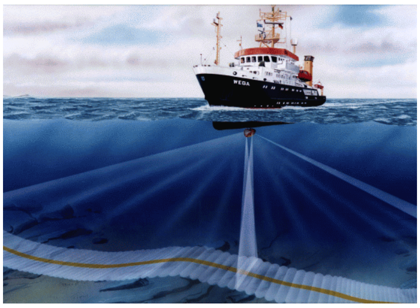
Generation and generalization of safe depth-contours for hydrographic charts using a surface-based approach

R.Y. Peters, H. Ledoux, B.M. Meijers

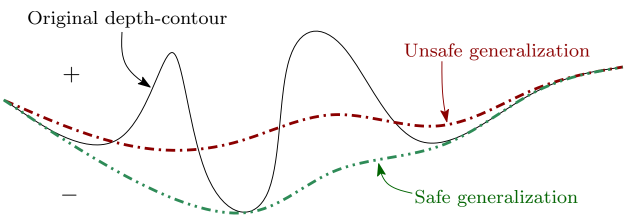
OTB Research Institute for the Built Environment,
Delft University of Technology

1 Introduction

Depth-contours are an essential part of any hydrographic chart—a map of a waterbody intended for safe ship navigation. Traditionally these were manually drawn by skilled hydrographers from a limited set of surveyed depth measurements. Nowadays this process of map making is shifted towards the digital domain, not in the last place because of the sheer size of the point clouds that result from modern surveying techniques such as Multi Beam Echo Sounding (MBES) (see Figure 1a). This is no trivial task, since the the produced depth-contours need to comply with the four hydrographic generalization constraints of safety, legibility (smoothness), topology and waterbody morphology. The challenge is to solve the generalization problem for these four constraints.



(a) Multi Beam Echo Sounding (MBES) is based on the principle of measuring the time of flight of hundreds of individual signal pulses.



(b) During generalization, contours can only be moved towards greater depth, indicated by a '−', in order to respect the safety constraint.

Figure 1

The hydrographic generalization constraints ensure that the resulting contours are compatible with the purpose of a hydrographic chart (to be an efficient and reliable guide for a shipper). Zhang and Guilbert (2011) describe these generalization constraints. In our own words these are:

1. The *safety constraint.* A hydrographic chart is primarily a depth-map of a waterbody. At every location, the indicated depth may never be deeper than the depth that was originally measured at that location. See Figure 1b. This is to guarantee that a ship never runs aground because of a faulty map.

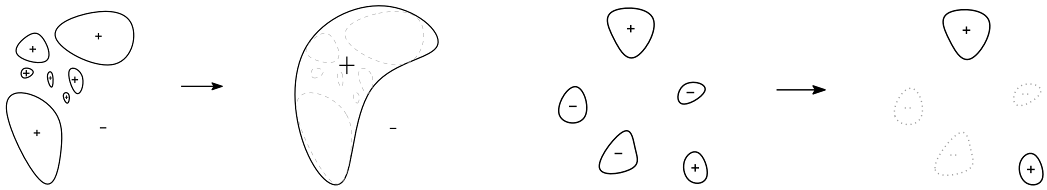
2. The *legibility constraint.* An overdose of information slows down the map reading process, thus only the essential information should be present on the map in a form that is clearly and efficiently apprehensible. This requires cartographic generalization.

3. The *topology constraint.* The topology of the depicted map elements must be correct. Contour lines for instance may not intersect.

4. The *morphology constraint.* The map should be as realistic and accurate as possible, i.e. the overall shape of the morphology of the underwater surface should be clearly perceivable and defining features should be preserved.

Note that the safety constraint is unique to hydrographic generalization. Subsequently it is not dealt with in generalization solutions for landforms. Moreover, and primarily related to the morphology constraint, as argued by the cartographer Imhof (1965): "One must never overlook the fact that (geographic) *surfaces* are being depicted with contours. A single line says very little. One line does not define a surface. Everything comes back, eventually, to the formation of the system of lines, that is, the surface." This is a clear argument for an approach that does not generalize contours independent from each other.

It should further be noted that the four hydrographic generalization constraints are sometimes incompatible with each other. For instance, the morphology constraint tells us to stay close to the measured shape of a waterbody. Yet, the legibility constraint forces us to deviate from that exact shape by disregarding details. And, for the sake of the safety constraint, contours can only be modified (to achieve legibility) so that the safety is respected at all times. It is therefore evident that the end result must be a reasonable compromise between the four generalization constrains, although the safety constraint can never be broken.



(a) Aggregation

(b) Omission

Figure 2: Generalization operators for hydrographic contours. The '+' and '−' symbols respectively indicate shallow and deep regions.

In order to satisfy the generalization constraints cartographic generalization, i.e. generalization of spatial data for cartographic visualization (Weibel, 1997), must be applied. Figure 2 illustrates two cartographic generalization operators. Aggregation (Figure 2a) reduces the number of contours by merging a group of nearby contours that represent local maxima. Omission (Figure 2b) means removing features that are insignificant for the map purpose, in the hydrography these are often contours that indicate particularly deep areas that pose no risk to safe ship navigation.

In this paper we show that the methods that are currently used in practice for the generation and generalization of depth-contours for hydrographic charts are not fully complying with the hydrographic generalization constraints. In fact the most critical constraint, for safety, is never strictly met at all.

We also introduce and demonstrate a surface-based generalization approach that is guaranteed to be safe. The concept is based on the Voronoi Diagram (VD) and the related Laplace spatial interpolation (Belikov et al., 1997; Hiyoshi and Sugihara, 1999), a variant of the more widely known Natural Neighbour or Sibson interpolation (Sibson, 1981). We have implemented this using the Delaunay Triangulation (DT), which is a unique mapping of the VD (i.e. its *dual*). Furthermore, we describe a number of generalization operators based on this concept. Results, based on real data, are provided as well.

2 Current approaches to generating and generalizing depth-contours

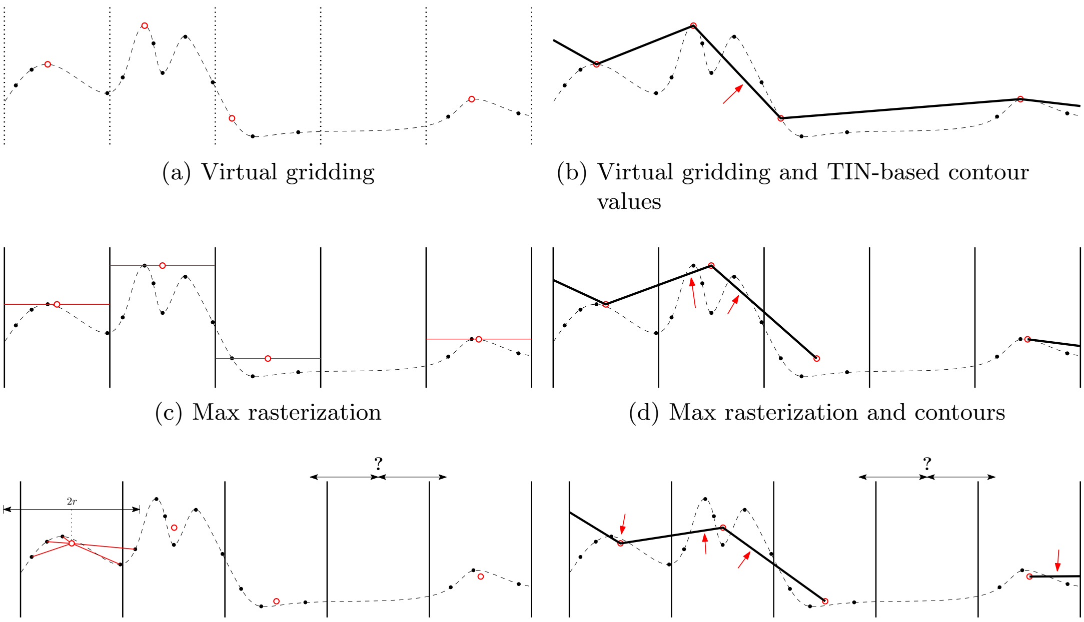
Practitioners use mostly two methods to generate depth-contours from randomly distributed input points. The first method is to use shallow filtering or gridding (Smith, 2003). Supposedly in favor of the safety constraint, only the shallowest point is retained for every cell in a grid or quadtree that is overlaid on the input data. Resulting points are either stored exactly (see Figure 3a), preserving their planimetric coordinates, or as a raster (see Figure 3c), in which case exact planimetric coordinates are discarded.

However, picking the shallowest point per grid cell does not guarantee safe contours in principle. The problem is that contour extraction algorithms perform a linear interpolation on top of the points present in the data structure. As can be observed from Figure 3b, this easily results in safety violations at 'secondary' local maxima in a single grid cell. The number and severity of these violations is related to the used resolution of the grid cells. A bigger cellsize will result in more and more severely violated points. Furthermore, when compared to triangulation, rasterization is likely to cause more extreme safety violations, because the depths are snapped to the cell's centers.

A second method that is used to generate contours is (some variant of) inverse distance weighting (IDW) interpolation (Shepard, 1968), that assigns raster cells a depth that is an inversely distance weighted sum of nearby points (see Figure 3e). From Figure 3f it is evident that this can also easily result in a violation of the safety constraint.

None of the above methods strictly respects the safety constraint. As a consequence, also any processing chain that uses one of these methods may not be absolutely safe.

While the above methods are able to achieve a certain form of generalization, most notably the somewhat arbitrary reduction of high frequency detail, i.e. the noisy character in raw contours (see Figure 6a), generalization can also be achieved by processing the contour lines themselves. However, two general problems arise



(a) Virtual gridding
(b) Virtual gridding and TIN-based contour values
(c) Max rasterization
(d) Max rasterization and contours
(e) IDW rasterization
(f) IDW rasterization and contours

Figure 3: On the left: profile views of different filtering and rasterization methods. On the right: the corresponding contours. Red arrows indicate where the safety constraint is violated with respect to the original points. Also note that in case a grid cell contains no data, no contours can be derived.

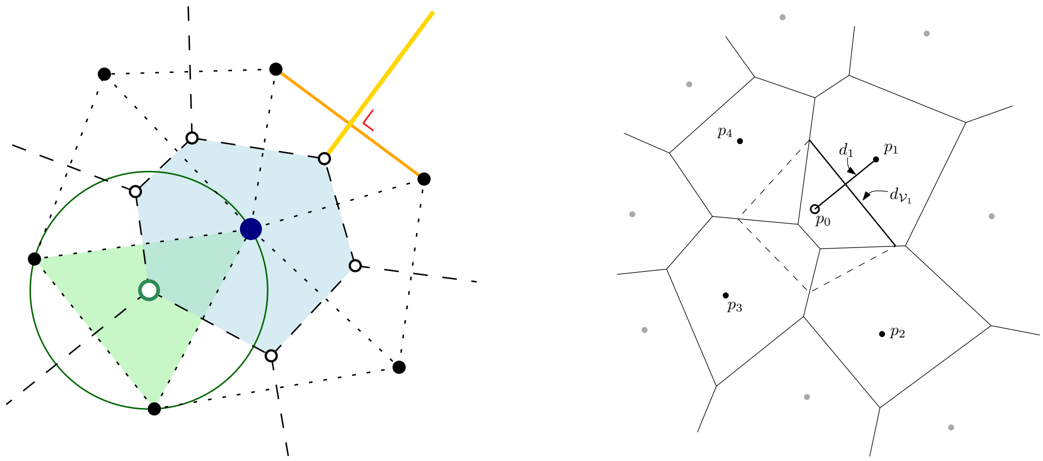
with line-based methods. Firstly, there is the problem of intersecting contour lines (the topology constraint). And secondly, these methods require safe and clean input contours to begin with. However, as described above, obtaining those safe contours is not a trivial task.

In reaction to the problem of intersecting contours, Hennau and De Wulf (2006) propose a method that combines a line-based smoothing technique with a TIN-based patch smoothing technique. Unfortunately, they do not consider the safety constraint.

A line-based generalization method that does respect safety is double-buffering (Smith, 2003). It works by buffering the set of input contours back and forth by some configurable buffering distance. A weakness of this approach is its non-adaptiveness, i.e. the buffer-distance is strongly dependent on local details in the contour lines. Guilbert and Lin (2007) and Guilbert and Saux (2008) use a spline-snake model to achieve hydrographic contour generalization in an iterative optimization. The authors do note that the preservation of safety comes at the price of a significantly slower convergence of iteration. In addition, the algorithm requires manual intervention in some cases.

3 A surface-based approach

We present a different approach to the problem of generating and generalizing hydrographic depth-contours from raw input points that is based on a continuous interpolation of the raw input points. It does not perform any kind of gridding or rasterization. Instead, it retains all input points at their exact planimetric coordinates during processing. Implementation-wise, the points are represented as vertices in a Delaunay Triangulation (DT), which can be considered a piecewise linear surface. We, however, define a continuous surface through the Voronoi Diagram (VD), a unique mapping of the DT (see Figure 4a), which conveniently exposes topological relationships of adjacency between vertices. Natural Neighbour or Sibson interpolation (Sibson, 1981; Gold, 1989) is a spatial interpolation method that exploits these relationships of spatial adjacency. The basic idea is to insert a point in the VD at the location that is to be interpolated, which of course creates a new Voronoi cell at that location. The area that is 'stolen' by this new cell from adjacent Voronoi cells (the Natural Neighbours), sets the weights in a weighted sum of the depths of those Natural Neighbours. The Laplace interpolant (Belikov et al., 1997; Hiyoshi



(a) The Voronoi Diagram (VD), drawn in dashed lines, uniquely maps to the Delaunay Triangulation (DT), drawn in dotted lines.

(b) Laplace interpolation. The depth at p_0 is defined as the weighted sum of the depths of its natural neighbours $p_1, \ldots, p_4$, with weights $w_i = \frac{d_{\mathcal{V}_i}}{d_i}$

Figure 4

and Sugihara, 1999) that we use, is a variant of Sibson interpolation. See Figure 4b. It is very similar to Sibson interpolation in terms of properties and results, but computationally faster to compute. Some interesting properties of these interpolation methods are:

1. Smooth: the derivative of the interpolated surface is continuous.
2. Adaptive: it performs well for varying configurations and density patterns of sample points.
3. Automatic: it requires no manual configuration.

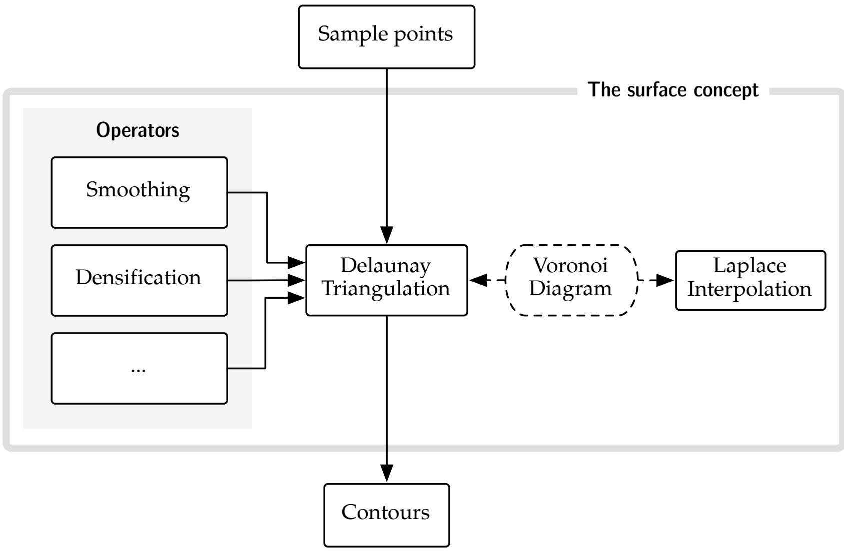


Figure 5: Overview Voronoi- and surface-based approach

Figure 5 schematically depicts how our surface-based approach works. Input points are triangulated and contours are extracted from the that triangulation. However, before extracting contours the triangulation is altered by means of several operators. These operators make use of the beneficial properties of the Laplace interpolation that is based on the triangulation, to achieve generalization of the surface. Topological correctness is guaranteed in the contours since they are directly derived from the triangulation. Therefore the topology constraint is satisfied.

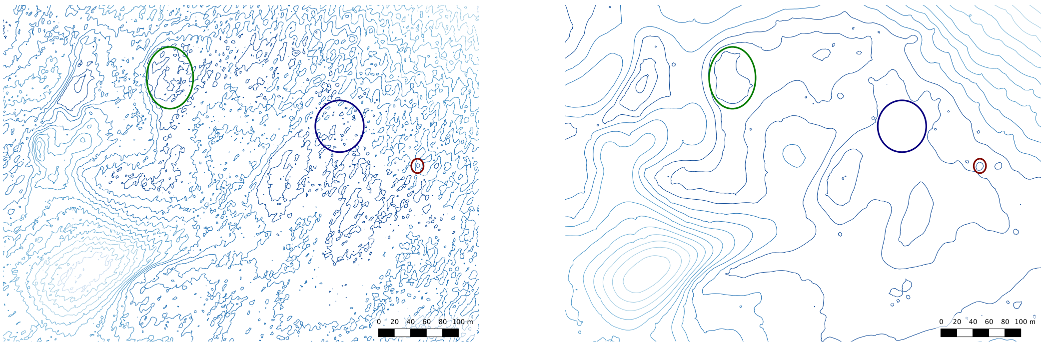
We have implemented the Voronoi- and surface based approach in the C++ programming language using several open source libraries, most notably the Computational Geometry Algorithms Library (CGAL) (Boissonnat et al., 2002).

3.1 Operators on the surface

Every operator that is defined in our surface-based approach is based on the Laplace interpolant (see Figure 4b) that we constrained in such a way that a sample point is never removed and never moved to a deeper depth. Consequently, all modifications in the shape of the surface are upwards, which implies that the safety constraint is respected at all times.

The surface-based smoothing operator

The smoothing operator re-evaluates the depth of a vertex using a Laplace interpolation of its Natural Neighbours. Only if the interpolated depth is higher than the original depth, the depth of the vertex is updated. Thus, smoothing does not change the planimetric coordinates of vertices, it only lifts vertices upwards. It can be performed either on a portion of a dataset or the whole dataset. Furthermore this operator can be applied any number of times, delivering more generalization with each pass. The smoothing operator, as defined here, not only smoothens the surface (i.e. by reducing its angularity as described by Kimerling and Muehrcke (2009)), but also simplifies it, in the sense that the overall complexity in the shape of the surface and its derived contours is reduced. The effect of the smoothing operator on



(a) Raw contours

(b) 100x Smoothed contours

Figure 6: The effect of the smoothing operator on the extracted contours (at every 50cm). The ellipses mark areas where aggregation (green), omission (blue) take place. Note that local maxima as the one indicated by the red ellipse are preserved.

the contour lines is demonstrated in Figure 6. The contours of Figure 6a are directly derived from the Delaunay Triangulation (DT) of input points. It is evident that the amount of clutter, the number of superfluous bends and the sharp angles in the contours lines are significantly reduced in Figure 6b, that demonstrates the same set of contours after one hundred simplification passes. As indicated by the ellipses by Figure 6, the sort of cartographic generalization illustrated in Figure 2 is effectively achieved. This is thus in service of the legibility constraint.

The surface-based densification operator

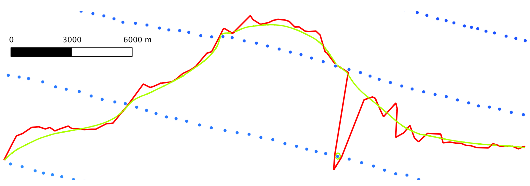
The aim of densification is to reduce the error between the continuous Laplace interpolation of input points and its computer representation. By inserting vertices at the circumcenters of large triangles, the resolution of the Delaunay triangulation is improved, and subsequently also the resolution of any contours that are derived from it. Naturally, these newly inserted vertices are assigned a depth value using the Laplace interpolant. The densification operator is particularly relevant for (parts of) datasets that have very sparse sampling. By densifying the DT that represents the surface, smooth contours, that serve the legibility constraint, can still be achieved. Figure 7a illustrates this.

3.2 Comparison with other methods

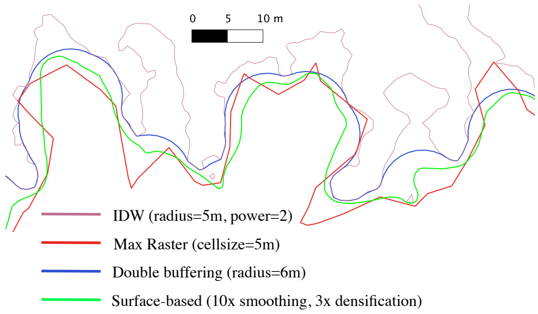
Figure 7b shows an overlay of representable results from the most common current methods and our approach.

4 Conclusion and future work

We have introduced a new approach for the generation and generalization of hydrographic depth-contours. Rather than performing generalization on input points or



(a) Densification: Comparison of corresponding contour lines. The red line is derived from a surface that was not densified, the green line is derived from a surface that was densified 3 times. Also shown are the sample points in blue.



(b) Comparison between methods for hydrographic contour generalization.

Figure 7

derived contour lines, we work on the surface that is inferred from the input points by means of the Laplace interpolant. All points are retained in the corresponding DT that serves as a basis for our generalization operators. Those operators are constrained such that no point is removed or moved downwards. In this manner we strictly respect the hydrographic safety constraint. Also the topology constraint is guaranteed, since contours are directly extracted from the generalized surface. As for the legibility and morphology constraint, these can—to some degree—be controlled by the number of smoothing passes that is applied on the surface. Less smoothing passes result in a contours that relate closer to the measured geomorphology, yet more smoothing passes result in more legible contour lines. Determining the optimal number of smoothing passes is still a manual process.

Future work includes searching for ways to automatically control the generalization process based on our surface-based approach. One idea is to conceptually link the surface with the contours, i.e. by performing specific surface generalization at places that are interesting based on analysis of the corresponding contours. Alternatively, such analysis might be performed directly on the surface. Relevant work in this area is presented by Zhang and Guilbert (2011) and Guilbert and Zhang (2012). A second area of interest is the addition of generalization operators to our surface-based approach. We are, for instance, working on a way to perform more specific aggregation in the surface by re-interpolating surface patches in between a group of peaks, using only those those peaks and the surrounding samples.

References

V.V. Belikov, V.D. Ivanov, V.K. Kontorovich, S.A. Korytnik, and A.Y. Semenov. The non-sibsonian interpolation: a new method of interpolation of the values of a function on an arbitrary set of points. *Computational mathematics and mathematical physics*, 37(1):9–15, 1997.

Jean-Daniel Boissonnat, Olivier Devillers, Sylvain Pion, Monique Teillaud, and Mariette Yvinec. Triangulations in CGAL. *Computational Geometry—Theory and Applications*, 22:5–19, 2002.

Christopher M. Gold. Surface interpolation, spatial adjacency and GIS. In Jonathan Raper, editor, *Three Dimensional Applications in Geographic Information Systems*, pages 21–35. Taylor & Francis, 1989.

E. Guilbert and H. Lin. Isobathymetric line simplification with conflict removal based on a b-spline snake model. *Marine Geodesy*, 30(1-2):169–195, 2007.

E. Guilbert and E. Saux. Cartographic generalisation of lines based on a b-spline snake model. *International Journal of Geographical Information Science*, 22(8): 847–870, 2008.

E. Guilbert and X. Zhang. Generalisation of submarine features on nautical charts. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume I-2, Melbourne, Australia, 2012.

Marc Hennau and Alain De Wulf. Smoothing contour lines of hydrographical maps: An optimised approach. In *Proceedings 15th International Congress of the International Federation of Hydrographic Societies*, pages 199–201, Antwerp, Belgium, 2006.

H. Hiyoshi and K. Sugihara. Two generalizations of an interpolant based on voronoi diagrams. *International journal of shape modeling*, 5(2):219–232, 1999.

E. Imhof. *Cartographic relief presentation*. ESRI Press (2007), 1965. ISBN 9781589480261.

A. Jon Kimerling and Phillip Muehrcke. *Map use: reading and analysis*. ESRI Press Academic, 2009.

D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings 23rd ACM National Conference*, pages 517—524, 1968.

Robin Sibson. A brief description of natural neighbour interpolation. In Vic Barnett, editor, *Interpreting Multivariate Data*, pages 21–36. Wiley, New York, USA, 1981.

Shepard M. Smith. The navigational surface: A multipurpose bathymetric database. Master's thesis, University of New Hampshire, 2003.

R. Weibel. Generalization of spatial data: Principles and selected algorithms. *Algorithmic foundations of geographic information systems*, pages 99–152, 1997.

X. Zhang and E. Guilbert. A multi-agent system approach for feature-driven generalization of isobathymetric line. *Advances in Cartography and GIScience. Volume 1*, pages 477–495, 2011.