Construction of the planar partition postal code map based on cadastral registration

Friso Penninga, Edward Verbree, Wilko Quak*, Peter van Oosterom
Delft University of Technology
Department of Geodesy
Jaffalaan 9
2628 BX DELFT
The Netherlands
+31 15 2783756
{E.Verbree, C.W.Quak, P.J.M.vanOosterom}@CiTG.TUDelft.NL
*Corresponding author

ABSTRACT

Accurate postal code maps could play an important role within GIS as the postal code has the potential to link the address description of buildings and their location in a certain global reference system. This relationship is possible in both directions: address matching and geocoding. These operators demand a certain mechanism in translating an exact geometric position (i.e. WGS84 coordinate) into a location indication (town, street, house number) and vice versa. As most built-up parcels are provided with a postal code this indicator could be used as the linkage. This paper describes the procedure, based on the Dutch cadastral registration, to obtain a reliable 6-position planar postal code map for the Netherlands. Problems with existing postal code maps, like intersecting of houses and arbitrary derived boundaries are avoided.

For a planar coverage, non built-up parcels having no assigned postal code should be assigned a plausible postal code. Therefore special attention is given to infrastructural parcels. These parcels are divided at their skeleton first and then piecewise attached to their neighbor parcels. This new approach results in very reliable postal code maps, which are visually attractive too as infrastructure lines can be regognized. The procedure is generic and can be applied to other administrative parcel information as well.

The algorithm is implemented using the Computational Geometry Algorithms Library (CGAL), and the possibilities and limitations of this library are addressed as well. The reliability of the derived planar postal code map is discussed and some results are shown by figures. A short overview of alternatives and improvements concludes this paper.

Categories and Subject Descriptors

H.4.m [**Information System Applications**]: Miscellaneous.

General Terms

Algorithms.

Keywords

GIS, Triangulation, Skeletonization

1. INTRODUCTION

Postal code maps could be used within GIS as they can link address descriptions of houses and offices to their location in a geometric reference system. This link is bidirectional: address matching is used to get a certain address specified by a coordinate and geocoding is used to allocate a coordinate to a given address. Current address matching and geocoding computations have however some limitations.

Most address matching algorithms use linear interpolation between two direct connected nodes of a street network. The nodes link to the addresses and given a regular distribution of houses along the street one can expect that the house with a specified address is at a certain distance between the two nodes. But a more complex allotment demands a more demanding algorithm.

As in most other countries the postal code is introduced in the Netherlands for efficiency in postal services. By demanding a uniform description of identification of houses and offices a more automatic dispatching of letters could be developed. The Dutch postal code consists of four digits and two characters, i.e. 2629JA. This so-called 6-position postal code is attached to an average of 16 neighboring postal addresses, or part of a street. By the combination of a postal code plus a house number each house or office can be identified uniquely. The Department of Geodesy at Delft, the Netherlands, is identified by 2629BX9. The postal code could be accordingly used as an address identifier.

The position of the postal address, and thus the postal code, is not obvious. The 'center' of a house identified by the postal address could be considered as the location of the postal code, but this derived point location of the postal code will be only a 'best guess'. That is because the coordinates are mostly arbitrarily allocated or calculated (i.e. the centroid) within the boundary of the building. And how to construct a 6-position planar postal code map out of several postal address locations? When this point location is used to assign a kind of neighborhood or Voronoi polygons serious consequences on the reliability of the results can occur. Or even worse results are possible, when the centroid of the individual address points is used as the key information in defining postal code areas. As shown in figure 1, these postal code areas will intersect built-up parcels and do not always consider boundaries like roads and rivers that restrict postal code areas in a natural sense.

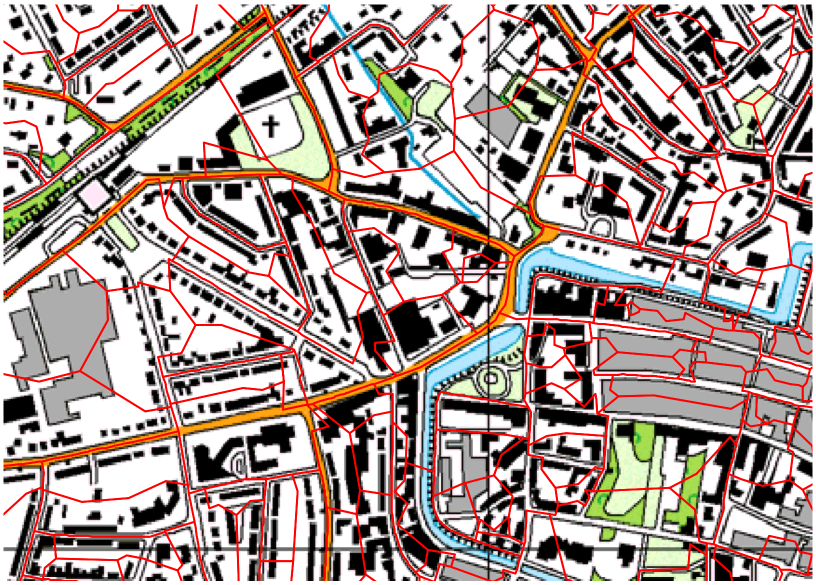

Figure 1. Voronoi polygons sometimes run through buildings and roads which results in bad Postal Code Areas

To solve these problems an approach based on the cadastral registration is proposed and implemented within the Computational Geometry Algorithm Library. The 6-postion postal code is now the efficient intermediary between an address and its location. Given a specific coordinate the address matching algorithm determines by a point in polygon search firstly the postal code area and given the postal code it will subsequently determine the parcel within the postal code area. This parcel points through the cadastral registration to the proper house description by town, address and number of the house. To the opposite the geocoding algorithm will now use the parcel as mean to obtain the proper 6-position postal code polygon

This method works without any problems for built-up parcels. The buildings are not only described by town, address and house number, but also with a postal code provided by the postal services. The parcels, which are not accommodated with a building, do not have a postal code. For a full coverage the most probable postal code should be attached to the remaining non-built parcels as well. To reflect the need to take the boundary of the infrastructural objects into account a procedure is developed where these are divided at their skeleton first. The remaining parts are then aggregated to their most likely neighbor parcel and thus postal coded. The final result will be a planar map that partitions the full coverage into non-overlapping adjacent polygons, where each polygon identifies a certain postal code and where the infrastructural objects are on the boundary of these postal code polygons. This idea is illustrated by figure 2.

This paper describes this procedure - based on the Dutch cadastral registration - to obtain a reliable planar partition postal code map for the Netherlands. The map is also more useful and attractive because of the expected coincidence of the created polygons with the infrastructure.

2. CREATING THE PLANAR PARTITION POSTAL MAP

2.1 The cadastral system as input for the planar postal code map

In the Dutch cadastral system the ownership, the culture code and the build-up code of parcels are registered in the AKR[1]-system and the boundaries are stored in the LKI[2]-system. Within the AKR-system an indication is given to the kind of use of the parcel. The parcel might be built up, or it is used as infrastructure (roads, etc.) or the parcel is in fact water (rivers, canals), etc. If the parcel is built up, it has an AKR object address and therefore a postal code. If the parcel does not has an AKR object address no postal code is known. Or, if the parcel has multiple AKR object

[1] AKR stands for Administratieve Kadastrale Registratiesysteem (in Dutch): 'Administrative Cadastral Registration System'.

[2] LKI stands for Landmeetkundig Kartografische Informatiesysteem (in Dutch): 'Information System for Surveying and Mapping'.

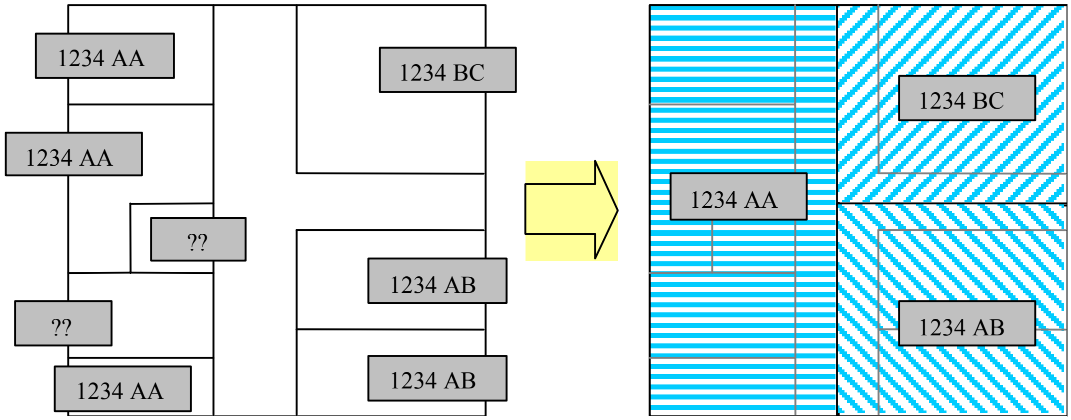

Figure 2. Cadastral situation (left) and its derived postal code areas (right).

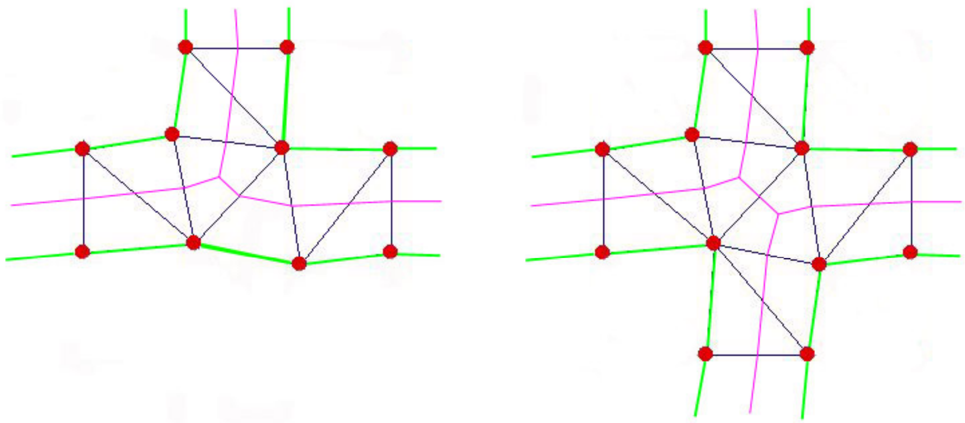

Figure 3. Skeletonized T-junction and crossing

addresses, this may also result in multiple postal codes. The last situation (i.e. a large apartment complex) is excluded in the algorithm, but in the recommendation we describe how the problem can be solved.

Via the AKR/LKI parcels the postal code has a kind of physical representation: the ownership boundaries. As these boundaries are often visible in the terrain, the postal code is too. By joining the parcels with the same postal code, postal code areas are created with boundaries that are visible in the terrain. As a postal code map that covers the complete plane is needed for the address matching, it is necessary to attach a postal code to the parcel where no postal code is given yet. This could be done quite straight forward, by an iterative attachment of the most likely postal code of one of the neighbor parcels. If this operation is performed on infrastructure like roads an undesirable situation occurs. Many houses (and thus parcels) located on one side of the road have a different postal code in comparison to the houses on the opposite site. Adding one certain postal code to the road will block the other postal codes. Address matching for point locations on the road will be quite difficult to perform. Instinctively the boundary between the two postal code areas is located at the road centerline. Therefore the left part of the road should be assigned to the postal code area at the left side and the right part of the road to the postal code area at the right side. Skeletonization is one of the key features of this new algorithm as it is used to split road parcels in order to add these separate parts to different postal code areas.

2.2 The skeletonization of infrastructural parcels

All parcels are incorporated into the planar map data structure of CGAL. This can be easily done because the boundaries of the parcels are given in the LKI database. Then the parcels identified in the cadastral registration as infrastructure or not-built up have to be skeletonized first. The skeletonization is performed by a constrained triangulation of these parcels. As a next step the skeleton is calculated using the method of DeLucia and Black[3]. The skeletonization method of Chithambaran [2] was not useful for our purpose because the resulting skeleton lines have may unwanted curves. The number of constrained edges in a triangle parameterizes this method. Figure 3 shows some examples of this method. Internal triangles with no constrained edges (0-triangles) are splitted by the skeleton in three parts. A 0-triangle is related to a junction. Triangles with one constrained edge (1-triangle) are splitted into two parts. A 1-triangle is related to a normal infra network connection. The triangles with two constrained edges (2-triangle) are also splitted into two parts, where the skeleton connects to an adjacent parcel. A 2-triangle is related to an endpoint in the network After these calculations some extra skeleton branches are calculated to connect the skeleton with the external boundary to enable splitting the parcel in more separate pieces. With this extended skeleton the original parcel is split, see figure 4. The splitted parcels are added into the planar map datastructure of CGAL.

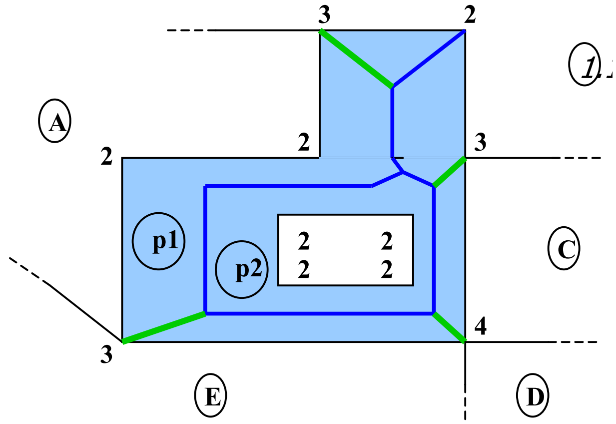

Figure 4. Parcel with skeleton and branches to connect to adjacent parcels

Figure 5. Iterative process assigning postal codes to (splitted) parcels. In each phase more unassigned parcels (gray, dark) get a postal code.

2.3 Assigning postal codes to the skeletonized and unassigned parcels

The infrastructural parcels without a known postal code are now all skeletonized and split into separate parts. These parts have to be assigned with the most plausible postal code of one of the neighbors. The same holds for the parcels without an AKR object address. The algorithm calculates the boundary length of all adjacent parcels and opts for the postal code with the largest length. Note that different neighbor parcels can contribute to the same postal code. This procedure will repeat until all parcels have adopted a postal code, shown in figure 5. A disadvantage of this procedure is the possibility to create two or more areas with the same postal code attached. In figure 6, the hatched areas are part of postal codes consisting of more than one area.

2.4 Uniting the parcels to postal code areas

Uniting the parcels with the same six position (e.g. 1234AB) postal code is done in the planar map by deleting the boundary segments with an identical postal code identifier at the left and right side

3. ASSIGNING A RELIABILITY MARK TO THE POSTAL CODE AREAS

The more iteration steps needed to assign a postal code to a parcel the less certain one is that it will be the 'correct' code. This reliability could be expressed into a relative reliability indicator Q. This Q takes into account the number of iterations to assign a postal code to a parcel. The postal code reliability indicator is normalized between 0 and 1 by dividing the result by the number of parcels in the postal code area. A postal code area created out of many infrastructural or non built-up parcels is thus given an indicator near 0; a postal code area created from built-up areas is

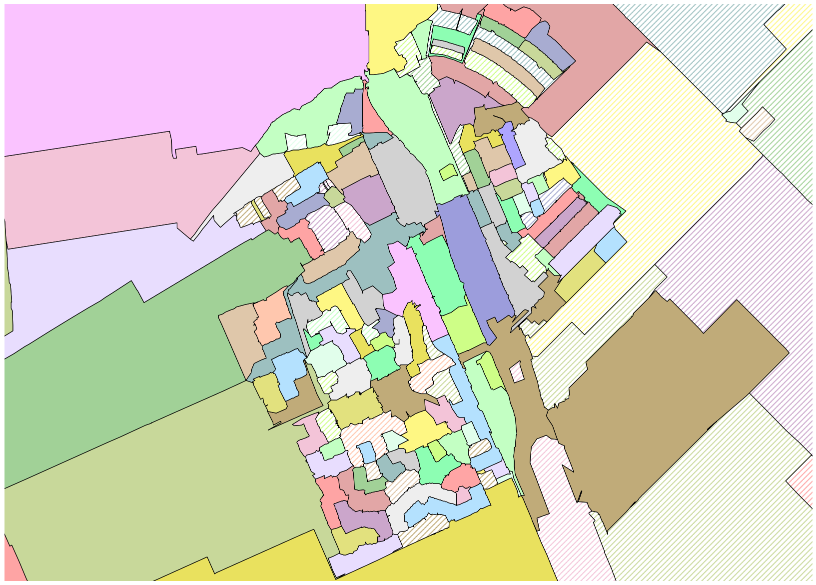

Figure 6. Assigned postal codes; notice disconnected postal code areas

Figure 7. Assigned reliability mark to postal code areas

given an indicator near 1. One can imagine that urban areas have a high indication, and rural areas a low one. This is shown in figure 7.

4. THE USE OF CGAL WITHIN CREATING THE PLANAR PARTITION POSTAL MAP

The complete procedure from the skeletonization to the assignment of the polygons has been programmed in the C++ language using the Computer Graphics Algorithms Library (version 2.4). CGAL[1] is a collaborative effort of several universities and research centers in Europe and Israel. The goal is to make the most important of the solutions and methods developed in computational geometry available to users in industry and academia in a C++ library. Our algorithm extensively uses the 'Planar Map' and 'Constrained Delaunay Triangulation' classes of CGAL. Within the Planar Map, the 'Trapezoidal Map' is used to locate points within the Planar Map.

Advantages of the CGAL library are its flexibility and efficiency. The user can extend the classes provided by CGAL with his own functionality while retaining the efficiency of CGAL functions. Also the types provided are highly parameterized (using C++ templates), so the user can often choose between a more efficient algorithm or to minimize the use of computer memory.

As a result of its flexibility the library has become quite complex. For an average GIS user it takes quite a lot of time to get used to the C++ programming language and the CGAL library. Also there are some pitfalls of choosing the right kernel. The kernel defines which number type is used to represent the co-ordinates of the spatial objects. In our software we used the exact real number representations for the co-ordinates with a homogeneous co-ordinate system. These are relatively expensive in storage and computation but robust.

5. RESULTS

The planar partition postal map algorithm is tested with a data set of the cadastral municipality of Zevenhuizen, in the southwest of the Netherlands. This test indicates some limitations in the use of the skeletonization method by DeLucia and Black. Especially junctions are difficult objects to treat properly. A four way junction is indicated by two adjacent or near 0-triangles, both with no constrained edges, see figure 3. This implicates at each crossing the creation of an edge of the skeleton between the center of the two 0-triangles. Another limitation are the 'false 2-triangles' and the 'shifted 0-triangles', as shown in figure 8 and 9. This results in juts and spikes within the skeleton and thus in the created postal code map, as shown in figure 8. Improving the smoothness of the skeleton via the removal of spikes or the improvement for skeleton junctions such as mentioned in [5] and [8], will directly result in better postal code maps.

But to show the possibilities of this method a part of the planar partition postal map of Zevenhuizen is shown in figure 10. This clearly indicates the connection between the houses and their postal code, although some limitations are visible and also shown in figure 11. The house in the southwest is upon a parcel that has been wrongly classified within the AKR-registration as being not built-up. As a result the parcel is split and intersects therefore the building.

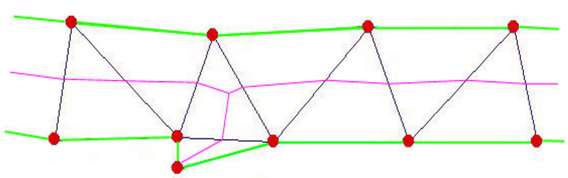

Figure 8. False 2-triangle causing a jut

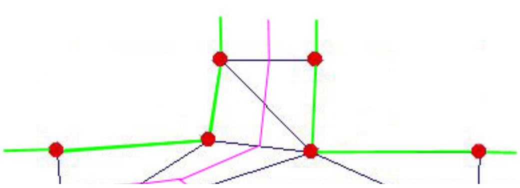

Figure 9. Shifted 0-triangle causing a spike in the skeleton

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

In this paper we presented a new method to derive a planar partition postal code map based on the geometric representation of cadastral parcels together with their associated administrative data, such as object addresses, culture and building codes. Infrastructural parcels (without cadastral address) are subdivided along their skeleton. Sub parcels are created from this skeleton completed with some additional edges from the skeleton to the original neighbor nodes in the parcel map.

The method has been implemented and tested on an actual data set. The results look very promising: better-looking planar postal code maps than the existing Voronoi-diagram based product (better readable due to infrastructural line work, which is recognized by users). Also, visual checks indicate that far less buildings are intersected by postal code boundaries and the data owner has acknowledged in their first reaction this improvement.

A quality indicator has been developed to indicate the reliability

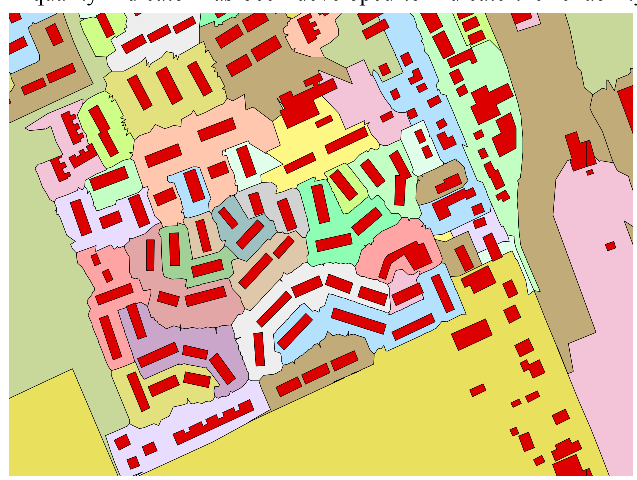

Figure 10. Results of test area Zevenhuizen

of the assigned postal code to the areas. However, one problem became apparent in the evaluation: some postal code regions result in disconnected areas. This could in fact be correct, but if possible it should be avoided as much as possible.

With respect to our implementation platform, we conclude the following: If you have a spatial problem that cannot be handled within a GIS, either for performance or functionality reasons, CGAL will be a good option to look at. However the powers of CGAL come at a price; the user should at least have some experience in C++.

6.2 Future Work

There are chances for a number of further improvements and the current algorithm could relatively simply be changed in the following manner:

Step 1. Assign postal codes (to non-infrastructural parcels without address) and glue parcels to larger areas until an infrastructural parcel is reached.

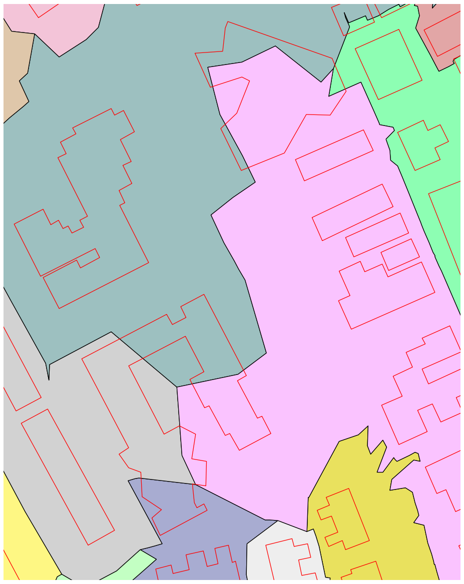

Figure 11. Intersected building by incorrect input data.

Step 2. Compute skeleton of infrastructural parcels and additional skeleton edges to the glued areas.

Step 3. Continue assigning postal codes to areas without address and glue them together.

The advantage of this method is that the 'postal' codes do not cross the roads, which is also in the real postal code world not the case (in the Netherlands). Another advantage is that less additional skeleton edges are needed, which will save computing time. It is however unclear if this will solve the mentioned problem of multiple areas for one postal code.

A quite different alternative implementation is to use a Triangulated Irregular Network (TIN) solely: for the parcels with address (all triangles get the postal code), for the infrastructural parcel (triangles have no postal code, but will be used in skeleton computation), for non-infrastructural parcels without address (triangles get no address, but may iteratively be assigned postal codes of neighbor triangles). It may even be useful to add additional points (called Steiner points) to the boundary and interior of the non-infrastructural parcels, in order to allow smoother growing. Further research has to be done in order to determine the usefulness of this alternative.

We will now finish the paper with other future work, not directly related to the main area computation algorithm:

1. Improve smoothness of skeleton via a) removal of spikes, or b) improvements for skeleton junctions such as mentioned in [5] and [8]

2. Investigate the quality of the administrative data (culture and building code) on the resulting planar postal code map.

3. Derive beside the most detailed 6ppc (6 position) also the larger postal code area units: 5ppc and 4ppc (or even 3ppc, 2ppc or 1 ppc) maps.

4. Produce more quantitative proof of the improved quality (also for larger data sets and different types of regions.)

7. ACKNOWLEDGEMENTS

The authors would like to express their acknowledgements to de Dutch Kadaster for use of their AKR and LKI databases for this research and all other research conducted by the Section GIS-technology within the Geo Database Management Centre of the Delft University of Technology. Finally we would like to thank Elfriede Fendel for the review of this paper.

8. REFERENCES

[1] CGAL Basic library manual. CGAL Consortium, 2002.

[2] Chithambaram, R. et al, ‘Skeletonizing polygons for map generalization’. In: Technical papers, ACSM-ASPRS Convention, Cartography and GIS/LIS, Volume 2. Bethesda: ACSM, 1991, p. 44-54.

[3] DeLucia, A. and T. Black, ‘A comprehensive approach to automatic feature generalization’. In: Proceedings of 13th International Cartographic Conference. International Cartographic Association, 1987, p. 169-191.

[4] Fabri, A. et al, On the design of CGAL, the Computational Geometry Algorithms Library (research report). Saarbrücken: Max-Planck-Institut für Informatik, 1998.

[5] Jones, C.B. et al, ‘Map Generalization with a Triangulated Data Structure’. Cartography and Geographic Information Systems, 22-4 (1995), p. 317-331.

[6] Oosterom, P.J.M. van, ‘Hartlijnen van wegennetwerken’. Informatieblad Kadaster, 7 (1999), p. 10-11 (in Dutch).

[7] Schut, T.G., en B. Maessen, Haalbaarheid 6PPC-vlakkenkaart (eindrapportage). Apeldoorn: Kadaster, 2000 (in Dutch).

[8] Uitermark, H. et al, ‘Semantic and Geometric Aspects of Integrating Road networks’. In: A. Vckovski et al (eds), Interoperating geographic information system: second international conference, INTEROP ‘99. Berlin: Springer-Verlag, 1999, p. 177-188.

[9] Wilschut, A. N., van Zwol, R., Brasa, N., Flokstra, J. (1998), Road collapsein magnum, in R. Laurini, ed., `6th ACM Workshop on Advances in Geographic Information Systems ACM-GIS'98, Washington DC, USA, 6-7 november 1998'.