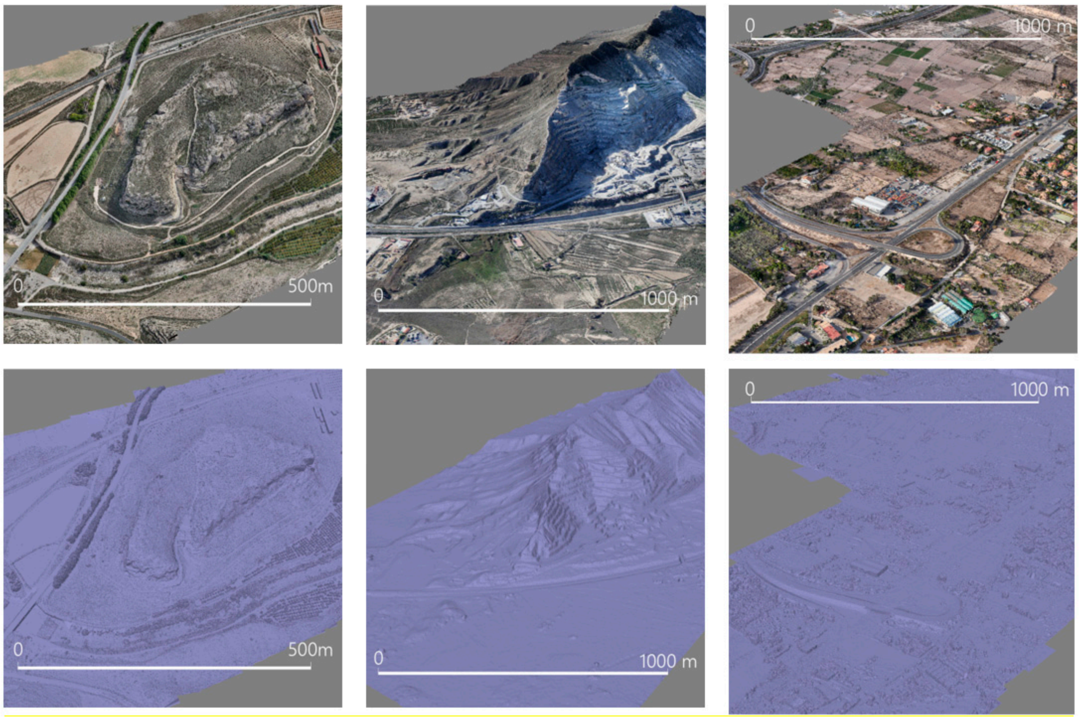


Figure 3. Obtained point clouds and DSMs from the photogrammetric processing and triangulation: Zone 1—Archaeology, resolution: 651 points/m^2 (**Left**); Zone 2—Quarry, resolution: 6 points/m^2 (**Center**); and Zone 3—Road, resolution: 92 points/m^2 (**Right**).

In order to make a more thorough comparison, the computational performance tests were assessed as much as possible, analyzing the reading, triangulation, and writing times. In those libraries that did not allow for this detailed analysis (e.g., LASTools), an analysis of the total computation time was carried out.

Next, (Figure 4) the graph corresponding to the computation times (in ms) (Y axis) for each of the libraries analyzed is shown, considering the different subsets of points previously indicated (X axis).

Please note that the obtained computation times come from the average computation time obtained for the three study cases using each subset of points. In order to analyze the computation time in the most homogeneous way, the definition of breaklines or geometrical constraints was not considered in this analysis.

In Figure 4 it is shown that some of the studied libraries impose a limit in the number of points to triangulate when they are used with a non-commercial license for scientific purposes (e.g., 500,000 points for Fade2.5D). Others, however, such as CGAL, do not impose any limits for triangulating point clouds in their original resolution. Regarding the efficiency analysis, the Fade2.5D library was the fastest of all analyzed. In contrast, the gDel3D and LasTools triangulation libraries appeared to be less rapid.

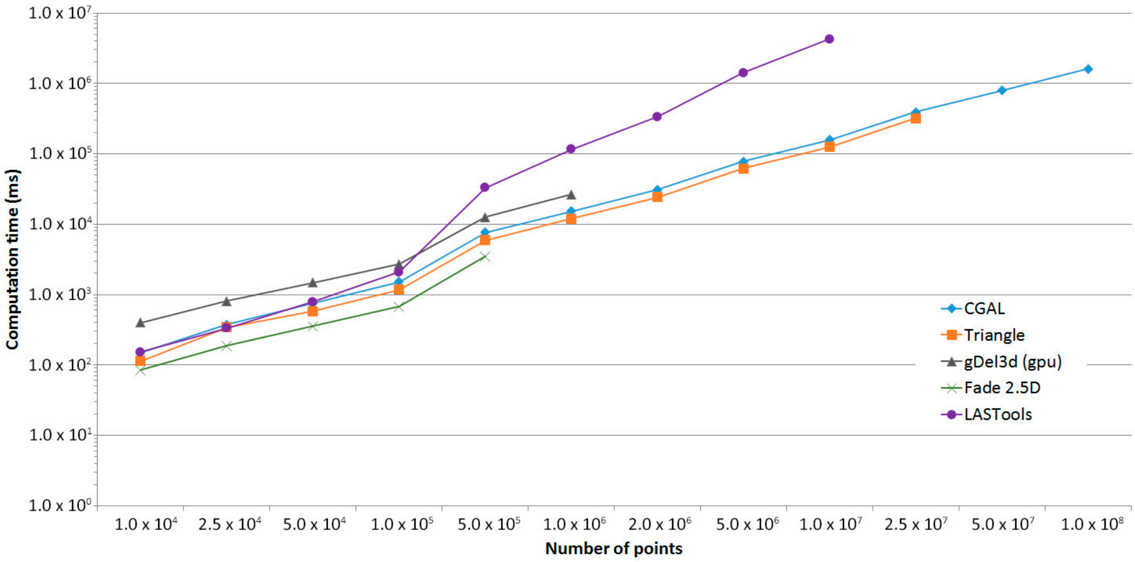


Figure 4. Comparison of triangulation time for the different algorithms used.

3.3. Geometry Quality Analysis

Each triangulation library's results were geometrically assessed in relation to the results obtained by accredited commercial software, "INPHO-DTMaster", which performed as the ground truth. In order to obtain a significant comparison, the comparison process was carried out by selecting three different samples (100 m × 100 m) of the different study cases:

- Plain sample. Involves two areas of flat terrain with small natural obstacles corresponding to the study cases *Zone 3 - Road* and *Zone 2 - Quarry*.
- Buildings/walls sample. Encompasses two areas of buildings or the presence of vertical walls, which were drawn from the study cases *Zone 3 - Road* and *Zone 1 - Archaeology*, respectively.
- Slope sample. Involves two areas of steep terrain with considerable slope that were extracted from the study cases *Zone 1 - Archaeology* and *Zone 2 - Quarry*.

The next figure (Figure 5) outlines the complexity of the triangulation results in these three samples, especially when buildings or trees exist and there are no breaklines or exclusion areas.

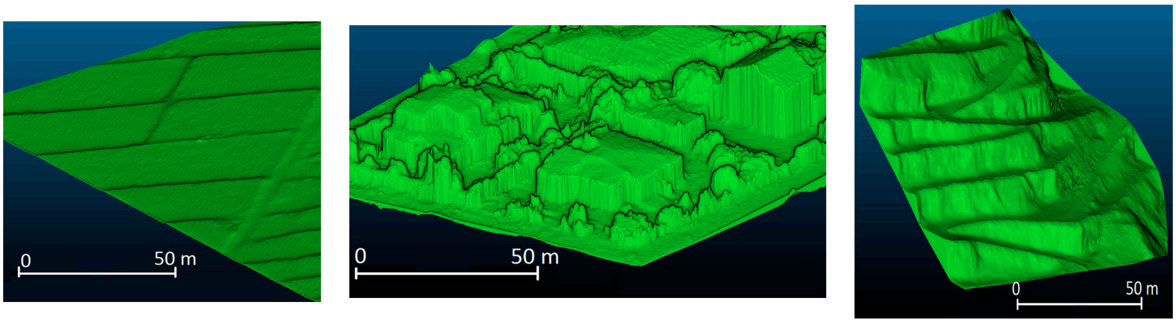


Figure 5. Analysis of the triangulation results using LASTool library in three different samples: (**Left**) plain sample; (**Center**) building and vegetation sample; (**Right**) slope sample.

Besides the computation efficiency, there are no significant differences between libraries when dealing with different samples. Nevertheless, the problems that exist are clear, confirmed in the statistical analysis (Table 1), when buildings or trees are triangulated automatically (Figure 5).

Table 1. Geometric quality based on Gaussian and robust estimators.

	Library	Gaussian Estimators		Robust Estimators		
		Mean (m)	**Standard Deviation (m)**	**Median (m)**	**P_{90}–P_{10} (m)**	**$P_{97.5}$–$P_{2.5}$ (m)**
“Buildings/walls” sample	CGAL	0.0313	±0.5575	0.0000	0.4697	2.4175
	Fade2.5D	0.0313	±0.5575	0.0000	0.4696	2.4175
	LASTools	0.0312	±0.5575	0.0000	0.4695	2.4176
	Triangle	0.0312	±0.5575	0.0000	0.4696	2.4176
“Plain” sample	CGAL	−0.0018	±0.0469	0.0000	0.0088	0.1336
	Fade2.5D	−0.0018	±0.0469	0.0000	0.0088	0.1335
	LASTools	−0.0018	±0.0469	0.0000	0.0088	0.1335
	Triangle	−0.0018	±0.0469	0.0000	0.0088	0.1336
“Slope” sample	CGAL	−0.0016	±0.0855	0.0000	0.0506	0.3159
	Fade2.5D	−0.0016	±0.0855	0.0000	0.0506	0.3159
	LASTools	−0.0016	±0.0855	0.0000	0.0506	0.3159
	Triangle	−0.0016	±0.0855	0.0000	0.0506	0.3159

To facilitate the statistical analysis (Table 1), average values resulting from the two study cases involved in each sample were calculated.

In an initial evaluation of the discrepancies, the central tendency and dispersion, estimated on the basis of a Gaussian hypothesis (mean and standard deviation), yielded a bias of 3 cm and an error dispersion of 55 cm. To check the reliability of these values, a graphical analysis based on qq-plots (Figure 6) was carried out, which allowed us to confirm the non-normality of the data. Thus, to provide a statistical analysis that ensures a better geometric quality assessment of the resulting triangulation, robust estimators of central tendency and dispersion [40] were used. The median was chosen as the robust estimator of the central tendency error. Regarding the dispersion error, since more than 50% of the points errors are less than 0.0001 m (i.e., all the triangulation strategies are based on the same vertices), the median absolute deviation (MAD) estimator is not able to produce significant results for the a priori precision; instead, an interpercentile range between the 2.5th and 97.5th percentiles was chosen, as well as the range between the 10th and 90th percentiles, which yielded the upper and lower error dispersion values for 95% and 80% of the sample points, respectively. These values correspond to 1.96 and 1.282 times the standard deviation for a Gaussian distribution, respectively. This robust approach allows us to exclude from the calculation the selection of statistics under a population hypothesis incompatible with the nature of the sample. However, Table 1 keeps the Gaussian values, so they can be compared with those resulting from the robust estimators, especially in the case of the error dispersion.

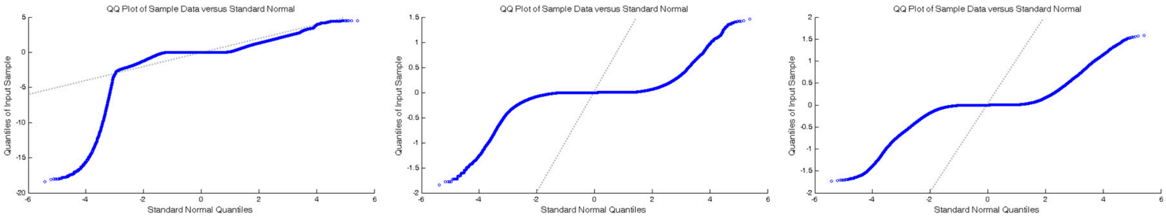


Figure 6. Example of the qq-plots for the CGAL library and the three samples: “Buildings/walls” (**left**); “Plain” (**center**); “Slope” (**right**).

With respect to the central tendency value, the robust measure provided by the median in all cases is zero, implying that no significant bias exists between the different algorithms and the study cases. Regarding the dispersion values, due to the non-normality of the samples, the values coming from the robust analysis should be taken into account for the analysis. It is worth noting that in all cases, there is a correlation between the interpercentile range and the standard deviation. However, the Gaussian relationship is not fulfilled between them, namely the 80% interpercentile range where all the samples are lower than 1.282 times the standard deviation. This is especially relevant for the plain sample, in which the robust dispersion range is more than 13 times better than the Gaussian prediction. In the other cases, it is around 3–4.3 times better when it comes to corroborating the stability and consistency of the methodologies. For the higher triangulation discrepancies, this tendency is minimized, since the Gaussian values fluctuate approximately by a factor of 0.7–1.1, as is shown in Table 1 for the 95% error interval.

4. Conclusions

In this paper, a comparative analysis of different open libraries or libraries which integrate available versions for the scientific community for the triangulation of large point clouds was performed. To carry out this study, an in-house tool, "DE-Delaunay Evaluation", was developed which compasses all the analyzed libraries, also including two modules for the comparative analysis in terms of computational time and geometric quality.

Given the current importance and proliferation of point clouds, which come from different sensors, this paper was designed to provide a comparative view, in terms of the computational efficiency and geometric quality of the final results of the different triangulation libraries. The study cases were chosen selecting representative samples for terrain and infrastructure modeling.

In regard to the geometric quality analysis, there no significant change between libraries, which is logical considering that all of them were executed automatically, without the definition of breaklines, and based on Delaunay principle. The most important difference was found in the computation time as well as the flexibility and ease of implementation of some libraries.

In regard to the volume of information (number of points) and memory management, bearing in mind that the study goal is the triangulation of very-high-spatial-resolution point clouds, it should be noted that the gDel, LASTools and Triangle libraries are not suitable nowadays to work directly with massive volumes of points. On the contrary, the Fade2.5D and CGAL libraries allow more efficient memory management, handling large and massive point clouds. In this regard, it is necessary to clarify that CGAL also suffers from memory management problems, since it does not work efficiently with memory paging. Conversely, Fade2.5D provides greater robustness in this respect, since it allows us to work in memory paging, which affects the algorithm's efficiency.

Regarding the ease of implementation, Fade2.5D stands at a superior level of abstraction compared to CGAL for the developer, allowing the use of its full functionality in a few lines of code. In contrast, the implementation of algorithms in CGAL is more complex, as consultation of the tool documentation and the interpretation and evaluation of complex terminology is necessary.

In relation to scalability with the possible inclusion of other triangulation algorithms, it should be noted that CGAL is presented as a complete library of computational geometry, encompassing powerful triangulation algorithms. Conversely, Fade2.5D incorporates only a variant of 2.5D triangulation, valid for terrain modeling and most civil engineering applications, but it could be limited for modeling complex objects.

Regarding the geometric quality (Table 1), it seems clear, based on the results obtained, that more complex objects such as buildings or walls, especially those that require the definition of breaklines, may entail greater geometric errors. This is caused by the differences between the resulting triangles and the presence of possible gaps within the mesh. The non-normality of the data in the case of large point clouds resulting from automatic processes is also clear, coming from passive or active sensors, and addressing an analysis of the data quality through non-parametric statistics and robust estimators versus classical Gaussian statistics is necessary. Therefore, it is recommended that large point clouds' quality control and their resulting DSM should be carried out initially by graphical procedures (e.g., qq-plots) which account for normal or non-normal data, and then by applying robust estimators (median and interpercentile range) which more reliably assess the geometric data quality.

This study was focused on libraries that can be used freely by any researcher, or at least under academic licenses. For this reason, some limitations in the number of points existed. In the future, these results could be analyzed using commercial licenses in order to deal with bigger point clouds.

Acknowledgments: This work has been partially supported by the company APLITOP through the project "Ánálisis de librerías C++ de triangulación de Delaunay de grandes nubes de puntos". The authors want to thank APLITOP for its confidence and support in advancing this research line.

Author Contributions: All authors conceived and designed the experiments. Luis Lopez-Fernandez carried out the software implementation. David Hernandez-Lopez performed the initial selection of C++ triangulation libraries and provided the study cases. Pablo Rodríguez-Gonzálvez and Damian Ortegal-Terol analyzed the data. Diego Gonzalez-Aguilera wrote the manuscript and all authors read and approved the final version.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, J.; Dowman, I.; Li, S.; Li, Z.; Madden, M.; Mills, J.; Paparoditis, N.; Rottensteiner, F.; Sester, M.; Toth, C.; et al. Information from imagery: ISPRS scientific vision and research agenda. *ISPRS J. Photogramm. Remote Sens.* **2016**, *115*, 3–21. [CrossRef]
2. Vosselman, G.; Maas, H.-G. *Airborne and Terrestrial Laser Scanning*; Whittles: Dunbeath, UK, 2010; Volume 318.
3. González, R.; Llamas, C.; Sánchez, M. Hyperspectral sensors integration in a RPAS investigation aerial platform. *Aerosp. Sci. Technol.* **2016**, *58*, 217–224. [CrossRef]
4. Drusch, M.; Del Bello, U.; Carlier, S.; Colin, O.; Fernandez, V.; Gascon, F.; Hoersch, B.; Isola, C.; Laberinti, P.; Martimort, P.; et al. Sentinel-2: ESA's optical high-resolution mission for GMES operational services. *Remote Sens. Environ.* **2012**, *120*, 25–36. [CrossRef]
5. Kraus, K. *Photogrammetry: Geometry from Images and Laser Scans*; Walter de Gruyter: Berlin, Germany, 2007.
6. Egels, Y.; Kasser, M. *Digital Photogrammetry*; CRC Press: Boca Raton, FL, USA, 2003.
7. Herrero-Huerta, M.; Felipe-García, B.; Belmar-Lizarán, S.; Hernández-López, D.; Rodríguez-Gonzálvez, P.; González-Aguilera, D. Dense canopy height model from a low-cost photogrammetric platform and LiDAR data. *Trees* **2016**, *30*, 1287–1301. [CrossRef]
8. Haala, N.; Kada, M. An update on automatic 3D building reconstruction. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 570–580. [CrossRef]
9. Tang, X.; Liu, Y.; Zhang, J.; Kainz, W. *Advances in Spatio-Temporal Analysis*; CRC Press: Boca Raton, FL, USA, 2007.
10. Bryan, B.A. High-performance computing tools for the integrated assessment and modelling of social-ecological systems. *Environ. Model. Softw.* **2013**, *39*, 295–303. [CrossRef]
11. Yastikli, N.; Cetin, Z. Classification of LiDAR data with point based classification methods. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B3*, 441–445. [CrossRef]
12. González-Aguilera, D.; López-Fernández, L.; Rodriguez-Gonzalvez, P.; Guerrero, D.; Hernandez-Lopez, D.; Remondino, F.; Menna, F.; Nocerino, E.; Toschi, I.; Ballabeni, A.; et al. Development of an all-purpose free photogrammetric tool. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B6*, 31–38. [CrossRef]
13. Peucker, T.K.; Fowler, R.J.; Little, J.J.; Mark, D.M. The triangulated irregular network. In Proceedings of the Digital Terrain Models Symposium, St. Louis, MO, USA, 9–11 May 1978; pp. 96–103.
14. Blandford, D.K.; Blelloch, G.E.; Cardoze, D.E.; Kadow, C. Compact representations of simplicial meshes in two and three dimensions. *Int. J. Comput. Geom. Appl.* **2005**, *15*, 3–24. [CrossRef]
15. Blelloch, G.E.; Miller, G.L.; Hardwick, J.C.; Talmor, D. Design and implementation of a practical parallel Delaunay algorithm. *Algorithmica* **1999**, *24*, 243–269. [CrossRef]
16. Chen, M.-B.; Chuang, T.-R.; Wu, J.-J. Parallel divide-and-conquer scheme for 2D Delaunay triangulation. *Concurr. Comput. Pract. Exp.* **2006**, *18*, 1595–1612. [CrossRef]
17. Yoon, S.-E.; Lindstrom, P.; Pascucci, V.; Manocha, D. Cache-oblivious mesh layouts. In Proceedings of ACM SIGGRAPH 2005, Los Angeles, CA, USA, 31 July–4 August 2005; pp. 886–893.
18. Isenburg, M.; Liu, Y.; Shewchuk, J.; Snoeyink, J. Streaming computation of Delaunay triangulations. In Proceedings of ACM SIGGRAPH 2006, Boston, MA, USA, 30 July–3 August 2006; pp. 1049–1056.
19. Vitter, J.S. External memory algorithms and data structures: Dealing with massive data. *ACM Comput. Surv.* **2001**, *33*, 209–271. [CrossRef]

20. Tu, T. A Scalable Database Approach to Computing Delaunay Triangulations. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2008.
21. Isenburg, M.; Lindstrom, P. Streaming meshes. In Proceedings of the 16th IEEE Visualization Conference VIS 05, Minneapolis, MN, USA, 23–28 October 2005; pp. 231–238.
22. Allègre, R.; Chaine, R.; Akkouche, S. A streaming algorithm for surface reconstruction. In Proceedings of the Symposium on Geometry Processing, Barcelona, Spain, 4–6 July 2007; pp. 79–88.
23. Agarwal, P.K.; Arge, L.; Yi, K. I/O-Efficient Construction of Constrained Delaunay Triangulations. In *Algorithms—ESA 2005, Proceedings of the 13th Annual European Symposium, Palma de Mallorca, Spain, 3–6 October 2005*; Brodal, G.S., Leonardi, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 355–366.
24. Isenburg, M. LAStools—Efficient Tools for LiDAR Processing. Available online: https://rapidlasso.com/lastools/ (accessed on 27 October 2016).
25. Fade2.5D. Available online: http://www.geom.at/fade2d/html/ (accessed on 27 October 2016).
26. Shewchuk, J.R. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry towards Geometric Engineering, Proceedings of the FCRC'96 Workshop, WACG'96, Philadelphia, PA, USA, 27–28 May 1996*; Selected Papers; Lin, M.C., Manocha, D., Eds.; Springer: Berlin/Heidelberg, Germany, 1996; pp. 203–222.
27. Shewchuk, J.R. Delaunay refinement algorithms for triangular mesh generation. *Comput. Geom.* **2002**, *22*, 21–74. [CrossRef]
28. Triangle. Available online: http://www.cs.cmu.edu/~quake/triangle.html (accessed on 27 October 2016).
29. Ruppert, J. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms* **1995**, *18*, 548–585. [CrossRef]
30. Lawson, C. Software for C1 surface interpolation. In *Mathematical Software III*; Rice, J.R., Ed.; Academic Press: New York, NY, USA, 1977; pp. 161–194.
31. Yvinec, M. CGAL User and Reference Manual. In CGAL 4.9—2D Triangulation; 2016. Available online: https://doc.cgal.org/latest/Triangulation_2/index.html (accessed on 29 December 2016).
32. CGAL. Available online: http://www.cgal.org/ (accessed on 27 October 2016).
33. Boissonnat, J.-D.; Devillers, O.; Pion, S.; Teillaud, M.; Yvinec, M. Triangulations in CGAL. *Comput. Geom.* **2002**, *22*, 5–19. [CrossRef]
34. Ye, S.; Daniels, K. Hierarchical Delaunay Triangulation for Meshing. In *Experimental Algorithms, Proceedings of the 10th International Symposium, SEA 2011, Kolimpari, Chania, Crete, Greece, 5–7 May 2011*; Pardalos, P.M., Rebennack, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 54–64.
35. Nanjappa, A. Delaunay Triangulation in R3 on the GPU. Ph.D. Dissertation, Visvesvaraya Technological University, Belgaum, India, 2012.
36. Cao, T.-T.; Nanjappa, A.; Gao, M.; Tan, T.-S. A GPU accelerated algorithm for 3D Delaunay triangulation. In Proceedings of the 18th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, San Francisco, CA, USA, 14–16 March 2014; pp. 47–54.
37. gDEL3D. Available online: http://www.comp.nus.edu.sg/~tants/gdel3d.html (accessed on 27 October 2016).
38. Shewchuk, R. Star splaying: An algorithm for repairing Delaunay triangulations and convex hulls. In Proceedings of the Twenty-First Annual Symposium on Computational Geometry, Pisa, Italy, 6–8 June 2005; pp. 237–246.
39. Trimple Inpho. DTMaster. Available online: http://www.trimble.com/imaging/inpho.aspx (accessed on 27 October 2016).
40. Rodríguez-Gonzálvez, P.; González-Aguilera, D.; Hernández-López, D.; González-Jorge, H. Accuracy assessment of airborne laser scanner dataset by means of parametric and non-parametric statistical methods. *IET Sci. Meas. Technol.* **2015**, *9*, 505–513. [CrossRef]