Using the constrained tGAP for generalisation of IMGeo to Top10NL model

Arjen Hofman[1], Arta Dilo[2], Peter van Oosterom[2], Nicole Borkens[3]

[1]Logica Nederland B.V., Geo ICT Public Sector, Laan van Zuid Hoorn 70
P.O. Box 3190, 2280 GD Rijswijk, the Netherlands
[2] Delft University of Technology, OTB, Section GIS-technology,
Jaffalaan 9, 2628 BX Delft, the Netherlands.
[3] Gemeentewerken Rotterdam, Department of Surveying,
Galvanistraat 15, 3029 AD, Rotterdam, the Netherlands
Email: arjen.hofman@logica.com, A.Dilo@tudelft.nl, P.J.M.vanOosterom@tudelft.nl,
N.Borkens@gw.rotterdam.nl

In this paper the results of merging large scale (1:1,000) and medium scale (1:10,000) topographic data into one structure are presented. This structure can be used as a single non-redundant representation for topographic data, which can be queried at any arbitrary scale (vario-scale) between the source scales. The solution is based on the constrained topological Generalised Area Partition (tGAP) structure, where the area objects of the medium-scale map function as region-constraints for the large-scale objects in the aggregation process. The result contains the accurate geometry of the large scale objects enriched with the generalization knowledge of the medium scale data (stored as references in the structure). The advantage of this constrained approach compared to the unconstrained tGAP is the higher quality of the aggregated maps. The ideas have been explored with real topographic data of the municipalities of Almere and Rotterdam, in which prototypes of the large scale IMGeo data and production versions of the medium scale Top10NL data are used. The ultimate goal is to have only one representation, which can then be maintained at the largest scale and is able to serve any arbitrary smaller scale (and efficiently produce output results comparable to the typical smaller scale representations as known today). This goal is not yet achieved, but this paper shows one step further in this direction and the results illustrate the feasibility of the ultimate goal.

KEYWORDS: map generalization, constrained tGAP structure, IMGeo model, Top10NL model,

1. Introduction

Automated generalisation is for most producers of geographic information a way to increase efficiency of map production within their organisation. It is a chance to do the data collection work only for a large-scale data and derive smaller scales out of this large-scale data automatically. All Dutch municipalities produce a large-scale topographic base map. Besides this, large municipalities in the Netherlands also produce their own medium- and small-scale topographic maps. The large-scale base map is produced from terrestrial surveying, while the map products at other scales are drawn from aerial photographs. This can be considered to be inefficient, since the real world that has to be represented is in all cases the same. Derivation of the smaller scale maps from one single source, the large-scale base map, would be the solution to this inefficiency problem. This inefficiency problem forms the basis of the research presented in this paper.

The research has been performed within the municipality of Rotterdam. The possibility to generalise a topographic dataset starting from a large-scale dataset is investigated in this paper. The large-scale dataset is a 1:1,000 dataset according to IMGeo, a new Dutch model on large-scale topography. The medium-scale dataset is a 1:10,000 dataset that is similar to the national 1:10,000 model, Top10NL, with some differences and additional information as well.

Top10NL is the current authentic registration on topography in the Netherlands. There are, in total, 10 authentic registrations in the Netherlands, spatial and non-spatial. Each authentic registration is the only allowed source in its field within the government, and is related to other registrations in the system of authentic registrations. The registration on topography should be connected to the registration on addresses and buildings (BAG) and to the cadastral registration. However, the geometry for BAG is required to have the accuracy of the large-scale 1:1,000 base map, and for this reason the Top10NL stands a bit outside of the system of authentic registrations. It would be better to have a scale independent authentic registration on topography. This paper shows that the 1:1,000 registration, IMGeo, can be connected to the 1:10,000 Top10NL.

Several solutions to generalisation are suggested in literature, with Multiple Representation Data Bases (MRDB) being a substantial part. Because nowadays map viewing is often done through web services, which are quite scale independent, we do not want a solution with fixed scales as in MRDB. Instead, we opt for the constrained tGAP structure, to create a vario-scale IMGeo. The constrained tGAP (Haunert et. al. 2007 submitted) is an idea that builds on the topological Generalised Area Partition (tGAP) structure. We apply the constrained tGAP to the IMGeo data, with Top10NL as constraints.

This paper starts with an introduction to the constrained tGAP in Section 2. The two models, IMGeo and Top10NL, as well as actual datasets according to these models (in the municipalities of Almere and Rotterdam) used in this research, are presented in Section 3. Section 4 discusses the different processing steps of data and presents the obtained results. The paper closes with conclusions in Section 5.

2. Constrained tGAP

The constrained tGAP is an extension of tGAP (van Oosterom 1990, van Oosterom 2005). The tGAP is (currently) working with spatial data in 2D, and assumes that the input objects in space form a partition, i.e. no overlap between objects, no gaps in space. The tGAP is a collection of structures that avoids storing redundant geometry by storing structural information representing generalisation knowledge (performed off-line and may take into account advanced criteria such as contextual information). For example, the generalisation process forms objects for smaller scales as aggregation of objects in the largest scale. The basic idea is to store the geometry only for objects of the largest scale, whereas for objects of smaller scales, i.e. objects created from generalisation, store references to the composing objects. The tGAP aims at minimal geometrical redundancy, and achieves it in two directions: using a topology model for storing objects of the largest scale; for any object of a smaller scale it stores references to features of the largest scale, which are used to construct this object. The filled structures can then be queried efficiently by selecting and composing the right objects to be visualised at a given scale.

This paper mainly uses the aggregation operation in the creation of the structure, but other generalization operations have been described earlier in the context of the tGAP structure; e.g. line simplification, typification, displacement, symbolization, collapse of area to line (or point) and assigning the parts to the neighbours of a subdivided area (van Putten and van Oosterom 1998, Ai and van Oosterom 2002); see Figure 1. These techniques would all fit into the presented structure. Actually, the splitting of a large scale area object is also presented in this paper, but in a different context than as a result of the area to line collapse operation (now as a potential result of small scale constraint regions splitting a large scale area object into parts; see section 4.2).

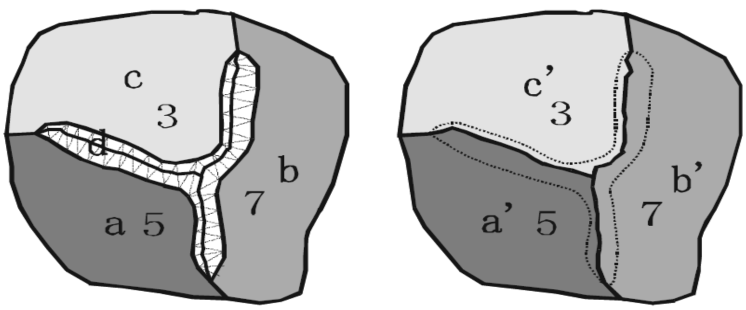

Figure 1: Area to line collapse and assigning the resulting parts to neighbour area's with weighted importance (taken from Ai and van Oosterom 2002)

The generalisation pre-processing is performed in iterations (steps). In each iteration, the least important object is merged to its most compatible neighbour, forming a new object. In the constrained tGAP, the pairwise merging is controlled by region constraints, which are pre-defined sets of the largest scale objects. Objects are allowed to merge only if they belong to the same region constraint. Two functions, *LeastImpFace* and *MostCompNeighbour*, calculate the least important face for each iteration, and its most compatible neighbour. Both functions have several input arguments, such area size of objects, and class of objects. They use a class weight table, and a class compatibility matrix. For iteration `i`, the function *LeastImpFace* returns the face with minimum importance (Eq. 1).

$$min\{\ area(\mathtt{x}) * weight(\mathtt{class}(\mathtt{x})) \mid \mathtt{x} \text{ a valid face in iteration } \mathtt{i}\ \} \qquad \text{Eq. 1}$$

The function *MostCompNeighbour* returns the face that has maximal compatibility with face `x` in iteration `i`, as calculated by Equation 2 (note the in the constrained tGAP, the candidates are limited to areas in the same region).

$$max\{\ compatibility(\mathtt{class}(\mathtt{x}), \mathtt{class}(\mathtt{y})) * area(\mathtt{y}) \mid \mathtt{y} \text{ is valid in iteration } \mathtt{i} \text{ and is a neighbour of } \mathtt{x}, \text{ and } \mathtt{region}(\mathtt{y}) = \mathtt{region}(\mathtt{x})\} \qquad \text{Eq. 2}$$

Figure 2 shows results of different steps of the constrained tGAP generalisation. Figure 2(a) shows the original data, i.e. objects of the largest scale, which is the starting step of the generalisation process. Constraints are shown in thick black line in the original data and all intermediate steps, Figure 2(a)-1(e). Figure 2(f) shows the region constraints; it is also the last step of the generalisation. Generalisation stops when all objects are merged up to these region constraints. In the intermediate steps, Figure 2(b)-2(e), it can be seen that merging is performed only inside region constraints.

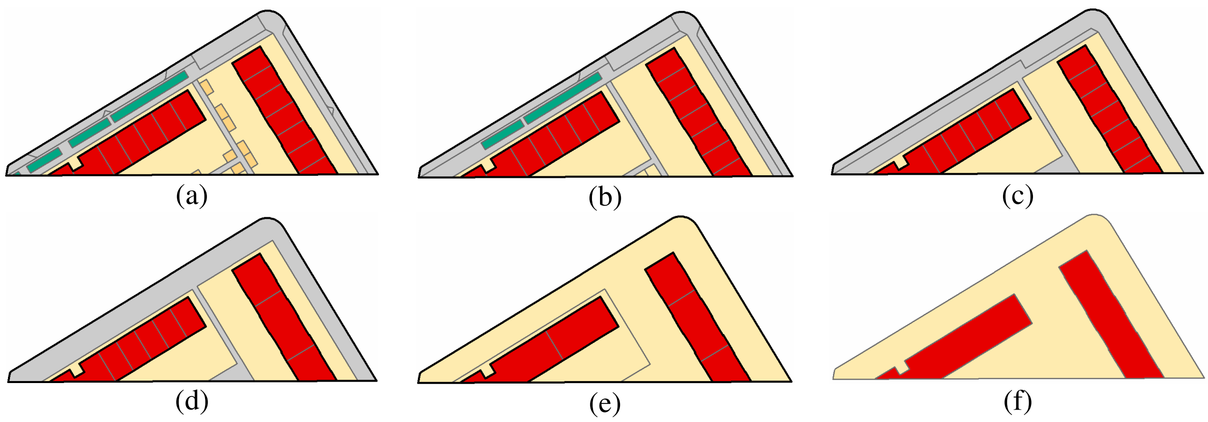

Figure 2: Results of different steps of constrained tGAP generalisation. Constraints are shown in thick black line through (a)-(e). (a) original data, 42 objects, (b) result of generalisation step 15, (c) result of step 25, (d) result of step 30, (e) result of step 35, (f) result of last generalisation step (equal to the region constraints).

The tGAP structures are stored as Oracle tables, and the constrained tGAP generalisation is performed by code written in PL/SQL. Oracle tables used for storing the constrained tGAP can be found in the appendix of this paper, and the full code for the constrained tGAP can be found in (Hofman, 2008, appendix G). Actually the region constraints subdivide the whole domain into parts that can be processed independently (and therefore also in parallel), also due to this the scope is more local. This is relevant when considering updates: the original tGAP iteratively looks for the globally least important feature, and a small change in the largest may cause quite a different resulting structure. This is not the case for the constrained tGAP, where the effect of a change is limited to a region constraint.

3. Description of models and datasets

IMGeo and Top10NL are both based on the same Dutch standard on geography, NEN3610 (NEN, 2005). This is the Dutch version of the 'General Feature Model' ISO 19109 to which top-level semantically meaning full classes have been added (water, road, terrain, etc.). The International Organisation for Standardisation (ISO) defines real-world objects in the ISO 19109 standard and translates them to geographical features (ISO, 2005).

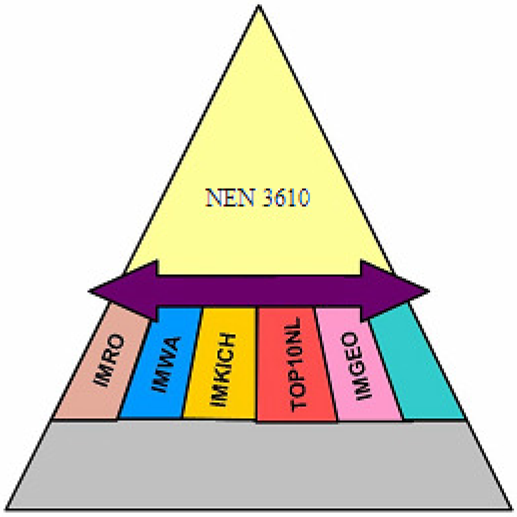

Figure 3: NEN3610 and a number of its derivatives (currently at least a dozen of these so called sector information models have been developed and most of them are also in production use).

Figure 3 shows a number of derivative sector specific information models (IM's) based on NEN3610. Among these there are two topographic models, IMGeo and Top10NL, which will be described in two separate subsections 3.1 and 3.2. Despite the fact that both models are derived from the same standard, there are differences between these models; these differences are described in subsection 3.3.

3.1 IMGeo model

IMGeo is an object-oriented large-scale topographic model. Until now there was only the line based GBKN (Dutch acronym for large scale base map) in the Netherlands. But with the introduction of IMGeo in 2007 the expectation is that this will become the new version of GBKN in the near future.

The main object classes in IMGeo consist of a main object that can be split into more individual object parts. For example a road can be split at every junction and still be identified as one road object. The main object classes of IMGeo are:

- Road
- Railroad
- Water
- Terrain
- Building
- Topographic element
- Civil work
- Registration area

3.2 Top10NL model

As already stated in the introduction, Top10NL is the current authentic registration on topography in the Netherlands. This means that every governmental organization is obliged to use Top10NL when working with a 1:10,000 map product from January 1st 2008 onwards. For municipalities that produce their own 1:10,000 map this applies from January 1st 2010. Since most of these large municipalities want to keep producing their own 1:10,000 maps, because of data content, up-to-dateness and economic reasons, it is important to integrate their own maps well with Top10NL. In this manner they might become the data produces in their territory for the Top10NL and deliver the data to the national service (the topographic department of the Dutch cadastre).

The main object classes of Top10NL are quite similar to the IMGeo classes, with the difference that Top10NL does not have separate class for civil works but includes them in the topographic elements class. Besides this Top10NL has three additional classes: functional area, geographical area and isolines.

3.3 Differences between IMGeo and Top10NL

The Top10NL model has been made before the IMGeo model. Although the official release of Top10NL was on January 1st 2008, the model already existed for some years. IMGeo has been created from 2006 and was finalised in October 2007. The Top10NL model has not been the starting point of the makers of IMGeo; they made their own model based on their own judgements. This has led to many small differences, which makes it hard to combine the models. The most striking differences are:

- IMGeo is collected by terrestrial surveying, Top10NL is collected from aerial photographs
- IMGeo is an area partition, Top10NL is not
- Top10NL has inherited the NEN3610 classes; IMGeo only makes reference to NEN3610 as a source (in the textual description and not in the model)
- Some different classes are modelled (see 3.2)

This can all be understood from the fact that the models have a different background. IMGeo is derived from the GBKN and Top10NL from its predecessor Top10Vector. GBKN is a topographic registration, whereas Top10Vector really was a topographic map.

The test data used shows also differences. Data from the municipality of Almere was used, because it was one of the few municipalities in the Netherlands that could provide IMGeo data in 2007 (when this research was carried out). The corresponding Top10NL shows mainly geometrical differences on buildings and roads and also some semantical differences. The geometrical differences give problems when assigning IMGeo objects to Top10NL regions, the semantical differences make that no simple mapping between IMGeo and Top10NL classes can be made. Later, tests were performed also on data from the municipality of Rotterdam (results are shown in Appendix C). Rotterdam data shows the same problems with buildings, but less problems with roads.

In Figure 4 it is shown what problems occur when overlaying buildings. On the IMGeo map, the IMGeo buildings are in red and overlaid with the transparent blue Top10NL building blocks. The figure shows that the building blocks in Top10NL are placed very inaccurately over the IMGeo buildings. The shift is very irregular and cannot be explained by parallax (error in photogrammetric data acquisition of objects with some relative height). The only explanation is that some displacement is added by the cartographer, though not needed according to the product specifications of Top10NL.The required accuracy of Top10NL is 4 meters and the differences indicated are all within this accuracy. However, this inaccuracy does not apply to all objects (and many other object types have better accuracy in the Top10NL).

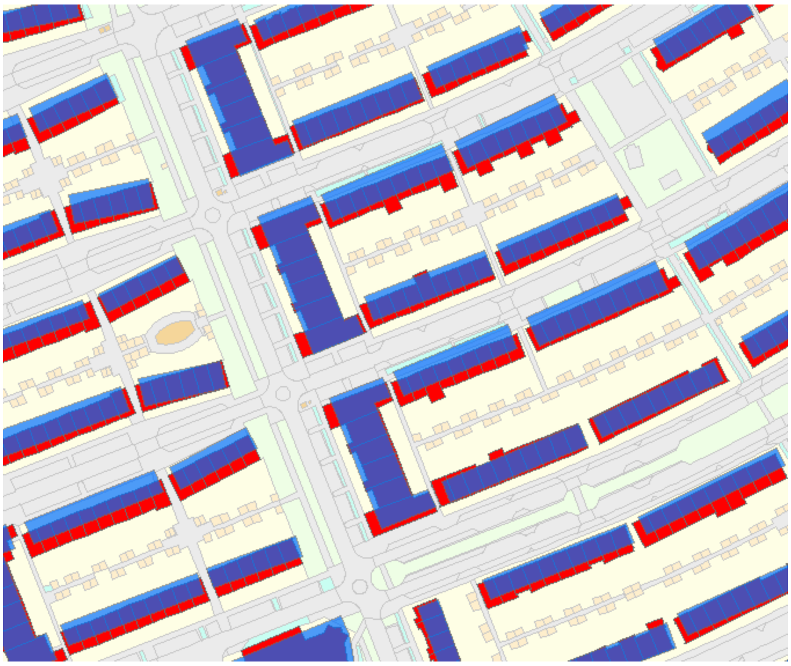

Figure 4: IMGeo buildings (in red) overlaid with Top10NL buildings blocks (in blue). The background is IMGeo data that is made transparent.

Figure 5 shows again the IMGeo map with the IMGeo roads in grey, overlaid with the transparent Top10NL roads in blue. In figure 5 it is shown that Top10NL road objects match geometrically very well with the IMGeo road objects. The problem shown in this figure is that there are far more road objects in IMGeo than in Top10NL. IMGeo data of Almere considers parking places and sidewalks as road, while Top10NL does not consider them roads. The road objects in IMGeo are not subdivided in different categories, because this was an IMGeo pilot project and the municipality of Almere did not have this data. The IMGeo data from the municipality of Rotterdam is better with respect to this issue; there is a more detailed classification of roads.

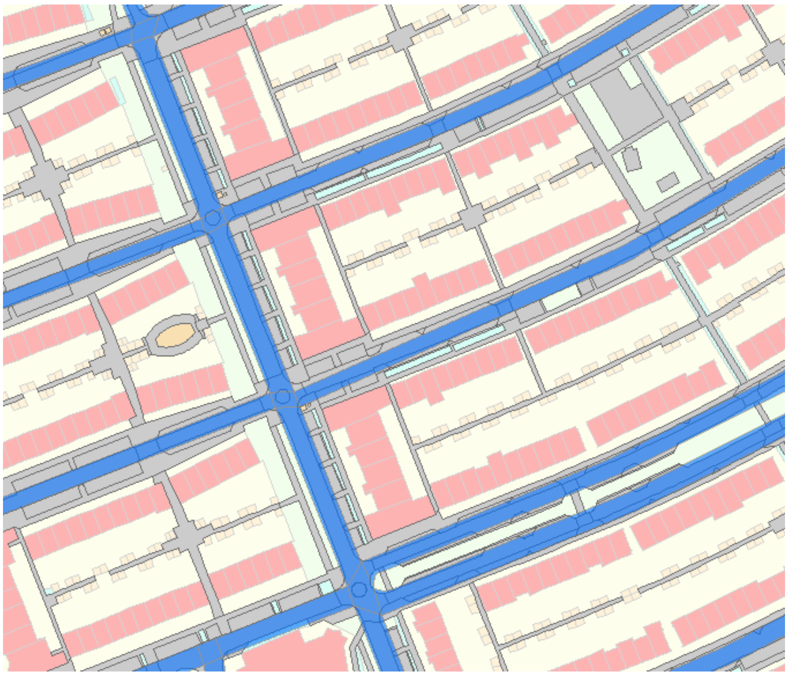

Figure 5: IMGeo roads (in grey) overlaid with Top10NL roads (in blue). The background is IMGeo data that is made transparent.

The semantical differences appear for example in areas with lots of larger shrubs. This can be shown in Top10NL as wood. Because the tGAP structure will be built from the large-scale objects (where for this type of terrain another classification is used), we cannot arrive in the end with a situation in which we have a totally different object after merging.

In the next section the test data presented in this section is processed. IMGeo objects are assigned to Top10NL objects (our region contraints) and the resulting data is processed by the tGAP code.

4. Data processing

The constrained tGAP has some assumptions, which put requirements on the data we want to generalize. The first assumption is that data has to form an area partition, which was not the case for IMGeo data of Almere and Rotterdam as overlapping areas did occur. The second assumption is that we know in advance to which region (constraint) each object of the original data belongs. We want to use Top10NL objects as region constraints for the IMGeo data, while the two are created independently. We need to assign a Top10NL object to every IMGeo object before we can run the constrained tGAP code. Finally, results of the generalization process are quite affected by weights and compatibilities, thus a good choice of these values is important.

This section will treat the solution for these three issues: creating area partitions in subsection 4.1; assigning Top10NL objects to IMGeo objects in subsection 4.2, and calculation of weights and compatibilities in subsection 4.3. Subsection 4.4 describes results of the constrained tGAP generalization on IMGeo data of Almere.

4.1 Resolving overlaps in IMGeo and Top10NL

IMGeo should normally be an area partition, as stated in Section 3.3. However this is not the case for the Almere test dataset, and the same is true for Rotterdam data. This is due to the fact that Almere was a pilot project for IMGeo and the Rotterdam data is also prototype IMGeo data ('pre-production' status).

The IMGeo test data is turned to an area partition using an ordering of classes with respect to their importance. For example, terrain objects are erased when overlapping with building objects. In case of a bridge a water object and a road object overlap; water object is erased. The rules for creating the partition are mentioned shortly below; the full treatment can be found in (Hofman, 2008).

- If 'Terrain' and another object overlap, the terrain part is erased.
- If 'Water' and 'Road' overlap, the water part is erased.
- If 'Road' and 'Other building' overlap, the road part is erased.

Also Top10NL is made an area partition, because the IMGeo objects have to be assigned to only one region. In case two or more Top10NL objects overlap, we would face the problem to which of those objects assign an IMGeo object. To avoid this, we resolve the overlaps in Top10NL by turning it to an area partition. The rules for creating a partition from Top10NL are:

- If 'Terrain' and another object overlap, the terrain part is erased.
- If 'Water' and 'Road' overlap, the water part is erased.

4.2 Relating IMGeo objects to Top10NL regions

The results of the previous processing are two data sets, each one being a partition. To assign IMGeo objects to region constraints, i.e. Top10NL objects, several methods were investigated (and implemented in an ArcGIS/Python environment). Four possible methods to assign IMGeo objects to Top10NL regions were developed:

1. The *simple overlay* method: An intersection between the models where every IMGeo object is split at the borders of the overlapping Top10NL region. In the end result only Top10NL geometry will be visible.
2. The *maximum area* method: The Top10NL region which overlaps the IMGeo object the most is the shape to which the whole IMGeo object is assigned to. The IMGeo geometry is kept in this method.
3. The *35%-split* method: If an IMGeo object belongs for more than 35 % to two Top10NL regions we consider this Top10NL geometry as enrichment of the structure; therefore the IMGeo object is split and new IMGeo objects are created. For all other IMGeo objects the maximum area method is applied.
4. The *building first* method: This method assigns IMGeo-buildings to a building region in case of some overlap with a Top10NL building block, without considering the amount of overlap. The other IMGeo objects are processed as in the maximum area method.

In the first method the structure does actually contain all geometries from both data sets and is therefore against the 'non geometric redundancy' principle of the tGAP structure (and it additionally contains the coordinates of the line segment intersections). However, the result is a structure that can be used quite nicely to morph between the two given input data sets.

In the second method an IMGeo object is always completely assigned to a Top10NL region. This implies that that the IMGeo objects belonging to a region can be partially outside the original Top10NL object. Looking at Figure 4 the reader can probably imagine that this method gives problems when assigning some individual IMGeo building areas to a Top10NL region. Since some IMGeo buildings have an overlap of less than 50% with the corresponding Top10NL building block, they are assigned to another region type (terrain), which give quite bad results: missing building blocks, see e.g. Figure 12(b).

Because the first two methods did each have their own drawbacks, the third method was investigated: have some limited amount of splitting. In case of an IMGeo object being overlapped for at least 35% with one Top10NL object and also at least 35% for another Top10NL object, the 'missing' Top10NL geometry could be considered to be of so much importance that it needs to be added to the IMGeo dataset. In such cases the IMGeo object is to be split. This might avoid assigning the complete IMGeo building to a non building block region in the Top10NL (and thereby completely losing the building). However, the results are not very satisfactory either as the bad overlap between the IMGeo buildings and the Top10NL building blocks cause some building blocks in the end result to be very narrow; see Figure 12(c).

Since we saw in subsection 3.3 that the buildings caused problems in the overlay, the problems arising with these methods are not surprising. A forth method was developed, called ‘building first’, which checks first all buildings, and assures that an IMGeo building gets assigned to a Top10NL building-block region in case of any overlap with a Top10NL building-block region, irrespective of the amount of overlap. All the other objects were processed with the technique of the maximum area overlap described earlier. Since this method proved to give the best results (based on our visual inspection of applying the various methods to our test data; see Appendices B and C), we decided to continue with data resulting from this last method.

4.3 Calculation of weights and compatibilities

The first results of the constrained tGAP were calculated without values for the weights and compatibilities. Involving weights and compatibilities led to better results. The first values for weights and compatibilities referred to the work of van Putten (van Putten and van Oosterom, 1998), which was the best practice for the moment. These values were tuned in order to get better generalization results. The method to come to good values for the weights and compatibilities was just trial and error (again based on visual inspection of the results). The optimum found for the weights and the compatibilities can be found in Tables 1 and 2.

Class Name	Code	Weight
Residence object / Building	1001	13
Other Building	1002	1
Road	2001	1,2
Water	3001	1,3
Lot	4001	9
Fallow lying terrain	4002	1
Plants	4003	0,9
Terrain (to be determined)	4004	0,1
Grass / Grassland	4005	1
Wood	4006	0,4
Other terrain	4007	0,3
Bin	5001	0,1

Table 1: Class weights determined for the vario-scale IMGeo.

From Class →	1001	1002	2001	3001	4001	4002	4003	4004	4005	5001
To Class ↓										
1001	1	0,99	0,5	0	0,99	0,9	0,5	0,5	0	0
1002	0,99	1	0,5	0	0,5	0,9	0,5	0,5	0	0
2001	0,5	0,5	0,99	0	0,5	0,5	0,95	0,9	0,5	0,95
3001	0	0	0	1	0	0	0	0	0	0
4001	0,9	0,95	0,95	0	1	0,9	0,9	0,9	0,8	0,9
4002	0,9	0,95	0,9	0	0,95	1	0,9	0,9	0,5	0,8

4003	0,5	0,5	0,5	0	0,8	0,5	0,99	0,95	0,9	0,9
4004	0,9	0,95	0,5	0	0,8	0,9	0	1	0,5	0,8
4005	0,5	0,5	0,9	0	0,8	0,5	0,95	0,95	0,99	0,95
5001	0	0	0	0	0	0	0	0	0	1

Table 2: Compatibility values for the vario-scale IMGeo.

With these values the best results for a vario-scale IMGeo can be achieved using the test data presented in this paper. These results follow in Section 4.4.

4.4 Results of constrained tGAP generalization in IMGeo data of Almere

Having the area partition of Almere (as created in subsection 4.1), and a region constraint for each object (as explained in subsection 4.2), together with the weight table and compatibility matrix (given in subsection 4.3), we can apply the constrained tGAP generalization for IMGeo data of Almere (and Rotterdam). This fills the tables with the initial data, i.e. the largest scale. The PL/SQL code of constrained tGAP fills the tables with results of generalization. Figures 6, 7 and 8 show the same part of Almere, for comparison: the largest scale, IMGeo (1:1,000); the result of constrained tGAP generalization for scale 1:10,000; and the Top10NL data of Almere. Figure 7 shows the results of this research. Comparing it to Figure 8, the Top10NL dataset, it can be stated that line simplification is still needed to further complete the results. However, it can also be concluded that the results in Figure 8 are quite a good generalization of the original data in Figure 6 (and based on the tGAP structure also all intermediate scales can be obtained; see Appendix B for some examples).

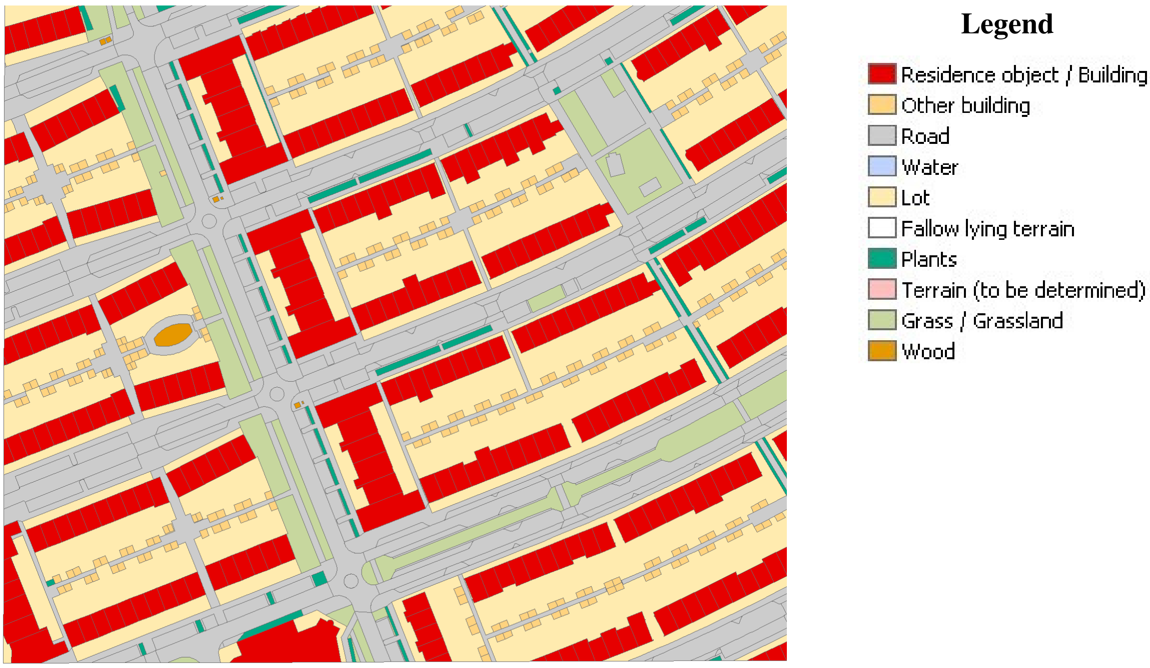

Figure 6 A part of original Almere IMGeo data at scale 1:1,000.

Figure 7: Results of the constrained tGAP for scale 1:10,000.

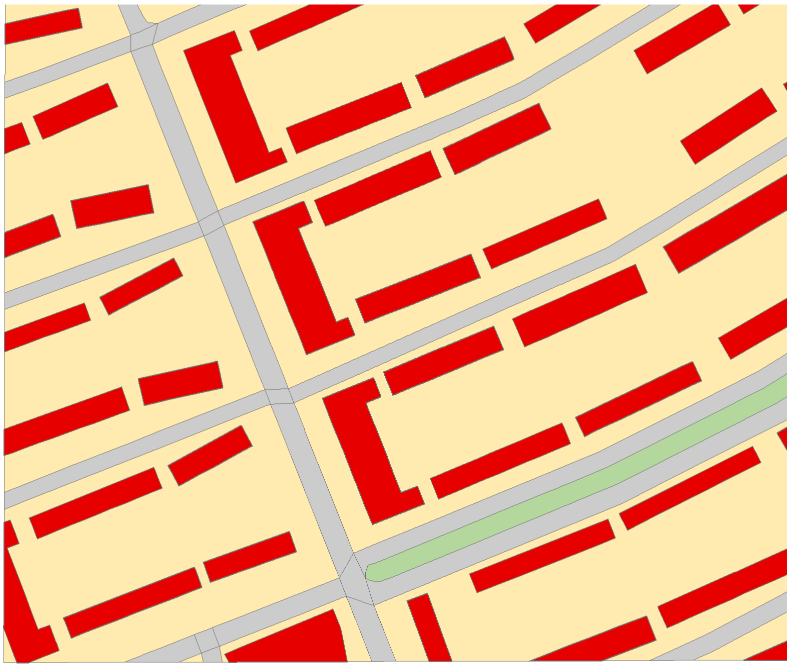

Figure 8: The Top10NL data for the same part of Almere.

5. Conclusions and Future research

In this research the possibilities of a vario-scale IMGeo were investigated. Since IMGeo is a new model its use is not yet very much investigated. This has been done in this research, also a comparison with Top10NL model. To be able to really integrate IMGeo and Top10NL in the constrained tGAP structure cooperation between the makers of the models will be necessary.

The constrained tGAP offers possibilities for the combination of two separate datasets. This paper presented a method to assign large-scale objects to medium scale regions. This can be done for other scales as well.

Once such assignment is done, constrained tGAP generalization can be performed on the large-scale objects, storing the results in the tGAP structure. Data stored in the tGAP structure is to be used to show maps for any scale between the large-scale and the scale of the region constraints. A final goal for the constraint tGAP would be a learning generalization process from the (first use of) region constraints, which could then run without constraints. For example, tuning of compatibility or weight values could be directed by the (initial) region constraints. In this research Top10NL was really necessary to get to good results. However, the vario-scale solution that the constrained tGAP offers is satisfactory.

Weight and compatibility values presented in the paper are results of tuning after several tests of performing constrained tGAP generalization with different values. Statistics taken from the overlay of Top10NL and IMGeo datasets can be useful for the calculation of weights and compatibilities. This possibility is a consideration for future research.

From the generalization results shown in previous section we see the need for line simplification. The use of Douglas-Peucker algorithm for line simplification is proposed in (van Oosterom 1990), and its implementation is given in (Meijers 2006). Simplification of buildings may require other techniques for better results. We are currently implementing the Visvalingam's algorithm for line simplification (Visvalingam and Whyatt 1993). Investigation of best results for line simplification is part of future research. Another direction of future research is the propagation of updates performed at the largest scale to the medium scale, and more generally the propagation of updates in the tGAP structure.

References

Ai, T. and van Oosterom, P. (2002). GAP-Tree Extensions Based on Skeletons. In: *Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling*, 501–513.

Galanda, M. (2003). *Automated Polygon Generalization in a Multi Agent System*, PhD thesis, Deptartment of Geography, University of Zurich.

Haunert, J. H., Dilo, A., van Oosterom, P. (2007). Using the constrained tGAP for progressive vector data transfer, Submitted to *Computers & GeoSciences*.

Hofman, A. (2008). *Developing a vario-scale IMGeo using the constrained tGAP structure*. MSc thesis Geomatics, Technical Delft University (www.gdmc.nl/publications).

International Organisation for Standardization (ISO), (2005). *ISO 19109:2005 Geographic Information - Rules for Application Schema*.

Meijers, M. (2006), *Implementation and testing of variable scale topological data structures,* MSc thesis Geomatics, Technical Delft University.

Nederlands Normalisatie Instituut (NEN) (2005) *Basis model Geo-informatie (in Dutch), NEN 3610.*

van Oosterom, P. (1990), *Reactive Data Structures for Geographic Information Systems*. PhD-thesis Department of Computer Science, Leiden University.

van Oosterom, P. (2005), Variable-scale Topological Data Structures Suitable for Progressive Data Transfer: The GAP-face Tree and GAP-edge Forest, *Cartography and Geographic Information Science*, 32 (4): 331–346.

Putten, J. and van Oosterom, P. (1998). New results with Generalized Area Partitioning. In: *Proceedings of the 9th International Symposium on Spatial Data Handling*, 485–495.

Visvalingam, M. and Whyatt J. D. (1993). Line Generalization by Repeated Elimination of Points, *Cartographic Journal*, 30 (1), 46–51.

Appendix A

Oracle tables storing data for the constrained tGAP, their relations, as well as primary and foreign keys. The label PK in front of a column name indicates that it is (part of) a primary key; label FK indicates that it is a foreign key; label pfK indicates that it is (part of) a primary key that is also a foreign key.

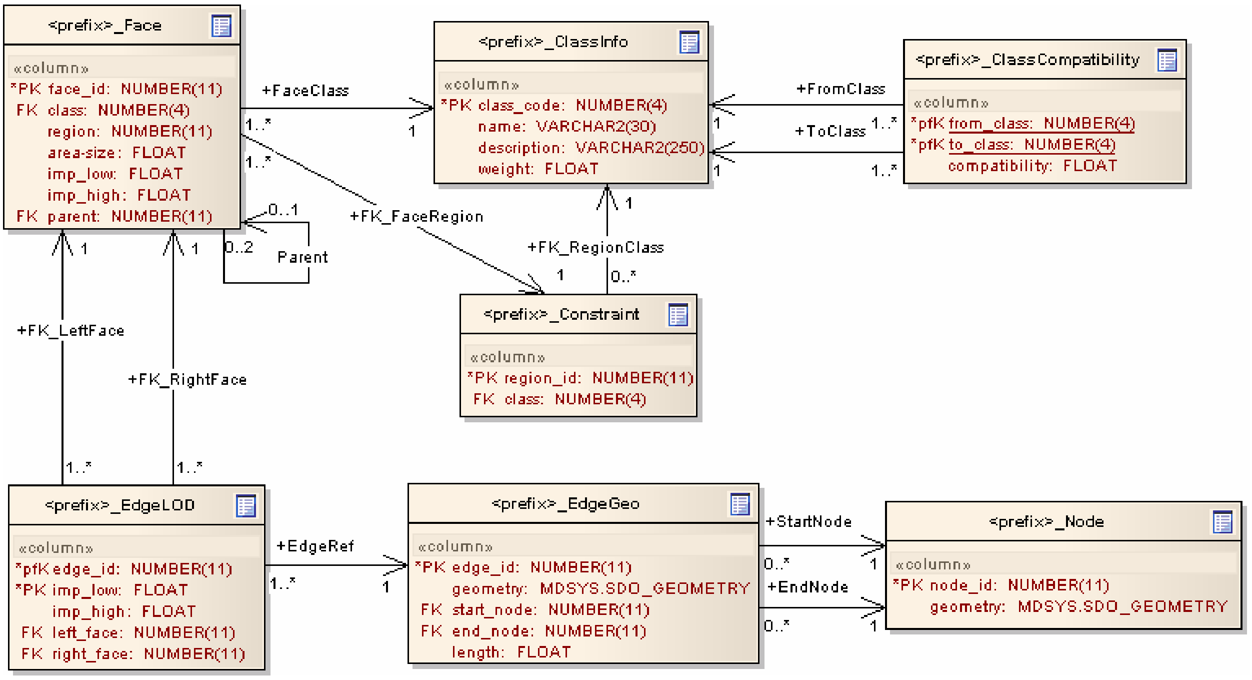

Figure 9: UML diagram of Oracle tables and relations storing data for the constrained tGAP.

Appendix B

This appendix shows results of the four approaches used for relating large scale to medium scale objects. Figures 10 and 11 show (part of) the original IMGeo and Top10NL data of Almere, respectively. Figure 12 shows results of the four different methods presented in this paper. Each one shows IMGeo objects from Almere, colored according to the region they are assigned to. It is visible that the different methods mostly have consequences for the way the buildings are classified. All illustrations in this appendix use the same legend with Figures 6-8.

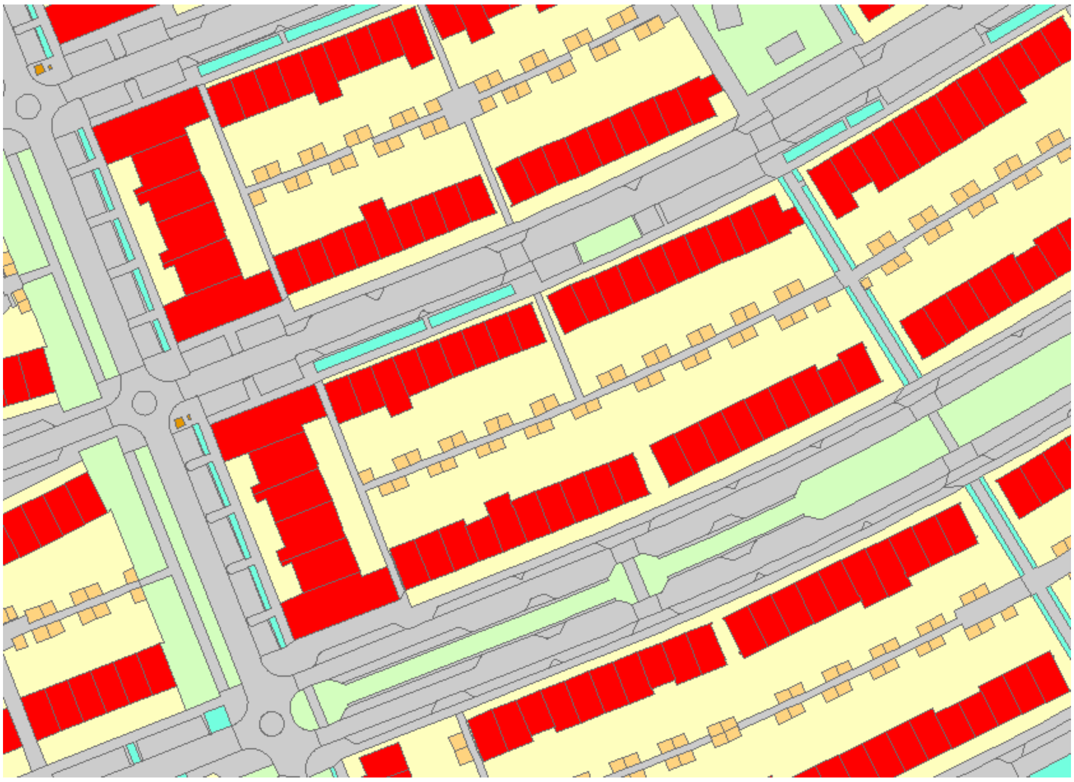

Figure 10: Part of the original IMGeo dataset of Almere.

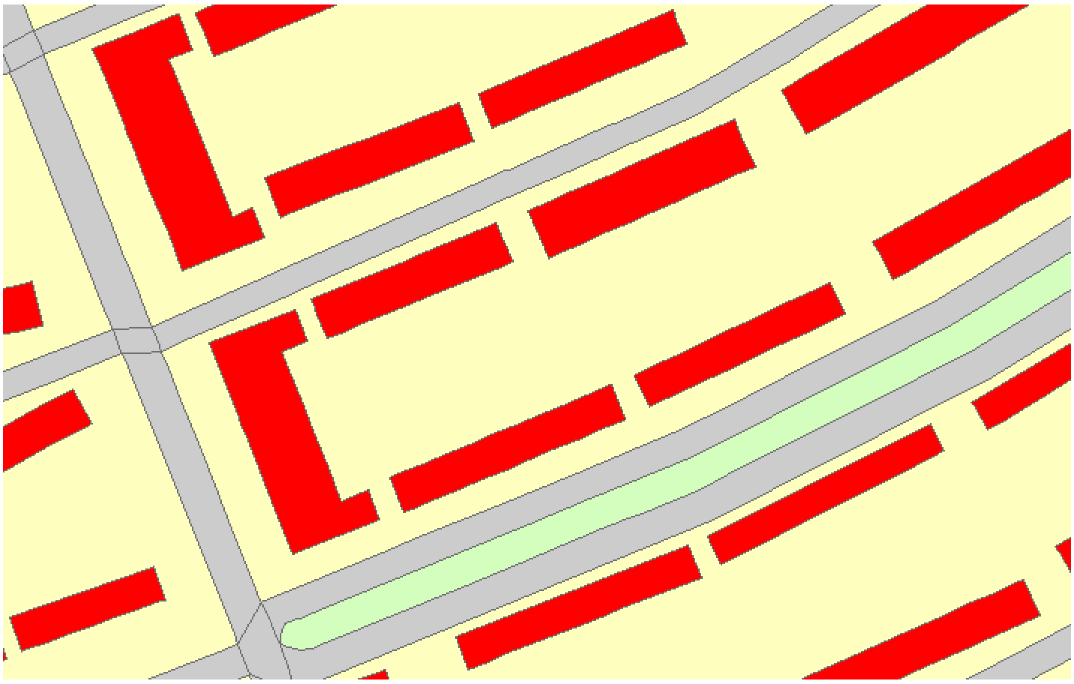

Figure 11: Part of the original Top10NL dataset of Almere.

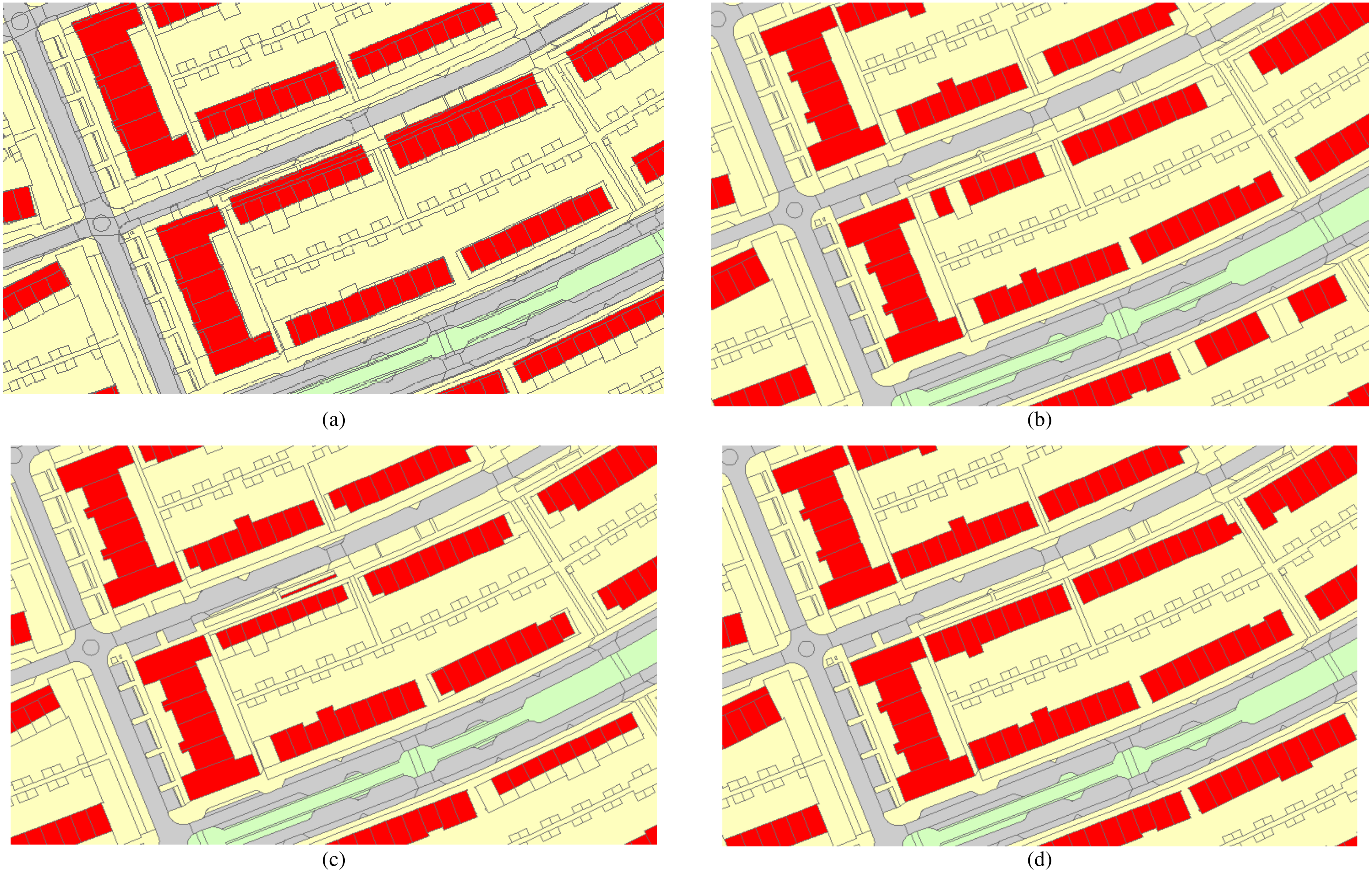

Figure 12: The results of the four methods visualised: (a) Simple overlay, (b) Maximum Area method, (c) Split method, (d) Building First method. The IMGeo objects are coloured according to the region they belong to, which gives an indication of the constrained tGAP end result.

Figures 13-19 show results of constrained tGAP generalization applied to Almere data processed with the building first method. Figure 13 shows the original IMGeo data, which is the input for the constrained tGAP generalization (iteration 0). Figures 14-17 show generalization at intermediate steps: 2000, 3000, 3500,and 3700. Figure 18 shows results of the last step of generalization (iteration 3804), and Figure 19 shows the corresponding Top10NL.

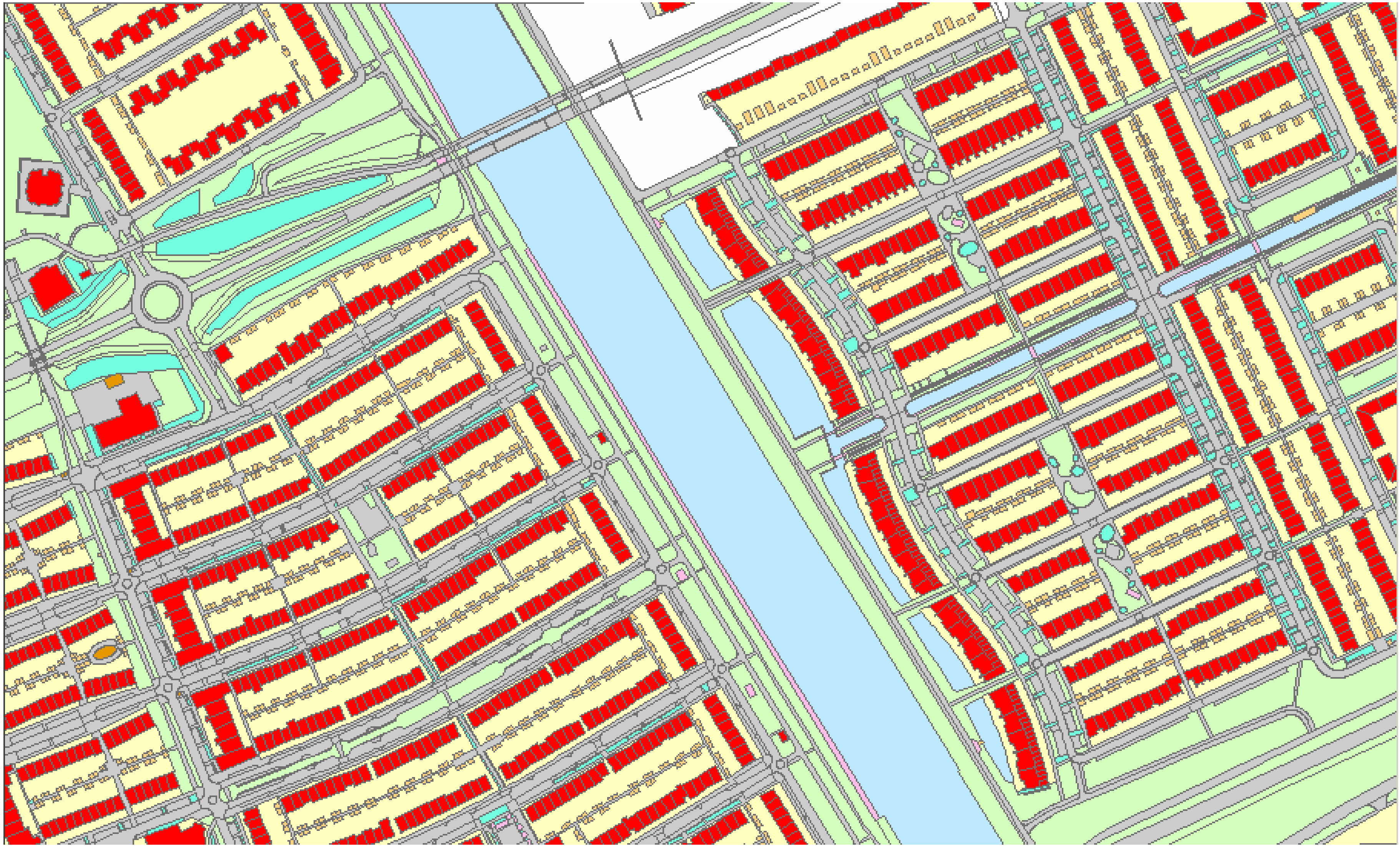

Figure 13: Almere: original IMGeo dataset given as input to the constrained tGAP.

Figure 14: Almere dataset: intermediate result of constrained tGAP at iteration 2000.

Figure 15: Almere dataset: intermediate result of constrained tGAP at iteration 3000.

Figure 16: Almere dataset: intermediate result of constrained tGAP at iteration 3500.

Figure 17: Almere dataset: intermediate result of constrained tGAP at iteration 3700

Figure 18: Almere dataset: final result of constrained tGAP (at iteration 3804).

Figure 19: Almere: the corresponding Top10NL dataset

Appendix C

This appendix shows some results of the tests done with data in the municipality of Rotterdam.

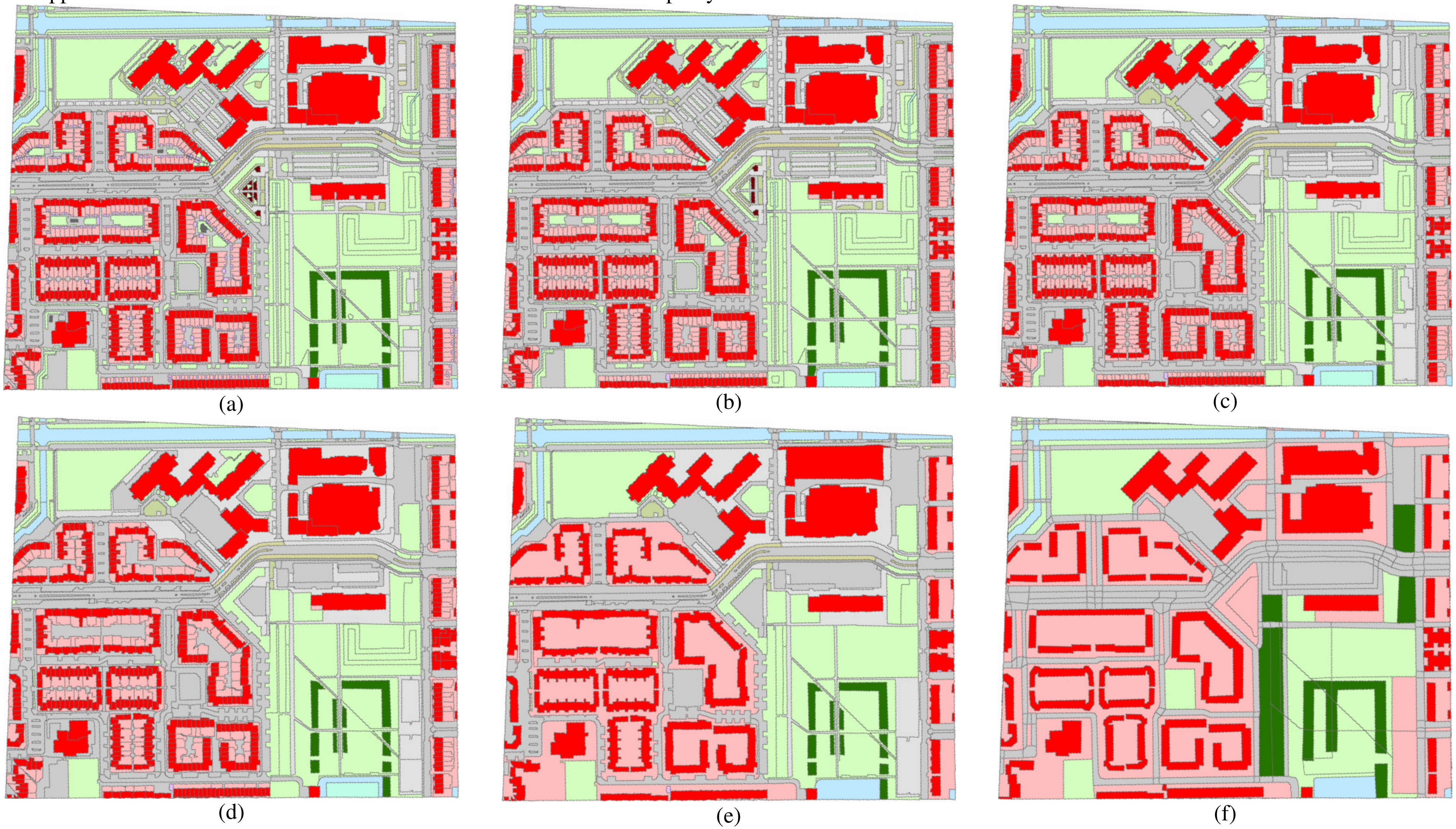

Figure 20: Rotterdam test data: (a) original IMGeo data - input to the constrained tGAP, (b) intermediate result of constrained tGAP at iteration 500, (c) intermediate result at iteration 1000, (d) intermediate result at iteration 1500, (e) final result of constrained tGAP, (f) Top10NL data.