

---

# Triangulation-free 3D Reconstruction from LiDAR Data

---

Yusuf Sahillioğlu

YUSUF@ALUMNI.UFL.EDU

Dept. of Computer & Inf. Science & Eng., Univ. of Florida, Gainesville, USA

## Abstract

We demonstrate a generic method for visualization of high-resolution unorganized and noisy 3D data points with a surface of significantly lower resolution. To this effect, we feed our algorithm with all the go Light Detection And Ranging (LiDAR) data and approximate it with a triangular mesh that is being deformed iteratively until convergence. Our main contribution is avoiding the direct triangulation of LiDAR points, a cumbersome step common in literature.

## 1. Introduction

Light Detection And Ranging (LiDAR) systems are capable of acquiring data to produce accurate digital elevation models by allowing the positioning of the footprint of a laser beam as it hits an object. The laser produces an optical pulse that is transmitted, reflected off an object, and returned to the receiver. The receiver accurately measures the travel time of the pulse from its start to its return. With the pulse traveling at the speed of light, the receiver senses the return pulse before the next pulse is sent out. Since the speed of light is known, the travel time can be converted to a range measurement. Combining the laser range, laser scans angle, laser position (from Global Positioning System – GPS), and laser orientation (from Inertial Navigation System – INS), accurate spatial object coordinates can be calculated for each laser pulse (Mosaic, 2001). While we focus on forest modeling here, there are plenty of other applications that are fed by LiDAR data, such as flood risk mapping, oil/gas explorations, real estate development, coastal zone mapping, and urban modeling. Our problem is, given a set of unorganized and noisy 3D points acquired by a LiDAR system, approximate the underlying surface by a 2-manifold triangular mesh. To this end, we employ a surface deformation framework and end up with a low-resolution mesh from the high-resolution LiDAR points. Our contribution is to avoid the computationally expensive 2D triangulation of the projected LiDAR points.

## 2. Related Work

LiDAR becomes an important and convenient data source. Many researchers are developing algorithms to extract a bare-earth model and building boundaries from the LiDAR data. A method for building reconstruction with LiDAR data (Sohn et al., 2004) combines filtered LiDAR data and cadastral building boundaries data, while another one (Tse et al., 2005) uses the Voronoi Diagram to trace building outlines, for which they extrude buildings via Computer Aided Design (CAD)-type Euler Operators to create a Triangulated Irregular Network (TIN) model and then applies these operators to modify the TIN, e.g., extrude buildings. As far as the bare-earth model generation, (Baligh et al., 2008) apply the wavelet denoising method followed by a convenient filtering for the extraction. Applicable to both bare-earth and building modeling, most researchers perform 2D triangulation (Bourke, 1989) of the projection of LiDAR data, followed by an elevation to 3D using the height information available. An optional edge collapse operation with convenient error metrics (Garland et al., 1997) on 2D triangulation helps building fair reconstructions in sufficiently low resolutions.

## 3. Algorithm

The deformation based algorithm that brings out the underlying shape of the input LiDAR data  $L$  is adapted from (Sahillioğlu et al., 2006). We guide their surface deformation framework by the tangent planes attached to some of the LiDAR points.

### 3.1 Tangent Plane Computation

Having subdivided the bounding box of  $L$  into a grid of cubes, we select a set of  $k$ -nearest neighbors of a point  $p$  inside the cube, which lets us fit a plane  $K$  to this set in the least squares sense, where  $K$  eventually gets assigned to  $p$  (see Figure (1)). For efficiency reasons, we deny the tangent plane computation for the points which already appear in the  $k$ -nearest neighbors list of a point with an assigned plane by assuming that there already exists a plane for them.

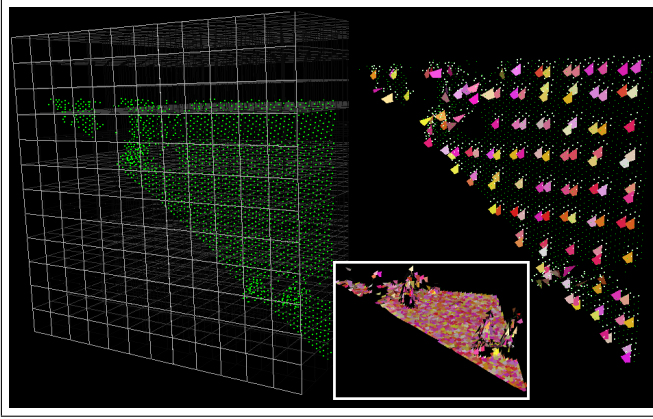


Figure 1. Grid of cubes (left) for fast tangent plane assignment (right). The larger the number of the tangent planes, the better the surface estimation is (framed in white).

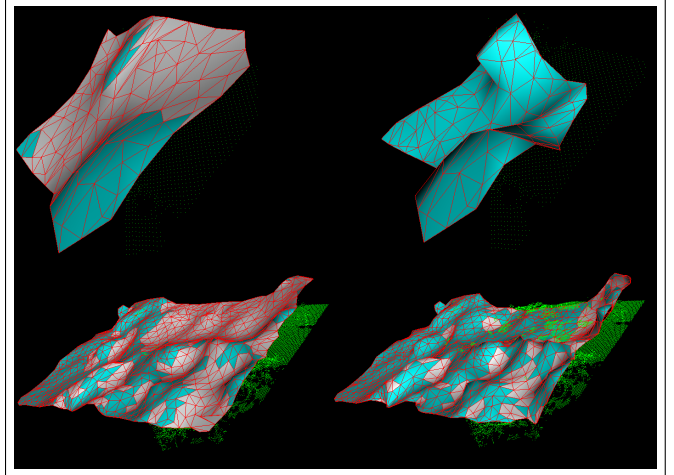


Figure 2. Low- (top) and high-resolution (bottom)  $M$  in deformation. White vertices are active whereas cyans are frozen. The higher the resolution of  $M$ , the better its evolution.

The accuracy and speed of the system depends on the cube size which is determined based on how high the resolution is wanted to be. Note also that, the tangent planes in use may be inconsistently oriented (Hoppe et al., 1992), but this problem is handled easily in our case thanks to our a priori that the mesh to be evolved,  $M$ , is initially outside  $L$ .

### 3.2 Deformation

In each iteration of the deformation process that evolves  $M$  towards  $L$ , each unfrozen/active vertex is moved in the direction of its normal with a conservative magnitude that prevents the formation of non-manifold triangles. In order to keep  $M$  as smooth as possible after moving all vertices, we apply the non-shrinking mesh fairing operation. The minimum edge length allowed on  $M$  is guaranteed to be  $\epsilon$  after edge collapses, and  $\epsilon/2$  is used as the magnitude of movement. Edge split operation, on the other hand, divides a long edge into two by introducing a new point in the middle which then lets  $M$  evolve further towards the concavities that would have never been reached otherwise. Edge flips merely keep the valences of vertices close to 6 for regularity purposes. These local mesh transform operations (Hoppe et al., 1993) as well as the fairing (Taubin, 1995) keeps  $M$  in a good shape at all times.

A vertex  $v$  is frozen up as soon as it passes to the other side of the tangent plane  $K$  of the closest LiDAR point  $p \in L$ . When a vertex is frozen/deactivated, a fine-tuning process follows in order to refine its final position; a binary search performed on the line segment with starting and final positions as endpoints, and  $v$  is pulled back to the point closest to  $K$ . Figure (2) shows deformation of  $M$  in progress.

### 3.3 Computational Complexity

The grid-based  $k$ -nearest neighbors for fast tangent plane computation is  $O((|L|/C)k)$  per grid cube and accumulates to  $O(|L|k)$  when summed over all cubes with  $C$  being the number of cubes and underlying LiDAR data distribution assumed to be uniform. Since  $k$ -nearest neighbors is repeated  $(|L|/C)/k$  times per cube, this preprocessing step indeed requires  $O(|L|k \times (|L|/C)/k = L^2/C)$  time. Once the guide tangent planes are made available, it takes  $O(|L|V_M)$  to perform one deformation iteration because all  $V_M$  vertices of the evolving mesh  $M$  need to find the closest LiDAR point to be able to realize whether they passed to the other side of the target tangent plane, or equivalently the target LiDAR surface. The number of deformation iterations depend on the distance of the initialization from the target LiDAR surface and is expected to be bounded by a small constant, e.g., 20 in our experiments, which renders the deformation complexity as  $O(|L|V_M)$ , and the overall complexity becomes  $O(L^2/C)$ .

## 4. Results and Discussion

We use the LiDAR data from (PSLC, 2007) and perform the traditional, i.e., direct triangulation based, generation of LiDAR surface as well to compare our surface estimation results with (see Figure (3)). Although the traditional one in its simplest form cannot reduce the resolution of the output, our solution can do so by simply initializing  $M$  as a low-resolution 2-manifold. Another advantage of our method is the clearance of the need of messing up with the direct triangulation of the noisy and huge LiDAR data.

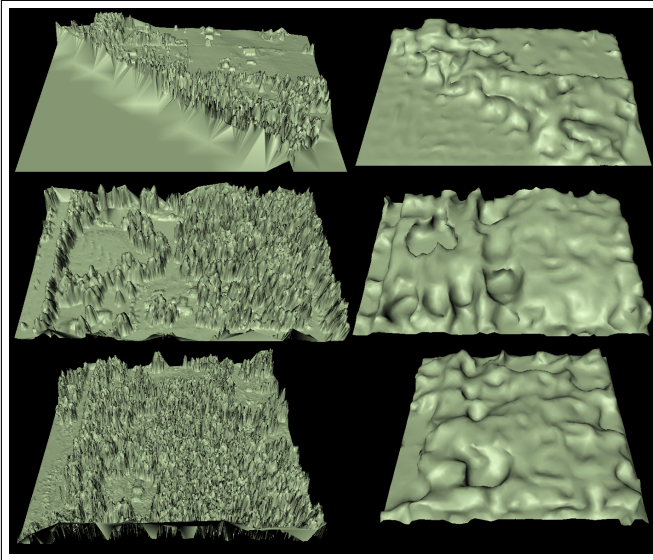


Figure 3. Three different target surfaces of 100K LiDAR points (left) approximated by meshes of 3K vertices (right).

## 5. Conclusions

We have described an iterative deformation-driven algorithm that is capable of being fed with unorganized and noisy data. In particular, we work with the LiDAR points although all data regardless of its size, connectivity type and noise level are quite welcome. Starting with a relatively low-resolution 2-manifold triangular mesh of a flat plane pointed towards the target surface implied by the LiDAR points, we iteratively deform it until all vertices are stabilized, i.e., convergence. With this scheme, we save ourselves from the computationally expensive 2D triangulation required by most of the related work. Besides, the resolution of the output surface can be controlled easily as the initial resolution of the evolving mesh stays the same at the time of convergence.

## References

- Mosaic Mapping Systems Inc. (2001). A White Paper on LiDAR Mapping.
- G. Sohn and I. J. Dowman (2004). Extraction of buildings from high resolution satellite data and LIDAR. *ISPRS/WGIII Automated Object Extraction*.
- R. Tse, M. Dakowicz, C. Gold & D. Kidner. (2005). Building Reconstruction Using LiDAR Data. *Dynamic and Multi-dimensional GIS* (pp. 156–161).
- A. Baligh, M. J. Valadan Zoej & A. Mohammadzadeh (2008). Bare Earth Extraction from Airborne LIDAR Data Using Different Filtering Methods. *ISPRS/WGIII*

*Processing of Point Clouds from Laser Scanners and Other Sensors*.

- P. Bourke. (1989). Triangulate: Efficient Triangulation Algorithm Suitable for Terrain Modelling. *Pan Pacific Computer Conference*.
- M. Garland and P. Heckbert. (1997). Surface Simplification Using Quadratic Error Metrics. *ACM SIGGRAPH* (pp. 209–216).
- Y. Sahillioglu and Y. Yemez. (2006). A Surface Deformation Framework for 3D Shape Recovery. *Multimedia Content Representation, Classification and Security, International Workshop (MRCSS)* (pp. 570–577).
- H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, & W. Stuetzle. (1992). Surface Reconstruction from Unorganized Points. *ACM SIGGRAPH* (pp. 71–78).
- H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, & W. Stuetzle. (1993). Mesh Optimization. *ACM SIGGRAPH* (pp. 19–26).
- G. Taubin. (1995). A Signal Processing Approach to Fair Surface Design. *ACM SIGGRAPH* (pp. 351–358).
- Puget Sound LiDAR Consortium. (2007). LiDAR Bare Earth and Top Surface DEM. The woodlands, TX: Terrapoint. Available: <http://pugetsoundlidar.ess.washington.edu/index.htm>. *PSLC Data*.