12-15-2007

The Hydraulic Spline: Comparisons of Existing Surface Modeling Techniques and Development of a Spline-Based Approach for Hydrographic and Topographic Surface Modeling

Maik Flanagin
University of New Orleans

The Hydraulic Spline: Comparisons of Existing Surface Modeling Techniques and Development of a Spline-Based Approach for Hydrographic and Topographic Surface Modeling

A Dissertation

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Engineering and Applied Science
Computer Science

by

Maik Flanagin

B.S. Massachusetts Institute of Technology 2000
M.Eng. Massachusetts Institute of Technology 2000

December 2007

Dedication

This dissertation is dedicated to my mother, Meta Flanagin, who provided valuable 'assistance' throughout my education.

Acknowledgments

I would like to thank the following people and organizations for their contributions and support: my adviser, Mahdi Abdelguerfi, Aurelian Grenetton, Nicholas Bioret, Jay Ratcliff, Michele Aurand, Tom Tobin, the New Orleans District of the U.S. Army Corps of Engineers, and the Naval Research Lab at Stennis Space Center .

Contents

Abstract

Creation of accurate and coherent surface models is vital to the effective planning and construction of flood control and hurricane protection projects. Typically, topographic surface models are synthesized from Delaunay triangulations or interpolated raster grids. Although these techniques are adequate in most general situations, they do not effectively address the specific case where topographic data is available only as cross-section and profile centerline data, such as the elevation sampling produced by traditional hydrographic surveys.

The hydraulic spline algorithm was developed to generate irregular two-dimensional channel grids from hydrographic cross-sections at any desired resolution. Hydraulic spline output grids can be easily merged with datasets of higher resolution, such as LIDAR data, to build a complete model of channel geometry and overbank topography.

In testing, the hydraulic spline algorithm faithfully reproduces elevations of known input cross-section points where they exist, while generating a smooth transition between known cross-sections. The algorithm performs particularly well compared to traditional techniques with respect to aesthetics and accuracy when input data is sparse. These qualities make the hydraulic spline an ideal choice for practical applications where available data may be limited due to historic or budgetary reasons.

Keywords - spline, LIDAR, interpolation, surface, triangulation, river

List of Figures

List of Tables

1

Overview

Southeast Louisiana faces a number of geographic challenges. The entire region was built from river sediment by the deltaic cycle, in which land accretes from soils deposited by the Mississippi river and subsides once the river finds a shorter flow gradient and changes course. Seasonal flooding, during which the river overtops its banks, also contributes to the land-building process, ensuring a stable system. Unfortunately, the same process that builds land also has the potential to flood properties of the local inhabitants. Construction of levees and river revetments in Louisiana under the U.S. Army Corps of Engineers Mississippi River and Tributaries Project and similar projects has mitigated the problem of river flooding, but has created other difficulties, principally the destabilization of the deltaic cycle. Without the flood-deposited sediments, levee-protected areas are sinking at an alarming rate from subsidence [1]. As southeast Louisiana sinks, it becomes more vulnerable to flooding and hurricane damage [2].

Flood control and hurricane protection projects, several of which are ongoing or completed, are vital to the region's survival. The Donaldsonville to the Gulf of Mexico Feasibility Study, conducted by the U.S. Army Corps of Engineers - New Orleans District (CEMVN)

and its partners, is currently underway to determine whether a new levee alignment would be a cost-effective solution to reduce the risk of damage from floods and hurricane-induced storm surge. This study is examining which variations in the levee alignment will produce the most benefit and how the new alignment will work in conjunction with a number of other projects, including the Morganza to the Gulf of Mexico Hurricane Protection Project [3] and the West Bank and Vicinity New Orleans Hurricane Protection Project, both of which may potentially tie into the new alignment and help reduce the storm surge risk for inhabited areas of southeast Louisiana. The Louisiana Coastal Restoration and Protection Project (LACPR) is also studying options for providing protection from 100-year event hurricanes through a combination of flood control and coastal restoration projects.

To measure the effectiveness of any course of action for these projects, hydraulic models are created to determine flood-prone areas and estimate potential damage. Two such models used by the U.S. Army Corps of Engineers (USACE) are the Hydrological Engineering Center's River Analysis System (HEC-RAS) [4], which computes water surface elevations and flow rates in river channels, and AdCirc [5], an advanced circulation modeling application that predicts the effects of hurricane storm surge on land. Constructing a quality hydraulic model depends on the development of an accurate topographic surface of the study area, with an emphasis on the waterways and adjacent banks. Generation of this surface is complicated by the necessity of combining disparate input data sources to yield a topography for the entire study area.

Two primary sources of input data are hydrographic survey data and Light Detection and Ranging (LIDAR). While hydrographic survey data is the more definitive and accurate source for channel elevations, its collection density has been historically limited by available funding. In contrast, very dense LIDAR data is cheaper to acquire and more readily available. However, LIDAR data is only reliable for unobstructed areas on land, not necessarily in river channels. This dissertation presents a technique to generate topographic surfaces for

hydraulic modeling by interpolating the sparse hydrographic survey data so that it may be merged with LIDAR data.

The rest of the dissertation further examines the difficulty of producing surfaces from LIDAR and hydrographic survey data and introduces the hydraulic spline algorithm as a means to merge data from these sources. Chapter 2 describes the data sources used in surface modeling, including cross-sections and profile centerlines. Chapter 3 describes some of the hydraulic simulation systems that require topographic surface models as data inputs, such as HEC-RAS and AdCirc. Chapter 4 reviews standard methods for processing topographic datasets and interpolating elevations of unknown point locations. Delaunay triangulation and geostatistical algorithms are discussed in detail and applied to a simple river geometry dataset. Chapter 5 introduces splines and explains the hydraulic spline algorithm and describes its processing steps. Chapter 6 continues with an algorithm specialization for addressing channel confluences. Chapter 7 provides a qualitative and quantitative assessment of the hydraulic spline and compares the accuracy of the algorithm with other surface generation techniques using root mean square error (RMSE) calculations. Finally, Chapter 8 describes the contribution of the hydraulic spline to several CEMVN projects.

2

Topographic Data Sources

Several sources of elevation data are available for constructing surface models. These sources vary greatly with respect to accuracy and data density. Specific examples used in modeling applications include the following:

2.1 Survey Data

There are two types of survey data used in hydraulic modeling: cross-sectional surveys and profile centerline surveys. Cross-sectional surveys include straight lines, called cross-sections, which are typically measured across a waterway, as shown in Figure 2.1. Ideally, survey cross-sections are oriented nearly perpendicular to the path of the river, but may vary based on conditions in the field. In almost all cases, topographic surveys augment the hydrographic surveys in cross-section overbank areas.

Profile centerline surveys include a single surveyed line called a profile centerline that is measured along the length of a waterway, typically in the middle of the channel between both banks. Sampling is usually dense enough to define the shape of the channel's path.

Depending on the scope or availability of funding, channel surveys do not always include profile centerline measurements. In these cases, the centerline must be digitized from aerial photography or acquired from other data sources.

Hydrographic surveys are normally derived from fathometer measurements collected along the cross-section of a channel for the vertical component and GPS coordinates referenced to known benchmark locations for the horizontal component. The depth measurements are subtracted from a gage reading referenced to a known vertical benchmark in order to obtain the elevations at the bottom of the channel. The survey benchmarks [6] are point locations with a known latitude, longitude, and elevation established by a geodetic authority such as the National Geodetic Survey (NGS). Most benchmark locations are tagged with a small brass or steel plate to help surveyors find them. By using benchmarks, surveyors can make adjustments to increase the accuracy of their surveys.

Although surveying provides the most accurate means of collecting data, the process is generally regarded as time-consuming, expensive, and labor-intensive. At the New Orleans District of the U.S. Army Corps of Engineers, survey data collection is either conducted by in-house staff or submitted by contractors and subjected to a quality assurance process. Survey files are received in the Corps of Engineer's standard EM file format, which is described in Engineering Manual EM 1110-1-1005 [7]. The EM file format uses coded records for survey metadata such as horizontal and vertical datums, state plane coordinate system zone, and the organization that conducted the survey. Additional coded records delimit data sections of point groups defining profile centerlines and cross-sections, in which each line following a record declaration is a comma-separated list of point identifier, easting, northing, elevation, and survey code. The easting, northing, and elevation values recorded in EM files can be parsed to assemble spatial features for cross-sections and profile centerlines.

Figure 2.1: Example of cross-sections and a profile centerline.

2.2 Airborne LIDAR (Light Detection and Ranging)

Airborne LIDAR offers a way to efficiently acquire high-resolution elevation data for large areas. LIDAR has proven itself to be cost-effective, while substantially speeding up project schedules [8], [9], [10]. Similar to RADAR systems, which emit radio waves that reflect off targets back to a receiver, airborne LIDAR systems emit laser pulses that reflect off the ground and are collected by a receiver. The lag time on the pulse returns is used to measure elevation values.

Although LIDAR can be used to collect elevation data over large areas, the fact that it is collected from the air presents difficulties in certain types of terrain. For instance, an airborne LIDAR system may not be able to fully penetrate tree canopies, reducing its usefulness in forests. In these cases, the multiple returns collected by the receiver can be processed with noise reduction algorithms such as Least Mean Squares (LMS) [11] to remove the majority of trees. However, to create a usable ground surface, LIDAR must be checked against aerial photography or ground-truthing to detect aberrations missed by automated

correction techniques.

In lakes and streams, the refractive properties of water tend to distort the results of the data collection, reducing the reliability of LIDAR data in these areas. Recent advances in LIDAR, such as the Scanning Hydrographic Operational Airborne LIDAR Survey (SHOALS) [12] include capabilities to sample coastal regions both on-shore and in the ocean. However, high sediment content and presence of floating vegetation mats in southeast Louisiana bayous continue to obstruct airborne data collection techniques.

A number of state and federal agencies, including the U.S. Army Corps of Engineers, FEMA, and the Louisiana Department of Environmental Quality, have sponsored collection of five meter resolution LIDAR for the entire state of Louisiana from 1999 to 2005. This LIDAR dataset is made publicly available on LSU's ATLAS website (http://atlas.lsu.edu) and is the source of all LIDAR examples in this paper.

2.3 SONAR (Sound Navigation and Ranging)

Multi-beam SONAR data uses sound waves to generate high-density digital elevation models underwater. The multi-beam system emits a wide-angle signal perpendicular to the path of the vessel, allowing it to collect elevation not only directly underneath the vessel, but also simultaneously within an angle to the left and right of the vessel. The U.S. Army Corps of Engineers regularly relies on SONAR data to visualize and analyze riverbeds [13], [14], for wreck management and dredging activities. Predictably, since SONAR is only collected underwater, away from the banklines, its usefulness is limited in terrain modeling. Furthermore, most ships that can be equipped with SONAR devices are too large to navigate some of the smaller bayous and canals of Louisiana.

SONAR data is often used to determine wreck locations. When visualized using SONAR point clouds or a Delaunay triangulation, channel cross-sections that do not show a typical

U-shape are likely to contain a shipwreck or other obstruction that may present a hazard to navigation. The same SONAR data, if taken from bank to bank, is commonly used as a data source from which to extract cross-sections for hydraulic modeling and bank stability analysis.

3

Hydraulic Simulation Applications

Hydraulic simulation applications are sophisticated programs that model aspects of hydraulic or hydrological systems. Simulations are normally used for tasks such as identifying flood-zones and calculating flood risk levels. Surface modeling is the most fundamental input component for hydraulic simulations.

3.1 HEC-RAS

The Hydraulic Engineering Center's River Analysis System (HEC-RAS) is an application that models water flow through channels based on one-dimensional hydraulic conservation of flow and conservation of energy equations. HEC-RAS is considered one-dimensional since lateral and semi-lateral flows are not directly modeled by the system. HEC-RAS utilizes three types of primary inputs: 1) Cross-section geometries of channels, 2) stream networks representing centerlines of channels, and 3) an input flow vector including both a water quantity and flow direction for each stream in the network (Figure 3.1). The program can model both steady-state flow, where the velocity of the water is constant, and unsteady-state

flow, where the velocity is variable and may even flow backward during a simulation. HEC-RAS simulation results are used to determine water surface elevations and wetted perimeters in a channel during a given storm event. These elevations and perimeters can be used to determine flooded areas. HEC-RAS has numerous applications in engineering and flood insurance rate determination.

For cases in unsteady-state flow modeling where water overflows the channel cross-sections, the system can be augmented with storage areas. Unlike cross-sections, which have specific spatial locations, storage areas are abstract representations of areas where a certain level of flooding occurs. Storage areas can be associated with one or more cross-sections. Water that overflows a cross-section is diverted to its associated storage areas and diverted from its storage areas when water recedes. Storage areas are defined with storage curves that map water volumes to water elevations in the storage areas.

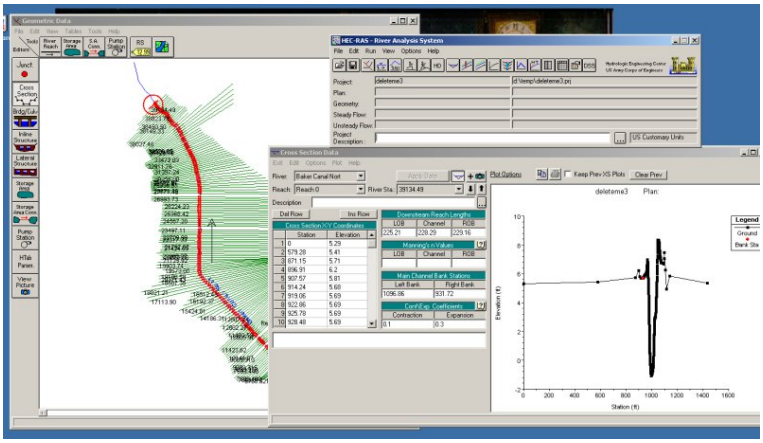

Figure 3.1: HEC-RAS model with geometry from Baker Canal North.

3.2 AdCirc

The Advanced Circulation System (AdCirc) is a finite element analysis system used to model hurricane storm surge and measure flood water inundation. The system calculates water movement on an irregular grid by solving the Generalized Wave-Continuity Equation

(GWCE) [15] and related equations using the finite element method over a series of time steps that correspond to the track of a simulated storm. Water depths are computed for each node in the network. The modeler can add weirs and boundaries to the model in order to determine flow rates across ridges in the terrain, such as roads and levees, and can input observation points to track depth changes at specific locations of interest. For example, a modeler can place observation points along the crest of an existing levee to measure the required crest height that would prevent overtopping failures.

The geometry of the AdCirc grid, like problems in finite element analysis, is modeled with an irregular grid such as the one shown in Figure 3.2. The use of an irregular grid allows the modeler to define more detail in areas of interest and less in other areas. This approach offers the best balance between precision and efficiency. In contrast, a regular grid such as a raster dataset would only have one resolution; either it would be too small to represent important details, or it would be too large to be realistically used in calculations.

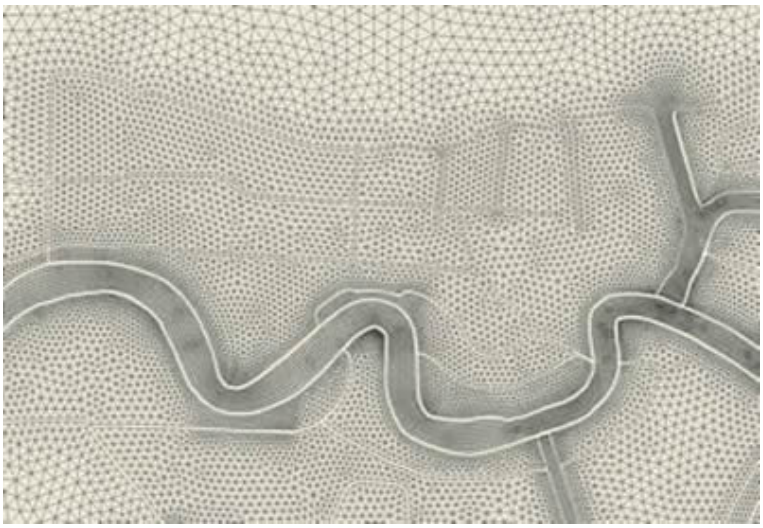

Figure 3.2: Sample AdCirc Grid in Southeast Louisiana

Details that are important to AdCirc geometry are the locations of levees and canals. Figure 3.3 shows more detail in the Orleans parish urban outfall canals and less detail in the neighborhoods between these canals. Levee points must be modeled in AdCirc as two

associated points separated by a discontinuity, which AdCirc will treat as a weir. In essence, a levee will block water until a certain overtopping threshold is reached, at which point the water will spill into the protected area behind the levee.

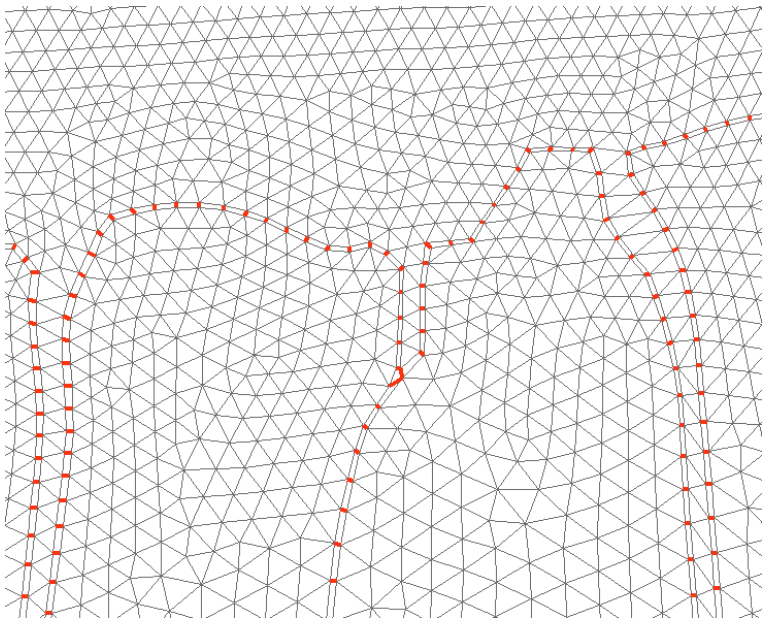

Figure 3.3: Closeup of an AdCirc Grid showing Orleans Canal, Bayou St. John, and London Avenue Canal. The red lines indicate weir associations.

Current versions of the AdCirc grid used for hurricane protection projects are based on elevation values acquired from a modified version of the FEMA LIDAR dataset combined with the best available bathymetric contours. Elevation values from these data sources are interpolated to define the elevations for points in the AdCirc grid.

3.3 Slope Stability Analysis

Slope stability analysis investigates the likelihood for earthen failures in embankments, levees, and natural slopes. Earth movements occur when the shear strength of the soil is exceeded by the stresses applied to it, including overburden weight and additional loads applied by moisture content in the soil or standing water.

Calculations for slope stability are usually reported as factor-of-safety values, which represent the ratio between the shear resistance of the soil and the shear resistance required to maintain soil equilibrium. To calculate slope stability, cross-sections of the embankment and soil boring data, including soil type classification and shear strength cohesion derived from triaxial tests, are required inputs. Methods such as the ordinary method of slices, the simplified Bishop method, and the USACE-modified Swedish method [16] apply static equilibrium equations and calculate the factor-of-safety along one or more potential slip surfaces. Slip surfaces can be modeled using wedges, circles, or non-circular shapes.

In the method of wedges, the slope is divided into an active wedge, a control block, and a passive wedge. The active wedge applies the driving force and the passive wedge contributes the resisting force. The active wedge is assumed to be inclined at an angle that is 45 degrees plus the effective friction angle of the soil [17]. The stability for each zone of unique soil type within an embankment is analyzed for its factor-of-safety at different possible slip locations along the slope. The stability of the weakest soil zone becomes the de facto stability of the embankment.

Cross-section data contributes the length and angle of the slope and is used in weight calculations for each wedge. Because boring log points may not fall close to surveyed cross-sections, it is necessary to interpolate cross-sections between existing surveyed cross-sections. A continuous surface provides a source for cross-section extraction that can be combined with boring log data for slope stability analysis.

4

Approaches to Surface Modeling

Techniques already exist for building surface models from the topographic data sources described in Chapter 2. In datasets where the point elevations are dense, such as the case with LIDAR data, a Delaunay triangulation can be applied, generating a reproducible surface that maximizes the smallest angle in each triangle. In the example of single-beam SONAR data, geostatistical analysis can be applied to the input points to determine the most likely representation of the surface [18]. Equally dense datasets such as SONAR and LIDAR are fairly easy to merge. Assuming they are based on the same vertical control information, SONAR and LIDAR can simply be concatenated together and manipulated as a single dataset. The difficulty lies in merging datasets of varying densities such as LIDAR and survey data. Once merged, these datasets can be used to provide the surface geometries required by the simulations described in Chapter 3.

4.1 Delaunay Triangulation

The first obvious approach to producing a surface might be to concatenate LIDAR coordinates with surveyed coordinates and apply a Delaunay triangulation to the combined points. The chief advantage of this approach is that every set of coordinates has exactly one Delaunay triangulation. The generated surface matches all input point elevations and creates logical intermediate values. A number of good implementations exist in commercial and publicly available software libraries such as the Triangle library [19]. Delaunay triangulation, however, tends to fail in the situation where one dataset is denser than the other. Figure 4.1 illustrates surveyed cross-section points (red triangles) and LIDAR points (green circles) in Falgout canal near Dulac, Louisiana. A Delaunay triangulation of these two datasets is shown in Figure 4.2. Details of the channel become lost between cross-sections since the LIDAR data cannot compensate for the sparse survey data.

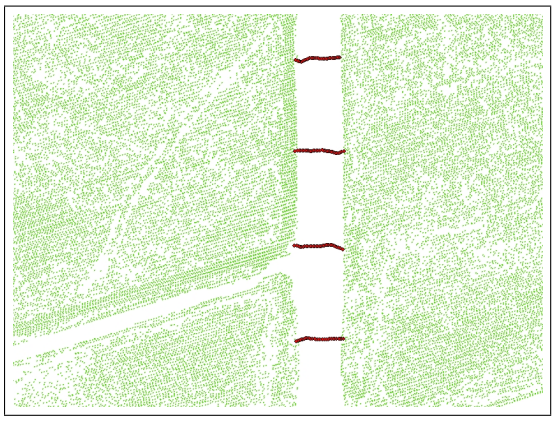

Figure 4.1: Surface input data. Green circles represent LIDAR points and red triangles represent channel cross-section points.

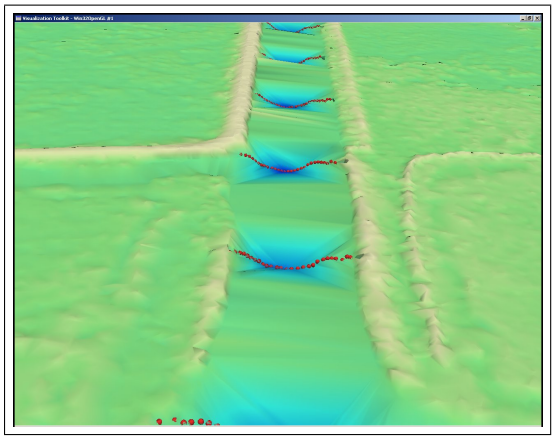

Figure 4.2: Delaunay triangulation of combined LIDAR and survey points. Red spheres represent survey cross-section points.

4.2 Geostatistics

Another option for surface modeling involves the use of geostatistical techniques such as the Inverse Distance Weighting function (IDW) or Kriging. These functions rely on the statistical properties of the available input data to fill in gaps between data points. Assuming a set of known input point coordinates, $[s_0, s_1, ...s_n] \in S$ where s is a coordinate (s_{ix}, s_{iy}, s_{iz}) and S is the two-dimensional space containing these input points, interpolation techniques can be used to derive an elevation estimator function, $\hat{Z}(s)$, that can predict the elevation value at an arbitrary point coordinate, s in space S. The structure of this estimator function is the following weighted average formula:

$$\hat{Z}(s) = \sum_{i=0}^{n} \lambda_i s_{iz}$$

In the above equation λ_i represents normalized weighting values ($\sum_{i=0}^{n} \lambda_i = 1.0$) and s_{iz} represents the elevation of known coordinate i. The challenge in interpolation lies in determining the appropriate values for the weights, $\lambda_i \in \| \ 0, n \ \|$ and, to a lesser extent, in choosing the set of known input coordinates to use in the calculation. The two most common

approaches used for interpolation, IDW and Kriging, rely on different techniques to calculate these weights.

The most practical use of the estimator $\hat{Z}(s)$ is to create elevation rasters by evaluating coordinates along a rectangular grid in the vicinity of the input coordinate points. The advantage of raster grids is twofold: 1) They can be represented by a floating-point GeoTIFF or a number of other published and widely-used file format standards. 2) Rasters can be streamed into memory, a process in which a client application loads and operates on rectangular subsets of the entire raster independently of the rest of the dataset. Streaming allows the raster to scale to the size limitations of the disk storage system. The disadvantage of raster grids is the disk space required to store a raster grid at the resolution required to model channels. Typically, the surface model area may be large relative to the channel and the modeler will not necessarily need the same level of detail outside the channel compared to inside the channel.

4.2.1 Inverse Distance Weighting (IDW)

In the case of the IDW algorithm [20], weight values are chosen by normalizing the distance of the n closest points to the coordinate at which the elevation is being calculated.

$$\lambda_i(s) = \frac{\frac{1}{d(s_i,s)^2}}{\sum_{j=1}^{n} \frac{1}{d(s_j,s)^2}}$$

In the equation above, function d is the Cartesian distance between two coordinate points. The weight evaluates to the inverse of the squared distance from the input point to a known coordinate divided by the sum of the distance of the input point to all known coordinates. A variation of the IDW algorithm uses all input point coordinates within a fixed radius instead of the n closest input point coordinates. The fixed-radius option limits predictions to coordinates that fall within a fixed distance of one or more of the input point coordinates;

All other values are left undefined.

An example of IDW is shown in Figure 4.3. Depressions form inside the channel where the cross-sections are present, but the result is not dissimilar from the Delaunay triangulation in areas where cross-sections are unavailable.

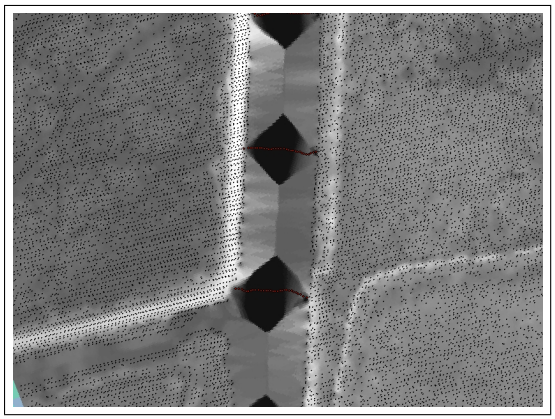

Figure 4.3: Example of surface generation using an inverse distance weighted (IDW) function with LIDAR and three cross-sections. Darker shades highlight lower elevations.

4.2.2 Kriging (Isotropic)

Like IDW, Kriging [21] techniques sum together weights multiplied with elevation values to deduce elevations at arbitrary locations. However, Kriging depends on knowledge of the correlation between every known input coordinate point.

The correlation matrix for a Kriging operation is based on a semivariogram model of the known input points. The semivariogram models the causal relationship between spatial distance from N known input points, $s_i, i \in [1, N]$ to determine the elevation of a target, unknown output point, s_0. A semivariogram calculation, γ for any two given points in a dataset is defined as follows:

$$\gamma(s_i, s_j) = \frac{1}{2}(s_{iz} - s_{j_z})^2 | i, j \in N$$

Theoretically, in the situation where there is a correlation between space and elevation, the variance in elevation decreases as the distance between two points decreases. Since points can be spaced far apart and have the same elevation, the input points are grouped together into bins based on user-defined distance, called a lag distance (ℓ), and normalized through mean or median calculations (Figure 4.4). A semivariogram plot, such as the one shown in Figure 4.5, is generated where the distance between known pairs of points (h) is plotted against the variance between points ($\gamma(h)$).

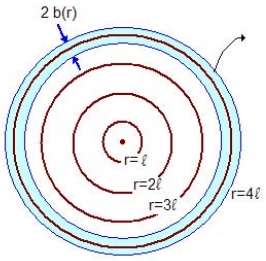

Figure 4.4: Creation of bins for Kriging. The rings show the lag distance relative to a known input point.

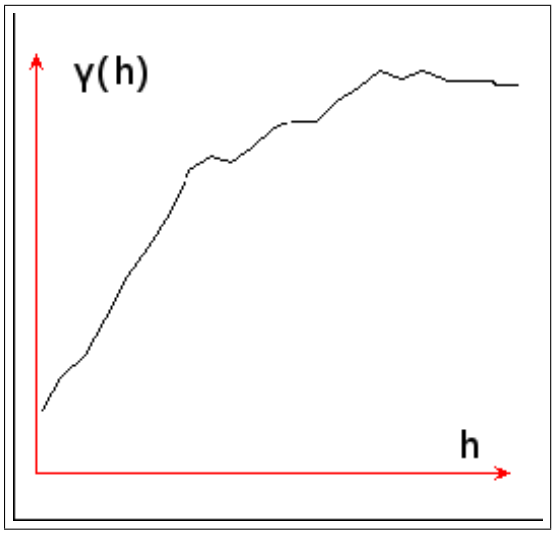

Figure 4.5: Plot of a typical empirical semivariogram. The semivariogram value (γ) increases as the distance (h) increases.

Once the empirical semivariogram values are calculated, they can be fitted to a model semivariogram based on least squares curve fitting. A model semivariogram is a simplification of the experimental semivariogram. Three commonly used model semivariograms are as follows:

Spherical Model:

$$\gamma(h) = \begin{cases} c_o + c\frac{hr}{2a} - \frac{h^3}{2a^3} \forall h \in [0, a] \\ c_o + c \forall h \in [a, \infty] \end{cases}$$

Exponential Model:

$$\gamma(h) = c_o + c(1 - e^{\frac{-h}{a}})$$

Gaussian Model:

$$\gamma(h) = c_o + c(1 - e^{\frac{-h^2}{a^2}})$$

In all three models, the nugget (c_0), sill (c), and range(a) variables represent well-defined parameters that can be adjusted to approximate the curve of the empirical semivariogram. The sill, c, represents the peak variance in the input data, which corresponds to the variance for the distance beyond which the data is uncorrelated. The distance beyond which the data becomes uncorrelated is called the range, a. The nugget, c_0, represents the maximum correlation between pairs of points as the distance between them approaches zero. Although this value is often zero, errors in data collection may produce points with the same coordinate, but different elevations. This phenomenon is called the nugget effect and is accounted for by the semivariogram model's nugget parameter. An example of a typical model semivariogram fitted to an empirical semivariogram is shown in Figure 4.6.

As shown in Figures 4.7, 4.8, and 4.9, the choice of model semivariogram affects the shape of the semivariogram curve. The exponential model, shown in Figure 4.9, approaches the

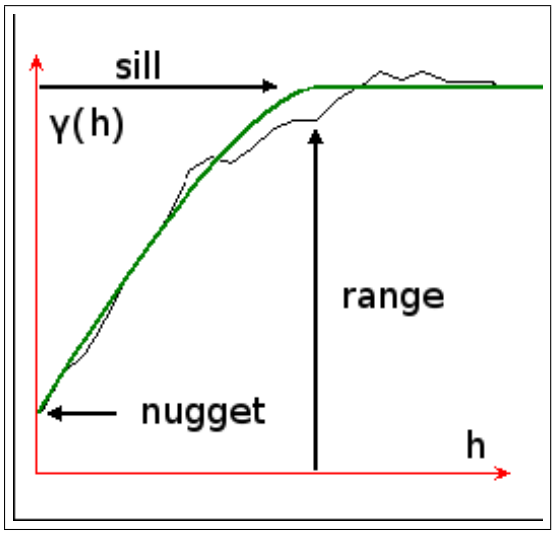

Figure 4.6: Plot of a typical empirical semivariogram (black) and its model semivariogram (green), including the annotated range, sill, and nugget.

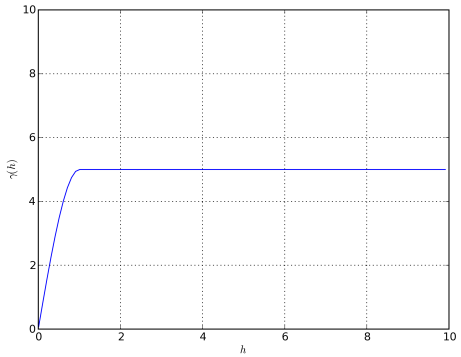

Figure 4.7: Spherical semivariogram model.

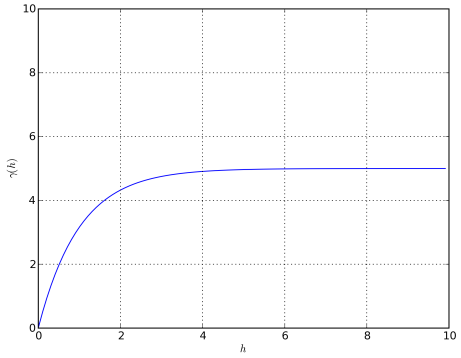

Figure 4.8: Exponential semivariogram model.

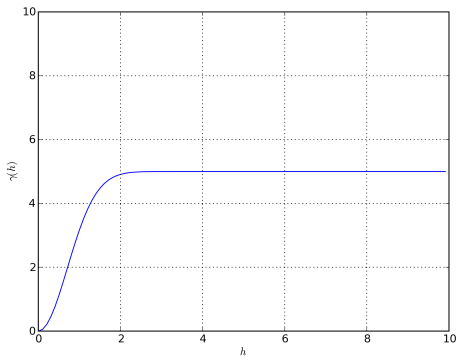

Figure 4.9: Gaussian semivariogram model.

sill asymptotically as the distance (h) approaches infinity, while the spherical model, shown in Figure 4.7, flattens at the sill as the distance approaches infinity. The Gaussian model, shown in Figure 4.9 resembles a normal curve with a small slope for low distances, then matches the asymptotic approach to the sill of the exponential model.

Values for the range, sill, and nugget can be derived in a number of different ways. They can be approximated manually by inspection of the empirical semivariogram plot or they can be approximated automatically by using a curve-fitting technique such as least squares adjustment.

The choice of theoretical model depends on the modeler's understanding of the topography. The spherical model, which is the most commonly-used model, produces an almost linear relationship between the distance and elevation correlation, while the exponential model produces a more abrupt transition. The Gaussian model retains high correlation at short distances, creating fairly flat, smooth surfaces near known points.

Once the model semivariogram parameters are established, elevations can be calculated at any location within the vicinity of the known points, S, to create a continuous elevation grid. The weight values are calculated from the semivariogram model equation using the distance from the target point to each of the the input points, $distance(s_0, s_i)$, as the input, h, for the semivariogram model. The weights for the input points are normalized such that

their sum adds up to 1.0. This is called ordinary Kriging to distinguish the method from other variations of Kriging.

The problem with Kriging is similar to the problem with IDW. Normal Kriging can not by itself handle the anisotropic nature of channel data, nor can it cope well with the potentially large spacings between known input points.

4.2.3 Kriging (Anisotropic)

In some cases, the correlation between points in a dataset is influenced as much by the angle between points as the distance between points. This angle factor is crucial to river channel modeling, where there will be lower variance between points aligned in the direction of flow, and greater variance between points aligned in the cross-section direction.

Rather than building one semivariogram, the anisotropic Kriging process produces multiple semivariograms, each for different angle ranges. The number of semivariograms is based on a user-defined angle size. For example, if the user were to choose an angle size of 30°, then 12 semivariograms would be produced, each one corresponding to a 30° wedge in which only pairs of points oriented within that angle range are included in the same semivariogram. Figure 4.10 illustrates how these semivariograms are built.

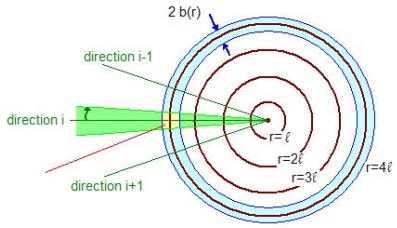

Figure 4.10: Creation of bins for the anisotropic variation of Kriging. Data points are segregated into wedges based on the angle of orientation between known input points and aggregated inside each wedge based on lag distance ranges.

Once the input points have been compiled into semivariograms such as the ones shown in Figure 4.11, they can be mapped using model semivariograms (4.12) and processed using

the same technique used in isotropic Kriging.

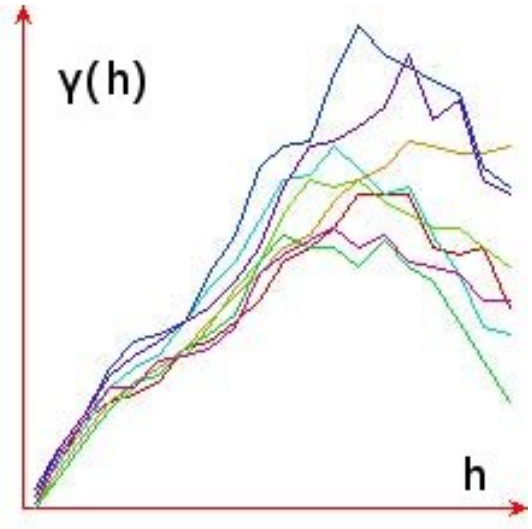

Figure 4.11: Compilation of empirical semivariograms from anisotropic Kriging. Each colored line represents a different angular wedge within the input data.

The processing time of Kriging scales poorly as more coordinates are considered and the size of the correlation matrix grows. A reasonable compromise is to use block Kriging, which considers only a limited neighborhood of input points at a time, since the contribution of points beyond a certain range has a negligible impact on the computation. This neighborhood could consist of either a fixed distance search radius or a fixed number of points search radius similar to the limiting techniques used in IDW.

4.2.4 Regularized Spline with Tension

A specialization of Kriging is the Regularized Spline with Tension (RST) [22], [23] algorithm. This algorithm generates slope curves from regular points (points with a non-zero gradient) within the input dataset. Using a smoothing function for interpolation, the RST algorithm applies the slope derivatives as a control of output surface curvature. The algorithm has weighting parameters that can control the tension, which can either conform tightly to the original dataset or allow more smoothing over the original data points.

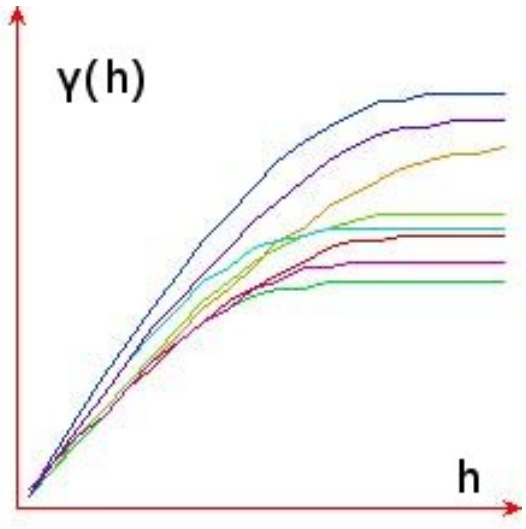

Figure 4.12: Representation of model semivariograms from anisotropic Kriging. Each colored line illustrates a different angular wedge within the input data.

The GRASS GIS software package features a prominent RST implementation that allows the user to define the tension of the generated surface. However, the tension in RST is global, and the detail of the output surface is only as good as the density of the input data. In sparse cases, the algorithm must be supplemented with additional information. For example, Cebecauer's work on surface modeling using RST found that supplemental features and contour lines were needed to produce a realistic surface in mountainous terrain [24].

4.2.5 Elliptical Interpolation of Channels

The creation of surface models in waterways presents unique challenges to geostatistical algorithms. Venkatesh Merwade's work on river channel representation [18] compared a variety of geostatistical techniques to model single beam SONAR input data. Merwade noted that modeling waterways is more complex than other surfaces due to the anisotropic properties of meandering river channels. He determined that Kriging with elliptical blocks that align to the channel boundaries was the optimal approach to use with single beam SONAR data.

Of all the techniques described in the existing body of literature, Merwade's elliptical method comes closest to matching the problem domain objective of generating topographic surfaces from sparse cross-section survey data. However, the single beam SONAR data used by Merwade provides a far greater coverage of the channel than cross-sectional data. Semivariograms calculated from traditional surveyed cross-sections display huge gaps corresponding to the dead zone between cross-sections. This makes elliptical block Kriging poorly suited for interpolating channel geometry from cross-section and profile centerline data.

5

The Hydraulic Spline Algorithm

For case studies in Louisiana channels, Delaunay triangulation and grid interpolation approaches were found to be insufficient for merging datasets of varying resolution. The cause of this deficiency is the lack of data between cross-section lines to match the density of LIDAR data points. However, in this particular problem domain, the engineer has knowledge about the path of the channel from either a surveyed data source or aerial photography. In either case, the engineer has a centerline of the channel. Based on this information, one possible solution is to interpolate the data between existing cross-sections to derive intermediate cross-sections along the centerline.

The Hydraulic Spline Algorithm [25], a generalization of the Waterway Generation Algorithm [26], exploits the interpolating property of Kochanek-Bartels splines in two dimensions to produce a two-dimensional irregular grid of quadrilaterals. This is accomplished by using separate spline templates (downstream and cross-channel) to create interpolated cross-sections, each having an equal number of points. These generated cross-sections are used to reconstruct the topology of the channel as an irregular grid. The generated irregular grid can then be merged with surrounding LIDAR data.

5.1 Splines

The spline is an algorithmic tool that can produce smooth curves and surfaces and has numerous applications in modern-day computer graphics [27]. Splines are curves defined by two or more control points. These control points map a normalized position along the curve (0.0 is one endpoint, while 1.0 is the other endpoint), α, to a data value for that α based on the spline's basis functions. Intermediate values can be generated by evaluating the curve at various values between 0.0 and 1.0.

There are two general categories of splines. Approximating splines, such as Bezier and B-splines, use a set of control points to define the shape of an output. Interpolating splines interpolate values through the actual control points. Of these types, interpolating splines are utilized by the hydraulic spline algorithm specifically for their inherent ability to preserve the input vertices within the limitations of the user-defined output granularity.

The most commonly used type of interpolating spline is the Hermite spline. Each control point in a Hermite spline is coupled with a controlling tangent vector as shown in Figure 5.1. The computation of a Hermite spline component is shown in the equation below:

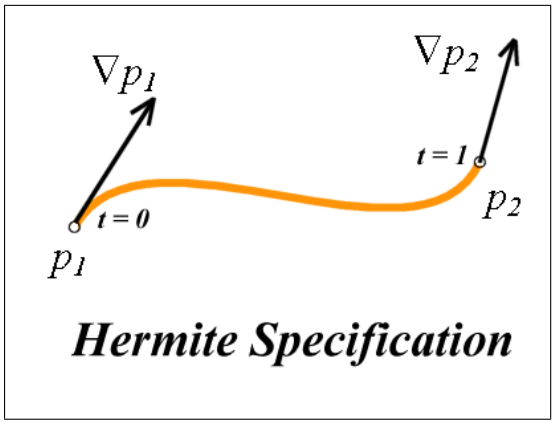

Figure 5.1: Visualization of a Hermite spline, controlling points ($p1$,$p2$), and tangents ($\bigtriangledown p1$, $\bigtriangledown p2$).

$$P(\alpha) = p_k(2\alpha^3 - 3\alpha^2 + 1) + p_{k+1}(-2\alpha^3 + 3\alpha^2) + Dp_k(\alpha^3 - 2\alpha^2 + \alpha) + Dp_{k+1}(\alpha^3 - \alpha^2)$$

$P(\alpha)$ is the component's output value, p represents the control point, and Dp represents the slope of p. Fortunately, two variations of the Hermite spline, the Cardinal spline and the Kochanek-Bartels spline [28], provide methods to manage spline curvature without calculating slope derivatives.

The Cardinal spline uses p_{k-1} and p_{k+2} control points, in addition to the p_k and p_{k+1} control points of the Hermite spline, to define a consistent curvature.

$$Dp_k = \frac{(1-t)(p_{k+1} - p_{k-1})}{2}$$

$$Dpk + 1 = \frac{(1-t)(p_{k+2} - p_k)}{2}$$

t represents a tension parameter parameter to define how tight to fit the evaluated points to the spline's control points. A value of 1 would result in linear interpretation along the spline's control points.

In contrast, the Kochanek-Bartels spline adds user-defined control parameters for the default tension, bias, and continuity attributes, while also using the same two extra control points as the Cardinal spline. In the following slope equations, b, c, and t represent bias, continuity, and tension respectively. Each of these parameters has a value in the range of [-1,1]. The bias parameter controls how far in front of or behind a control point the curvature occurs. The continuity parameter determines how smooth the change in slope is from one curve section to the next.

$$Dp_k = \frac{(1-t)[(1+b)(1-c)(p_k - p_{k-1}) + (1-b)(1+c)(p_{k+1} - p_k)]}{2}$$

$$Dp_k = \frac{(1-t)[(1+b)(1+c)(p_{k+1}-p_k)+(1-b)(1-c)(p_{k+2}-p_{k+1})]}{2}$$

In practical applications, tension is the most important variable of the hydraulic spline. Some applications, such as well-defined natural creeks and bayous, may work well with a smooth and loose spline; typical man-made drainage canals consist of trapezoid-shaped low and high flow cross-sections and may be defined by only four or seven points. Since the lines are perfectly straight in these instances, the tension should be set to 1 for man-made channel geometries. A sample Kochanek-Bartels spline is shown in Figure 5.2.

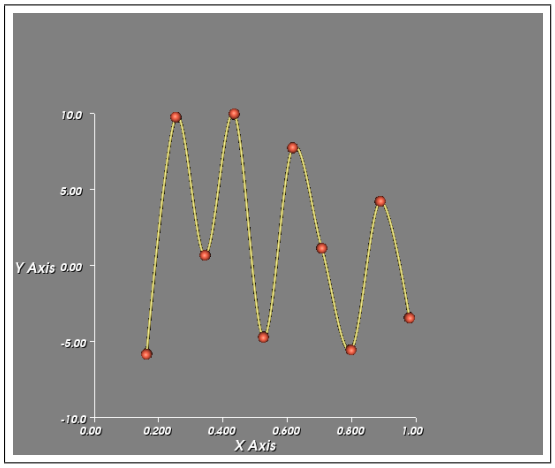

Figure 5.2: Example of a Kochanek-Bartels spline (shown in yellow) with default tension, bias, and continuity values of 0. Red circles indicate control points.

5.2 Input Cross-Section Interpolation

The first step in the hydraulic spline algorithm is to interpolate the input cross-sections, using a Kochanek-Bartels spline for the x,y, and z components of the cross-section points, assigning each cross-section the same number of vertices. The motivation is twofold: First, in the output mesh, each vertex will be connected by line segments to the corresponding vertices in the immediate upstream and downstream cross-sections. Having an equal number

of points in each cross-section makes it possible to link the quadrilaterals together as a mesh. Second, interpolating the cross-sections gives the user control over the detail level of the output mesh. Two template splines, one for each dimension of the irregular grid, have a tension parameter that is copied to all other splines, giving the user control of the output cross-section's tension.

To interpolate the cross-sections along the profile, the intersection points of cross-sections and profile centerline must be established (Figure 5.3). The intersection points are the locations where the α values are chosen and associated with their corresponding cross-sections in order to build the interpolation splines. As shown in lines 1-4 of Algorithm 1, the α value is computed as the Euclidean distance along the profile path from the first end point to the point of intersection divided by the total length of the profile. Cross-sections that do not intersect the profile at any point are not used to define the spline.

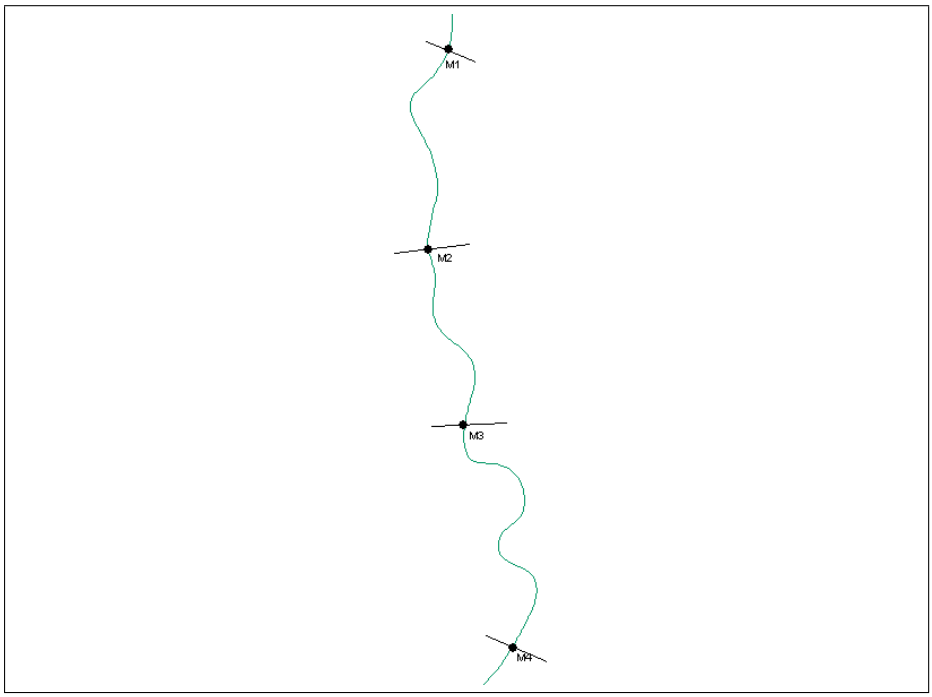

Figure 5.3: Hydraulic Spline input cross-sections and centerline. Points M1, M2, M3, and M4 represent intersection points between cross-sections and the profile centerline.

```
1  for i = 1 to size(CrossSections) do
2  |   P ← CrossSection_i ∩ Profile;
3  |   alphas(i) ← (distance from Profile end point to P)/ (total length of Profile);
4  end
5  for j = 0 to NumberOfCrossSectionSamples do
6  |   for k = 0 to size(CrossSections) do
7  |   |   M ← midpoint(CrossSection_k);
8  |   |   Compute (r, θ, φ) for CrossSection_k[j] relative to M;
9  |   |   AddPoint(splines(j), alphas(k), (r, θ, φ));
10 |   end
11 end
```

Algorithm 1: Hydraulic Spline Algorithm setup.

Once the cross-sections are either associated or filtered out, spline generation can begin. The process proceeds as described in lines 5-11 of Algorithm 1 and illustrated in a hypothetical channel in Figure 5.4. Since each sampled cross-section has an been interpolated to have an equal number of points, a separate spline can be generated for each point position in every cross-section. For instance, the first point (or leftmost point) of each sampled cross-section is added to a first spline, while the second point of each cross-section is added to a second spline, and so forth. Each cross-section uses the cross-section's midpoint as a local origin and interpolates the sampled points relative to this local origin. However, rather than relying on traditional Cartesian coordinates, the splines are based on spherical coordinates of the cross-section points as shown below in Figure 5.5. Relative to the midpoint M of a cross-section, coordinates (r, ϕ, θ) are computed for each numbered cross-section point using the following conversion equations:

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \arctan\left(\frac{y}{x}\right) - \alpha$$

$$\phi = \arccos\left(\frac{z}{r}\right)$$

Spherical coordinates offer a distinct advantage over Cartesian coordinates in waterway channel spline construction. In the Cartesian coordinate system, the X, Y, and Z (relative

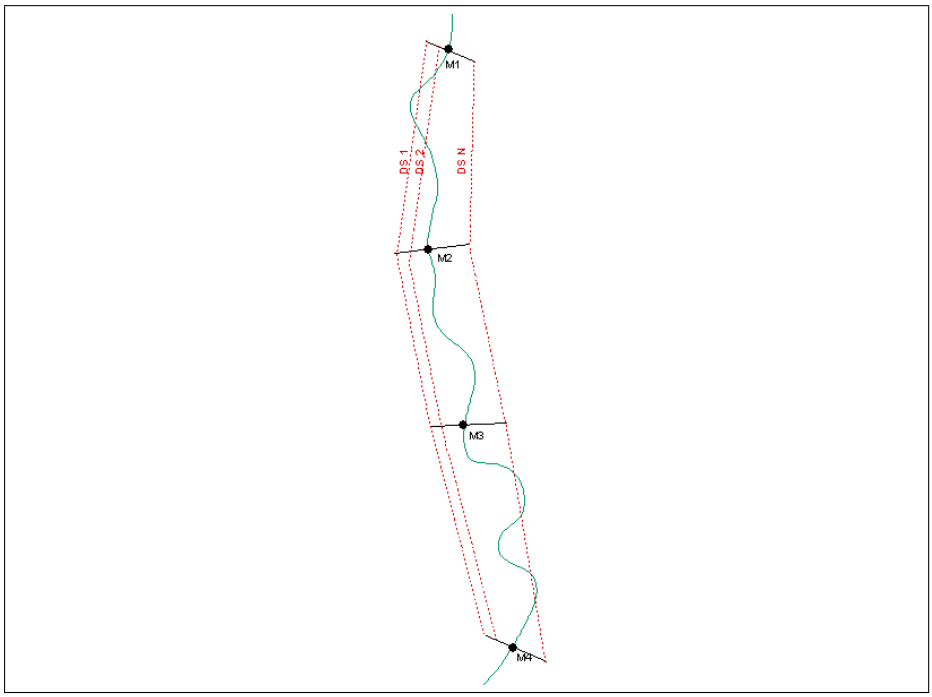

Figure 5.4: Downstream spline control point associations. Red dashed lines show associations amongst vertices. DS1 is a spline composed of the first point in each cross-section. DS2 is a spline composed of the second point in each cross-section. DSN is a spline composed of the Nth point in each cross-section.

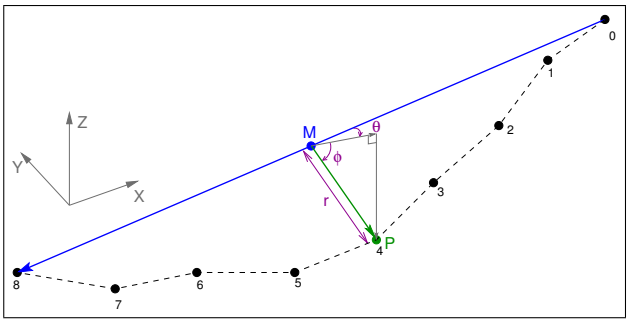

Figure 5.5: Cartesian to spherical coordinate conversion.

to the local origin) components are added to the corresponding spline, which itself is merely three one-dimensional splines. Waterways in deltaic plains have a tendency to meander, rendering Cartesian coordinates inadequate for splining cross-sections. As shown in Figure 5.6, the waterway narrows quite inappropriately around turns if no original cross-sections are located in these bends.

Since meanders may occur along the path of a waterway channel between cross-sections, it becomes necessary to interpolate the global angle of the path, adding a fourth component to the spline system. However, if the algorithm needs to track an angle, spherical coordinates can be used without requiring an additional spline component. As shown in Figure 5.6, cross-sections interpolated using spherical coordinates follow the profile without narrowing.

5.2.1 Supplementing the Profile Centerline

Aliasing can occur when splines are used to interpolate cross-sections that are spaced excessively far apart. Specifically, the natural thalweg, or deepest path along the bottom of the channel, may disappear and reappear in a different location, like the Atchafalaya River example in Figure 5.7. In such cases, auxiliary profile lines can be added to control the transitions between cross-sections. The multi-spline feature of the algorithm computes the normalized distance of the intersection point between the auxiliary profile lines and the cross-section to determine how to evaluate output cross-sections. This is done by evenly dividing the output evaluations between the breaks created by the profile centerline and auxiliary profile lines. Algorithm 2 provides an overview of the multi-spline evaluation process. The α values for the left and right endpoints of the cross-section line, which evaluate to 0.0 and 1.0 respectively, and the α values of the intersecting centerlines are split into ranges. Based on the target number of output points, evaluation is evenly split among these ranges.

The obvious profile lines to use, in addition to the thalweg, are the left and right banklines of the channel. For example, if two banklines are added as auxiliary profile lines, and

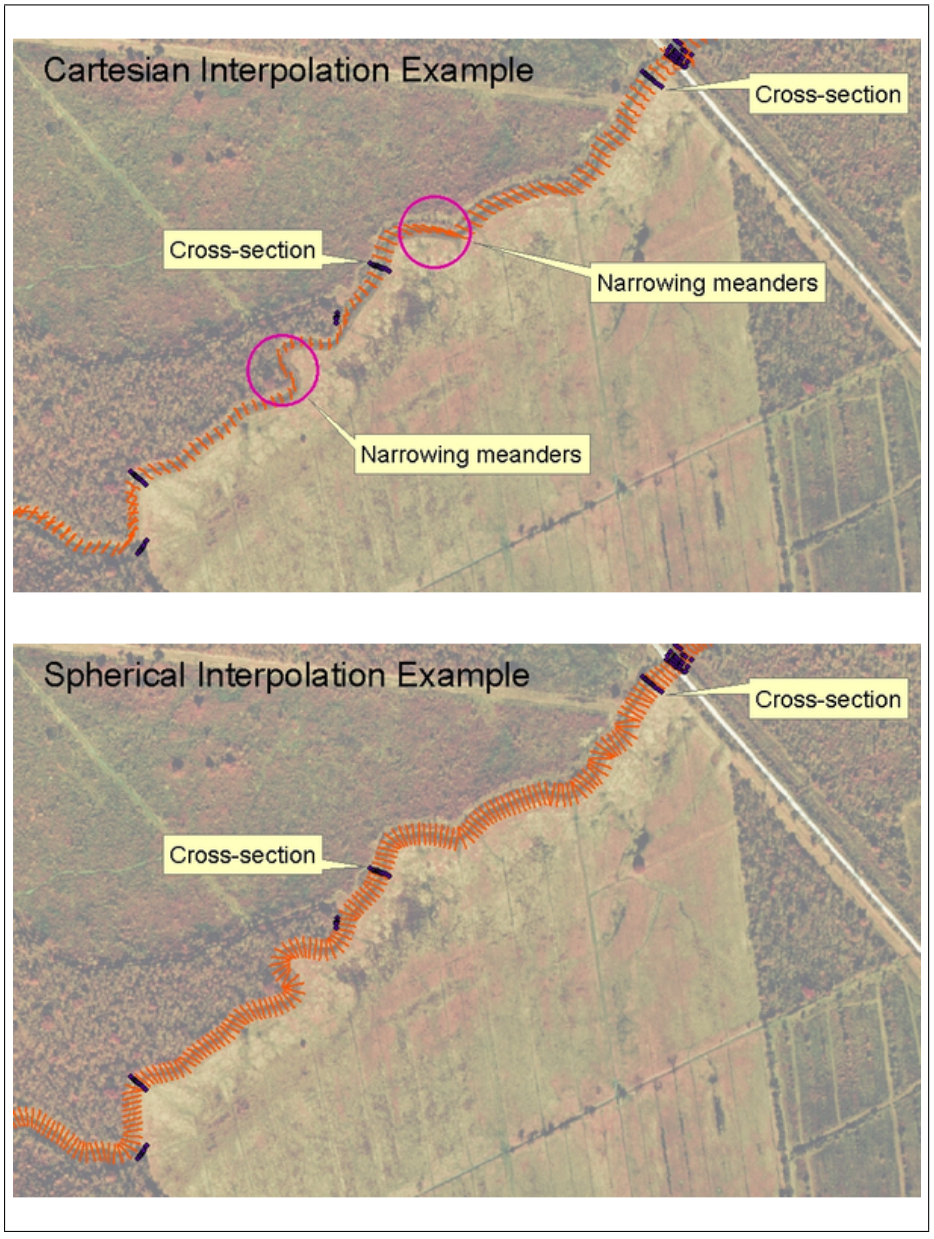

Figure 5.6: (Above) Cartesian interpolation: Interpolated cross-sections between original cross-sections narrow around the meanders. (Below) Spherical interpolation: Interpolated cross-sections preserve channel width around the same meanders.

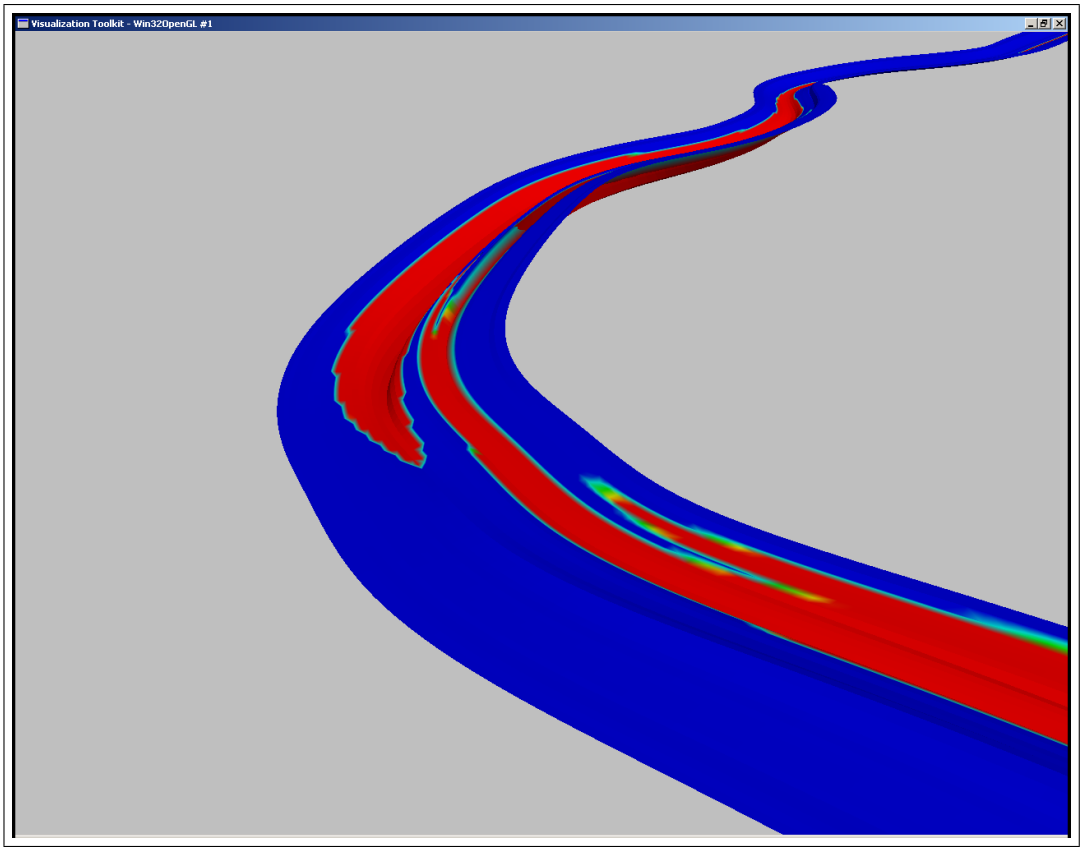

Figure 5.7: Atchafalaya River with single centerline and even cross-section point spacing. Deeper areas are shown in blue and shallower areas are shown in area. The thalweg disappears and reappears downstream.

$ResampledCenterlineAlphas \leftarrow \emptyset$;
$CenterlineAlphas \leftarrow \{0.0\} \bigcup CenterlineAlphas$;
$CenterlineAlphas \leftarrow CenterlineAlphas \bigcup \{1.0\}$;
$PointsPerBin \leftarrow NumberOfCrossSectionSamples/$`size(`$CenterlineAlphas$`)`;
for $i = 1$ *to* `size(`$CenterlineAlphas$`)` $- 1$ **do**
    for $j = 1$ *to* `size(`$PointsPerBin$`)` **do**
        $ResampledCenterlineAlphas \leftarrow CenterlineAlphas_i + \frac{(j-1)(CenterlineAlphas_i - CenterlineAlphas_{i+1})}{PointsPerBin}$;
    end
end

Algorithm 2: Hydraulic Spline Algorithm α resampling based on multiple profile lines.

each output cross-section is to contain twenty points, then five points would be on the left descending bank, five points on the left side of the channel, five points on the right side of the channel, and five points on the right descending bank. Figure 5.8 contrasts cross-section points interpolated by a constant spacing with cross-section points that were interpolated using additional profile lines. Both the shape of the cross-section and the number of evaluated points (shown as blue triangles) are the same. The only difference is that points in the multiple profile line case are grouped evenly between the control points (shown as red circles). Figure 5.9 shows the result of applying multi-spline cross-section control points: the thalweg is preserved along the channel's path.

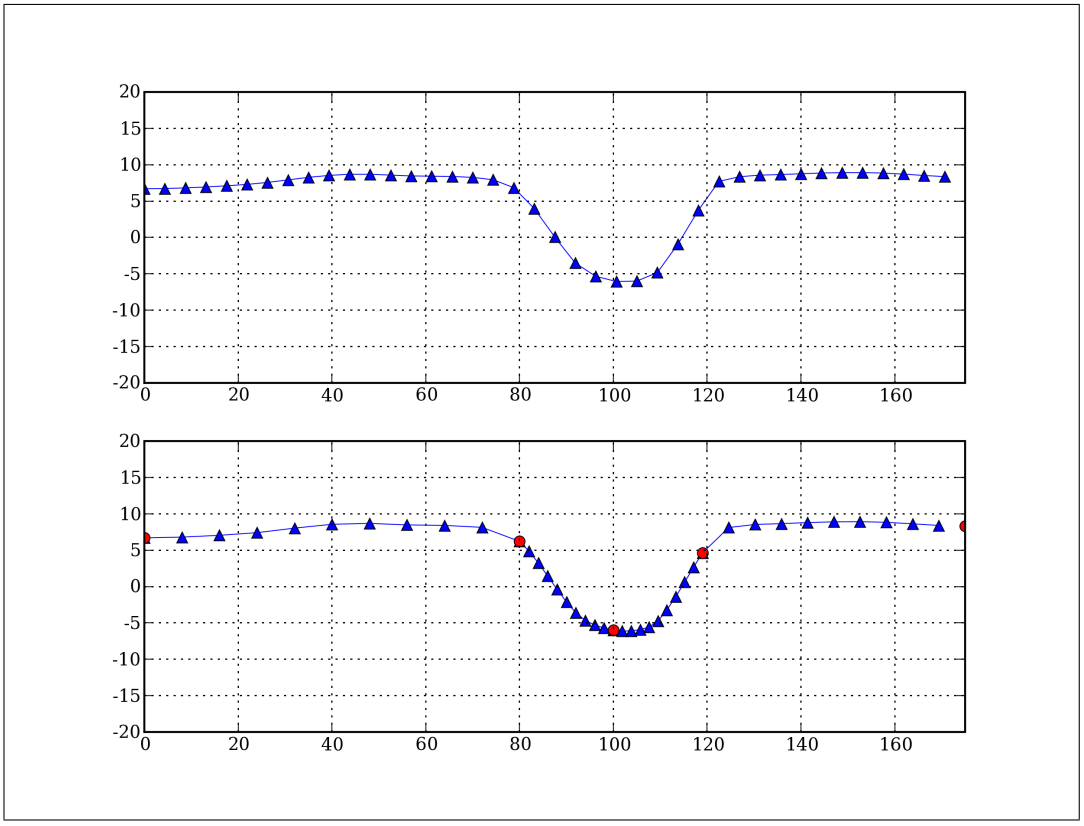

Figure 5.8: Above: typical cross-section evaluated at even-spacing intervals. Bottom: typical cross-section evaluated using additional profile lines located at red circles.

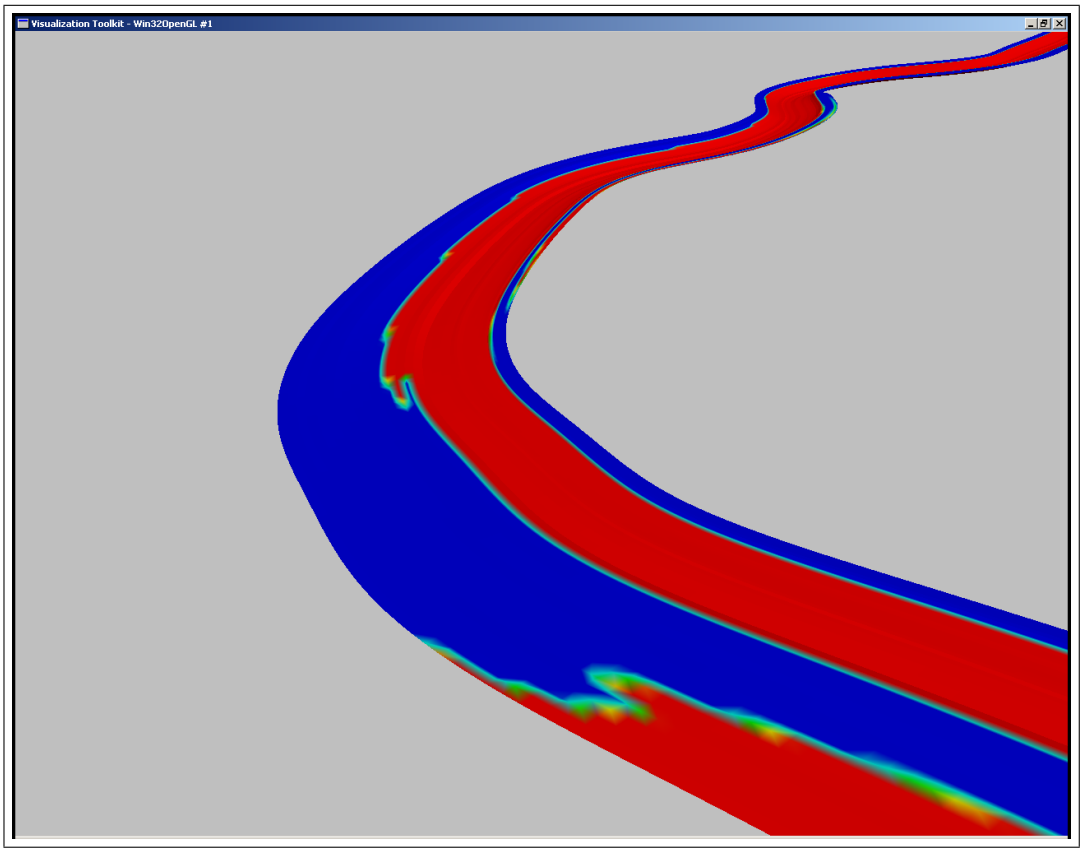

Figure 5.9: Atchafalaya River with dynamic centerline and multiple control profiles. Shallow areas are shown in red and deeper areas are shown in blue. The thalweg is preserved is this case.

```
1 for i ← 1 to NumberOfProfileSamples − 1 do
2 |   for j ← 1 to NumberOfCrossSectionSamples − 1 do
3 |   |   CreatePolygon(P_{i−1}^{j−1}, P_{i−1}^{j}, P_i^{j}, P_i^{j−1});
4 |   end
5 end
```

Algorithm 3: Evaluation process to create irregular mesh once splines are initialized. $NumberOfProfileSamples$ and $NumberOfCrossSectionSamples$ are user-defined inputs.

5.2.2 Hydraulic Spline Evaluation

Once splines are constructed from cross-section coordinates, they can be evaluated anywhere along the centerline's path. To create a surface, the spline can be evaluated at regular intervals along the centerline to produce output cross-sections as shown in Algorithm 3. The end result is M cross sections, each with N points, that follow the original waterway profile and whose shapes are defined by the original cross-sections, where M is the number of profile samples and N is the number of cross-section samples. Since its topology is regular, the resulting interpolated data can be used to build a grid as shown in Figure 5.10.

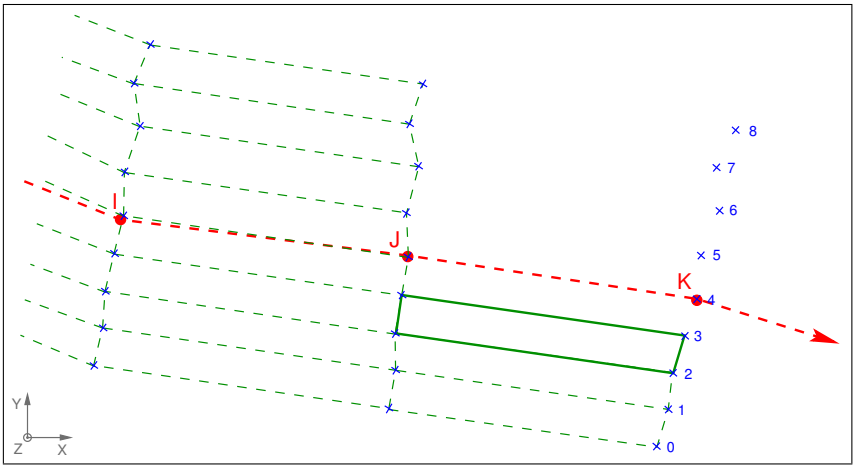

Figure 5.10: Basic mesh generation. Dashed green lines represent the part of the mesh already created, while the solid lines represent the current mesh polygon.

5.2.3 Waterway Correction

Sometimes, cross-sections surveys are taken well past the banks of the waterway and, as shown in Figure 5.11, intersect one another whenever the profile takes a sharp turn for a dense interpolation. Unfortunately, such cases can occur and must be accommodated. The only alternative to checking intersection points between cross-section lines would be to determine where the cross-section hits the bank and clip accordingly, essentially obliterating all survey information collected on land.

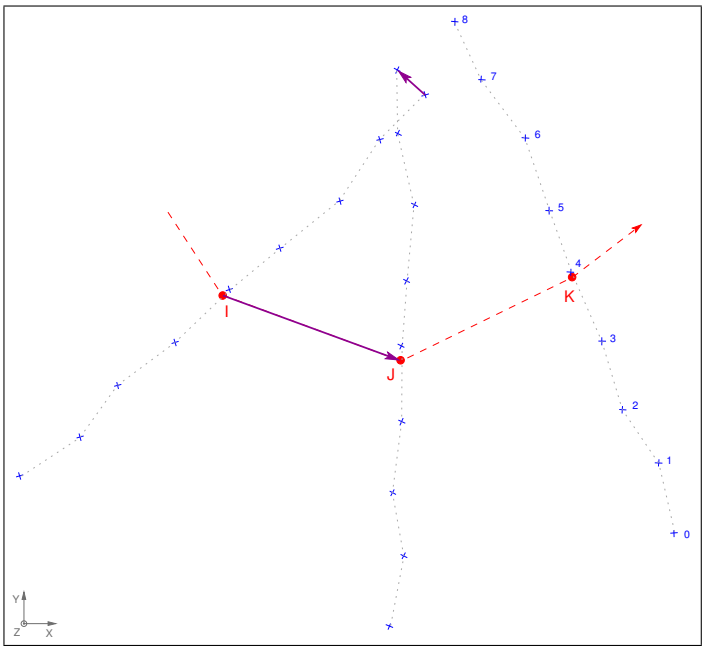

Figure 5.11: Two interpolated cross-sections intersecting each other.

A corrective step is added after the spline evaluation stage to repair potentially broken topology. The most straight-forward approach is to use the dot product calculation to determine if the vector formed by a given point in $CrossSection^{i-1}$ and its corresponding point from $CrossSection^{i}$ faces the same direction as the vector formed by the associated profile points for those two cross-sections. If $\overrightarrow{IJ} \cdot \overrightarrow{P_I^i P_J^i} < 0$, P_J^i is merged with P_I^i, producing a triangle to replace the degenerate polygon. The result is shown in Figure 5.12.

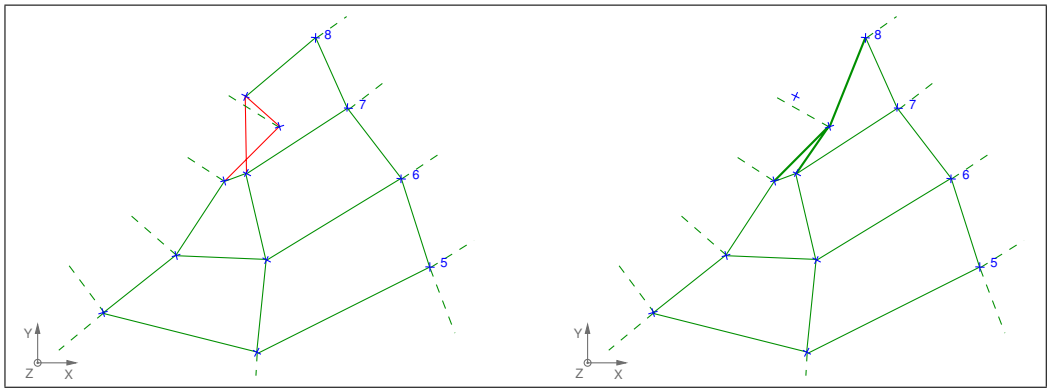

Figure 5.12: Simple Intersection Filtering.

```
1  begin
2      for i ← 1 to NumberOfProfileSamples − 2 do
3          for j ← 1 to NumberOfCrossSectionSamples − 1 do
4              if P_{i−1}^j P_i^j · Q_{i−1} Q_i < ε then
5                  P_i^j ← P_{i−1}^j;
6              else
7                  if P_i^j P_{i+1}^j · Q_i Q_{i+1} < ε and P_{i−1}^j P_{i+1}^j · Q_{i−1} Q_{i+1} > λ then
8                      P_i^j ← P_{i+1}^j;
9                  end
10             end
11         end
12         for j ← 1 to NumberOfCrossSectionSamples − 1 do
13             CreatePolygon(P_{i−1}^{j−1}, P_{i−1}^j, P_i^j, P_i^{j−1});
14         end
15     end
16     CreateLastPolygon();
17 end
```

Algorithm 4: Correction Algorithm

The filtering process corrects intersections between $CrossSection^i$ and $CrossSection^{i-1}$. To achieve a better result, filtering is also applied to $CrossSection^{i+1}$. To filter a cross-section i, the algorithm searches for intersections with the interpolated $CrossSection^{i-1}$ and $CrossSection^{i+1}$. Algorithm 4 shows the steps of this filtering process with Q_i representing the ith profile point.

Using the non-null constant ε circumvents arithmetic precision issues and merges close points together. The constant λ is necessary to check that the $CrossSection^{i+1}$ is better than the $CrossSection^i$. In the implementation, $\lambda = 0.5 \cdot \|\overrightarrow{Q_{i-1}Q_{i+1}}\|$. Although the previous algorithm works quite well in most cases, very sharp turns in a waterway can create so many

interpolated cross-sections intersections that it fails to correct them (Figure 5.13).

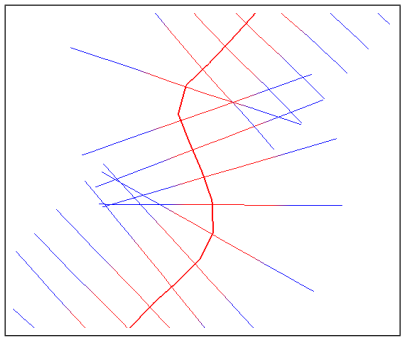

Figure 5.13: An example of multiple intersecting cross-sections.

To mitigate such problems, interpolated cross-sections must be prevented from intersecting too many subsequent cross-sections. Therefore, the algorithm uses a sorted brute-force intersection algorithm to compute all intersections between all interpolated cross-sections. Then, if an interpolated cross-section intersects more than three subsequent cross-sections, half of the cross-section's points are discarded where the crossings occur. An example of this filtering is shown in Figure 5.14.

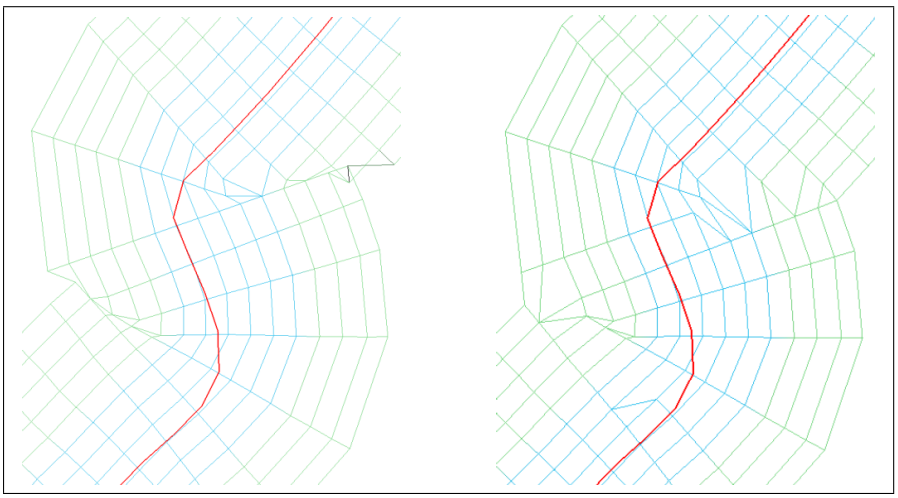

Figure 5.14: Multiple Intersection Filtering.

5.3 Merging LIDAR with Survey Data

Once a mesh is generated from the interpolated cross-sections, it can be merged with the LIDAR data in the same area. Currently, the desired effect is to use the boundary from the waterway mesh to remove points that would fall within the waterway channel and concatenate the remaining LIDAR with the points from the waterway mesh, using a constrained Delaunay triangulation. The strength of this approach draws from the assumption that survey data is inherently more accurate than LIDAR data. However even if this is true, the accuracy of this approach is limited by the density of the original cross-section survey measurements. In some cases, part of a cross-section survey is taken on land, meaning that using LIDAR inside the generated waterway might also yield acceptable results.

5.3.1 LIDAR Clipping

During the clipping phase of the process, each LIDAR point is tested to determine if it falls within any of the waterway quadrilaterals. If banklines are provided as auxiliary centerlines, then the user has the option to choose whether to omit points outside the wetted perimeter of the channel or outside the entire evaluated surface. The actual test applied to these points is the point-in-polygon algorithm in which a scan-line is taken at a fixed X value and the number of crossings are counted [29]. This approach can accommodate situations in which the polygon happens to be concave. Typically, the process of clipping points from the waterway would be the most time-intensive operation in the entire terrain generation process. Not only does this operation run in $O(mn)$ execution time (where m is the number of LIDAR points and n is the number of quadrilaterals), but LIDAR input contains in the order of up to tens of millions of point coordinates and generated waterways may contain up to hundreds of thousands of quadrilaterals.

A number of spatial indexing structures can be found in literature for improving the ef-

ficiency of the clipping operation, such as Quadtrees and R-Trees. For this implementation, an in-memory implementation of an R-Tree [30] is used to index the waterway quadrilaterals to facilitate searching. The R-Tree organizes spatial data into a balanced hierarchical structure similar to a B-Tree used in database implementation. Each level has up to a fixed number of nodes represented with minimum rectangular bounding-boxes computed from the extent of the node's children. The leaf nodes are the bounding boxes of actual polygons in the generated waterway dataset. Once the R-Tree is constructed, queries can be performed in $\theta(\log N)$ time by traversing through the tree and pruning cases where the current test point falls outside a node's bounding box, thus reducing the aggregate computation time of the clipping operation to $\theta(M \log N)$.

An example of an R-Tree is shown above in Figure 5.15 and Figure 5.16. In this particular example, the R-Tree has three levels. (A, B) represent the first level. (a, b, c, d) represent the second level. Rectangles 1-8 represent bounding boxes of individual elements in a dataset, comprising the R-Tree's third layer. A search query that intersects with top-level rectangle A, but not top-level rectangle B, would traverse the R-Tree to the second level and query a and b, which fall inside of A, but not c or d.

The advantage that the R-Tree offers for clipping channel data is the ability to balance itself and provide consistent node query access time regardless of how the data is distributed in the spatial domain. In contrast, Quadtrees perform most efficiently when given an even spatial distribution. Meandering river channels are typically long and narrow, filling only a portion of their minimum bounding rectangle with actual channel data.

5.4 Constrained Triangulation

Surface topology construction is performed by combining the generated waterway surface and the LIDAR points to form a constrained Delaunay triangulation using the generated

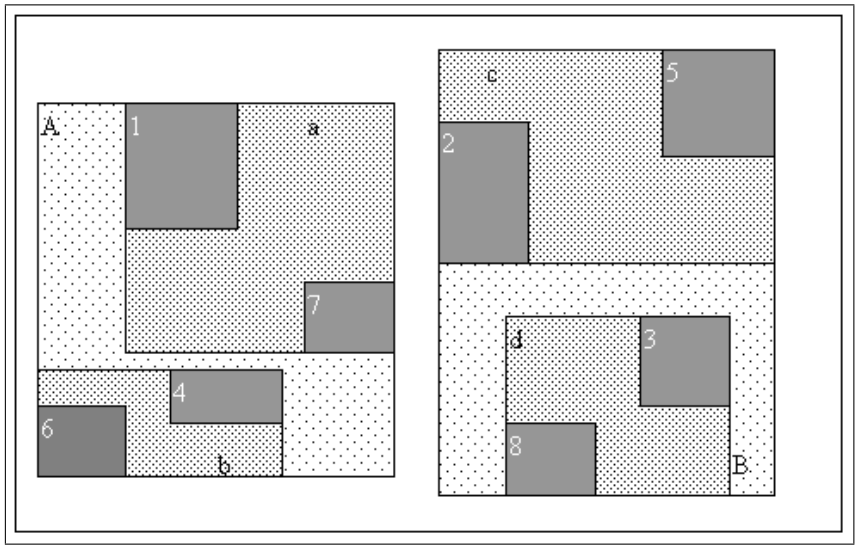

Figure 5.15: R-Tree example. Boxes 1-8 represent the minimum bounding rectangles of individual geometries.

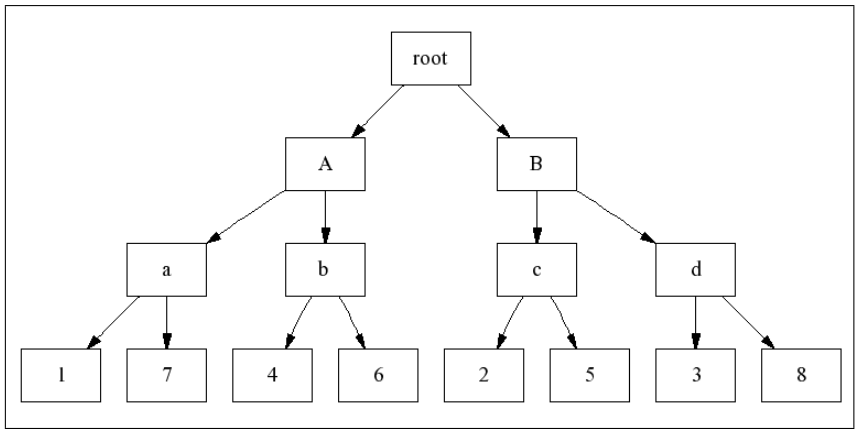

Figure 5.16: R-Tree representation of a typical dataset. Each node has an arrow pointing to its children. Leaf nodes (nodes 1-8) have no children and represent the minimum bounding rectangles of individual geometries.

waterway surface topology as the constraints. The constraints play a crucial role in determining final topology; the waterway constraints prevent LIDAR points along the bank from infiltrating the channel geometry, which would produce unnatural high points inside the channel. An initial two-dimensional Delaunay triangulation is created from the combined dataset. Next, the triangles that intersect a waterway surface line segment are discarded and replaced with a retriangulation on either side of the given line segment. An example of the resulting topology from a constrained Delaunay triangulation merged with an interpolated channel is shown in Figure 5.17.

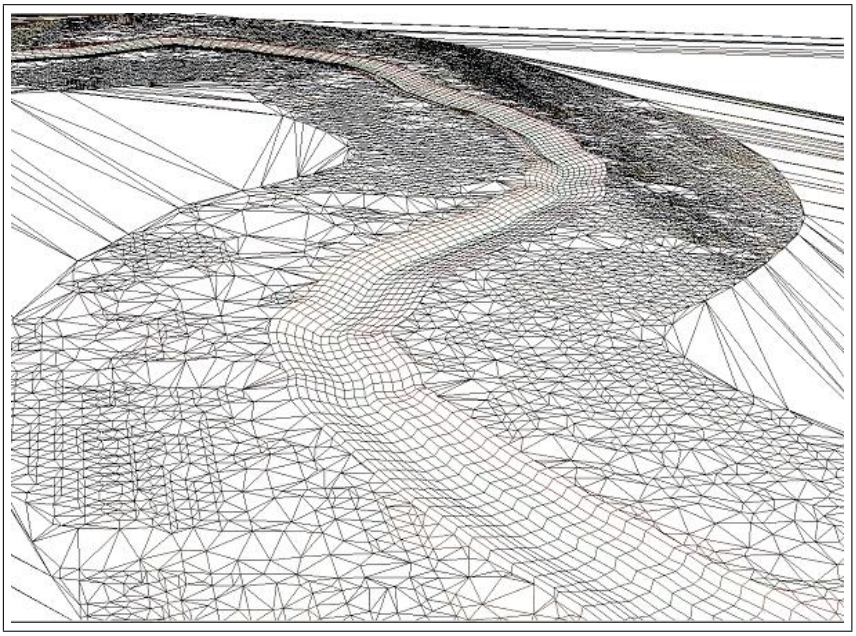

Figure 5.17: Close-up of merged hydraulic spline output and adjacent overbank LIDAR in Black Bayou. LIDAR triangulation is constrained to areas outside the spline-interpolated channel.

6

Channel Confluence Modeling

While individual channels can be modeled using the hydraulic spline interpolation described in Section 5.2, the majority of hydraulic modeling projects will require geometry for a full network of channels. A channel does not exist in isolation. Rather, it is a component of a larger network of streams and rivers formed by soil erosion and sedimentation over time. A channel may have multiple tributaries and distributaries that an engineer may want to model, depending on the required level of detail for a project. The locations where tributaries and distributaries meet the main channel are called confluence areas. If channels are modeled separately, the banks of one channel will cut off the path of related channels in confluence areas. Several approaches were prototyped for modeling confluence areas: anisotropic Kriging yielded the most promising results for modeling confluence area hydrography.

6.1 Hydraulic Spline Network Approach

Merging the output meshes for channels modeled with the hydraulic spline may not produce the expected result, since the bank of one channel will cut off the other channel if insufficient

cross-section data exists in the confluence area. Figure 6.1 shows an example of both a tributary and a distributary in the Bayou Des Allemands basin. The channel forks into two channels and combines back into a single channel further downstream at a junction point called a confluence. However, the hydraulic spline algorithm can be modified to model channel confluence by aggregating individual channels before the merge step. First, the centerlines are divided into reaches based on centerline junction points as shown in Figure 6.2. Next, a breadth first search (BFS) algorithm [31] is applied to trace the shortest route from all terminating segment nodes to all other terminating segment nodes. Associations between segments are recorded in a network (Figures 6.3 and 6.4). Unique centerlines are generated for each combination of segments that form a distinct path. Each of these paths is used as the centerline for an individual hydraulic spline processing step. The output channel surfaces are merged together using Delaunay triangulation.

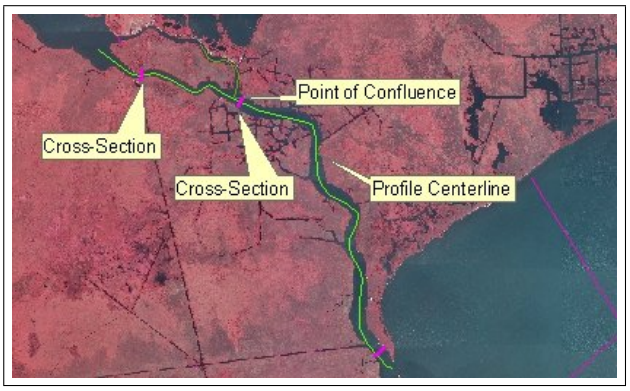

Figure 6.1: A simple confluence in Bayou Des Allemands. The river splits into two channels in the northwest, which merge back together downstream of the split.

Even with the networked approach, the resulting output surface may not meet accuracy requirements because of the abrupt change created by a confluence in the channel geometry. Abrupt changes contradict the focus of the hydraulic spline, which is to create geometry where cross-sections exist and to generate a smooth surface everywhere else within the channel. An example output surface is shown in Figure 6.5. While one canal clearly transitions

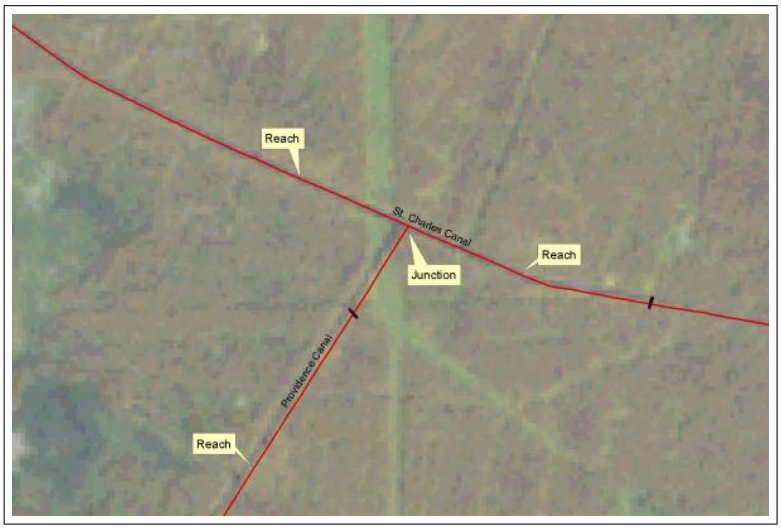

Figure 6.2: Map view of a confluence areas showing reaches and junctions.

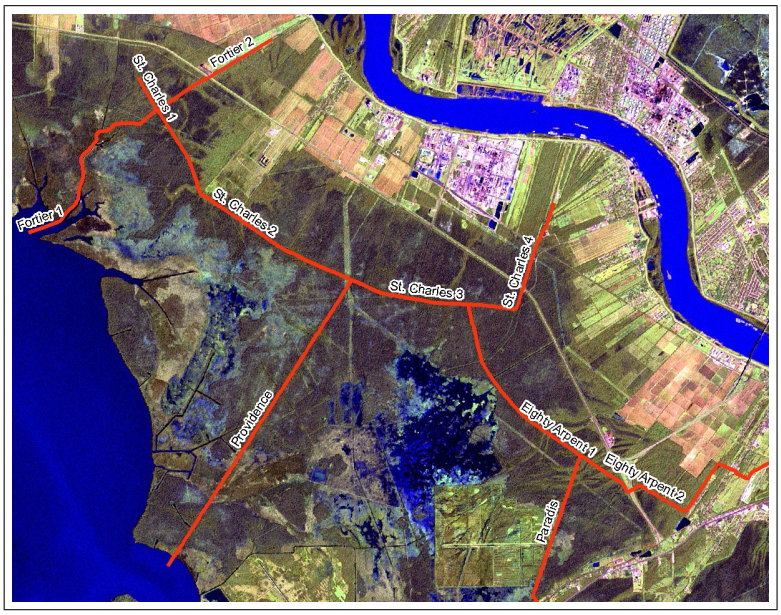

Figure 6.3: A channel segment network for a fairly complex of set canals and bayous.

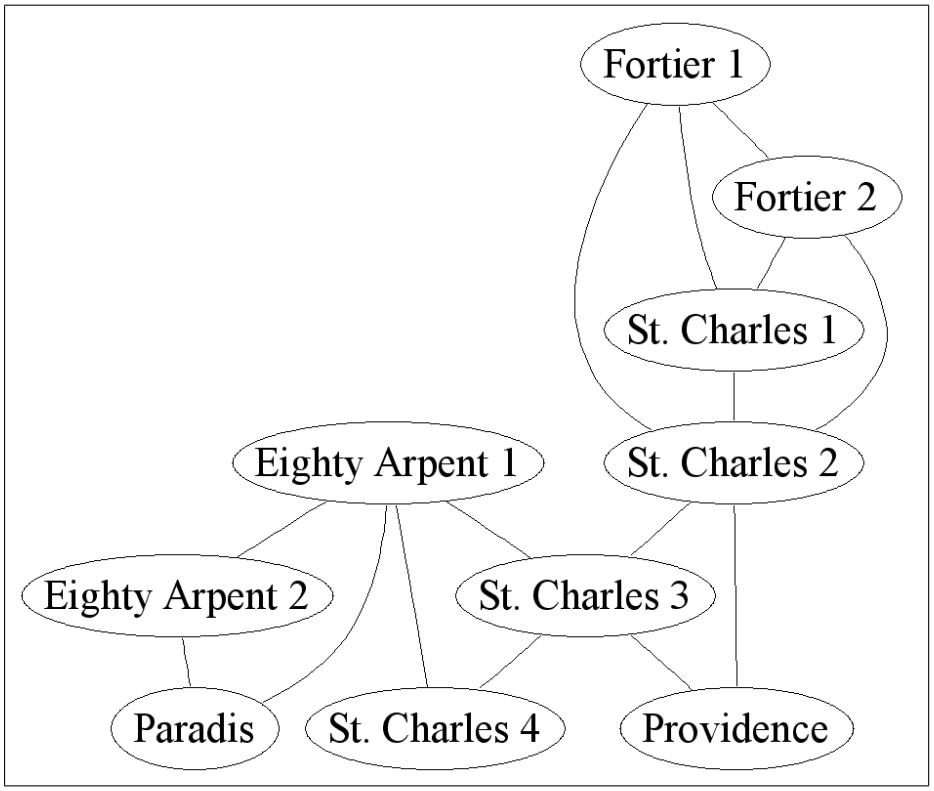

Figure 6.4: Graph representation of the channel reach network shown in the map in Figure 6.3.

into the other canal, small spikes can be seen near the corners of the junction.

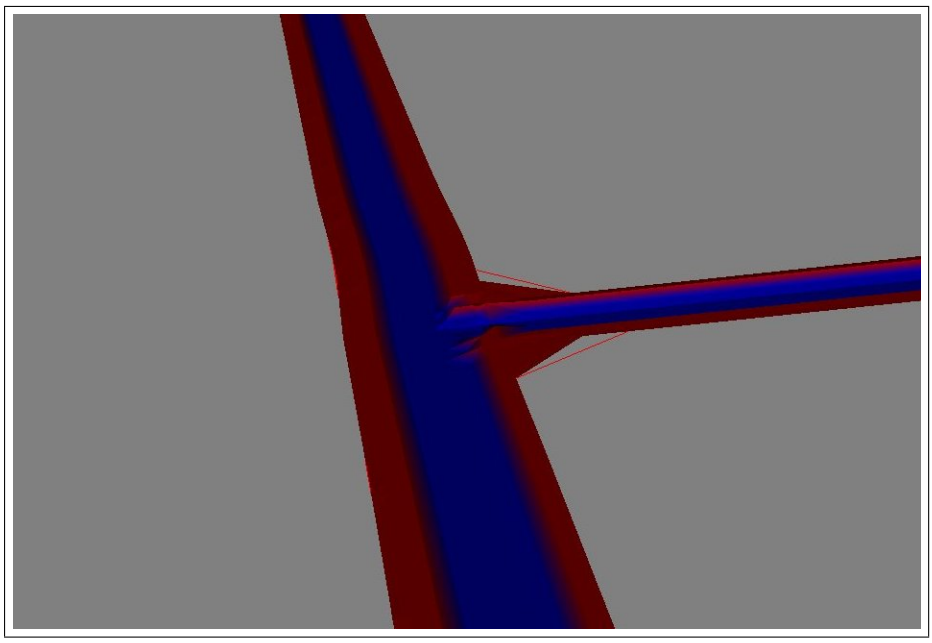

Figure 6.5: Rendering of a canal confluence in Vacherie. Data spikes appear along the bank when confluence cross-sections are not available.

6.2 Anisotropic Kriging Approach

As an alternative approach, anisotropic Kriging can also be used to model channel confluence areas. Kriging relies on manual processing steps and user judgment, but can produce more accurate results if executed properly. Rather than dividing the channel system into reaches based on profile centerlines, the Kriging approach divides the channel system into continuous areas with similar directions of anisotropy. Once these areas are determined, the input cross-section points are clipped and processed separately using anisotropic Kriging as described in section 4.2.3. Elevation grids are generated as Kriging output for each area and merged back together into a continuous surface. This process is illustrated in Figure 6.6.

The key to a good surface model is to successfully merge output surface grids. Concatenating grids together can produce obvious discontinuities as shown in Figure 6.7. Fortunately,

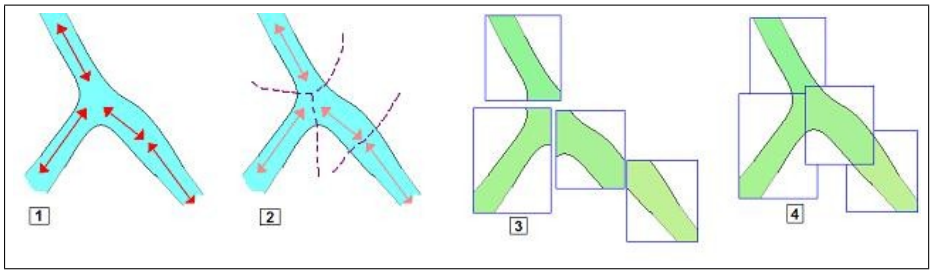

Figure 6.6: Overview of Kriging-based confluence modeling. First, the main directions of anisotropy are determined(1). Areas are clipped into pieces for Kriging(2). Grid surfaces are generated (3) and merged back together(4).

raster grids allow for simple analysis of overlap areas since the user can control the output resolution and the pixel alignment. For each coordinate location represented in the surface model, there may exist zero, one or more pixels that overlap. To determine the elevation for overlapping pixels, a weighting function can be employed in a fashion similar to the IDW process. To determine the merged pixel elevation, $\hat{p}$, from N partial elevation grids, P, the weighting function would be as follows:

$$\hat{p} = \frac{\sum_{i=0}^{N} p_{i_z} w_i(p_i, P_i)}{\sum_{i=0}^{N} w_i(p_i, P_i)}$$

In the above function, p_{i_z} represents an elevation from one of the partial elevation grids, at the same coordinate as $\hat{p}$. w represents a weighted average function that is calculated based on the distance of $\hat{p}$ from the center of the grid, based on the extent of the world coordinates for the partial grid (P_{XMin},P_{XMax},P_{YMin},P_{YMax}):

$$w(p, P) = |w_x(p, P) w_y(p, P)|$$

where

$$w_x(p,P) = \begin{cases} \frac{p_x - P_{XMin}}{P_{XMax} - P_{XMin}} \forall P_{XMin} <= p_x <= P_{XMax} \\ 0 \forall P_{XMin} > p_x \bigcup p_x > P_{XMax} \end{cases}$$

and

$$w_y(p,P) = \begin{cases} \frac{p_y - P_{YMin}}{P_{YMax} - P_{XYin}} \forall P_{YMin} <= p_x <= P_{YMax} \\ 0 \forall P_{YMin} > p_y \bigcup p_y > P_{XYax} \end{cases}$$

Figure 6.7: Surface model produced by merging multiple anisotropic Kriging outputs. Black represents deeper areas. A discontinuity appears in the middle of the channel.

The difficulty of blending is rooted in the choice of the overlap range. If the overlap is too small, the grid will contain discontinuities. If the overlap is excessive, the grid may contain artifacts from one grid that should not appear in the overlapped grid. However, the choice of overlap size can be made easier by examining the frequencies of the individual grids and creating bandpass mosaics [32].

The multi-step process of Kriging and merging the resulting surfaces can remove the spiking effect in the confluence area. Figure 6.8 shows the Tiger Pass confluence in Venice, Louisiana (near the Mississippi River Delta). In this area, the visualization shows no spiking or discontinuity in the middle of the confluence area. However, the channel may require post-processing as Kriging can not exactly determine the edge of the channel based on the input data. The green lines represent bank areas that were generated and illustrate a false hydrography. The confluence Kriging process would need to be supplemented with a polygon of the water's edge in order to clip out the valid channel area.

A more detailed example showing the Mississippi River Delta is shown in Figure 6.9. The white lines indicate input cross-sections. Not only does the visualization of the output surface looks fairly realistic, but the input cross-section surface elevations match surface elevations within the output grid.

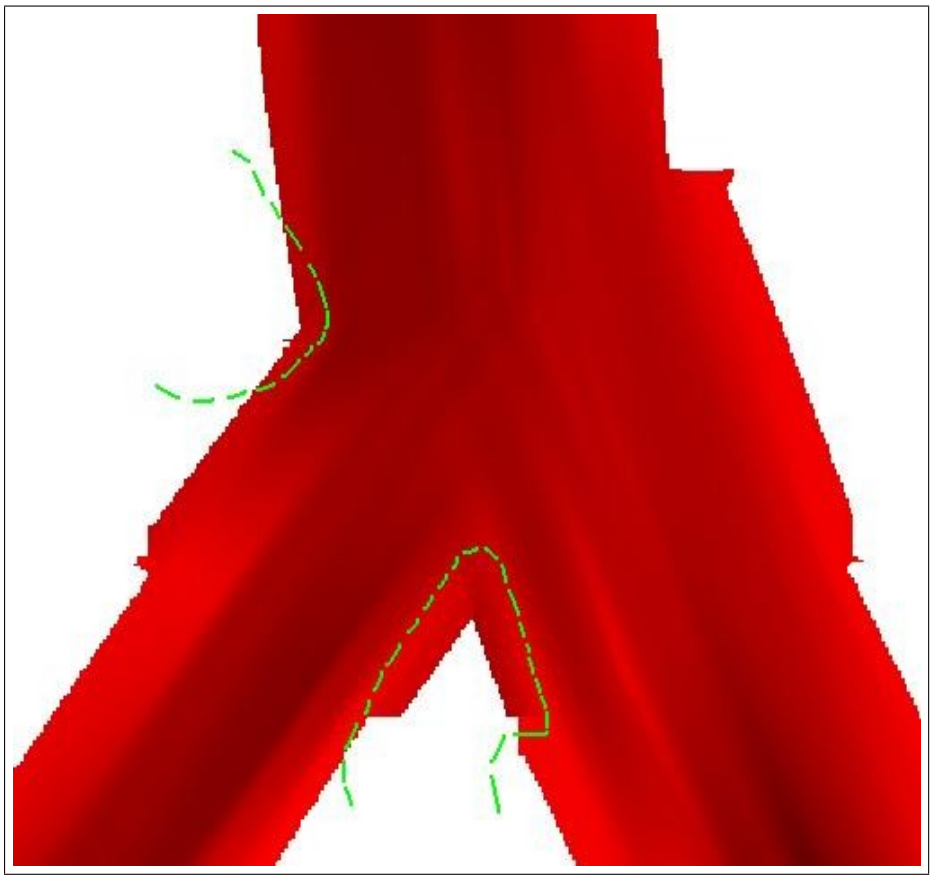

Figure 6.8: Sample confluence modeled with anisotropic Kriging. Green lines represent the calculated water's edge. Surface elevations on the landside of these edge lines are invalid, since all input was hydrographic.

Figure 6.9: Confluence at the Mississippi River Delta. Input cross-sections are shown in white.

7

Results

Numerous tests were conducted using channels from ongoing projects. Two particularly interesting cases are where the number of input cross-sections is very sparse and where the number of input cross-sections is very dense. Bayou Becnel, taken from the Donaldsonville to the Gulf of Mexico project, exemplifies the case with sparse input. The Mississippi River Hydrobook (or more formally, the Mississippi River Navigation and Hydrographic Survey Book) [33] serves as an example of a channel with dense cross-section data. These experiments validate the following goals for the hydraulic spline algorithm:

1. The algorithm must be able to reproduce the original cross-section at the α value where that cross-section intersects the path of the centerline.

2. The generated cross-sections must form a smooth and continuous progression from one original cross-section to the next.

7.1 Implementation

The primary platform for the development of this dissertation and its result testing is the Visualization Toolkit (VTK) [34]. Not only does VTK provide tools to create a desirable visual representation through OpenGL, but it also offers a component-based processing architecture, in which each operation, such as triangulation or decimation, is implemented as a single class. Multiple classes can be instantiated and chained together in a lazy-evaluation pipeline. Additional components can easily be implemented and added to these processing chains. VTK includes built-in support for several different types of splines including the Kochanek-Bartels spline used for this algorithm, as well as a geometric data abstraction class called vtkPolyData, and a probe algorithm for data sampling used during accuracy testing.

Three extension libraries were written specifically as part of this dissertation. The first extension library is the VTK Geography library, a family of classes to support the import and export of various geospatial file formats, such as shapefiles, Microstation CAD files, PostGIS, and ArcSDE, as well as support for coordinate system conversions. The library also includes the in-memory R-Tree used to perform clipping operations for LIDAR and channel merge operations. The second extension library is the VTK Hydraulic library, which includes a class for the hydraulic spline, and classes that encapsulate the hydraulic spline to generate cross-sections and irregular grids. Additional utilities were added to support bankline creation and survey file formats. The third extension library is the VTK Kriging library used for channel confluence modeling and surface blending.

Additional software was used for testing and mapping. Python and matplotlib were used to create the XY plots from the outputs of the probe algorithm. ArcGIS was used to create many of the figures that include locational reference maps.

7.2 Sparse Case: Bayou Becnel

The intended application area for the hydraulic spline algorithm is the case where there are only a few cross-sections available. Bayou Becnel, shown in Figure 7.1, has a meandering centerline and cross-sections spaced approximately one mile apart, making this channel an excellent example for the sparse case. Figure 7.2 shows a comparison of Delaunay triangulation and the hydraulic spline. Clearly, Delaunay triangulation is incapable of creating a surface from such little information; it does not take into consideration the centerline like the hydraulic spline approach does. Consequently, triangulation is unable to render the channel in the way a user would expect.

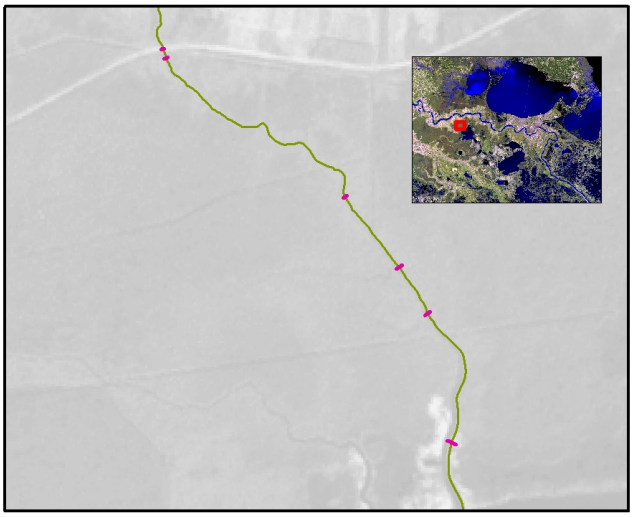

Figure 7.1: Map of area covering Bayou Becnel.

7.3 Dense Case: Mississippi River Hydrobook

Although intended for sparse cross-sections, the hydraulic spline algorithm should also handle dense cross-sections and render a surface similar to a Delaunay triangulation. Cross-sections

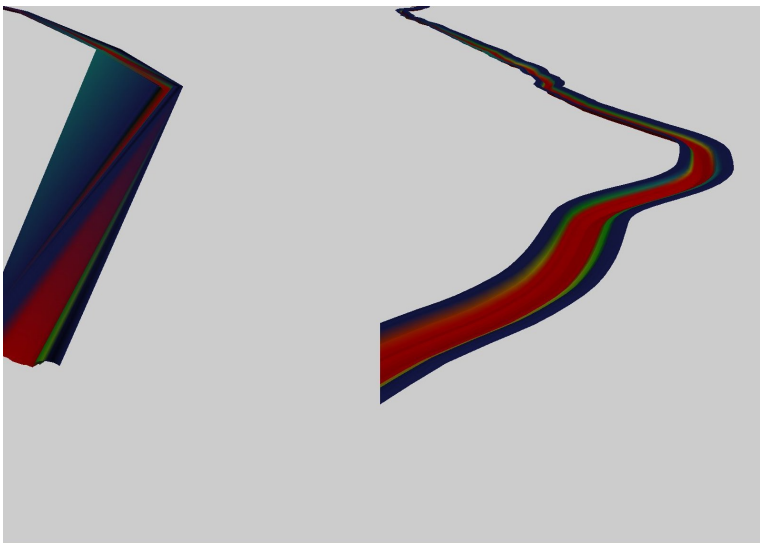

Figure 7.2: A comparison of Delaunay triangulation (left) and hydraulic spline processing (right) in Bayou Becnel.

in the Mississippi River Hydrobook, spaced at a relatively close interval of one thousand feet as shown in Figure 7.3, provide a good example of dense cross-section data. Figure 7.3 shows a small section of the Mississippi River near English Turn. The Mississippi River example, with its large number of input cross-sections, should ensure that the algorithm can scale to the most intensive real world examples.

The first requirement of the algorithm is that it must reproduce the input cross-sections at each of the α values where the input is defined within the confines of the requested number of output points. Figure 7.4 shows three cross-sections from the input dataset and the corresponding cross-sections generated by the hydraulic spline using 50 points in the generated cross-section. The algorithm reproduces the input cross-sections.

The next requirement for the algorithm is that it must provide a smooth transition from one cross-section to the next. Figure 7.5 shows the transition from one cross-section to the next using a loose default tension for the Kochanek-Bartels spline parameters. There is a noticeable progression from one cross-section to the next.

The processing time for the hydraulic spline algorithm was tested against VTK's Delau-

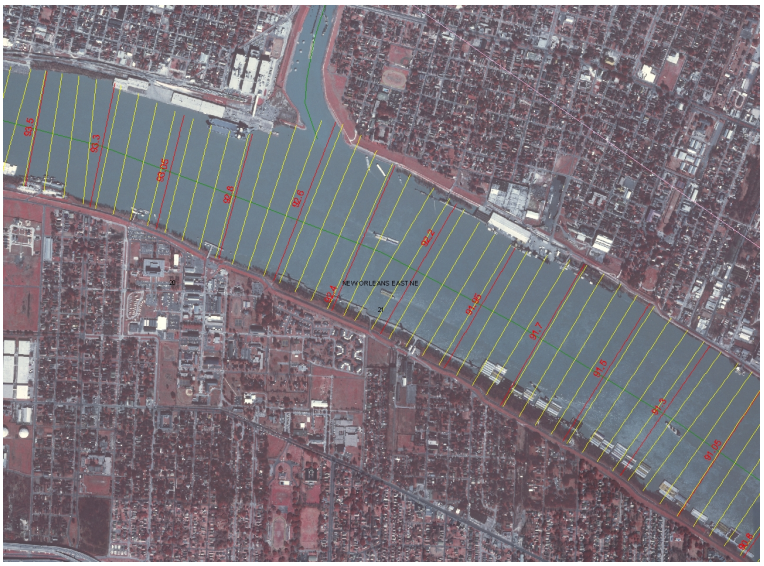

Figure 7.3: Section of Mississippi River Hydrobook data. Original surveys are shown in red. Interpolated cross-sections are shown in yellow.

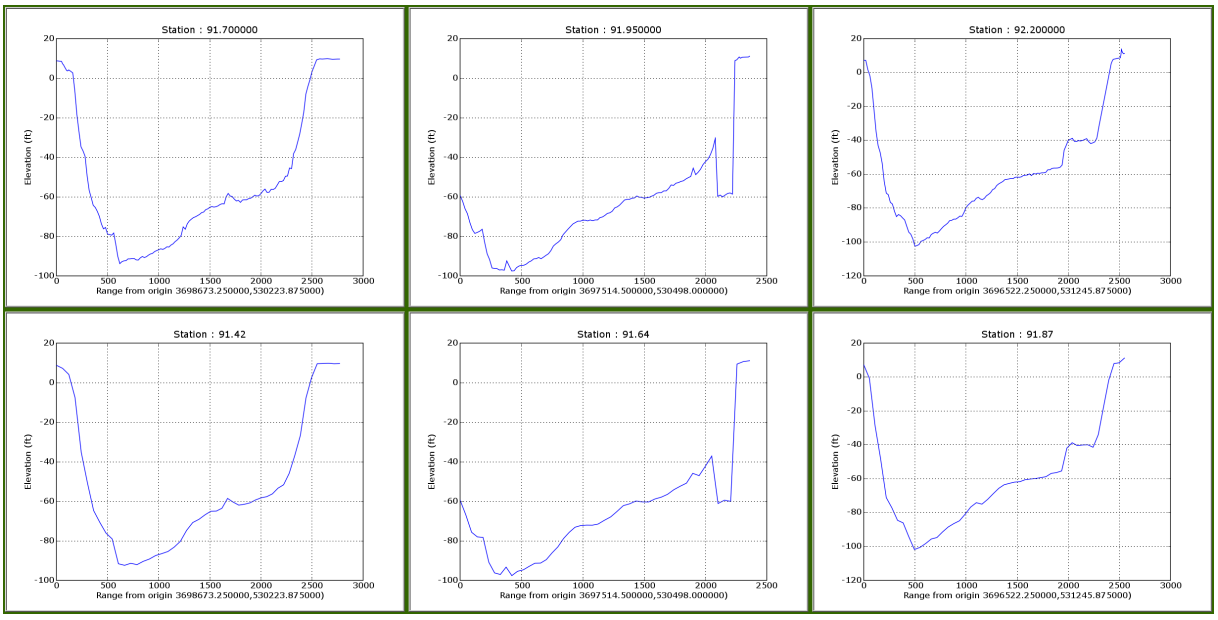

Figure 7.4: (Above) Cross-sections from the original Mississippi River Hydrobook. (Below) Interpolated Cross-sections evaluated at the α positions of the original cross-sections.

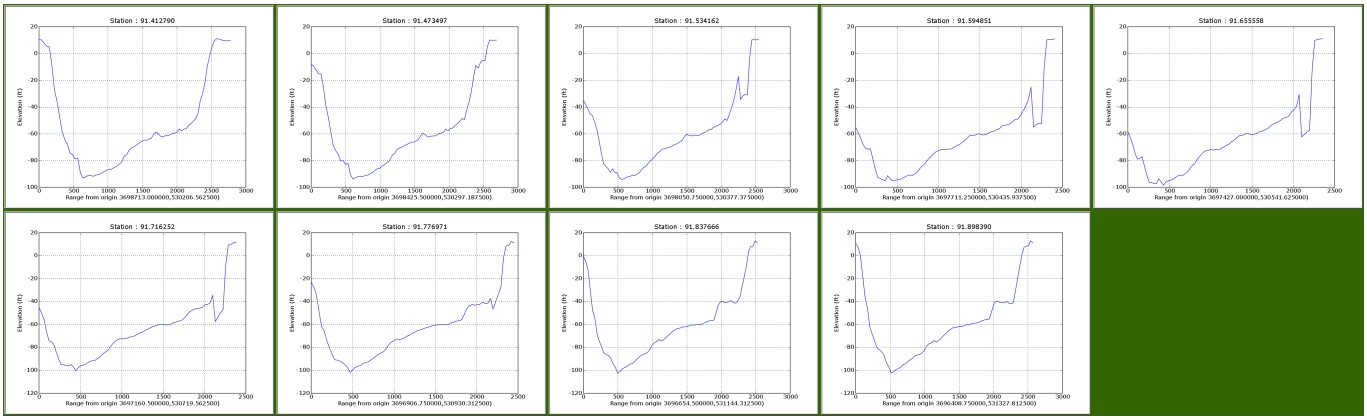

Figure 7.5: Interpolated cross-sections generated from the Mississippi River Hydrobook showing a progression at 5 foot intervals.

Table 7.1: Comparison of execution time for hydraulic spline and Delaunay triangulation.

	Points	Cells	Execution Time (seconds)
Hydraulic Spline	233,300	230,868	17.70
VTK Delaunay Triangulation	228,197	271,382	163.31

nay triangulation for the Mississippi River Hydrobook dataset. Table 7.1 shows test results in which the hydraulic spline algorithm can generate more geometry in less time.

Hydraulic splines attempt to solve the specific problem of channel generation, rather than the more general problem of surface generation, which gives the method a distinct advantage with respect to performance to other techniques. Much of the calculation effort is pushed to the spline setup phase. Evaluation of the spline is a linear time operation with respect to the quantity of requested output cells. Triangulation computation time is dependent on the number of input points and IDW computation time is dependent on search radius size. In terms of space, the hydraulic spline algorithm is just as limited as triangulation, in that the memory used is proportional to the size of the input dataset. In contrast, the space used in IDW processing is constant when the output is a raster, since data can be streamed as part of the processing.

As an added bonus, as shown in Figures 7.6, 7.7, and 7.8, the hydraulic spline algorithm produces a surface more aesthetically pleasing than a triangulated surface, but without the

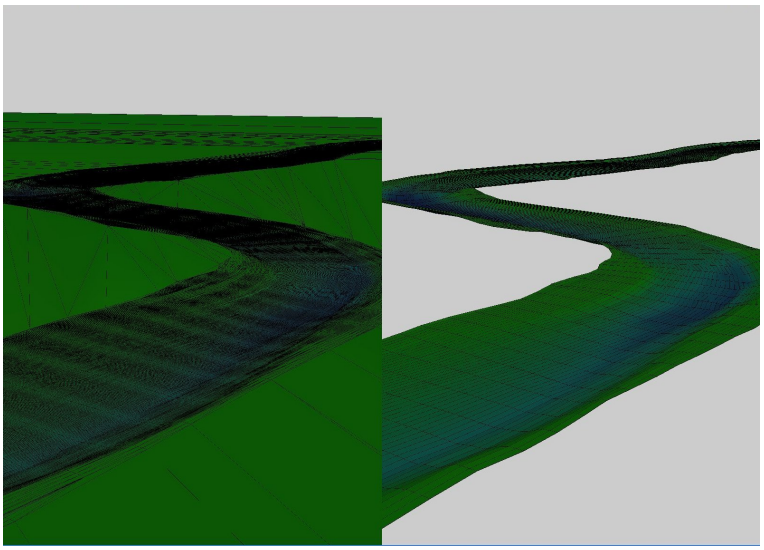

Figure 7.6: A comparison of Delaunay triangulation (left) with a hydraulic spline surface (right) for the Mississippi River.

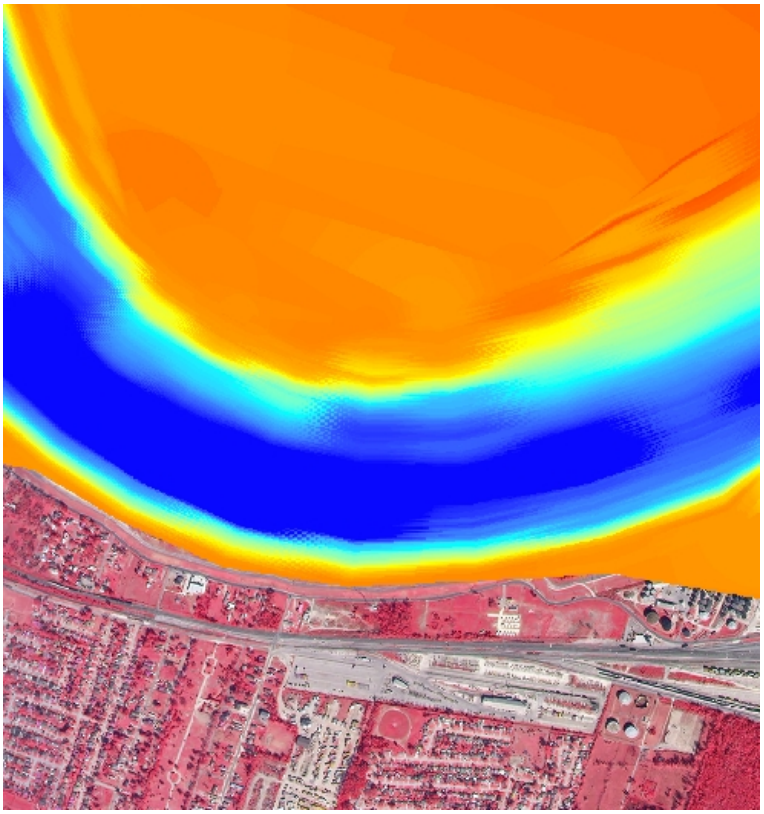

Figure 7.7: A rasterized triangulation visualization of Mississippi River cross-sections. Long jagged triangles are visible in transition areas and a false surface appears on the north bank since the triangulation has no constraints.

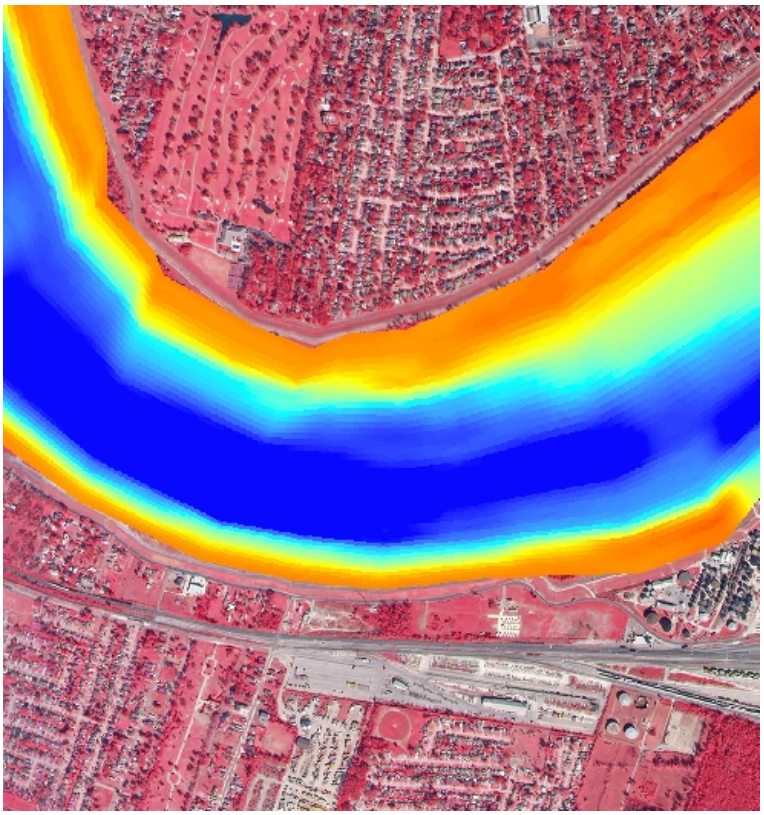

Figure 7.8: A rasterized visualization of a Mississippi River surface generated using the hydraulic spline. Transitions are smoother than in the triangulated surface and banks are defined with external constraints.

superfluous boundary triangles created in Delaunay triangulation. These large boundary triangles are a serious problem in practical application. Typically, a user would have to set a maximum triangle area or specify a circumcircle tolerance to filter out extraneous triangles. Specifying a tolerance that is too high creates holes in the data. Specifying a tolerance that is too low fails to filter out spurious triangles. Developing a filtering tolerance can involve a significant amount of trial and error. In the end, the user may have to manually remove extraneous triangles. In contrast, the hydraulic spline-generated surface does not have to be pruned and can be easily merged with LIDAR data as shown in Figure 7.9.

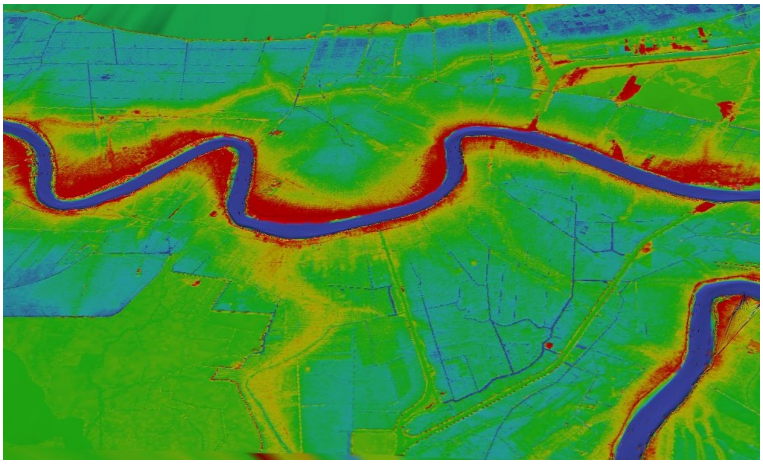

Figure 7.9: LIDAR data for the New Orleans metropolitan area merged with a hydraulic spline interpolation of the Mississippi River.

7.4 Accuracy

Several tests were run on the output surface generated by the hydraulic spline algorithm and compared with original input data to calculate absolute accuracy measurements. Absolute accuracy is a measurement of the difference between the elevations of the known input points and the elevations of the evaluated surface at the same locations. To evaluate elevations at known locations, a process known as elevation probing in the Visualization Toolkit (VTK) is employed. Elevation probing interpolates elevations from the vertices of a polygon containing

the sample point based on a weighted summation using the distance between the sample point and each polygon vertex. Using the input point elevations (s_{iz}) and the probed point elevation ($\hat{S}(s_i)$), the root mean squared error (RMSE) formula (shown below) can be used to calculate absolute accuracies.

$$RMSE(\hat{S}) = \sqrt{\frac{\sum_{i=1}^{n} \left((s_{iz}) - \hat{S}(s_i)\right)^2}{n}}$$

The dataset used for testing is a subset of the Mississippi River Hydrobook cross-sections in St. Charles, Jefferson, and Orleans parishes. The input cross-sections consist of 130 line features spaced approximately 1000 feet apart with each feature containing an average of 135 vertices. Elevations range from -199 feet to 35 feet, with a mean elevation -56.1 feet and a standard deviation of 45.9 feet. To feed the hydraulic spline runs, a profile centerline was digitized (without elevation values) from 1 meter resolution color-infrared imagery.

7.4.1 Quantified Spline Accuracy

Hydraulic spline output was tested to determine if it met the fundamental requirement that the elevations of an output cross-section must match the elevations of the input cross-sections at the locations where input cross-sections are defined. This test was implemented by first calculating the α of the intersection of an input cross-section section with the profile centerline. Next, the hydraulic spline was evaluated at that α value and a cross-section was generated. Points were sampled from both input and output cross-sections and the RMSE was calculated for each sample. Multiple tests were run with varying numbers of output cross-sections points. As demonstrated in Figure 7.10, the RMSE value converges on 0.0 as the number of output cross-section points increases.

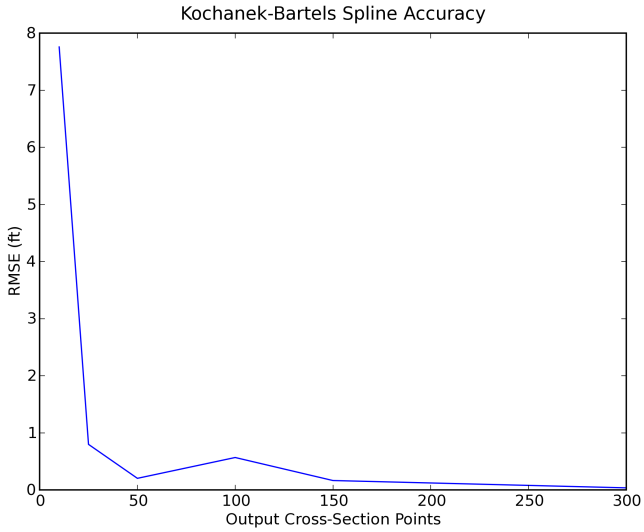

Figure 7.10: RMSE calculations for a single output cross-section evaluated at the furthest upstream intersection point. RMSE between output cross-section and original cross-section approaches 0 as the number of interpolated cross-section points increases to equal the number of points in the original cross-section (approximately 135 points).

7.4.2 Delaunay Triangulation and IDW vs. Hydraulic Spline at Varied Resolutions

Absolute accuracy of the hydraulic spline output was tested against the Delaunay triangulation technique, commonly used in practice for hydrographic modeling, as well as against the IDW algorithm. To simulate sparse data, six input cross-section datasets were generated: the first input dataset included cross-sections every thousand feet (the original subset of Mississippi River Hydrobook cross-sections shown in Figure 7.11), the second input dataset (Figure 7.12) consisted of cross-sections every 2000 feet (every other cross-section from the original dataset was dropped), the third (Figure 7.13), fourth (Figure 7.14), fifth (Figure 7.15), and sixth (Figure 7.16) datasets consisted of cross-sections spaced 4000, 8000, 16000, and 32000 feet apart respectively, each created by dropping every other cross-section from the previous dataset.

Once the test case cross-section datasets were assembled, output surfaces were created using Delaunay triangulation, IDW, and hydraulic spline. For the IDW algorithm, the output surface was a 5 foot resolution raster grid, while the Delaunay triangulation and hydraulic spline outputs were vector datasets. For the hydraulic spline surface, multiple runs were conducted employing several variations of the number of output cross-sections and the number of output cross-sections points. The resulting output surfaces were probed with points from the original cross-section dataset (1000 foot spacing) and RMSE values were calculated.

Figure 7.11: Input cross-sections from the Mississippi River Hydrographic survey of 2004 shown in red. Cross-sections are spaced approximately 1,000 feet apart.

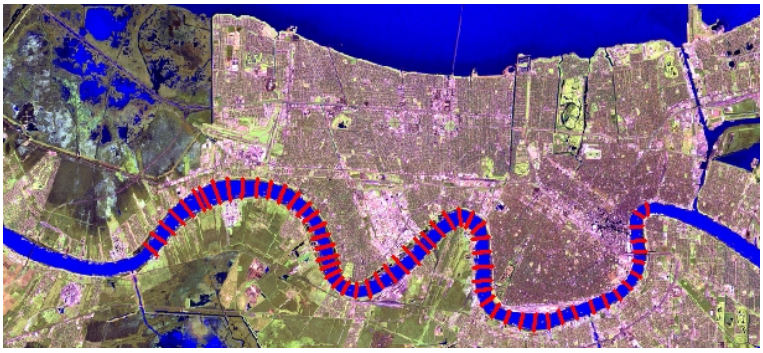

Figure 7.12: Subset of input cross-sections from the Mississippi River Hydrographic survey of 2004 shown in red. Cross-sections are spaced approximately 2,000 feet apart.

The absolute accuracy of triangulation is shown in Figure 7.17. Absolute accuracy is plotted against output cross-section spacing for each of the six test case datasets. For many

Figure 7.13: Subset of input cross-sections from the Mississippi River Hydrographic survey of 2004 shown in red. Cross-sections are spaced approximately 4,000 feet apart.

Figure 7.14: Subset of input cross-sections from the Mississippi River Hydrographic survey of 2004 shown in red. Cross-sections are spaced approximately 8,000 feet apart.

Figure 7.15: Subset of input cross-sections from the Mississippi River Hydrographic survey of 2004 shown in red. Cross-sections are spaced approximately 16,000 feet apart.

Figure 7.16: Subset of input cross-sections from the Mississippi River Hydrographic survey of 2004 shown in red. Cross-sections are spaced approximately 32,000 feet apart.

of the test case datasets the cross-section spacing limits the number of points that can be sampled. Delaunay triangulation will only create a surface from known points, and parts of the channel can fall outside the set of known points. This effect can be seen in Figure 7.18 where the Delaunay triangulation misses most of the river banks. In fairness, points from the original cross-sections that do not fall within the output Delaunay triangulation are excluded from accuracy calculations. The percentage of original points included in the triangulation outputs for the six triangulation runs is shown in Figure 7.19.

The absolute accuracy of IDW is calculated for each of the six test case datasets. As indicated by the results in Figure 7.20, the accuracy measured against the original cross-sections is not as precise due to the fact that the elevations were probed from a 5 foot resolution grid, rather than the exact inputs points as used by Delaunay triangulation. The RMSE increases dramatically and starts to level off.

Figures 7.21, 7.22, 7.23, 7.24, 7.25, and 7.26 compare the RMSE of sampled points from triangulation and IDW to the RMSE of sampled points generated by the hydraulic spline. One figure is included for each test case dataset of cross-sections. Since the hydraulic spline algorithm parameterizes the number of input cross-section points to use and the number of output cross-section lines to generate, multiple output surfaces can be generated. This contrasts to Delaunay triangulation where every input point is used and a single predetermined

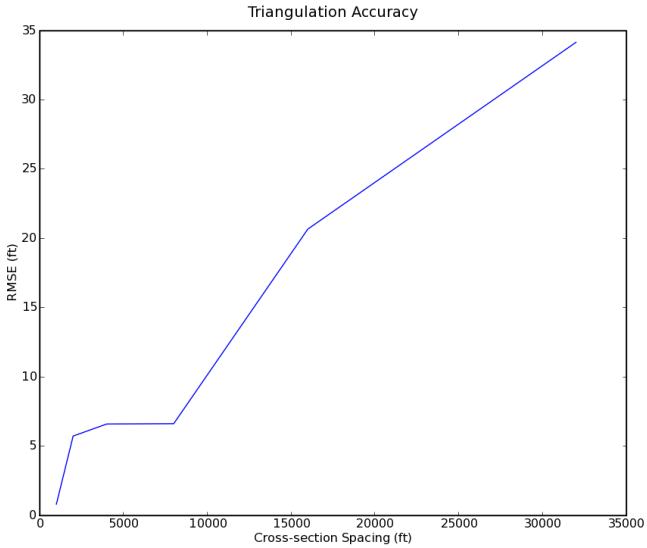

Figure 7.17: Plot of probed triangulation accuracy versus cross-section spacing of test case datasets. Error is zero when all cross-sections are used and increases linearly as cross-section spacing increases.

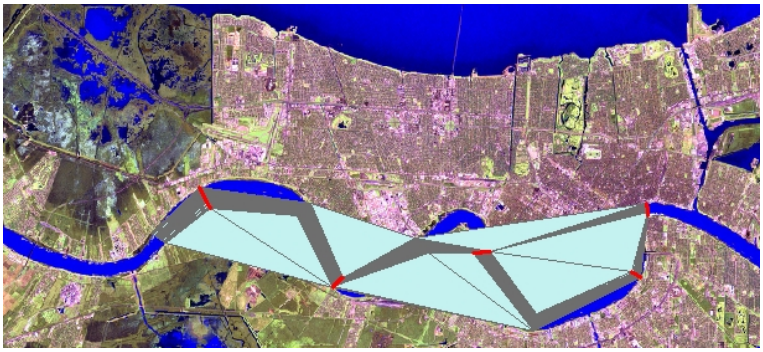

Figure 7.18: Delaunay triangulation of test case cross-sections (spaced 16,000 feet apart). Edges are shown in gray. Some of the river bends are clearly missed.

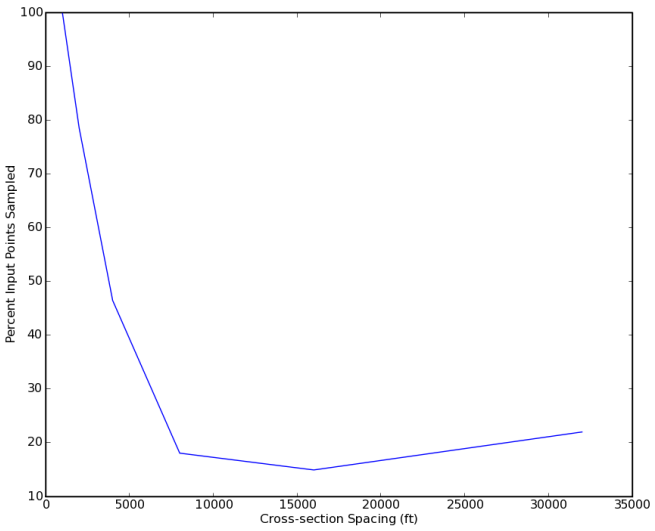

Figure 7.19: Plot of the percentage of sampled input points versus cross-section spacing for input datasets identified by cross-section spacing. When all cross-sections are used, all original points are sampled. As less cross-sections are used for triangulation, the percentage of sampled points decreases.

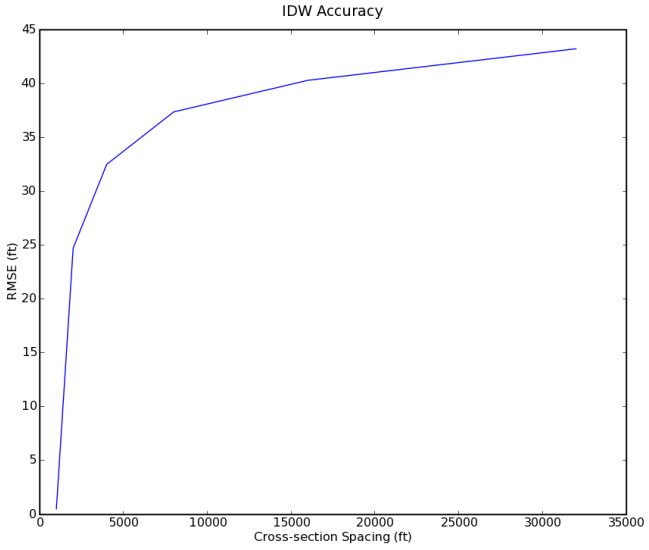

Figure 7.20: Plot of probed IDW accuracy versus cross-section spacing. Error rises sharply as cross-section spacing increases.

surface is generated. While one run was executed using triangulation, several runs were executed using the same input cross-sections for varying numbers of output cross-section points and outputs cross-sections.

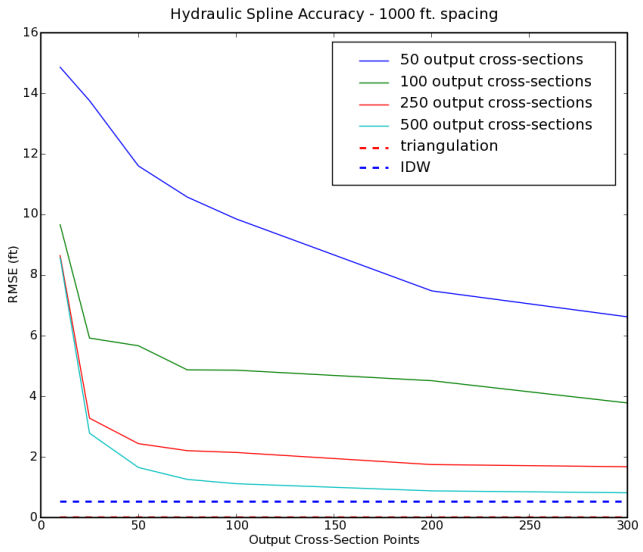

Figure 7.21: Plot of probed hydraulic spline accuracy versus number of output points generated per cross-section using test case with cross-sections spaced 1000 feet apart. Separate color plots are shown for varying numbers of generated cross-sections. The error for triangulation in this case is zero since every output surface point elevation matches an input point. Each hydraulic spline cross-section contains an average of 135 points.

As shown in the these plots, the accuracy of the hydraulic spline surface can be quantitatively more accurate than triangulation output if the spacing between cross-sections is greater than every fourth cross-section. When the input cross-sections were spaced less than 4000 feet apart, the Delaunay triangulation proved more accurate. The accuracy of the hydraulic spline remains competitive with the Delaunay triangulation as the number of input cross-sections increases.

Predictably, the choice of output parameters also influences the accuracy of the hydraulic spline surface. As the number of output cross-sections is increased, or the number of output cross-section points are increased, the accuracy improves exponentially and converges on

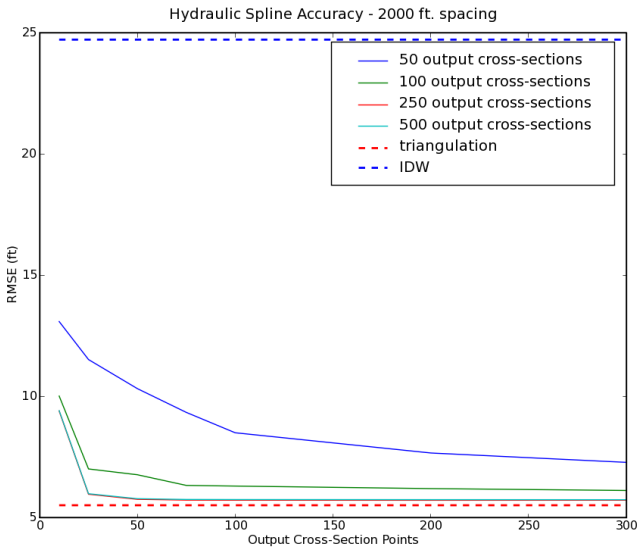

Figure 7.22: Plot of probed hydraulic spline accuracy versus number of output points generated per cross-section using the test case with cross-sections spaced 2000 feet apart. Separate color plots are shown for varying numbers of generated cross-sections. A dashed red line indicates the probed triangulation accuracy using the same input cross-section spacing. Each hydraulic spline cross-section contains an average of 135 points.

a single value, indicating that each input dataset has a limit on the number of output cross-sections and cross-section points beyond which the accuracy will not improve. In this example, setting the number of output cross-sections and cross-section points to values greater than the number of input cross-sections (130) and average points per cross-section (135) guarantees the optimal accuracy for hydraulic spline surfaces.

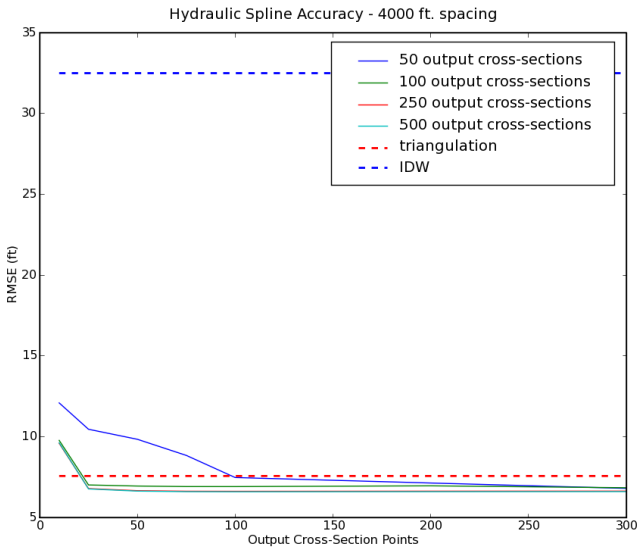

Figure 7.23: Plot of probed hydraulic spline accuracy versus number of output points generated per cross-section using the test case with cross-sections spaced 4000 feet apart. Separate color plots are shown for varying numbers of generated cross-sections. A dashed red line indicates the probed triangulation accuracy using the same input cross-section spacing. Each hydraulic spline cross-section contains an average of 135 points.

7.4.3 Profile Variations

Another series of tests were executed using variations of the profile centerline. Vertices from the original centerline were randomly repositioned within 10 feet horizontally of the original points and errors were calculated for each of the previous runs using 30 variations of the profile centerline. Table 7.2 lists the minimum and maximum RMSE values for each test case dataset. The relatively small variation in minimum and maximum RMSE demonstrates that the influence of the choice of profile centerline on accuracy is minor compared to the influence of the number of input cross-sections. The implication of this test result is that the choice of centerline has negligible impact on the output surface compared to the number of output cross-sections or number of cross-sections points chosen by the user. Therefore, a user would not have to be concerned about the exact location of the digitized centerline. As long as cross-sections are similar in their bank-to-bank coverage (i.e. all cross-sections extend

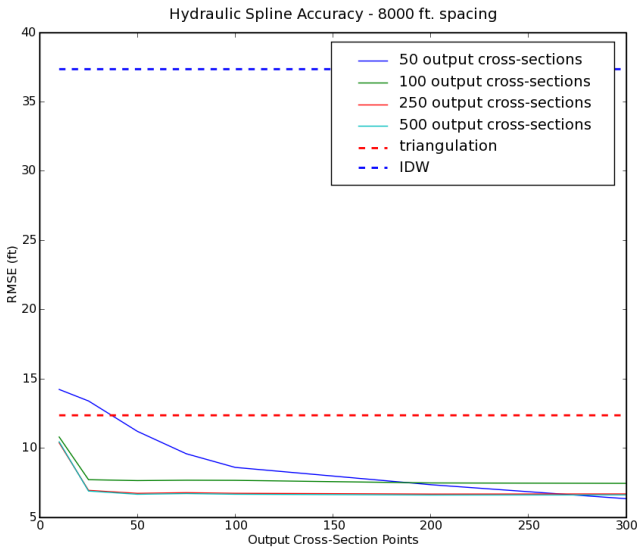

Figure 7.24: Plot of probed hydraulic spline accuracy versus number of output points generated per cross-section using the test case with cross-sections spaced 8000 feet apart. Separate color plots are shown for varying numbers of generated cross-sections. A dashed red line indicates the probed triangulation accuracy using the same input cross-section spacing. Each hydraulic spline cross-section contains an average of 135 points.

only to the bankline, or all cross-sections include overbank measurements), a single profile that approximates the channel centerline can be used by the hydraulic spline to generate accurate channel hydrography.

Input CS Spacing	Output CS Count	Output CS Point Count	Min RMSE	Max RMSE
1	50	25	13.63	13.89
	50	200	7.29	7.52
	200	25	3.29	3.34
	200	200	2.09	2.17
2	50	25	11.26	11.73
	50	200	7.52	7.87
	200	25	5.94	6.11
	200	200	5.67	5.86
4	50	25	10.32	10.69
	50	200	6.98	7.31
	200	25	6.69	6.89
	200	200	6.54	6.73
8	50	25	13.09	13.78
	50	200	6.74	7.97
	200	25	6.79	7.20
	200	200	6.48	6.98
16	50	25	20.05	21.12
	50	200	14.29	18.87
	200	25	19.96	21.47
	200	200	19.84	21.58
32	50	25	13.63	13.8
	50	200	35.09	37.79
	200	25	32.93	34.36
	200	200	33.69	35.15

Table 7.2: Minimum and maximum RMSE values for hydraulic spline surfaces generated for different test case datasets and different output parameters. Minimum to maximum RMSE values are calculated for 30 possible variations of the profile centerline location within the Mississippi River Hydrobook example.

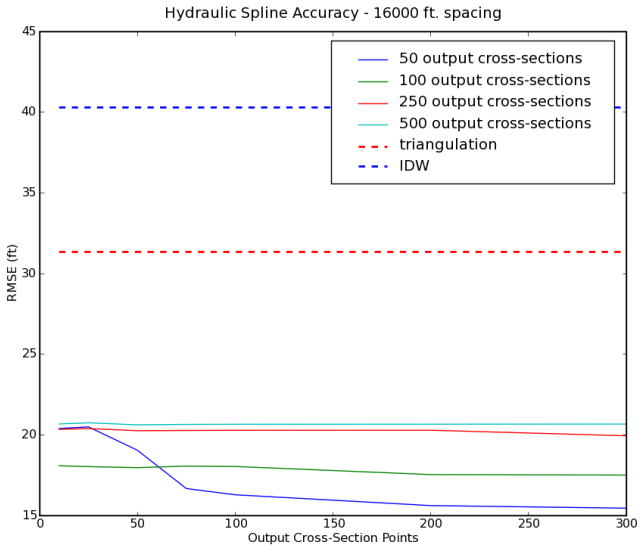

Figure 7.25: Plot of probed hydraulic spline accuracy versus number of output points generated per cross-section using the test case with cross-sections spaced 16000 feet apart. Separate color plots are shown for varying numbers of generated cross-sections. A dashed red line indicates the probed triangulation accuracy using the same input cross-section spacing. Each hydraulic spline cross-section contains an average of 135 points.

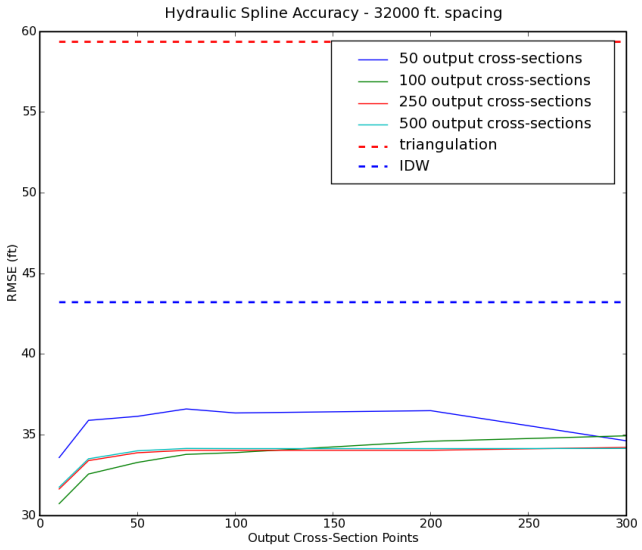

Figure 7.26: Plot of probed hydraulic spline accuracy versus number of output points generated per cross-section using the test case with cross-sections spaced 32000 feet apart. Separate color plots are shown for varying numbers of generated cross-sections. A dashed red line indicates the probed triangulation accuracy using the same input cross-section spacing. Each hydraulic spline cross-section contains an average of 135 points.

8

Hydraulic Spline Projects

The hydraulic spline algorithm has yielded tremendous benefit to a number of projects that have had to rely on historic, sparse or incomplete data. Although each project had a different goal, all started with channel data based on cross-section and profile centerline surveys, making them ideal candidates for the hydraulic spline.

8.1 Donaldsonville to the Gulf Hydraulic Modeling

By 2004, the Donaldsonville to the Gulf of Mexico Feasibility Study was well underway to determine the cost benefit ratio of constructing a new levee system, such as the one shown in Figure 8.1, between the Mississippi River and Bayou Lafourche. A feasibility study is conducted solely to compare the total cost of building a project with the value of properties and other assets in the area the would be protected by the project. As such, the money spent on a feasibility study is only a small fraction of the money that will be spent on the actual project and, consequently, the data collected for feasibility studies tends to be sparse.

For the Donaldsonville to the Gulf of Mexico feasibility study, survey cross-sections were

collected a mile apart along several key waterways in the study area. To generate usable channel geometry for flow-modeling in HEC-RAS, the U.S. Army Corps of Engineers used the hydraulic spline algorithm to model several basins using the limited survey data and the FEMA LIDAR. This data was later imported into an AdCirc grid for storm surge analysis. Figure 8.2 shows a HEC-RAS screenshot displaying cross-sections of Baker canal that were generated by the hydraulic spline algorithm.

Figure 8.1: A map produced for public meetings showing a possible levee alignment for the Donaldsonville to the Gulf of Mexico Feasibility Study.

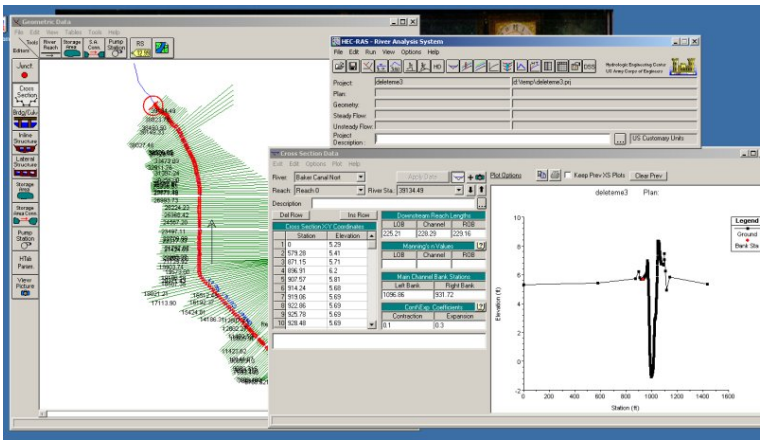

Figure 8.2: HEC-RAS model with geometry from Baker Canal North.

8.2 FEMA Digital Flood Insurance Rate Mapping

In Ascension and Lafayette parishes, the Federal Emergency Management Agency (FEMA) needed to update its flood insurance rate maps based on the latest hydrological data and determine flood risk for residents and flood insurance costs throughout the area. The hydraulic spline was used to create channel geometries for Bayou Carencro, Ill des Cannes, Black Bayou and several other channels, and merged with LIDAR data to create geometries for HEC-RAS models. The output water surface elevations were used to generate contours and classify regions into the different FEMA-designated flood zones.

8.3 IPET

After hurricane Katrina, the federal government commissioned a project to study the performance of the southeast Louisiana hurricane protection system in order to make recommendations on how to better protect the region in the future. Researchers from the U.S. Army Corps of Engineers, universities, and other independent professional organizations joined together to form the Interagency Performance Evaluation Task Force (IPET) [35].

As a component of the IPET study, an AdCirc model was developed using elevations adjusted to the new NAVD88 2004.65 vertical datum and epoch. The 2004 Mississippi River Hydrographic Survey cross-section data was first adjusted to the new datum and epoch using an IPET-developed IDW interpolation grid of known benchmark elevation changes between the new 2004.65 epoch and the previous 1996 epoch. The adjusted survey data was interpolated with the hydraulic spline algorithm to contribute Mississippi River hydrography to the AdCirc grid. With this information, the IPET team was able to model the elevation of storm surge as it travels up the Mississippi River for various storm scenarios. In 2005, IPET compiled their findings into a report that is currently being used to design a new 100

year protection system for southeast Louisiana and more effectively communicate the risk of living in the area based on this level of protection.

8.4 Conclusion

The hydraulic spline algorithm provides a powerful new instrument for surface modeling. It supplements existing algorithms like triangulation and geostatistical analysis and caters to the specialized problem domain of creating surfaces within channels in ways that more generalized algorithms cannot. The hydraulic spline is parameterized to produce an irregular grid of the user's desired dimensions. Since the algorithm is a composition of splines, it preserves and parameterizes various aspects of surface generation such as tension, as well as parameters that are more natural to users, such as number of output points and number of output cross-sections, rather than esoteric parameters such as lag distance and angle of anisotropy.

The hydraulic spline algorithm has been used successfully in several major feasibility studies and future work is planned to adapt the algorithm to levee modeling to augment development of a levee database [36] and assist project planning efforts and engineering of south Louisiana levee systems. Conveniently, levee data is also composed of cross-sections and centerlines, effectively making a levee an inverted channel. This proposed expansion of the hydraulic spline into the new domain of levee modeling will enable engineers to develop deficiency volumes based on the difference between a levee section's original design cross-sections and its surveyed cross-sections.

Bibliography

[1] Kurt D. Shinkle and Dr. Roy K. Dokka. NOAA technical report NOS/NGS 50: Rates of vertical displacement at benchmarks in the Lower Mississippi Valley and the Northern Gulf Coast. Technical report, U.S. DEPARTMENT OF COMMERCE National Oceanic and Atmospheric Administration National Ocean Service, July 2004.

[2] U.S. Army Corps of Engineers. Louisiana Coastal Area Ecosystem Restoration Study, November 2004.

[3] Laura McKnight. Morganza-to-the-Gulf not Found in Federal Budget. *Houma Today*, February 2005.

[4] Gary Brunner. *HEC–RAS River Analysis System: Hydraulic Reference Manual*. 609 Second Street Davis, CA 956-4687, 2001.

[5] R.A. Leuttich, Johannes Westerink, and N.W. Scheffner. AdCirc: An Advanced Three Dimensional Circulation Model for Shelves, Coasts and Estuaries. Technical report, US Army Corps of Engineers, Washington DC, 1991.

[6] A. Genetti. *Engineering and Design - Survey Markers and Monumentation*, September 1990.

[7] R. Vanantwerp. *Engineering and Design - Topographic Surveying*, August 1994.

[8] M. Flood. Commercial development of airborne laser altimetry-A review of the commercial instrument market and its projected growth. *International Archives of Photogrammetry and Remote Sensing*, 32, 1999.

[9] David Veneziano, Reginald Souleyrette, and Shauna Hallmark. Evaluation of LIDAR for highway planning, location, and design. In *Pecora 15/Land Satellite Information IV/ISPRS Commission I/FIEOS 2002 Conference Proceedings*, 2002.

[10] Minnesota Department of Natural Resources. A white paper on developing a high-resolution digital elevation model (DEM) and floodplain mapping program, June 2002.

[11] N.H. Younan, H.S. Lee, and D.L. Evans. Extracting digital terrain models in forestry using LIDAR data. *Geoscience and Remote Sensing Symposium*, 5:2070–2072, 2001.

[12] J.R. Banic and A.G. Cunningham. Airborne laser bathymetry: A tool for the next millennium. *EEZ Technology*, 3:75–80, August/September 1998.

[13] John Hall. Dike Curtails Burrwood Bayou Scour Threat. *Riverside*, 15(3), May-June 2004.

[14] Mark Schliefstein. Filling in the gap. *Times-Picayune*, pages A1,A8, May 2004.

[15] J. H. Atkinson, J. J. Westerink, and J. M. Hervouet. Similarities between the quasi-bubble and the generalized wave continuity equation solutions to the shallow water equations. *International Journal for Mathematics in Fluids*, 45:689714, 2004.

[16] M. Walsh. *Engineering and Design - Slope Stability*, October 2003.

[17] William Warren Caver Jr. *Slope Stability in Soft Layered Soil Systems*. PhD thesis, Oklahoma State University, May 1973.

[18] Venkatesh Mohan Merwade. *Geospatial Description of River Channels in Three Dimensions*. PhD thesis, University of Texas at Austin, August 2004.

[19] Jonathan Richard Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Selected papers from the Workshop on Applied Computational Geormetry, Towards Geometric Engineering*, pages 203–222. Springer-Verlag, 1996.

[20] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 ACM National Conference*, pages 517–524.

[21] D.G Krige. *A statistical approach to some mine valuations and allied problems at the Witwatersrand*. PhD thesis, University of Witwatersrand, 1951.

[22] H. Mitasova and L. Mitas. Interpolation by regularized spline with tension: I. Theory and implementation, 1993.

[23] Helena Mitasova, Lubos Mitas, and Harmon Russel S. Simultaneous spline approximation and topographic analysis for LIDAR elevation data in open source GIS, 2005.

[24] T. Cebecauer, J. Hofierka, and M. Suri. Processing digital terrain models by regularized spline with tension: tuning interpolation parameters for different input datasets. In *Proceedings of the Open source GIS - GRASS users conference 2002*, September 2002.

[25] M. Flanagin et al. Hydraulic splines: A hybrid approach to modeling river channel geometries. *Computing in Science and Engineering*, 9(5):4–15, September/October 2007.

[26] Maik Flanagin, Aurélien Grenotton, Jay Ratcliff, Mahdi Abdelguerfi, and Kevin Shaw. Automated terrain generation using LIDAR and waterbody survey data. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1185–1189, New York, NY, USA, 2005. ACM Press.

[27] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice.* Addison Wesley, 2 edition, 1996.

[28] Doris H. U. Kochanek and Richard H. Bartels. Interpolating splines with local tension, continuity, and bias control. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 33–41. ACM Press, 1984.

[29] Eric Haines. Point in polygon strategies. In Paul Heckbert, editor, *Graphics Gems IV*, pages 24–46. Academic Press, 1994.

[30] Antonin Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, pages 47–57. ACM Press, 1984.

[31] R. Rivest. *Introduction to algorithms*. MIT Press, Cambridge, MA, USA, 2001.

[32] Edward H. Adelson, C. H. Anderson, J. R. Bergen, Peter J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6), 1984.

[33] U.S. Army Corps of Engineers. *Mississippi River Navigation Book*. http://www.mvd.usace.army.mil/navbook/riverMain.aspx, 2007.

[34] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit*. Prentice Hall PTR, Upper Saddle River, NJ 07458, third edition, 2002.

[35] E. Link. *Performance Evaluation of the New Orleans and Southeast Louisiana Hurricane Protection System*. U.S. Army Corps of Engineers - Interagency Performance Evaluation Team, June 2006.

[36] D. McDaniel, M. Flanagin, and M. Aurand. Rebuilding the Levees with GIS. In *Proceedings of the ESRI User Conference 2007*, June 2007.

Vita

Born in Ulm, Germany and raised in the town of Duson, Louisiana, Maik Flanagin attended the Massachusetts Institute of Technology where he graduated in 2000 with a Bachelors of Science in Computer Science and Engineering and a Masters of Engineering in Computer Science and Electrical Engineering. He was ranked seventh all-time in career rushing yards for the MIT football team and won the Jerry Nason Senior Achievement Award. Maik has worked in internships for IBM in Böblingen and the MIT Media Lab, specializing in the visualization of large, complex data sets. In 2001, Maik joined the New Orleans District of the U.S. Army Corps of Engineers as a contractor to develop GIS and visualization systems, while working on his doctorate degree in Science and Applied Engineering. He is currently developing geospatial software systems to support levee reconstruction projects and aid in the struggle to preserve coastal Louisiana.