Abstract — We present a tunable, morphologically aware, lossy, geospatial compression algorithm based on right-triangulated irregular networks (RTIN). After constructing a RTIN network of the initial gridded data, a set of criteria is used to selectively remove vertices that minimally impact the overall shape of the mesh. This approach allows for the reduction in size of geospatial data in a manner that minimizes the actual loss of fidelity to the base data, while maximizing the reduction in data size. The resulting thinned mesh thus possesses a high degree of resolution in areas where the original data possesses high variability and each point carries usable information regarding the underlying structure of undersea or land terrain and lower degrees of resolution in areas that are flatter and each point thus contributes less value. As the applied criteria are tunable to user specifications, this algorithm is widely applicable from deep ocean to coastal regions.

Index of Terms – Right Triangular Irregular Networks, Bathymetry, Computational Geometry, Databasing

I. INTRODUCTION

As the resolution and accuracy of modern sensor technologies has improved, the difficulty in handling the very large geospatial datasets they produce has grown. From storage to processing to dissemination, the issues presented by massively dense spatial datasets offer numerous challenges in a variety of fields including, but not limited to, bathymetry, topography, and meteorology.

Given that in any modelled geospatial surface there will be some regions of a generally flat nature, it should be clear that not every portion of a dataset contributes significantly to the surface representation of the set. Thus is the question raised: is there a manner in which only those portions of the dataset which contribute meaningful information to the dataset as a whole be found, selected, and used independently of those portions with little overall bearing to the surface?

The algorithm presented here, Gridded Irregular Network (GRIN™), is an attempt to achieve these aims. Using this algorithm, a gridded dataset is subdivided into triangles. Points that can be removed without destabilizing the triangular mesh are detected and evaluated for their overall contribution to the original dataset’s fidelity. Non-significantly contributive points are removed and their containing triangles merged together in a pattern designed to maximize the thinnability of the remaining points. The final result is a triangular mesh approximation of the original dataset, with larger triangles (and fewer component points) for flatter areas with more complicated regions possessing smaller triangles and greater resolution, up to the original resolution of the data.

II. BACKGROUND

1. Top-Down Approach

Introduced by Evans et al. [1], right triangulated irregular networks (RTINs) are the fundamental structure utilized by this algorithm. A modification of the established triangulated irregular network (TIN) structure, the RTIN structure was introduced in an effort to provide many of the surface approximation qualities of a TIN while avoiding TIN’s corresponding increase in data density and complexity. As part of the RTIN’s introduction, several other core components of this algorithm were also introduced, including the usage of thresholding criteria to achieve different thinning results and, conceptually, the bottom-up approach used by this algorithm.

Originally intended to approximate a surface rapidly for visualization, the process described within this paper uses what is termed the “Top-Down Approach”. Starting with just the four corner points of
the input grid, the Top-Down Approach adds further triangles by splitting the larger triangles at the midpoint of their hypotenuse. The RTIN is thus refined until a certain preset limit is reached. As originally described, instead of using this approach to create a single thinned RTIN mesh, a hierarchy of triangles were created so that different thinned meshes could be generated in real-time to provide the experience of flying or walking through the virtual surface. Of particular note, in the majority of runs new points, will be introduced into the resulting RTIN as the midpoints of the various refined hypotenuses will not match up with grid points from the original input dataset.

Suárez and Plaza [2] present a basic coarsening algorithm for use in RTINs, although still within Evans’ top-down approach. The coarsening approach used here starts with the mesh of the finest triangles created via Evans’ subdividing top-down approach. The resultant hierarchical structure is then used to find triangles that can be merged together. Suárez and Plaza also provides demonstrative proof of the effectiveness of thinned RTINs in approximating the surface of the original full mesh with only a fraction of the points and a small amount of induced error.

Guibas and Stolfi [3] provide the quadedge data structure which is the in memory representation of the RTIN utilized by this algorithm. The quad-edge structure is based on the insight that each edge lies exactly between two faces upon the mesh and connects two vertices. Each quad-edge structure then contains a pointer to the two vertices it connects and the four connecting edges around the adjacent mesh faces. Thus, by assuming the edge to be directional in nature, one can easily traverse around either face via a quick pointer lookup. This allows for the quick traversal of the mesh in an arbitrary direction and, indeed, allows for the lookup of an arbitrary point within the mesh in $O(n)$ time. More importantly this structure allows for an alternative to the hierarchical binary tree used in both Evans and Suárez, and introduces the possibility of the bottom-up approach adopted here.

2. Bottom-Up Approach

The top-down approach first chosen by Evans [1] initializes the RTIN at its coarsest resolution, possessing only the four edge vertices and consisting of only two bisecting triangles splitting the square grid. Each triangle is then subdivided into smaller triangles until an arbitrary limit would be reached. For a visual approximation this is a perfectly suitable approach, but when dealing with real world bathymetric/topographic data points, the loss of the original data point locations negated the use of the thinned RTIN result. This loss occurs because the Top-Down Approach produces a grid that is $2^{N+1}$ by $2^{N+1}$ in size, where N is an integer greater than or equal to 1. Thus, the original grid has meet this restrictive dimensionality requirement.

Instead of the Top-Down Approach, the GRIN algorithm employs a Bottom-Up Approach. Pajarola presents this concept and a formal algorithm in [4-6]. The Bottom-Up Approach begins with the grid at the original resolution and performing a triangulation upon it. This means every point retained within the thinned RTIN is an actual point present within the original data as well. In addition to retaining all original points, the approach also needs to allow for non $2^{N+1}$ by $2^{N+1}$ sized grids, including rectangular grids. Thus, input datasets have no imposition that the end result is a square bisected by two triangles.

While [4-6] published the Bottom-Up Approach, we have not found implementation of it in the literature for digital elevation models (DEM), either for land or bathymetry. Our implementation of the algorithm in GRIN includes a process to serialize the resulting thinned RTIN into a tightly compressed file containing only a few pieces of metadata and those points not removed in the thinning process. While the average amount of thinning will vary both on the data and criteria being used, we found upwards of seventy percent of points removed were not uncommon. Further, with our serialization process each remaining point only requires 12 bytes of data, resulting in an extremely compact on disk result.

III. GRIN THINNING ALGORITHM

Initially, the input data to GRIN must be a DEM grid. The X and Y node spacings can be differ. The dimensions of the grid can be square or rectangular; however, grids possessing dimensions of $2^{N+1}$ by $2^{N+1}$ will be fully reducible to the coarsest possible state of two triangles. Since this approach only coarsens the original mesh, the initial resolution of the input grid will be the maximum resolution of the resulting thinned mesh. Thus, the initial grid input into GRIN should be as dense as possible within the processing limitation of the computer used.
match the edge creation process from the top-down approach, so that if every checked point passes the criteria evaluation the triangular network will reduce to as close to the two bisecting triangles as the grid’s dimensionality allows.

As with the right-angle point detection process, there are two different edge selection methods depending on whether the point detected is within the mesh or along the borders of the dataset. When along the outer edge of the grid only one edge orientation is valid as the outer grid edge cannot be removed and is thus automatically invalid. Within the mesh interior, however, the edge selection process depends on both the iteration count of the algorithm and the location of the point being evaluated.

On the first and every subsequent \(n\)th even-numbered iteration, the edge selected only depends on the row in which the point being considered is found. Figure 2 illustrates this concept. Algorithm 1 details the edge selection process. Points along the bottom-most non-edge 2\(n\) rows use the vertical edge orientation, with the next 2\(n\) rows using the horizontal edge orientation, and repeating thereon. For odd \(n\)th iterations, the edge selected depends on both the row and column in which the point being considered is found. Starting from the left, each column alternates between 2\(n\) repetitions of the two possible edge orientations (vertical and horizontal), with the bottom-most non-edge 2\(n\) rows beginning with vertical orientations and the next 2\(n\) rows beginning with horizontal orientations, and repeating thereon.

![Figure 1: A properly tessellated 5x5 right triangulated network](image)

1. Point Selection

Starting at the bottom leftmost point, the initial input grid is triangulated using a tessellated pattern that subdivides the smallest square cells described by the grid. This pattern alternates the direction of the bisecting edge as show in Fig. 1. This alternation ensures that the resulting right-triangulated network can be reduced, all the way to four edge points and two bisecting triangles if the grid is of a proper \(2^{N+1}\) by \(2^{N+1}\) dimensionality.

The actual thinning of the triangular network occurs in an iterative manner, starting with a detection of points bordering only right angles. For each of these “right-angle points” detected, an orientation for evaluation and potential deletion are found based on the iteration of the algorithm and the point’s location within the mesh, so as to maximize the thinnability of the mesh. Each point is then evaluated for potential removal and, if found to be a valid removal point, removed along with one or two of its attached edges. This edge removal results in the merging of the former point’s containing triangles, from four or two to two or one, respectively. Any right-angle points evaluated as invalid for removal are marked as invalid for removal from all future iterations. This process iterates until no unmarked right-angle points are found.

2. Edge Selection

After a point is identified as surrounded by right-angles and therefore viable for removal, an edge must be selected for point evaluation as well as edge removal if the point is ultimately found to be removable. The edge selected must also attempt to maximize the potential thinnability of the triangular mesh, to avoid reaching a premature end of iteration. The edge selection process in GRIN is designed to

<table>
<thead>
<tr>
<th>Iteration 0</th>
<th>Iteration 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-hhhhhh-</td>
<td>-hhhhhh-</td>
</tr>
<tr>
<td>vvvvvvv</td>
<td>vhhvhvvhv</td>
</tr>
<tr>
<td>vhvvhhhhv</td>
<td>vvhhvvhv</td>
</tr>
<tr>
<td>vvrvvvvv</td>
<td>vvhhvvhv</td>
</tr>
<tr>
<td>vhvvhhhv</td>
<td>vhvvvhhv</td>
</tr>
<tr>
<td>vhvvhvvv</td>
<td>vhvhhvvh</td>
</tr>
<tr>
<td>vhvvvvhv</td>
<td>vhvvhhvh</td>
</tr>
<tr>
<td>vvvhvvhv</td>
<td>vvvhvvhv</td>
</tr>
<tr>
<td>-hhhhhhh-</td>
<td>-hhhhhh-</td>
</tr>
</tbody>
</table>

![Figure 2: Vertical, “v” and horizontal “h” edge orientations in the first two iterations of a 9x9 grid during the GRIN thinning process. Each “h” or “v” entry in a row signifies the orientation of each edge between two initial grid points during the initial stage of the criteria evaluation process detailed in Section III.2. The entry “-” means an exterior corner point.](image)

3. Removal Criteria

For the datasets and use cases that factored into our implementation two criteria were used. The first is a threshold criterion that evaluates a point by comparing the point’s value with the interpolation of the value at the point’s location using the endpoints of the edge selected in
the previous step. If the difference between the actual value of the point and the interpolated value is within a specified threshold, the point is removed. This criterion is simple to understand in function; set a threshold of 10 and if the difference between the interpolation and the actual value is less than 10, the point is removed.

The second criteria is a percent criteria that evaluates a point by comparing the difference between the point’s value and the interpolation of the value at that point with a specified percentage of the point’s original value. Functionally, this criterion is best understood as a protective criterion for low valued points and thus was only used in combination with the threshold criteria during testing and development. Used solo, the criteria would remove more high valued points while maintaining low valued ones.

4. Serialization and Reconstruction

In order to properly serialize the mesh each of the triangular faces must be maintained and thus, classically, serializations of a triangular network resulted in either larger file sizes or the majority of the points reduction only offsetting the less compact format required to record the triangle faces. For this algorithm in particular, however, it was realized that the order in which points were evaluated was procedural in nature. If the points removed in the first iteration of the thinning process match the points removed from the first iteration of a previous run, then not only will the next iteration also evaluate the same points as the next iteration of the previous run, but the eventual surface of the thinned result will be identical to the original.

As such, when serializing the thinned result of this algorithm, only those points that were not removed by the thinning process need to be recorded, along with metadata describing the extents of the grid as well as the spacing between points. When reconstructing the mesh from the serialized format, a grid of the same extents as the original is produced with all points except for those present in the serialized format marked as invalid points. Then, a custom criterion is applied so that any point marked as invalid is removed. Since all points marked as invalid are points that were removed in the initial run, the same mesh surface as produced originally is recreated. The metadata used to recreate the grid can largely be ignored and thus the reduction in the original dataset’s points correlates directly to a reduction in the size of the serialized triangular mesh.

IV. CASE STUDIES

We tested the GRIN algorithm using two sets of combined bathymetry and topography data from the NOAA Coast Relief Model [7] for the United States. These test cases were run using a Macintosh desktop with 12GB of memory and two 2.4GHz quad-core Intel Xeon processors. The algorithm was coded in Java. Each case took ~ 90 minutes to run.

1. Initial Study - Mississippi Coastal Region

We first applied this algorithm to test data of bathymetry and topography of the Gulf of Mexico centered around the Mississippi coast as shown in Figure 4. The original grid spacing before reduction was 250m. This grid was 887 by 1332 in size, containing 1,181,484 points. The combined reduction criteria used is change in height < 30 meters and change in percentage of water depth < 1%. The algorithm reduced the grid by 66.3% of the original size to ~ 400,000 points in approximately 2 minutes. Figure 4 shows the reduced result. For visualization of the result, we fit the data to a Delaunay mesh from Matlab.

<table>
<thead>
<tr>
<th>Algorithm 1: Bottom-Up RTIN Edge Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>This algorithm is for interior edge selection for thinning using Bottom-Up Approach.</td>
</tr>
</tbody>
</table>

/* Input: right-angle point p possessing an x and y component, iteration counter i = 0, ..., n, ..., N-1 */
/* Output: Edge orientation to use for criteria evaluation/triangle merging */

1: \( \alpha = p.y \% 2^\left\lfloor \frac{i}{2} \right\rfloor + 1 \)
2: \( \beta = p.x \% 2^\left\lfloor \frac{i}{2} \right\rfloor + 1 \)
3: If \( i \) is even
  4: If \( \alpha \leq 2^\left\lfloor \frac{i}{2} \right\rfloor \) \&\& \( i \!\neq\! 0 \)
    5: Then orientation is vertical
    6: Else orientation is horizontal

7: Else
8: If \( \alpha \leq 2^\left\lfloor \frac{i}{2} \right\rfloor \) \&\& \( i \!\neq\! 0 \)
9: If \( \beta \leq 2^\left\lfloor \frac{i}{2} \right\rfloor \) \&\& \( i \!\neq\! 0 \)
10: Then orientation is vertical
11: Else orientation is horizontal
12: Else
13: If \( \beta \leq 2^\left\lfloor \frac{i}{2} \right\rfloor \) \&\& \( i \!\neq\! 0 \)
14: Then orientation is horizontal
15: Else orientation is vertical

Figure 3: When interpolating point B', different criteria values create different ranges of acceptance.
2. Multi-Run Study - Galveston, Texas Area

We then applied the algorithm to denser data set of bathymetry and topography of the Gulf of Mexico centered near Galveston, Texas as shown in Figure 5. The original grid spacing before reduction was 250m. The dataset contained a total 16,785,409 points, arranged in 4097 rows and 4097 columns. Note that this grid is $2^N + 1$ by $2^N + 1$, where $N = 12$. We applied 46 sets of reduction criteria; Table 2 provides results three examples. In these cases, the algorithm took $\sim 90$ minutes. After reduction, we then reconstructed the bathymetry surface (c.f. Section III.4) back to the original grid dimensions to obtain an RMS error.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Maximum Difference (m)</th>
<th>RMS Error (m)</th>
<th>Total Point Deletions</th>
<th>Thinning Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01% and 1m</td>
<td>1.4234m</td>
<td>0.0247</td>
<td>11,698,335</td>
<td>69.69%</td>
</tr>
<tr>
<td>0.3% and 3m</td>
<td>4.2976m</td>
<td>0.2730</td>
<td>15,090,330</td>
<td>89.90%</td>
</tr>
<tr>
<td>10% and 10m</td>
<td>16.1533m</td>
<td>2.6266</td>
<td>16,371,142</td>
<td>97.53%</td>
</tr>
</tbody>
</table>

Table 1: Thinning results from three different sets of combined reduction criteria for the Galveston, Texas multi-run study.

Figure 4 – A thinned mesh derived from a grid originally showing the Mississippi coast demonstrates how deeper, flatter surfaces are thinned while closer shallows are retained.

IV. DISCUSSION

The extra time required to run the algorithm when going from the first to second case was a factor of 45, although the number of points increased by a factor of 14. This extra does not include an analysis of the file reading and writing. According to [5-6], the computation time is $O(N)$. Note that no effort has yet been expended to parallelize the algorithm, although several steps in the process would seem to lend themselves very well to a parallel approach.

As shown in the second column of Table 2, some points are not removed even when the differences between neighbors exceeded the initial set threshold. A cumulative distribution analysis shows that for this case, roughly 5% to 7% of the remaining points exceeded the height threshold. In testing this expansion past the threshold was generally found to be no greater than fifty percent and rare enough to not have a noticeable effect on the results. We hypothesize two possible reasons: 1) the greater height difference lay along the opposite edge orientation chosen for evaluation and 2) a point that should have been retained may become eligible for removal because in an earlier iteration a point that would have prevented it from being retained was itself removed.

For non-$2N+1$ by $2^N+1$ input grids, testing indicates that the reduction in maximal thinning begins small and grows the further a dataset departs from the ideal grid.
dimensionality. Pragmatically, we anticipate that this is rarely a concern outside of truly oddly dimensioned grids. Preliminarily, there does seem to be an underlying function of the resulting RTIN’s geometry that from the input dimensions. For grids of dimension $2^n+1 \times 2^m+1$ ($M$ is an integer > 0) the number of points remaining after maximally thinned is

$$f(n, m) = 4 + \sum_{i=1}^{m-n} 2^{i-1}, \text{ where } n \leq m \quad (1)$$

Table 2 lists three possible dimensional arrangements (in addition to the previously established $2^n+1 \times 2^n+1$).

<table>
<thead>
<tr>
<th>Grid Dimensions</th>
<th>Maximally Thinned Points Remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^n+1 \times 2^n+1$</td>
<td>4</td>
</tr>
<tr>
<td>$2^n+1 \times 2^{n+1}+1$</td>
<td>5</td>
</tr>
<tr>
<td>$2^n+1 \times 2^n+2^n+1$</td>
<td>6</td>
</tr>
<tr>
<td>$2^n+1 \times 2^{n+1}+2^{n+1}$</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 2: Dimension Expressions and Maximal Thinnability

Due to the underlying geometries of the triangular mesh, when triangles are merged post point-removal either two triangles are merged into one or four are merged into two. Two configurations of edges and points allow for this merging, both characterized by the angles between the connected edges of the point being examined all being right angles. One case consists of points laying along the edge of the grid and connected to three edges forming two right angles. The other case is for points within the interior of the mesh connected to four edges forming four right angles.

Currently these points are found via a simple exhaustive search of all unmarked points, but ongoing effort is directed at pre-computing which points will fall into one of these two configurations. These points should be, at a minimum, calculable until a nearby point is marked as invalid for removal.

V. SUMMARY

The algorithm detailed here is the first application of the RTIN bottom-up approach applied to bathymetric/topographic with the purpose of DEM thinning. This process allows for the creation of non $2^n+1$
by $2^n+1$ dimensioned, while datasets not possessing the $2^n+1$ by $2^n+1$ dimensionality have proved to be irreducible to four edge points and two bisecting triangles, they often are reducible enough that any reduction in potential thinning is small. Further, there seems to be an initial functional relationship between the input grid’s dimensions and the resulting RTIN’s maximum thinnability.

The use of the dual thinning criteria for bathymetric data ensures this process is applicable both in the deep ocean and along shallows and shoreline. Of course, the criteria used for the final point evaluation is irrelevant to the overall functioning of the algorithm, and should generally use values chosen based on the need of the end user and the data being thinned. Possible other criteria include, but are not limited to: criteria that evaluates the point’s position within the graph, either via a generalized bounding box or using some manner of keep/discard list; criteria that tracks how many points are removed and halts thinning after reaching a preset limit; and criteria that removes points with uncertainty above a certain threshold. Importantly, it is possible to chain criteria, requiring a point to satisfy all criteria used in order to qualify for removal.

Our serialization process enables us to maintain the arrangement of triangle faces within the mesh while only actually recording the individual points, enabling us to directly translate the point reduction of the thinning process into the resulting serialized mesh’s file size. By thus enabling grids of any dimensionality to be thinned of any points found to not be meaningfully contributing to the overall mesh surface, this algorithm provides an intelligently customizable and tunable compression algorithm applicable to all bathymetric grids.

REFERENCES


