

Constrained tGAP for generalization between scales: the case of Dutch topographic data

Arta Dilo, Peter van Oosterom*

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

Arjen Hofman

*Logica Nederland B.V., Geo ICT Public Sector,
P.O. Box 3190, 2280 GD Rijswijk, the Netherlands*



*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)

Topographic data provide a general description of the earth's surface, including roads, rivers, buildings and vegetation; more explanation in Section 3.

In order to avoid conflict of interest, Jean-Claude Thill, Editor-in-chief of Computers, Environment and Urban Systems did edit this paper.

Constrained tGAP for generalization between scales: the case of Dutch topographic data

Arta Dilo, Peter van Oosterom*

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

Arjen Hofman

*Logica Nederland B.V., Geo ICT Public Sector,
P.O. Box 3190, 2280 GD Rijswijk, the Netherlands*

Abstract

This article presents the results of integrating large- and medium-scale data into a unified data structure. This structure can be used as a single non-redundant representation for the input data, which can be queried at any arbitrary scale between the source scales. The solution is based on the constrained topological Generalized Area Partition (tGAP), which stores the results of a generalization process applied to the large-scale dataset, and is controlled by the objects of the medium-scale dataset, which act as constraints on 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 constraint tGAP structure. The advantage of this constrained approach over the original tGAP is the higher quality of the aggregated maps. The idea was implemented with real topographic datasets from the Netherlands for the large- (1:1 000) and medium-scale (1:10 000) data. The approach is expected to be equally valid for any categorical map and for other scales as well.

Key words: automatic map generalization, object matching, constrained tGAP

*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)

Topographic data provide a general description of the earth's surface, including roads, rivers, buildings and vegetation; more explanation in Section 3.

In order to avoid conflict of interest, Jean-Claude Thill, Editor-in-chief of Computers, Environment and Urban Systems did edit this paper.

1. Introduction

In recent years we have seen the increased use of digital maps on the Internet. A multitude of web applications are built upon maps. A typical requirement of such applications is the adaptation of maps to a Level of Detail (LoD) that is appropriate for the extent shown on the screen at any time. The representation of a small area in detail results in a so-called large-scale map, while the representation of a large area in lesser detail is called a small-scale map (Jones and Ware, 2005). Traditionally, the generation of small-scale analog maps via selection and adaptation of content from the large-scale analog maps is performed by cartographers and is called map generalization. The widespread use of digital maps has stimulated research on automatic map generalization.

Existing approaches to automatic map generalization can be categorized into three groups. The solution offered by the first group, Multiple Representation Data Bases (MRDB), consists of a set of maps at fixed scales, often together with links between objects in the different scales (National Center For Geographic Information Analysis, 1989; Kilpelainen, 1997; Hardy and Dan, 2005; Parent et al., 2006). A request for a certain map scale is answered by selecting from the pre-defined set the scale that is the closest. The second group consists of techniques that automatically generate a map at a target scale from a large-scale dataset, mostly optimization algorithms (de Berg et al., 1998; Ware and Jones, 1998; Barrault et al., 2001; Haunert, 2007). Although these algorithms produce good generalization results, the problem is that their real-time performance is not appropriate for on-line generalization. The third group consists of techniques that perform an off-line generalization, whose results are stored and can then be used for on-line map generation at any scale (Buttenfield, 2002; Brenner and Sester, 2005; van Oosterom, 2005; Bo et al., 2008).

In many countries, topographic services still produce maps at a fixed set of scales. Although the data for these maps are stored digitally, the map production and often the data collection as well, are performed separately for each scale. More and more, these maps are made available via the Internet, where they are requested to display a continuous range of scales. The available scales often reveal a considerable leap in the level of detail from one scale to the other. The same happens with other types of maps, for example soil maps, land cover maps, etc.

In this article we propose a method to perform generalization for a continuous range between two given scales. The method builds on the topological Generalized Area Partition (tGAP) (van Oosterom, 1994, 2005), and is called constrained tGAP. The constrained tGAP stores results of a generalization performed between two scales: a large-scale dataset whose geometry is stored,

and a smaller scale dataset that controls the generalization process so that the large-scale dataset is gradually transformed into the smaller scale dataset. Generalization is performed on an area partition; thus the large-scale dataset is required to be an area partition. Area objects of the smaller scale data set act as region constraints in the generalization process, i.e., they restrict the aggregation of large-scale objects only inside the region constraints. The method proposed here consists of two stages: the first stage matches objects of the large-scale dataset to objects of the smaller scale dataset, which act as region constraints in the next stage; the second stage compiles additional information needed for the constrained tGAP and performs the generalization. As a case study we use large- and medium-scale topographic data from the Netherlands.

The remaining part of this article is organized as follows. Section 2 provides an overview of automatic generalization techniques, and methods of object matching. Section 3 introduces the two datasets, large- and medium-scale Dutch topographic data. The two following sections contain the proposed method: Section 4 explains the matching between objects in the two datasets, and Section 5 describes how the constrained tGAP works. Section 6 presents the results of the constrained tGAP generalization of the Dutch topographic data. Section 7 discusses open issues and future research. Finally, Section 8 concludes the article.

2. Previous work

Most of the research in MRDB is related to multi-scale databases, that is to say databases that allow an object to be characterized by several representations of its geometry, each one at a different scale (Vangenot, 2004). Hierarchical data structures are used to organize multiple representations, where levels in the hierarchy correspond to different LoDs (Stell and Worboys, 1998; Timpf, 1998). The Multiple Representation Management System of Friis-Christensen and Jensen (2003) and Friis-Christensen et al. (2005) provides mechanisms for maintaining and restoring the consistency between different representations of the same object. The MurMur project extends to multi-representation facets other than spatial resolution by including semantic aspects (Parent et al., 2006). They propose the use of stamps to denote representations and the use of specific relationships to relate alternative representations of the same phenomenon (Vangenot et al., 2002; Vangenot, 2004; Parent et al., 2006).

An important aspect of the MRDB is the existence of links between objects in the different representations. These are used to control consistency between different representations, and also

for the propagation of updates (Kilpelainen, 1997). Sester et al. (1998) propose approaches for linking objects in different datasets of a similar scale. Linking is seen as a matching problem, and measures are proposed for matching objects in two datasets of a similar scale. The linking of objects in two datasets of different scales is treated as a derivative of generalization, that is, a representation of the smaller scale is generated from the large scale via a generalization technique, e.g., aggregation, which creates the links between the large scale objects and aggregates in the smaller scale. Friis-Christensen and Jensen (2003) discuss techniques for matching objects in different datasets, which can be grouped into techniques that perform attribute comparison, and techniques that perform spatial matching.

Map generalization aims at retaining as much as possible of the relevant information from the source map, taking into consideration cartographic rules for legibility (Jones and Ware, 2005). This is often translated to an optimization problem with a set of constraints and objectives to be satisfied to the best solution. The optimization in map generalization is often performed by applying meta-heuristics such as hill-climbing and simulated annealing (Ware and Jones, 1998). In recent years, the application of multi-agent systems has become a mainstream approach (Barrault et al., 2001). This allows different generalization operators to be integrated. Galanda (2003) discusses this approach in the context of polygon maps, using a hill-climbing strategy for optimization. Researchers in the field of computational geometry have proposed global and deterministic optimization approaches, for example, to solve the line simplification problem (de Berg et al., 1998). Often, specialized heuristics are needed to find efficient algorithms. Aggregation tasks in map generalization are formalized as an optimization problem in (Haunert and Wolff, 2006), and a deterministic approach, based on mixed-integer programming, is used to solve it. Cheng and Li (2006), as well as van Smaalen (2003), discuss criteria that need to be considered for automated aggregation.

A compromise approach between the previous two, pre-generalization (MRDB) and on-line generalization, is provided by a third group of works. The principle is that a multi-scale database stores the geometry of the highest LoD together with the changes performed by a generalization algorithm, which are then used to reconstruct a required LoD on-line. These works are tightly coupled with the progressive transfer of data in a web environment. Buttenfield (2002) presents an algorithm for the gradual refinement of polyline coordinates based on the line simplification algorithm of Douglas and Peucker (1973); it assures that topological properties are preserved. Brenner and Sester (2005) present a method to gradually refine building ground plans. A sequence

of refinement increments results from an inverted sequence of simplification steps. Operators for the simplification and reconstruction of simplified data are proposed by Yang (2005) and Yang et al. (2005). The simplification is performed by the constrained removal of vertices; a consistent topology is maintained by enforcing a set of rules. Reconstruction operators restore the original data from the simplified versions. Ai et al. (2005) propose a ‘change accumulation model’, which sees the difference between two consecutive representations as an addition or subtraction of change patches, and thus the difference between a source and a target scale as an accumulation of such changes. Taking the example of a river network, Bo et al. (2008) present a structure based on the ‘change accumulation model’, which is used for progressive transfer. Bertolotto and Egenhofer (2001) and Follin et al. (2005) generally express differences between different given vector maps by atomic generalization and refinement operators.

The method we propose in this paper falls into this last group. The original tGAP (topological Generalized Area Partition) (van Oosterom, 1994, 2005) stores the geometry of the highest LoD, the input dataset, and keeps track of the generalization process, which performs a binary merge of areas until all is merged to one area that forms the full extent. The constrained tGAP requires two datasets as input, the highest LoD dataset, and a smaller scale dataset, which is expected to be a good generalization of the highest LoD dataset. The generalization transforms the highest LoD dataset towards the smaller scale dataset. The geometry of the highest LoD and the transformation steps are stored in the constrained tGAP structure, which can then be used for progressive transfer in the web (van Oosterom et al., 2006; Haunert et al., 2009). In Haunert et al. (2009) we used the constrained tGAP for generalization between a given large-scale dataset, and a smaller scale computed via an optimization method (Haunert and Wolff, 2006). For the method proposed in this paper, the smaller scale dataset is produced by an independent process, i.e., not derivable from the highest LoD dataset, which imposes different requirements to the generalization between the scales.

3. Datasets of large and medium scale

In this study we use large- and medium-scale topographic data from the Netherlands, at 1:1 000 and 1:10 000 scales, respectively. Large-scale topographic data in the Netherlands are mostly produced by municipalities, while smaller scale maps are produced by the Geo-information department of the Dutch Cadaster. A new model, IMGeo, is created for large-scale data, which reflects the content of existing topographic data at the scale 1:1 000. Top10NL is the model for topographic

data at the scale 1:10 000. The datasets employed in this study follow the two models, IMGeo and Top10NL. We used data from the municipality of Almere, because it was one of the few municipalities that could provide IMGeo data when this research started. Later we received IMGeo test data from Rotterdam that were used to test the results of the generalization, which are shown in Appendix A. All the illustrations in the article were prepared with data from Almere.



Figure 1: Large-scale topographic data from the city of Almere, the Netherlands.

Topographic maps supply a general image of the earth's surface: roads, rivers, buildings, often the nature of the vegetation, the relief and the names of the various mapped objects (Kraak and Ormeling, 1996, pg. 44). The two models, IMGeo and Top10NL, endorse this meaning of

topography. It should be noted that in our research relief is not included as we focus on the area partitioning. The main classes of IMGeo are Road, Railway, Water, Building, LayoutElement and Terrain. Class LayoutElement has 11 subclasses, each representing a different kind of a topographic element, e.g., Bin, StreetFurniture and OtherBuilding. The class Terrain contains anything that is not buildings or infrastructure, for example, forest, grassland, fallow land, etc. Although the (area) objects of IMGeo should form an area partition, this was not the case for the test data. We processed the test data and created a partition. Overlaps occurred between terrain objects and objects of other types, also between roads and water or layout elements. The overlaps were resolved by imposing an importance order on classes: layout elements have the highest importance, followed by roads and water, while terrain has the lowest importance. This order is translated to rules, e.g., ‘if a Terrain object overlaps with another object, subtract the overlap from the Terrain object’ (see Hofman et al. (2008) for a complete set of rules).

The IMGeo test data allows categorization mainly on classes. The class Terrain can be further categorized by the LanduseType attribute values, and the class LayoutElement allows 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, for the land-use values of Terrain; bin, and other buildings as subclasses of the LayoutElement class. Figure 1 shows the IMGeo data for part of the city of Almere, the Netherlands.

The Top10NL model contains all the IMGeo classes, which have similar attributes to their corresponding classes in IMGeo. We would expect the topographic model of a smaller scale to be less detailed than the model of a larger scale. This is not always the case for Top10NL compared to IMGeo, e.g., Top10NL differentiates between different kinds of forests, while IMGeo classifies them all as forest. The Top10NL objects were categorized based on the main classes. Additionally, objects of class Terrain were further categorized according to the LanduseType attribute values. A new attribute, region-class, was created for the Top10NL data, with the following values: building; road; water; grassland, forest, and other terrain from the land-use values of Terrain. Terrain objects had overlap with objects of other classes. There were also overlaps between road and water objects. We also processed the Top10NL data using the importance order of classes, and created a partition. Figure 2 shows Top10NL data from the city of Almere, covering a larger area than Figure 1. The rectangle in the middle of the map shows the extent of the map in Figure 1.

Some of the categories have the same semantics in both models, e.g., Road and Water. Some



Figure 2: Medium-scale topographic data from the city of Almere, the Netherlands. The rectangle in the middle of the image shows the extent of the map of Figure 1.

categories have similar semantics, e.g., Building in IMGeo is a residence unit, while in Top10NL it is a building block, Grass in IMGeo is similar to Grassland in Top10NL. There are categories of IMGeo that do not have a one-to-one match to the categories of Top10NL, e.g., Other building from IMGeo can be a Building or Other terrain in Top10NL, while Other terrain of Top10NL can be Lot or Fallow land in IMGeo. Semantic matching is considered in the object matching described in the next section.

4. Object matching

The matching of objects is often based on their spatial overlap (Friis-Christensen et al., 2005). A large-scale object is assigned to a smaller scale object that contains it fully, or to a large extent.

The different levels of accuracy of the two scales and differences in semantics cause problems for the automatic matching. The problems encountered with the datasets we used were mainly with buildings and roads, as illustrated in Figures 3 and 4.



Figure 3: Large-scale building objects (in red) overlaid with medium-scale buildings blocks (in blue). The background is the large-scale dataset in dimmed colors.

Figure 3 shows IMGeo buildings in red, overlaid with Top10NL building blocks shown in blue. Top10NL building blocks are shifted from the IMGeo buildings. The shift is very irregular and cannot be explained by parallax (error in the photogrammetric data acquisition of objects with some relative height). The only explanation is that some displacement is added by the cartographer. The displacement is within 4 meters, which is the required accuracy of Top10NL.

Figure 4 shows IMGeo roads in gray, overlaid with transparent Top10NL roads in blue. It can be seen that the position of Top10NL roads matches the IMGeo roads very well. The problem is that

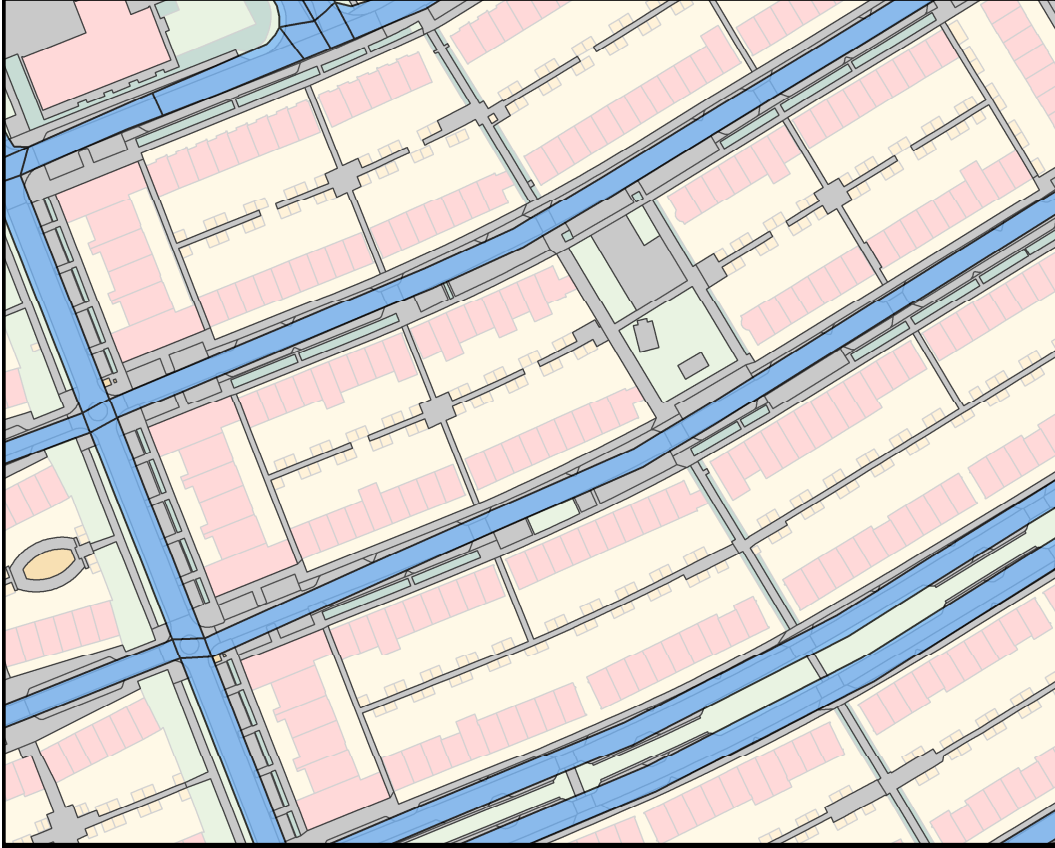


Figure 4: Large-scale road objects (in gray) overlaid with medium-scale roads (in blue). The background is the large-scale dataset in dimmed colors.

there are far more road objects in IMGeo than in Top10NL. The IMGeo data of Almere considers parking places and sidewalks as roads, while Top10NL does not. The IMGeo road objects in the Almere dataset are not subdivided into different categories (the production of the IMGeo dataset was a pilot project and the municipality of Almere did not have these data at its disposal). The IMGeo data from the municipality of Rotterdam has a more detailed classification of roads.

Top10NL objects serve as constraints in the generalization process, which requires every IMGeo object to be assigned to a Top10NL object. Several methods were investigated for assigning IMGeo objects to Top10NL objects and their results were compared. The following Sections 4.1 – 4.4 treat each method separately. A center is defined for a region constraint, which is an IMGeo object that has the same category as the constraint, i.e., the Top10NL object to which it belongs. The

center is used by the generalization algorithm to cast its category to the objects within a constraint. Section 4.5 explains the selection of a center for a region constraint.

4.1. Simple overlay method



Figure 5: The results of the simple overlay method. Objects are colored according to the medium-scale region to which they are assigned, which gives an indication of the constrained tGAP end result

The first test was a simple overlay; the resulting dataset contains the geometry of both the IMGeo and the Top10NL data. The Top10NL data have more details for some classes, like roads that have often a more detailed geometry, e.g., in traffic circles. The method creates a dataset with a richer geometry than the IMGeo data. Feeding this dataset to the constrained tGAP generalization produces a smooth morphing from IMGeo to Top10NL. However, the overlay introduces many small objects without semantics, e.g., sliver polygons resulting from the displacement of Top10NL objects with regard to the corresponding IMGeo objects. Figure 5 shows the overlay dataset. Examples

of sliver polygons are shown inside blue ovals: the double boundary line for road parts at oval numbered 1, the different shape of road sections at oval 2, and the shifted Top10NL buildings that intersect with non-building IMGeo objects at 3 and 4. Furthermore, the 1:10 000 map obtained from generalization would have the Top10NL geometry. This can be seen in Figure 5 where objects are colored on the category value coming from the Top10NL. The constrained tGAP generalization will create an object for the 1:10 000 map from each block of adjacent objects with the same color. The accuracy of the IMGeo data is 28cm–56cm, which is better than the Top10NL accuracy, thus we want to preserve the IMGeo accuracy in the generalization results. Therefore we looked for another method for matching objects.

4.2. Largest overlap method

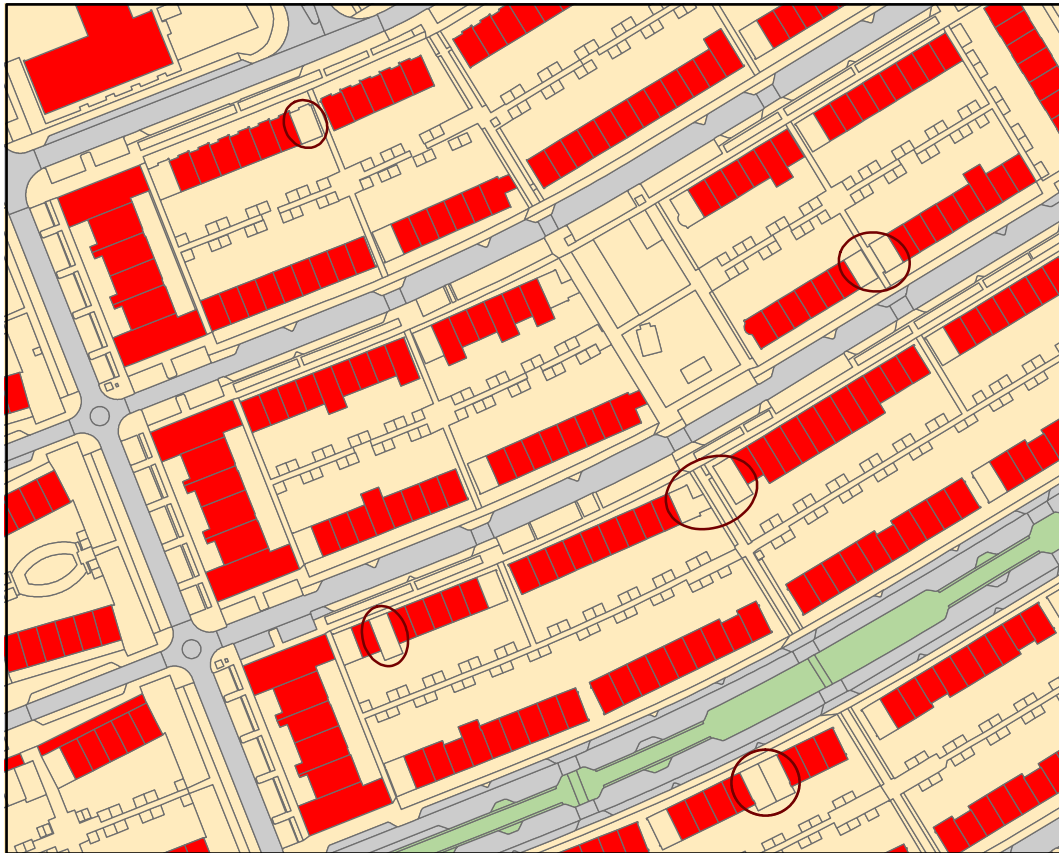


Figure 6: The results of the largest overlap method. The large-scale objects are colored according to the medium-scale region to which they belong, thus indicating the end result of the generalization.

This method assigns an IMGeo object to the Top10NL object with which it has the largest overlap. This implies that the IMGeo objects belonging to a region constraint can be partially outside the original Top10NL object. The geometry of IMGeo will be preserved in the generalization results. This method reveals problems with buildings: some IMGeo buildings have a larger overlap with other objects than (the corresponding) building block, often with Terrain objects. This produces missing buildings at the end or in the middle of a building block, as indicated by the dark red ovals in Figure 6.

4.3. Split method



Figure 7: The results of the 35% split method. Objects are colored according to the medium-scale region to which they belong, indicating the end result of the generalization.

The largest overlap method gave unsatisfactory results. On the other side, we know that the Top10NL contains more details for some classes, e.g., roads, thus Top10NL geometry can be an

enrichment for IMGeo geometry. We modified the rule: if an IMGeo object belongs for more than 35% to two Top10NL objects, then we split the IMGeo object in two new objects. All the other IMGeo objects are processed based on the largest overlap rule. This avoids assigning the complete IMGeo building to a non-building block region in the Top10NL, thus completely losing the building. Figure 7 illustrates the results of this method. Objects created by this method are colored according to the category of the Top10NL object to which they are assigned. Blocks of adjacent objects of the same color will be the objects of the 1:10 000 scale resulting from the generalization. As can be seen in Figure 7, the results were still unsatisfactory: a small overlap between IMGeo buildings and Top10NL building blocks creates very narrow building objects in the final result (see examples marked 1, 2, and 4 in Figure 7), a largely displaced Top10NL building block overlapping a courtyard by more than 35% creates a false building from part of the courtyard (see example 3 in Figure 7), and there are still missing buildings (examples 4 and 5 in Figure 7).

4.4. Building first method

Most of the unsolved problems in the previous methods were related to buildings. This could be expected from the observations about the two datasets given at the beginning of this section, and also illustrated in Figure 3. A fourth method was developed, which first checks all buildings and assures that an IMGeo building gets assigned to a Top10NL building block in case of any overlap, regardless of the amount of overlap. All the other objects were processed with the largest overlap method described earlier. Figure 8 illustrates the results of this method. Again, IMGeo objects are colored according to the category of the Top10NL object to which they are assigned, thus indicating 1:10 000 objects that will result from the constrained tGAP generalization.

To compare the results of these four methods we calculated ratios of correctly assigned objects to the total number of objects. A correctly assigned object is an IMGeo object that is assigned to a Top10NL object of the same category. The ratios were 0.47, 0.65, 0.64, and 0.68 for the overlay, largest overlap, 35% split, and building first methods, respectively. The last method assures the highest number of IMGeo objects correctly assigned to Top10NL region constrains. Therefore we chose the data produced by this last method as the input for the second stage, constrained tGAP generalization. The results shown in the rest of the article are generated employing the dataset produced by the ‘Building first method’.

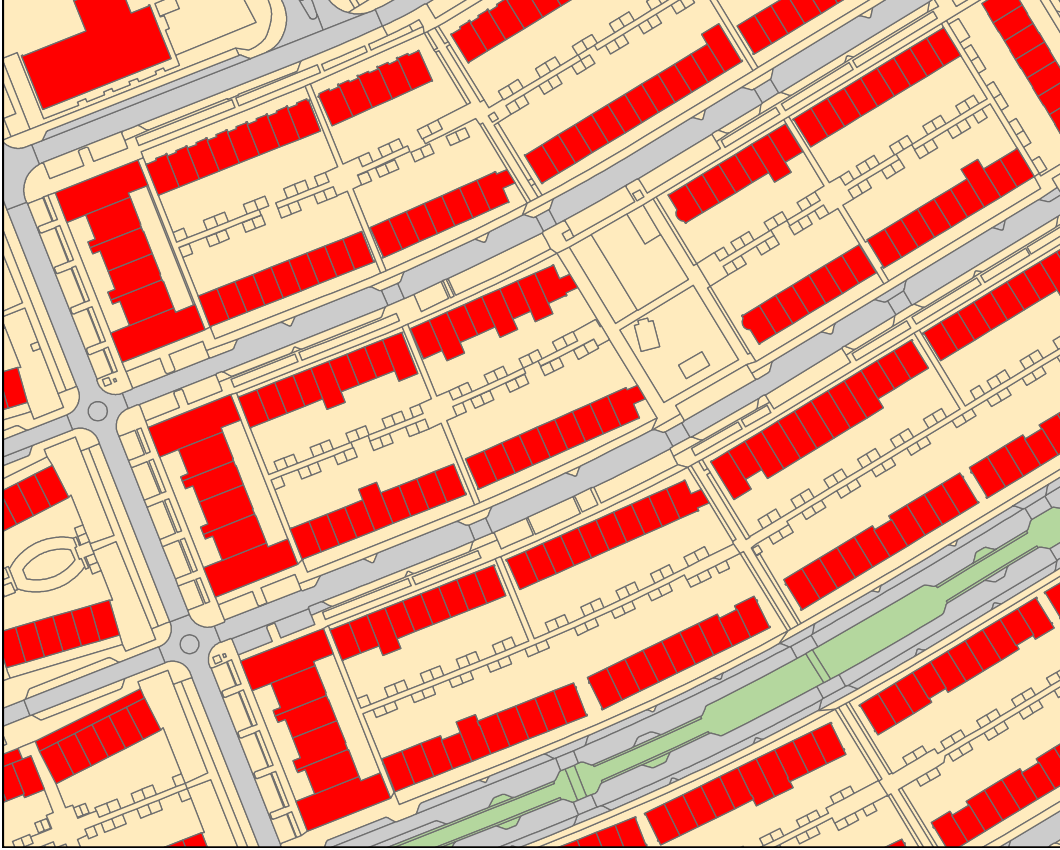


Figure 8: The results of the building first method. The large-scale objects are colored according to the medium-scale region too which they belong, indicating the final result of the generalization.

4.5. Assigning centers to region constraints

A center is an IMGeo object of the same category as the Top10NL object to which it is assigned. Each Top10NL object induces a region constraint, which is a collection of IMGeo objects that are assigned to this Top10NL object. For each region constraint, we look for an IMGeo object that has the same category as the constraint, and select the largest object to be the center of the constraint. A center is used in the generalization as a seed that will spread its category to the surrounding objects within the constraint.

There are two categories for Top10NL objects, Forest and Other terrain, which are not present in IMGeo objects. Top10NL objects belonging to these categories do not have centers. The aggregation of IMGeo objects that belong to such region constraints is driven only by weight and compatibility

values (given in Table 1).

5. Generalization between scales

We have a large-scale dataset and aggregates of large-scale objects that represent objects of a medium-scale dataset, derived from the matching described in Section 4.4. We want to perform a generalization that gradually transforms the large-scale objects to their aggregates, producing in this way a gradual transformation from the large-scale map to the medium-scale map. This generalization is performed by the constrained tGAP.

The constrained tGAP works with spatial data in 2D, and requires the data to form an area partition, i.e., there are no overlapping areas. The generalization process, which is prepared off-line, merges objects of the largest scale to form objects for smaller scales, and keeps track of this merging. The goal is minimal geometrical redundancy, and is achieved in two ways: using a topological model for storing objects of the largest scale, thus avoiding the redundant storage of shared boundaries between neighbors 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, thus avoiding redundancy between scales. Topological consistency is assured by the topological model that stores the largest scale data, and the correct topological references that are stored in the generalization process. Section 5.1 explains the constrained tGAP generalization process, and Section 5.2 describes the data stored by the constrained tGAP.

5.1. Constrained tGAP generalization

The generalization process is performed in steps. In each step, the least important object is merged to its most compatible neighbor, forming a new object. This pairwise merging is controlled by region constraints: objects are allowed to merge only if they belong to the same region constraint. Inside the constraints, the generalization results are driven by the importance and compatibility values of objects. The importance value of an object v is calculated from the area size, and the weight of the object's category: $Imp(v) = area_size(v) \cdot weight(class(v))$. The compatibility between two objects u and v is calculated from the length of the shared boundary, and the compatibility values between their categories: $Comp(u, v) = bnd_length(u, v) \cdot compatib(class(u), class(v))$. The generalization stops when all the objects are merged up to the region constraints.

Figure 9 shows the results of the (unconstrained) tGAP generalization in the first row, and the constrained tGAP generalization in the second row, for the same steps. The constraints are shown

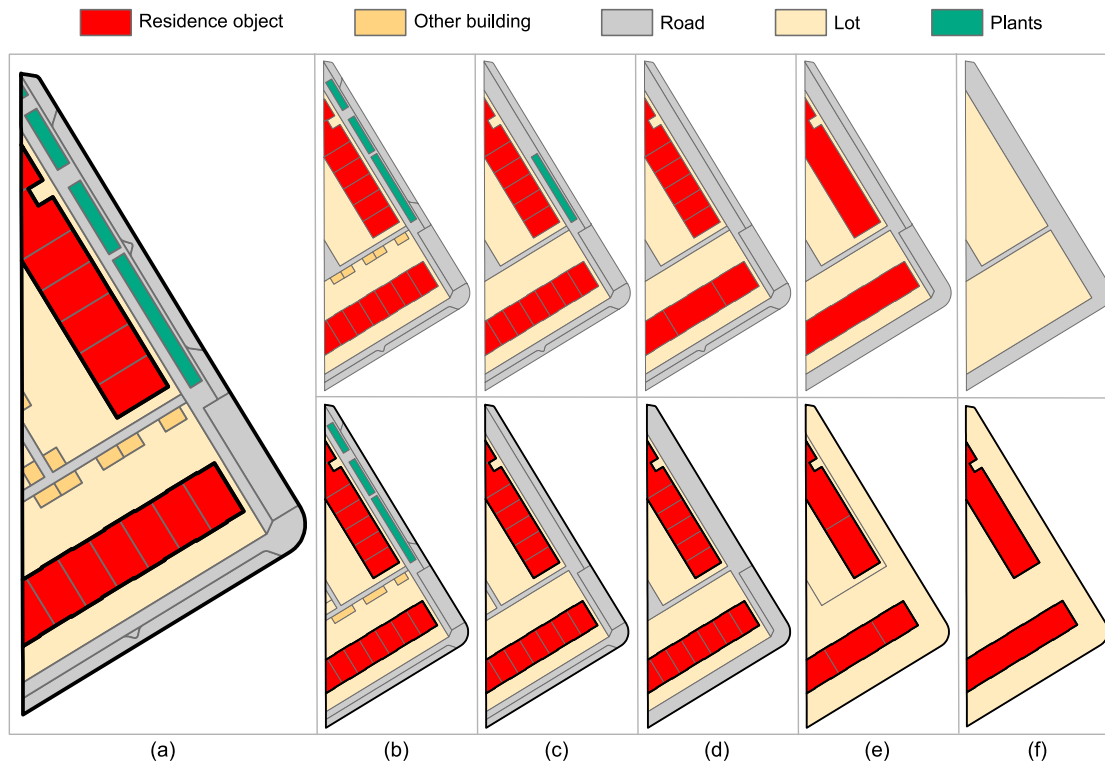


Figure 9: Results of the tGAP and the constrained tGAP generalizations for the same steps: (a) original data and constraints; (b)–(f) results of steps 10, 20, 25, 35, and 39, respectively; the first row shows the results of the tGAP generalization, the second row the results of the constrained tGAP generalization. Constraints are shown in a thick black line for the constrained tGAP; step 39 is the last one for constrained tGAP, and it is also the region constraint.

in a thick black line (second row). Figure 9(a) shows the large-scale data, a small city block taken from the IMGeo data of Almere. Figures 9(b)– 9(f) show the results of the generalization steps 10, 20, 25, 35, and 39, respectively. Step 39 is the last step of the constrained tGAP generalization. Default weight and compatibility values were used in the tGAP generalization: weights equal to 1 for all the categories, compatibility value 0.1 for any two different categories, and value 1 for the same category. The tuned values for weights and compatibilities, which are given in Table 1, were used for the constrained tGAP. In the generalization results in the first row we see that the largest areas take over their neighbors because of the default values applied for weights: buildings disappear after step 35 because lots have a larger area; sidewalks, categorized as Road in the IMGeo test data, remain longer because they cover a larger area. The application of tuned values in the second row produces different results: plants have a low importance (i.e., weight), as a result they

disappear earlier; lots are more important than roads, therefore they remain longer. It can be seen that for the constrained tGAP, merging is only performed inside the region constraints. This directs the generalization towards the medium-scale map, which is reproduced in the last step of the constrained tGAP.

The topological model used for the largest scale data consists of faces, edges, and nodes. Each edge contains references to its left and right faces, as well as to its start and end nodes. Geometry is stored for edges and nodes, whereas the geometry of a face is to be constructed by a topology builder algorithm that collects the edges referring to it as its left or its right face. The generalization starts at step 0 with all the objects (i.e., faces) of the largest scale being valid. A generalization 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 begins at this step. If one of the merged faces was a center, then the new face becomes a center. The starting and the ending validity values are stored for every face during the generalization process. For each step, we keep track of the changes occurring at the boundary edges of the two merged faces. An edge disappears if it is part of the common boundary of the two merged faces. The other edges from the boundary of the two merged faces still exist, but the reference to the left or the right face changes: a new version is created for such edges, with the same geometry, but an updated left or right face reference. A validity range is recorded for each version of an edge.

The constrained tGAP generalization 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 topological model in database tables. Face and edge information, as well as information on the class weights and compatibilities, are read from the database tables into arrays in memory. The face array is sorted by importance values, and is always maintained in this order after the removal of merged faces and the 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. The edge array is correspondingly updated by removing common edges, and updating the references for the other boundary edges. Changes happening at each generalization step are reflected in the database tables (explained in the next section).

```

1: fill face array with info: face-id, class, importance, region-id, center, area /* SQL statements collecting face */
2: fill edge array with info: edge-id, left-face-id, right-face-id /* and edge information, as well as */
3: get class weights and compatibilities into memory arrays /* weight and compatibility info */
4: while face array is not empty do
5:     get neighbors of face[1] that are in the same region constraint into neighbors array
6:     if neighbors array is not empty then
7:         get the most compatible neighbor from neighbors into best-nbhd
8:         compile information for the new face

```

```

9:         remove the two merged faces from face array
10:        insert the new face in the right order in the face array
11:        insert the new face into the Face table
12:        update edge array          /* consequence of merging the two faces */
13:        update the Edge table
14:    else                                /* face[1] is a region constraint */
15:        if face[1] is not a center then
16:            update the record in Face table: set class to region-class
17:        end if
18:        remove face[1] from the array
19:    end if
20: end while

```

Lines 1–3 read face and edge information for the large-scale data, and weight and compatibility values from the database tables to memory arrays. Lines 5–7 select the most compatible neighbor for the current least important face. Line 8 compiles information for the new face coming from the least important face and its most compatible neighbor. If one of the two faces is a center, then the new face becomes a center and it acquires the class of the center. Otherwise, the class of the most compatible neighbor becomes the class of the new face. Lines 9–13 update the face and edge tables and memory arrays. Lines 15–17 ensure that a face that is enlarged to a region constraint acquires the class of the region, and line 18 removes it from the face array.

5.2. Constrained tGAP data

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

Information about area objects is stored in the $\langle \text{dataset} \rangle$.Face table: an identifier, the category to which it belongs, the region constraint, an attribute with value ‘T’ or ‘F’ that defines whether the area is a center, the area size, and the validity range as [imp_low, imp_high). Information about edges is split into two tables: $\langle \text{dataset} \rangle$.EdgeGeo and $\langle \text{dataset} \rangle$.Edge. The first table contains an identifier for an edge, references to its start and end nodes, the geometry, and its length. Table $\langle \text{dataset} \rangle$.Edge stores the left and right faces of an edge as they change during the generalization, 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 the tables is that part of the edge information, $\langle \text{dataset} \rangle$.Edge, changes frequently as result of the creation of the tGAP structure, while the other part, $\langle \text{dataset} \rangle$.EdgeGeo, is static. Node information is stored in the $\langle \text{dataset} \rangle$.Node table. The table

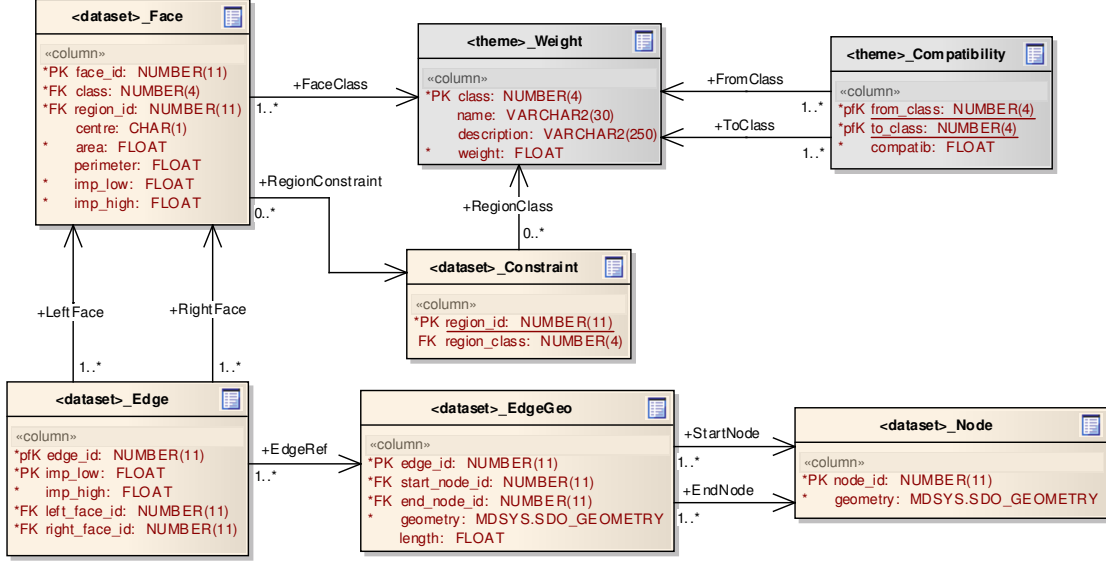


Figure 10: UML diagram of the tables and relationships that store the constrained tGAP information in Oracle Spatial. PK denotes a primary key, FK a foreign key, and pFK a foreign key that is part of a primary key.

(theme)_Weight stores information about categories: their codes as referred in the Face table, name and description, and the weight. The table (theme)_Compatibility stores the compatibility value of changing from the from_class to the to_class. Note that the tables do not store the medium-scale geometry. This is derived from the large-scale geometry via references.

6. Results of generalization

The inputs to our generalization method are the large-scale data, IMGeo, which have a high accuracy, and the medium-scale data, Top10NL, which are produced following expert (cartographer) rules. We consider the Top10NL data to be a good representation for the medium-scale map. Knowledge from the medium-scale map is transferred through the matching process to the region constraints, which control the generalization process of the constrained tGAP. The end result of our generalization is a map of scale 1:10 000 that is built upon the original medium-scale map. Besides the 1:10 000 map, our generalization method provides maps in the range of scales of the two input datasets. Maps for all the scales in the range 1:1 000 to 1:10 000 are compiled from the large-scale data features and the references stored during the generalization.

We applied the constrained tGAP generalization using a set of default values for weights and

compatibilities: weights equal to 1 for all the categories, which means that the importance of objects is only based on the area size; the compatibility value equals 0.1 for objects of different categories, and the compatibility value is 1 for objects of the same category. The results were not good, so we tuned the weight and compatibility values. We referred to weight and compatibility values from van Putten and van Oosterom (1998), and tuned them further in order to produce better generalization results.

From-class code →		<i>1001</i>	<i>2001</i>	<i>3001</i>	<i>4001</i>	<i>4002</i>	<i>4003</i>	<i>4004</i>	<i>4005</i>	<i>5001</i>	<i>5002</i>
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↓										
<i>Building</i>	<i>1001</i>	1.00	0.50	0.00	0.90	0.90	0.50	0.50	0.00	0.00	0.99
<i>Road</i>	<i>2001</i>	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00
<i>Water</i>	<i>3001</i>	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<i>Lot</i>	<i>4001</i>	0.50	0.90	0.00	1.00	0.95	0.95	0.95	0.80	0.95	0.95
<i>Fallow land</i>	<i>4002</i>	0.90	0.90	0.00	0.50	1.00	0.90	0.90	0.50	0.80	0.95
<i>Plants</i>	<i>4003</i>	0.50	0.50	0.00	0.50	0.50	0.99	0.95	0.90	0.90	0.50
<i>Terrain unk.</i>	<i>4004</i>	0.90	0.50	0.00	0.50	0.90	0.00	1.00	0.99	0.95	0.95
<i>Grass</i>	<i>4005</i>	0.50	0.90	0.00	0.80	0.50	0.95	0.95	0.99	0.95	0.50
<i>Bin</i>	<i>5001</i>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
<i>Other Build.</i>	<i>5002</i>	0.99	0.50	0.00	0.50	0.90	0.50	0.50	0.00	0.00	1.00

Table 1: The weight and compatibility values determined for categories of the large-scale data.

The tuning of values was trial and error, based on visual inspection of the results, which aim is a gradual transformation of the large-scale dataset toward the medium-scale dataset. The optimum values found for the weights and the compatibilities are given in Table 1. The second row contains the weights, and the rest of the table contains compatibility values.

Buildings are important and should be retained through all the scales of the maps. We ensure this by granting a high weight value to the category *Building*. The IMGeo dataset of Almere considers parking places and sidewalks as roads, while the Top10NL dataset does not. This results in more road objects in the IMGeo data as compared to the Top10NL data. To achieve a gradual decrease of roads during the generalization, we attach a relatively small weight to roads, as well as a very low compatibility of other categories to roads (see row *2001*, most of the values are 0) while allowing relatively high compatibilities of roads to other categories (see column *2001*). We consider water to be compatible only with itself, and incompatible with other categories: compatibility of water to any other category equals 0, and also compatibility of any other category to water equals 0. The IMGeo categories, lot, fallow land, terrain (unknown), bin and other building have a similar

semantics with the category terrain of Top10NL, thus we want the objects of these categories to aggregate together. We make *Lot* an attractive category by granting it a high weight value and high compatibilities of the other categories to lots, while allowing it to merge to the other categories, which is ensured by the medium-high compatibility values of lots to the other categories.



(a) large-scale data (scale 1:1 000)



(b) generalization result at scale 1:5 000

Figure 11: Results of the constrained tGAP generalization for Almere data: (a) original large-scale data, (b) result of the constrained tGAP generalization for the scale 1:5 000.

The iterative process of the constrained tGAP allows us to build a relation between an iteration

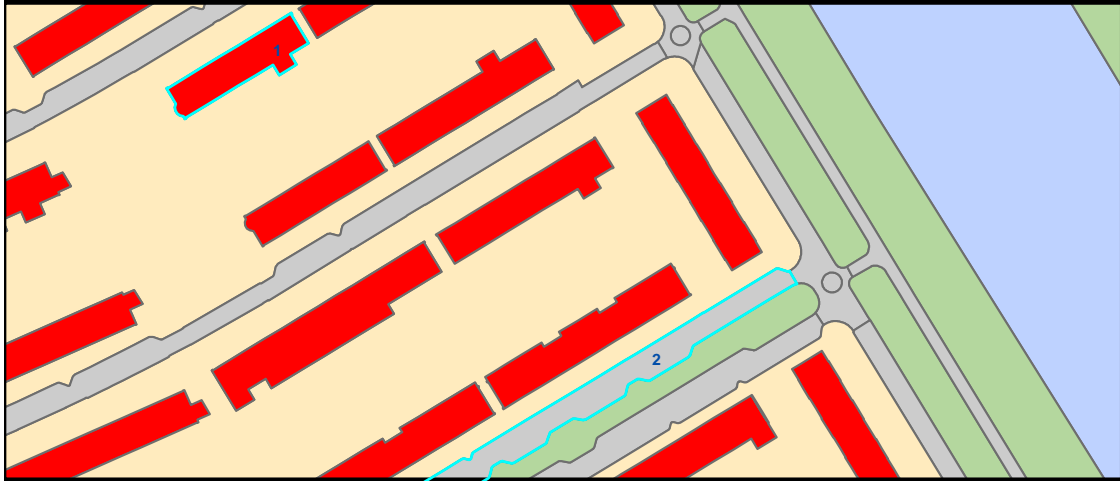
step i and the number of its valid faces (i.e., area objects): $no - objects(i) = no - IMGeo - object - i$. The relation between the scale and the number of map objects often follows certain rules, e.g., Equation 3 of (Töpfer and Pillewizer, 1966, pg. 12) states that the ratio between the number of map objects is equal to the ratio between the scales. We use these two relations to build a linear mapping between the iteration steps in the constrained tGAP and the map scales (see Haurert et al. (2009) for the mapping formula).

The maps of Figures 11 and 12 illustrate results of the generalization using weight and compatibility values of Table 1. They show the same part of Almere, for comparison: the original IMGeo data (scale 1:1 000) in 11(a), the result of the constrained tGAP generalization for the scale 1:5 000 in 11(b), the result of the constrained tGAP generalization for the scale 1:10 000 in 12(a), and the original Top10NL data (scale 1:10 000) in 12(b). Comparing Figure 12(a), the result of this research, with Figure 12(b), the Top10NL dataset, it can be seen that line simplification is still needed to further complete the results (compare the boundary lines of the selected objects: the building labeled 1 and the road labeled 2). However, it can also be concluded that the results in Figure 12(a) are a good generalization of the original data in Figure 11(a). Examples of the constrained tGAP generalization applied to Rotterdam data are given in Appendix A.

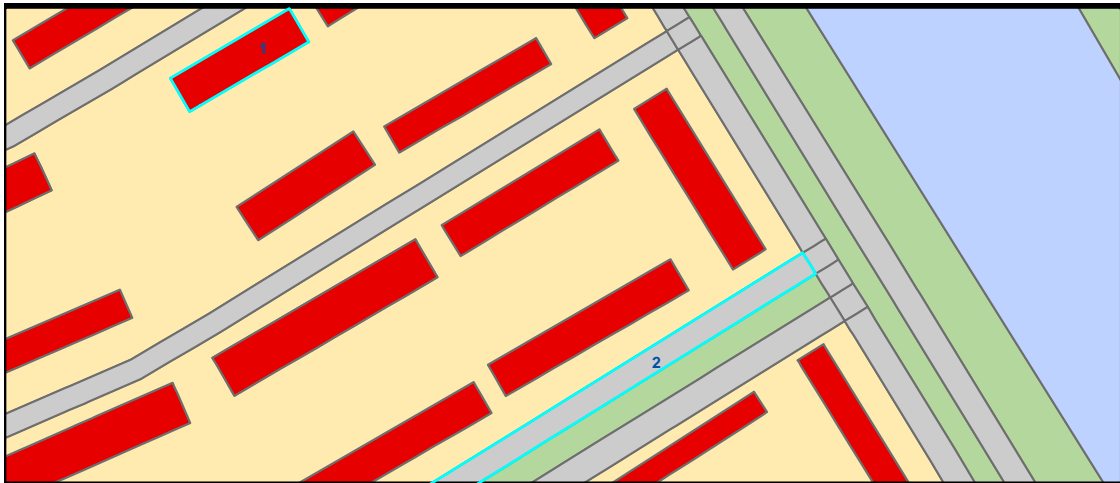
7. Future research

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

- New (non-aggregated) categories in smaller scales
- Use of statistics for fine-tuning the weight and compatibility values
- Inclusion of line simplification
- Dynamic version of constrained tGAP structure
- Explicit treatment of (temporal) inconsistencies in input data
- Constrained tGAP with more than two input scales



(a) generalization result at scale 1:10 000



(b) original data for scale 1:10 000

Figure 12: Results of the constrained tGAP generalization for Almere data: (a) result of the constrained tGAP generalization for the scale 1:10 000, (b) medium-scale data (1:10 000) for the same part of Almere.

Adding new, non-aggregated, categories in smaller scales may be required for certain object categories, which does not exist at larger scales (also not in parts or fragments). When studying the content of the Dutch topographic maps at different scales, in a few exceptional cases a complete new category occurs. An example of such a category is an air-space region. These regions are quite large and it does not make sense for them to be displayed in a large-scale map fragment as most

likely one of three options occurs: the map fragment is completely outside the air-space region, the map fragment is completely inside the air-space region, or a part of the boundary of the air-space region can be seen. This does not make much sense, therefore air-space regions are only displayed at smaller scales.

The weight and compatibility values presented in this contribution are the results of fine-tuning after several tests of performing the 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 in order to enhance the fine-tuning. Using overlay statistics may also be an approach to arrive at an initial setting for the weight and compatibility values when using the constrained tGAP method with other data sets (than the ‘well-known’ Top10NL and IMGeo).

From the generalization results shown in the 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). The simplification of buildings may require other techniques for better results. We are currently implementing the Visvalingams algorithm for line simplification (Visvalingam and Whyatt, 1993). The investigation of better results for line simplification, including topological correctness (i.e., lines in the complete dataset form a planar graph), is a topic for 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.

Inconsistencies between input data from different scales, due to reasons other than accuracy issues (e.g. temporal displacements), do negatively influence the quality of the constrained tGAP structure. These inconsistencies might be partially solved by using spatio-temporal topographic data sets and fixing 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.

The approach described in this contribution to integrate the 1:1 000 and 1:10 000 topographic data into a unified data 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 among several fixed scales. The method for matching large-scale object to medium-scale object can be applied between every pair of scales, probably after

some modifications: the ‘building first’ method is specific to the large-scale data, while the ‘largest overlap’ method is applicable to every scale. The order of creation of the overall constrained tGAP structure, bottom-up (starting with the two largest scales) or top-down (starting with the two smallest scales), does influence the overall result. The reason for this is the following: the original 1:10 000 geometry (see Figure 12(b)) is different from the derived constrained tGAP 1:10 000 geometry (see Figure 12(a), which is based on the 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 one, because decisions are based on the geometry that is actually used to represent the larger scale object in the constrained tGAP structure. However, one could also argue that the top-down approach is a good one, as the original geometries may carry the best information to make decisions on which set of objects should be assigned to the (next higher level) region. More testing is needed before a definite conclusion can be reached on the best approach.

It is important to recognize that, whatever the method of linking large-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 can be deleted. Only the largest scale and the resulting constrained tGAP structure (which encapsulates the human cartographer’s 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, and it becomes even more urgent to create the BLG-tree as well, either based on Douglas-Peucker or other techniques (see van Oosterom (2005)).

8. Conclusions

Despite the fact that this paper uses topographic data from the Netherlands, this does not mean that the application of the developed techniques is limited to either data from the Netherlands or to topographic data. Of course, the tGAP parameters such as class importance and class compatibility have to be defined for other types of data. An example of the application of constrained tGAP techniques to German land cover data is described in (Haunert et al., 2009) (but the small scale data in this paper does not have an independent source). Perhaps even more important than serving the needs of National Mapping Agencies and other geo-information producers, the vario-

scale data structures, such as the constrained tGAP, will also better serve the users via efficient vario-scale geo-information delivered in a geo-information web service context based on progressive transfer (van Oosterom et al., 2006).

In this article, we presented an approach to the integration of datasets of two different scales and generalization between the scales. We used large- and medium-scale topographic data from the Netherlands. The first stage of the process is object matching between the two datasets. Four methods were proposed for object matching, which are all based on the spatial overlap between objects, and which differ from each other in their consideration of the amount of overlap or the treatment of specific categories. We chose the method that produced the highest number of correctly matched objects between the two datasets. The output of the matching process is used in the second stage, the generalization that is performed by the constrained tGAP. Objects of the medium-scale dataset are used as constraints in the generalization process. They restrict the aggregation of the large-scale objects only inside these constraints, resulting in a gradual transformation of the large-scale dataset into the medium-scale dataset.

The constrained tGAP generalization stores the geometry of objects of the large-scale dataset, and references to features of this dataset that enable the construction of objects for the other scales. This ensures minimal redundancy of geometries and consistency between scales. The use of a topological model for storing the large-scale data ensures minimal redundancy and consistency at this horizontal level. The constrained tGAP data can be queried to produce maps for any scale in the range of scales of the two input dataset.

Large- and medium-scale Dutch topographic data were used in this research. The approach is equally valid for other datasets and other scales. The constrained tGAP generalization can be used for any categorical map, e.g., soil or land cover maps, and any pair of scales.

It is our hope that in future only the largest scale in a dynamic constrained tGAP structure needs to be updated, and that the cartographer would check 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 geo-information (at different scales), this also guarantees consistency between scales and the use of intermediate scales as well; e.g., for smooth zooming or for progressive transfer (van Oosterom et al., 2006; Haunert et al., 2009).

References

- Ai, T., Li, Z., Liu, Y., 2005. Progressive transmission of vector data based on changes accumulation model. In: Fisher, P. (Ed.), *Developments in Spatial Data Handling: 11th International Symposium on Spatial Data Handling (SDH'05)*. Springer, Berlin, Germany, pp. 85–96.
- Barrault, M., Regnauld, N., Duchene, C., Haire, K., Baeijs, C., Demazeau, Y., Hardy, P., Mackaness, W., Ruas, A., Weibel, R., 2001. Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalization. In: *Proceedings 20th International Cartographic Conference (ICC'01)*, Beijing, China. Vol. 3. pp. 2110–2116.
- Bertolotto, M., Egenhofer, M. J., 2001. Progressive transmission of vector map data over the world wide web. *GeoInformatica* 5 (4), 345–373.
- Bo, A., Ai, T., Xinming, T., 2008. Progressive transmission of vector map on the web. In: *Proceedings XXIst ISPRS Congress*, Beijing, China. No. XXXVII(Part B2) in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. pp. 411–418.
- Brenner, C., Sester, M., July 2005. Continuous generalization for small mobile displays. In: Agouris, P., Croituru, A. (Eds.), *Next Generation Geospatial Information*. Vol. 1. Routledge, UK, pp. 33–42.
- Buttenfield, B. P., 2002. Transmitting vector geospatial data across the internet. In: Egenhofer, M. J., Mark, D. M. (Eds.), *Proceedings Second International Conference on Geographic Information Science (GIScience'02)*, Boulder, CO, USA. Vol. 2478 of *Lecture Notes In Computer Science*. Springer, Berlin, Germany, pp. 51–64.
- Cheng, T., Li, Z., 2006. Toward quantitative measures for the semantic quality of polygon generalization. *Cartographica* 41 (2), 135–147.
- de Berg, M., van Kreveld, M., Schirra, S., 1998. Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and Geographic Information Systems* 25 (4), 243–257.
- 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.

- Follin, J.-M., Bouju, A., Bertrand, F., Boursier, P., 2005. Multi-resolution extension for transmission of geodata in a mobile context. *Computers & Geosciences* 31 (2), 179–188.
- 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.
- Friis-Christensen, A., Jensen, C. S., Nytnun, J. P., Skogan, D., 2005. A conceptual schema language for the management of multiple representations of geographic entities. *Transactions in GIS* 9 (3), 345–380.
- Galanda, M., 2003. Automated polygon generalization in a multi agent system. Ph.D. thesis, University of Zurich, Switzerland.
- Hardy, P., Dan, L., July 2005. Multiple representation with overrides, and their relationship to dlm/dcm generalisation. In: *8th ICA Workshop on Generalisation and Multiple Representation*. ICA Commission on Generalisation and Multiple Representation, A Coruna, Spain.
URL http://aci.ign.fr/Acoruna/Papers/Hardy_Lee.pdf
- Hauert, J.-H., 2007. A formal model and mixed-integer program for area aggregation in map generalization. In: *Proceedings Photogrammetric Image Analysis (PIA'07)*. No. XXXVI(PART 3/W49A) in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. ISPRS, pp. 161–166.
- Hauert, J.-H., Dilo, A., van Oosterom, P., 2009. Constrained set-up of the tGAP structure for progressive vector data transfer. *Computers & Geosciences*, in press.
- Hauert, 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., Dilo, A., van Oosterom, P., Borkens, N., 2008. Using the constrained tGAP for generalisation of IMGeo to Top10NL model. In: *11th ICA Workshop on Generalisation and Multiple Representations*, Montpellier, France.
URL http://aci.ign.fr/montpellier2008/papers/27_Hofman_et_al.pdf

- Jones, C. B., Ware, J. M., 2005. Map generalization in the web age. *International Journal of Geographical Information Science* 19 (8-9), 859-870.
- Kilpelainen, T., 1997. Multiple representation and generalisation of geo-databases for topographic maps. Ph.D. thesis, Finnish Geodetic Institute, Finland.
- Kraak, M.-J., Ormeling, F., 1996. *Cartography: Visualization of Spatial Data*. Longman.
- 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.
- Parent, C., Spaccapietra, S., Zimányi, E., 2006. The MurMur project: modeling and querying multi-representation spatio-temporal databases. *Information Systems* 31 (8), 733-769.
- Sester, M., Anders, K.-H., Walter, V., 1998. Linking objects of different spatial data sets by integration and aggregation. *Geoinformatica* 2 (4), 335-358.
- Stell, J., Worboys, M., 1998. Stratified map spaces: A formal basis for multi-resolution spatial databases. In: *Proceedings 8th International Symposium on Spatial Data Handling (SDH'98)*. pp. 180-189.
- Timpf, S., 1998. Hierarchical structures in map series. Ph.D. thesis, Technical University Vienna, Austria.
- Töpfer, F., Pillewizer, W., 1966. The principles of selection. *The Cartographic Journal* 3 (1), 10-16.
- 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 Geographical Information Science* 32 (4), 331-346.

- van Oosterom, P., de Vries, M., Meijers, M., 2006. Vario-scale data server in a web service context. In: Proceedings 9th ICA Workshop on Map Generalisation and Multiple Representation, Portland, OR, USA.
- 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.
- van Smaalen, J. W. N., 2003. Automated aggregation of geographic objects. Ph.D. thesis, Wageningen University, The Netherlands.
- Vangenot, C., August 2004. Multi-representation in spatial databases using the mads conceptual model. In: 7th ICA Workshop on Generalisation and Multiple Representation. ICA Commission on Generalisation and Multiple Representation, Leicester, UK.
- Vangenot, C., Parent, C., Spaccapietra, S., 2002. Modelling and manipulating multiple representations of spatial data. In: Symposium of Geospatial Theory, Processing and Applications. Ottawa, Canada.
- Visvalingam, M., Whyatt, J. D., 1993. Line generalization by repeated elimination of points. *Cartographic Journal* 30 (1), 46–51.
- Ware, J. M., Jones, C. B., 1998. Conflict reduction in map generalization using iterative improvement. *GeoInformatica* 2 (4), 383–407.
- Yang, B., 2005. A multi-resolution model of vector map data for rapid transmission over the internet. *Computers & Geosciences* 31 (5), 569–578.
- Yang, B., Purves, R., Weibel, R., 2005. Efficient transmission of vector data over the internet. *International Journal of Geographical Information Science* 21 (2), 215–237.

A. Results of the constrained tGAP for Rotterdam data

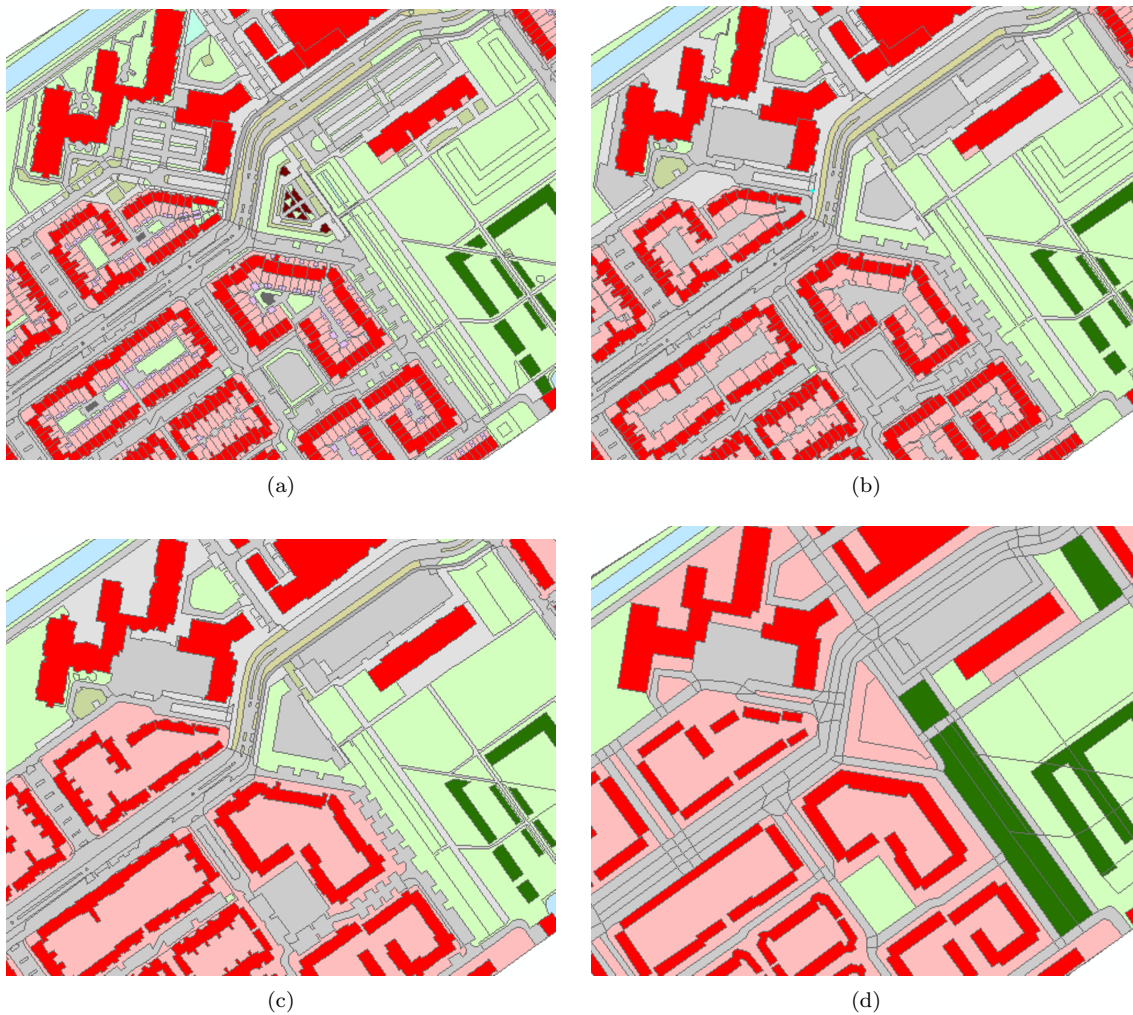


Figure 13: Results of the constrained tGAP generalization for Rotterdam data: (a) original IMGeo data, (b) results of generalization for the scale 1:3 000, (c) results of generalization for the scale 1:10 000, (d) the corresponding Top10NL data.