ECRONICON OPEN ACCESS

EC OCEANOGRAPHY
Review Article

Manipulating Sonar Bathymetry into Virtual Reality Using Adaptive Terrain and Graph-Theoretic Meshing Algorithms to Acquisition World War II U.S. Navy Liberty Vessels in the Gulf of Mexico Utilizing the Hadoop Ecosystem

Wilbert A McClay[1-15]* and Elaine Stewart McClay[16-18]

[1]Department of Linguistics, Tulane University, New Orleans, Louisiana, USA
[2]Labstream Corporation, St. Charles, Missouri and Los Angeles, California, USA
[3]U.S. Navy, New Orleans, Louisiana, USA
[4]Department of Criminal Justice and Forensics, University of New Haven, West Haven, CT, USA
[5]Tulane University School of Engineering, New Orleans, Louisiana, USA
[6]Department of Information Systems, Northeastern University, Boston, Massachusetts, USA
[7]U.S. Department of Defense, Washington, DC, USA
[8]U.S. Department of Energy, Washington, DC, USA
[9]Danish Defense Forces, Copenhagen, Denmark
[10]Wolfsmilch Drones, O'Fallon, Missouri, USA
[11]Lawrence Livermore National Laboratory, Livermore, California, USA
[12]Brandeis University, Waltham, Massachusetts, USA
[13]Harvard University, Cambridge, Massachusetts, USA
[14]JustUs Youth Development Automated Behavioral Analysis Corporation, St. Charles, Missouri, USA
[15]Southern University and A&M College, Baton Rouge, Louisiana
[16]Fisk University, Nashville, Tennessee, USA
[17]Whittier University, Whittier, California, USA
[18]Smith College School of Social Work, Northampton, Massachusetts, USA

***Corresponding Author:** Wilbert A McClay, Department of Linguistics, Tulane University, New Orleans, LA, USA.
Email ID: wmcclay@tulane.edu.

Received: April 15, 2025; **Published:** January 12, 2026

Abstract

The incentive for this article is to design a virtual environment that fosters the use of uniform and nonuniform gridding tactics to handle large-scale bathymetry data. The understanding of depth and sonar imagery has been a growing interest of many oceanographers and scientists, hence proper handling of massive sonar data could improve coverage of the sea terrain. The bathymetry data will be acquired from the World War II U.S. Navy Liberty Vessels sunken in the Gulf of Mexico. The implementation of these non-uniform gridding tactics should prove worthwhile, allowing finer resolution and approximation of the data set while utilizing a lower dimension of the data set. Heuristic terrain simplification algorithms are based upon the fundamentals of divide and conquer algorithms in greedy programming. The simplification method proposed in this work is a multi-pass decimation method, which begins with a Delaunay triangulation of the input data of 167,358 triangles and then a reduction to 83,679 triangles to remove outliers. Hence, the results of the heuristic terrain simplification algorithm produced a significant 50% percent compression ratio with optimal connectivity of the bathymetric data and still produced an accurate approximation of the bathymetric data set. The estimation of the terrain simplification algorithm granted exemplary results with respect to triangular compression, where in the original image was composed of 167,358 triangles was easily parsed into the Hadoop Ecosystem with Apache Spark Directed Acyclic Graph Engine (DAG) with Apache Scala utilization to formulate Resilient Distributed Data Sets for Amazon AWS Apache Cassandra (NoSQL Databases) to demonstrate quintessential optimization and scalability for sonar bathymetric data compression.

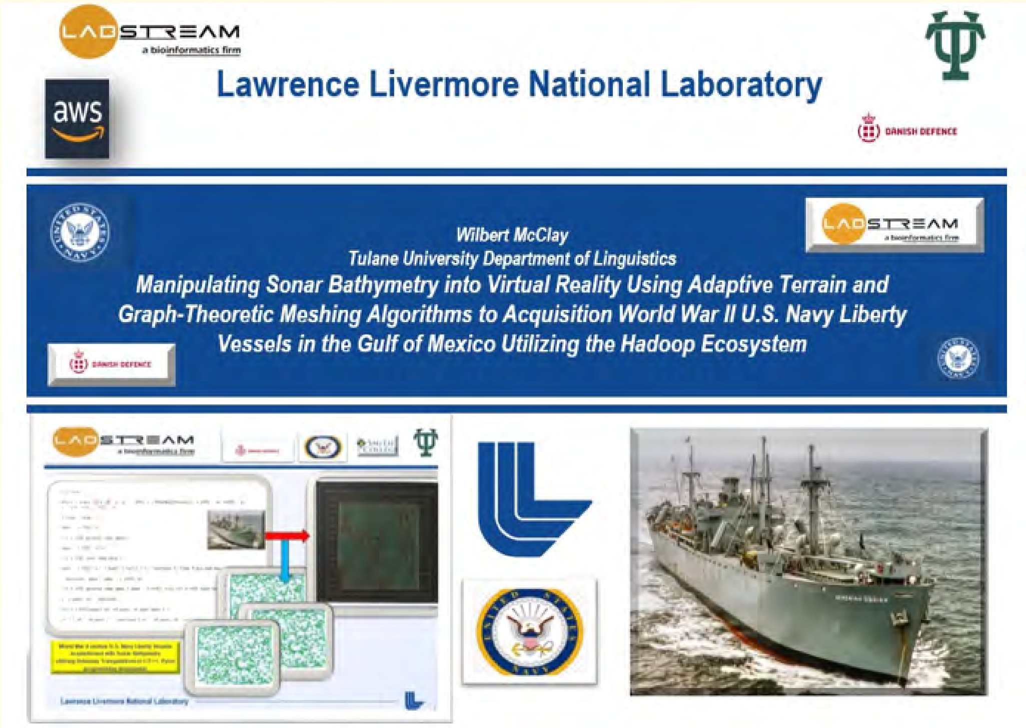
Keywords: *Bathymetry; Delaunay Triangulation; Minimal Spanning Trees; Uniform Gridding; Hadoop Ecosystem; Apache Spark Directed Acyclic Graph(s) DAG; Apache Scala; Amazon AWS; Apache Cassandra*

Introduction

Manipulating massive data sets for computational science has proved to be a rigorous process due to gridding tactics that are based upon using partial differential equation analysis. The incentive for this project is to design a virtual environment that fosters the use of uniform and non-uniform gridding tactics to handle large-scale bathymetry data set analysis. The implementation of these non-uniform gridding tactics should prove worthwhile allowing finer resolution and approximation of the data set while utilizing a minimal sample of the bathymetric data set. Heuristic terrain simplification algorithms are based upon divide and conquer algorithms in greedy programming. The goal in acquisitioning massive sonar bathymetry data sets is to acquire a maximum amount of bathymetric data, but only utilize a fraction of the bathymetric data to form a legitimate approximation.

The importance of the relationship between computer engineering and oceanography is a legitimate and growing concern. This has attracted the interest of many government agencies, such as the Department of Defense, and contractors in private industry. The knowledge of acquiring and analyzing bathymetric data has been a growing and diligent concern of many oceanographers and scientists.

The first question to understand is exactly what is bathymetry? For our purpose, the analysis of bathymetry will concern multibeam bathymetry, where multibeam bathymetry is collected and processed to create shaded relief images that depict seafloor morphology. The understanding of depth and sonar imagery can be extracted from multibeam bathymetry gathered by multibeam sonar. Thus, the question can be modified to what is multibeam sonar? Briefly, multibeam sonar is a system that illustrates fan-shaped coverage of the sea floor with the similarity of a side scans sonar, yet the output data is in the form of depths rather than images. The importance of analyzing bathymetry is that it corresponds to the Liberty Vessels. The Liberty Vessels were first launched on Sept. 17, 1941 [21], at Baltimore Naval 4 shipyard and were named after Patrick Henry. During World War II the United States mass-produced a standardized 10,000-ton ship with a cruising speed of 16 knots called Liberty Vessels, illustrated in figure(s) 1-1(a), below.



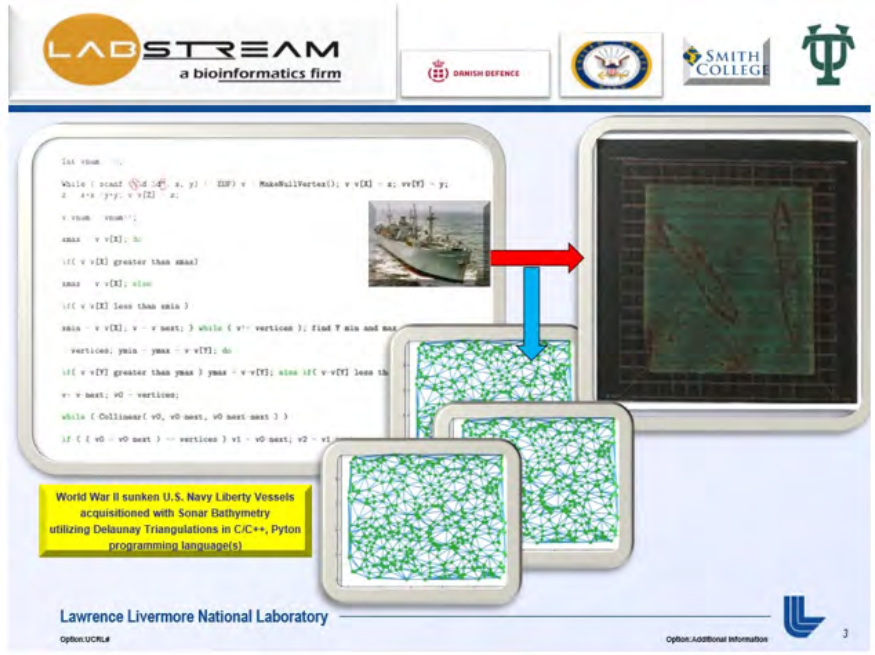
Lawrence Livermore National Laboratory

Wilbert McClay

Tulane University Department of Linguistics

Manipulating Sonar Bathymetry into Virtual Reality Using Adaptive Terrain and Graph-Theoretic Meshing Algorithms to Acquisition World War II U.S. Navy Liberty Vessels in the Gulf of Mexico Utilizing the Hadoop Ecosystem

Figure 1



```
int vnum ...;
While ( scanf ("%d %d", x, y) != EOF) v = MakeNullVertex(); v v[X] = x; vv[Y] = y;
z = x*x +y*y; v v[Z] = z;
v vnum = vnum++;
xmax = v v[X]; do
if( v v[X] greater than xmax)
xmax = v v[X]; else
if( v v[X] less than xmin )
xmin = v v[X]; v = v next; } while ( v != vertices ); find Y min and max
 vertices; ymin = ymax = v v[Y]; do
if( v v[Y] greater than ymax ) ymax = v v[Y]; else if( v v[Y] less th
v= v next; v0 = vertices;
while ( Collinear( v0, v0 next, v0 next next ) )
if ( ( v0 = v0 next ) == vertices ) v1 = v0 next; v2 = v1
```

World War II sunken U.S. Navy Liberty Vessels acquisitioned with Sonar Bathymetry utilizing Delaunay Triangulations in C/C++, Pyton programming language(s)

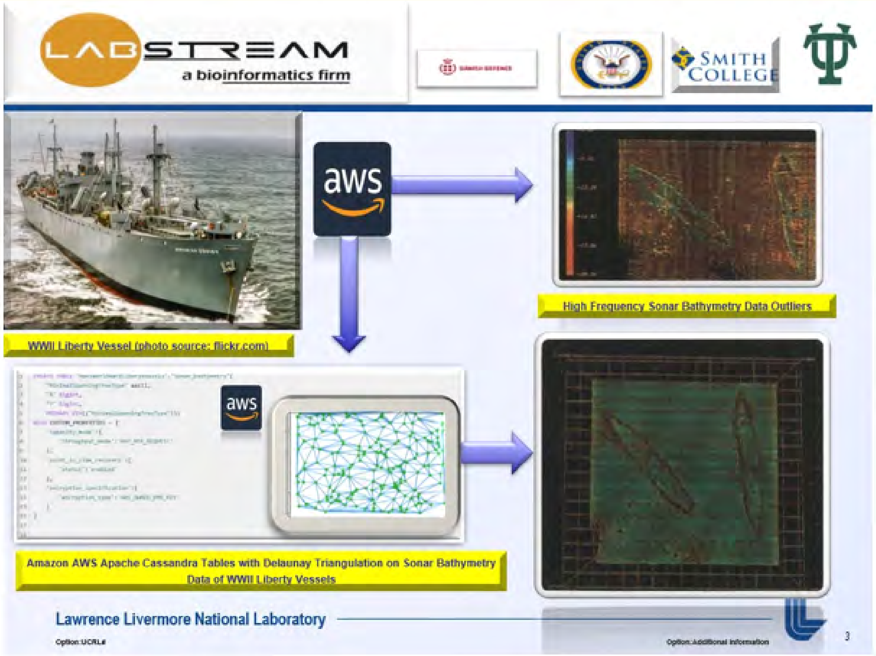
Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional Information

3

Figure 1a: *Big data virtual environment with uniform and non-uniform gridding for sonar bathymetry of World War II liberty vessels sunken in the Gulf of Mexico.*



WWII Liberty Vessel (photo source: flickr.com)

High Frequency Sonar Bathymetry Data Outliers

```
CREATE TABLE [illegible]
    [illegible]
    "X" [illegible],
    "Y" [illegible],
    [illegible]
WITH CUSTOM_PROPERTIES = {
    'capacity_mode':{
        'throughput_mode':'[illegible]'
    },
    'point_in_time_recovery':{
        'status':'enabled'
    },
    'encryption_specification':{
        'encryption_type':'[illegible]'
    }
}
```

Amazon AWS Apache Cassandra Tables with Delaunay Triangulation on Sonar Bathymetry Data of WWII Liberty Vessels

Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional Information

3

Figure 1b: *Big data virtual environment with uniform and non-uniform gridding using Amazon AWS Apache Cassandra NoSQL database for sonar bathymetry of World War II liberty vessels sunken in the Gulf of Mexico.*

Geographical data information

In the subsequent section "Understanding Sonar", a more detailed and elaborate approach will give a full understanding of multibeam sonar and its usage to acquire bathymetric data.

The multibeam sonar data used in this project was collected on a survey vessel in the South Horn of the Gulf of Mexico at a latitude of 30.16' and longitude of -88.75'. The South Horn in the Gulf of Mexico is located at 30 degrees North latitude and 88.75 West longitude. As illustrated in figure 2, the average depth of the water was 20 meters in the South Horn of the Gulf of Mexico, and it also illustrates the problems of bathymetric data resolution and the aspects of holidays and artifacts, which will be discussed in later sections. Also, a comparison of figure 2 to 3 illustrates the significance of the way the bathymetry was acquired and the significance of outliers in the bathymetric data set illustrated in figure 3. The bathymetric data points shown in red are some of the significant outliers that will be eliminated using the techniques of uniform and non-uniform gridding and terrain simplification. As you will notice from figure 2 and 3, the data set is a massive, dense, scattered data field, and therefore required the use of uniform and non-uniform gridding to develop a legitimate terrain mapping.

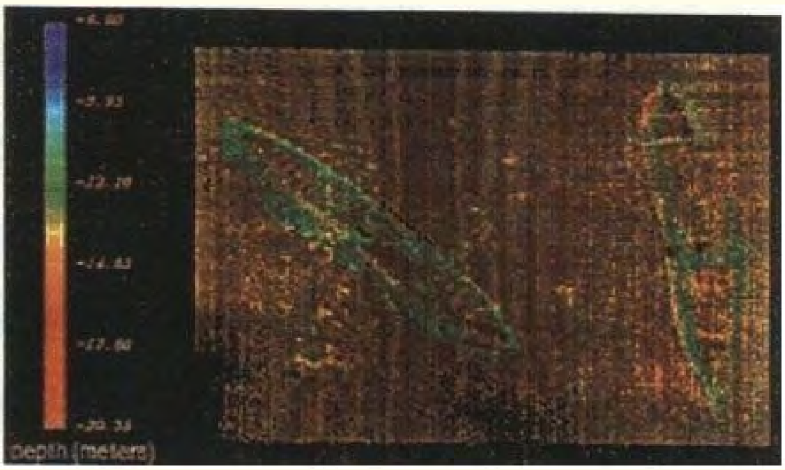


Figure 2: *Materials and methods.*

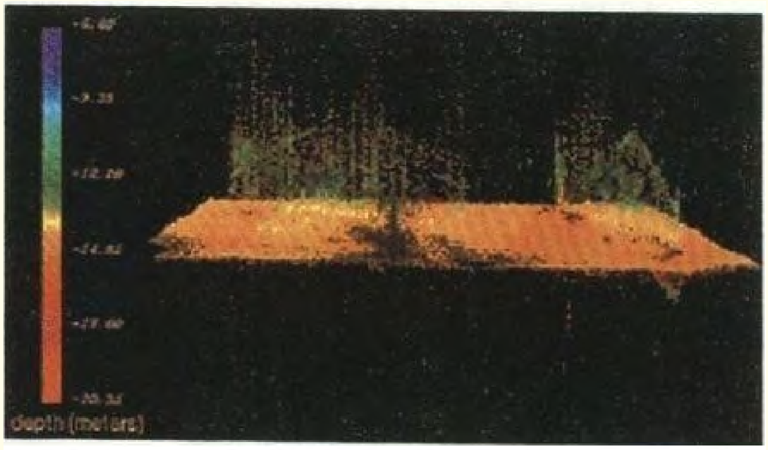


Figure 3: *Results (High-frequency data and outliers).*

Citation: Wilbert A McClay and Elaine Stewart McClay. "Manipulating Sonar Bathymetry into Virtual Reality Using Adaptive Terrain and Graph-Theoretic Meshing Algorithms to Acquisition World War II U.S. Navy Liberty Vessels in the Gulf of Mexico Utilizing the Hadoop Ecosystem". *EC Oceanography* 1.1 (2025): 01-27.

To understand the concept of terrain mapping, a height field must be defined to give general meaning to a triangular irregular network. First, a height field is defined as a set of samples over a planar domain. The most common type of height field is terrain data. For this article, our concern is with bathymetric data as a particular height field. Many applications such as computer graphics for entertainment, and virtual underwater environments use height fields. In practice, it is desirable to render height field terrain data rapidly, but also with high resolution. The bathymetric height field typically uses polygons or, more explicitly, the triangle as an object description since this is the benchmark standard for all graphics hardware. This is a primary concern because raw data can be easily transformed into polygons by placing edges between neighboring samples. The significance in size for terrain description is most important because the larger the bathymetric terrain data, the more expensive it is to render a significant model. This is important, because if an application has computer animation where the display is updated many times per second, then a slow rendering process is useless.

Background

The purpose of this article is to generate a simplified model of a height field from the original model and render it in a virtual application. The model should use as few triangles as possible, and the process should be as rapid as possible. The simplification algorithms chosen include well-known tools such as a randomized quick sort algorithm and the Delaunay triangulation algorithm. However, the use of a terrain simplification algorithm is necessary, primarily because of the redundancy of overlapping triangulation. The simplification method proposed in this article is a multipass decimation method, which would begin with a triangulation of all the input points and then iteratively delete vertices from the triangulation. This gradually simplifies the approximation. As shown from the figure 2 diagram of the bathymetric data set, there were over 3,600,00 points accumulated from the sonar readings, of that data portion 650,000 were used for the primary analysis. The bathymetric data set was invoked and edited in raw binary form. The data had to be corrected for Endianness and converted to IEEE floating-point numbers. Once this process is finished the data in x and y and formatted into x, y, and z triplets from a large-dense scattered data plot. The bathymetry is noted as being a large-dense scattered data set because in certain portions of the bathymetry it is very dense and in other parts the bathymetry is very sparse or scattered. This approach leads to the gridding analysis scheme. It involves the data to be specifically gridded in accordance to maximize a specific number of data points to each cell size within the grid.

The process in cropping of the raw execution of the bathymetric data composed of 3,600,000 points had to be significantly sliced down to an image space of 650,000 points, which was defined by this formula:

$$x_1 = (\text{pntx} - x_{min})/x_{max} - x_{min} * (\text{NUMCOLS} - 1 + 0.5)$$

$$y_1 = (\text{pnty} - y_{min})/y_{max} - y_{min} * (\text{NUMCOLS} - 1 + 0.5)$$

Respectively, x_1 and y_1 correspond to the bounding box region or extents to transfer the cropping of the data set from image space to computational point space, by normalizing the values of the maximum and minimum values of the x and y arrays. The cropping at the extents of (-88.751383, 30.164613) and the extents of (-88.749436, 30.16594), was done to find the correct position in computational space to that within image space, and thus explains the extended floating-point precision of the computational values. In addition, it was done as a more computationally conventional diagram to find the better locality of the two Liberty Vessels.

The first terrain simplification method used for the data filtration was the process of uniform gridding, where a regular grid uses samples in x and y. From the bathymetric data set provided from the cropped data set we can develop a bivariate function represented as H (x, y) = z, where H is the underlying surface provided and the point (x, y, z) lies on the actual surface. Since the bathymetric data set is a large-dense scattered data plot, the points on the surface were reconstructed using a Euclidean interpolator where each point that was selectively chosen was closest to the z average. In theory, this means the reconstruction operator T maps a function defined over a scattered set of points in a continuous domain (a function such as H) to a function defined at every point in the domain. Thus, the principality behind this criterion by building a triangulation of the surface to yield values of points that are not necessarily part of a grid. The z average is chosen with minimal deviation known as the shoal bias figure 2 diagram which represents minimum depth. In addition, this is the notion of triangular irregular network mapping, which is often referred to as (TIN) mapping. In the following sections, the basis of a height field will be described in more detail and how a height field relates to a minimum spanning tree. In addition, the simplification algorithms such as a randomized quick sort and Delaunay triangulation will also be examined in more detail, which will lead to the final analysis of a terrain simplification algorithm.

A description of bathymetric data

To understand the acquisition of sonar data you must first understand the meaning of sonar and the analysis techniques used to manipulate bathymetry. First, there are two types of sonar, which are known as active and passive sonar. The passive sonar is capable of only hearing sounds that are generated externally, where such indications of this aspect could be a submarine, fish, etc. Thus, these sounds are accumulated through a hydrophone, and these sounds are formed in a direction of one-way travel time. The one-way travel time indicates the time it takes the sound to travel from the source. However, for the acquisition of bathymetric data, the analyzed type of sonar used was active. Active sonar distributes a signal, which it generates over a period. The signal of the active sonar is transmitted through the water where it will reflect off-targets. The reflection of these echoes is retrieved by the active sonar's hydrophone and manipulated. The instrumentation of an active sonar system is composed of a projector and a hydrophone, which are referred to as transducers. The projector transforms the electric signals into underwater pressure waves, which are more commonly referred to as sound waves. The hydrophone, however, does the opposite in its retrieval of bathymetric information, where it is used to retrieve information. In addition, in some cases, the transducers can perform both tasks of transmission and retrieval, and later in the article will indicate how it affected the bathymetric data set. The design or more commonly the shaping of the sonar can be used to form a variety of beam types and shapes that are dependent upon transducer manipulation. Transducers can transmit signals in isotropic or directional beams which can be displayed as a time impulse or as continuous signals. The power level of the signal can be handled as a controlled output. This leads to our discussion of sound waves and more effectively illustrates from the graphic examples that element factors, such as the fluid medium, temperature, and depth are an indication of how the sonar data are retrieved. First, a sound wave is categorized as a wavelength and a frequency. For our sonar, the number of waves that pass any specific point in one second is known as the frequency of the wave. The frequency of the wave is measured in cycles per second or Hertz. The distance between the waves is the wavelength and the rate at which the waves pass is an indication of their speed; for our purpose, a sound wave propagates at 1500 meters per second as the velocity of sonar. The cogent reason for understanding sonar is to foster a correct implementation process for the data from computational space to a virtual environment. The diligent detail to sonar acquisition is the basis for formulating correct gridding tactics and will help to guarantee finer resolution of the data in question.

Understanding of resolution of sonar data

To first understand the resolution of sonar data the issue is to understand the type of active sonar that was needed to measure the bathymetry. As stated, earlier in the introduction the type of sonar used to acquire the bathymetry was the usage of a multibeam sonar. A multibeam sonar system acquires bathymetry by providing a fan-shaped coverage of the ocean floor, and as stated earlier, the output of the data is in the form of depths rather than as images. In addition, the multibeam system measures and records the time for the acoustic signal to travel from the transmitter or transducer to the ocean floor or object, i.e. Liberty Vessels, and back to the receiver. As shown in figure 2 and 3, the illustration of the multibeam sonar signature is apparent as it is easily noticed by the shaping of the imagery acquired by the sonar. In addition, multibeam sonars differ from side-scanning sonars by being attached to the survey vessel, as opposed to being towed from behind. Theoretically, sonar forms a point source of energy (transducer) which will radiate outwards from the wavefront forming a spherical shell. As the radius of the shell increases the sound intensity decreases. A sample model of the spreading loss from the unit range of R initial to a range of R is 10 log where: $10\log((R^2)/(I^2))=20\log R$.

Some energy is lost to the air where sound waves are diverted into a layer of water, energy is lost because of friction in molecules, chemical changes in seawater, foreign material, etc. The concept of the coverage area on the ocean floor is a ratio that is dependent on the depth of the water, which is usually four times the water depth and illustrates the powerful significance of this type of active sonar. The unification between frequency and sound range clarifies a huge variety of conations for the use of active sonar. Thus, if a sonar wishes to retrieve transmit sound pulses at long ranges while covering a significantly wide area in a short duration of time, then a low-frequency source is best to use. The low-frequency source is best because it attenuates high-frequency signals and hence has a lower noise ratio. For the acquisition of the two Liberty Vessels, there is a clear indication that high-frequency noise was present in the data collection. As you will notice from figure 3, it shows clear indications of outliers that were presented which are an indication of noise or measurement error, it also illustrates the poor resolution of the sonar in certain aspects of the accumulation of the bathymetric data.

The poor resolution of the bathymetric data is not because of the graphics but is an indication of how the sonar retrieved the data. Low-frequency sound typically has longer wavelengths and often longer pulse widths, which indicates the amount of time the sonar is active, it (sonar) grants a lower resolution of the bathymetric information. Thus, the low-frequency soundwave tends to have a smoothing process on the bathymetry causing the edges to not have a sharper resolution. The process of gridding or binning the data must be utilized to ameliorate the active sonar acquisition of the data. The target of the sonar also presented a dilemma since the two Liberty Vessels were submerged within the ocean surface. To pinpoint the target accuracy of the two Liberty Vessels an accurate acoustic profile must be displayed. From figure 2, many tracks were passed over the two Liberty Vessels to obtain a significant acoustic profile. Also, many variables affect the sonar data based on the phenomenon which affects the resulting sonar records, which depict the seabed and its various targets. The elemental factors such as the environmental conditions which are wind, waves, currents, density gradients, form temperature, and salinity changes, are all aspects that affect the quality and interpretability of the sonar data. The elemental factors affect the resolution of the sonar data predominantly. This is because after the acoustic energy is reflected from the bottom and waterborne discontinuities are received by the transducers, it is displayed on the monitor. Thus, when a sound pulse is emitted from the transducers, the signal it attenuates is propagated very rapidly. The absorption reduces the strength of the outgoing pulse and then returns echoes due to physical and chemical elements within the ocean. Since the bathymetry measurements taken were done in the South Horn Gulf of Mexico, the process of absorption is much more rapid than in fresh water. From figure 2, the elemental factors provide a significant reason for the low resolution in certain aspects of the bathymetry data set. The absorption causes a linear reduction of echo amplitude with range when measured in decibels (dB). Since beam spreading is an apparent process in the sonar readings the absorption causes an exponential reduction of returned energy with respect to range. These apparent distortions within the sonar readings are indications of environmental factors but are also dependent upon the yaw and pitch of the sonar. In addition, another factor that caused distortions in the acquisitions of sonar data is known as range data compression. Thus, each data point within the bathymetric data set has some form of range data distortion that is dependent upon lateral, or range, distortion. Later, in the paper, this will lead to the aspect of pattern clustering and how the effects of distortion cause problems in terms of image compression.

Graphical resolution and sonar acquisition

Resolution in any sonar system is not only a function of the sonar beam but is also a function of the display mechanism. Therefore, when using a video display on a graphics software package such as OpenGL, it is imperative to keep in mind that a constant size display has a limit to pixel size and that shorter ranges provide higher displayed resolution. Thus, the resolution of each pixel is a function of the range divided by the number of pixels in the display that are spread across that range. The bathymetry data set collection of the two Liberty Vessels was considered small in comparison to its time and size duration of approximately 4 million points. The significance of shorter ranges is that shipping time for the area covered is increased for small object resolution and reduces the effect of signal loss and noise. This is primarily due to insonifications tactics, more commonly known as pings. The ping rate is dependent upon the discretion of the operator of the sonar therefore if the insonification is set at lower ping rates then the target in question may not even be recognizable at all.

Uniform gridding

The theory and analysis of gridding is the basis for approximating any large-scale data set. The data set being analyzed is compromised within two domains corresponding to its massive size and density. The underlying principle of uniform gridding is based upon separating the diagonals of rectangles in a Cartesian grid. The Cartesian grid is based on the sorting of two monotonically increasing sets where x_i, i=1,..,n and y_i, j=1,..,m.

Thus, the gridded points have explicit coordinates (x_i, y_j) that determine a cellular matrix of the domain which consists of rectangles. The simple uniform gridding is done by the forming of one of the edges with one of the diagonals of the rectangular cells which form the triangulation of the domain. In this application and as well as in others where terrain meshing algorithms are needed a mathematical approach for designing the grid is used, such as a criterion where a minimum jump is in the normal vector, etc. For our illustration and purpose, we chose the triangulation for the uniform gridding based on the Euclidean estimator algorithm, where a point was chosen based upon its bias closest to the z average. Moreover, the aspect of fitting partial differential equations to a structured grid or the fitting of large data sets to a structured grid by interpolation illustrates the need for the credulous argument of curvilinear grids. A curvilinear grid is specified with two geometric principles or arrays where x_{ij}, y_{ij}, i=1,...,M; j=1,...,N, where each cell within the matrix C_{ij} consists of the quadrilateral with the boundary illuminated by (x_{ij}, y_{ij}) which maps to ($x_{i+j,}$ y_{i+j}), which then maps to (x_{ij}, y_{ij}). Thus, it is presumed that these four points from a polygon that is based on the criteria that these points which form the polygon are nonintersecting. In addition, this places certain constraints on the formation of the number of geometric cellular triangulation within the curvilinear grid. An example of this process is posted in Scientific Visualization [2]. As noticed the cellular matrix forms, the entire triangulation process known as our domain, which is more explicitly known as the union of all the cells which can be triangulated within the matrix.

Problems of uniform gridding

The uniform gridding method used earlier is a typical form of refinement method, however, this will not suffice for our data in question. First, this is a large dense scattered data plot taken with a multibeam sonar, that has included a given number of facets or holidays within the image readings. A uniform gridding approach will not render a correct approximation of the data set, primarily because in the areas of no information or data points it gives false illusions or spikes. This represents that a more appropriate algorithm must be used to explain this bathymetric data set with better resolution.

Later, in the paper, the importance of error measures because of gridding will be discussed, but for now, the topic of non-uniform gridding will be analyzed. Basically, there are two primary forms of gridding visualization. The first discussed is used primarily in the analysis of partial differential equations, and the second non-uniform gridding typically uses Delaunay or the Voronoi triangulation for scattered data plots.

Overall, the purpose of this paper is to find the most effective greedy insertion principle using the Delaunay algorithm to approximate our bathymetric data set. The Delaunay Triangulation from Theorem 5.7.2, O'Rourke [6]. The Delaunay triangulation of a set of points in two dimensions is precisely the projection to the xy- plane of the lower convex hull of the transformed points in three dimensions, transformed by mapping upwards to the paraboloid $z = x^2 + y^2$. This means that the Delaunay triangulation algorithm works by the principle that all perimeter edges of the containing polygon are analyzed on the criterion of mathematical validity, that if an edge is valid. The edge must first pass the circle test. In addition, if edges from P are not inscribed or lie outside the circumcircle of the triangle that is on the opposite side of the edge from A. "All invalid edges must be swapped with the other diagonals of the quadrilateral containing them, at which point the containing polygon acquires two new suspect edges. The process continues until no suspect edges remain" O'Rourke [6]. With respect to the convex hull in three dimensions, this applies to the region of the circumcircle. More explicitly it is the defining criterion of the Delaunay triangulation because it is based on the parabolic principal $z = x^2 + y^2$, thus a map of the parabolic is defined as (x_i, y_i) (x_i, y_i, x_i^2, y_i^2). Thus, the convex hull is based on a set of three-dimensional points and with disregard for the "top" faces of the hull. All those faces whose outward normally points upward, in the aspect of being a positive dot product with the z-axis vector. The equation of the tangent plane above the point (a, b) is $z = 2ax + 2by – (a^2 + b^2)$, and more explicitly this follows directly to the equation $z = 2ax – a^2$.

Designing a curvilinear grid

As previously discussed, this gives a general overview of the understanding of the design for a uniform grid. However, we must now explore how to design a particular gridding algorithm to accommodate the two Liberty Vessels bathymetric data sets. Meaning, that it is now time to explore a more rigorous approach illustrated in VectorGrid.c, which illustrates the complexity in design for a large dense scattered data plot.

First, the transformed region is always comprised of contiguous rectangular blocks as shown in figure 2. The occurrence is because one of the curvilinear coordinates is parametrized by a constant on each segment of the physical boundary. Thus, each segment of the physical boundary corresponds to a plane segment of the boundary of the transformed region which is parallel to a coordinate plane in that region. The completion of the boundary of the transformed region then is composed of a series of plane segments that intersect all at right angles. In addition, the coordinates of the transformed region can thus be considered simple counters identifying the points on the grid. Where the idea being the transformed region is comprised of a collection of rectangular blocks. It is therefore easy to hypothesize that to identify the grid points with integers values of the curvilinear coordinates within each block. As illustrated in VectorGrid.c. In addition, it also clarifies where to place the Cartesian coordinates of a grid point in S_{ijk}, where the subscripts (i, j, k) here indicate the position of (S_i, S_j, S_k) in the transformed region. Also, within each cell or block, the curvilinear coordinates are then taken to vary as S_i= 1, 2, …, I over the grid points, where I is the count of the number of points within the S_i direction. As seen within the program VectorGrid.c, it illustrates the aspect of partitioning each gridded point within each cellular region, Moreover, for grid points on a boundary segment of the transformed region, the gridded point will be placed in S_{ijk} with one index fixed. The cells of the transformed region will also be composed of four exterior boundaries, of which all are plane segments connected at right angles. The occurrence of interior boundaries can be avoided if desired by dividing the cells into the nomenclature of smaller cells. The boundary segments in “the transformed plane may correspond to actual segments of the physical boundary or may correspond to cuts in the physical region”, Thompson [8]. In addition, these cuts may not necessarily be physical boundaries but are interfaces across the field within the bathymetry. An illustration showing an aspect of interfaces across the boundary is shown in figure 2, where shown in the cellular region with respect to -88.75 and 30.17 degrees there are shown interfaces across the boundary of the bathymetry. As shown in illustration figure 2.

In addition, a boundary segment in the transformed cellular region corresponds to such a cut that it then interfaces across one cell which relates to complete continuity to another cell, or to another cell side which corresponds to its own region. Thus, the dependence is placed upon the type of grid generation used, the Cartesian coordinates of the grid points on a physical boundary segment which may either be specified as “may be free to move over the boundary in order to satisfy a condition, such as orthogonality, or the angle at which coordinate lines intersect the boundary” Thompson [8]. To set up the configuration of the transformed region, a mapping must be established between each (exterior or interior) segment of the boundary or a segment of a cut in the physical region. Meaning, that the first step is to position points on the physical boundary, or on a cut, which are to correspond to corners of the transformed region where it can either be exterior or interior. As an example, consider a two-dimensional simple connected region, where four points on the physical boundary are selected to correspond to the four corners of the empty rectangle that forms the transformed region. Now, consider any of these for points, where one specifies a curvilinear coordinate which will run from that point to one of the two neighboring corner points, while the other cell point will run to the other neighbor. Thus, the mapping of the points that correspond to the coordinates will connect to the opposite pairs of corner points. In addition, since the curvilinear coordinates are to be assigned integer values at the grid points, “Si is to vary from 1 at one corner to a maximum value, I1, at the next corner, where I1 is the number of grid points on the boundary segment between these two corners. Therefore, proceeding clockwise from the lower left corner, the Cartesian coordinates of the four corner points are placed in $S_{i,\ 1}$, $S_{i,j}$, and $S_{I,1}$ where I1 = I and I (squared) = J” Thompson [8].

Divide and conquer algorithms

As noticed the size of these bathymetric data sets plays a central role in the complexity and the amount of time it takes the CPU to run algorithms to do these triangulations. Before these triangulations can be tested on the bathymetric data sets it must first satisfy the optimal time in terms of complexity. Meaning, that before any reduction algorithm can successfully optimize the data sets it must be first evaluated in terms of running time processes. Thus, one of the first criteria to maximize the best optimal running time complexity is to use a quick sort algorithm on the initial bathymetric data set, which is used as a preprocessing approach before sending the data set to a uniform or non-uniform grid structure. The concept of curvilinear grids and nonuniformity is spoken of in the paper before divide and conquer algorithms. This primarily illustrates how large data sets are filtered into rectangular gridding algorithms, therefore meaning the user needs to know exactly what type of gridding algorithms they intend to use before sorting the initial data set.

The general approach of a divide and conquer algorithm is to divide the data set into subsets of A and B, solve the problem for A and solve the problem for B and merge the results into a solution of *AUB*. For the bathymetric data sets containing the two Liberty vessels, the data set was purposely quickly sorted before the initial gridding process took place, to minimize the running time complexity of the reduction algorithms. This process of quick- sorting the bathymetric data sets partitions the bathymetry of x, y, and z triplets into a filtered process, this makes it easier for such processes as the Euclidean estimator algorithm to find the point closest to the z average and ameliorates the resolution of the image, by selectively being able to estimate the cell size for the gridding algorithm.

One will notice that this leads to a form of data mining and moreover, this becomes an illustration of data dependent triangulation where the bathymetric data set determines the criteria for the type of triangulation and gridding algorithms needed. The use of divide and conquer algorithms are the predecessors for heuristic terrain simplification algorithms and reduce complexity when using massive data sets. Procedures reduce the demand of trying to partition the entire data set using traditional dynamic programming. The process of linearly interpolating each x, y, and z triplet is ameliorated by using a divide and conquer algorithm to sort the data and filter it into the appropriate gridding algorithm.

Data-dependent triangulations

Understanding data-dependent triangulations means the understanding of function analysis dependent upon the make-up of the data set. The modeling function for a data-dependent triangulation is based upon a bivariate function of (F_i, x_i, y_i), I =1,…,N. A novel approach is to define a schematic function which is to first form a triangulation of a convex hull of the independent data (x_i, y_i), I =1,…,N, and then define F to be a piecewise linear over this triangulation. This yields a continuous function that interpolates the data. Since any form of triangulation is to form a minimum spanning tree, the purpose is to form an acyclic network, which optimizes the least, the weighted cost function of the triangular irregular network. Therefore, the basic idea of a data-dependent triangulation is that any triangulation of the independent data (x_i, y_i), i=1,…,N will suffice for this approach, however, the credence is in selectively choosing the modeling bivariate function F (x_i, y_i) must be chosen carefully to do the triangulation. Choosing a triangulation method whether it be Delaunay, Voronoi, or some other method, depends upon the ordering which is imposed on the collection of all possible triangulations within the convex hull within the triangular domain. The first criteria is to find a local cost function for each edge which is denoted by S (F_t, e_i). The goal is to find an optimal data-dependent triangulation which is defined to have the smallest associated vector with the sum of possible least weighted cost functions through the triangular irregular network. In addition, it is known that a global minimum exists within the triangular irregular network even though it may not be unique, and it may not be so easy to find computationally.

Data mining tactics

Designing large bathymetric data sets to form a particular height field often is the result of noticing clumps or clusters in the bathymetry, which are intended for visual analysis. Often clustering procedures are used as a form of data mining to recognize much more intricate structures within the bathymetry. Previously, in the paper, we mentioned using the Euclidean estimator algorithm to recognize the point closest to the z average. The Euclidean estimator algorithm is analogous to the nearest-neighbor algorithm in the sense that it can be viewed as a minimum spanning tree. Given a minimal spanning tree, we can find the data clustering produced by a reduction algorithm, such as the Delaunay, or Nearest-Neighbor algorithm.

The process for the minimal spanning tree algorithm works as follows. Given a set of data points, which are strung out into long chains, a minimal spanning, tree forms a natural skeleton for the data domain. If we designate the diameter path as the longest path through the trees, then a chain will be characterized by the shallow depth of branching off the diameter path. For example, Duda and Hart [1] state, in contrast, for a large cloud of data points, the trees will usually not have an obvious diameter path, but rather a several distinct, near-diameter paths. For any of these, an appreciable number of nodes will be off the path. While slight changes in the locations of the data points can cause major rerouting of a minimal spanning tree, as stated in Duda and Hart [1]. Thus, the use of minimal spanning tree algorithms is useful because it grants statistics on edge length distribution. From this example represented is a diagram of a dense cluster that is embedded within a sparse one. Primarily, the lengths of the edges of the minimal spanning tree exhibit two distinct clusters that would easily be detected by a minimum-variance algorithm. Therefore, by deleting all edges longer than some intermediate value, we can extract the dense cluster as the largest connected component of the remaining graph.

The significant relation of this to the Liberty Vessel bathymetric data set is that the bathymetry was composed of a large, dense, scattered data field where the data itself had to be non-uniformly gridded to get a better resolution of the data set. The use of clustering in connection with a minimal spanning tree algorithm clarifies the significance that certain regions, i.e., Liberty Vessels and artifacts, can be selectively chosen and gridded using the techniques of clustering within a minimal spanning tree algorithm. Furthermore, the aspect of error measure associated with this form of triangular irregular network gridding brings forth the precarious nature of how the data must be selectively chosen from a particular cluster within the bathymetric data set. If the clustering is not done with careful analysis and if the minimal spanning tree algorithm does not ameliorate the least-weighted path through the triangular irregular network, then the quality control of the bathymetric image is extremely difficult.

In conclusion, it is easily discerned that the minimal spanning tree forms the correct framework for the careful reduction of a bathymetric data set. It is the careful analysis of estimating edge weight with respect to topology growth that formulates the design of an optimal minimal spanning tree. In this project, two criteria form the basis of analysis; the spacing of the data set nodes and the weighted-edge length that formed the path to each node. If the path length from the edge node to its neighboring node is optimal, then an efficient minimal spanning tree is designed.

Error measures

Fuzzy modeling procedures are often in a series of clusters based upon a priority. The priority ranks clusters in terms of hierarchical importance to improve resolution within areas of significant importance. The problem in this article is to increase the resolution of the Liberty Vessels and consider their masses within the bathymetry data as areas of critical analysis. Thus, the understanding of error measures based upon the sage of a Euclidean interpolator minimal spanning tree reduces the error and compensates for the gridding algorithm by producing a finer resolution containing the significant Liberty Vessel clusters. The problem is one of isolating the two Liberty Vessels using the aspect of clustering algorithms to find the minimal spanning tree with the least weighted edges to approximate the data set; a different approach of lower-dimensional space was used to interpolate the data set. A method to attack this problem is to try to represent the data points as points in some lower-dimensional space in such a way that the distance between points in the lower-dimensional space corresponds to the dissimilarities between the points in the original space. Using the low-dimensional and multidimensional scaling approach verifies our need for clustering because the Euclidean Estimator algorithm was used primarily for the purpose of selecting data values closest to the z-average. In addition, Duda and Hart [1] also state, "The general process of finding a configuration of points whose interpoint distances correspond to dissimilarities is often called multidimensional scaling".

The preceding argument introduces the topic of error functions and the role it represents in the area of data analysis. For the sake of example, we shall assume that the distances between the n samples $x_1,\ldots,x_n$ and $y_1,\ldots,y_n$, for which the n(n-1)/2 distance d_{ij} between image points are as close as possible to the corresponding original distances. These functions involve only the distances between points and are invariant to the rigid-body motions of the configurations. Thus, all these functions are normalized such that their minimum values are invariant to the dilatation of the sample points.

Hierarchical cluster reduction

In using pattern classification using dimensional reduction is that a clear representation of the data is a concern. Therefore, the greatest emphasis is placed upon groups of clusters with the greatest variability, where the greatest difference is in terms of class relative to the standard deviations, thus Duda and Hart [1] coined this as "multiple discriminant analysis".

Procedure hierarchical cluster reduction

The following algorithm to produce cluster reduction [1]:

1. Let d = d and Fi = (xi), I = 1,….d.

2. If d = d', stop

3. Compute the correlation matrix and find the most correlated pair of distinct clusters of features, say fi and Fj.

Where the distinct clusters of features are the two Liberty Vessels

4. Merge Fi and Fj, delete Fj, and decrement d by one

5. Go to Loop.

Cluster reduction

The idea of clustering using this argument has two strong fallacies in terms of bathymetric data set analysis. The first impediment to the bathymetry of the two Liberty Vessels is that there is a small deviation in data analysis. Therefore, it means even for the bathymetry that is coded into a sorted list, the deviations within the data set are so minimal that agglomerative reduction can either have a very minimal effect on terrain simplification, or a subversive effect if done out of proportion.

Hierarchical clustering

In this section, hierarchical clustering is presented as a substitution for agglomerative-based clustering. The idea of clustering is to focus on the aspect of terrain simplification but also utilize the aspect of pattern recognition to clarify areas of significant interest. The ideology of using hierarchical clustering is analogous to the scheme of using variable-length coding procedures, where areas of interest are coded with a longer bit frequency than that of the redundant areas. The procedure of variable length coding as an aspect of hierarchical clustering is useful for this bathymetric data set because the two primary areas of interest are the two Liberty Vessels. However, the problem of isolating the two Liberty Vessels from image space to computational data point analysis presents an aspect of pattern recognition.

The first criterion of hierarchical clustering procedures is to divide the areas into two distinct classes, agglomerate and divisive. An agglomerative or bottom-up, clumping procedure starts with singleton clusters and forms the sequence by successively merging clusters. The use of divisive clustering (top-down, splitting) procedures starts with all the samples in one cluster and forms the sequence by successively splitting clusters. To analyze the two Liberty Vessels the choice of Agglomerative clustering was utilized primarily because the purpose of moving from one clustering level to the next is computationally more feasible. Thus, to manipulate the bathymetric data set into selective sampling clusters, the process of interpolating the bathymetric data set into selective sampling clusters, the process of interpolating the bathymetric data set to isolate certain parameters is a necessity.

The usage of a nearest-neighbor function was initiated where points were selected with respect to the closest average of the depth buffer. This means that the acyclic graph of the triangular irregular network was based upon a process of selecting a certain node length with respect to the minimal depth average. It is a computationally hard problem to distinguish the distance between all the clusters within the bathymetric data set. And in addition, it is an NP-hard problem to selectively distinguish the removal of points from a data set based on the aspects of one algorithm. Therefore, the principle behind analyzing the bathymetric data set is to selectively distinguish clusters based upon an agglomerative process. The use of an agglomerative process where the bathymetric data set had a clustering section based upon the nearest neighbor algorithm. (See code Vectorgrid.c in Appendix) The second pattern recognition process was based upon clustering sing higher frequency data associated with the two Liberty Vessels. The higher frequency data is associated with the Liberty Vessels primarily because the insonifications from the sonar grant a higher resolution of the Liberty Vessels within the image based on the reflectance of their metallic surface. The clustering procedure transcends into the data set because it grants clustering criteria already established on the basis of pattern recognition with respect to the frequency domain of the bathymetry.

Theory of pattern recognition

In terms of the graph-theoretic methodology, one can associate the thresholding distance with the fact that a data point can be indicative or associated with a cluster. This is based on the principle that the distance between respective bathymetry points is less than a certain gamma function. Thus, these criteria can be generalized to mean that to judge whether any independent points within the data set belong to the same cluster. The function of S_{ij} represents a matrix that depends upon the similarity graph in which nodes correspond to points and an edge joins node I and node j if and only if $S_{ij} = 1$. Therefore, for the clustering algorithm, a similarity graph was designed to fulfill the criteria of bathymetric data points for clusters. In addition, this builds a “single-linkage algorithm”, where for two samples alpha and beta are in the same cluster only if and only if there exists a chain where alpha_1,...,alpha_k such that alpha is similar to alpha1 and to alpha2. And in conclusion for the remainder of the chain. For a complete linkage algorithm, all samples in a given cluster must be similar to one another, and no sample can be in more than one cluster. As noticed from program terrain Simplification.c there is a clustering algorithm that implements this form of data association after the respective Delaunay triangulation is formed. The clustering algorithm is performed directly after the Delaunay triangulation as a pattern recognition approach for clustering but also for the emphasis of developing a better minimal spanning tree algorithm to bolster the Delaunay triangulation. Therefore, the nearest-neighbor algorithm can be interpreted as an algorithm for designing or finding a minimal spanning tree. In addition, by implementing a minimal spanning tree it is easier to find clustering produced by the nearest-neighbor algorithm and the data-dependent triangulation. In further analysis, by observing the structure of a minimal spanning tree it can easily be deduced that removal of the longest edge will produce the two-clustering grouping, and therefore removal of the next longest edge produces the three-clustering group. Finally, the analogy grants that for a minimal panning tree to be established it is considered an undirected graph, such that the graph produces G = (V, E) where V is the set of nodes or points and E is the set of possible interconnections between pairs of nodes. For each edge known to be a (u,v) element of E, there is a weight function associated with the cost to connect u and v. The emphasis is to design an acyclic graph or subset T intersects or equal to E that connects all of the vertices and whose total weight is minimized: w(T) = Summation q(u, v). The minimal spanning tree is formed when T forms an acyclic graph and connects all the vertices, and is moreover considered to span the graph of G.

Cost of Delaunay triangulation algorithm

The importance of cost analysis is to form an optimal spanning tree that formulates a Euclidean estimator that compensates for edge weight cost within the minimal spanning tree and forms an optimal Delaunay triangulation to render the best resolution of the data set. To locate one point in the Delaunay mesh domain there are certain factors that must be parameterized, such as time to locate a point, time to insert a vertex in the mesh, and a factor that represents the number of passes. Within each pass, a classification costs in three criteria where the selection is done to pick the best point. Next, an insertion to insert a vertex into the mesh is performed, and finally, the errors at grid points are recalculated. To find the best points, a pass or scanning through O (n) points reforming comparisons is completed. Then a performance of a single mesh inserted which has a cost represented as K. For every unused point, of which there are O (n-i) = O (n), we must also perform a recalculation. This is shown in the code for the Delaunay triangulation:

```
Int x, y, z
Int vnum = 0;
While ( scanf (“%d %d”. x, y) != EOF)
v = MakeNullVertex();
v-v[X] = x;
v-v[Y] = y;
z = x*x +y*y;
v-v[Z] = z;
v-vnum = vnum++;
```

performing comparisons performs the analysis of recalculation to just whether or not vertice is justified.

```
v = vertices;
xmin = xmax = v-v[X];
do
if( v-v[X] greater than xmax)
xmax = v-v[X];
else
if( v-v[X] less than xmin )
xmin = v-v[X];
v = v-next;
} while ( v!= vertices );
```

find Y min and max

```
v = vertices;
ymin = ymax = v-v[Y];
do
if( v-v[Y] greater than ymax ) ymax = v-v[Y];
else
```

Citation: Wilbert A McClay and Elaine Stewart McClay. “Manipulating Sonar Bathymetry into Virtual Reality Using Adaptive Terrain and Graph-Theoretic Meshing Algorithms to Acquisition World War II U.S. Navy Liberty Vessels in the Gulf of Mexico Utilizing the Hadoop Ecosystem”. *EC Oceanography* 1.1 (2025): 01-27.

```
if( v-v[Y] less than ymin ) ymin = v-v[Y];
v= v-next;
v0 = vertices;
while ( Collinear( v0, v0-next, v0-next-next ) )
if ( ( v0 = v0-next ) == vertices )
v1 = v0-next;
v2 = v1-next;
```

Therefore, for the algorithm stated above, the comparison and recalculation involve a mesh location where the results are queried to find the points containing the triangle and interpolation to find the value of the approximation at that point. Thus. The cost for each location is K and the cost for each interpolation is O(1). The cost for location might increase with successive passes, since the number of vertices in the triangular terrain mesh at the beginning of the pass is i+4, making the cost of recalculation o each pass equal to O(nK). The worst-case cost of location, K, and the cost of insertion, I, are mesh dependent. Consider a planar triangulation with v vertices total, a significant number of vertices on the boundary, having 3vv(boundary)3 edges and 2v-v(boundary)-2 triangles. The assumed or typical approximation for the cost of the vertices on the boundary is v(b) = $O((v)^{1/2})$, in which case the number of edges is approximately 3v and the number of triangles about 2v, however, the number of edges and the number of triangles is representable each with a cost of O(v). The aspects of point location within the Delaunay mesh are the usage of the simple "walking method" stated by Garland and Heckbert [3], where their ascertains are that the algorithm starts on an edge of the mesh, and iteratively walks through the mesh toward the target point until it arrives at the target. Garland and Heckbert [3] state that if it were to start in a very deleterious spot, with a mesh of i+4 vertices, it might have to walk across almost all O(i) edges. In addition, Delaunay mesh insertion involves locating the containing triangle at a point, inserting a new vertex there, and possibly swapping some edges. The worst estimate of complexity that is dependent upon location will require O(i) time, and O(i) edges will have to be swapped. Moreover, Delaunay mesh location and insertion can require the time linear in the correspondence with the number of points in the triangular mesh, and K = I = O(i) complexity. Therefore, the worst estimate, for the costs per pass for the Delaunay mesh algorithm is O(n) for selection, O(i) for insertion, and O(in) for recalculation. The dominant factor is based in terms of recalculation; therefore, the total worst-case cost is O(in) = $O(m^2n)$. However, the suspected worst-case complexity is very unlikely for most situations if the algorithm is written for the best case for the Delaunay triangulation. It is the expected cost behavior that must be analyzed since the Delaunay triangulation is a minimal spanning tree algorithm. The expectation for the cost for a random-access location of the Delaunay mesh with v vertices for a typical point distribution is only K = $O((n)^{1/2})$. But, if successive location assignments are in proximity, then the location procedure can start its search at the triangle returned by the previous call, so that very few steps will be needed to find the next target point. Moreover, the expected cost of location is K = O(1). The Delaunay mesh triangulation algorithm used thus far, without terrain simplification, uses point location assignments where almost always they are near one another. Almost all the point location assignments are made in the process of scanning the unused points. The scanning procedure moves in a process across each row in order, where the successively scanned points are almost always nearby and are usually direct neighbors. With this consideration, it is possible to construct meshes in which both insertion and location will have linear cost. In fact, there is no precedence that states that this form of the linear cost will not occur. The insertion of the location of the query assignments involves a location that does not illustrate as much spatial coherence as location queries due to recalculation. Thus, the expected cost is approximately K = $O((i)^{1/2})$. The other cost of insertion is due to the edge swaps, and on average, the number of edge swaps is constant, with correspondingly constant time. Therefore, the process of using insertion with the Delaunay triangulation grants the expected cost for insertion to be I = $O((i)^{1/2})$. The assumption also that K = O(1) and I = $O((i)^{1/2})$ in the expected case, the costs per pass for the Delaunay mesh algorithm are O(n) for selection, $O((i)^{1/2})$ for insertion, and O(n) for recalculation. Therefore, the dominance terms are the process of selection and the process of recalculation, verifying the total excepted cost with respect to complexity to be O(n) = O(mn).

Faster complexity of Delaunay

The algorithm for the Delaunay Triangulation stated above gives the general synopsis for the complexity of the generalized Delaunay Triangulation. However, for our purpose, the usage of the Delaunay algorithm from O'Rourke [6] has a complexity of O(nlogn) using the aspect of incremental Delaunay Triangulation. For the sake of discussion, one can easily notice that the generalized complexity of the Delaunay algorithm has an expensive best-case estimate of O(MN), and a worst-case complexity of $O(m^2n)$ which is profligate in terms of complexity. The incremental Delaunay triangulation algorithm follows the scheme that an edge is valid if it passes the circle test: if A lies outside the circumcircle of the triangle that is on the opposite side of the edge from A, as stated in previous sections. Therefore, all invalid edges must be swapped with the other diagonal of the quadrilateral containing them, at which point the containing polygon accepts the two new suspect edges. The following node implements:

```
do

tEdge temp;

temp = e->next;

if ( e->adjface[0]->visible and

e-> adjface[1]->visible )

 e interior: mark for deletion.

e->delete = REMOVED;

else if ( e->adjface[0]->visible ||

e->adjface[1]->visible )

e border: make a new face.

e->newface = MakeConeFace( 1, p);

e = temp;

while ( e = !=edges);

return TRUE;
```

The process continues until no suspect edges remain.

Manipulation of the Delaunay algorithm

The process of insertion demands that after the point is inserted it will have edges that emanate from the corners of a surrounding polygon. Thus, the polygon defines the area in which the triangulation has been altered, and hence it defines the area or region, in which the approximation has changed. Garland and Heckbert [3] note this as the update region. The change in coherence is what allows the process of significant optimization of the Delaunay algorithm. Therefore, an array is defined denoted as Cache which contains error values indexed by the set of input points. Moreover, with respect to our previous algorithms, the new process for insertion only needs a few modifications, where a GREEDY INSERT to initialize the Cache and change Insert to update values in Cache that have changed due to insertion and make the main loop of Greedy Insert look up values on Cache rather than calling the Error explicitly. In addition, the grid points in the update region are located by scanning the triangles surrounding the inserted point. Moreover, the update region is overestimated by computing the bounding box of these triangles. The Garland and Heckbert [3] algorithm follow the principle:

```
Faster Delaunay Algorithm
Insert(Point p);
    Mark p as used
    Mesh_Insert(p)
    for all points q inside the triangles
    incident on p
        Cache[q] := Error(q)
Greedy Insert;
    Initialize mesh to two triangles w/height field grid for all input points p
        Cache p := Error(p)
    while not Goal_Met do
        best:=nil
        maxerr:=0
        forall input points p
            err:= Cache p
            if err > maxerr then
                maxerr:= err
                best:= p
        Insert best
```

Locality manipulation

In conclusion, with the original algorithm, the three categories with respect to selection, insertion, and recalculation are unchanged, but now the performance will result in less computation, reducing the complexity. If A is the area of the update region on pass i. Therefore, we are still doing a location query for each recalculated point, so the cost per pass for recalculation is O(AK) for location and O(A) for interpolation. In the worst case, is the area to be updated reduced by only a constant amount on each pass, so A = O(n-i). Hence, this is only true in certain abstract cases, it is still a possibility. The cost for recalculation is the same as in the original algorithm, and it remains the dominant cost, therefore the worstcase complexity for the faster Delaunay algorithm is still unchanged with a complexity of $O(m^2n)$.

The minimal spanning tree of the Delaunay triangulation

In previous sections, the greedy divide and conquer algorithms were interpreted as predecessors to the heuristic terrain irregular network mapping algorithm. The minimal spanning tree algorithm is the subset for the Delaunay triangulation algorithm and forms the field of the mathematical basis for the formation of triangular irregular network mapping. In addition, without totally understanding the purpose or knowledge of a minimal spanning tree the implementation of non-uniform gridding tactics would be irrelevant and the algorithmic complexity with respect to gridding and optimal network topology would be compromised significantly. A minimal spanning tree is a subset of the Delaunay triangulation: minimal spanning tree (MST) subsets D(P). where D represents the acyclic graph with P nodes, O'Rourke [6]. A minimal spanning tree has many implications throughout computer science for approximating network topology. The network topology referred to in this paper is the triangular irregular network known as TIN mapping. The minimal spanning tree algorithm has optimal significance because by manipulating the edges in the network topology. Thus an (MST) or minimal spanning tree is a set of points where a minimum length tree covers all the points:" the shortest tree whose nodes are precisely those in the set" O'Rourke [6]. The estimation of edge length relative to the minimal spanning tree utilized the Euclidean interpolator algorithm, noted in previous sections, the Euclidean estimator algorithm is important primarily because it forms the Euclidean Minimum Spanning Tree (EMST). The formation of an edge length with respect to the Delaunay triangulation is a given measurement of the Euclidean length of the segment connecting its endpoints. As noted, in earlier sections, the development of the entire triangular irregular network was formulated based on the criterion for data selection involved using a Euclidean interpolar algorithm to decimate the data set and formulate a height field. Once the height field is determined, it can be deduced that its relation relative to a minimum spanning tree forms the resultant EMST. The basis of the minimum spanning tree for the formation of a height field also forms the acyclic graph of the Delaunay triangulation, then a network topology of a Euclidean minimum spanning tree is proven.

Algorithm cost of the minimal spanning tree

The importance of cost, least-weighted edge length, and complexity are important in many graphs theoretic algorithms utilizing minimal spanning trees. For a set of points defined within a network topology, there are (n:2) edges, where the complexity of sorting is O(n^2logn). Since the type of minimal spanning tree, it caused the algorithmic complexity to reduce to O(nlogn). The reduction in complexity is a result of the Euclidean interpolator where the criterion of point selection is based upon the average z-buffer relative to the height field. Thus, the idea of a convex hull illustrates the importance of the Euclidean interpolator algorithm and how it determines the complexity of the resultant Delaunay triangulation. As noted in subsequent chapters, the convex hull of a set of data points is the smallest convex polygon in its interior. The convex hull is important in minimal spanning tree algorithms because it describes the formation of the Delaunay triangulation, as justified by, O'Rourke [6]. The maximization of the minimum angle is centered around the objective of obtaining an optimal convex hull and is the basis of the Delaunay triangulation. In Computational Geometry in C, O'Rourke [6] states, "The Delaunay triangulation T =D(P) is maximal with respect to the angle fatness relation: T greater than or equal to T' for any other triangulation T' or P". Thus, the description of angle-fatness describes the relation between two triangles of the same point within a set. In addition, the partitioning of the convex hull formulates the maximization of the minimal angle, meaning that the smallest angle is maximized over all the triangulations. Maximization with respect to the minimal angle is important because it reduces the number of slivers or very small triangles which are the result of data outliers, and it is the basis for determining an optimal terrain simplification algorithm.

Heuristic terrain simplification

The usage of faster selection and optimization is a very important part of interpretation and analysis when analyzing the Delaunay triangulation algorithm. Since the Delaunay triangulation algorithm is a minimal spanning tree, the usage of an optimization algorithm makes it superb for terrain simplification algorithms. The minimal spanning tree is defined as a connected, undirected graph, and each bathymetry point can be similar to a set of pins in a graph G = (V, E), where V is the set of pins, and e is the set of possible interconnections between pairs of pins for respective (u, v). Thus, for each edge (u, v) there is a weight w (u, v) specifying the cost needed to connect u and v. The use of an optimal data-dependent Delaunay triangulation chooses and removes a weighted edge dependent upon its parent node, where an edge is selectively chosen if its weight is greater than its parent's node edge. The meaning of this algorithmic process is to design an optimal triangulation that reduces the number of indices concerning edge length based upon the parent node within the triangular irregular network. Also, this is a form of terrain simplification since it lowers the aspect of redundancy. The terrain simplification procedure is also analogous to a heap extraction procedure, where the growth of the heap per sweeping is bounded by a constant value or the edge length from the previous parent node. Also, the entire net growth is the number of triangles recorded and the compression, or terrain simplification is based upon the number of indices removed concerning the number of indices from the original minimal spanning tree or triangular irregular network.

Performance evaluation

The terrain simplification procedure algorithm takes an optimal triangulation procedure since it reduces the number of triangulations within the original minimal spanning tree. But the algorithm used to selectively remove indices is based upon a greedy strategy, where indices are removed with respect to their original parent node and not the initial key node of the entire minimal spanning tree algorithm. The node's edge is considered safe if its edge does not violate the invariant parent edge of the triangular irregular matrix, and each step of including or removing indices forms a subset with the minimal spanning tree algorithm. Thus, noted from previous sections which examined the cost and complexity of a given Euclidean minimal spanning tree. The reduction of vertices is based upon obtaining the maximal for the minimal angle to reduce the number of vertices by obtaining larger triangles. As noticed from the Delaunay image. There is a stronger density of triangles within the major quadrants of the image, but in the areas of facets and holidays, there is a slightly noticeable but minimal change within the image, as shown in figure 4.

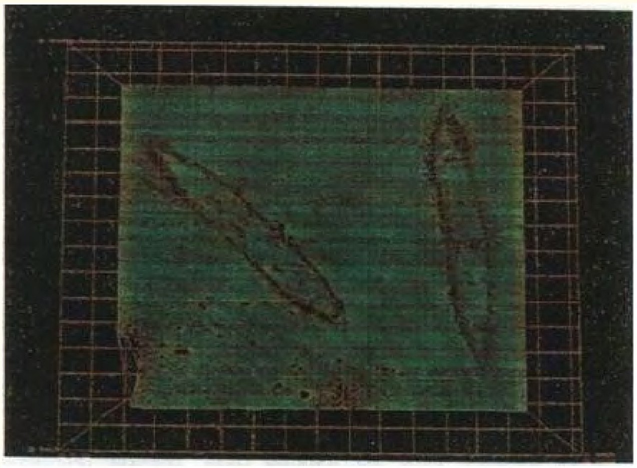


Figure 4: *Delaunay triangulation.*

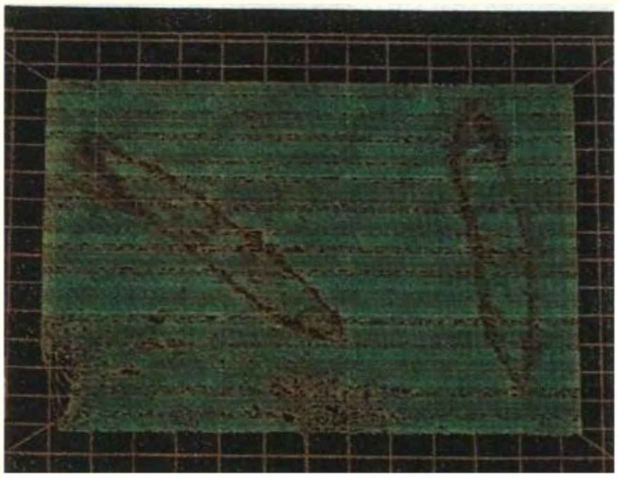


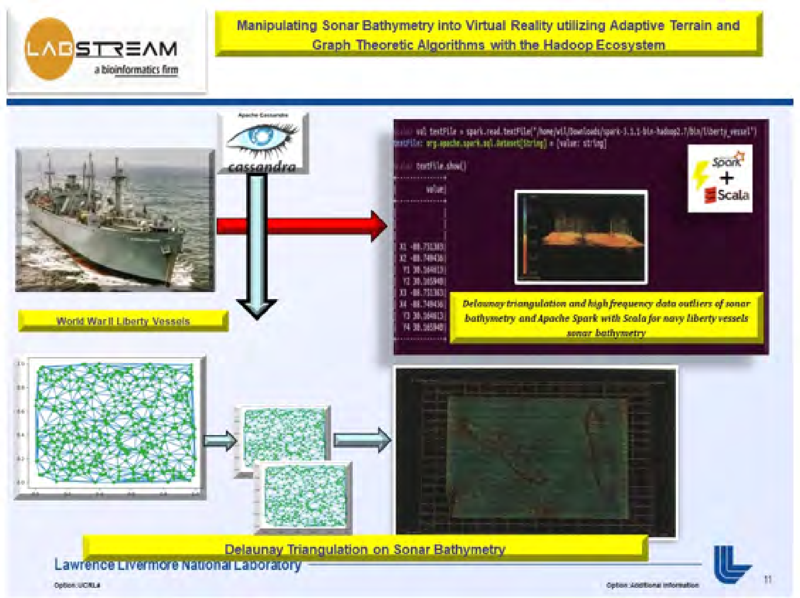
Figure 5: *Terrain simplification.*

The effect of the network topology differed in regions of apparent facets that were also explicit in the original image. This is because a significant number of indices were removed from the triangular irregular network. The minimal spanning tree had fewer nodes to form triangular connectivity. This illustrates the sparse density of triangles and illuminates the regions of facets and holidays much more apparent since these regions typically had edge lengths with a higher cost than their parent node. In addition, the acquisition of the bathymetric data concerning temperature, insonifications procedures, and salinity also compromised the resolution of the data and illustrated the track separation or scalloping effect of the multibeam sonar. The approximation of the terrain simplification algorithm granted excellent results concerning triangular compression, where the original image was composed of 167,358 triangles. In the terrain simplification image, the number of triangles was reduced to 83,679 triangles, which is a 50% compression ratio.

Hadoop ecosystem

Hadoop 1.0.0 was originally released by Apache in 2011, consisting of mainly the Hadoop Distributed File System (HDFS) and MapReduce. The Hadoop Platform soon became realized as an ecosystem, which is constantly evolving where each unit in the ecosystem facilitated a specific data analysis or data storage need. We selected the Hadoop Ecosystem as a data analysis warehouse because of its scalability, performance, and fault tolerance. The Hadoop Ecosystem represents data in terms of key/value pairs. The utilization of the Hadoop NoSql database(s), HBase, data is represented as a collection of wide rows. The atomic structures in HBase make global data processing using MapReduce and row-specific reading/writing using HBase is more demanding [9,13]. In this paper, we presented the acquisition of the Sonar Bathymetry into Virtual Reality utilizing Amazon AWS Apache Cassandra and Apache Spark with Scala as essential tool(s) in the Hadoop Ecosystem that provides an CQL (Cassandra Query Language) dialect for querying data stored in the Hadoop Distributed File System (HDFS), other filesystems that integrate with Hadoop, such as MapR-FS and Amazon's S3 and NoSQL database(s) (HBase and Cassandra).

Furthermore, the acquisition of the bathymetric data concerning temperature, insonifications procedures, and salinity also compromised the resolution of the data and illustrated the track separation or scalloping effect of the multibeam sonar. The approximation of the terrain simplification algorithm granted excellent results with respect to triangular compression, where in the original image was composed of 167,358 triangles was easily parsed into the Hadoop Ecosystem with Apache Spark Directed Acyclic Graph Engine with Apache Scala utilization to formulate Resilient Distributed Data Sets to demonstrate quintessential optimization and scalability for sonar bathymetric data compression with the terrain simplification imaging the number of triangles within a reduced scale to 83,679 triangles, which is a 50% compression ratio, illustrated in figure 1b, while the Hadoop file for the coordinates is illustrated in figure 6a, 6b and figure(s) 7a-7h.



Manipulating Sonar Bathymetry into Virtual Reality utilizing Adaptive Terrain and Graph Theoretic Algorithms with the Hadoop Ecosystem

```
scala> val textFile = spark.read.textFile("/home/wil/Downloads/spark-3.1.1-bin-hadoop2.7/bin/liberty_vessel")
textFile: org.apache.spark.sql.Dataset[String] = [value: string]

scala> textFile.show()
+-------------+
|        value|
+-------------+
|             |
|             |
|             |
|X1 -88.751365|
|X2 -88.749436|
|Y1 30.164613|
|Y2 30.165040|
|X3 -88.751365|
|X4 -88.749436|
|Y3 30.164613|
|Y4 30.165040|
+-------------+
```

Delaunay triangulation and high frequency data outliers of sonar bathymetry and Apache Spark with Scala for navy liberty vessels sonar bathymetry

World War II Liberty Vessels

Delaunay Triangulation on Sonar Bathymetry

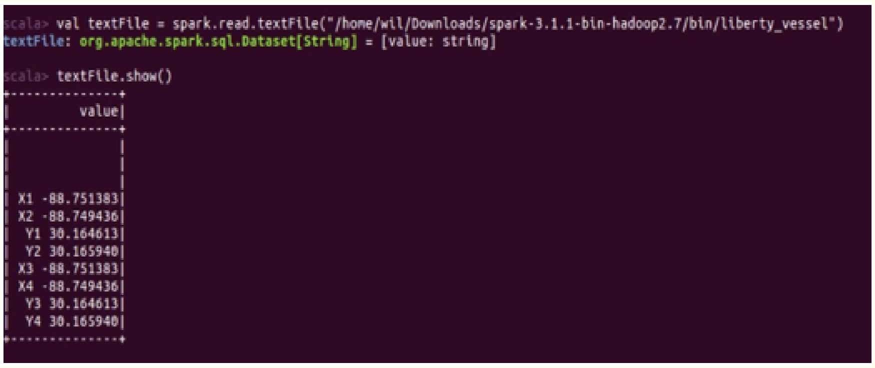
Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional information

11

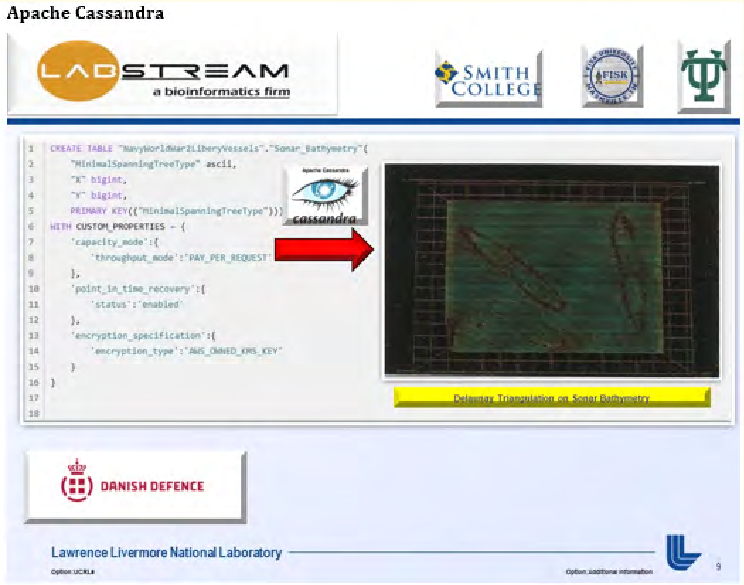
Figure 6a: *Hadoop Apache Spark Scala.*



```
scala> val textFile = spark.read.textFile("/home/wil/Downloads/spark-3.1.1-bin-hadoop2.7/bin/liberty_vessel")
textFile: org.apache.spark.sql.Dataset[String] = [value: string]

scala> textFile.show()
+--------------+
|         value|
+--------------+
|              |
|              |
|              |
| X1 -88.751383|
| X2 -88.749436|
|  Y1 30.164613|
|  Y2 30.165940|
| X3 -88.751383|
| X4 -88.749436|
|  Y3 30.164613|
|  Y4 30.165940|
+--------------+
```

Figure 6b: *Hadoop Apache Spark Scala Script.*



Apache Cassandra

```
CREATE TABLE "NavyWorldWar2LiberyVessels"."Sonar_Bathymetry"(
    "MinimalSpanningTreeType" ascii,
    "X" bigint,
    "Y" bigint,
    PRIMARY KEY(("MinimalSpanningTreeType")))
WITH CUSTOM_PROPERTIES = {
    'capacity_mode':{
        'throughput_mode':'PAY_PER_REQUEST'
    },
    'point_in_time_recovery':{
        'status':'enabled'
    },
    'encryption_specification':{
        'encryption_type':'AWS_OWNED_KMS_KEY'
    }
}
```

Delaunay Triangulation on Sonar Bathymetry

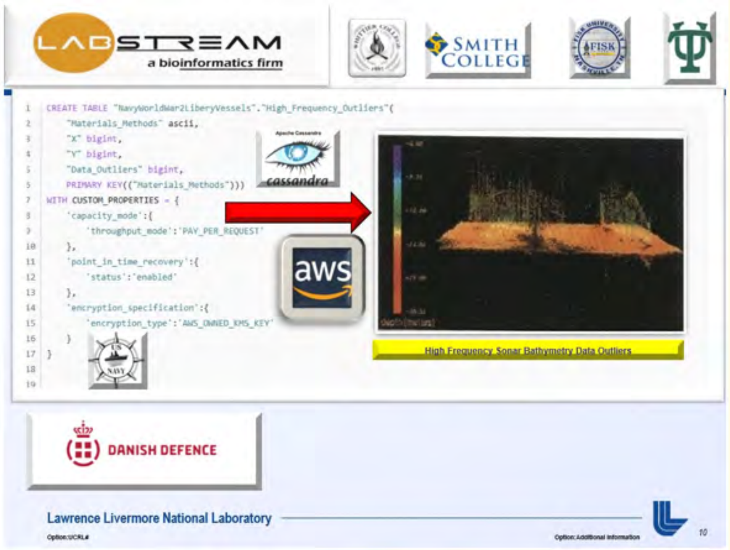
Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional information

9

Figure 7a: *Apache Cassandra Navy Liberty Vessels Sonar Bathymetry Keyspace(s) and Table(s).*



```
CREATE TABLE "NavyWorldWar2LiberyVessels"."High_Frequency_Outliers"(
    "Materials_Methods" ascii,
    "X" bigint,
    "Y" bigint,
    "Data_Outliers" bigint,
    PRIMARY KEY(("Materials_Methods")))
WITH CUSTOM_PROPERTIES = {
    'capacity_mode':{
        'throughput_mode':'PAY_PER_REQUEST'
    },
    'point_in_time_recovery':{
        'status':'enabled'
    },
    'encryption_specification':{
        'encryption_type':'AWS_OWNED_KMS_KEY'
    }
}
```

High Frequency Sonar Bathymetry Data Outliers

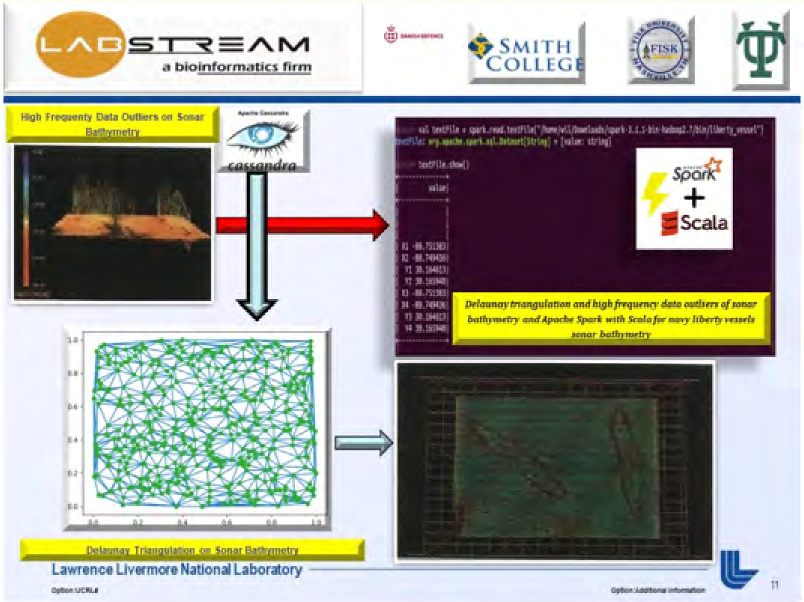
Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional information

10

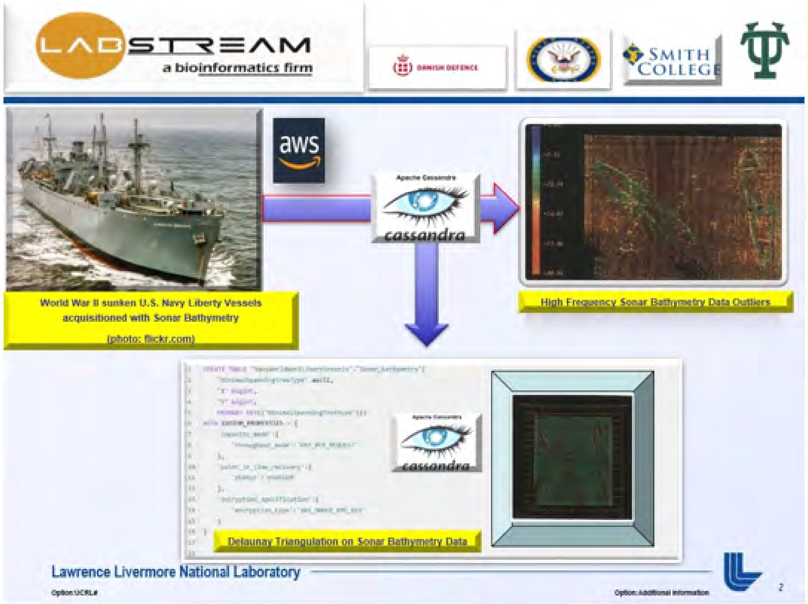
Figure 7b: *High frequency data outliers of sonar bathymetry and Apache Cassandra Navy Liberty Vessels Sonar Bathymetry Keyspace(s) and Table(s).*


LABSTREAM
a bioinformatics firm
SMITH COLLEGE
FISK
High Frequency Data Outliers on Sonar Bathymetry
Apache Cassandra
cassandra
Spark
+
Scala
Delaunay triangulation and high frequency data outliers of sonar bathymetry and Apache Spark with Scala for navy liberty vessels sonar bathymetry
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Delaunay Triangulation on Sonar Bathymetry
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional information
11

```
[illegible] val textFile = spark.read.textFile("/home/wil/Downloads/spark-3.1.1-bin-hadoop2.7/bin/liberty_vessel")
textFile: org.apache.spark.sql.Dataset[String] = [value: string]

[illegible] textFile.show()
+------------+
|       value|
+------------+
|            |
|            |
|            |
|X1 -88.751383|
|X2 -88.749430|
|Y1 30.164613|
|Y2 30.165940|
|X3 -88.751383|
|X4 -88.749430|
|Y3 30.164613|
|Y4 30.165940|
+------------+
```

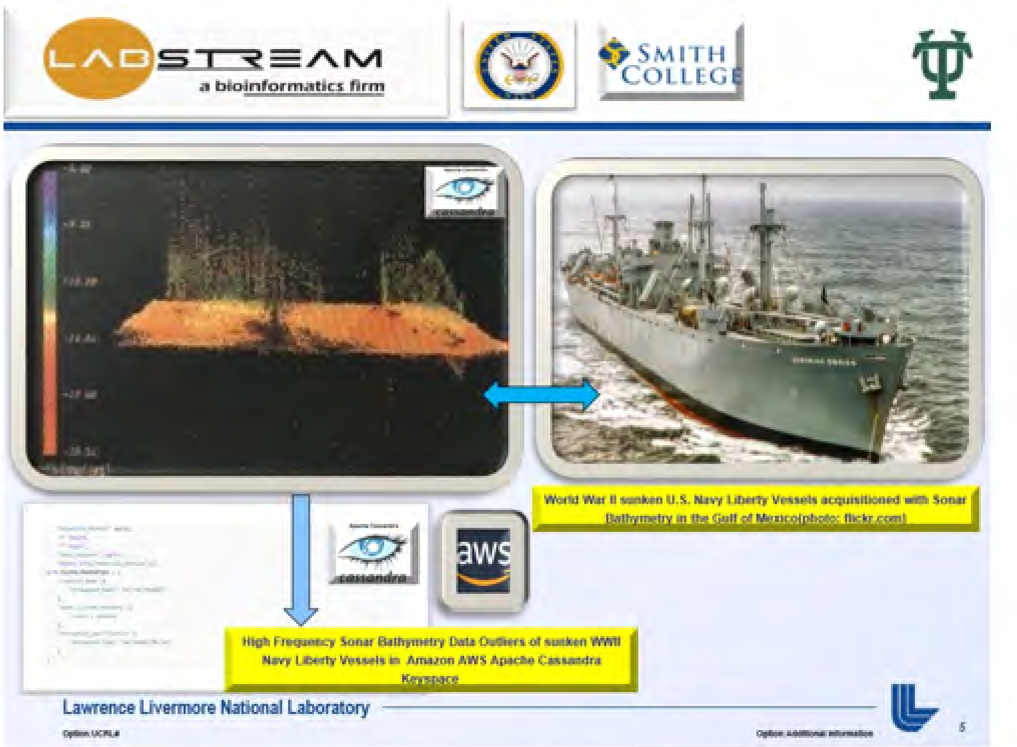
Figure 7c: *Delaunay triangulation and high frequency data outliers of sonar bathymetry and Apache spark with scala for navy liberty vessels sonar bathymetry.*


LABSTREAM
a bioinformatics firm
DANISH DEFENCE
SMITH COLLEGE
aws
Apache Cassandra
cassandra
World War II sunken U.S. Navy Liberty Vessels acquisitioned with Sonar Bathymetry
(photo: flickr.com)
High Frequency Sonar Bathymetry Data Outliers
Apache Cassandra
cassandra
Delaunay Triangulation on Sonar Bathymetry Data
Lawrence Livermore National Laboratory
Option:UCRL#
Option:Additional information
2

```
CREATE TABLE "[illegible]"."Sonar_Bathymetry"(
    "MinimalSpanningTreeType" ascii,
    "X" bigint,
    "Y" bigint,
    PRIMARY KEY(("MinimalSpanningTreeType")))
WITH CUSTOM_PROPERTIES = {
    'capacity_mode':{
        'throughput_mode':'PAY_PER_REQUEST'
    },
    'point_in_time_recovery':{
        'status':'enabled'
    },
    'encryption_specification':{
        'encryption_type':'AWS_OWNED_KMS_KEY'
    }
}
```

Figure 7d: *Delaunay triangulation and high-frequency data outliers of sonar bathymetry with Apache Cassandra Keyspaces and Apache Cassandra Tables and Apache spark with scala for navy liberty vessels sonar bathymetry.*

Citation: Wilbert A McClay and Elaine Stewart McClay. "Manipulating Sonar Bathymetry into Virtual Reality Using Adaptive Terrain and Graph-Theoretic Meshing Algorithms to Acquisition World War II U.S. Navy Liberty Vessels in the Gulf of Mexico Utilizing the Hadoop Ecosystem". *EC Oceanography* 1.1 (2025): 01-27.



World War II sunken U.S. Navy Liberty Vessels acquisitioned with Sonar Bathymetry in the Gulf of Mexico(photo: flickr.com)

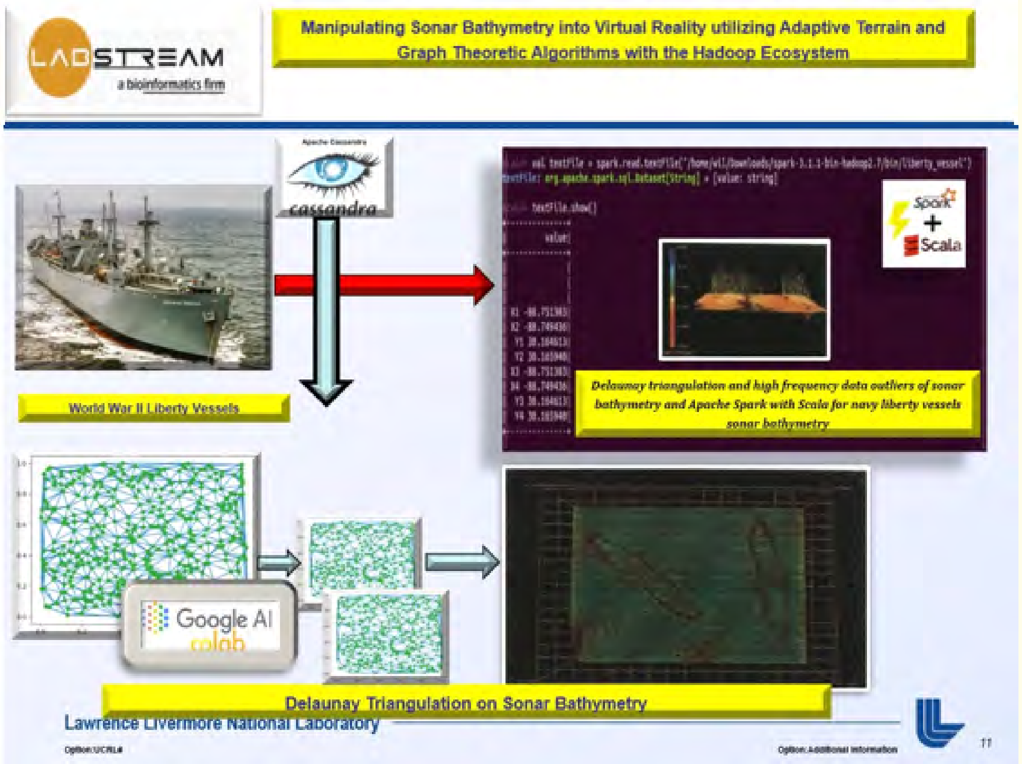
High Frequency Sonar Bathymetry Data Outliers of sunken WWII Navy Liberty Vessels in Amazon AWS Apache Cassandra Keyspace

Lawrence Livermore National Laboratory

Option: UCRL#

Option: Additional information

5



Manipulating Sonar Bathymetry into Virtual Reality utilizing Adaptive Terrain and Graph Theoretic Algorithms with the Hadoop Ecosystem

```
scala> val textFile = spark.read.textFile("/home/wil/Downloads/spark-3.1.1-bin-hadoop2.7/bin/liberty_vessel")
textFile: org.apache.spark.sql.Dataset[String] = [value: string]

scala> textFile.show()
+-------------+
|        value|
+-------------+
|             |
|             |
|X1 -88.751383|
|X2 -88.749436|
|Y1 30.164613|
|Y2 30.165946|
|X3 -88.751383|
|X4 -88.749436|
|Y3 30.164613|
|Y4 30.165946|
+-------------+
```

World War II Liberty Vessels

Delaunay triangulation and high frequency data outliers of sonar bathymetry and Apache Spark with Scala for navy liberty vessels sonar bathymetry

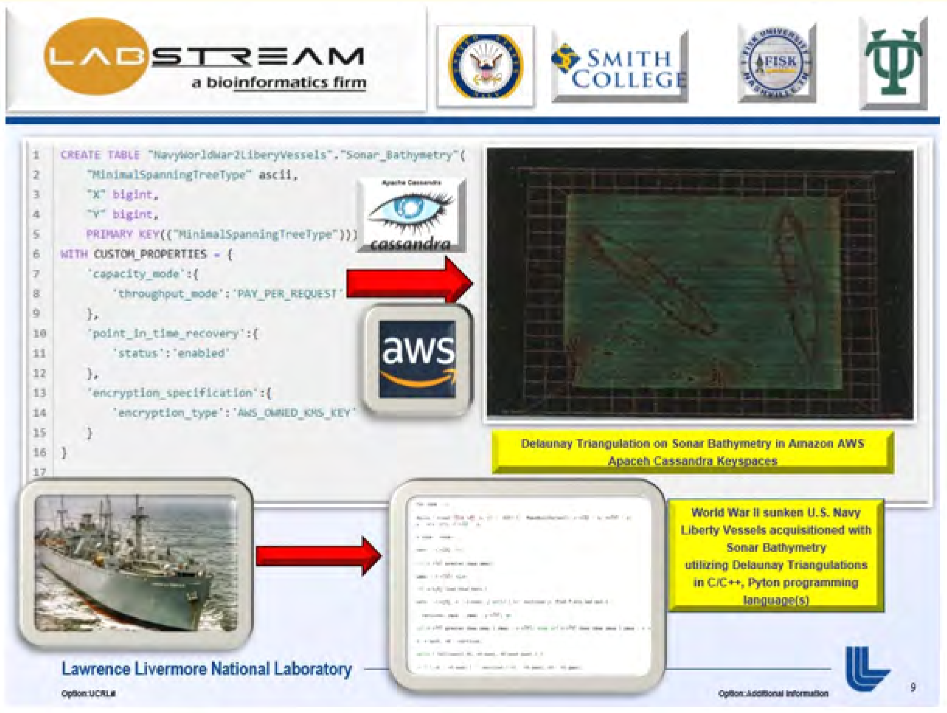
Delaunay Triangulation on Sonar Bathymetry

Lawrence Livermore National Laboratory

Option: UCRL#

Option: Additional information

11



```
CREATE TABLE "NavyWorldWar2LiberyVessels"."Sonar_Bathymetry"(
    "MinimalSpanningTreeType" ascii,
    "X" bigint,
    "Y" bigint,
    PRIMARY KEY(("MinimalSpanningTreeType")))
WITH CUSTOM_PROPERTIES = {
    'capacity_mode':{
        'throughput_mode':'PAY_PER_REQUEST'
    },
    'point_in_time_recovery':{
        'status':'enabled'
    },
    'encryption_specification':{
        'encryption_type':'AWS_OWNED_KMS_KEY'
    }
}
```

Delaunay Triangulation on Sonar Bathymetry in Amazon AWS Apaceh Cassandra Keyspaces

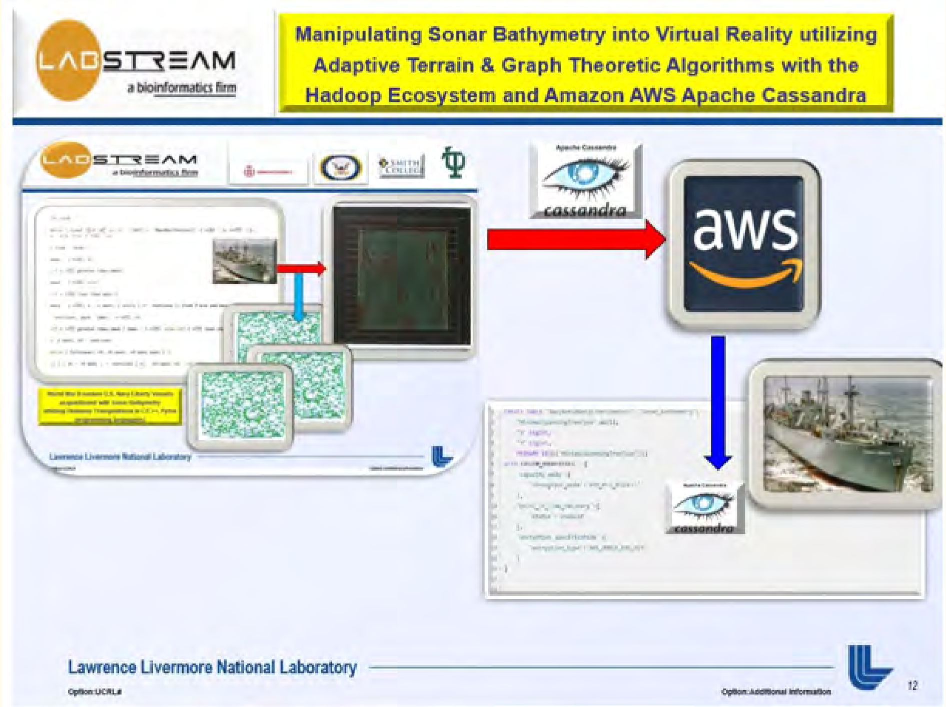
World War II sunken U.S. Navy Liberty Vessels acquisitioned with Sonar Bathymetry utilizing Delaunay Triangulations in C/C++, Pyton programming language(s)

Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional Information

9



Manipulating Sonar Bathymetry into Virtual Reality utilizing Adaptive Terrain & Graph Theoretic Algorithms with the Hadoop Ecosystem and Amazon AWS Apache Cassandra

Lawrence Livermore National Laboratory

Option:UCRL#

Option:Additional Information

12

Figure 7e-7h: *Delaunay triangulation and high-frequency data outliers of sonar bathymetry, Delaunay Triangulation iteration(s) with Apache Cassandra Keyspaces and Apache Cassandra Tables and Apache Spark with Apache Scala for Navy Liberty Vessels Sonar Bathymetry.*

Conclusion

The greedy terrain simplification and Delaunay Algorithm could be utilized in other ways to manipulate and extrapolate information from a bathymetric height field. Certain examples of other applications would be to try and preserve a certain region to eliminate finer connectivity and diminish aesthetic appeal based upon the slope of the edges and not edge weight cost. However, the purpose of this project was to reduce a significant number of triangles within the non-uniform minimal spanning tree, but still, maintain optimal triangular irregular network and aesthetic appeal. The resolution of the image rendered was compromised particularly in areas of significant outliers and slivers causing the holiday and faces within the image to be more apparent. The necessity to pursue more rigorous and scientific development in triangular irregular networks has been thoroughly recognized. The results of the heuristic terrain simplification used for this project were analogous to the Jarvis method where vertices are reduced based upon angle-fatness, yet for this application, the basis was edge length cost done in comparison to the parent node of each index. The result of this heuristic algorithm produced a 50% compression tactic but utilized optimal connectivity and still gave an accurate approximation of the data set. Initial future concerns in triangular irregular network mappings (TINS) are to reduce the number of slivers which is formed primarily because outliers tend to have low angle fatness. The casual effect of slivers is mainly an aspect of data-dependent triangulations where high-frequency noise, holidays, and outliers are the significant contributions of these data error based triangulations or slivers. An approach to diminish the effect of slivers is to swap edge lengths or formulate an optimal triangulation based totally on the criterion of optimal angle fatness. The heuristic terrain simplification algorithm which this project utilized diminished edge length based upon edge weight cost concerning the invariant parent node indices. Another factor to take into consideration for future work is the problem of high-frequency noise associated with the data. High-frequency data is a significant contribution to outliers and are the initial causes of slivers. The high-frequency data is the factor of a data dependent triangulation process where point selection is ideally based upon formulating a convex hull or the smallest polygon enclosing a set of points. Yet, the dilemma of designing an optimal convex hull is compromised when high-frequency data or outliers are devised in the data point selection. The process of removing all the outliers from the data without causing significant lossy compression to the data set is difficult, and the association of high-frequency data to its point selection within a convex hull is an ideal consideration. The method to remove the outliers concerning point selection within a convex hull is to use the least-squares method for fitting an optimal data-dependent triangulation stated by Garland and Heckbert [3]. The fundamental idea of a triangular irregular network topology is that it is the inherent offspring of a minimal spanning tree. For hierarchal-based clustering, a Euclidean minimal spanning tree proved useful in determining the distinctive clusters of points of the two Liberty Vessels within minimal error.

Notably, the approximation of the terrain simplification algorithm granted excellent results with respect to triangular compression, where the original image was composed of 167,358 triangles that were easily parsed into the Hadoop Ecosystem with Apache Cassandra KeySpaces and Apache Cassandra tables with Apache Spark Directed Acyclic Graph Engine with Scala utilization to formulate Resilient Distributed Data Sets to demonstrate quintessential optimization and scalability for sonar bathymetric data compression using latitude and longitude coordinates in the South Horn of the Gulf of Mexico, illustrated in figure 6a with the terrain simplification imaging the number of triangles within a reduced scale to 83,679 triangles, which is a 50% compression ratio.

The minimal irregularity of the network topology with respect to the facets and holidays in the results of sonar data acquisition, high-frequency data, and outliers are noted as future concerns to further this project.

Author Contributions

Conceptualization: W.M.; data acquisition: W.M; writing: W.M. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

The U.S. Navy Naval Oceanographic Office (NAV)) at Stennis Space Center, Mississippi specifically NAVO Director(s), Dr. Al Lewando, PhD, MBA, MPA, Dr. Martial Carr, PhD., Dr. Ed Beeson, PhD, Andy Haas M.Sc.Eng, and NAVO/Tulane University Tenured Advisor Professor(s)

Andrew Martinez, PhD, Dr. Judith Maxwell, PhD, Dr. Fred Petry, PhD,. Additionally, Law Enforcement/Security Manager, Ms. Shannon Collison, U.S. Navy & Tulane University Alumni/Amazon AWS Chief Computer Scientist, Dr. Orlando Karam, PhD, Labstream Corporation/Brandeis University Alumni Attorney, Rayee Lumer, Esq, my younger brother(s) University of Missouri/Tulane University Alumni, Akil McClay, MHA, M.Sc., Louisiana State University Alumni, Nazareth McClay, Emory University Alumni, Gabadiah McClay, Attorney/Aunt Regina McClay, Esq, Aunt Velma Stewart, M.Ed & Cousin, Ms. Kia Johnson Stewart, Mrs. Towando Woods, M.Sc., Uncle Carlton & Aunt Verla McClay, (grandparents), World War II U.S. Navy Seabee, Wilbert McClay, Sr & Mrs. Albertha Henderson McClay, and (grandparents) World War II Louisiana Sheriff Deputy John E. Stewart & Mrs. Velma Stewart. Finally wish to highly acknowledge the United States Navy Department of Defense Graduate School Programs, Danish Defense Forces in the Country of Denmark, my National Merit Scholar Alumni, Fisk University Alumni, Whittier University Alumni, Smith College School of Social Work Graduate Alumni Mother, Mrs. Elaine Stewart McClay, MSW, LCSW, and Southern University Alumni, Meharry Medical College Alumni/U.S. Army Captain/Physician, Father, Dr. Wilbert McClay, Jr., M.D., support this paper.

Conflicts of Interest

The authors declare no conflict of interest.

Data Availability Statement

The datasets generated and/or analyzed during the current study are available from the corresponding author (Wilbert A. McClay[1-15]) on request.

Bibliography

1. Duda and Hart. "Pattern classification and scene analysis". Addison and Wesley (1982).
2. Duda., *et al*. "Pattern classification". John Wiley & Sons (2012).
3. Earnshawn RA., *et al*. "Scientific visualization". Addison and Wesley (1992).
4. Garland and Heckbart. "Fast polygon and approximation of terrain and height fields". Carnegie Mellon (1997).
5. Abraham Kandel. "Fuzzy mathematical techniques with applications". Addison and Wesley (1986).
6. Craig Lindley. "Practical image processing in C". John Wiley and Sons, Inc., (1991).
7. O'Rourke. "Computational geometry in C". Second Edition, Cambridge University Press (1998).
8. Ronald L Rivest., *et al*. "Introduction to algorithms". Massachusetts Institute Technology Press (1993).
9. Thompson., *et al*. "Numerical grid generation". North-Holland (1985).
10. McClay WA., *et al*. "A real-time magnetoencephalography brain-computer interface using interactive 3d-visualization and the Hadoop ecosystem". *Journal of Brain Sciences* 5.4 (2015): 419-440.
11. MongoDB (2018).
12. Attias H. "ICA, graphical models, and variational methods". In: Independent component analysis: principles and practice; Roberts S, Everson R, Eds.; Cambridge University Press: Cambridge, UK (2001): 95-112.
13. Smith KT. "Big data security: The evolution of Hadoop's security model" (2013).

14. Rodriguez M. “Big graph data on Hortonworks data platform” (2012).

15. The Apache HBase Reference Guide, 2014 Apache Software Foundation (2015).

16. Aven Jeffrey. “Sams Teach Yourself Apache Spark”. SAMS (2016).

17. McClay WA. “A magnetoencephalographic/encephalographic (MEG/EEG) brain-computer interface driver for interactive iOS mobile videogame applications utilizing the Hadoop ecosystem, MongoDB, and Cassandra NoSQL databases”. *Diseases* 6.4 (2018): 89.

18. McClay W., *et al*. “Amplitude modulated phase only filtering and high dimensional warping for registration on MRI images”. Proceedings SPIE 6310, Photonic Devices and Algorithms for Computing VIII, 63100P (2006).

19. Wilbert A McClay. “Mobile data and individual client diagnostic acquisition from PGWS call analysis utilizing spark and the Hadoop ecosystem”. *Journal of Remote Sensing and GIS* (2016).

20. “World War II Liberty Vessel Photographic Images”, https://www.flickr.com.

21. Attias H. “Planning by probabilistic inference”. In Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA (2003).

Volume 1 Issue 1 January 2025