Mesh Simplification for Building Typification

Dirk Burghardt and Alessandro Cecconi
Department of Geography, University of Zurich
Winterthurerstr. 190, 8057 Zurich, Switzerland
burg@geo.unizh.ch, acecconi@geo.unizh.ch

Abstract. This paper describes a deterministic approach for the *typification* of buildings which uses several levels of details for the derivation of intermediate scales. The typification procedure is modeled as a two-tier process, with the steps *positioning* and *representation*. While the first step determines the number and the position of the building objects with respect to the requested scale, the representation step is used to calculate size and orientation for the replacement buildings. The positioning is performed using a mesh optimization technique adapted from the area of computer graphics. The approach was originally developed for surface reconstruction and mesh simplification. With help of one additonal parameter the building density could be controlled, e.g. it is possible to emphasize areas with high building density.

Introduction – Motivation

In the context of webmapping automated generalization plays an important role for the adaption of cartographic presentation on the user specifications. So far two different approaches are discussed, on one side the on-the-fly creation of cartographic presentations from one base data (process-oriented approach) and on the other side the use of pregenerated, independent maps (representation-oriented approach) (Weibel 1997). While the on-the-fly production is not completly realizable due the efficiency of generalization operators in the foreseeable future, the use of pregenerated cartographic presentations are not flexible enough for webmapping. An alternative is the combination of the above-mentioned approaches, through the use of a multiscale database. They are used to manage several digital cartographic models with different levels of detail and allow the creation of adapted cartographic presentations from them through automated generalization. Following advantages resulting from this, so-called derivation-oriented approach (Cecconi 2003):

- The generalization process will be accelerated.
- Missing generalization opertors could be compensated with access to several levels of detail.
- Any adjustments on the user specification are possible concerning scale, content, etc..

In this paper a method for the *typification* of the feature class building is presented, which uses two levels of detail (LOD_{25} and LOD_{200} [1], see also Figure 1) for the derivation of intermediate scales.

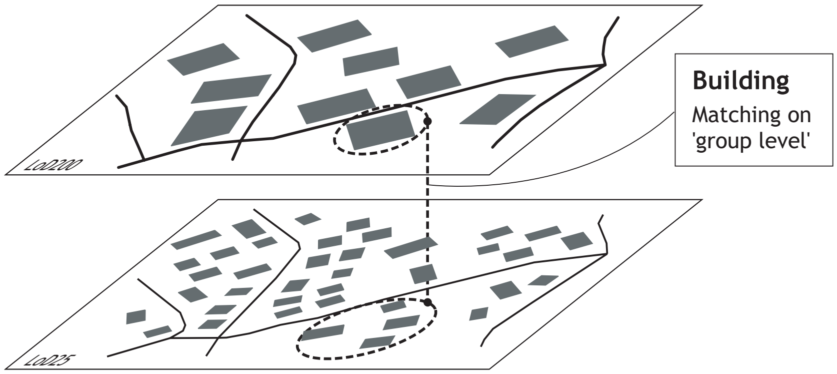


Figure 1: Two level of detail (LOD_{25} and LOD_{200}) are used for the typification of the feature class building. Data: VECTOR25/200 © swisstopo (BA034957).

It is also shown how the matching process between individual objects and groups could be realized to speed up the generalization operations. If there are existing links between objects of different levels of detail or if they could be created, the administration and updating process would be simplifyed too.

The operator *typification* is used for the transformation of an initial set of objects into a subset, while maintaining the distribution characteristics and pattern of the original set. Sester and Brenner (2000) proposed an approach for typification of 2D-structures of similar type and size (e.g. buildings). The method is based on Kohonen Feature Maps, a neural network learning technique. The prominent property of this unsupervised learning method is the fact that the neurons are adapted to a new situation, while keeping their spatial ordering. The approach is non determenistic as a result of random selection of neurons at the beginning, which means after rerun the algorithm different results will be achieved. We suggest a determenistic approach for the typification of the feature class building with adaption of a method from computer graphics, so-called mesh optimization.

Theory of mesh optimization

The mesh simplification technique is related to *mesh optimization techniques* well known in the research area of computer vision which hold great potential for many applications. These kinds of techniques have been mainly used for surface reconstruction from sampled data, occurring in many scientific and engineering domains.

---

[1] VECTOR25 and VECTOR200 define the data models for the scale of 1:25'000 and 1:200'000, provided by the Swiss Federal Office of Topography.

The main reason for using a mesh optimization technique is to reduce the amount of data. Thereby three goals or purposes can be defined:

i) faster rendering,
ii) reduced storage volume, and
iii) simpler manipulation.

A lot of research work has taken place in this topic. Mesh optimization, as considered by Turk (1992) and Schroeder et al. (1992), refers to the problem of reducing the number of faces in dense meshes (usually made up of triangles). The contribution of this early work was the development of a method for smoothly interpolating between models representing the same object at different levels of detail. Two main disadvantages, however, impede using this approach in our work. First, the method is best suited for models that represent curved surfaces. Secondly, the required algorithm is very complex and time-consuming and thus not adapted for the context of on-demand web mapping.

The research work of Hoppe et al. (1993) has shown that mesh optimization can be put into use effectively in at least two applications: surface reconstruction from unorganized points, and mesh simplification (the reduction of the number of vertices in an initially dense mesh of triangles). Their principal idea is to describe mesh simplification as an optimization problem, defining an energy function E that directly measures deviation of the final mesh from the original. To solve the optimization problem they minimize the energy function E that captures the competing objectives of a tight geometric fit and a compact representation. One of the main disadvantages concerning this approach is the very time-consuming computation of the energy minimizing function E. Based on the method of Turk (1992) and Hoppe et al. (1993) for mesh optimization, an adapted, less time-consuming energy minimizing function has been developed for on-demand web mapping. The problem considered in here can be stated as follows:

„Given a collection of data points X in $\mathbf{R}^2$ and an initial triangulation mesh M_0, a mesh M_f with a smaller number of vertices is sought that fits the original data well.“

Figure 2. illustrates the main idea of reducing the amount of vertices of a mesh (*mesh simplification*) for multiple representations preserving the original characteristic vertex distribution. On the left hand side of the Figure, a dense mesh made up of a large number of vertices describes the front of a „face“. In the middle part, the amount of vertices is reduced maintaining the typical outline and characteristic of the original form. On the right hand side a strongly simplified version, with a fraction of the original vertices of the starting data set, is depicted.

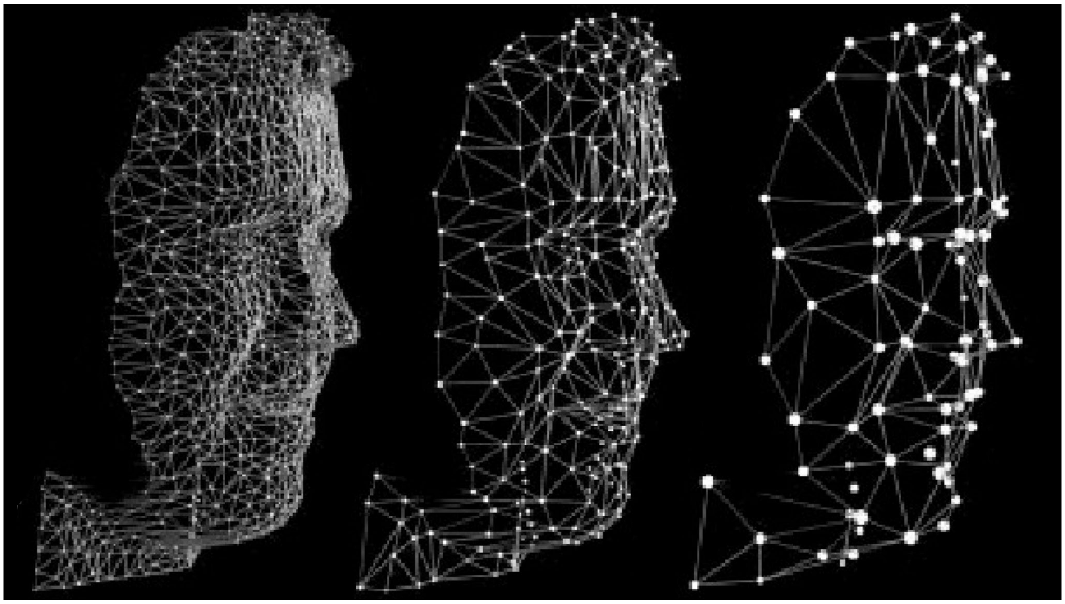


Figure 2: Example of reducing the number of vertices of a mesh (from Dudziak (2000))

Each state best represents the original form. Thus the selection of points plays a crucial role and must be optimized to preserve the original arrangement. Regions with a dense number of vertices should be maintained in each state or level as dense areas and vice versa with sparse parts. This is well illustrated in Figure 2. Transferring this idea to cartography and particularly for building representation, the various states of the „face“ in Figure 2 can be looked upon as different scales or LODs of a map. The vertices do thereby not display the buildings themselves but, for example, the centers of gravity of building objects.

Starting from a mesh M_0 with n vertices (representing the centers of gravity of the individual buildings) a new mesh M_j with (n-1) vertices is sought. This new mesh should best represent the original one with minor changes. Hoppe et al. (1993) define a mesh M as a pair (K,V), where: K is a simplicial complex representing the connectivity of the vertices, edges and faces; $V = \{v_1,...,v_m\}$, $v_i \in \mathbf{R}^3$ is a set of vertex positions defining the shape of the mesh. To obtain a mesh that provides a good fit to the original point set X an energy function $E(K,V)$ is defined where:

$$E(K,V) = E_{dist}(K,V) + E_{rep}(K) + E_{spring}(K,V)$$

By varying number, position and connectivity of the vertices a minimization of this value is looked for. The distance E_{dist} is equal to the sum of squared distances from the point set X of the mesh. The value E_{rep} is proportional to the number of vertices. E_{spring} is looked upon as a regularizing term and describes the sum of the edge lengths. Using this energy function E for a mesh optimization in principle provides the possibility of observing several constraints which restrict the generalization process (such as the distance between the objects, building alignment, etc.). But since, on the one hand, the minimization of the energy function *E(K,V)* is computationally very time-consuming (Puppo and Scopigno 1997) and on the other, only *one* constraint (distance between the objects) should be considered here, a simpler function, more adapted for on-demand web mapping has been derived.

Mesh simplification adapted for typification

Before explaining the various steps of the proposed mesh simplification technique in more detail, the full approach of typification should be discussed. The typification procedure, as shown in Figure 3, is composed of two steps which are not independent and interact with each other:

- **Position**: Determining the *number* and the *position* of the new objects with respect to the requested scale;
- **Representation**: Creation of a new building objects at the determined position.

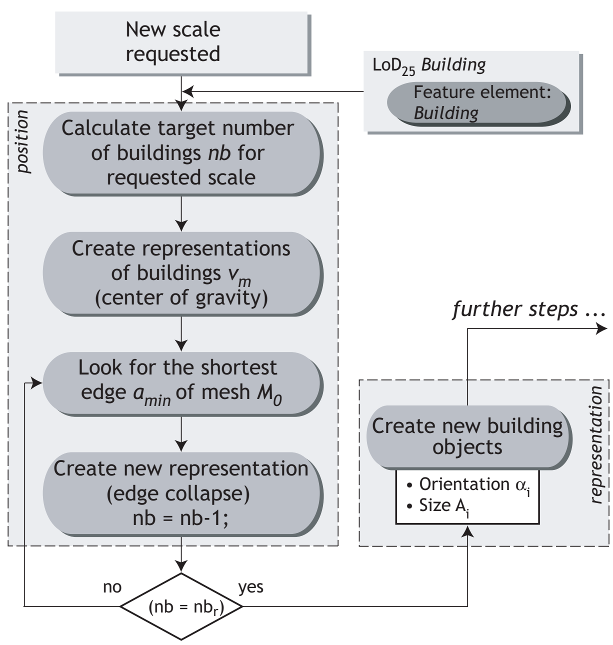


Figure 3: Flowchart for the typification operator of the generalization process for the feature class building, carried out by the mesh simplification technique. The highlighted part on the left computes the position of the new placeholders, while the right part defines the representation.

The typification process (mesh simplification) is an iterative process where the termination criterion is dependent on the requested map scale m_r. By means of the original number of objects nb_o and the scale value m_r, an adapted number of building objects $nb_r(nb_o, m_r)$ is computed for terminating the iteration. The basic idea is, that two objects which lie next to each other can be replaced by a new one between them, called a representative (placeholder). This simple operation is known as **edge collapsing** (Hoppe 1996) and is sufficient for effectively simplifying meshes. As shown in Figure 4, an edge collapse transformation unifies two adjacent vertices v_s and v_t into a new single vertex v_n.

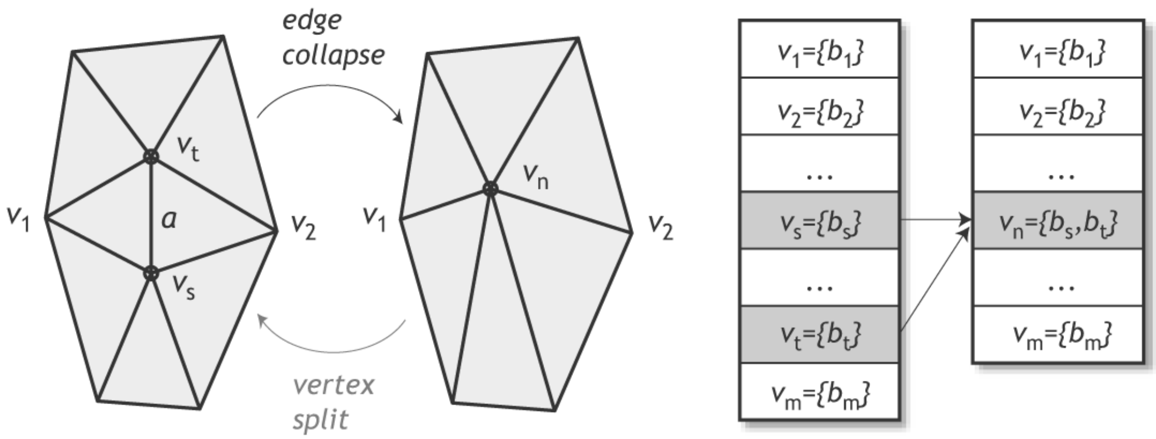


Figure 4: The method of edge collapsing for mesh simplification (left). On the right a sequence of edge collapses is shown. v_i describes the center of gravity of building b_i.

The inverse transformation vertex split adds a new vertex v_t and thus two new faces to the original mesh. Since typification must reduce the number of objects, the inverse case is not relevant here. Transferring the idea of edge collapsing to the feature class building, where each vertex represents a building object, helps to solve the first step of the typification process - *position*.

For the second step *representation*, each remaining vertex v of the final mesh M_f must know which building or buildings it represents. For example, in Figure 4 the placeholder v_n represents the vertices v_s and v_t and thus the building objects b_s and b_t. From the geometric information of the represented objects (e.g. area A, orientation α) a new best fit building object (i.e. a placeholder) must be created for the requested map scale m_r.

The main phases of the mesh simplification technique for the feature class building are illustrated in Figure 5. It explains the first step of the typification process (position).

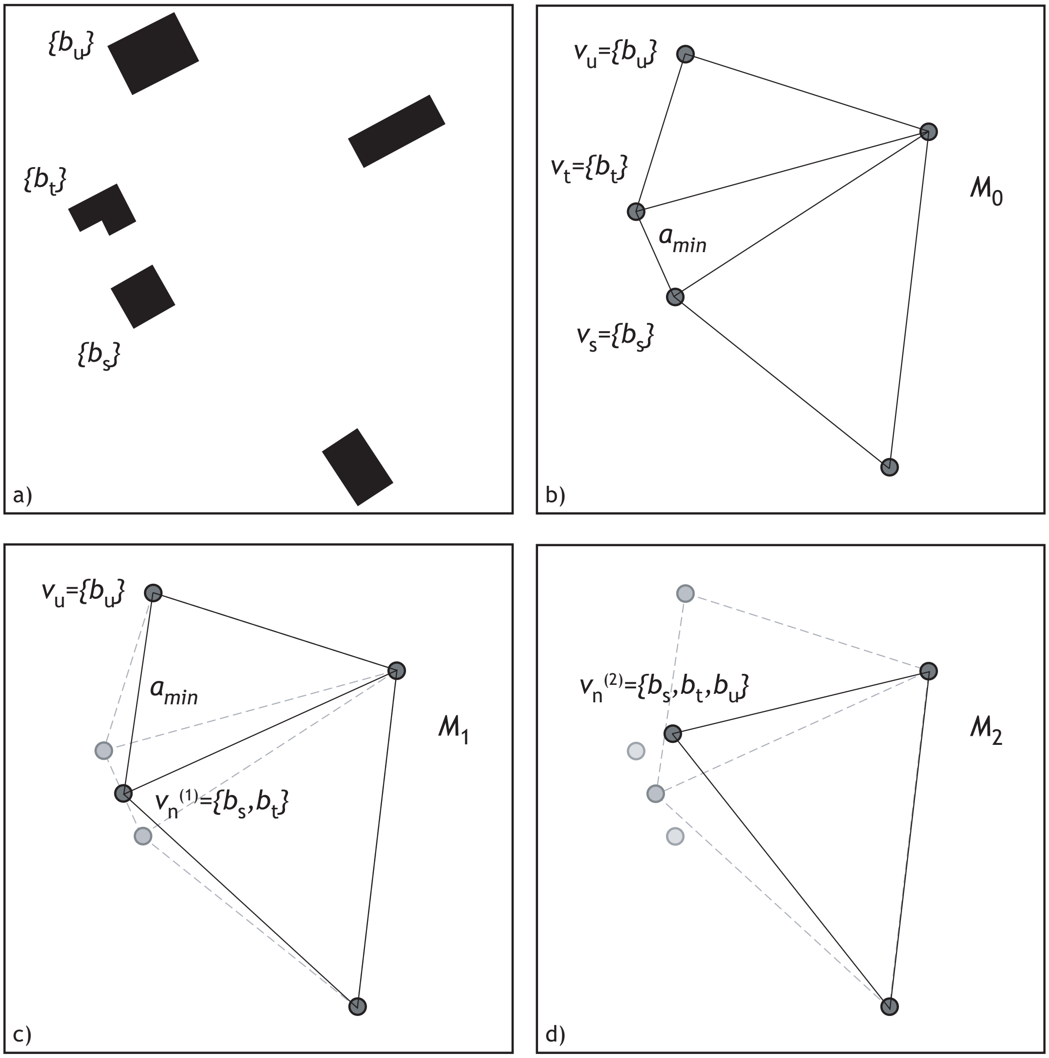


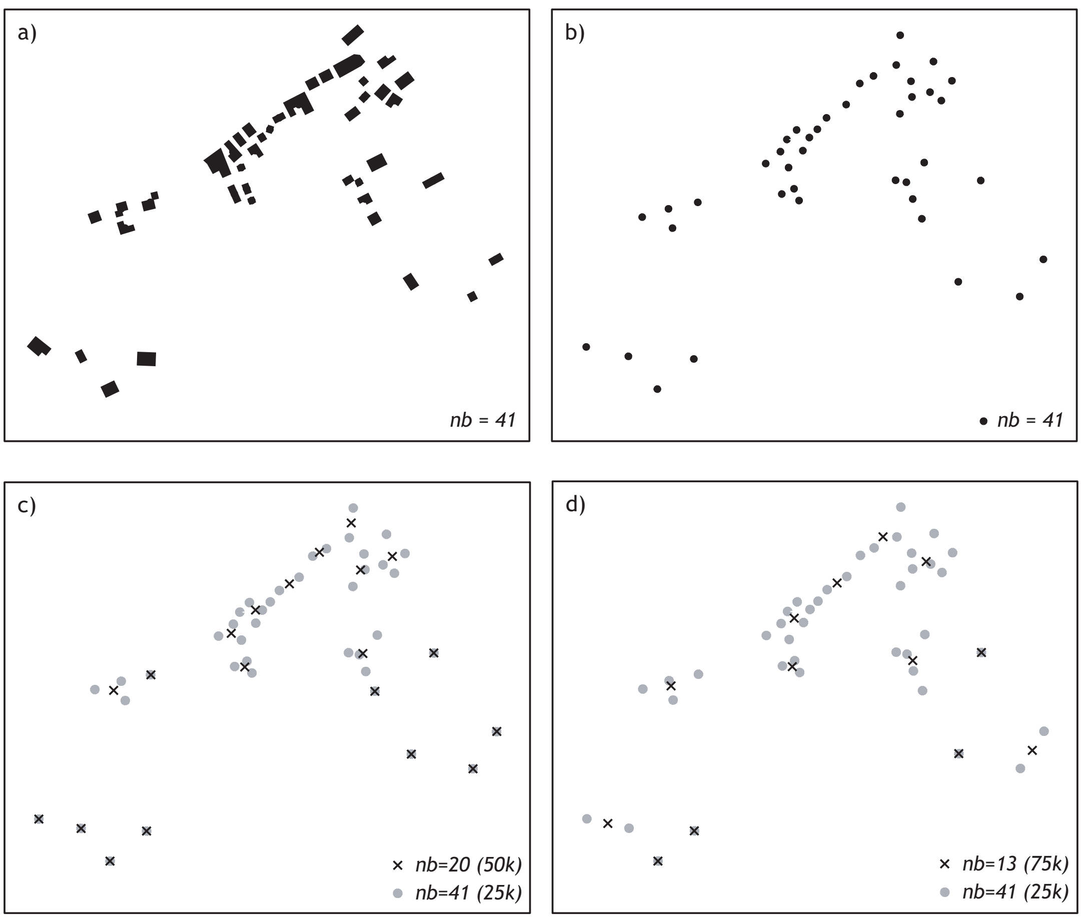
Figure 5: Phases of the mesh simplification technique for the step *position*

Let X be a set of points with vertices $v_1,..., v_m$ representing the centers of gravity of the building objects $b_1,..., b_m$ in LOD_{25} and M_0 a mesh over this point set. In a first iteration loop a new mesh M_1 is sought, where two vertices v_s and v_t are replaced by a new one $v_n^{(1)}$ as illustrated in Figure 5c). The criterion for the selection of these two vertices is that they describe the shortest edge a_{min} of the mesh M_0. The new vertex $v_n^{(1)}$ is defined as center of a_{min} and thus lies between v_s and v_t. By searching the shortest edge and replacing v_s and v_t through $v_n^{(1)}$ the modifications take place locally and thus the main characteristics of the mesh will not be disturbed. For the calculation of the position of the new vertex $v_n^{(1)}$ all affected vertices (and thus buildings) must be considered: v_s, v_t and v_u. Hence, the position of $v_n^{(2)}$ is fixed by the center of gravity of the vertices which are replaced (show in Figure 5d)). The iteration is continued until the current number of vertices is smaller than the value $nb_r(nb_{25},m_r)$.

There are several ways to calculate number of objects for the target map scale. One approach is to keep the „black-white“ ratio between buildings and background constant for all scales. The number nb_r results from the summed area of buildings in LOD_{25} divided by the average of the building size at the target scale. Another way is to use Töpfer's radical law (Töpfer and Pillewizer 1966):

$$nb_r = nb_{25}\sqrt{\frac{m_{25}}{m_r}}$$

After termination this iteration process each remaining vertex represents one or more of the original buildings. Besides the amount of the created objects also their positions [*x,y*] plays an important role. The new placeholders must best portray the original nature of each group. In the following figures the results computed with the mesh simplification technique are illustrated. A comparison of original objects and created placeholders illustrates the obtained results.



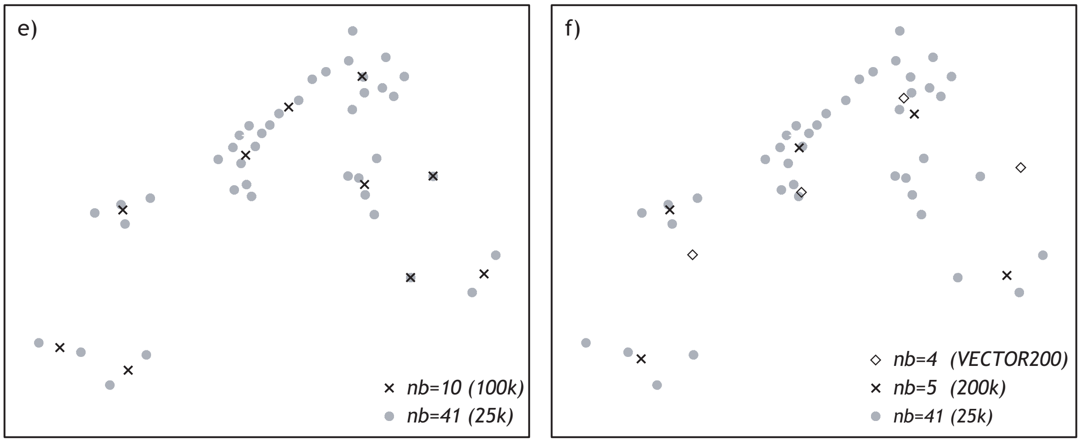


Figure 6: The positions of the placeholders for the following scales: a) buildings in LOD_{25}, b) 1:25'000 (original data set), c) 1:50'000, d) 1:75'000, e) 1:100'000 and f) 1:200'000 with the objects' center of gravity in LOD_{200} (all not to scale). Data: VECTOR25/200 © swisstopo (BA034957).

Figure 6a) shows the building objects of LOD_{25} (VECTOR25) with *nb*=41 elements. In Figure 6b) only the centers of gravity representing the original data set are displayed. These points compose the vertices of the mesh M_0 as discussed in the previous section and define the starting point of the mesh simplification technique. With the iterative process of *edge collapsing* a number of vertices are removed or replaced by new ones. Figure 6c) shows the result for the requested scale of 1:50'000 displayed as black points whereby the amount is decreased to *nb*=20. In the background the vertices depicted in gray describe the vertices of M_0. In some cases vertices coincide and only the black ones (1:50'000) are visible. The placeholders for scale 1:75'000 are displayed in Figure 6d) where the number decreases to *nb*=13. In Figure 6e) the mesh is composed of only *nb*=10 vertices. Finally, in Figure 6f) a comparison with the positions of the vertices of LOD_{200} can be done. In contrast to LOD_{200} the computed solution results in 5 objects, whereby three of them are describing a similar position. Evaluating the created placeholders for each scale with the original data set of LOD_{25} it can be noticed that reasonable results have been achieved maintaining the main characteristics of the group through all scales.

Control of building density

Since our approach is based on looking for the vertices describing the shortest distance (shortest triangle edge), areas with high building density would be thinned out more strongly than sparsely populated regions. On the one hand this could be desirable to avoid too many objects in urban areas, but on the other hand it would distort the characteristics of an area. To meet this requirement a correction factor f_a is included in the calculation of the distance. This factor f_a allows to elongate the real edge length a between the vertices and thus to decrease the thinning process in densely populated areas. It depends on the number of objects r that a vertex represents (for example the value for v_n^2 is r=3) as well as the term s_u which must be set by the user in advance meeting the following conditions:

$$0 \le s_u \le 1 \quad \rightarrow \quad \begin{array}{l} s_u = 0.0: \text{no correction is done} \\ s_u = 1.0: \text{correction by means of the original number of represented objects} \end{array}$$

From these two values the factor f_a can be defined as follows

$$f_a = s_u \cdot (r - 1) + 1$$

As described above the shortest edge a_{min} of all edges a_i ($\forall$i) of the mesh M_0 is looked for:

$$a_{min} = \min a, \quad \text{where} \quad a = f_a \cdot \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2}$$

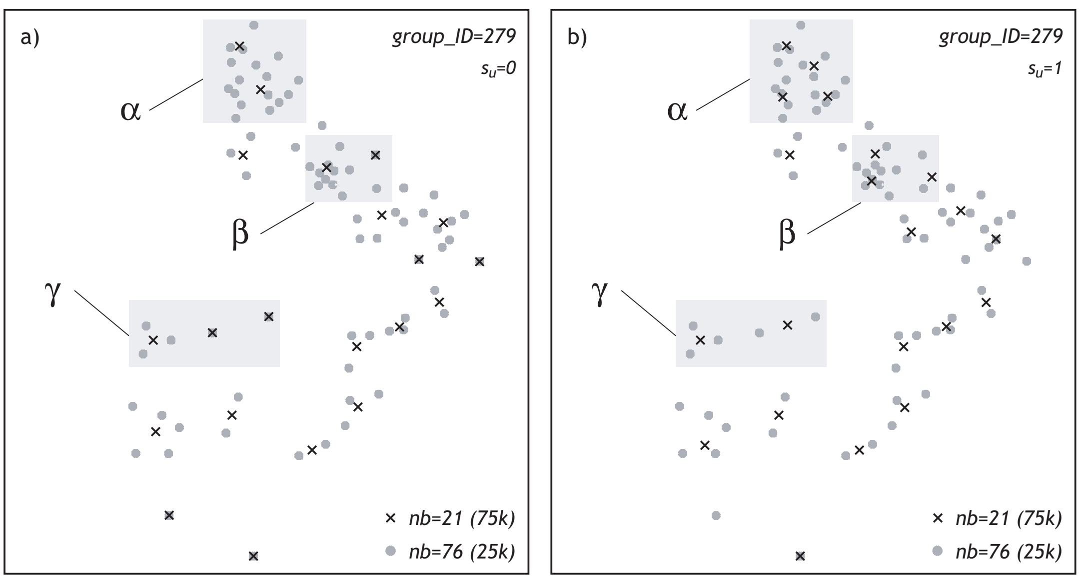


Figure 7: The positions of the placeholders with a) s_u = 0.0 and b) s_u = 1.0 (both not to scale). Data: VECTOR25 © swisstopo (BA034957).

The examples discussed so far have been computed with the factor s_u = 0.0 which implies that no correction has been done concerning the distance between the vertices. To incorporate this quality the value s_u has been introduced to decrease the thinning process in densely populated areas. The difference between computing a representation with s_u = 0.0 or s_u = 1.0, respectively is displayed in Figure 7 a) and b) for the scale 1:75'000 (s_u may range between 0.0 (no correction) and 1.0 (maximum correction)).

For both results the number of objects to be displayed for the requested scale is equal (*nb*=21). What changes are the positions of the centers of gravity of the placeholders. On the left side (Figure 7 a)) the highlighted area α is represented by only two buildings while on the right part of the figure (b)) four objects are displayed within the same area. Comparing these two areas reflects the effect of the factor s_u: computing the result with a high value for s_u ($s_u \rightarrow 1.0$) takes the density of the building objects more into account than with low values ($s_u \rightarrow 0.0$). The same effect can be found in the second highlighted example β where places with a high density of buildings are represented better in b) than in a). Working with low values for s_u ($s_u \rightarrow 0.0$) prefer single seated objects as illustrated in the highlighted area γ.

Shape construction of replacement buildings

The next step of the typification process (*representation*) creates for each vertex a building object from the ones it represents. For example in Figure 5, for the vertex $v_n^{(2)}$ the vertices v_s, v_t and v_u determine the shape of the new building object $b_n^{(2)}$.

Thereby the two attributes

- *area A* [*width, length*]
- *orientation* α

of each represented object are considered. The selected approach is based on the idea that the newly created building object $b_n^{(2)}$ should on the one hand best represent the largest object of the group and on the other hand also attempt to maintain the characteristics of the whole group.

Area A

Comparing the map series of the Swiss National Mapping Agency it can be stated that most buildings are strongly simplified at a scale of 1:100'000 and smaller and thus less detailed. The shapes are usually represented as rectangles to meet the minimum separability distance as defined in Spiess (1990). In the context of on-demand web mapping where the minimum separability distances are more severe (owing to the coarse display resolution) the depiction of buildings should be kept very simple. The concept here is also to define each new object as rectangle, whereby the area $\overline{A_n}$ is computed out of the average of the pertaining buildings $\sum A_k / nb$ in LOD_{25}. As a consequence of the necessity to keep the minimal dimensions of buildings a scaling factor f_{area} is used:

$$\overline{A_n} = \frac{\sum_k A_k}{nb} \cdot f_{area} = \frac{\sum_k A_k}{nb} \cdot \underbrace{\frac{m_r^2}{m_{25}^2} \cdot \left(1.0 - \Psi \cdot \frac{m_r - m_{25}}{m_{200} - m_{25}}\right)}_{f_{area}}$$

The first term m_r^2 / m_{25}^2 of the scaling factor compensates the reduction in size while the scale changes. Without the second part the buildings size would be always like in LOD_{25} independent from scale. The second part is used to consider the size of buildings in LOD_{200}. Their influence increases when the requested scale is closer to LOD_{200}. Parameter Ψ has to be calculated from the ratio between original building size $\overline{A_{LOD200}}$ and calculated building size $\overline{A_n}$ for LOD_{200}:

$$\Psi = 1.0 - \frac{m_{25}^2}{m_{200}^2} \cdot \frac{\overline{A_{LOD200}}}{\overline{A_n}}$$

Out of this corrected value $\overline{A_n}$ the *width* and *length* of the new object can be obtained. The ratio *width* / *length* of the new building must be the same as for the largest represented object in LOD_{25}.

Orientation α

For the orientation α of the new building only the value of the largest object of the group is considered. The reason for this approach is that the orientation of the largest building can be assumed to be most representative for the represented objects and influence or even dominate the characteristics of its environment. Hence, the orientation α of the new and the largest object are equal.

Figure 8 displays the result of typification computed for different scales whereby the examples b) - f) have been scaled to the same size as a).

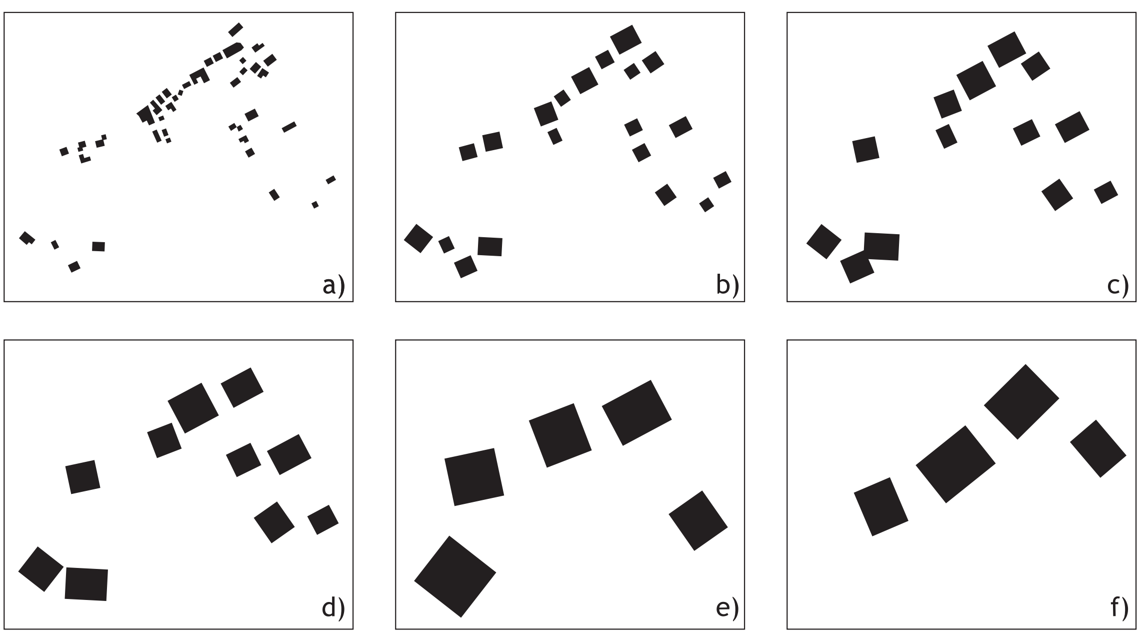


Figure 8: The positions and dimensions of the placeholders for different scales: a) 1:25'000 (representing LOD_{25}), b) 1:50'000, c) 1:75'000, d) 1:100'000, e) 1:200'000 and f) 1:200'000 (representing LOD_{200}) (all at scale 1:25'000).
Data: VECTOR25/200 © swisstopo (BA034957).

Figure 8a) shows the original data set of LOD_{25} with *nb*=41 objects. For scale 1:50'000 (Figure 8b)) the amount of objects decreases to *nb*=20. The dimension [*width,length*] of each object depends on the dimensions of the represented objects, while the orientation α is fixed by the same orientation as the largest building. In Figure 8c) *nb*=13 objects are represented for scale 1:75'000. As can be seen overlap problems arise in the lower part of the figure. As one possible solution the elimination of one object can be taken into consideration. Figure 8d) shows *nb*=10 objects for scale 1:100'000. The last two Figure 8e) and f) display the buildings objects for scale 1:200'000: e) the computed objects and f) the data set of LOD_{200}. Comparing the size of the buildings at scale 1:200'000 a similarity can be assessed. Note that for all examples shown the constraints of minimal distance between individual objects has not been taken in account. All illustrations are depicted at 1:25'000. Figure 9 shows the same situation, but now at the corresponding target scale.

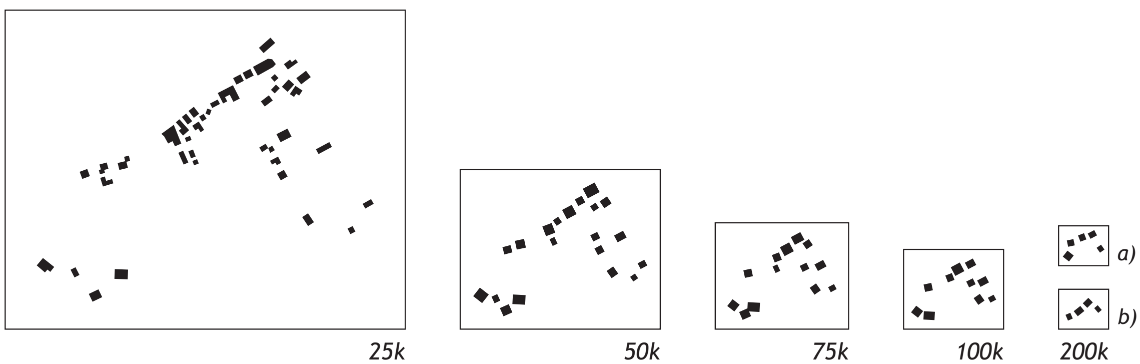


Figure 9: Typification of buildings computed for different scales with correct proportions. For scale 1:200'000 the computed (a)) and the original (b)) representations are shown. Data: VECTOR25/200 © swisstopo (BA034957).

Figure 10 gives a complete example with the maps from LOD_{25} and LOD_{200} for comparison as well as the calculated position of placeholders and the resulting map for scale 1:75'000.

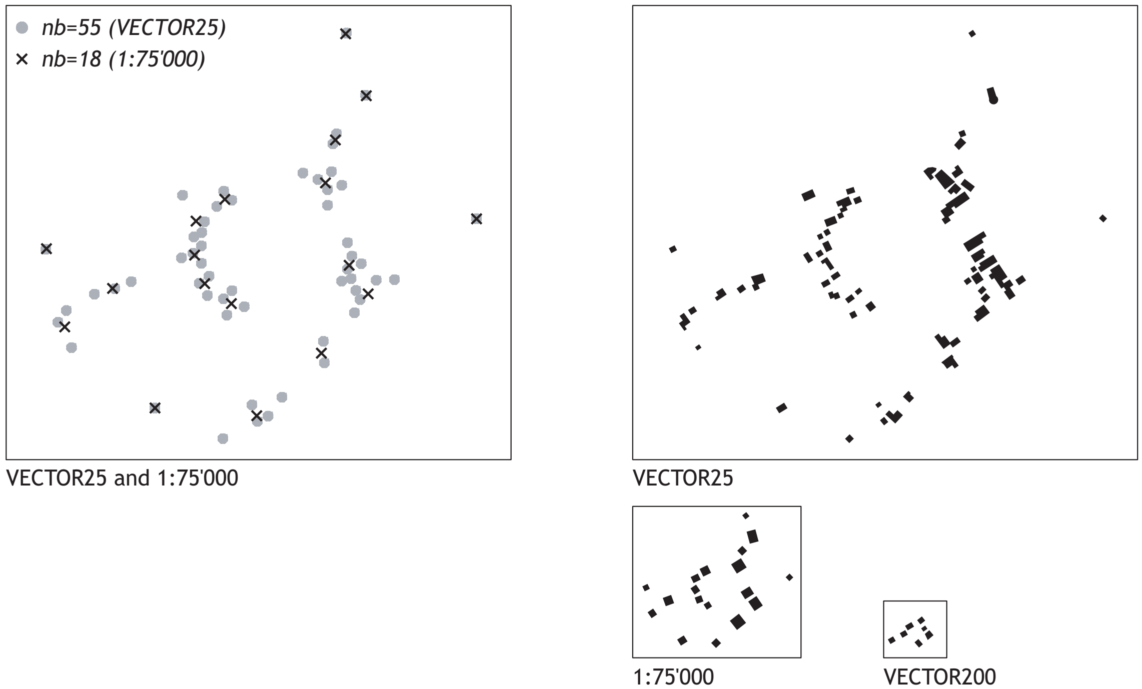


Figure 10: Left: The positions of the placeholders for scale 1:75'000 (not to scale). Right: The dimensions of the placeholder computed out for scale 1:75'000.

Limitations and possible improvements

For both parts of the approach limitations can be found. They may be summarized in the following four points:

- In the present implementation the mesh simplification technique fails to maintain alignments of buildings. Such particular patterns could be preserved by introducing additional constraints to the method. Constraints could be generated by an off-line preprocessing step that detects buildings alignments (for instance, using the method proposed by Christophe and Ruas (2002)) and observed in the algorithm by setting increased weights for buildings in alignment structures.

- The positions of the placeholder are only dependent on the center of gravities of the objects it represents. In some cases, especially when representing large or important buildings, a combined geometric and semantic-based edge collapsing could be of advantage to keep these special building objects. The selected approach of mesh simplification must be enhanced by the possibility of using additional parameters (e.g. importance of the building objects).

- The computed area of a placeholder depends on the areas of all represented buildings and is derived from the average of the area of all included objects. Even if a placeholder represents buildings with extremely different sizes the average size is displayed. This can influence the local nature of these few objects. A similar problem arises for example if one large-sized building and several small ones are used to compute the area of a placeholder. In this case the size of the new object will be too small compared to the large building. This situation leads to an unfortunate depiction of the placeholder. To better determine the size of a placeholder a statistical evaluation of the represented building objects can be accomplished.

- The minimum separability distances are taken into account only indirectly by the two given states LOD_{25} and LOD_{200}. For the computed placeholders this constraint is not respected in the current implementation. A solution in context with on-demand web mapping could be to downsize the concerned building objects and thus gain space between them in order to avoid overlaps and congestion. An example of overlaps created by excessive building sizes is given in Figure 11.

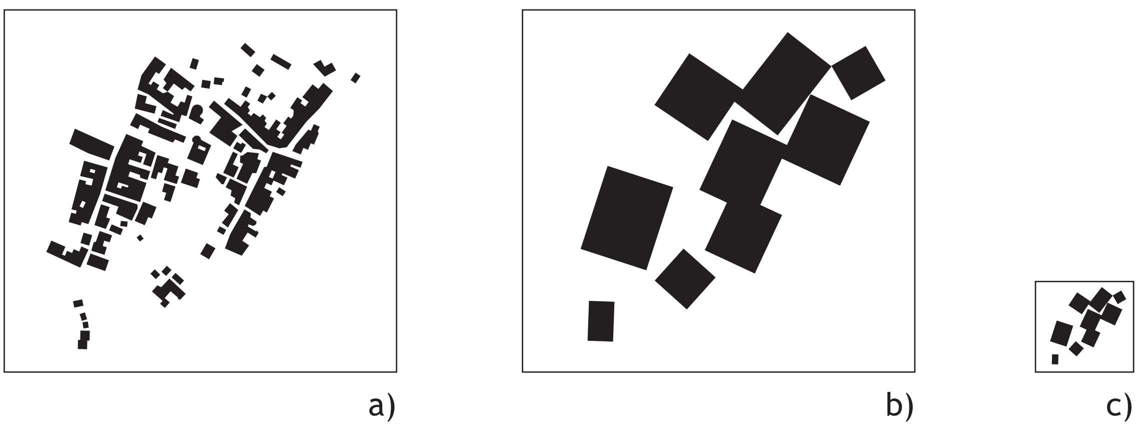


Figure 11: Example for computed objects that are disproportionate for the requested scale. Figure a) LOD_{25} with 57 objects, b) result computed for scale 1:100'000 (at scale 1:25'000), and c) computed result at scale 1:100'000 (9 objects). Data: VECTOR25 © swisstopo (BA034957).

Figure 11a) shows the original data set (LOD_{25}) for the feature class building. Figure 11b) and c) display the computed representation with the mesh simplification technique for the scale 1:100'000. Besides the overlap problem the size of the objects is not appropriate for the requested scale. The problem is that very large buildings in LOD_{25} define the basis for computation of the building sizes for the placeholders at 1:100'000. For that reason some objects are about eight times the size required by the constraints defining the minimum perceptual limits Spiess (1990). In this case the size of the objects can be reduced to meet the requirements mentioned. A comparison can be done with Figure 11d), where the dimensions of the objects are well proportioned.

With the possibility of defining the correction factor s_u in the range of [0.0,1.0] the user can influence the kind of map he/she wants to have. The differences between the representations concerning the positions of the building objects for ($s_u \rightarrow 0.0$) or ($s_u \rightarrow 1.0$) are significant. The advantage of using this method of mesh simplification is that any scale can be generated out of the two border data sets LOD_{25} and LOD_{200}.

Literature

Boffet, A. and Rocca Serra, S. (2001): *Identification of Spatial Structures within Urban Blocks for Town Characterisation.* In: Proceeding of the 20th ICA/ACI Conference, Beijing, pp. 1974-1983.

Brazile, F. (2001): *Semantic Infrastructure and Methods to Support Quality Evaluation in Cartographic Generalization.* Ph.D. thesis, Department of Geography, University of Zurich.

Cecconi, A. (2003): *Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping.* Ph.D. thesis, Department of Geography, University of Zurich.

Christophe, S. and Ruas, A. (2002): *Detecting Building Alignments for Generalisation Purposes.* In: Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. (1993): *Mesh Optimization.* Computer Graphics, 27 (Annual Conference Series), pp. 19-26.

Hoppe, H. (1996): *Progressive Meshes.* Computer Graphics, 30 (Annual Conference Series), pp. 99-108.

Regnauld, N. (1996): *Recoginition of Building Clusters for Generalization.* In: Proceeding of the 7th International Symposium on Spatial Data Handling, Delft, pp. 4B.1-4B.14.

Saalfeld, A. (1987): *It Doesn't Make me Nearly as CROSS.* Some Advantages of the Point-Vector Representation of Line Segments in Automated Cartography. International Journal of Geographical Information Systems, 1 (4), pp. 379-386.

Schroeder, W. J., Zarge, J. A., and Lorensen, W. E. (1992): *Decimation of triangle meshes.* Computer Graphics, 26 (2), pp. 65-70.

Sester, M. and Brenner, C. (2000): *Typification Based on Kohonen Feature Nets.* In: Proceedings of the 1st International Conference on GIScience, Savannah (GA), pp. 21-22.

Spiess, E. (1990): *Siedlungsgeneralisierung.* In: Kartographische Publikationsreihe, Volume 10. Schweizerische Gesellschaft für Kartographie, Zürich.

Töpfer, F. and Pillewizer, W. (1966): *The Principles of Selection: a Means of Cartographic Generalization.* The Cartographic Journal, 3 (1), pp. 10-16.

Turk, G. (1992): *Re-Tiling Polygonal Surfaces.* Computer Graphics, 26 (2), pp. 55-64.

Weibel, R. (1997): *Generalization of Spatial Data: Principles and Selected Algorithms.* In: M. van Kreveld, J. Nievergelt, T. Roos, and P. Widmayer (editors), Algorithmic Foundations of Geographic Information Systems, Lectures Notes in Computer Science, Springer, Berlin, pp. 99-152.