3D Modeling of Roofs from LiDAR Data using the ESRI ArcObjects Framework

Nadeem Kolia

Thomas Jefferson High School for Science and Technology
August 25, 2005

Mentor: Luke A. Catania, ERDC-TEC

Abstract

The goal of this project was to develop a program that would automate the process of generating high resolution three dimensional building models from Light Detection and Ranging (LiDAR) Elevation data. The program was developed in Microsoft Visual Basic (VB) and ArcObjects as a Dynamic Linked Library (DLL) for ArcScene 3D. The program relies upon 2D footprints previously generated by the Advanced LiDAR Exploitation System (ALES), but requires no further user interaction or input.

Introduction

LiDAR data is being collected by the army to aid in campaign and mission planning. Commercial ArcGIS software is capable of representing LiDAR data in three dimensions, however only an outline of buildings is visible and the true structure of the building is masked. ALES is capable of generating 2D footprints of buildings which can then be extruded to the average height of building to give it a 3D appearance and a flat roof. However, most buildings have more complex roof types and thus flat building models would be insufficient. Extracting high resolution building models from LiDAR data would provide information about the urban environment that would be essential in the planning operations. The LiDAR Modeling System (LMS) developed by the University of Southern California (USC) is capable of extracting the entire 3D geometry of a building, but requires the user to select the roof type and points that define the specified roof type. Thus, it is slow and time consuming and not feasible for rapidly modeling an urban area. Software is needed that will automate the process of extracting building models with the need of little to no user interaction.

Background

The army deploys aerial laser scanning to quickly collect data about an urban environment that could be used to help in mission planning. LiDAR data was collected over several different areas, including San Francisco, Baltimore, Washington D.C Mall, Fort Benning, and the Engineering Research and Development Centers (ERDC) Topographical Engineering Center (TEC). The LiDAR data is converted to a raster with elevation values for every meter accurate to within 10 centimeters.

The raster can be loaded into ESRI ArcGIS software such as ArcMap and ArcScene. Both ArcMap and ArcScene are capable of classifying the elevation data on a grayscale and provide a way of visualizing the raster data. Additionally, ArcScene is capable of visualizing the data in three dimensions by extruding the raster values. ArcObjects can be used to develop custom tools that take advantage of the built in features and capabilities of ArcScene.

Procedure

A program had to be designed and developed to implement an algorithm for generating building models. Instead of making a program that ran over the entire raster dataset, it would be more efficient to run for each building. Before this could be done, individual buildings had to be isolated from the raster for future processing. This required another program that could intelligently and accurately recognize and eliminate extraneous terrain data. Instead of creating a new program for this specific task, the 2D footprints generated by ALES were adapted to this task. Using 2D footprints to isolate one building at a time lowered the memory requirements and decreased the processing time required for modeling buildings.

The LMS software that is capable of modeling buildings with user interaction requires two inputs from the user. First it requires the user to recognize and select the type of roof that will be modeled, and then requires the user to select a few key points that could be used to generate the model. The number and type of key points required depends on the roof type. For example, modeling a sphere would require two key points. One point would define the center of the sphere; the other key point would be an arbitrary key point on the surface of the sphere and would in essence be providing information about the radius of the sphere. With the center and the radius of the sphere, it would be fairly easy to model. For a flat roof surface, as many points as there are edges in the polygon defining the roof are required to accurately create a model of the roof. Since the program will automate this process, it must replace the user and be able to first recognize a particular roof type and then find the points that define that particular roof type.

Though only given elevation data, there are two sets of data values which are useful for this purpose. They are elevation or height values, which are given, and slope values, which can be easily calculated. In fact the 3D Analyst Toolbox in ArcScene provides a tool which is capable of calculating the maximum slope in any direction from a particular point, or for all the points in the raster dataset.

Now that we have height and slope datasets easily accessible as well as the area which we wish to model, we can begin to implement an algorithm that will recognize the roof as a particular type and then find the key points to define the roof.

The roof types considered shall be generalized as being a flat roof, slanted roof, peaked roof, cylinder, tower, sphere, or ellipsoid. A flat roof has a polygonal base and is extruded to the same height for each point of the base. A slant roof has a rectangular base with one side higher than the opposite and the adjacent sides of equal height. A peaked roof is two slanted roofs that meet at their peeks, the roof typical of most houses. A cylinder is a flat roof with a circular base. A tower is similar to a peaked roof except more than two slanted roofs meet at one high point. The sphere and the ellipsoid can be parts of their respective shape, and are usually so. Many buildings, especially larger ones, employ various combinations of these roofs and therefore the shapes must be recognized and modeled separately. Furthermore, a hole may exist within the roof, which adds another dimension to the problem.

The first approach used in modeling the buildings relied solely upon height data and did not take advantage of slope data available. The first step in this process was to break up the region of interest into groups of continuous points with the same height. Because roofs are not made perfectly and the measurements taken by the LiDAR instrument is only accurate to within 10cm, a height tolerance had be used to distinguish points of the same height. If the two heights have a difference less than or equal to the height tolerance, then they can be considered to be of the same height and can be grouped together.

Next, for each group of points a convex hull is created. This significantly reduces the number of points in the group and creates a simply polygon that ArcGIS is capable of drawing. Once the convex hull has been created, an intersection is taken with the original area that was selected for modeling. This is necessary because the data originally analyzed in order to find groups of points was larger than the area of interest. This is because a clip created from the raster data using a 2D footprint or user input will be rectangular and lie along the coordinate axis of the raster. However, the models dont usually lie along the coordinate axis and dont necessarily have a rectangular shape, thus the actual area being modeled is larger and encompassing of the building itself. (See Figure 1) By taking the intersection of the convex polygon with the building the area outside the building but inside of the raster clip is eliminated. (See Figure 2)

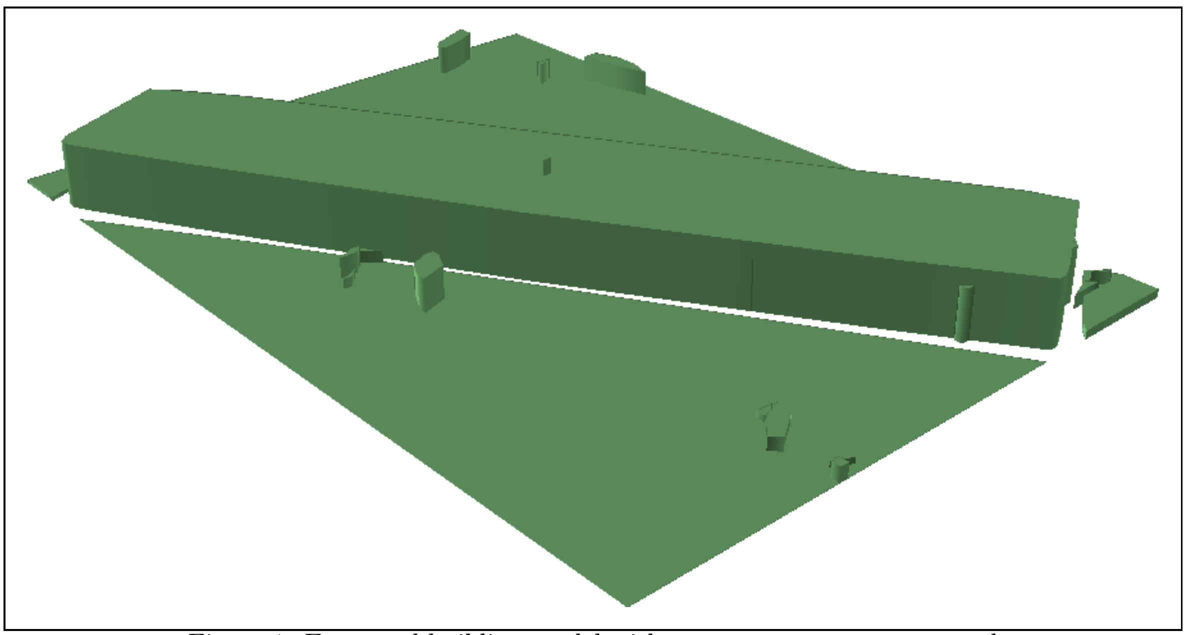

Figure 1: Extracted building model without extraneous area removed

The convex polygons are indicative of the shell of a feature in the roof of the building, but dont contain information about the inside of the roof and thus obscure the true form of the building. To add details of the interior of the roof to the building model I added points to each convex hull. For each vertex of the convex hull, the highest point that could be

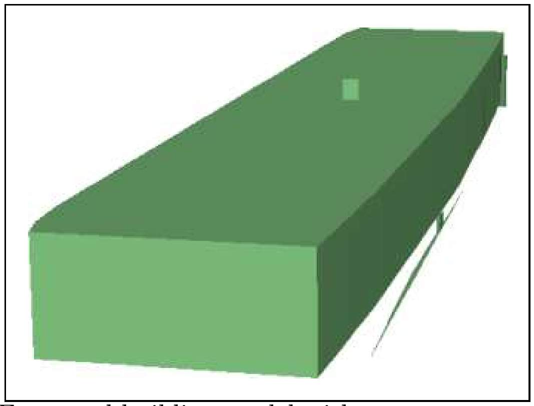

Figure 2: Extracted building model with extraneous area removed

reached by taking the steepest path from that vertex was added to the convex hull. ArcGIS uses these interior points to create a polygon that defines the interior shape of the polygon. The result is a polygon donut, the interior points will create a hole in the convex hull we had before. (See Figure 3)

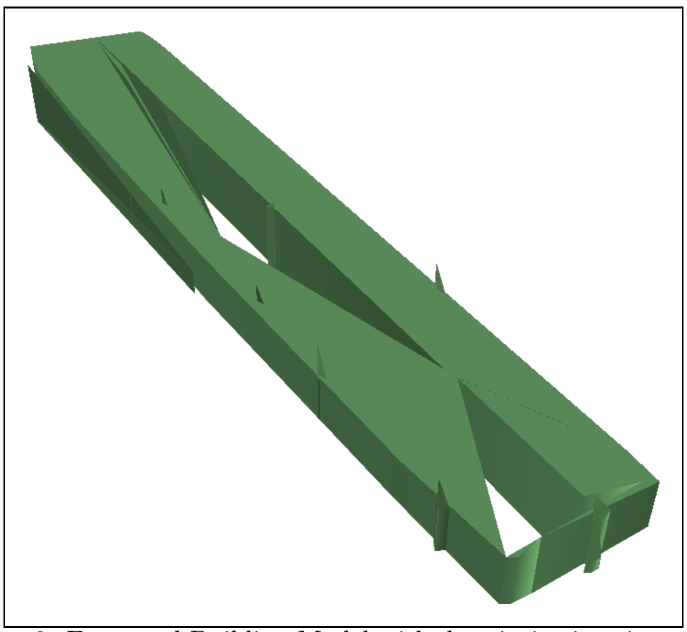

Figure 3: Extracted Building Model with descriptive interior points

Despite the obvious drawback of having a hole in the model, the hole does model the interior of the building and increases the accuracy of the polygon model. The first attempt to fill the hole was basically creating a new polygon with just the interior points that were added later on. This worked, but not perfectly, for ArcGIS used a convex hull for the interior solid and thus the polygon did not completely fill the hole. (See Figure 4)

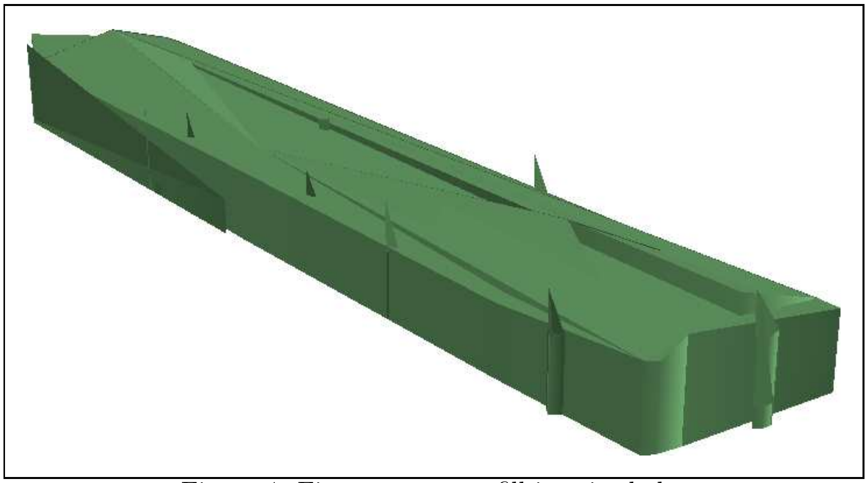

Figure 4: First attempt to fill interior holes

Upon inspection of the donut polygon, it became obvious that ArcGIS had modeled the shape as one polygon as an incomplete ring instead of two disjoint polygons one for the outside and one for the inside. ArcGIS has the capability of defining each polygon as a collection of rings which is closer to the polygon taught in mathematics. A polygon created from a collection of rings can have holes defined by exterior and interior rings. Very simply, area inside an interior ring is considered a hole in the polygon, while area inside an exterior ring is part of the solid polygon. So now the original convex hole was set as the exterior ring of one polygon, and a convex hull of the interior points was set as the interior ring of the same polygon. This creates a true donut shape, but it more or less similar to the holed polygon from before. Another polygon is created with the exterior polygon set to the convex hull of the interior points. This effectively and exactly fills the hole.

The model now consisted of a completely solid shape that had both interior and exterior features. There was still a problem, not all of the interior points previously discovered were being represented only the ones that formed the interior convex hull were affecting the model. Furthermore, the most interior points most often the points that were also the highest and thus would improve our model the most were the ones being left out.

To fix this, the process of creating a polygon from an exterior and interior ring was modified slightly to run as many times as possible. First using the collection of points that is the vertices of the original convex hull and all the interior points, a convex hull is created and set as the exterior ring of a new polygon. Then, these points are removed and a new convex hull is created and set as the interior ring of the same polygon. These two steps are repeated on the remaining points an indefinite number of times, until there are so few points remaining that a convex hull can no longer be created. This improves greatly upon the old process because it allows for as much detail as possible for the modeling of the interior roof of the building. (See Figure 5)

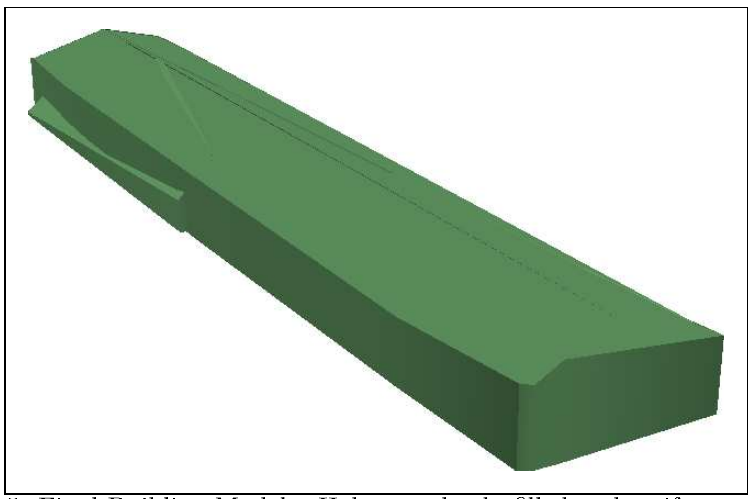

Figure 5: Final Building Model – Hole completely filled and artifacts removed

There are several aspects of this technique that could be improved upon and the technique is far from being capable of accurately modeling all of the various types of building roofs. It cant recognize and model a hole in the center of any building, no matter of what shape. It is also not able to model a sphere or ellipsoid realistically. Another problem with the models generated is that many small shapes are added to each model along with the few major shapes actually necessary. The smaller models add extrusions and fine detail but obscure and hinder the display of the more important major shapes. These number of these smaller models can be reduced my removing shapes smaller than a certain area. However, care must be taken to ensure the minimum required area is not set too high, for this will remove shapes that may be essential to a building model.

Conclusion

The process explained above is successful in modeling buildings to an extent. It does correctly differentiate between several different roofs types in one building, and will model each of them separately. Its major shortcomings come from its lack of ability to actually recognize the type of roof it is modeling. Because of this the program is unable to refine models by snapping them to key points. It is also unable to remove all extraneous shapes that are already modeled within larger shapes and are unnecessary detailed. The benefit of this approach is that it is capable of creating models for roofs not originally considered and is more versatile than a program that would try to characterize a roof from a built-in set of roofs.

To improve upon this process slope values should be taken into consideration, and the program should attempt to recognize roof types. This would be especially beneficial in modeling spheres since spheres can't be easily created from a group of polygons. It would also provide for the identification and modeling of holes, a roof feature currently ignored by the program.

Acknowledments

First off, I would like to thank my family for always supporting me. I would like to thank my mentor Luke Catania for giving me the opportunity to work on this project, being patient and helpful as I attempted to learn VB and ArcObjects, allowing me the flexibility to tackle the problem in my own way, and helping me design and develop my algorithm. Thanks to

Chris Gard for giving me additional feedback and direction with my project and Tom Witte for providing humor at work. Thanks to Branch Chief Vineet Gupta for his support and encouragement. Thanks to the rest of the SEAP students here: Raamin Mostaghimi, Dmitry Portnoy, and Sebastian Wood for making work so much more enjoyable and interesting. Finally, thanks to the countless others at TEC who helped me with an innumerable number of little things.