3D TERRAIN SKELETON APPROXIMATION FROM CONTOURS

K. Matuk[†], C.M. Gold[‡], Z. Li[†]

[†]The Department of Land Surveying & Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong.
(krzysiek.matuk,lszlli)@polyu.edu.hk
[‡]School of Computing, University of Glamorgan, Pontypridd, CF37 1DL, Wales, UK.
christophergold@voronoi.com

KEY WORDS: DEM/DTM, Modelling, Reconstruction, Multiresolution, Generalization

ABSTRACT:

In this paper we describe a method for the approximation of the three dimensional skeleton of a terrain model. We based our analysis on the contour line model which lets us easily construct separate two dimensional skeletons for each layer. This approach helps to get rid of construction of the three dimensional Voronoi diagram which is both time and space consuming. The resulting skeleton may find various application like generalization of terrain models or quick collision detection.

1 INTRODUCTION

Representation of the terrain surface plays an important role in many aspects of the modeling. Over the years many different models of the terrain surface have been proposed. From contour line models, through grids to scattered points all of these models attracted the attention of many authors. Recent rapid development of laser scanning techniques focus a great deal of attention on models based on scattered points. However, number of points obtained from laser scanner can be counted in billions, especially in case of the terrain models, which are usually very large.

Analysis and modeling of any big object, since usually memory and speed of the system are limited, makes terrain models the bottleneck of all processes which are using terrain models in any way.

One of the ways to simplify analysis of the objects is to deal with its skeleton. Unfortunately the number of points in the model and a complexity of the three dimensional techniques for extraction of the medial axis from the Voronoi diagram seems to be the main obstacle in achieving this goal.

Instead of computing the full three dimensional skeleton, we decided to try to build its approximation. The process uses as an input contour line terrain model which is widely used in practice and easy available.

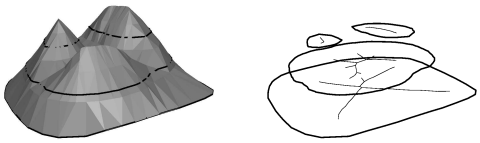


Figure 1: Contours and their skeletons extracted for each layer separately.

The process starts from building the separate, two dimensional Delaunay triangulations for each layer of the model. Next, the medial axis of the each layer is extracted. The layers are then stacked one over another. Finally, all skeleton vertices in all layers, except the lowest one, are visited. When a vertex is visited its nearest neighbor in the layer immediately below is found. Both vertices are then connected by an arc.

As a result, skeletons in all layers and arcs between them form a cyclic graph which represents the approximation of the three dimensional skeleton of the input terrain model.

2 BACKGROUND

To build the approximation of the three dimensional skeleton we had to employ various techniques. We start by constructing triangulation of each layer and extracting the skeleton from it using the one step crust and skeleton extraction algorithm designed by Gold (1999). In the next step a potential residual is computed according to the work of Ogniewicz and Kübler (1995). At the same time closed polygons are detected and skeleton parts marked as internal or external. Finally we employ a variation of the Li-Openshaw (Li and Openshaw, 1992) algorithm for line generalization to extract the most important parts of the skeletons.

2.1 One step crust and skeleton extraction

One of the simplest algorithms for extraction of the skeleton from samples placed along two dimensional curve has been proposed by Gold and Snoeyink in Gold (1999) and Gold and Snoeyink (2001).

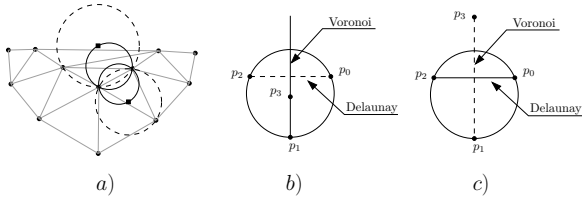


Figure 2: One step crust and skeleton extraction.

Crust and skeleton extraction relies on the fact, that the Delaunay edge is a part of the crust if a circle build on both samples forming Delaunay edge (Figure 2, $\overline{p_0p_2}$) and one of the vertices of the dual to Voronoi edge $\overline{p_1p_3}$ does not contain other Voronoi vertex.

2.2 Potential residual functions

A central place in skeleton analysis seems to be occupied by the works of Ogniewicz et al. In Ogniewicz et al. (1993), Ogniewicz and Kübler (1995) and Ogniewicz and Ilg (1992), they presented the theory and some of the numerous applications for skeletons of two dimensional objects. Their work presents methods for the extraction of potential functions, such as:

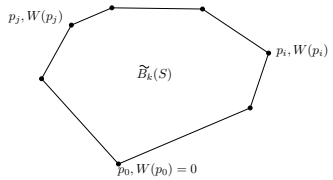


Figure 3: Boundary potential function.

- potential residual ΔR_P
- circularity residual ΔR_C
- bi-circularity residual ΔR_B, and
- chord residual ΔR_H

In our approach we utilize only the potential residual ΔR_P, because its value is dependent only on the size of a feature described by a particular skeleton edge, not the importance of the feature for the shape. Other residual functions are more suitable for extraction of the important parts of the skeleton (see (Ogniewicz, 1992)), but in our future work we would like to use scale depended factors for generalization of the contour lines, which generates a need for preservation of true values of boundary distances.

The potential residual for a skeleton edge dual to a Delaunay edge between two points: p_i and p_j, $(p_i, p_j \subset \widetilde{B_k}(S)$ can be computed using following formula:

$$W_k(p_i, p_j) = \begin{cases} \min(|W(p_i) - W(p_j)|, & \widetilde{B_k}(S) \\ \Sigma_w - |W(p_i) - W(p_j)|) & closed \\ \\ |W(p_i) - W(p_j)| & \widetilde{B_k}(S) \\ & non - closed \end{cases} \quad (1)$$

In equation 1 the value of $\sum_w$equals the length of the whole boundary of a closed polygon, while $W(p_i)$ and $W(p_j)$ are the potentials in the vertices p_i and p_j, respectively.

The value of the residual potential can be computed by first computing the potential in each vertex, with respect to some vertex for which the potential, without lose of generality, can be set to zero (Figure 3, p_0).

2.3 Shape reconstruction from slices

Closely related to the contour lines in terrain modeling is reconstruction of shapes from two dimensional cross sectional images. This is a very intensively studied topic in visualization of human organs using computer tomography. In works by Boissonnat (1988) and Bajaj et al. (1996), detailed descriptions of the techniques utilized for shape reconstruction from planar cross sections are found.

The techniques presented in those works as well as an approach described by Geiger (1993) will help us to find adjacency relationships between skeleton branches in neighboring layers.

3 CONTOUR LINES GENERALIZATION

Contour lines are the most wide used model for representation of terrain models in Cartography and GIS. In parallel with the rapid development of computer systems, they were more and more utilized for storing and processing geographical data.

The natural need for reduction of those data resulted in many algorithms for reduction of the complexity of the contour lines. The description of some of the approaches presented by different authors can be found in the works of McMaster et al. (McMaster (1989),McMaster (1987), McMaster and Shea (1992)) as well as other authors: (Douglas and Peucker. (1973), Gold and Thibault (2001), Li and Sui (2000)).

As can be seen in later parts of this paper, we will apply a slightly modified method presented by Li et al. in Li and Openshaw (1992) and Li and Openshaw (1993). The method is known as the Li-Openshaw algorithm and is based on the definition of the smallest visible object (SVO).

The smallest visible object is given by equation:

$$SVO = S_t * D * (1 - \frac{S_f}{S_t}) \quad (2)$$

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

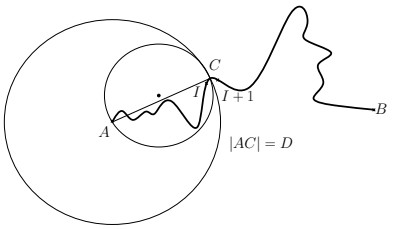


Figure 4: Li-Openshaw algorithm for line generalization.

The main idea of this algorithm relies on the removal of some of the features which are invisible at the desired scale of the map.

To simplify a curve from A to B in Figure 4 by using this algorithm the following steps must be performed:

1. Estimate the SVO by using equation 2.
2. Determine the starting point of the algorithm (point A on Figure 4).
3. Compute the mean value of the coordinates of A and intersection of the circle with radius D equal to SVO, and the center at A with the curve being simplified (C). This can be done by using following two equations:

$$(X - X_A)^2 + (Y - Y_A)^2 = SVO^2$$

$$\frac{Y - Y_I}{Y_{I+1} - Y_I} = \frac{X - X_I}{X_{I+1} - X_I}$$

4. Advance start point to intersection point C and repeat step 3 until point B is reached.

This algorithm works in the vector mode, however according to the suggestions given by the authors in (Li and Openshaw, 1992) the best results can be obtained using the raster-vector approach.

4 EXTRACTION OF THE CRUST AND SKELETON INFORMATION FROM SOURCE DATA.

The process starts from the creation of the set of 2D Delaunay triangulations, one for each layer and dependent on the z coordinate (or elevation) of the samples. This approach opens a possibility for parallel processing of the data which will result in improvement of the overall processing performance on multiprocessor machines. However, parallelization of the algorithm is out of the scope of this paper, and will not be explored deeper here.

Separating the triangulation of the each layer lets us to treat it as a set of points, distributed along the planar curves. Unfortunately at this moment the triangulation does not contain any information, neither which Delaunay edge forms an edge of the contour line nor which Voronoi edge is a part of skeleton.

To extract skeleton edges we will utilize a technique developed by Snoeyink and Gold which uses fact that that a Delaunay edge is a part of the crust or its dual is a part of the skeleton.

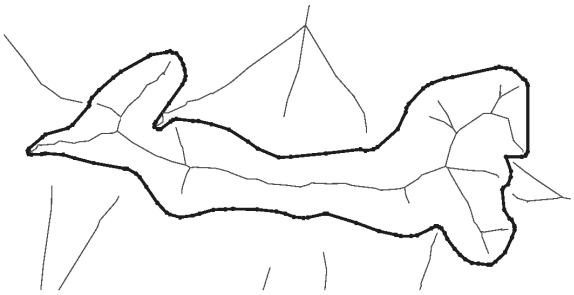


Figure 5: One step crust and skeleton extraction.

Our approach relies on analysis of the skeleton. To distinguish which edges are parts of the skeleton we will assign a flag for each Delaunay edge. The flag set to TRUE, will indicate when the dual, to the Delaunay edge, is a part of skeleton. The opposite value of the flag, will mean that this particular Delaunay edge is an element of the crust. In our case of course this is equal to assigning the Delaunay edge to a contour line.

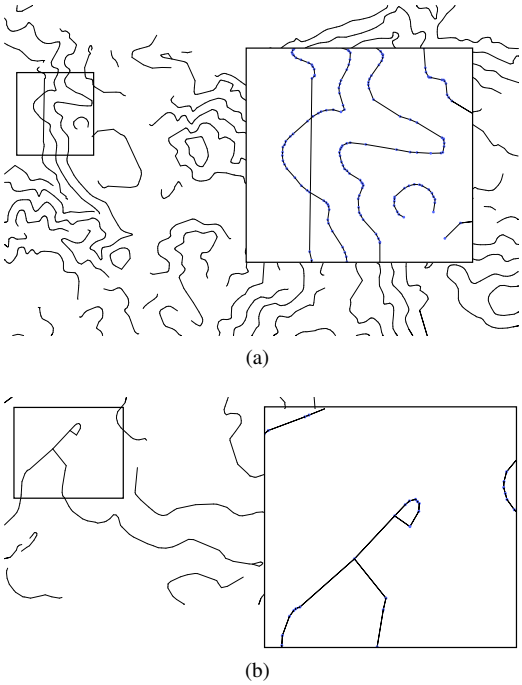
Labelling the Delaunay edges has been performed in two ways. As the criteria for the selection of the approach we used properties of the input data.

At first *.xyz files were used. The coordinates of the points were extracted from ArcInfo coverage files. The straightforward solution in this case seems to be the utilization of the one step crust and skeleton extraction algorithm.

Source coverage files were also used. They contain information about point coordinates, as well as the polygons which those points form. This information was later used for proper labeling of the Delaunay edges as a part of the contour lines.

The first of the approaches relies on the sampling density, which in the case of real data is often not good enough. Besides, unsupervised labelling of the Delaunay edges may sometimes cause edges to be labelled as crust while they actually do not belong to any of the contour lines. At the same time some edges which actually are part of a contour line were labelled as a skeleton (Figure 6).

To get rid of this problem, the results of crust/skeleton extraction were postprocessed. We performed some simple statistical analysis of the length of the crust edges in every layer and eliminated outliers from the set by labeling all the edges for which



(a)

(b)

Figure 6: Wrong crust edges due to undersampling: intersections (a) and loops (b).

the length was greater than some threshold as skeleton edges. In order to make this process easily configurable standard statistical technique for outliers detection has been used (Tabachnick and Fidell, 1996).

The threshold was selected experimentally using the following formula:

$$l > \bar{l} + n * \sigma \tag{3}$$

where:

$\bar{l}$- mean length of the crust edge
n - integer value selected experimentally
σ - standard deviation of the edge's length

In our case possible outliers will all be greater than the mean length of the crust edges. Due to this only positive values of n should be taken into consideration. At the same time n can not be too small in order to prevent disconnection of proper crust edges. An experimentally selected value of n equal to 6 seems to be a reasonable choice. This value caused all too long crust edges to be marked as skeleton. At the same time some of the edges which actually should keep their label were also changed, however the number of disconnected proper crust edges remain at an acceptable level. The distribution of the crust edges lengths before and after the outlier elimination can be observed in Figure 7.

Unfortunately in the areas of high curvature and poor sampling density, some unwanted edges, labelled as a crust, still remain. They result in loops in the middle of the contour line or joins between neighboring lines. To remove those errors, we compute the number of the crust edges going out from every vertex of the triangulation. If the vertex has more than two outgoing crust edges, one of the edges is disconnected. In our data it has never happened that there were more than three outgoing crust edges. The edge to disconnect was chosen as that which creates the largest angle with the remaining two edges (Figure 6b)).

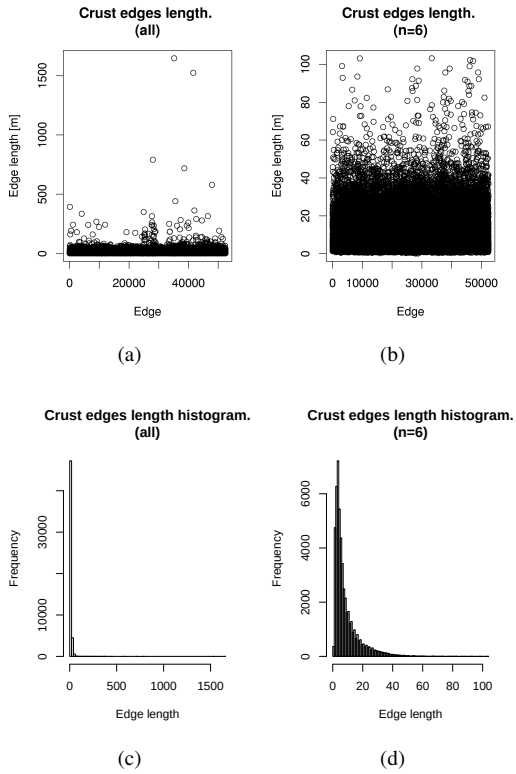


Figure 7: Distributions of the crust edges after extraction (a) and (c) and after removal of crust edges for n=6 (b) and (d)

Another way to obtain the necessarily information can be the usage of the advantages of vector data. Source data were obtained as ArcInfo coverage files, which already contain information about polygons. By using this information for each edge of the polygon the corresponding edge of the Delaunay triangulation was found and labelled as the crust edge.

Unfortunately, any effort to improve quality of data was insufficient to correct all errors present in real data. This forced us to set following assumptions:

1. Source data can contain only closed polygons
2. Open polygons are allowed only on the boundary of the dataset.
3. A closed polygon can not contain any other polygons inside, whether it is closed or not.

The first two assumptions protect us from situations when errors in the data cause skeleton edges to intersect the boundary of the polygon. In the future we will try to find a solution which will be able to handle all problematic cases.

The third assumption, was set to avoid the presence of background skeletons inside the closed polygons.

4.1 Classification of the skeleton edges

After extraction of the contour and skeleton information from the source data, each layer should contain polygonal chains forming open or closed polygons. For the sake of simplicity we assume that there are no open polygons except on the boundary of the data. Skeleton edges present in each layer can be divided into three categories:

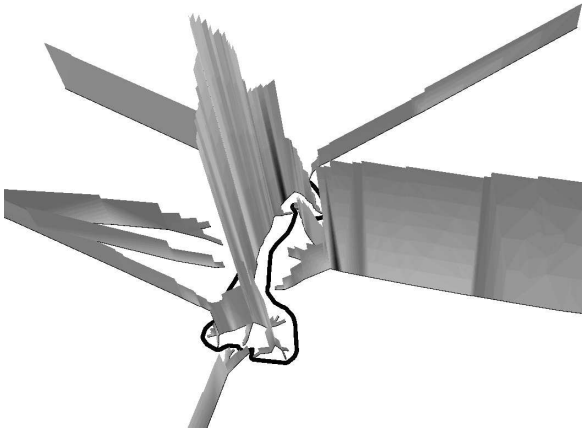


Figure 8: Perspective view of the potential residual.

- internal, placed inside closed polygons
- external, placed outside closed polygons, but dual to the Delaunay edges joining samples from the same polygon
- background, dual to Delaunay edges joining samples from two different polygons

The distinction between background skeleton edges and those belonging to a polygon can be easily done by checking the polygon ID number assigned to samples in the previous step. Another distinction which has to be made is separation of the internal skeleton parts from the external ones.

Since information about residual values is already present in our data we decided to employ it to made this distinction. As can be seen in Figure 8 the largest value of the residual has a skeleton edge placed inside the polygon.

4.2 Simplification of the skeleton

By their nature skeletons contain in their structures information about every, even the smallest perturbation present at the boundary. Sometimes even a perturbation related to a very small precision error, present in sample coordinates, may generate a long "hair" in the medial axis. To avoid the presence of the unwanted branches in our output we decided to simplify the structure of the skeleton by retracting those branches, which have residual function value smaller than some threshold.

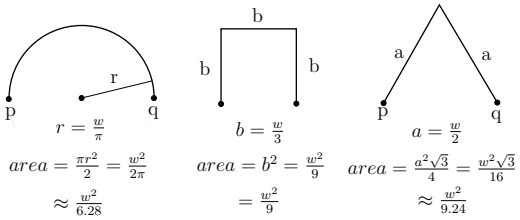


Figure 9: Area covered by features of different shape.

Visibility of an object in different scales is related to the area covered by it. The bigger area covered the smaller scale in which an object will be visible. In order to do so, all features having a perimeter smaller than the threshold feature with the largest area (for a fixed perimeter) must be found (Figure 9). For a fixed value of a perimeter (or the boundary length) different shapes will cover different areas. In (Ogniewicz, 1992) three basic shapes are analyzed in terms of boundary length: circle, square and uniform triangle. Here we follow similar path and for a given boundary

length w we want to evaluate which shape of a feature will have the biggest area.

It can be proven (see Weisstein (2005)) that out of all shapes, the circle is the one which encloses the greatest area for a given value of the perimeter. The same result can be observed in Figure 9 where areas covered by the basic shapes are presented.

To estimate the value of the threshold we used the natural principle as described in (Li and Openshaw, 1992) and (Li and Openshaw, 1993). 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. To adapt it to be used together with the residual functions we suggest using a slightly modified version of the natural principle. To eliminate skeleton branches which define features "invisible" at some particular scale we used following threshold during generalization:

$$\Delta R_H(e) \leqslant \frac{d\pi}{2} \tag{4}$$

where d is the diameter of the SVO as can be obtained from equation 2. Geometrically this can be interpreted as a half arc from p_i to p_j (Figure 10).

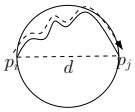


Figure 10: Adaptation of the natural principle to simplification of the skeleton.

We assumed that the smallest object which can be drawn at a map scale has diameter of $0.5 - 0.7$ mm (parameter D in equation 2), and using a source data scale which is $1 : 10\,000$ we attempt to simplify the skeleton for a few different values of the output scale. The results of elimination of the invisible skeleton branches using the threshold obtained by using equation 4.2 can be seen in Figure 11.

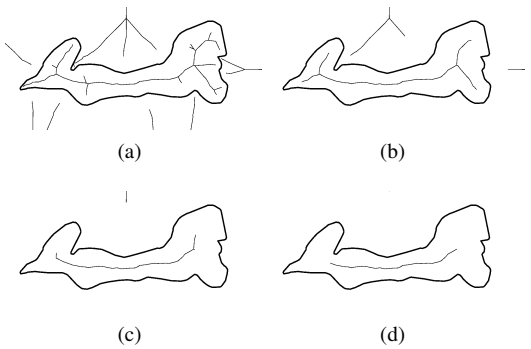


Figure 11: Simplification of the skeleton for different scales: a) 1:20 000, b) 1:50 000, c) 1:100 000, d) 1:200 000.

5 BUILDING ADJACENCY RELATIONSHIP BETWEEN LAYERS.

This step of our construction process is similar to algorithms used in reconstruction of the shapes from computer tomography. However in contrast to approach presented eg. in (Geiger, 1993), our approach simply visits all of the skeleton vertices in each layer. For each vertex, its nearest neighbor in the layer right below is found and connected by the edge to this vertex. Obviously the lowest layer is skipped in this step.

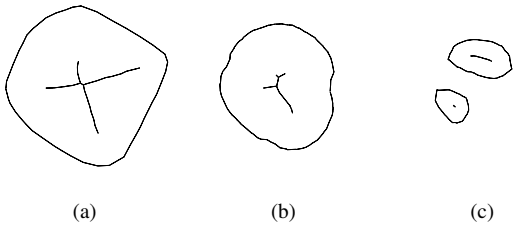


Figure 12: Separate layers and their skeletons.

In Figure 12 three separate layers of the simple terrain model are presented. Skeletons were simplified to make images clearer. Stacked contours can be seen in Figure 13 (gray lines denote joins between layers).

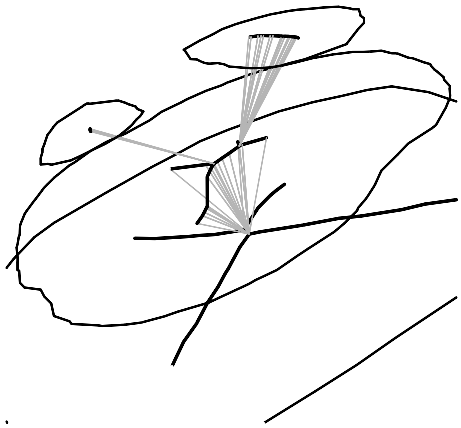


Figure 13: Skeletons in separate layers joined together.

6 CONCLUSIONS AND FUTURE WORK

The resulting skeleton approximation can now be used for generalization of the contour lines to prevent intersections of the contour lines in different layers. In this case, the graph which approximates the skeleton can be used as a search structure. During the simplification of one layer of the source data, neighboring layers can be checked for possible intersections. Since search (in neighboring layers) can be performed starting from a triangles nearest to the currently processed vertex and the triangles are available in constant time $O(1)$, so presumably queries checking for intersections cam be performed very effectively. Another application could be, after some postprocessing, a quick collision detection, based on the simplified skeleton approximation.

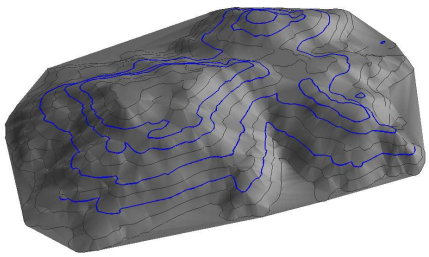
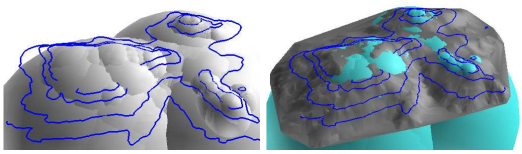


Figure 14: Terrain model with contour lines superimposed.

To detect collision with terrain, a skeleton vertex can be used with some weight, representing the radius of a sphere, attached.

Some preliminary results are presented in Figure 15a. The source terrain model from Figure 14 is approximated by spheres. The spheres were constructed by using three vertices (a, b, c) of a triangle from one layer (L_i) and a vertex nearest to the circumcenter of the circle constructed on triangle Δabc in the layer placed right over L_i. In Figure 15b terrain model superimposed with its spherical approximation can be observed. The skeleton was simplified before construcing the spheres.

Contour lines of each layer can be processed separately, which in the case of our test data allowed processing without need for the development of an out-of-core solution. At the same time in the case of smaller objects (fewer points in layer) a few layers can be loaded into memory and processed at the same time.



(a) (b)

Figure 15: Spheres built for each layer.

In the future, except for the simpler skeleton, we would like also to obtain a simpler shape of the boundary polygon, which eventually will lead us to a skeleton based algorithm for contour line generalization.

Unfortunately, our assumption about closeness of the polygons will have to be modified in future research, due to the presence in real data, of open polygonal chains.

Acknowledgements

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.

REFERENCES

Bajaj, C. L., Coyle, E. J., and Lin, K.-N., 1996. Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Processing*, 58(6):pp. 524–543. ISSN 1077-3169. doi:http://dx.doi.org/10.1006/gmip.1996.0044.

Boissonnat, J.-D., 1988. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing*, 44(1):pp. 1–29. ISSN 0734-189X.

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

Geiger, B., 1993. Three-dimensional modeling of human organs and its application to diagnosis and surgical planning. Technical Report 2105, Institut National de Recherche en Informatique et Automatique, INRIA Sophia Antipolis, France.

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*, pp. 189–196. ACM Press, New York, NY, USA. ISBN 1-58113-068-6.

Gold, C. M. and Snoeyink, J., 2001. A one-step crust and skeleton extraction algorithm. *Algorithmica*, 30(2):pp. 144–163.

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

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):pp. 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):pp. 19–29.

Li, Z. and Sui, H., 2000. An integrated technique for automated generalization of contour maps. *The Cartographic Journal*, 37(1):pp. 29–37.

McMaster, R. B., 1987. Automated line generalization. *Cartographica*, 24(2):pp. 74–111.

McMaster, R. B., 1989. Numerical generalization in cartography. *Cartographica*, 26(1). Monograph 40.

McMaster, R. B. and Shea, K. S., 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*, pp. 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*, pp. 875–883.

Ogniewicz, R. L., 1992. *Discrete Voronoi Skeletons*. Ph.D. thesis, Swiss Federal Institute of Technology.

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

Tabachnick, B. G. and Fidell, L. S., 1996. *Using multivariate statistics*. HarperCollins College Publishers (New York, NY), Cambridge, MA, USA. ISBN 0673994147.

Weisstein, E. W., 2005. Isoperimetric problem. *From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/IsoperimetricProblem.html*.