Automated Terrain Generation Using LIDAR and Waterbody Survey Data

Maik Flanagin
University of New Orleans
maik@cs.uno.edu

Aurélien Grenotton
University of New Orleans
agrenott@cs.uno.edu

Jay Ratcliff
U.S. Army Corps of Engineers
jay.ratcliff@usace.army.mil

Mahdi Abdelguerfi
University of New Orleans
mahdi@cs.uno.edu

Kevin Shaw
Naval Research Laboratory
shaw@nrlssc.navy.mil

ABSTRACT
LIDAR data provides a dense and precise sampling of ground elevations on land, but not of bathymetric elevations in a waterway. By supplementing LIDAR data with densely interpolated waterway survey data, terrain models can be generated to approximate LIDAR-derived topography combined with waterway bathymetry. Three-dimensional interpolation techniques are used to generate a dense waterway bathymetry, as well as the bounding region to clip and fill the LIDAR data with this bathymetry. Consequently, the resulting terrain is fit for both visualization and further processing in hydraulic and hydrological models.

Keywords: LIDAR, visualization, terrain-modeling.

1. OVERVIEW
Hydraulic and hydrological models are essential to the planning process for major flood control projects. Hydraulic models are used for many tasks including computing watershed basin flows, channel flood stages, and predicting flood impacts for real events and hypothetical scenarios. Two such models used extensively by the U.S. Army Corps of Engineers are the Hydrological Engineering Centers River Analysis System (HEC-RAS)[1], which computes water surface elevations and flow rates in river channels, and ADCIRC[2], an advanced three-dimensional circulation modeling system that predicts the effects of hurricane storm surge on land.

Naturally, for any model to produce satisfactory results, it must be based on accurate data. In the case of hydraulic and hydrological models, the most important dataset is the ground elevation. Particularly important is the elevation data in waterways, also known as bathymetry. Herein lies a fundamental problem: as of this writing, there seems to be no reasonable way to acquire elevations in large waterway-riddled areas, with which to build a complete, dense, and accurate digital elevation model (DEM).

2. RELATED WORK
The U.S. Army Corps of Engineers has tried a few different approaches to constructing digital elevation models for its projects. Some approaches involve sophisticated remote sensing systems such as Scanning Hydrographic Operational Airborne LIDAR Survey (SHOALS)[3], which performs LIDAR data collection in coastal areas both on-shore and in the ocean. Of course, while useful for coastal models, the technology used in SHOALS is not yet at a point where data can be collected from rivers, bayous, or canals, all of which are typically much muddier than the ocean. Other approaches involve merging multiple data sources to produce a single, composite model. A good example of this approach is the New Orleans district’s GeoAce program[4], in which models can be generated to visualize and analyze river bottoms by applying a Delaunay triangulation[5] to the points from both LIDAR and SONAR datasets. Although such a merge may be the ideal approach to modeling, SONAR data is not be available for all waterways of interest due to cost and logistic issues.

3. LIDAR-SURVEY HYBRID TERRAIN MODEL
The alternative approach combines LIDAR with survey data taken underwater. The problem with this approach is that while LIDAR is significantly dense (sample points could average a one-foot spacing), survey data is very sparse. Sim-
ply concatenating the two sources results in a useless surface, virtually indistinguishable from the original LIDAR. If only survey data were somehow more dense, then combining it with LIDAR would prove as simple as combining SONAR with LIDAR.

This very concept forms the basis of the LIDAR-survey hybrid terrain models. First, a waterway generation algorithm creates a mesh representing the surface of the waterway with a density approximating that of the LIDAR data. Afterwards, LIDAR points whose coordinates fall within the mesh’s perimeter are filtered out and the remaining LIDAR points are concatenated to the mesh points and triangulated. The result is a complete terrain model of both ground surface and riverbed geometry.

### 3.1 Waterway Generation

The interpolation algorithm generates waterways channels using a spline-based approach[6]. The input data sources for the waterway generation derive from cross-section and profile survey data as shown in Figure 1. Cross-sections are survey measurements usually taken from one bank of a waterway to the opposite bank in a straight line. Ideally, these cross-sections are nearly perpendicular to the path of the river.

Profile centerlines are surveys taken along the length of the river, typically in middle of the channel between both banks. Sampling is usually dense so as to define the shape of the channel’s path. Unfortunately, some profile surveys are not taken and must be created by tracing a waterway’s path on an aerial photograph using a CAD program.

![Figure 1: Example of cross-sections and profile centerline.](image)

#### 3.1.1 Waterway Spline Construction

The first step to turn this data into a waterway mesh is to resample these inputs so as to achieve the desired level of mesh detail. In addition to the cross-section and profile data sources, the number of cross-section and profile samples must also be provided to the algorithm in order to determine the number of points to generate. A three-dimensional spline is generated from each cross-section and evaluated at even spacings. Effectively, all sampled cross-sections have the same number of points. Similarly, a three dimensional spline is created for the profile and evaluated at even-spacing to generate the profile to be used for interpolation. In the interpolation, a cross-section is generated at each sampled profile point.

To properly interpolate the cross-sections along the profile, it must be established where the cross-sections intersect the profile line. It is at these intersection points that the α values are chosen and associated with their corresponding cross-sections in order to build the interpolation splines. As shown in lines 1-4 of Algorithm 1, the α value is computed as the Euclidean distance along the profile path from the first end point to the point of intersection divided by the total length of the profile. Cross-sections that do not intersect the profile at any point are not used to compose the spline.

Once the cross-sections are either associated or filtered out, the spline generation can begin. The process proceeds as described in lines 5-11 of Algorithm 1. Since each sampled cross-section has an equal number of points, a separate spline can be generated for each point position in the cross-sections. For instance, the first point (or leftmost point) of each sampled cross-section is added to a first spline, while the second point of each cross-section is added to a second spline, and so forth. Each cross-section uses the cross-section’s midpoint as a local origin and interpolates the sampled points relative to this local origin. However, rather than relying on traditional Cartesian coordinates, the splines are based on the spherical coordinates of the cross-section points as shown below in Figure 2. Coordinates \((r, \theta, \phi)\) are computed relative to the midpoint \(M\) of a cross-section. The motivation for computing cross-sections in such a way is to preserve the channel geometry around meanders in the waterway path.

#### 3.1.2 Waterway Spline Evaluation

Once splines are constructed from these cross-section spherical coordinates, evaluation can begin. Each spline is evaluated, as shown in Algorithm 2, at even intervals according

![Algorithm 1: Waterway Generation Algorithm.](image)
Figure 2: Cartesian to spherical coordinates.

Figure 3: Basic mesh generation. Dashed green lines represents the part of the mesh already created, while the solid one represents the current mesh polygon.

This algorithm is straightforward so long as there are no degenerate cases. Sometimes, however, cross-sections surveys are taken well past the banks of the waterway and, as shown in Figure 4, intersect one another whenever the profile takes a sharp turn for a dense interpolation. Unfortunately, such cases can occur and must be accounted for.

3.1.3 Waterway Correction

A corrective step is taken at this point in the waterway generation process to repair potentially broken topology. The first approach to solve this problem uses the dot product to determine if the vector formed by the given point in \( CrossSection^i \) and its corresponding point from \( CrossSection^i \) faces the same direction as the vector formed by the associated profile points for those two cross-sections. If \( \overrightarrow{P_i P_j} \cdot \overrightarrow{Q_i Q_{i+1}} < 0 \), \( P_j \) is merged with \( P_i \), producing triangles (or even lines in more worse cases) to replace the degenerate polygons. The result is shown in Figure 5.

```plaintext
Algorithm 2: Waterway Evaluation Algorithm.

1. for i = 1 to NumberOfProfileSamples - 1 do
2.     for j = 1 to NumberOfCrossSectionSamples - 1 do
3.         CrossPolygon(i-1, j, i, j + 1, i + 1, j);  # generate process to repair potentially broken topology.
4.     end
5. end
```

Figure 4: Two interpolated cross-sections intersecting each other.

The filtering process corrects intersections between \( CrossSection^i \) and \( CrossSection^i + 1 \). To achieve a better result, we extended this to \( CrossSection^i + 1 \). To filter the cross-section \( i \), we look for intersections with the interpolated cross-sections \( i - 1 \) and \( i + 1 \). Algorithm 3 shows the steps of this filtering process with \( Q_i \) representing the \( i \)th profile point.

Using the non-null constant \( \varepsilon \) circumvents arithmetic precision issues and to merge close points together. The constant \( \lambda \) is necessary to check that the \( CrossSection^i + 1 \) is better than the \( CrossSection^i \). In the implementation, \( \lambda = 0.5\|Q_{i-1}Q_{i+1}\| \). Although the previous algorithm works quite well in most cases, very sharp turns in a waterway can cre-

```plaintext
Algorithm 3: Correction Algorithm.

begin
1. for i = 1 to NumberOfProfileSamples - 2 do
2.     for j = 1 to NumberOfCrossSectionSamples - 1 do
3.         if \( \overrightarrow{P_i P_j} \cdot \overrightarrow{Q_{i-1}Q_{i+1}} < \varepsilon \) then
4.             \( P_i \leftarrow P_j \);
5.         end
6.         if \( \overrightarrow{P_i P_j} \cdot \overrightarrow{Q_{i-1}Q_{i+1}} < \varepsilon \) and \( \overrightarrow{P_{i-1}P_{i+1}} \cdot \overrightarrow{Q_{i-1}Q_{i+1}} > \lambda \) then
7.             \( P_j \leftarrow P_{i+1} \);
8.         end
9.     end
10. end
11. end
12. CrossLastPolygon();
```

Figure 5: Simple Intersection Filtering.
ate so many interpolated cross-sections intersections that it fails to correct them (see Figure 6).

Figure 6: Two examples of multiple intersecting cross-sections.

To head off such disasters, interpolated cross-sections must be prevented from intersecting too many subsequent cross-sections. Therefore, we use a sorted brute force intersections algorithm to compute all the intersections between all the interpolated cross-sections. Then, if an interpolated cross-section intersects more than three subsequent cross-sections, half of the cross-section’s points are discarded where the crossings occur. An example of this filtering is shown in Figure 7.

Figure 7: Multiple Intersection Filtering.

3.2 Merging LIDAR with Survey Data

Once a mesh is generated, it can then be merged with the LIDAR data in the same area. Currently, the desired effect is to use the boundary from the waterway mesh to clip points that would fall within the waterway channel and concatenate the remaining LIDAR with the points from the waterway mesh, using a constrained Delaunay triangulation. The strength of such an approach derives from the assumption that survey data is inherently more accurate than LIDAR data. However even if this is true, the accuracy of this approach is limited by the density of the original cross-section survey measurements. In some cases, part of a cross-section survey is taken on land, meaning that using LIDAR inside the generated waterway might not be such a bad idea.

4. RESULTS

The waterway generation algorithm produces a realistic terrain surface, even when merged with LIDAR data. As shown in Figure 8, the waterway topology of Bayou Black is preserved and combined with triangulated banks up to a hundred feet out from the bayou’s edge.

Figure 8: Wireframe mesh of Bayou Black.

Figure 9 compares the original LIDAR with the merged LIDAR. The left image shows the before picture: LIDAR data from an area that contains a canal is simply triangulated and rendered without any supplemental data, producing a terrain surface that is mostly flat in the area where the canal should run. By contrast, the image on the right shows the canal generated by the interpolation process described earlier merged with the LIDAR; the merge produces depth in the channel and even defines the natural levees along its banks.

Figure 9: Triangulated LIDAR for the area around Baker Canal South. The image on the right includes the canal.

5. CONCLUSION AND FUTURE WORK

In conclusion, the terrain modeling system provides reasonable results for merging waterway surveys and LIDAR data. The system produces highly defined waterway geometry based on real data, merged with topographic elevations to form a complete DEM readily available for hydraulic models and geospatial analysis. However, waterways rarely exist in isolation of other channels. Waterways will typically start and or end in another waterway or even a waterbody such
as a lake or ocean. In such cases modeling multiple waterways with sparse survey data would result in discrepancies at junction points. One waterway could have an open channel at the intersection area, while another waterway could have a bank at the intersection area. Future work should entail a method to keep the channels open in these areas of confluence.

6. REFERENCES


