Elsevier Editorial System(tm) for Computers, Environment and Urban Systems
Manuscript Draft

Manuscript Number:

Title: Constrained tGAP for generalisation between scales: the case of IMGeo and Top10NL data

Article Type: Special Issue: Multiple representations

Keywords: map generalisation, constrained tGAP structure, IMGeo data, Top10NL data

Corresponding Author: Dr. Arta Dilo, PhD

Corresponding Author's Institution: Technical University of Delft

First Author: Arta Dilo , PhD

Order of Authors: Arta Dilo , PhD; Arta Dilo, PhD; Peter J van Oosterom, PhD; Arjen Hofman, MSc

Abstract: 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 between the source scales. The solution is based on the constrained topological Generalised Area Partition (tGAP) structure, which stores the results of a generalisation process applied on the large scale dataset, and is controlled by the objects of the medium scale dataset that act as region constraints for the large scale objects. 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 quality 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.

Dear CEUS editor,

Hereby we submit a paper for the special issue on 'Generalisation and Multiple Representation'.

Kind regards
Arta Dilo

(on behalf of all authors)

Constrained tGAP for generalisation between scales: the case of IMGeo and Top10NL data

Constrained tGAP for generalisation between scales: the case of IMGeo and Top10NL data

Abstract

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 between the source scales. The solution is based on the constrained topological Generalised Area Partition (tGAP) structure, which stores the results of a generalisation process applied on the large scale dataset, and is controlled by the objects of the medium scale dataset that act as region constraints for the large scale objects. 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 quality 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.

Key words: map generalisation, constrained tGAP structure, IMGeo data, Top10NL data

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 the large scale map and derive smaller scale maps out of this large scale data automatically. Another advantage is the increased consistency between the different map scales produced in this manner compared to independent map production per scale. 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 map is produced from terrestrial surveying, while the map products at other scales are drawn from aerial photographs. Data for the large scale map, 1:1 000, follow the IMGeo model, a new Dutch model on large scale topography. Data for the medium scale map, 1:10 000, comply with Top10NL, another model on topography.

Top10NL is the current authentic registration on topography in the Netherlands. There are several other 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. Figure 1 shows six authentic registrations, and their relations. In this figure it is shown that the intention of the system of authentic registrations is to relate all authentic registrations. The registration on topography is connected to the cadastral registration and to the registration of buildings.

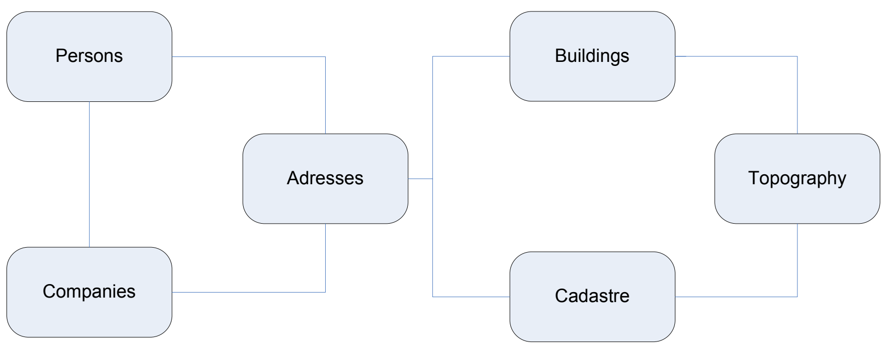


Figure 1: System of authentic registrations in the Netherlands

The research presented in this paper has been performed within the municipality of Rotterdam (Hofman, 2008). Its objective is the generalisation of topographic datasets, starting from the large scale, IMGeo data, to the medium scale, Top10NL data. As a result of generalisation, it assures the connection between the 1:1 000 registration, IMGeo, and Top10NL (1:10 000) registration.

Several solutions to generalisation are suggested in literature, with Multiple Representation Data Bases (MRDB) being a substantial part (Friis-Christensen and Jensen, 2003; Kilpelainen, 1997; National Center For Geographic Information Analysis, 1989). Because nowadays map viewing is often done through web services, which are 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., 2008) is an idea that builds on the topological Generalised Area Partition (tGAP) structure (van Oosterom, 1994, 2005). The constrained tGAP structure stores results of a generalisation performed between two scales: a large scale dataset, which geometry is stored in the structure, and a smaller scale dataset that controls the generalisation process so that the large scale dataset is gradually transformed into the smaller scale dataset. Generalisation is performed on an area partition, thus, it requires the large scale dataset to be an area partition. Area objects of the smaller scale data set act as region constraints in the generalisation process, i.e., they restrict aggregation of large scale objects only inside the region constraints. In the original constrained tGAP (Haunert et al., 2008) the smaller scale regions are computed via an optimization method (Haunert and Wolff, 2006). Because the smaller scale regions are derived from the large scale objects there is a perfect match. In this paper, IMGeo data is fed to the constrained tGAP as the large scale dataset, and Top10NL data form the constraints. As the Top10NL regions are produced independently, there is no perfect match, and a large part of the research concerns matching the large scale objects to the small scale regions.

The rest of the paper is organised as follows. Section 2 introduces the two models used in this research, IMGeo and Top10NL, as well as actual datasets according to these models (in the municipalities of Almere and Rotterdam). Section 3 discusses the processing steps that prepare data for the constrained tGAP generalisation. Section 4 explains how the constrained tGAP works, and Section 5 presents the results of the constrained tGAP generalisation. Section 6 discusses open issues and future research, and Section 7 concludes the paper.

2. Models and datasets

IMGeo and Top10NL are both based on the same Dutch standard on geography, NEN 3610 (Nederlands Normalisatie Instituut (NEN), 2005). This is the Dutch version of the 'General Feature Model' ISO 19109 (International Organisation for Standardization (ISO), 2005). Top class of NEN 3610 is GeoObject that represents any spatial object. Under this general class there are 14 main classes, most of which have their own subclasses. Figure 2 shows the UML diagram of the main classes of NEN 3610.

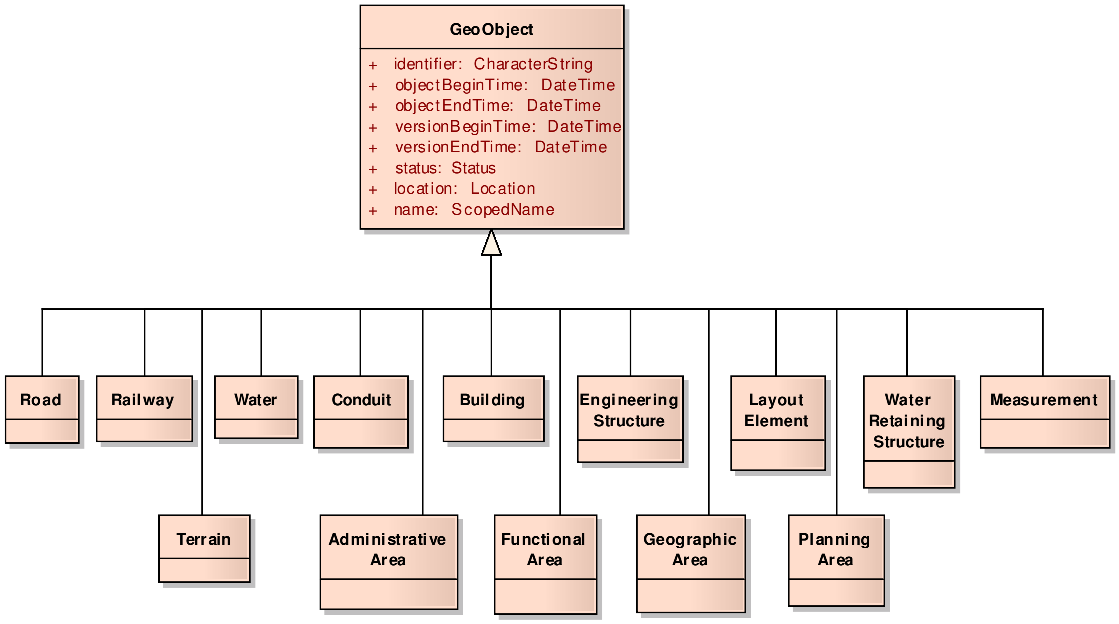


Figure 2: UML diagram of NEN 3610 main classes.

2.1. 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 it will become the new version of GBKN in the near future.

As for NEN 3610, the top object of IMGeo is GeoObject. The main classes of IMGeo are: Road, Railway, Water, Terrain, Building, LayoutElement, EngineeringStructure, AdministrativeArea, corresponding to the NEN 3610 classes with the same name. IMGeo models have fewer classes than NEN 3610. The UML diagram in Figure 3 shows classes of the IMGeo model. Most of the main classes consist each of a subclass that models a part-whole relation, e.g., Road and RoadPart. Class LayoutElement has 11 subclasses, each representing a different kind of a topographic element. Only eight of the subclasses represent area objects, which are the objects of interest for this research: Bin, Sign, Installation, Case, Pylon, OtherBuilding, Separation, and StreetFurniture.

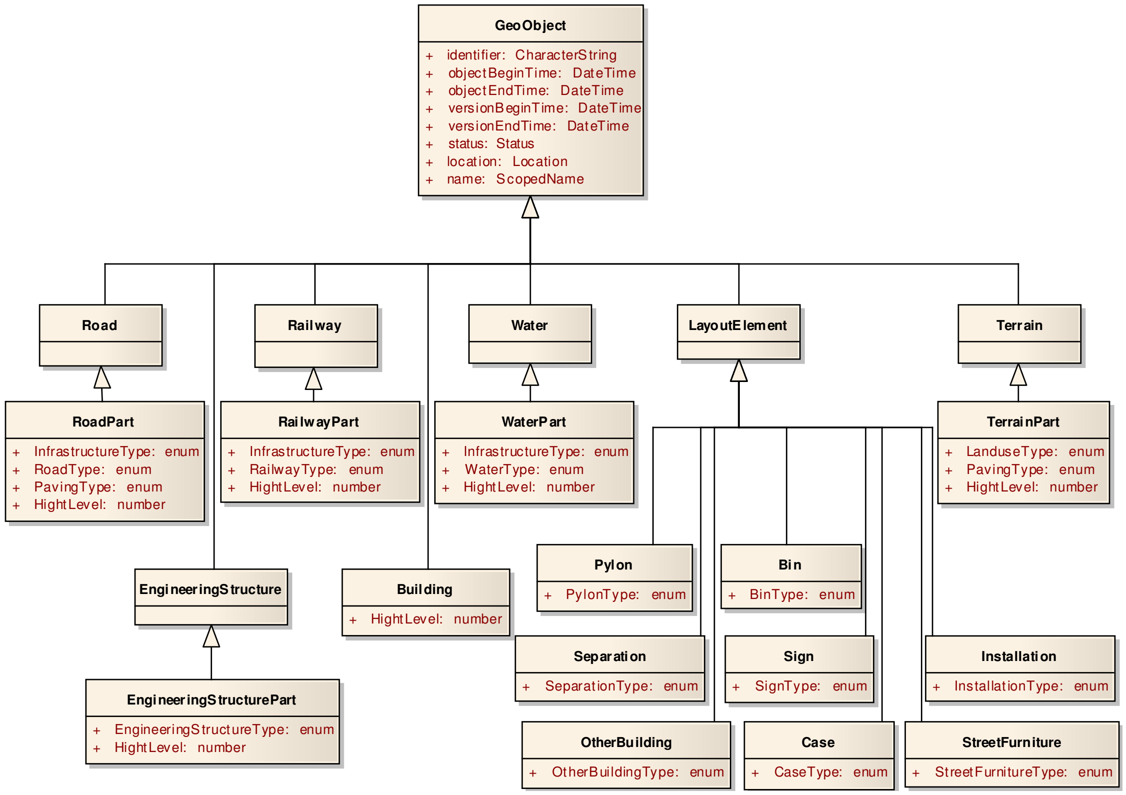


Figure 3: UML diagram of IMGeo classes.

2.2. Top10NL model

Top10NL is the current authentic registration on topography in the Netherlands, which means that every governmental organization is obliged to use Top10NL when working with a 1:10 000 map product. 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 that is the topographic department of the Dutch cadastre.

Top10NL inherits classes of NEN 3610: Road, Railway, Water, Terrain, Building, LayoutElement, AdministrativeArea, FunctionalArea, GeographicalArea. It has an additional class that is not in NEN 3610, Relief. Compared to IMGeo, it has all IMGeo classes, and a few more. There is another difference with IMGeo; class EngineeringStructure is not modelled separately in Top10NL, but this content is included in the LayoutElement class. Figure 4 shows a part of the Top10NL class model; Top10NL class Terrain is a subclass of NEN 3610 class Terrain.

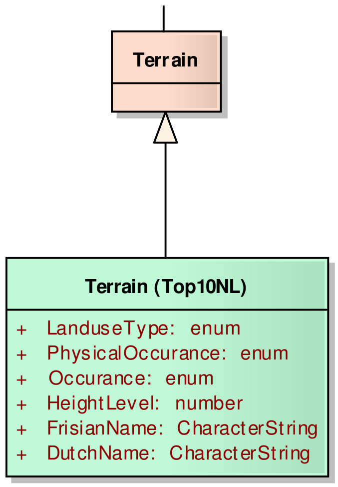


Figure 4: Part of the UML diagram of Top10NL classes: class Terrain of Top10NL is a subclass of the corresponding Terrain class from the NEN 3610 model.

2.3. Differences between IMGeo and Top10NL

Top10NL and IMGeo models are both derived from NEN 3610. NEN 3610 is a general model, containing classes that can be used by different sectors that operate with spatial data. Figure 5 shows a number of derivative sector specific information models (IM's) based on NEN 3610. Among these, there are the two topographic models, IMGeo and Top10NL. There are many other sector specific models, of which the sectors Spatial Planning/Development, Water, and Cultural History are also illustrated in Figure 5. Classes in NEN 3610 are very general, in order to give the sector models a certain amount of freedom to choose what attributes to model. For the models we discuss in this paper, IMGeo and Top10NL, it can be argued that they form the same sector, topography. The distinction is that IMGeo is the model for large scale topography and Top10NL is for medium scale topography. Thus, only a difference in level of detail would be expected. There are though more differences between the models, which lie in the history of modelling.

2.3.1. Historical background

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 (Bakker and Kolk,

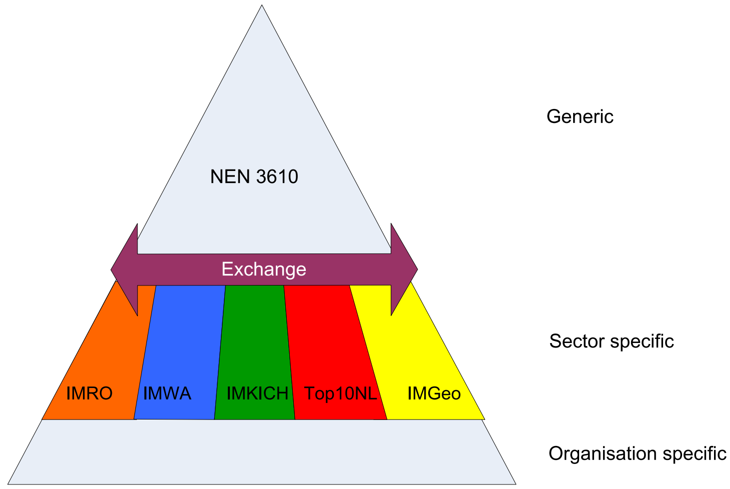


Figure 5: NEN 3610 and a number of its derivatives. Currently at least a dozen of these so called sector information models have been developed and most of them are also in production use.

2003). The predecessor of Top10NL is Top10Vector. Because the Top10Vector model wanted to comply to the OGC standards at the end of last century, they made an object oriented model called Top10NL. Parallel to this, GeoNovum, responsible for the development of NEN 3610, started making this general model object oriented as well. For this reason Top10NL and NEN 3610 developments did influence each other. IMGeo has been created from 2006 and was finalised in October 2007. When IMGeo was made all definitions from NEN 3610 were strictly followed. IMGeo is derived from the Dutch Large Base Map (GBKN), and was designed independently from Top10NL and Top10Vector. The predecessor of IMGeo, GBKN, is a topographic registration, which is used for municipal maintenance organisations like utilities and public environment. Top10Vector was really a topographic map, meant for the user to orientate him/herself.

There are more differences between the two products, related to the data acquisition, and the modelling process. IMGeo is mainly collected by terrestrial surveying, Top10NL is collected from aerial photographs. Top10NL has inherited the NEN 3610 classes; IMGeo only makes reference to NEN 3610 as a source (in the textual description and not in the model). This different background has led to differences is the models.

2.3.2. Differences in the models

In this subsection some of the most striking modelling differences are mentioned. For example, IMGeo is an area partition, Top10NL is not. IMGeo models roads as area objects only, while

Top10NL models centrelines of roads as additional geometry. Class 'Railway' is modelled very different in both models. IMGeo looks at the area around the actual railway and models this as an area (Spoorbaandeel in Dutch) and the actual rails are topographical elements within this area, objects of class LayoutElements. In Top10NL the centreline of the rails is the 'Railway' and the area around it is modelled in the object class 'Terrain'. There are several differences in the kind of LayoutElement objects distinguished in both models. IMGeo is more focused on urban elements, because these are used for municipal registrations; Top10NL has more topographical elements intended to make the map reader be able to orient him/herself in the terrain.

In a topographical sector model one can expect the information of a smaller scale to be less detailed than on a larger scale. Thus, we would expect the Top10NL classes and attributes to be a subset of the IMGeo classes and attributes. At some places in the model this is not the case. For example, Top10NL defines four kinds of wood while in IMGeo only the attribute value 'wood' is distinguished.

2.3.3. Test data

The test data used shows some of the differences mentioned. Data from the municipality of Almere was initially used, because it was one of the few municipalities in the Netherlands that could provide IMGeo data when this research was started 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 difficult the mapping between IMGeo and Top10NL object categories, which are defined from classes and attribute values. Later during the spring of 2008 when IMGeo test data for Rotterdam was produced, tests were performed also on data from the municipality of Rotterdam. Rotterdam data shows the same problems with buildings, but fewer problems with roads.

In Figure 6 it is shown what problems occur when overlaying buildings. 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.

Figure 6: IMGeo buildings (in red) overlaid with Top10NL buildings blocks (in blue). The background is IMGeo data that is made transparent.

However, this inaccuracy does not seem to apply to all objects (as many other object types have better accuracy in the Top10NL).

Figure 7 shows IMGeo roads in grey, overlaid with the transparent Top10NL roads in blue. It can be seen 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.

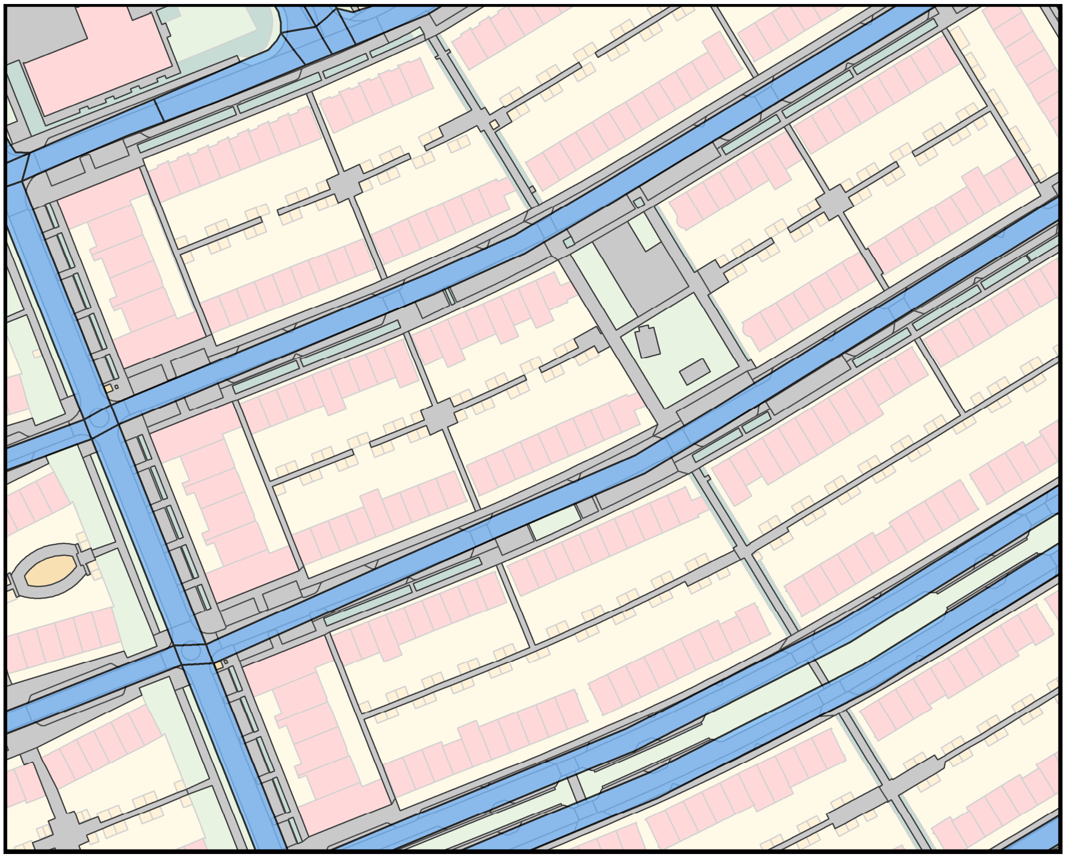


Figure 7: IMGeo roads (in grey) overlaid with Top10NL roads (in blue). The background is IMGeo data that is made transparent.

Besides the geometric discrepancies shown above, the two dataset exhibit semantical differences as well. The IMGeo test data allows categorisation mainly on classes. Only class Terrain can be further categorised based on LanduseType attribute values. Class LayoutElement allows further categorization based on its subclasses. We created a new attribute, 'class', to store these categories: building; road; water; lot, fallow land, plants, other terrain, and grass, from land-use values of class Terrain; bin, and other building as subclasses of LayoutElement class. Top10NL data has more attributes for each of the object classes, but most of them are difficult to compare with the categories created for IMGeo data. Top10NL objects are categorised based on classes, and for object of class Terrain the attribute LanduseType was used for further categorisation. A new attribute, 'region-class' was created for Top10NL data, with the following values: building; road;

water; grassland, wood, and other terrain from land-use values of class Terrain. An example of different classification in IMGeo and Top10NL area areas with lots of larger shrubs. This can be shown in Top10NL as wood. Although Terrain class of IMGeo has a LanduseType value 'wood', the large shrubs are classified as 'plants'.

3. Preparation of data for the constrained tGAP generalisation

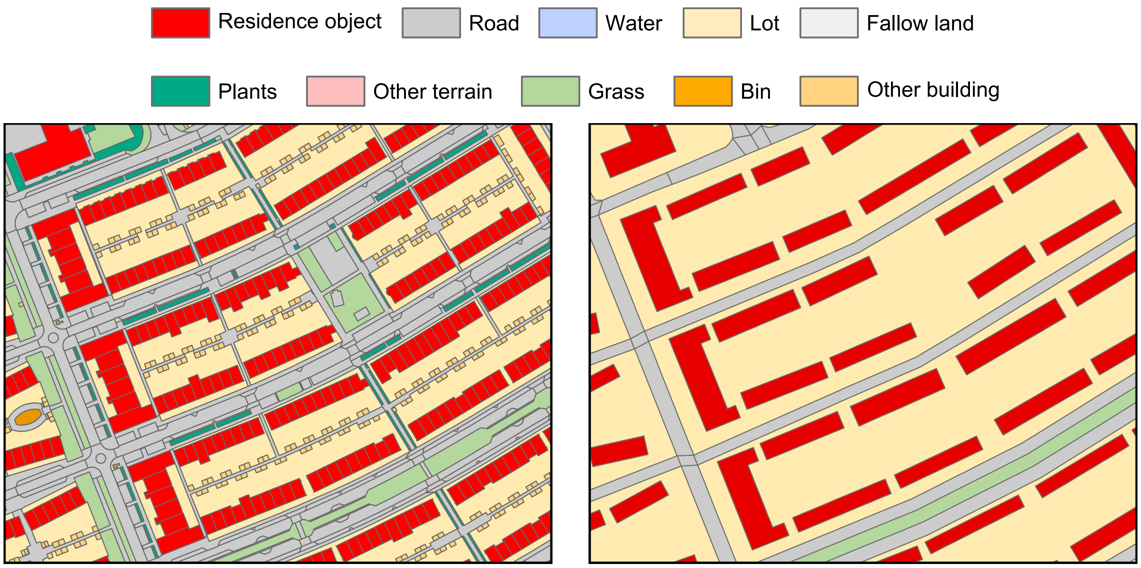
The constrained tGAP has some assumptions, which put requirements on the data we want to generalize. The first assumption is that data has to form an area partition, which was not the case for IMGeo data of Almere and Rotterdam, as overlapping areas did occur. The second assumption is that we know in advance to which region constraint an object of the large scale 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 the IMGeo objects to Top10NL regions before we can run the constrained tGAP code. This section provides the solution for these issues: creating area partitions in Section 3.1, assigning IMGeo objects to Top10NL regions in Section 3.2, and in Section 3.3 creating centres for Top10NL regions from an IMGeo object that holds the same class as the region constraint.

3.1. Resolving overlaps in IMGeo and Top10NL

IMGeo should normally be an area partition, as stated in Section 2.3.2. However this is not the case for the Almere test dataset. This is due to the fact that Almere was a pilot project for IMGeo. Rotterdam data was also prototype IMGeo data, 'pre-production' status, and as such had fewer overlapping problems.

The IMGeo test data is turned to an area partition using an ordering of classes with respect to their importance. Overlaps were happening between terrain objects and objects of any other type; also between roads and water or other buildings. Based on the importance order, the following rules were executed to remove overlaps:

- 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.



(a) IMGeo data

(b) Top10NL data

Figure 8: Area partitions for the IMGeo and Top10NL data, which are used for the constrained tGAP generalisation.

An IMGeo object needs to be assigned to a Top10NL region. To reduce ambiguity, we create first an area partition for the Top10NL data. Resolving overlaps in Top10NL data was performed in a similar manner as for IMGeo. The following rules were executed for creating a partition from Top10NL:

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

The resulting partitions for a part of Almere, after the cleaning operations, can be seen in Figure 8.

3.2. 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. Four possible methods to assign IMGeo objects to Top10NL regions were developed; these are treated separately in the following sections.

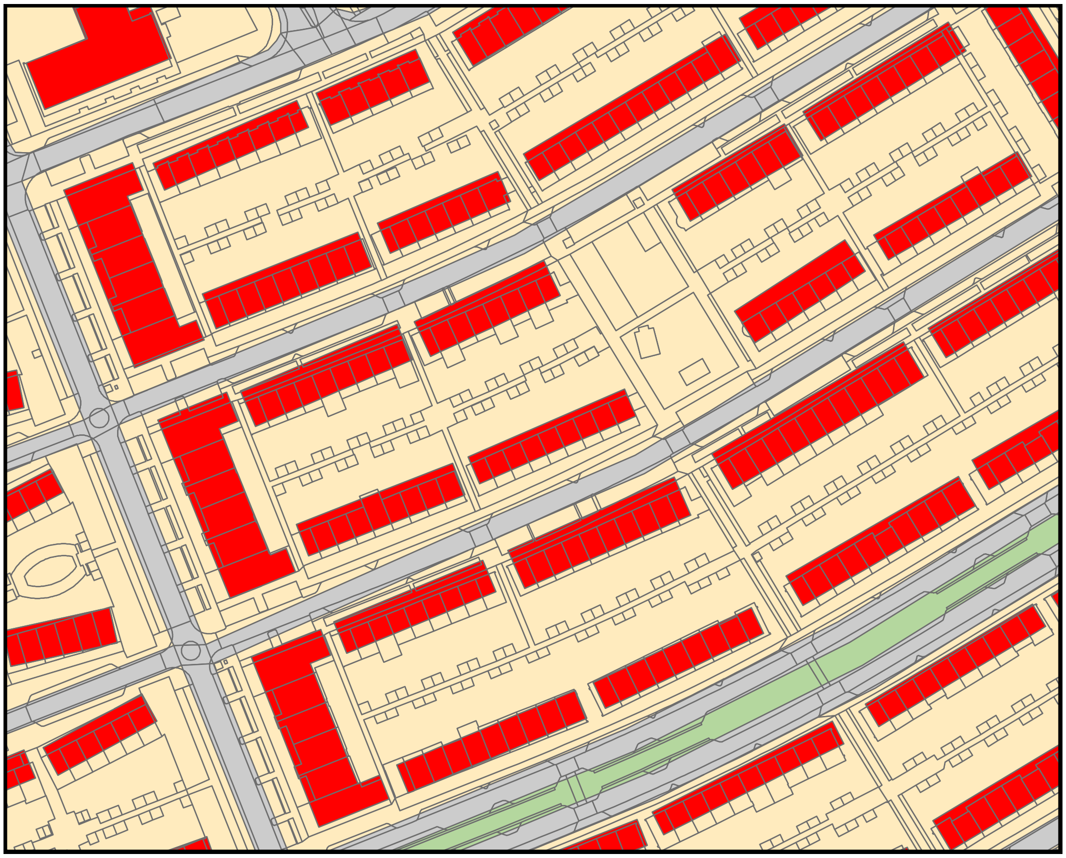


Figure 9: The results of simple overlay method. The IMGeo objects are coloured according to the region they belong to, which gives an indication of the constrained tGAP end result

3.2.1. Simple overlay method

This is the complete intersection between the two datasets, IMGeo and Top10NL data. Every IMGeo object is split at the borders of the overlapping Top10NL region. The resulting dataset contains the geometry of both datasets. Filling the constrained tGAP structure through generalisation starting from this dataset, allows a nice morphing between the IMGeo and Top10NL datasets. The map obtained from the tGAP structure for the scale 1:10 000 would give the Top10NL geometry. Figure 9 shows the overlay dataset, coloured on the class value coming from Top10NL. This gives the idea of how the final result of generalisation looks like.

In this 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. It contains also the coordinates of the line segment intersections.

3.2.2. Maximum area method

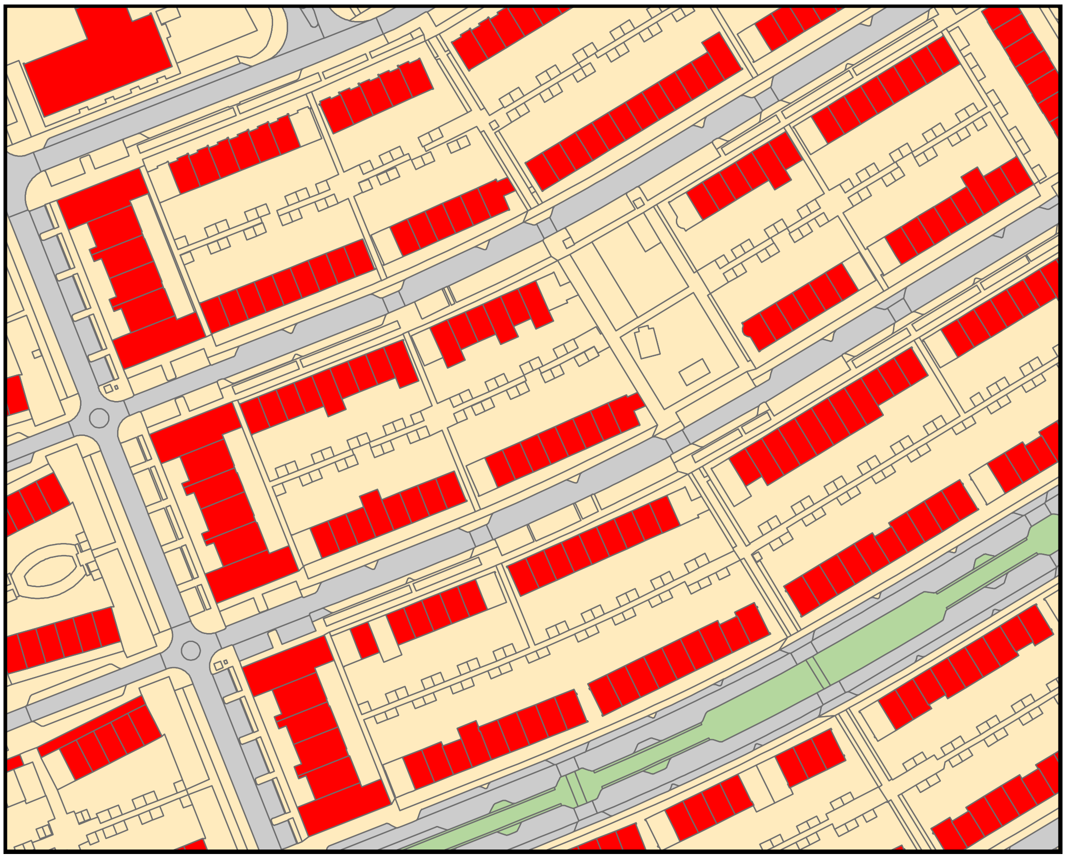


Figure 10: The results of maximum area method. The IMGeo objects are coloured according to the region they belong to, which gives an indication of the constrained tGAP end result.

The Top10NL region which overlaps the IMGeo object the most is the region to which the whole IMGeo object is assigned to. 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 region. In the end result the geometry of IMGeo is kept. Looking at Figure 10 it can be seen 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 (often to terrain), which give quite bad results: missing buildings, as can be seen in Figure 10.

3.2.3. Split method

Figure 11: The results of 35% split method. The IMGeo objects are coloured according to the region they belong to, which gives an indication of the constrained tGAP end result.

If an IMGeo object belongs for more than 35% to two Top10NL regions, we consider the 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. 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. A small overlap between the IMGeo buildings and the Top10NL buildings cause some building blocks in the end result to be too narrow; see Figure 11.

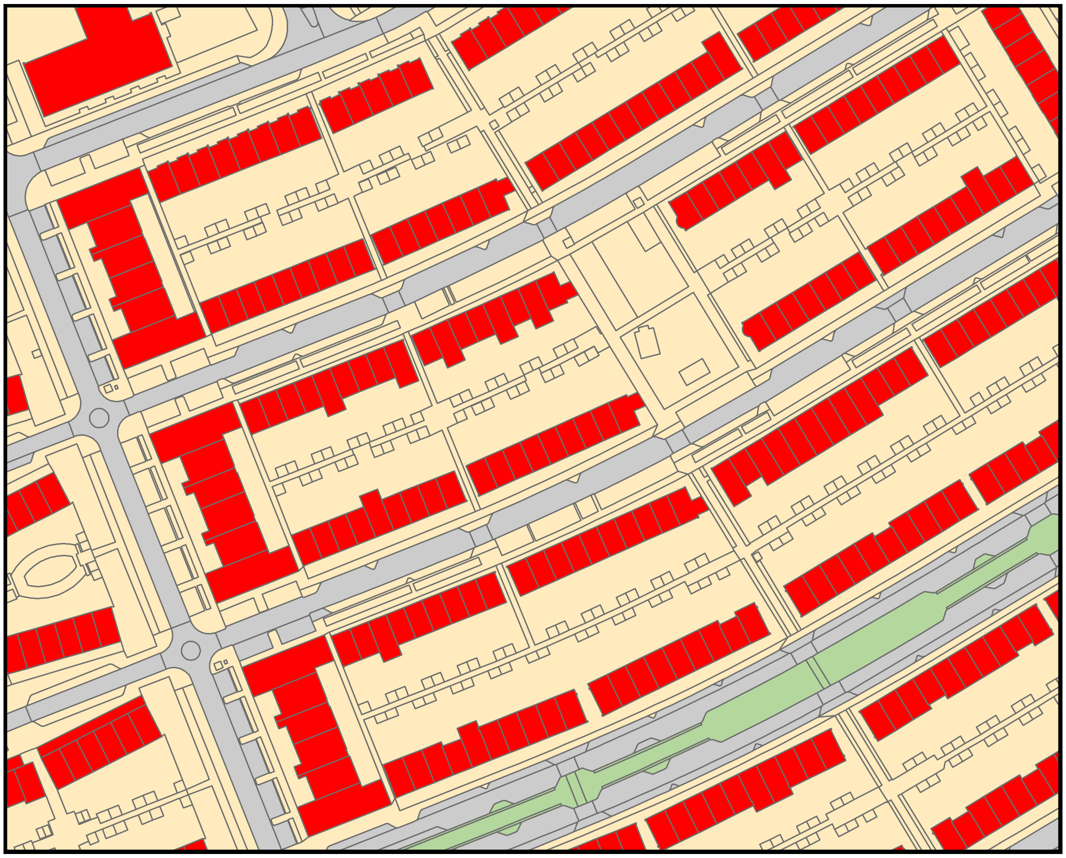


Figure 12: The results of building first method. The IMGeo objects are coloured according to the region they belong to, which gives an indication of the constrained tGAP end result.

3.2.4. Building first method

Since we saw in Section 2.3.3 that the buildings caused problems in the overlay, the problems arising with the previously described methods are not surprising. A fourth method was developed, called 'building first', which checks first all buildings, and assures that an IMGeo building gets assigned to a Top10NL building-block in case of any overlap with a Top10NL building-block, 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), we decided to continue with data resulting from this last method.

3.3. Assigning centres

For every region constraint, that is every Top10NL object, we look for an IMGeo object belonging to the region that has the same class as the region. The largest object that has the same class with the region constraint is called a centre. A centre is used during generalisation to enforce its class value to the new object created from the merging of the centre with another object. This assures that at the end of the generalisation process, the class of the centre (that is also the class of the constraint) is preserved.

There are two categories for Top10NL objects, wood and other terrain, that are not found on IMGeo objects. Top10NL regions belonging to these categories do not have centres. Merging of IMGeo objects that belong to such Top10NL regions is only driven by weight and compatibility values between IMGeo objects. Because the tGAP structure is built from the large scale objects, we cannot arrive in the end with a situation in which we have a totally different object class after merging, which is not occurring in the large scale input dataset.

4. The constrained tGAP

The constrained tGAP is an extension of tGAP (van Oosterom, 1994, 2005). As for the tGAP, the term stands for both a generalisation process, and the structures that store the results of this generalisation. It is currently working with spatial data in 2D, and requires the data to form a partition, i.e., no overlap between area objects, and no gaps. The generalisation process, which is prepared off-line, merges objects of the largest scale to form objects for smaller scales, and keeps track of this merging in the structures. Minimal geometrical redundancy is aimed for, and achieved in two directions: using a topology model for storing objects of the largest scale (avoiding redundant storage of shared boundaries between neighbours at the largest scale); for any object of a smaller scale references are stored to features of the largest scale, which are used to construct this object (avoiding redundancy between scales). The filled structures can be queried efficiently to select features for composing the right objects to be visualised at a given scale.

4.1. Generalisation process and structures

The generalisation process is performed in steps. In each step, 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. Generalisation stops when all objects are merged up to the region constraints. Inside the constraint regions, the generalisation results are driven by the importance and compatibility values of objects. Importance value of an object v is calculated from the area size, and the weight of the object class: $Imp(v) = area\_size(v) \cdot weight(class(v))$. Compatibility between two objects u and v is calculated from length of the shared boundary, and compatibility values between their classes: $Comp(u, v) = bnd\_length(u, v) \cdot compatib(class(u), class(v))$.

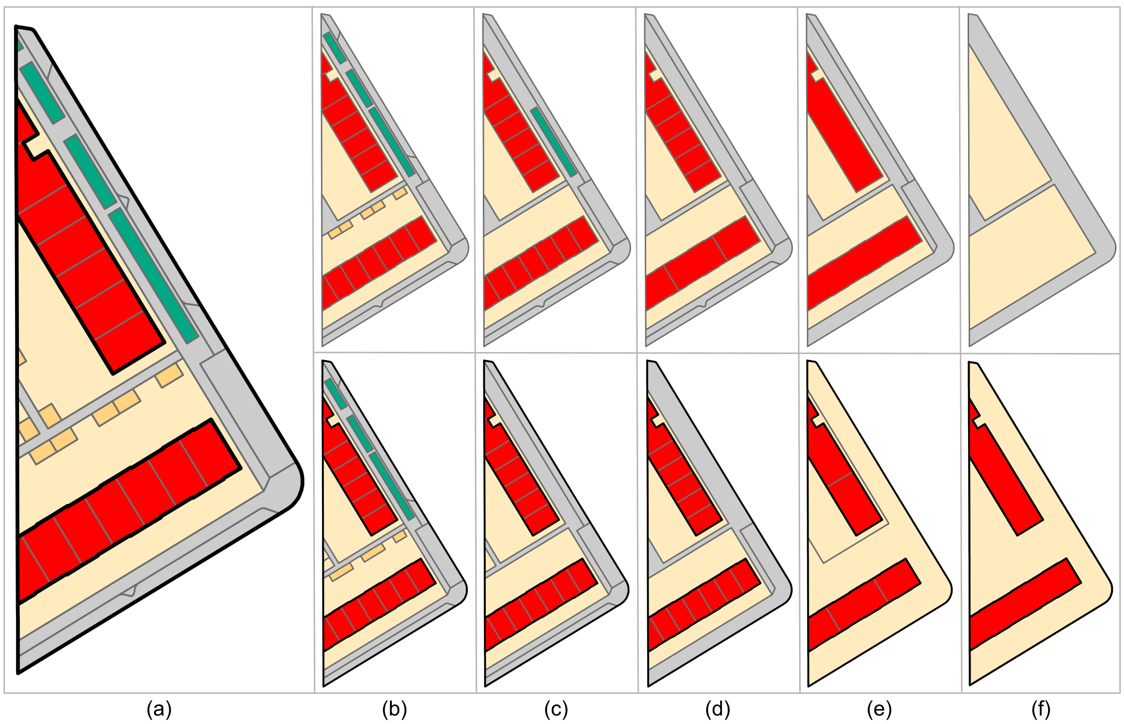


Figure 13: Results of tGAP and constrained tGAP generalisations for the same steps: (a) original data and constraints; (b)–(f) result of step 10, 20, 25, 35, and 39, respectively; first row has results of tGAP generalisation, second row has results of constrained tGAP generalisation. Constraints are shown in thick black line for the constrained tGAP; step 39 is the last for constrained tGAP, and it is also the region constraints.

Figure 13 shows the results of the unconstrained tGAP generalisation in the first row, and constrained tGAP generalisation in the second row, for the same steps. Constraints are shown in thick black line (second row). Figure 13(a) is the largest scale data, a small city block taken from the IMGeo data of Almere. Figures 13(b)– 13(f) show results of generalisation steps 10, 20, 25, 35, and 39, respectively. Step 39 is the last step of the constrained tGAP generalisation, and it is also the map of region constraints. It can be seen that for the constrained tGAP, merging is performed only inside region constraints.

Generalisation results are stored in the constrained tGAP structure, which contains a face and an edge hierarchy (van Oosterom, 2005; Haunert et al., 2008). The topological model that we use consists of faces, edges, and nodes. Each edge holds references to its left and right face, as well as to its start and end node. Geometry is stored for edges and nodes, whereas the geometry of a face is constructed by a topology builder algorithm that collects edges referring to it as left or right face. Generalisation starts at step 0 with all objects (i.e., faces) of the largest scale being valid. A generalisation step merges two objects to a new face; the two merged faces become invalid (i.e., stop existing) in this step, and the validity of the new face start at this step. The starting and the ending validity value is stored for every face during the generalisation process. For every step, we keep trace of changes happening at the boundary edges of the two merged faces. An edge disappears if it is part of the common boundary of the two merged areas. The other edges from the boundary of the two merged areas continue existing, but the reference to the left or right face changes: a new version is created for such edges, the geometry is unchanged, but the left or right face reference is changed at the current step. An alternative would be to also merge to edges (adjacent to the shared edge which is removed) in cases where this is possible, and create longer edges for the smaller scales, as described in the original GAP-tree proposal (van Oosterom, 1994, 2005). A validity range is recorded for every version of an edge. The next section describes how this information is stored in Oracle Spatial.

4.2. Implementation in Oracle Spatial

The information for the constrained tGAP structures is stored in Oracle 10g Spatial. Figure 14 shows the UML diagram of tables that store the constrained tGAP information in Oracle Spatial. Arrows associating tables show foreign key relationships; cardinalities are shown when different from 1. Primary keys and foreign keys are shown by symbols PK, FK, and pfK symbolises a foreign key that is part of a primary key.

Information about area objects is stored in ⟨dataset⟩_Face table: an identifier, the class to which it belongs, region constraint, an attribute with value 'Y' or 'N' that defines whether the area is a centre, the area size, and the validity range as [imp_low, imp_high). Information about edges is split in two tables: ⟨dataset⟩_EdgeGeo and ⟨dataset⟩_Edge. The first table contains an identifier for an edge, references to start and end node, the geometry, and its length. Table

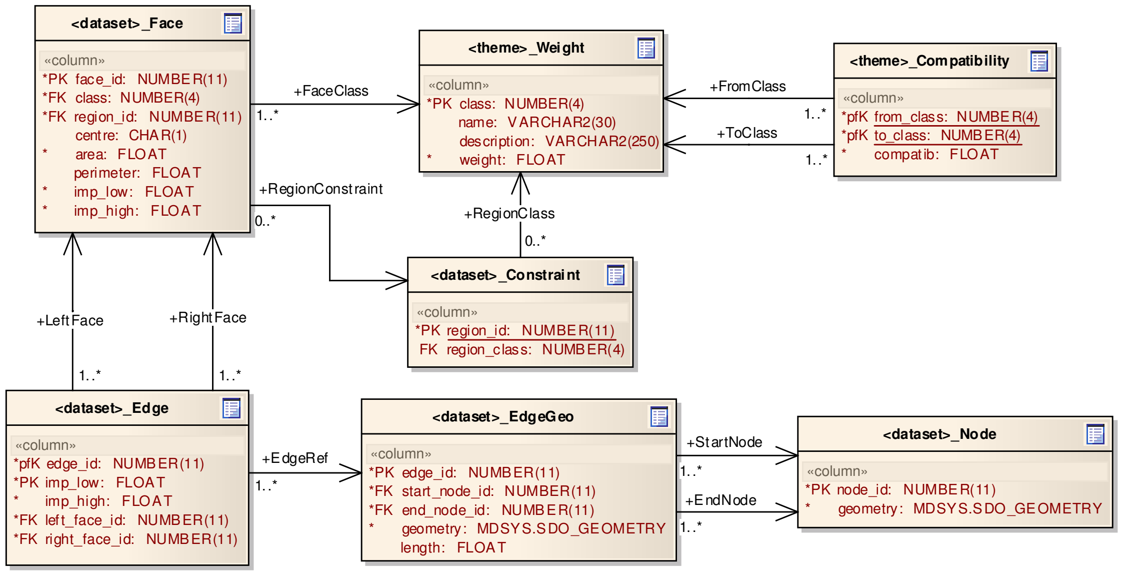


Figure 14: UML diagram of tables and relationships that store the constrained tGAP information in Oracle Spatial. PK shows a primary key, FK shows a foreign key, and pfK shows a foreign key that is part of a primary key.

⟨dataset⟩_Edge stores left-, right-face of an edge as they change during the generalisation, and the corresponding validity range [imp_low, imp_high); the combination edge_id, imp_low is unique and it is the primary key of the table. The reason for splitting is that a part of the edge information, ⟨dataset⟩_Edge, does change frequently as result of the creation of the tGAP structure and the other part, ⟨dataset⟩_EdgeGeo, is static. Node information is stored in ⟨dataset⟩_Node table. Table ⟨theme⟩_Weight stores information about classes: code as referred in Face table, name and description, as well as class weight. Table ⟨theme⟩_Compatibility stores the compatibility value of changing from the from_class to the to_class.

The constrained tGAP generalisation is implemented as PL/SQL code, which algorithm is given below. It starts with the largest scale data that is stored in a left-right topology model, and forms an area partition. Face and edge information, as well as information of class weights and compatibilities, is read from source data tables into arrays in memory. Face array is sorted by importance value, and is always kept in this order after removal of merged faces and insertion of new faces (a priority queue). The first element of the array is processed in each iteration, as it has the lowest importance value. Edge array is correspondingly updated, by removing common edges, and updating the references for the other boundary edges. Changes happening at each generalisation step are reflected in the ⟨dataset⟩_Face and ⟨dataset⟩_Edge tables.

```
/* Read source data tables by executing SQL statements */
fill face array with info: id, class, importance, region, centre, area
fill edge array with info: id, left-face, right-face
get class weights and compatibilities into memory arrays
while face array is not empty do
    get neighbours of face[1] that are in the same region constraint into neighbours array
    if neighbours array is not empty then
        get the most compatible neighbour from neighbours into best-nbhd
        compile information for the new face
        remove the two merged faces from face array
        insert the new face in the right order in the face array
        insert the new face into the Face table
        update edge array        /* consequence of merging the two faces */
        update the Edge table
    else                         /* face[1] is equal to a region constraint */
        remove face[1] from the array
    end if
end while
```

5. Performing the constrained tGAP generalization for IMGeo and Top10NL data

Having the area partition of Almere and a region constraint for each object, as explained in Section 3, we can apply the constrained tGAP generalisation using a set of default values for weights and compatibilities: equal weights for classes, which means importance of objects is only based on the area size; compatibility value equal to 0.1 for objects of different classes, and compatibility value 1 for objects of the same class. For tGAP generalisation in Figure 13 we used these default values. We applied these values for the whole Almere dataset. The results were not satisfactory, therefore we looked into improving the weight and compatibility values.

5.1. Calculation of weights and compatibilities

The first values for weights and compatibilities referred to the work of 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 trial and error (again based on visual inspection of the results). The optimum found for the weights and the compatibilities are given in Table 1. 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 5.2.

From-class code ⟶		1001	2001	3001	4001	4002	4003	4004	4005	5001	5002
	Weight	13.0	1.20	1.30	9.00	1.00	0.93	0.90	0.88	0.10	1.00
Class name	**To-c**↓										
Building	1001	1.00	0.50	0.00	0.99	0.90	0.50	0.50	0.00	0.00	0.99
Road	2001	0.50	0.99	0.00	0.50	0.50	0.95	0.90	0.50	0.95	0.50
Water	3001	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Lot	4001	0.90	0.95	0.00	1.00	0.90	0.90	0.90	0.80	0.90	0.95
Fallow land	4002	0.90	0.90	0.00	0.95	1.00	0.90	0.90	0.50	0.80	0.95
Plants	4003	0.50	0.50	0.00	0.80	0.50	0.99	0.95	0.90	0.90	0.50
Other terrain	4004	0.90	0.50	0.00	0.80	0.90	0.00	1.00	0.99	0.95	0.95
Grass	4005	0.50	0.90	0.00	0.80	0.50	0.95	0.95	0.99	0.95	0.50
Bin	5001	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Other Building	5002	0.99	0.50	0.00	0.50	0.90	0.50	0.50	0.00	0.00	1.00

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

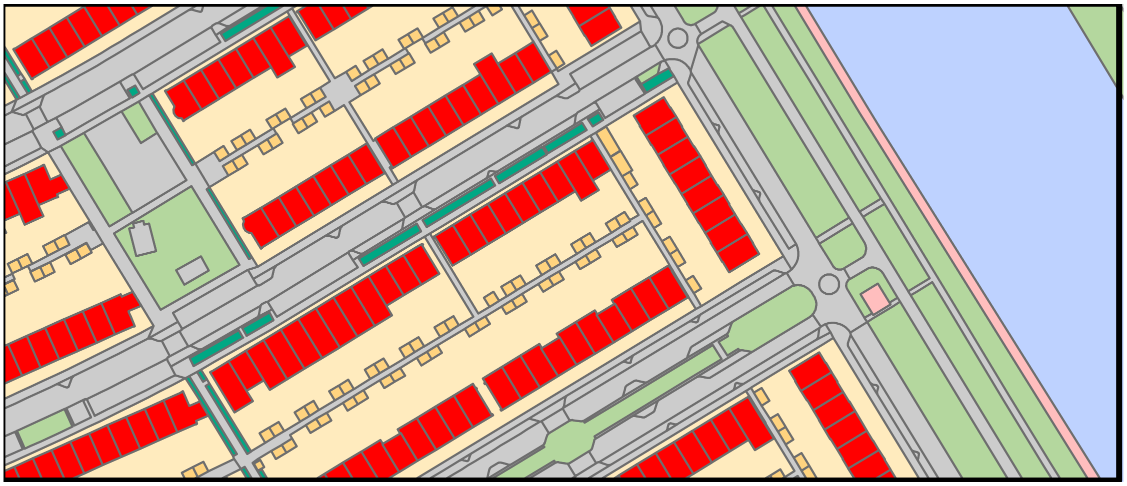
5.2. Results of generalization

Using the prepared input data, together with the weight table and compatibility matrix, we can apply the constrained tGAP generalization for IMGeo data. The PL/SQL code of the constrained tGAP fills the tables with results of generalization. Figures 15 and 16 show the same part of Almere, for comparison: the largest scale IMGeo (1:1 000) in 15(a); the result of constrained tGAP generalization for scale 1:5 000 in 15(b); the result of constrained tGAP generalization for scale 1:10 000 in 16(a); and the corresponding Top10NL data in 16(b). Figure 16(a) shows the results of this research. Comparing it to Figure 16(b), 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 16(a) are quite a good generalization of the original data in Figure 15(a). Based on the tGAP structure also all intermediate scales can be obtained. Examples of constrained tGAP generalisation applied to Rotterdam data are given in the Appendix A.

6. Future research

In this section we describe the open issues we encountered when conducting the research described in this paper. In many cases we also propose suggestions for resolving these open issues, but most of these are not yet implemented or tested in our current system. These issues are:

- Non neighbouring objects within one region. An approach for this situation has been suggested in (Ai and van Oosterom, 2002). It basically means that a least important object is merged

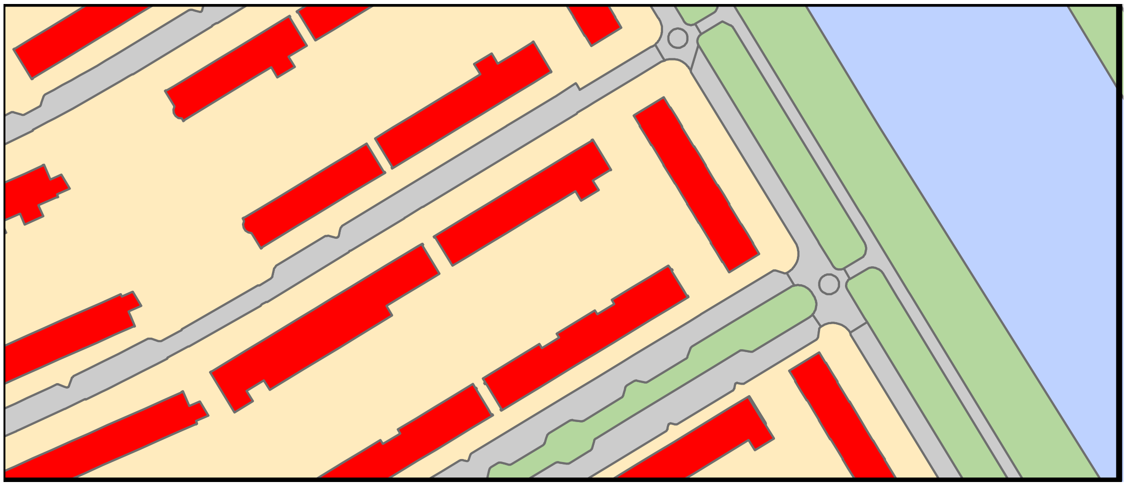


(a) IMGeo data

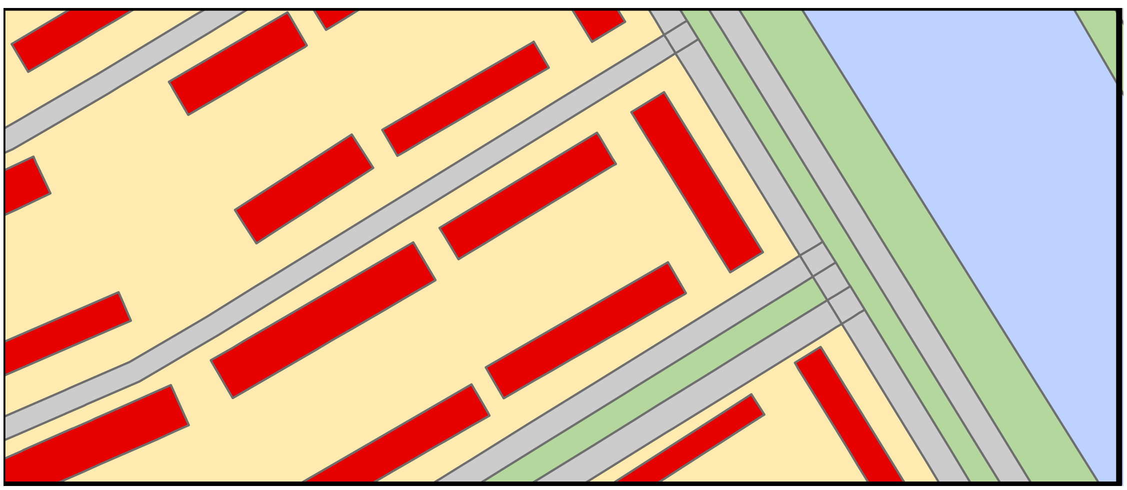
(b) step 3408

Figure 15: Results of constrained tGAP generalisation for Almere data: (a) original IMGeo data, (b) result of constrained tGAP generalisation for scale 1:5 000.

with a compatible object that is not the direct neighbour. This implies that the space lying in between these two objects is also included in the aggregated class. Via a triangulation the space in between is computed and after that one can check which third object is involved (or more objects are involved) and if this would be allowed. Most likely only a part of this third object is needed and therefore the third object is split. This part of the third object is



(a) step 3834



(b) Top10NL data

Figure 16: Results of constrained tGAP generalisation for Almere data: (a) result of constrained tGAP generalisation for scale 1:10 000, (b) Top10NL data for the same part of Almere.

included in the aggregation with the other two objects.

- Non-aggregated classes in smaller scales. When studying the content of the Dutch topographic maps at the different scales, in a few exceptional cases a complete new class occurs. And example of such a class is an air-space region. These regions are quite large and do not make sense to be displayed in a large scale map fragment as most likely one of the three options does occur: map fragment is completely outside air-space region, map fragment is completely inside air-space region, or a part of the boundary of air-space region can be seen. This does not make much sense, therefore air-space regions are only displayed at smaller scales.

- Inconsistencies between input data from different scales, due to reasons other than accuracy issues, e.g., temporal. These might be partially solved by using spatio-temporal topographic data sets and using a moment in time for which all topographic scales are available. After creating the constrained tGAP with these data, the newer data of the largest scale could be used to update the tGAP structure and propagate significant changes upwards to the smaller scale objects.

- Preference for end-region classification when creating constrained tGAP. Currently, it is possible that after merging all the objects assigned to a region, the class of the last object is different from the target region. By having 'localized' importance and compatibility functions a natural path to the classification of the target region might be obtained, e.g., objects with classification similar to the target classification are made more important and in the compatibility function the objects with the same class as the target region are made more attractive.

- Differences in classification systems used at the different scales, e.g., roundabout on 1:10 000 and not on 1:1 000, urban-lot on 1:10 000 and separate buildings and yards on the 1:1 000. Define special importance and collapse functions for these special cases, e.g., high compatibility of yard to adjacent building. Also take the decision to reclassify at a certain moment. In the traditional tGAP structure always the class of the winner (most important) object is kept; e.g., building. In case of a composition with its own classification at a smaller scale, this classification might be used further; e.g., urban-lot.

- From the generalization results shown in previous section we see the need for line simplification. The use of Douglas-Peucker algorithm (Douglas and Peucker, 1973) for line simplification is proposed in (van Oosterom, 1994), and its implementation is given in (Meijers, 2006). Simplification of buildings may require other techniques for better results. We are currently implementing the Visvalingams algorithm for line simplification (Visvalingam and Whyatt, 1993). Investigation of best results for line simplification is part of future research.

- Another direction for future research is the automatic propagation of updates performed at the largest scale to the medium scale, and more generally the propagation of updates in the tGAP structure. This means that the structure would become a dynamic structure. Also, if the past states are not forgotten, but included, then the result will be a vario-scale spatio-temporal structure.

- Inclusion of scales smaller than 1:10 000 in the constrained tGAP structure; e.g., such as the 1:50 000 and 1:250 000 scales.

The last issue will be discussed in a bit more detail. The approach described in this paper to merge the 1:1 000 and 1:10 000 topographic data in one structure could also be applied between other scales; e.g., 1:10 000 and 1:50 000, or between 1:50 000 and 1:250 000 (or even more fixed scales). The four methods to relate a large scale object to a smaller scale region can be applied between every pair of scales. For the simple overlay method the result is overlaying all geometry from all scales. This causes a lot of fragmentation and therefore this method does not seem to be appropriate for this situation, despite the fact that it is an attractive approach to morph between two fixed scales. The maximum area method does only use geometry from the largest scale and therefore does not have a problem with the fragmentation. The order of creation the overall constrained tGAP structure, bottom-up (starting with largest scales) or top-down (starting with smallest scales), does influence the overall result. The reason for this is the following: the original 1:10 000 geometry (see Figure 16(b)) is different from the derived constrained tGAP 1:10 000 geometry (see Figure 16(a), which is based on 1:1 000 geometry) and this does influence the process of relating the 1:10 000 object to the 1:50 000 regions as the geometry of the two 1:10 000 scales is different. Probably the bottom-up approach is a better, because decisions are based on the geometry which is actually used to represent the larger scale object in the constrained tGAP structure. However, one could also argue that top-down is a good approach as the original geometries may carry the best information to take the decision which set of objects is assigned to the (next higher level) region. More testing is needed before a definite decision can be taken on the best approach.

In case of the 35% split method, some geometry of the smaller scale is introduced at the larger scale. Because of this reason, a top-down approach is proposed as in theory some geometry from the smallest scale might be pushed down via the intermediate scales until the largest scale is reached. It is an unlikely situation as this geometry was originally only present at the smallest scale and not at the larger scales and at the same time it is supposed to be significant according to the 35% rule. As the 35% split method only applies to the objects that fall under the 35% rule, the other objects follow the maximum area method, but now also in a top-down manner.

The building first method is a bit specific for the larger scales, as on the smaller scales no individual buildings are represented any more. Despite this fact, this type of rule could also be applied when mapping objects between two other fixed scales, e.g., to maintain a very important object class as good as possible: assign this to a region of the same class in case there is any amount of overlap. As the larger scale geometry is used to represent the smaller scale composite objects (region counterparts) in the tGAP structure, it again does seem more fair to follow a bottom-up approach, because of the same reasoning as in case of the 35% split method. Again it must be stressed that more testing is needed.

It should be remembered that whichever method of coupling larger scale objects to smaller scale regions (all terms are relative), it is the final result that is important: the constrained tGAP that integrates several fixed scales in a truly vario-scale structure. Therefore, after the constrained tGAP creation, the smaller scales and can be thrown away. Only the largest scale and the resulting constrained tGAP structure (which has encapsulated the human cartographer knowledge) have to be maintained from this point onwards. It should further be noted that as the range of scales becomes larger, the need for line generalization is only increasing, therefore it becomes more urgent to also create the BLG-tree, either based on Douglas-Peucker or other techniques, see (van Oosterom, 2005).

7. Conclusions

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 the Top10NL model has been made. To be able to really integrate IMGeo and Top10NL in the constrained tGAP structure cooperation between the makers of the models will be necessary in order to remove model inconsistencies, which can not be solved via automatic conversions.

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 more and 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. Though the constrained tGAP structure has initially been tested and optimized for Almere data, is was later applied to Rotterdam data. Despite the fact that this was quite different, again the results were satisfactory, indicating that the approach is quite generic.

In this research it was shown that the generalization quality has much improved by using the Top10NL regions (compared to the unconstrained generalization of the IMGeo data). Another result from the constraint tGAP investigations would be learning from generalization process using the region constraints and then observing how tGAP creation is different from the unconstrained tGAP creation; what kind of aggregations are made with which composition and frequency. For example, tuning of compatibility and weight values could be directed by collecting statistics obtained from the generalization under the region constraints. 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.

It is the intention that in the future only the largest scale in a dynamic constrained tGAP structure is updated and that the cartographer checks the automatically generated propagation of changes to the smaller scales (higher levels in the tGAP structure). In most cases this is expected to be of sufficient quality. If not, then the structure could be corrected manually in the tGAP structure by changing the incorrect selections and aggregations. Besides a more efficient updating procedure for the geo-information (at different scales) this also guarantees consistency between scales and also all intermediate scales can be used; e.g., for smooth zooming or for progressive transfer (Haunert et al., 2008; Meijers, 2006).

References

Ai, T., van Oosterom, P., 2002. GAP-tree extensions based on skeletons. In: Richardson, D., van Oosterom, P. (Eds.), Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling. Springer, Berlin, Germany, pp. 501–513.

Bakker, N., Kolk, B., August 2003. Top10NL, a new object-oriented topographical database in GML. In: Proceedings 18th International Cartographic Conference (ICC'03), Durban, South Africa.

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

Friis-Christensen, A., Jensen, C. S., 2003. Object-relational management of multiply represented geographic entities. In: SSDBM'2003: Proceedings of the 15th international conference on Scientific and statistical database management. IEEE Computer Society, Washington, DC, USA, pp. 183–192.

Haunert, J.-H., Dilo, A., van Oosterom, P., 2008. Constrained set-up of the tGAP structure for progressive vector data transfer. Submitted to Computers & Geosciences.

Haunert, J.-H., Wolff, A., 2006. Generalization of land cover maps by mixed integer programming. In: Proceedings 14th International ACM Symposium on Advances in Geographic Information Systems (ACM-GIS'06). pp. 75–82.

Hofman, A., February 2008. Developing a vario-scale imgeo using the constrained tgap structure. Master's thesis, TU Delft.
URL http://www.gdmc.nl/publications/2008/Vario-scale_IMGeo_using_constrained_tGAP.pdf

International Organisation for Standardization (ISO), 2005. ISO 19109:2005 Geographic Information - Rules for Application Schema. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39891.

Kilpelainen, T., 1997. Multiple representation and generalisation of geo-databases for topographic maps. Ph.D. thesis, Finnish Geodetic Institute, Finland.

Meijers, M., June 2006. Implementation and testing of variable scale topological data structures. Master's thesis, TU Delft.
URL http://www.gdmc.nl/publications/2006/Variable_scale_topological_structures.pdf

National Center For Geographic Information Analysis, 1989. The research plan of the national center for geographic information and analysis. International Journal of Geographical Information Science 3 (2), 117–136.

Nederlands Normalisatie Instituut (NEN), 2005. Basis model geo-informatie, NEN 3610. http://www2.nen.nl/nen/servlet/dispatcher.Dispatcher?id=253335.

van Oosterom, P., 1994. Reactive Data Structures for Geographic Information Systems. Oxford University Press, UK.

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.

van Putten, J., van Oosterom, P., 1998. New results with generalized area partitionings. In: Proceedings of the International Symposium on Spatial Data Handling, SDH'98, Vancouver, Canada. pp. 485–495.

Visvalingam, M., Whyatt, J. D., 1993. Line generalization by repeated elimination of points. Cartographic Journal 30 (1), 46–51.

A. Results of constrained tGAP for Rotterdam data

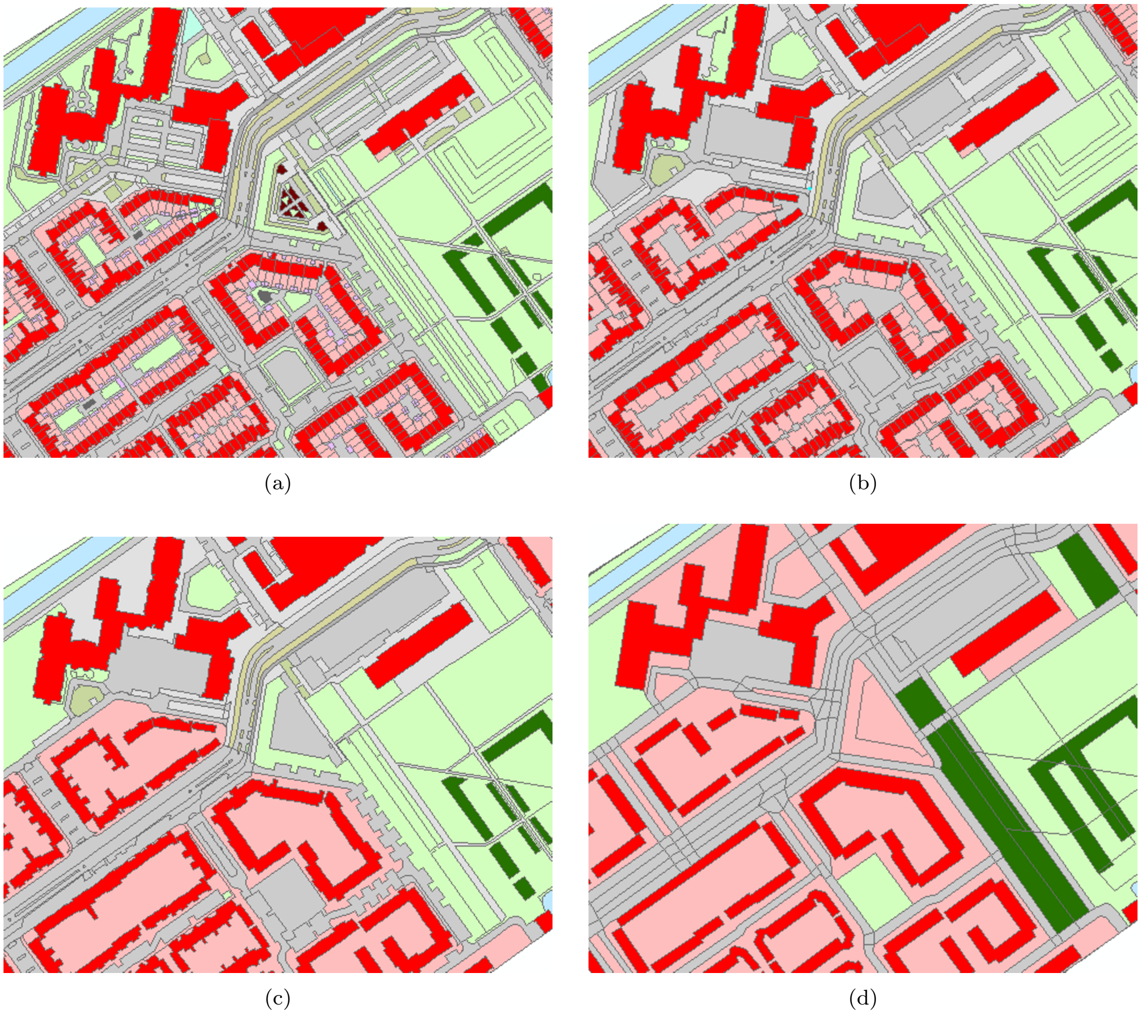


Figure 17: Results of constrained tGAP generalisation for Rotterdam data: (a) original IMGeo data, (b) results of generalisation for scale 1:3 000, (c) results of generalisation for scale 1:10 000, (d) corresponding Top10NL data.

* Title Page (WITH author details)

Constrained tGAP for generalisation between scales: the case of IMGeo and Top10NL data

Arta Dilo*, Peter van Oosterom

OTB, Section GIS-technology, Delft University of Technology, Jaffalaan 9, 2628 BX Delft, the Netherlands

Arjen Hofman

Logica Nederland B.V., Geo ICT Public Sector, Laan van Zuid Hoorn 70, P.O. Box 3190, 2280 GD Rijswijk, the Netherlands

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 between the source scales. The solution is based on the constrained topological Generalised Area Partition (tGAP) structure, which stores the results of a generalisation process applied on the large scale dataset, and is controlled by the objects of the medium scale dataset that act as region constraints for the large scale objects. 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 quality 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.

*Corresponding author

Email addresses: A.Dilo@tudelft.nl (Arta Dilo), P.J.M.vanOosterom@tudelft.nl (Peter van Oosterom), arjen.hofman@logica.com (Arjen Hofman)

Preprint submitted to Elsevier *November 1, 2008*