Skeleton Based Contour Line Generalization

Krzysztof Matuk[1], Christopher Gold[2], and Zilin Li[1]

[1]The Department of Land Surveying Geo-Informatics
The Hong Kong Polytechnic University, Hong Kong.
kmatuk@op.pl, lszlli@polyu.edu.hk
[2]GIS Research Centre, School of Computing, University of Glamorgan
Pontypridd CF37 1DL, Wales, UK.
christophergold@voronoi.com

Abstract

Contour lines are a widely utilized representation of terrain models in both cartography and Geographical Information Systems (GIS). Since they are often presented at different scales there is a need for generalization techniques. In this paper an algorithm for the generalization of contour lines based on skeleton pruning is presented. The algorithm is based on the boundary residual function and retraction of the skeleton of contour lines. The novelty of this method relies on pruning not only the internal skeleton branches, but also those skeleton branches placed outside the closed contour polygon. This approach, in contrast to original method which was designed for closed shapes is capable of handling also open polygonal chains.
A simplified version of the skeleton is extracted in the first step of the algorithm and in the next a simpler boundary is computed. The simpler boundary as shown in this paper, can be found using three different ways: detection of stable vertices, computation of an average vertex and approximation of the boundary by Bezier splines.

1 Introduction

Generalization is one of the most important processes in cartography and GIS. As stated by Keates (1989) , generalization is a process of adjusting the representation of a phenomenon for adaptation to a map scale. A more complex description of generalization has been presented by McMaster and Shea (1992). The authors divide the process of digital generalization into three parts: the analysis of philosophical objectives (why to generalize), cartometric evaluation (when to generalize) and spatial and attribute transformations (how to generalize). The philosophical objectives, besides adapting the map to viewer needs and reducing complexity, also minimize the use of resources

(storage, processing time) and are also important to spatial accuracy and aesthetics.
Contour lines are the most widely used model for the representation of the terrain models in Cartography and GIS. Starting from the Douglas-Peucker algorithm (Douglas and Peucker. (1973)), trough ϵ-band by Perkal (1958), natural principle by Li and Openshaw (1993) and line generalization based on the analysis of shape characteristics by Wang and Muller (1998), contour lines can be generalized in a number of ways.
In this work yet another answer to the question: "how to generalize ?" is given in the form of a new algorithm for generalization of contour lines. The algorithm utilizes a medial axis of a two dimensional object as described by Blum (1967). In particular the algorithm makes use of a medial axis extracted from samples taken from the boundary of a shape or placed along an open curve. Extraction of the medial axis (or skeleton) from scattered points is described by several authors Attali (1997), Amenta et al. (1998) and Gold and Snoeyink (2001). The algorithm by Gold (1999) and Gold and Snoeyink (2001) appears to be the least complicated and is used in this study.
A simplification of a shape is possible by pruning its skeleton. Two interesting, from the point of view of this study, algorithms have been proposed by Gold and Thibault (2001) and Ogniewicz and Kübler (1995). The first is an iterative process, based on pruning the leaves of the skeleton only. The algorithm moves a leaf vertex of the skeleton to its parent. Simultaneously with pruning the skeleton, the outlying vertex of the boundary is moved onto the circle centered on the parent skeleton vertex.
The second algorithm is based on the boundary potential functions. The value of the boundary potential makes possible extraction of significant parts of a shape and its skeleton. This method has been proven to be a controllable and stable method for shape simplification. Unfortunately, due to taking into consideration internal skeleton branches only, the method does not apply to open polygonal chains.

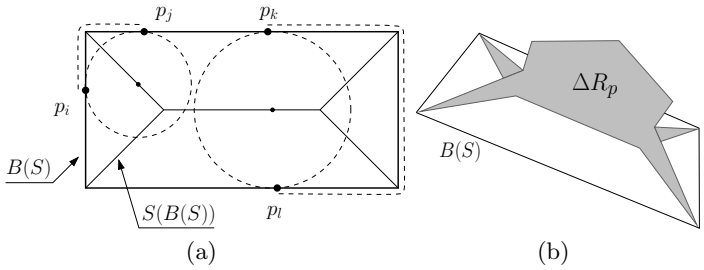

Fig. 1. Boundary distance between two points (a) and, a perspective view of the boundary potential function (b).

In the algorithm presented in this paper the potential residual function ΔR_p is utilized as a measure of a feature size. The boundary potential residual as shown in Figure 1 is the distance along a boundary, between the two samples connected by a Delaunay edge (eg. $\overline{p_i p_j}$). The value of the distance is assigned to the Voronoi edge dual to $\overline{p_i p_j}$.
In order to find a useful cut off threshold, the idea of defining the smallest visible object (SVO) offered by Li and Openshaw (1992, 1993) is borrowed. The main idea of their algorithm relies on the removal of some of the features which are invisible at the desired scale of the map. The method is known as the Li-Openshaw algorithm and is based on the definition of the smallest visible object (SVO) given by equation:

$$SVO = S_t * D * (1 - \frac{S_f}{S_t}) \tag{1}$$

where:

S_t - scale factor of the source map
D - diameter of the SVO at the map scale. In the range of this diameter all information about the shape of the curve can be neglected
S_f - desired map scale factor

The organization of this paper is as follows: Section 2 gives definitions of symbols and definitions used in further sections. The new algorithm and results obtained of its application to some real data is described in Sections 3 and 4. Finally Section 5 presents conclusions and plans for future development of the algorithm.

2 Definitions

The algorithm presented in this work utilizes a few different techniques. In order to make the argument consistent, the introduction of some common symbols and definitions may be useful. Most are slight modifications of the terms described by Amenta et al. (1998), Ogniewicz and Ilg (1992), Ogniewicz et al. (1993), and Ogniewicz and Kübler (1995). Others are widely used in computational geometry.
The input of the algorithm is a set of samples taken from the boundary of a two dimensional shape. The shape is denoted as F and the samples obtained from its boundary as S ($S \subset F$). In order not to limit considerations to closed shapes, the samples can be also taken from an open curve.
The two dimensional Delaunay triangulation of the sample boundary points is used as a starting point. It is denoted as $\mathcal{DT}(S)$, the dual graph of the Delaunay triangulation, the Voronoi diagram is denoted as $\mathcal{V}(S)$. At the same time, for any edge e, $e \in \mathcal{DT}(S)$, the symbol $\mathcal{D}(e)$ denotes the Voronoi edge dual to e; which means that $\mathcal{D}(e) \in \mathcal{V}(S)$.
The boundary of the closed shape F can be approximated by a polygon. At

the same time, a polyline can be used to approximate the boundary of an open curve. Since a distinction is not made between open and closed objects at this stage, both are denoted as $\hat{B}(S)$. The input set is not restricted to one boundary, which means that $\hat{B}(S)$ may be composed of many open curves and/or many closed shapes. This leads to the following statement: $\hat{B}(S) = \{\breve{B}_1(S), \ldots, \breve{B}_N(S)\}$, where $\breve{B}_k$ is any shape of the input set.
The medial axis of some shape F is a set of points having at least two nearest neighbors on the boundary of F. It has been proven by Brandt (1994) that when the sampling density approaches infinity, the Voronoi diagram of the boundary samples of a two dimensional shape converges to the medial axis of F. Sampling density in real data is never infinite, hence the medial axis of F can only be approximated. The approximation of the medial axis or the skeleton is denoted as $\mathcal{S}(\breve{B}_k(S))$.

3 Polygon and polyline simplification

As mentioned in the previous sections, the existing algorithms based on skeleton pruning have some disadvantages. These shortcomings prevent them from being applied for controllable generalization of contour line models of terrains. The method proposed by Ogniewicz et al. needs information about which skeleton part is internal for each polygon. While the algorithm by Gold and Thibault (2001) is able to prune the leaves of the skeleton only. It can not perform more extensive pruning. The algorithm presented here combines the advantages of both methods. It is able to perform a simplification by retraction of the skeleton branches on both sides of a curve. The simplification is not limited to skeleton leaves and is driven by the boundary potential function.

3.1 Simplification of a skeleton

The simplification process starts from the extraction of the skeleton and computation of the boundary potential values for each Voronoi edge which is part of the skeleton. This is followed by extraction of the skeleton edges which have boundary potential greater than some threshold t ($t \in R^+$)(Figure 2). This leads to the following definitions:

Definition 1. *For some threshold $t \in R^+$, $\mathcal{S}(\breve{B}_k(S),t)$ denotes the skeleton of $\breve{B}_k(S)$ simplified for threshold t.*

Definition 2. *For some threshold $t \in R^+$, $\breve{B}_k(S,t)$ denotes the boundary $\breve{B}_k(S)$ simplified for threshold t.*

The pruning of the skeleton causes elimination of two dimensional features of a contour line. This operation, if performed in a coordinated way on all the contour lines, should prevent them from intersecting one another. The coordination must assure that if one polygon is placed inside the other before

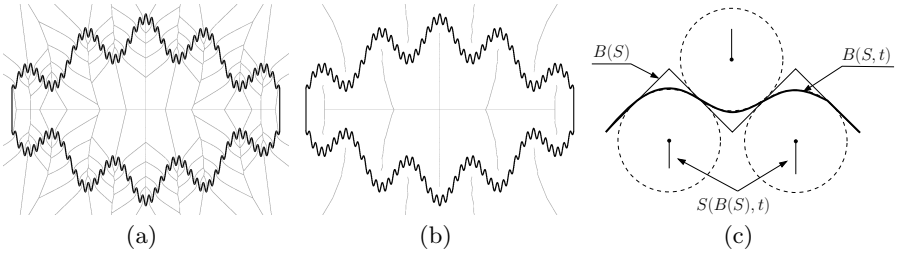

Fig. 2. Simplification of the skeleton. Source shape and its skeleton (a) and the same shape with the skeleton simplified for some threshold (b). Interpolation of the shape between medial circles placed on the opposite sides of the boundary (c) (bold line).

simplification, it will remain enclosed by the external polygon after simplification as well. The same condition should be fulfilled in the case of the variations on the contour lines or in other words two dimensional features. This means that not only must the direction of a simplification be the same for all corresponding two dimensional features but also the amount of simplification has to remain on a similar level.

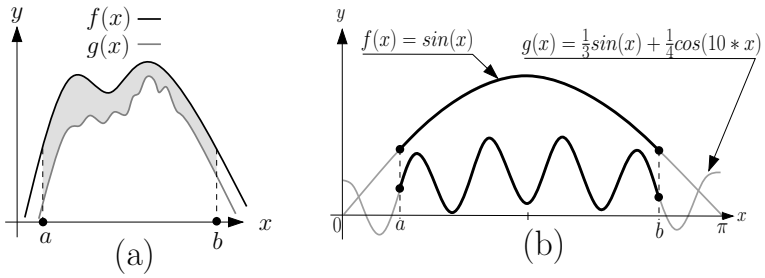

Fig. 3. Area enclosed by $f(x)$ is greater than area enclosed by $g(x)$ (a) the same is not always true in the case of the perimeter (b).

A simplification parameter chosen for driving the simplification has to be both independent of the planar position of a contour line and also an elevation of the contour line. It should also be invariant under horizontal and vertical translations. Another restriction is the complexity of its computation. Cumulation of a terrain model size and the computation complexity of the simplification parameter may seriously affect the performance of the simplification algorithm.
Two main quantitative parameters are related to the size of a closed planar figures. One is the area and the other the perimeter. Both are invariant under translation and rotation and both can be computed in linear time ($O(n)$). As can be seen in Figure 3(a) the area enclosed by an object ($g(x)$) placed inside

another object ($f(x)$) is always smaller. Unfortunately this is not always true in the case of the perimeter. In fact it is quite easy to show a counterexample (Figure 3(b)). However, the situation as presented in Figure 3(b) is very unlikely to occur in the case of contour line data. Additionally the perimeter has been well described in the literature as the parameter for the skeleton based shape simplification. Owing to these reasons, the perimeter is used in the later parts of this study.

3.2 Extraction of the simpler shape from the simplified skeleton

In order to perform the skeleton based shape simplification, the problem of interpolation of the shape of medial circles, centered in the leaves of the simplified version of a skeleton (Figure 2(c)), must be solved. This problem is approached in this study in three different ways:

- detection of the stable vertices on the boundary
- computation of the average vertex
- approximation of the new shape by Bezier splines

The first method is based on the leaves of the simplified skeleton. Every leaf of the simplified skeleton is a Voronoi edge. This Voronoi edge is generated as a dual to some Delaunay edge between two sample points (Figure 4(a)). The two samples are generators of an important skeleton branch, which means that they are also important for the whole shape and may be selected as a part of its representation.

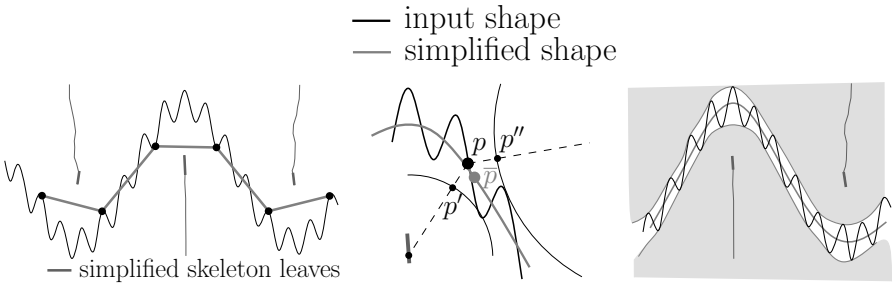

Fig. 4. Stable vertices detection (a), an average vertex computation (b) and approximation by the Bezier splines (c).

The method, based on computation of the average vertex, treats both sides of the curve, being simplified, as two separated entities. Since the algorithm does not make a distinction as to which side is internal or more privileged,

every sample is pruned to circles on two sides of the curve. This causes the sample to take more than one position after simplification (points p' and p'' in Figure 4(b)). Since none of the positions is more privileged, average coordinates of the sample are computed and returned as its position after the simplification (point $\overline{p}$ in Figure 4(b)).

The third method is similar to the computation of the average vertex. It also treats two sides of the curve separately. This results in two shapes bounding possible simplified curves on both booth sides (gray regions in Figure 4(c)). The new, simplified shape can be obtained by creating a curve which is an interpolation between the two bounding shapes.

3.3 Generalization by detection of the stable vertices

The algorithm presented here performs the skeleton retraction on both sides of the input shape. Figure 5(a) shows simplification of a shape for a small pruning threshold. The endpoints of the Delaunay edges ($\overline{s_1 s_2}$, $\overline{s_3 s_4}$, $\overline{s_5 s_6}$) dual to the leaves of the simplified skeleton are not influenced by retraction of the skeleton branches on the opposite side of $B(S)$. The threshold is small enough and does not cause conflicts on the neighboring features. With regard to the selected pruning threshold a new shape is obtained. The new shape is represented by vertices s_i, $i \in \{1 \dots 6\}$ and potentially endpoints of the input, if $B(S)$ is a polyline.

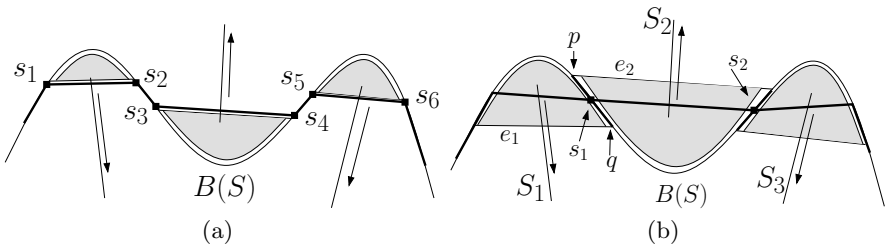

Fig. 5. Small pruning threshold (a), no conflicts occur and a bigger pruning threshold (b) conflicts on boundary occur

A more complicated situation occurs when the pruning threshold is greater than in the previous case. Figure 5(b) shows the situation, when some of the parts of $B(S)$ are pulled in opposite directions by adjacent skeleton branches. In order to obtain simplification of the boundary a vertex in which retraction balances must be found.

Let's consider the Delaunay edges e_1 and e_2 presented in Figure 5. They are dual to the skeleton leaves, being the result of pruning the skeleton branches

Algorithm 3.1: Shape generalization by detection of stable vertices.

Data: Samples ($p_i \in \mathcal{P}$) from a shape ($\mathcal{S}$) , simplification threshold $t \in \mathbb{R}$
Result: Samples from a simpler shape, $p'_i \in P'$

```
for each B̆_k(S) ∈ B̂(S) do
    if B̆_k(S) is polyline then Add endpoints to ImpB(S) and SimB(S);
    for s ∈ B̆_k do
        n ← number of importance markers of s;
        if n > 0 then Add s to ImpB(S);
    end
    DetectStableVertices(ImpB(S));
end
```

s_1 and s_2 placed on the opposite sides of the boundary $B(S)$. If a Delaunay edge $e \in \mathcal{DT}(S)$ and $\mathcal{D}(e) \in \mathcal{S}(\breve{B}_k(S)$,t), e is called an importance marker of its endpoints.
In the case of retraction of S_1 the process can not retract the boundary $B(S)$ further than the boundary vertex q. While in the case of S_2, it will retract the boundary $B(S)$ to at most vertex p. For simplicity let us assume that the point of equilibrium is in the middle of the boundary between p and q. The case when $p = q$ may also occur, but in this case it can be also assumed that this vertex is the point of balance. The pseudocode of the algorithm can be found in Algorithm 3.1 and Algorithm 3.2 sections.

3.4 Generalization by computation of an average vertex

The simplification process based on the simplification of the skeleton can be seen as a retraction of a boundary sample (p) to its parent circle (c_t). The parent circle is a first circle meet during the traversal from a skeleton vertex (C_0) (adjacent to the boundary sample) to the leaf node (C_t) of a simplified skeleton (Figure 6).

On a plane, every non self intersecting curve has two sides. The retraction can process in both directions. Additionally there can be more than one adjacent skeleton branch on each side of the curve. This means that every boundary sample can be retracted to at least two circles. In the case presented in Figure 6(b) the vertex p_i is a sample from a sinusoidal curve. A simplification of the curve for some threshold causes p_i to be retracted to the circles c_{i_1} and c_{i_2}, centered in C_{i_1} and C_{i_2} respectively.
The vertex p_i after a retraction for some threshold to c_{i_1} takes the position denoted as p_{i_1} and after the retraction to c_{i_2} takes position denoted as p_{i_2}. Since there is no additional information about which of the circles is more important or which is internal, both positions are equally privileged. The new position of p_i after simplification is computed by averaging coordinates of p_{i_1} and p_{i_2} and denoted as p'_i.
The simplification threshold in the case of the average vertex method must

Algorithm 3.2: DetectStableVertices()

Data: Points important to a shape $ImpB(S)$
Result: Stable vertices from $ImpB(S)$

```
for each q ∈ ImpB(S) do
    if q is leaf then
        p ← predecessor of q (previous leaf) ;
        n ← number of importance markers of q;
        i_q, i_p ← importance markers of q and q;
        if two of importance markers of q are on opposite sides of B̆_k then
            Add q to SimB(S);
        end
        else
            if i_q and i_p are on the same side of B̆_k ∩ pq ∈ DT(S) then
                Add p and q to SimB(S);
            end
            if i_q and i_p are not on the same side of B̆_k then
                Add ⌊(p + q)/2⌋ to SimB(S);
            end
            p ← q;
        end
    end
end
return SimB(S);
```

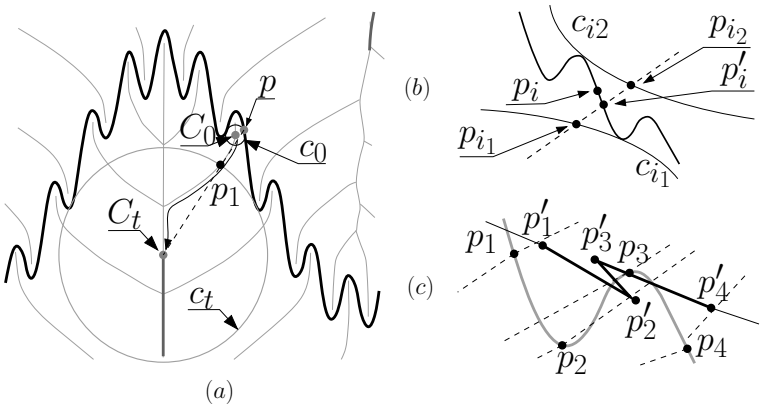

Fig. 6. Average vertex method, point p_i is retracted to circles c_{i1} and c_{i2}

be selected very carefully ownig to a possibility of generation of cracks on a simplified curve (Figure 6(c)). A solution for this problem can be execution of the average vertex algorithm in a few iterations, starting from a very small simplification threshold and gradually increasing it in every iteration. The same method will solve the problem presented in Figure 3(b). The first iter-

ations remove all high frequency components from the longer internal curve and cause very little effect on the shorter, external one.

3.5 Generalization by approximation by the Bezier splines

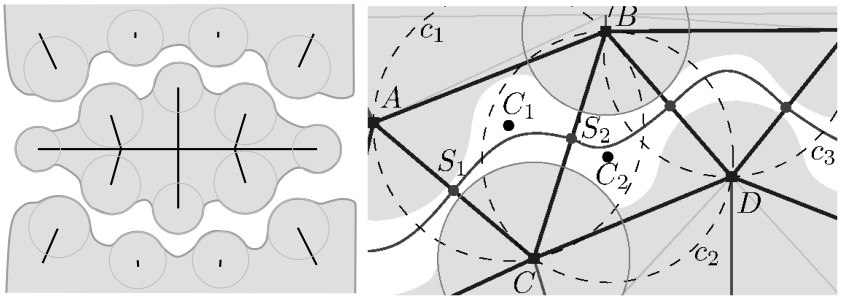

Fig. 7. The Bezier splines method. Simplified skeleton (a) and triangulated. Triangles constructed on different skeleton branches are taken into consideration (eg. ΔABC and ΔBCD) (b) and the splines approximating the new shape are constructed.

As mentioned above, the extraction of a simpler shape can be based on interpolation between two bounding regions (gray areas in Figure 7(a)). The regions are created as the result of independent retraction of the skeleton in two directions. In order to compute the control points for the splines the simplified skeleton is triangulated. Some triangles are constructed on skeleton vertices which belong to the skeleton branches placed on opposite sides of the shape being simplified. In the next step, a spline is constructed passing through points S_1 and S_2 (Figure 7 (b)). The points are placed half way between the boundaries of the medial balls centered in points A and B (point S_1) and B and C (point S_2). The control point for the spline between points S_1 and S_2 is the circumcenter (C_1) of the circle c_1 constructed on the triangle ΔABC. The circumcenter C_2 of the circle c_2 is a control point for the next part of the spline. The circle c_2 is constructed on the triangle ΔBCD.

3.6 Generalization, adaptation of the SVO

According to the definition presented in Li and Openshaw (1992), the smallest visible object (SVO) is equal to a circle with a radius related to the scales of both the target and the source map. In order to adapt it for use with residual functions, a slight modification of the condition used by Li and Openshaw (1992) has been made:

$$\Delta R_H(e) \leqslant \frac{d\pi}{2} \tag{2}$$

where d is the diameter of the SVO, as can be obtained using Equation 1. The meaning of the threshold obtained from Equation 2 is that the algorithm will remove all features for which the boundary length is smaller than half the perimeter of the circle representing SVO.

4 Evaluation of the results

The algorithm has been tested on three datasets: Maple, Jin Island and Layer 6. Each have slightly different characteristics and required some preprocessing before generalization. The maple dataset contains samples taken from the boundary of a maple leaf. The samples were distributed closely enough for the algorithm by Gold and Snoeyink (2001) to extract the shape of the maple leaf as a single closed polygon. For another test, the contour lines of Jin Island in Hong Kong at the scale 1:10 000 were used. Data points from each layer of the input were triangulated on the first pass. On the next pass, each edge of the input was found in the corresponding triangulation and labeled as a part of the crust. All other Delaunay edges were considered as dual to the skeleton edges. The last dataset, Layer 6, was prepared as a set of points from one of the layers of Jin Island data. The crust and skeleton were extracted using the one step crust and skeleton extraction algorithm. Tests were conducted on a Pentium(R) M 1500 MHz laptop with 512 MB of RAM memory. For computation of the Delaunay triangulation, the Linux based CGAL library was used with lazy exact arithmetic.
The Maple leaf dataset was simplified for three different values of the smallest visible object: 5, 10, 25. They were substituted to Equation 2 and yield in following values of a cut off threshold ($\Delta R_H(e)$): 7.85, 15.71, 39.27 receptively. Figure 8 presents the results of the simplification.

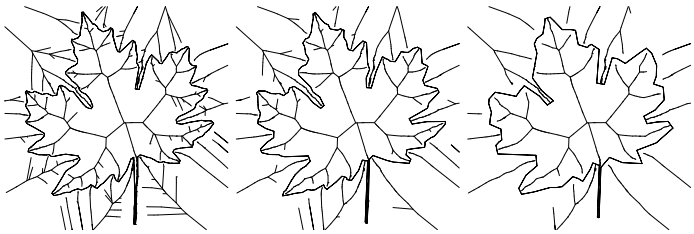

Fig. 8. Maple leaf simplified for SVO=5 (a), simplified for SVO=10 (b), simplified for SVO=25 (c)

The contour lines representing Jin Island in Hong Kong are in the scale 1:10 000. Tests have been conducted for reduction to 1:20 000 and 1:50 000 and 1:100 000, using different values of the SVO in output scale. Figure 9(a) presents the source contour lines of Jin Island at a scale of 1:10 000. Figures 9 (b)... (c) present results of the generalization to scales 1:20 000 and 1:50 000, and the diameter of the SVO in the map scale set to 0.8 mm. As can be noticed in the images for scales 1:20 000 (Figure 9(b) and 1: 50 000 (Figure 9(c) the algorithm behaves quite well and performs fully automatic generalization.

(a) (b) (c)

Fig. 9. Jin Island, a) source data, b) simplified for d=8, c) simplified for d=32.

The preprocessing speed in case of the extraction of the crust and skeleton information, using the one step algorithm, was on average around 1900 points/s. A noticeable drop in performance occurred in the case of extraction of the crust/skeleton information from coverage files (1100 points/s). Extraction of a simplified boundary was performed from more than 1000 vertices per second up to over 10 000 vertices per second.

4.1 Skeleton pruning vs. manual generalization.

Figure 11 shows results of applying the algorithms based on the stable vertex method and average vertex to contour lines of Jin Island given in scale 1:1000. The results are compared to contours simplified manually with the help of the Douglas-Peucker algorithm to scale 1:10 000.

5 Conclusions and future work

The proposed algorithms give good results in reasonable time. Their main advantage is that pruning can be parametrized using different types of parameters. Not only the boundary length, as shown in this paper but also the

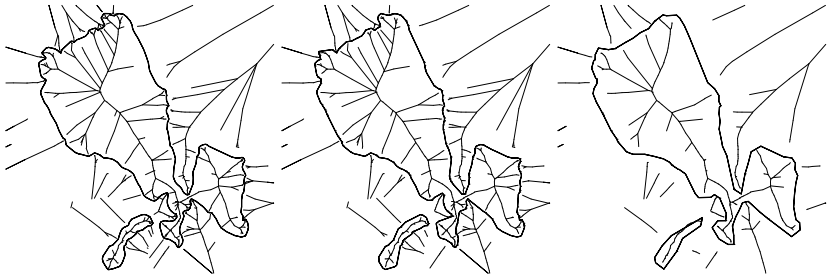

Fig. 10. Layer 6 of the contour lines of Jin Island, a) source data, b) simplified for d=8, c) simplified for d=32.

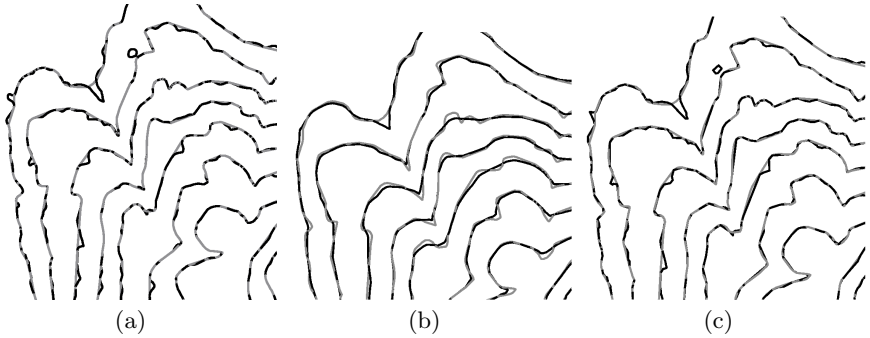

Fig. 11. Simplification of Jin Island (a) average vertex, SVO=4.5 m, (b) average vertex, SVO=45 m, (c) stable vertex, SVO=4.5 m. Gray line represent contour lines generalized manually, and black results of skeleton based generalization.

area covered by features and the parameters presented in work by Ogniewicz et al. may be used. According to tests performed so far the average vertex algorithm does not cause intersections between neighbouring contour lines even when applied with a big simplification threshold. However some cracks may occur when the pruning threshold is too big. In this case simplification should be performed for a few iterations with smaller thresholds, before applying the desired threshold.

Some problems which appear during simplification of bent, featureless objects (Figure 12) using the stable vertex method are the subject of ongoing research. The stable vertex algorithm also does not take into consideration the size of neighbouring features and computes the stable vertex exactly in the middle of two leaf vertices, generated by skeleton branches placed on the opposite sides of the shape. Hopefully this issue can be solved in the future.

Storage of each layer of the contour line model in separate triangulations makes possible parallelization of the process. This should result in better per-

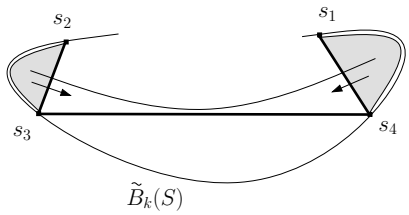

Fig. 12. Simplification of the bent, feature less shape.

formance on multiprocessors machines.

The work presented here is a part of a bigger project. The aim of this project is to remove three dimensional terrain features by simplification of 2D cross sections. All algorithms presented here were in fact designed to achieve this goal. In the future it is hoped to give theoretical guarantees for the algorithms. The guarantees should show that skeleton based simplification prevents contour lines from intersecting one another. Utilization of the area as the parameter driving the generalization process seems to be promising. Future research focuses on utilization of the area as well as on further development of the Bezier splines approximation algorithm.

Acknowledgments

The work described in this paper was fully supported by The Hong Kong Polytechnic University.

Terrain data where obtained thanks to courtesy of the Survey and Mapping Office, Lands Dept., HK SAR Government.

The code was implemented in C++ and uses CGAL library CGAL (2005) for the Delaunay triangulation.

References

Amenta N, Bern M, and Eppstein D (1998) The Crust and the B-Skeleton: Combinatorial Curve Reconstruction. Graphical models and image processing: GMIP, 60(2):125–.

Attali D (1997) r-Regular Shape Reconstruction from Unorganized Points. In *Proceedings of the 13th ACM Symposium on Computational Geometry*, pages 248–253.

Blum H (1967) A Transformation for Extracting New Descriptors of Shape. In *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press.

Brandt JW (1994) Convergence and continuity criteria for discrete approximations of the continuous planar skeleton. CVGIP : Image Understanding, 59(1):116–124.

CGAL (2005) http://www.cgal.org.

Douglas DH and Peucker TK (1973) Algorithms for the reduction of the number of points required to present a digitized line or its caricature. The Canadian Cartographer, 10(2):112–122.

Gold C (1999) Crust and Anticrust: A One Step Boundary and Skeleton Extraction Algorithm. In *SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*, pages 189–196. ACM Press, New York, NY, USA. ISBN 1-58113-068-6.

Gold CM and Snoeyink J (2001) A One-Step Crust and Skeleton Extraction Algorithm. Algorithmica, 30(2):144–163.

Gold CM and Thibault D (2001) Map generalization by skeleton retraction. In *Proc. 20th Int. Cartographic Conf. (ICC 2001)*, pages 2072–2081. International Cartographic Association.

Keates JS (1989) *Cartographic Design and Production*. Longman Scientific and Technical, New York, 2nd edition.

Li Z and Openshaw S (1992) Algorithms for automated line generalization based on a natural principle of objective generalization. International Journal of Geographical Information Systems, 6(5):373–389.

Li Z and Openshaw S (1993) A natural principle for the Objective Generalization of Digital maps. Cartography and Geographic Information Systems, 20(1):19–29.

McMaster RB and Shea KS (1992) *Generalization in Digital Cartography*. Association of American Geographers, Washington, D.C.

Ogniewicz R and Ilg M (1992) Voronoi Skeletons: Theory and Applications. In *Proceedings IEEE Conference Computer Vision and Pattern Recognition, CVPR*, pages 63–69. IEEE Computer Society, Los Alamitos, California. ISBN 0-8186-2855-3.

Ogniewicz R, Szekely G, and Naf M (1993) Medial manifolds and hierarchical description of 2d and 3d objects with applications to MRI data of the human brain. In *Proceedings 8th Scandinavian Conference on Image Analysis*, pages 875–883.

Ogniewicz RL and Kübler O (1995) Hierarchic Voronoi skeletons. Pattern Recognition, 28(3):343–359.

Perkal J (1958) An attempt at objective generalization, Translated by W. Jackowski from Julian Perkal, Proba obiektywnej generalizacji. Geodezja I Kartografia, VII(2):130–142.

Wang Z and Muller JC (1998) Line Generalization Based on Analysis of Shape. Cartography and Geographic Information Systems, 25(1):3–15.