9-2013

Gradual Generalization of Nautical Chart Contours with a Cube B-Spline Snake Model

Dandan Miao
University of New Hampshire, Durham

Brian R. Calder
University of New Hampshire, Durham, brian.calder@unh.edu

Gradual Generalization of Nautical Chart Contours with a Cubic B-spline Snake Model

Dandan Miao
Center for Coastal Ocean Mapping
University of New Hampshire
Durham, NH, USA
dmiao@ccom.unh.edu

Brian Calder
Center for Coastal Ocean Mapping
University of New Hampshire
Durham, NH, USA
brc@ccom.unh.edu

***Abstract*—B-spline snake methods have been used in cartographic generalization in the past decade, particularly in the generalization of navigational charts where this method yields good results with respect to the shoal-bias rules for generalization of chart contours. However, previous studies only show generalization results at particular generalization (or scale) levels, and the user can only see two conditions: before the generalization and after generalization, but nothing in between. This paper presents an improved method of using B-spline snakes for generalization in the context of nautical charts, where the generalization process is done gradually, and the user can see the complete process of the generalization.**

Keywords—gradual contour generalization, B-spline, snake active contour model, cartographic generalization

I. Introduction

Contours are one of the primary bathymetric features on nautical charts. They depict the geomorphologic shape of the seafloor, indicate the shallow area, and provide safety of navigation information for mariners. Nautical charts make a distinction between isobaths (i.e. a line that connects all points with the same depth) and contours (i.e., a line that contains all points shallower [shoaler] than a given depth). The method described here is concerned with contours, since they are a more general description of a depth boundary, and required for maintenance of navigational safety when constructing a chart.

Charts are generally constructed from multiple sources of bathymetric data (for example, soundings from various sources, contours, indications of obstructions, etc.) and non-bathymetric data (e.g., floating aids to navigation, shore-line constructions, tides and currents, etc.). Traditionally, charts were constructed at a particular scale of representation in order to depict the information at a level of detail suitable for the intended use (e.g., very large scale, perhaps 1:5,000 for docking charts, through to very small scale, perhaps 1:1,000,000 or less, for planning an ocean crossing). Most often, the source surveys for the charts were conducted at a scale twice that of the largest scale charts for the area being surveyed and smaller scale charts were constructed from the larger scale charts by a process of generalization. As the scale of the chart changes, the contents shown on the chart are necessarily different since the space available to represent any given physical area is smaller: the detail available at the largest scale cannot be shown clearly at smaller scales. Clarity of representation is essential in a chart in order to provide a useful working document, and to promote navigational safety for surface vessels. Generalization is the process of choosing which contents should been shown and how they will be represented on the chart to achieve these goals.

More recent practice has been to construct fully electronic charts (i.e., Electronic Navigational Charts [ENCs]) for use in computer-based bridge navigation systems. These systems allow the user to zoom in and out essentially continuously and therefore require that the display system (either an Electronic Chart System [ECS] or Electronic Chart Display and Information System [ECDIS]) provide generalized data to the user on demand. Currently, such systems select the best chart available for the region from a set of charts (typically the chart with the closest scale match to that required), and display it, generalizing only within the limits of the scale minimum and maximum information coded into the chart's source data. Since these systems are essentially autonomous of the cartographer once the source data is supplied, automatic methods for generalization are even more important than they are in the traditional paper-based chart construction pipeline: here they need to be safe, and preferably aesthetically pleasing, without human intervention.

There are two main aspects of generalization: model generalization and cartographic generalization. Model generalization, also called database generalization, is generalization in the conceptual level of the data representation, while cartographic generalization, also called graphic generalization, is about the changes in the geometric shape of chart features. This paper focuses primarily on cartographic generalization. Many previous studies have been conducted on the topic of cartographic generalization, but most have been concerned with land map generalization; the generalization of nautical charts has not been widely studied.

Nautical charts differ from land maps in that they do not intend to faithfully represent the true nature of the seafloor in the area of interest, or, necessarily, all of the other components in the region. The goal, rather, is to provide a representation of

978-0-933957-40-4 ©2013 MTS

the area that is as faithful to the known true configuration of the seabed as possible (in as much as the – usually limited – source data provides information on the true configuration of the seabed), modified such that the information is inherently safe for surface navigation. So, for example, the nautical cartographer might move an indicated sounding in order to improve the clarity of the display, or intentionally modify the representation in order to suggest to the mariner that an area of the chart is unsuitable for transit. In all cases, the nautical chart must obey shoal-bias rules, meaning that the chart always shows the shallowest depth at a given position, or a modification of the known configuration of the seafloor such that the depth indicated on the chart is shoaler than the cartographer knows the water to be. This difference requires the process of nautical chart generalization to be very different from land map generalization.

A previous study by Guilbert and Saux [2] introduced a B-spline snake method to nautical chart contour generalization. This method demonstrates several generalization operators, and takes the shoal-bias rule into consideration. However this process only creates results at a given level of generalization, and there is no intermediate result between the original chart scale and the generalized scale. But in reality, when a chart with a generalization function is been displayed on an Electronic Chart System (ECS) or Electronic Chart Display and Information System (ECDIS) screen, it is more appropriate to have the generalization happen smoothly as the user zooms in and out between scales. Current generalization studies all provide generalization result at some given generalization level, but no research has shown gradual generalization on a nautical chart; this paper addresses that question.

II. BACKGROUND

A. Cubic B-spline curve definition

A B-spline curve is a parametric function defined on an interval I=[a,b] $\subset \mathbb{R}$ in $\mathbb{R}^2$ [2]. For $u \in$ I, the curve

$$f(u) = \sum_{i=0}^{m} Q_i N_i^k(u) \tag{1}$$

is a function of the control points $Q_i \in \mathbb{R}^2$ and the basis functions N_i^k. The basis functions are piecewise polynomial functions of suitable degree; here, cubic B-splines are used. The basis functions are defined recursively,

$$\begin{cases} N_i^1(u) = \begin{cases} 1 \text{ if } u_i \le u \le u_{i+1} \text{ and } u_i < u_{i+1} \\ 0 \quad \text{otherwise} \end{cases} \\ N_i^j(u) = \dfrac{u - u_i}{u_{i+j-1} - u_i} N_i^{j-1}(u) + \dfrac{u_{i+j} - u}{u_{i+j} - u_{i+1}} N_{i+1}^{j-1}(u) \text{ for } 2 \le j \le k \end{cases}$$

and the first, second and third order basis function[3] are therefore as shown in Figure 1.

This work was sponsored by NOAA grant NA10NOS4000073, which is gratefully acknowledged.

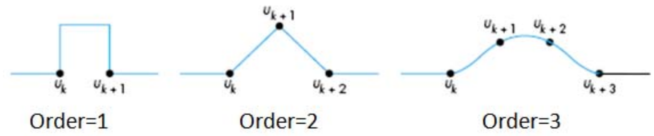

Figure 1: First three basis functions for the cubic B-spline approximation to a generic curve. The u_i are the knot vector values that define the shape of the basis functions.

The basis functions are positive and non-zero on a local interval given by the knot vector [u_i], with $0 \le u_0 \le u_1 \le u_2 \le \cdots \le u_n \le 1$ [2]. Cubic B-splines are used to ensure C^2-smoothness (i.e., continuous second derivatives) at each knot.

B. Snake model

Snakes, also called active contours, were first used in image processing. A snake is a curve defined within an image domain that can move under the influence of internal forces that describe the curve itself and external forces computed from the image data [1]. The snake is defined through a parametric curve $X(u) = [x(u), y(u)],\ u \in [0,1]$, on which the forces are defined through an energy-like term

$$E_{total} = \int_0^1 (E_{\text{int}}(X(u)) + E_{ext}(X(u)))du \tag{3}$$

where $E_{\text{int}}(l(u))$ is the internal energy of the curve, describing the smoothness, and $E_{ext}(l(u))$ is the external energy, which indicates external constraints on the system. In the system defined here, these constraints correspond to the shoal-biasing rule; when the external energy is minimized, the shoal-bias constraint has been satisfied. In general, the algorithm seeks a shape of the curve to balance the effects of the internal and external energies such that the resultant curve is as smooth as possible while still satisfying the external constraints (which may be either hard constraints – i.e., that must be satisfied – or soft constraints that express a degree of preference).

In the general snake method, the internal energy is represented as:

$$E_{\text{int}} = \int_0^1 \frac{(\alpha |X'(u)|^2 + \beta |X''(u)|^2)}{2} du \tag{4}$$

where $X'(u)$ and $X''(u)$ are the first and second derivative of $X(u)$ with respect to u, and α and β are weighting parameters that control the balance between the snake's tension and rigidity [1], and are adjusted to emphasize the required features for the given problem.

C. B-spline snake representation

For use in this work, the geomorphological constraints depend mainly on the rigidity of the snake, and therefore the value of α is set to zero [4]. The general formulation uses the second derivative at each knot as an approximation to the curvature at that point, but computing this directly is difficult in practice, so an approximation is made by finding the internal angle φ_i between three consecutive points on the curve, P_{j-1}, P_j, and P_{j+1}, and then approximating the curvature [4] as

$$\kappa(u_i) = \frac{\sin(\varphi_i)}{\frac{1}{2}\left\|\hat{P}_{j+1} - \hat{P}_{j-1}\right\|} \tag{5}$$

resulting in an internal energy of

$$E_{\text{int}} = \int_0^1 \frac{\beta\left|\kappa(u)\right|^2}{2} du \tag{6}$$

This internal energy clearly has a minimum where the curvature is uniquely zero everywhere, representing a straight line, as might be expected.

In image processing applications [5], snakes are often used to match contours in the image. The external energy term therefore often uses distance between the current location and some image-derived contour information. In the case of contour generalization, however, there is no definite target since the ENC contours move continuously offshore as the scale of the chart decreases. The primary constraint, therefore, is that the generalized snake should be on the seaward side of the original curve, and the external energy can be set to a one-sided function [4],

$$E_{ext}(X(u_j)) = \begin{cases} \dfrac{\left\|X_0(u_j) - X(u_j)\right\|^2}{\varepsilon_{vis}^2} & \text{if } X(u_j) \text{ on the wrong side} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

(where $X_0(u_j)$ is the original curve) such that there is only a penalty when the constraint is broken. The penalty term here increases according to the severity with which the generalized curve crosses to the wrong side of the original, but uses a normalization term to represent the 'minimum visualizable distance,' set according to the target scale of generalization. This reflects the fact that lines on the chart display are non-ideal, and have a defined thickness.

III. GRADUAL GENERALIZATION

The total energy is a function of each point on the B-spline snake, and as the minimum value of a function is where the first derivative of the function is zero, the snake minimizes the energy function such that

$$\nabla E_{\text{int}} + \nabla E_{ext} = 0 \tag{9}$$

Solving (9) stably requires appropriate numerical approximation of E_{int} and E_{ext}, which is covered following.

A. *Numerical solution to the first derivative of energy term*

A snake is an active contour that moves from the starting position to the desired position where the total energy is minimized. This process is incremental, with the curve moving in multiple steps as the energy is minimized; computation of the gradients in (9) provides the direction and magnitude of the steps.

The first derivative of E_{int} is

$$\nabla E_{\text{int}} = \beta\kappa(u)\nabla\kappa(u) \tag{10}$$

where $\nabla\kappa(u)$ is the first derivative of curvature $\kappa(u)$ at each point of the curve. Different approximations to $\nabla\kappa(u)$ can lead to different performance of the snake, and if the approximation method is not chosen correctly there may be spikes or inappropriately large step sizes in the iteration. Here, $\nabla\kappa(u)$ has been approximated by calculating $\frac{\partial\kappa(u)}{\partial x}$ and $\frac{\partial\kappa(u)}{\partial y}$ and then setting

$$\nabla\kappa(u) = \left(\frac{\partial\kappa(u)}{\partial x}, \frac{\partial\kappa(u)}{\partial y}\right) \tag{11}$$

In order to compute the changes of curvature $\kappa(u)$ in the x and y directions, a new pseudo point u_j' is created very close to the original point u_j (Figure 2) as defined following, and curvatures $\kappa(u_j)$ and $\kappa(u_j')$ are generated using (5). The partial derivative is then approximated as

$$\partial\kappa(u_j) = \kappa(u_j) - \kappa(u_j') \tag{12}$$

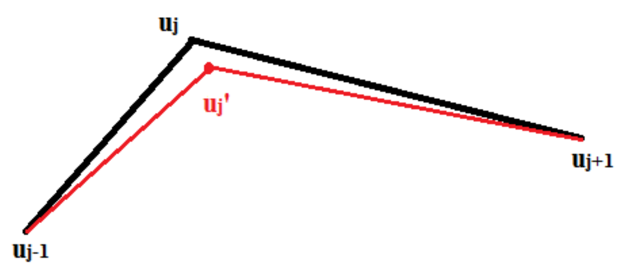

Figure 2: Geometry of the curvature derivative computation. The pseudo-point u_j' is used to compute an estimate of the rate of change of curvature using (11) and (12).

The partials ∂x and ∂y are the difference of u_j' and u_j in x and y directions. For convenience in finding the ∂u_j in x and y directions u_j' is moved solely in the x or y direction: so if the coordinates of point u_j is (x_{u_j}, y_{u_j}), then $u_{j,x}'$ will be

$(x_{u_j}+\gamma, y_{u_j})$, where γ is a very small constant value, and respectively for $u_{j,y}$'. Then, the ∂x and ∂y will be equal to γ, and $\frac{\partial \kappa(u_j)}{\partial x}$ and $\frac{\partial \kappa(u_j)}{\partial y}$ can be calculated as

$$\frac{\partial \kappa(u)}{\partial x} = \frac{\kappa(u_j) - \kappa(u_{j,x}')}{\gamma} \tag{13}$$

$$\frac{\partial \kappa(u)}{\partial y} = \frac{\kappa(u_j) - \kappa(u_{j,y}')}{\gamma} \tag{14}$$

Although this approximation is easy to calculate, the method can have numerical problems. For example, when the angle $\angle u_{j-1}u_j u_{j+1}$ is significantly smaller than 90 degrees, and the points u_{j-1}, u_j, u_{j+1} are in certain positions, the modified position of u_j that makes the angle larger will not be inside the original angle (Figure 3).

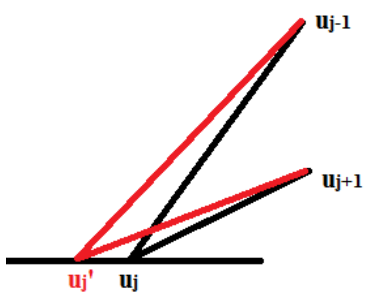

Figure 3: Geometry of the perturbed point with acute angles. Displacing the point under these circumstances can lead to poor estimates of rate of curvature change.

So, for acute angles, another approximation is used. Instead of only moving u_j' along the x or y axis, u_j' is moved along line $u_{j-1}u_j$, and then along line $u_j u_{j+1}$ to point u_j'' (Figure 4).

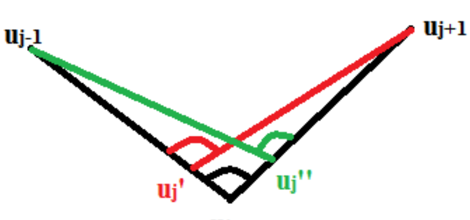

Figure 4: Geometry of rate of change of curvature computation with acute angles. Two displaced points are used to provide a pair of estimates from which the rate of change can be computed.

At point u_j', the difference of curvature $\partial\kappa(u_j')$ is $\kappa(u_j)-\kappa(u_j')$, while at point u_j'', the difference of curvature $\partial\kappa(u_j'')$ is $\kappa(u_j)-\kappa(u_j'')$. Then, $\frac{\partial \kappa(u_j)}{\partial x}$ is the sum of $\frac{\partial \kappa(u_j')}{\partial x}$ and $\frac{\partial \kappa(u_j'')}{\partial x}$,

$$\frac{\partial \kappa(u_j)}{\partial x} = \frac{\kappa(u_j)-\kappa(u_j')}{\sqrt{(x_{u_{j-1}}-x_{u_j})^2+(y_{u_{j-1}}-y_{u_j})^2}}(x_{u_{j-1}}-x_{u_j}) + \frac{\kappa(u_j)-\kappa(u_j'')}{\sqrt{(x_{u_{j+1}}-x_{u_j})^2+(y_{u_{j+1}}-y_{u_j})^2}}(x_{u_{j+1}}-x_{u_j}) \tag{15}$$

$$\frac{\partial \kappa(u_j)}{\partial y} = \frac{\kappa(u_j)-\kappa(u_j')}{\sqrt{(x_{u_{j-1}}-x_{u_j})^2+(y_{u_{j-1}}-y_{u_j})^2}}(y_{u_{j-1}}-y_{u_j}) + \frac{\kappa(u_j)-\kappa(u_j'')}{\sqrt{(x_{u_{j+1}}-x_{u_j})^2+(y_{u_{j+1}}-y_{u_j})^2}}(y_{u_{j+1}}-y_{u_j}) \tag{16}$$

Since the second method is moving the point u_j in its neighbor point's direction, the gradient of curvature approximated by this method will be, comparatively, smaller than the first method. For consistency, the gradient of curvature computed by the second method should be multiplied by a constant value. After testing, constants with 1:20 ratio are used here, meaning that the constant for the second method should be 20 times larger than the constant for the first method. This ratio attempts to ensure the snake has similar step size at all points.

The external energy in (8) can be rewritten

$$E_{ext} = \frac{(x_u - x_{u0})^2 + (y_u - y_{u0})^2}{\varepsilon_{vis}^2} \tag{17}$$

so that the ∇E_{ext} is simply the vector with components

$$\frac{\partial E_{ext}(u_j)}{\partial x} = \frac{2}{\varepsilon_{vis}^2}(x_{u_j} - x_{u0_j}) \tag{18}$$

$$\frac{\partial E_{ext}(u_j)}{\partial y} = \frac{2}{\varepsilon_{vis}^2}(y_{u_j} - y_{u0_j}) \tag{19}$$

where u_0 are the knots on the snake curve.

B. Process of generalizing a set of contours

For generalization of a set of contours, the following algorithm is used:

Input: a set of polyline (open) and closed polygon contours.

Represent all polyline contours as B-spline curves with 80% of the original curve points, reducing the number if the approximation problem becomes singular.

Preprocess closed polygon contours; equally distribute (by distance) the points on the closed contour, and add points on the closed polygon, so that the distance between each point is smaller than 1/100 of the perimeter of that closed polygon.

1 `Repeat`

2 Calculate the distance between the current snake position and all other features in the data, and find the feature that is closest to the current snake, and the closest approach distance for that feature.

3 `If` the closest approach distance is smaller than a threshold (4 m here due to the line thickness observed) indicating that the two features are too close and need to be aggregated, and this feature has not been aggregated before:

3.1 Mark the closest feature as having been aggregated,

3.2 Delete the closest feature from the active dataset,

3.3 Find the two segments that connect the current curve and closest feature,

3.4 Add those two segments and the remaining part of the closest feature into the current snake.

4 `End if`

5 `If` the distance between any two neighbor points on the current snake is larger than a suitable threshold (1/60 of the total length of current contour was chosen empirically):

5.1 Add points at all segments where two original points are too far away from each other.

6 `End If`

7 `If` the distance of any two neighbor points on the current snake is smaller than a suitable threshold (1/600 of current snake length was chosen empirically):

7.1 Find the set of all points that are within a threshold distance of their neighbors,

7.2 Find the sub-set of all groups of at least three consecutive points in sequence,

7.3 Delete the first point in each group of three continuous points.

8 `End If`

9 Calculate the step size of the current snake with (13)-(19), and move the current snake to the next step.

10 `End Repeat`

IV. Results

The data used to test the gradual generalization method is from the ENC data for Portsmouth Harbor, NH, as portrayed in US ENC US5NH02M. All US raster charts and ENCs may be downloaded free from http://www.nauticalcharts.noaa.gov.

A. *Simple line case*

The method was first tested on a single contour. The original curve had 787 points, and in the first step of B-spline snake approximation, the total number of points was decreased to 607. The curve was then generalized with the above method.

Figure 5 shows an intermediate step in the generalization, while Figure 6 shows a subset of the approximating curves between the original (dark blue) and final generalized curve.

Figure 5: Intermediate stage of the generalization process. Note that the generalized (light blue) curve does not extend more seawards that the most seaward point of the original (dark blue) curve, but has removed extraneous detail from the landward side as expected. Note also the smoothness of the curve.

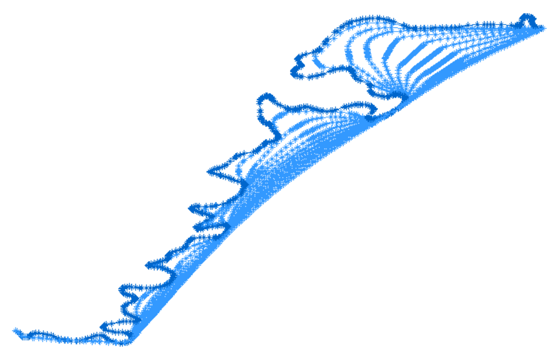

Figure 6: A composite of a subset of the generalization steps for the curve of Figure 5, showing the gradual level of generalization increase as the target scale of representation decreases. Note how the final contour is a very gentle curve only constrained to the endpoints and most seaward extent of the original curve.

B. *A set of contours*

The method was also tested on a set of contours that contains a single polyline (open) contour to be generalized, and a set of closed contours that represent the same bathymetric depth, and are located to the seaward side of the contour being treated. Figure 7 shows the input data.

Figure 8 shows an initial stage of generalization, where the target contour (light blue) has been generalized from the original (dark blue), but has yet to encounter any of the other closed contours. Figures 9-11 illustrate the situation where a number of contours have been aggregated into the target contour as it has been increasingly generalized. The shape of the contours being aggregated can be seen to be preserved from Figure 7, with smooth transitions being created as the target contour encounters the landward-most point of the each contour. Eventually, all of the contours are aggregated, and the result, Figure 12, is a very smooth contour that maximizes the outer hull of all of the contours, while smoothing the segments between the promontories.

Figure 7: Input data for the second experiment. The blue contour is being generalized, while the grey closed contours represent the same bathymetric depth, but are located seaward of the target contour.

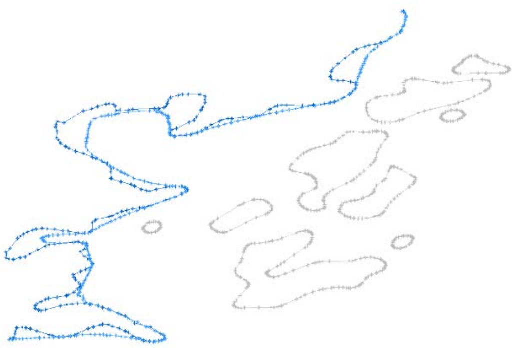

Figure 8: An early stage of generalization, where no other contours have been encountered: only cartographic generalization has been applied to the target contour.

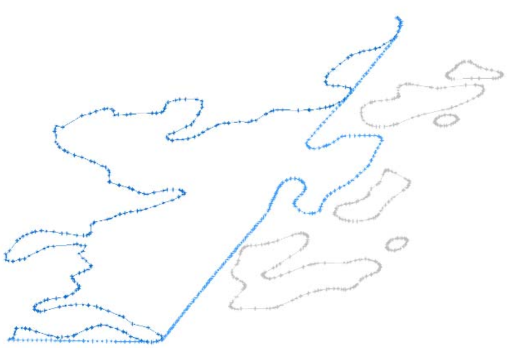

Figure 9: The generalized result after a number of contours have been aggregated. Note that the seaward shape of the contour currently being aggregated has been preserved.

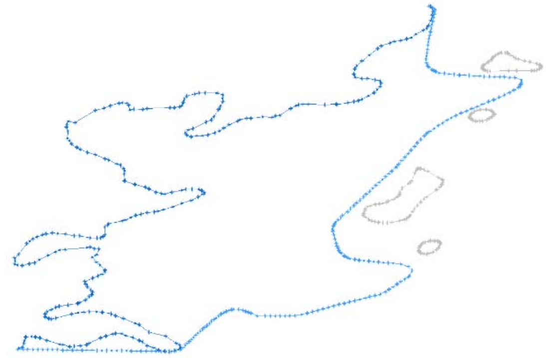

Figure 10: A further stage of aggregation, preserving the seaward shape of the contours so far aggregated, but with smoothed transitions.

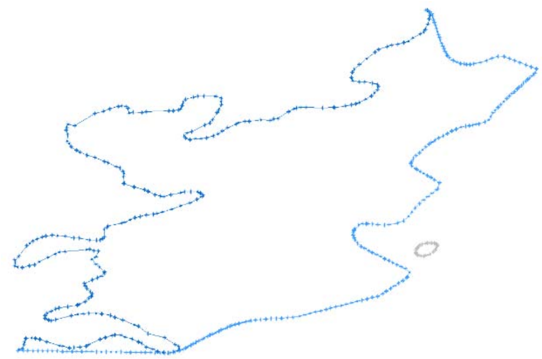

Figure 11: A late stage example of aggregation, with one final contour to be aggregated. Note the smoothed shape of the northern boundary of the generalizing snake, which continues to be smoothed as the process continues (c.f. Figure 12).

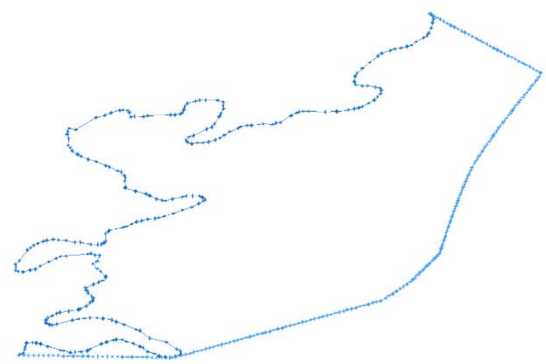

Figure 12: The final stage of generalization. All of the closed contours have been aggregated and generalized, so that the result (at much lower scale) preserves only the outer promontories of the originals, with smooth transitions between them.

V. Discussion

The gradual generalization method is useful as it provides a scale-less process of generalization, with the mid-method results of this generalization covering many scales. It also combines cartographic generalization and model generalization, which generates better results than using either one of them alone. However, there is room for improvement of this method, for example in simultaneous generalization of multiple contours, interaction with other bathymetric and non-bathymetric features, and further model generalizations. Future work will be on specifying more general input data types, finding a suitable workflow for those data types, and evolving a general workflow for complete chart data.

REFERENCES

[1] C. Xu, and J. L. Prince, "Snakes, shapes, and Gradient vector Flow" IEEE Transactions on Image Processing, 7(3) 359-369, 1998.

[2] E. Guilbert and E. Saux, "Cartographic generalisation of lines based on a B-spline snake model" International Joournal of Geograohic Information Science, 22(8) 847-870, 2008.

[3] E. Angel, D. Shreiner, Interactive Computer Graphics, 6th Edition, Addison-Wesley, 2012, Chapter 10.

[4] E. Guilbert, E. Saux, and M. Daniel, "Conflict Removal between B-spline Curves for Isobathymetric Line Generalization Using a Snake Model", Cartography and Geographic Information science, 33(1) 37-52, 2006.

[5] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active Contour Models", International Journal of Computer Vision, 321-331, 1987.