Deriving Scale-Transition Matrices from Map Samples for Simulated Annealing Based Aggregation

Birgit Kieler[1], Jan-Henrik Haunert[2] and Monika Sester[1]

[1]*Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Hanover, Germany*
[2]*Lehrstuhl für Informatik I, Universität Würzburg, Germany*

The goal of the research described in this paper is to derive parameters necessary for the automatic generalization of land-cover maps. A digital land-cover map is often given as a partition of the plane into areas of different classes. Generalizing such maps is usually done by aggregating small areas into larger regions. This can be modeled using cost functions in an optimization process, where a major objective is to minimize the class changes. Thus, an important input parameter for the aggregation is the information about possible aggregation partners of individual object classes. This can be coded in terms of a transition matrix listing costs that are charged for changing a unit area from one class into another one.

In our case we consider the problem of determining the transition matrix based on two datasets of different scales. We propose three options to solve the problem: 1) the conventional way where an expert defines manually the transition matrix, 2) to derive the transition matrix from an analysis of an overlay of both datasets, and 3) an automatic way where the optimization is iterated while adapting the transition matrix until the difference of the intersection areas between both datasets before and after the generalization is minimized. As underlying aggregation procedure we use an approach based on global combinatorial optimization.

We tested our approach for two German topographic datasets of different origin, which are given in the same areal extent and were acquired at scales 1:1,000 and 1:25,000, respectively. The evaluation of our results allows us to conclude that our method is promising for the derivation of transition matrices from map samples. In the discussion we describe the advantages and disadvantages and show options for future work.

Keywords: Multi-scale representation; generalization; aggregation; optimization; knowledge acquisition

1. Introduction and related work

A fundamental research topic in cartography is the derivation of maps in different scales using generalization processes. The vision in generalization is to provide a base scale in highest resolution and automatically derive the smaller scale maps from it. Even if automatic generalization has shown some considerable success in recent years, this vision has not yet been reached (Mackaness et al. 2007). Methods for the automation of individual generalization operations have been proposed, e.g. line simplification (Douglas & Peucker 1973), building simplification (Staufenbiel 1973), road network typification (Richardson & Thomson 1999), and displacement (Harrie 1999) – however, what is still lacking are approaches for the complex interplay of different operations when applied to different spatial situations.

The main difficulties rise from the fact that the preconditions for applying an operation depend on the object and its context. Defining context and constraints for given generalization tasks has been investigated in a recent project of the EuroSDR (Foerster & Stoter 2008). In order to satisfy multiple, possibly competing constraints, the use of optimization approaches is promising (Harrie & Weibel 2007). Among them, the principle of software agents has been proposed (Lamy et al. 1999), as well as mathematical optimization (Sester 2005) and heuristics (Ware & Jones 1998).

The classical method for area aggregation was presented by van Oosterom (1995) – the so-called GAP-tree. In a region-growing approach, areas that are too small to be represented and visualized in the target scale are merged with neighbouring areas. The selection of the neighbours depends on different criteria, mainly geometric and semantic constraints, e.g. length of common boundary. Recently, Haunert (2008) introduced a global optimization approach for the problem of area aggregation in order to overcome fundamental restrictions that a mere local aggregation has. He used mathematical programming in order to be able to define both soft and hard constraints for the underlying problem. Instead of determining the optimal solution with mixed-integer programming also heuristics like simulated annealing can be applied, which have the advantage of computational efficiency.

A basic problem in aggregation is to define the compatibility or similarity of object classes. Different authors have proposed methods to derive such measures from given data models ((Rodriguez & Egenhofer 2004), (Yaolin et al. 2002)). Schwering (2008) reviews different approaches for defining semantic similarity measures. For the definition of semantic similarity between object classes ontology-based (Kavouras et al. 2005) and instance-based approaches are proposed ((Duckham & Worboys 2005), (Kieler et al. 2007)).

In our case the differences between the datasets result both from different underlying ontologies *and* from different (but related) scales. Thus, especially part-of relationships are relevant for the different conceptualizations of the data in the datasets.

Thus, in the context of aggregation the issue of similarity has to be discussed thoroughly: on the one hand, an object should be merged with its most similar neighbour in order to assure that the general spatial situation does not change too drastically. However, in the course of aggregation some objects may completely disappear, without being able to identify a semantically similar neighbour. Consider e.g. a building in a forest, which will be merged with the forest area in smaller scales – although their object classes are semantically very different. Thus, merging also reflects the preservation and enhancement of dominant and important features in a local environment. Therefore, in the remainder of the paper we will no longer use the notion of similarity matrices, but scale-transition matrices instead.

The hypothesis of our paper is based on these observations: Using existing datasets of different scale but similar thematic contents and a given optimal aggregation scheme can reveal the scale-transition relationships between different datasets. The resulting relationships can be used for the generalization of other high resolution datasets later on.

The paper is structured as follows. First the aggregation scheme based on global optimization is presented (Section 2). Then we describe the datasets that are used in this study, and the necessary preprocessing steps (Section 3). Then we present our strategies for the determination of the transition matrices in Section 4. The presentation and evaluation of the results obtained in our experimental tests is given in Section 5. Section 6 summarizes the achievement and gives an outlook on future work.

2. Aggregation procedure based on global optimization

We now sketch our method for area aggregation; for a more detailed presentation we refer to Haunert (2008).

Aggregation means to replace a set of input objects by a smaller set of output objects such that there exists a many-to-one mapping between the elements of both sets. In our case both the input objects and the output objects are areas, each belonging to a single land-cover class. Both the input dataset and the output dataset are planar partitions, that is, there are no gaps and no overlapping areas. The output areas must satisfy hard size constraints defined for the target scale, more precisely, they must not be smaller than a certain size.

In order to ensure that all output areas are large enough, we usually need to accept class changes, for example, a small forest area may need to be aggregated with its surrounding farmland areas, resulting in a large farmland area. Obviously, the amount of such class changes should be kept small, that is, large areas should keep their classes.

There are a few area aggregation methods that ensure size constraints, for example, the method of van Oosterom (1995) iteratively merges pairs of areas until all areas are large enough. In each step of this process the *least important area* is merged with its *most compatible neighbour*; both the importance and the compatibility can be defined based on different criteria, for example, the sizes and the classes of the areas.

Though the method of van Oosterom (1995) yields areas of sufficient size we observed a severe drawback: even if we apply class-based compatibility measures, the classes change a lot. Always selecting the most compatible neighbour is greedy and may yield bad output maps. Due to this reason we developed a new aggregation method by combinatorial optimization that minimizes class changes while ensuring output areas of sufficient size (Haunert 2008). Additionally, our method allows the compactness of areas to be maximized, which is necessary to obtain geometrically simple shapes. We tested different combinatorial optimization methods including an exact approach (mixed-integer programming) and heuristics (for example, simulated annealing), see Haunert (2007) for a comparison. In contrast to our exact approach, our heuristic approaches offered near-optimal solutions in modest time. Therefore, we applied the simulated annealing method in the experiments presented in this paper.

We now discuss the objective function that we applied in our optimization approach. In order to quantify class changes, we introduce a class distance $\mathrm{d}\colon \Gamma^2 \to \mathbb{R}_0^+$ with $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_k\}$ being the set of all object classes. With this distance we express the global cost for class changes as

$$f_{class} = \underset{v \in V}{\qquad} w\ \ v\ \ \cdot d\ \ \gamma\ \ v\ \ ,\gamma'\ \ v\ \ ,$$

with V being the set of all areas in the input dataset, $w\colon V \to \mathbb{R}^+$ their size, $\gamma\colon V \to \Gamma$ their class before aggregation, and $\gamma'\colon V \to \Gamma$ their class after aggregation. The class distance d reflects the scale-transition relationship between object classes, which in some cases also reflects the semantic similarity, i.e. changing an area from farmland to a similar class such as grassland is cheap but changing the same area to settlement is expensive. Defining appropriate class distances, however, is a difficult process. Existing approaches (e.g. Yaolin et al. 2002) rely on class hierarchies and attributes, which may not be defined in the given data model. Therefore, we approach the problem of deriving class distances by analyzing given map samples, see Section 4.

In addition to class distances we consider the compactness of shapes. We define the cost for the non-compactness of an output region by

$$c = perimeter^2.$$

We denote the total cost for non-compact regions, that is, the sum of c over all output regions, as $\mathrm{f}_{\mathrm{noncomp}}$. Note that, with this cost function, there is no bias towards larger or smaller output regions: For a single square of unit size we would charge the cost $f_{noncomp} = c = 4^2 = 16$. If we cover the same space by four squares of size ¼, each square would contribute a cost of $c = 2^2 = 4$, thus again the total cost would be $f_{noncomp} = 4c = 16$. The size of the shapes in the solution does not affect $f_{noncomp}$.

To express a trade-off between both objectives we define the cost

$$f = s \cdot f_{class} + \ 1 - s \ \cdot f_{noncomp} \ ,$$

with $s \in \ 0,1\ $ being a weight factor.

In contrast to our exact optimization approach by mixed-integer programming, our simulated annealing does not guarantee results that strictly satisfy the size constraints. It is possible, however, to penalize regions that are too small and thereby to generate solutions in which most output areas have the required size. We do this by adding a term in the cost function. Let θ be the required size in the target scale. For a region of size W with $W < \theta$ a penalty equal to $\theta - W$ is charged. We denote the sum of all such penalties as f_{size}. The cost function becomes

$$f' = r \cdot f_{size} + \ 1 - r \ \cdot f \ ,$$

with $r \in \ 0,1\ $ being a weight factor.

The simulated-annealing method requires additional parameters which are usually determined through experiments: the initial temperature T_S, the final temperature T_E, and the number of iterations κ.

The requirements for the application of this algorithm are twofold: firstly, the object classes of input and target scale have to be the same. Secondly, the data has to be given in terms of an area partition.

3. Presentation and preprocessing of the test data

3.1 Test datasets

For our investigations we used two topographic datasets from different origins and with different resolutions (see Figure 1). Our test area has an extent of approximately 3 km x 2 km and is located in Goslar, a city in Lower Saxony in Germany. On the one hand we used the German digital cadastral information system ALK (Automated Cadastre Map) with a scale of 1:1,000 and on the other hand the official German topographic database ATKIS with a scale of 1:25,000. Both datasets are independently captured, maintained, and updated. Each dataset contains area, line, and point objects. Besides the geometric description, each object is also described semantically using the object classes and attributes described in the respective object catalogues. The classifications in the two object catalogues are similar – since both datasets describe topographic objects, yet they are different.

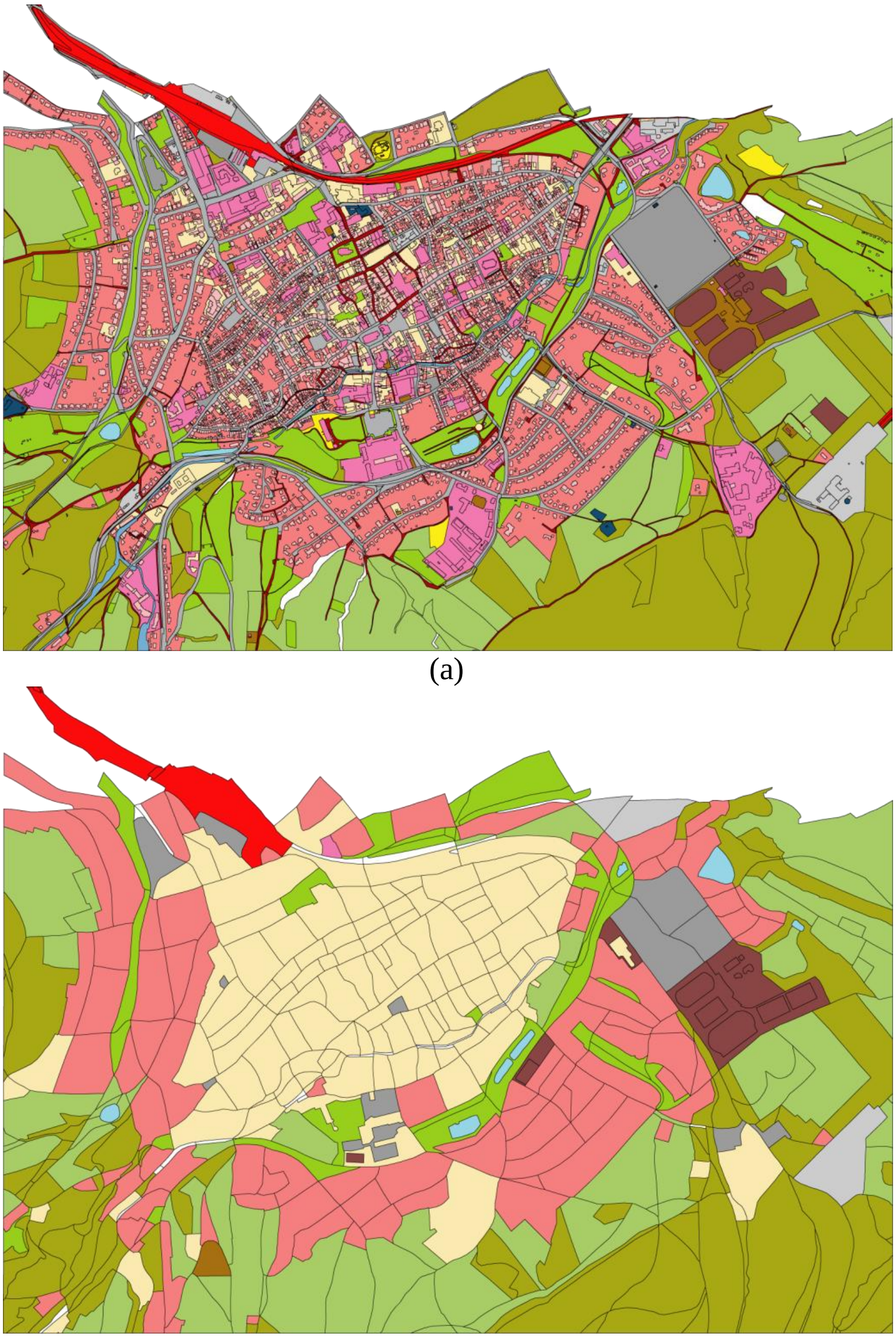
As the data does not satisfy the requirements of our aggregation approach we preprocessed the data according to our needs. These individual processing steps are presented in the following section.

3.2 Preprocessing steps of the input data

The preprocessing was divided into two parts. The first part considers the semantic level. The given object classes of both involved datasets differ in the detail level, due to the different scales: ALK contains 95 classes, whereas ATKIS contains 22 classes. As the datasets are provided and acquired by different organizations, also different object catalogues are used. As discussed above, for our aggregation algorithm the object classes in both datasets have to be identical. Hence we had to define new object classes from the existing classes of both datasets and came up with 16 common object classes (see Table 1).

The second part of preprocessing is carried out on the geometrical level. Since our approach is restricted to area objects, the data has to be preprocessed in order to form a planar partition. That means that overlapping areas and gaps in between have to be eliminated. The next preprocessing steps were only accomplished to the large-scale ALK dataset. In order to reduce the number of the involved objects, all adjacent objects of the same object class were merged. This lead to larger connected regions.

Some object classes in the large-scale dataset do not exist anymore in the small-scale dataset. Often object classes remain but a change of the geometry type is possible, e.g. a change from an area to a line or even to a point. In the large-scale dataset, narrow and elongated objects like roads, paths and rivers are represented as areas, whereas they are represented as lines in the small-scale dataset. In order to make the datasets more compatible, we eliminated all narrow objects by applying a collapse operator (see Figure 2) (Haunert & Sester 2008). This operator comprises two steps. In the first step all objects of the aforementioned object classes were merged and a skeleton was constructed from the resulting area. This operation leads to a decomposition into multiple fragments on both sides of the centreline. Subsequently each fragment can be merged with the adjacent polygon. After the collapse, a line simplification process follows. With this step smoother polygon boundaries are obtained. We applied the line simplification algorithm to all ALK objects (see Figure 2). With these preprocessing steps the number of ALK objects decreased from 8,172 to 3,797 objects.



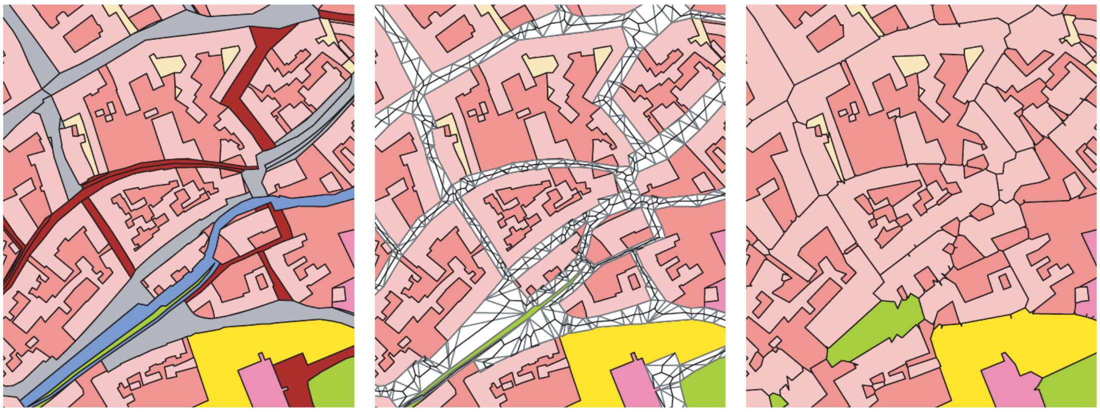
(a)

(b)

Figure 1. (a) The original datasets ALK (1:1,000) and (b) ATKIS (1:25,000).

Table 1. Definition of new common object classes from individual original object classes: in the table, the number of objects and the total *size* in the corresponding object classes are listed.

Object classes	# ALK	ALK size [m²]	# ATKIS	ATKIS size [m²]	Textual description
0900	2132	556,336	-	-	Residential buildings
1100	90	352,309	1	3,387	Public buildings and open space
1300	750	1,111,388	95	1,194,455	Residential area
1400	386	229,145	116	1,133,078	Commercial and service area
1700	32	92,936	3	77,002	Trade and industrial area
2800	17	39,952	1	10,223	Recreation area
4100	7	76,345	15	124,099	Sports area
4200	129	584,185	34	275,646	Green space
2500	45	12,928	-	-	Supply and waste management
5300	63	159,571	12	153,786	Square
5400	6	91,942	5	87,833	Railway area
9500	9	21,544	24	25,972	Unused area
6200	30	677,811	61	1,089,066	Grassland
7000	77	1,188,142	85	1,045,026	Forest
9300	15	22,489	-	-	Historical site
8800	9	26,302	7	23,752	Lake



(a) (b) (c)

Figure 2: Enlarged parts of: (a) The original ALK dataset including narrow polygons, like roads (light grey), paths (dark red) and rivers (blue). (b) Triangulation-based skeleton for the narrow polygons. (c) Resulting map after collapse operator and line simplification.

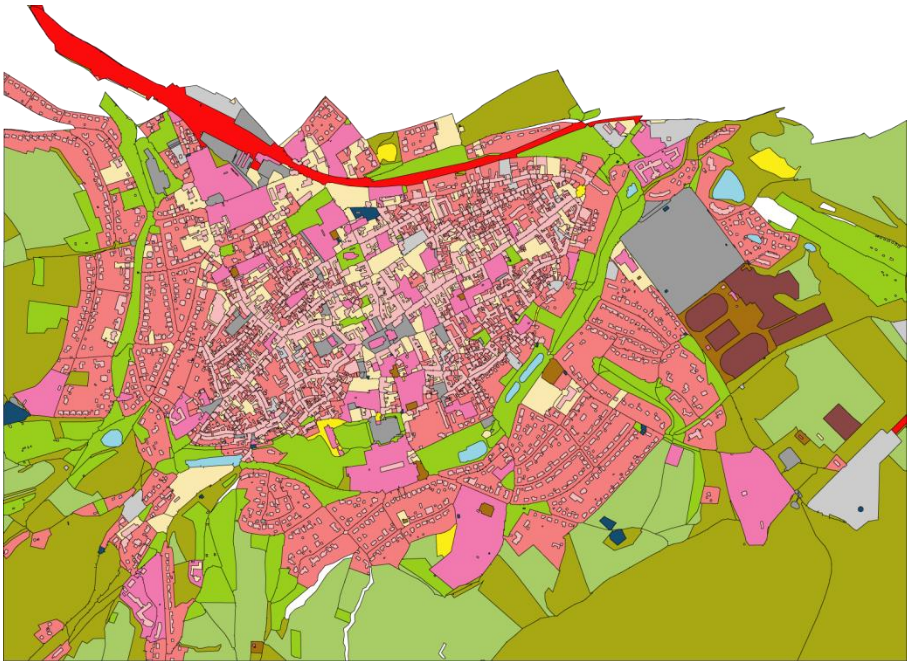


Figure 3: Preprocessed ALK data, after collapsed narrow polygons and simplified polygon borders.

4. Determination of transition matrices

The transition matrix between the involved object classes is an important control parameter for the aggregation process. It defines the class distance d and thereby the cost for class change f_{class}. In this section we present three options to derive this matrix. The underlying assumptions and objectives of these options are different, and thus will also lead to different results, described in section 5.

1) Typically an expert sets up a transition matrix T_1 manually based on his/her experience with the generation of a generalized dataset. In this case the major goal is to preserve similar object classes and thus enhance the underlying structure of the large-scale dataset. To this end, the expert first has to decide which object classes are similar, and therefore are assigned low cost values in the transition matrix, and which ones are important and have to be preserved and thus are assigned high costs. The higher the costs, the more expensive the class change will be. After defining suitable costs for specific class changes, the generalized dataset will be produced. Based on the resulting outcomes, modifications in the cost functions can be made and an iteration of the process started.

   A positive aspect of this method is that the method will end up with a satisfying map, as the expert is in the loop to optionally modify the cost functions to change the result. On the negative side, however, the process is time consuming as ultimately for each generalization task an expert is needed who is familiar with the specifications and meaning of the involved data. As this is not affordable when it comes to the processing of increasing numbers of datasets (e.g. integrated via Web-services) we propose two other options for automatically deriving transition matrices.

2) Sometimes we have, in addition to the dataset that we want to generalize, also a dataset in the target scale (e.g. consider the topographic map series of

National Mapping Agencies). In this case the scale transition matrix can be learned from the two datasets. The first idea is to reveal the relationships between the object classes using a geometric overlay of the datasets (this idea will be refined in case 3). To this end we calculate the intersection areas between the objects of both datasets. We summarize the results in the *intersection matrix* I_0, where each of its cells $I_0(i,j)$ contains the total area of changes from class γ_i to γ_j. Then we convert these values into percentages K_0 , where $K_0 \ i,j \ = I_0(i,j)/area \ \gamma_i$, and $area \ \gamma_i$ is the total area size of all objects of object class γ_i. Subsequently, we set up the transition matrix T_2 in order to get a low cost value for a high intersection rate and vice versa, with $T_2(i,j) = 100 - K_0(i,j)$.
This way of determining class correspondences to reflect the class transitions has both advantages and disadvantages. On the positive side is the simple and straightforward computation of the cost values for the transition matrix. On the negative side, it is not guaranteed that the result does completely reflect the underlying class correspondences. If, for example, the target dataset does not reflect the dominant object classes of the source dataset, we may lose these object classes. Furthermore, the intersection matrix describes the global percentages of the class transitions over the whole dataset – it may, however, lead to the fact that the object classes with higher likelihood are dominating the other possible transitions, leading to a different distribution in the resulting object classes.

3) In order to avoid the disadvantage of the second approach, we propose to imitate the iterative learning process of an expert. Therefore we use the derived relationships between the given datasets from the second approach as reference and then iteratively adapt the result of the aggregation to the real situation – both with respect to the number and area size of the resulting object classes, and their relative distribution. If the total area size of the aggregated solution concerning the individual object classes is higher than in the target dataset, the likelihood of transition has to be reduced. This is done in an adaptive way. We start the aggregation process with an initial transition matrix T_2. Then we improve the transition matrix during the following iterative process, until the defined stop criterion is fulfilled:

a. Calculation of the intersection matrix I_0 according to approach (2).
b. Aggregation using simulated annealing with transition matrix T_2; as a result we obtain a new intersection matrix I_n; $n =$ iteration step.
c. Calculation of the difference matrix $D_n = I_n - I_0$.
d. Set max to the maximal value of D_n at value of $D_n \ i,j$. We assume that the algorithm produces a too high class change $i \rightarrow j$, thus we can improve it by increasing the cost for this class change.
e. Increase the cost for this class change by a constant factor x. As a result we obtain a new transition matrix T_n, with $T_n = T_{n-1} + \ x$.
f. If max is greater than a certain threshold, repeat the procedure from step (b) with the modified transition matrix $T_{n\,.}$ As result we obtain a new intersection matrix I_{n+1}.

5. Experiments and evaluation of results

The generalization process using the aggregation algorithm described in Section 2 applying the different transition matrices was tested with the test area (see Figure 1). For our investigations we used the preprocessed ALK data (see Figure 3) and the original ATKIS data (see Figure 1(b)).

In the following we present the results of the aggregation approach for the three different transition matrices separately. After specifying the input parameters for each case, we show the resulting maps and evaluate them in two ways, firstly visually and secondly concerning the total area sizes of the object classes compared with the input datasets (see Table 1 and Table 5). The control parameters for the simulated annealing algorithm have been chosen the same for all scenarios, as follows: Initial temperature $T_S = 10^6$, final temperature $T_E = 10^3$, number of iteration $m = 3{,}797{,}000$ (which corresponds to 1000 times the number of objects), weight factor for compactness $\mathrm{s} = 0.8$ and the penalty factor $\theta - \mathrm{W} = 10^5$. In order to simplify the problem, we also defined the same target object size $\theta = 10{,}000$ m² for all object classes. All control parameters were determined empirically.

1) The first transition matrix T_1 (see Table 2) was defined manually, that is, we took the role of a cartographic expert. The goal of this first option was to generate a generalized dataset which reflects the structure of the detailed ALK dataset best. Based on our knowledge of the semantics of the object classes, we decided which object classes are similar or not and defined the cost values accordingly. For example the object classes *2800*, *4100* and *4200* represent recreation areas. We assessed them as similar classes and therefore we assigned short distances values in between them, i.e. 5. We wanted to preserve the object class *1400* thus we defined high cost values, i.e. 200, for changing this class into another one. The transition matrix T_1 is not symmetric; the main diagonal is filled with zeros, which indicates that existing classes should be kept with higher priority.

   In order to evaluate the result of our approach using combinatorial optimization (see Figure 4(b)), we also show the result obtained with a classical region-growing approach (Figure 4(a)). The main difference between these two solutions is the fact that our approach produces more compact results, i.e. leads to a more clear and simple visualization, thus a better generalization. Comparing the results with the original datasets confirms our assumption, that the maps are more similar to ALK than to ATKIS. That is, the city center is not dominated by only one object class, like *1400* in ATKIS, but is more heterogeneous, consisting of different settlement areas. However, also several object classes, like *2500*, *2800*, *8800*, *9300* and *9500* were eliminated, as they did not meet the given target object size.

   In Table 5 the total area sizes of the object classes for each test scenario are shown. Based on these values we find that the differences between the resulting areas to ALK are lower than to ATKIS in almost all classes, which also reflects our goals for this first case. For the future we recommend that important object classes like water objects *(8800)* have to be preserved, regardless of their size. This implies that the minimum size of the target objects has to be set differently for different object classes.

Table 2. Transition matrix T_1 defined by an expert

	900	1100	1300	1400	1700	2500	2800	4100	4200	5300	5400	6200	7000	8800	9300	9500
900	0	100	5	100	100	100	100	100	100	100	100	100	100	100	100	100
1100	100	0	100	5	5	5	100	100	100	100	100	100	100	100	100	100
1300	5	100	0	100	100	100	100	100	100	100	100	100	100	100	100	100
1400	200	200	200	0	200	200	200	200	200	200	200	200	200	200	200	200
1700	100	5	100	100	0	100	100	100	100	100	100	100	100	100	100	100
2500	100	5	100	100	100	0	100	100	100	100	100	100	100	100	100	100
2800	100	100	100	100	100	100	0	5	5	100	100	100	100	100	100	100
4100	100	100	100	100	100	100	5	0	5	100	100	100	100	100	100	100
4200	100	100	100	100	100	100	5	5	0	100	100	100	100	100	100	100
5300	100	100	100	100	100	100	100	100	100	0	100	100	100	100	100	100
5400	100	100	100	100	100	100	100	100	100	100	0	100	100	100	100	100
6200	100	100	100	100	100	100	100	100	100	100	100	0	100	100	100	100
7000	100	100	100	100	100	100	100	100	100	100	100	100	0	100	100	100
8800	100	100	100	100	100	100	100	100	100	100	100	100	100	0	100	100
9300	100	100	100	100	100	100	100	100	100	100	100	100	100	100	0	100
9500	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	0

2) The second result is based on the transition matrix T_2 (see Table 3) which was derived from the intersection matrix I_0 as described in Section 4 (2). We note that on the one hand the main diagonal of T_2 is filled, that is, keeping an object class can be more expensive than changing it into another one. On the other hand the matrix is also not symmetric, that is, a class change from one class to another can be more expensive than vice versa.

   With this approach the goal was to generate a generalized dataset which is similar to the ATKIS dataset. Figure 4(c) displays the resulting map applying T_2 in our proposed optimization-based aggregation algorithm. Visually the result confirmed our assumption. That is, the city center is mainly represented of polygons of commercial and service areas *(1400)*; the center is surrounded by residential areas *(1300)* and by green areas. None of the object classes which are only represented in ALK do exist in the resulting map, for example *0900*, *2500* and *9300*. More areas from *1300*, *5400* and *7000* are generated compared to ALK and ATKIS. The highest class changes are in the object classes *1100* and *1400*. In the generalized dataset the total area size of *1100* is much higher than in the ATKIS dataset and much lower than in ALK. For the object class *1400* we observe the opposite case. So we can conclude that in general the values in the scale-transition matrices were able to reflect the fact that in ATKIS new object classes dominate (class *1400*) and other object classes are reduced or no longer present (class *1100*), but some, e.g. *1300,* are overrepresented.

   The shapes of the polygon borders of the generalized map are directly not comparable with the polygons of the ATKIS data, because the polygon boundaries of the ATKIS data are oriented on the local road network. As our aggregation method focuses on the creation of geometrically compact polygons, this spatial constraint of the road network gets lost.

Table 3. Transition matrix T_2 from intersection matrix I_0

	900	1100	1300	1400	1700	2500	2800	4100	4200	5300	5400	6200	7000	8800	9300	9500
900	100	100	68	33	100	100	100	100	100	100	100	100	100	100	100	99
1100	100	99	92	22	98	100	100	99	100	95	99	99	96	100	100	100
1300	100	100	25	79	100	100	100	100	98	100	100	99	99	100	100	100
1400	100	100	74	37	99	100	100	96	99	100	99	98	100	100	100	99
1700	100	100	91	84	39	100	100	100	100	100	90	97	99	100	100	100
2500	100	100	94	38	98	100	100	100	99	98	100	90	82	100	100	100
2800	100	100	96	85	100	100	100	24	100	100	100	98	97	100	100	100
4100	100	100	100	100	100	100	100	17	100	100	100	83	100	100	100	100
4200	100	100	90	95	99	100	98	100	68	98	100	60	94	100	100	99
5300	100	100	98	84	100	100	100	99	99	27	94	100	99	100	100	100
5400	100	100	97	92	98	100	100	100	100	99	31	94	99	100	100	90
6200	100	100	100	100	100	100	100	100	100	100	100	1	99	100	100	100
7000	100	100	99	100	100	100	100	99	95	100	100	90	19	100	100	100
8800	100	100	96	83	100	100	100	100	99	100	100	99	100	25	100	98
9300	100	100	100	62	100	100	100	100	76	100	100	96	66	100	100	100
9500	100	100	99	100	100	100	100	100	100	100	100	9	93	100	100	100

3) Finally, we present the results based on the transition matrix T_3, which was learned during our proposed iterative process as described in Section 4 (3). The primary goal for this case was to generate a generalized dataset that reproduces the underlying object class relationships between ALK and ATKIS from the overlay procedure of approach (2).

Due to extensive computing time we changed the number of iterations for the simulated annealing process from $m = 3{,}797{,}000$ to $m = 379{,}700$. We restricted our tests to a number of iterations of $n = 50$. When evaluating the results we identified two criteria that our solution should fulfil: both the sum s_{D_n} of the individual elements of D_n and the max value should be small. As we aim at finding a trade-off between both criteria, the product of both values $s_{D_n} \cdot max$ gives us a reliable indication for a good solution.

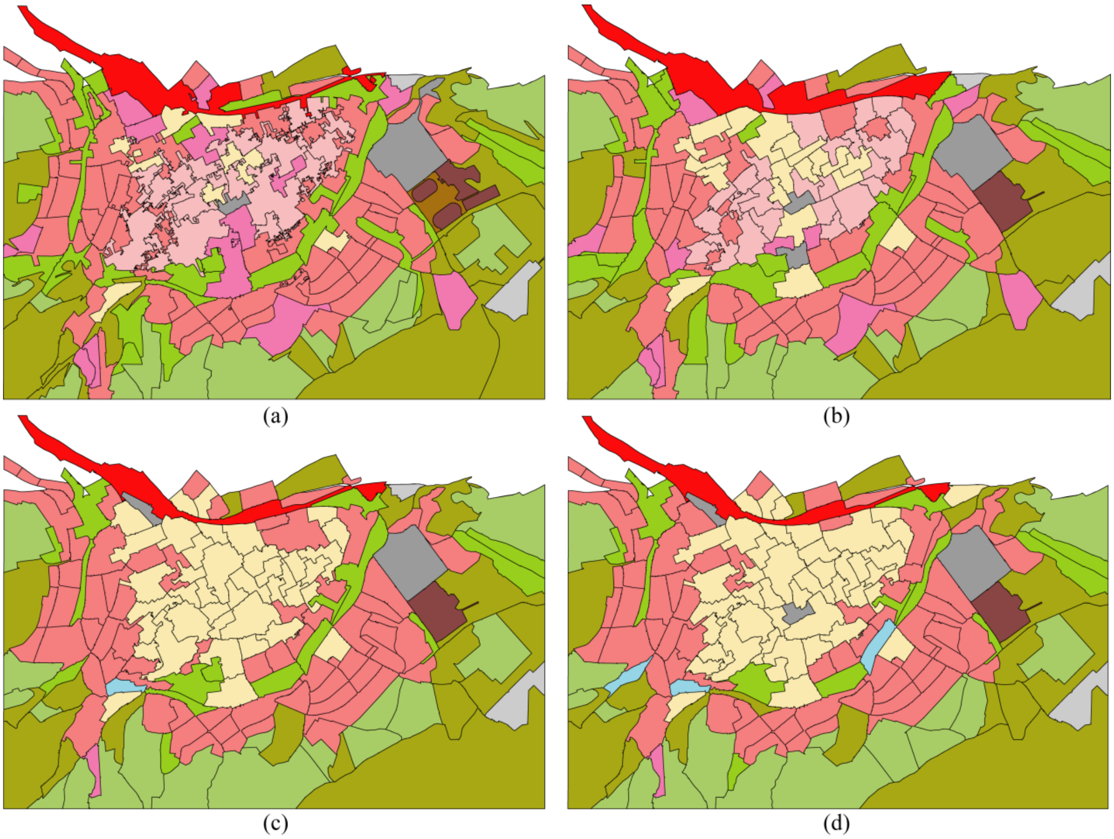
In our case, iteration 22 (of 50) yields the positive minimum of the product. The corresponding transition matrix is shown in Table 4 and the resulting map in Figure 4(d).

At a first glance, the result comes up to our expectation, an improvement compared to the result of case 2. In this case objects of the object class *8800* (lake) and *5300* (square) appear, reflecting the given scale-transition relations between the input datasets. These objects are created at locations where in both input datasets objects of these types exist. The objects are over-represented, because we defined a minimum target object size for all object classes. This strong constraint leads to the fact that this object creation is enforced. Taking a closer look at the area comparison the obtained result reveals to be more similar to the result of (2) and therefore also to the original ATKIS dataset. With a growing iteration rate we expect the results to be a trade-off between ALK and ATKIS, with an enhanced effect of prominent objects.

However, the results give us hints for further improvement. Instead of comparing the results of the iterations with the initial intersection result, we plan to compare it with the area distribution in the target dataset. In this way, the goal, namely to adapt the matrix elements in a way that it leads to the generation of a similar situation as in the target dataset, will be better reflected. Secondly, also individual target area sizes have to be determined for the different object classes. The threshold have to be adapted on the one hand to the minimum area size that are visually perceivable in the target scale; on the other hand the value also has to be guided from the real distribution of the object class in the datasets.

Table 4. Transition matrix T_3 learned in our iterative process (Iteration 22)

	900	1100	1300	1400	1700	2500	2800	4100	4200	5300	5400	6200	7000	8800	9300	9500
900	100	100	68	33	100	100	100	100	100	100	100	100	100	100	100	99
1100	100	99	92	22	98	100	100	99	100	95	99	99	96	100	100	100
1300	100	100	39	79	100	100	100	100	98	100	100	99	99	100	100	100
1400	100	100	74	37	99	100	100	96	99	100	99	98	100	100	100	99
1700	100	100	91	84	39	100	100	100	100	100	90	97	99	100	100	100
2500	100	100	94	38	98	100	100	100	99	98	100	90	82	100	100	100
2800	100	100	96	85	100	100	100	24	100	100	100	98	97	100	100	100
4100	100	100	100	100	100	100	100	17	100	100	100	83	100	100	100	100
4200	100	100	90	95	99	100	98	100	84	98	100	60	94	100	100	99
5300	100	100	98	84	100	100	100	99	99	27	94	100	99	100	100	100
5400	100	100	97	92	98	100	100	100	100	99	31	94	99	100	100	90
6200	100	100	100	100	100	100	100	100	100	100	100	1	99	100	100	100
7000	100	100	99	100	100	100	100	99	95	100	100	90	33	100	100	100
8800	100	100	96	83	100	100	100	100	99	100	100	99	100	25	100	98
9300	100	100	100	62	100	100	100	100	76	100	100	96	66	100	100	100
9500	100	100	99	100	100	100	100	100	100	100	100	9	93	100	100	100



(a)

(b)

(c)

(d)

Figure 4. Generalized maps resulting from different transition matrices: Case 1) Defined by an expert (a) using a region-growing approach and (b) using our optimization approach. (c) Case 2) Derived from the intersection matrix I_0 (d) Case 3) Learned during iterative process.

Table 5. Resulting area sizes of the object classes separated for each test scenario.

Object classes	Scenario (1) [m^2]		Scenario (2) [m^2]	Scenario (3) [m^2]
	Region Growing	Global Optimization		
0900	497,824	411,750	-	-
1100	363,208	237,774	13,524	13,524
1300	1,267,225	1,264,048	1,413,334	1,348,856
1400	123,809	323,336	802,586	876,486
1700	54,914	60,612	60,612	50,084
2800	33,047	-	-	-
4100	64,199	57,981	57,981	57,981
4200	565,741	517,557	385,378	301,084
2500	-	-	-	-
5300	122,421	128,273	104,979	117,365
5400	175,736	209,245	135,289	135,289
9500	-	-	-	-
6200	703,143	688,045	943,008	970,084
7000	1,271,860	1,344,706	1,314,087	1,324,486
9300	-	-	-	-
8800	-	-	12,548	48,087
# objects	161	166	162	162

6. Summary and outlook on future work

Haunert (2008) has presented an optimization-based aggregation method, which ensures results of high quality with respect to two criteria: it minimises object class changes and maximises the compactness of shapes. The result of the aggregation method is heavily dependent on the input parameters, which reflect the compatibility of aggregation objects, i.e. their scale-transition relationships. In this paper we have presented three different options for the determination of these compatibility values. One was the conventional way, where an expert determines the values by a thorough investigation of object catalogues. The main objective was however to automate the task in order to overcome the time consuming way of a manual definition. Therefore we used two datasets of different scales for learning the scale-transition parameters: Firstly, we derived the transition matrix by a simple geometric overlay analysis of the data instances. Secondly, we developed an iterative learning process of the transition matrix, imitating the interactive adaptation of a human operator. The measure that we minimized was the difference of the intersection areas of object classes between both datasets before and after the generalization.

In the paper we could show that it is possible to derive parameters for scale-transition matrixes to be used for aggregation. The parameters can be learned from two existing datasets. The underlying idea is furthermore, to use these parameters also to generalized datasets, where only the large-scale information is available. This, however, is subject to our further investigations. Similarly, we think that the approach is general enough to derive arbitrary functional relationships from two datasets relating to the same area. Thus it is not restricted to the scale-transition relationships.

The work relies on several assumptions and simplifications that have to be investigated in future work. The first relates to the definition of common object classes which was necessary for using the current version of the aggregation algorithm. Obviously, this definition influences the result of the aggregation procedure. In order to make it independent of this initial definition the algorithm has to be extended. Another issue in improving our aggregation method is to consider different size constraints for different object classes to avoid that important object classes, for example lakes, will be eliminated or overrepresented.

Acknowledgements

This research is funded by the German Science Foundation (DFG). The support is gratefully acknowledged.

References

Douglas, D. & Peucker, T. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer **10**(2): 112-122.

Duckham, M. & Worboys, M. (2005). An algebraic approach to automated geospatial information fusion. International Journal of Geographical Information Science **19**: 537-557.

Foerster, T. & Stoter, J. (2008). Generalisation Operators For Practice - A Survey at National Mapping Agencies. Proceedings of the 11th ICA Workshop on Generalisation and Multiple Representation.

Harrie, L. & Weibel, R. (2007). Modelling the overall process of map generalization. Generalization of geographic information: cartographic modelling and applications. Mackaness, W., Ruas, A., & Sarjakoski, T. Oxford, Elsevier.

Harrie, L.E. (1999). The constraint method for solving spatial conflicts in cartographic generalization. Cartography and Geographic Information Science **26**(1): 55-69.

Haunert, J.-H. (2007). Optimization Methods for Area Aggregation in Land Cover Maps. In: Proceedings of the 23rd International Cartographic Conference (ICC'07), 4-10 August 2007, Moscow, Russia.

Haunert, J.-H. (2008). Aggregation in Map Generalization by Combinatorial Optimization. Thesis (PhD). Leibniz Universität Hannover, In: Reihe C, Heft 626 of Veröffentlichungen der Deutschen Geodätischen Kommission, Verlag der Bayerischen Akademie der Wissenschaften, Munich, Germany.

Haunert, J.-H. & Sester, M. (2008). Area collapse and road centerlines based on straight skeletons. GeoInformatica **12**(2): 169-191.

Kavouras, M., Kokla, M. & Tomai, E. (2005). Comparing categories among geographic ontologies. Computers & Geosciences **31**(2): 145 - 154.

Kieler, B., Sester, M., Wang, H. & Jiang, J. (2007). Semantic Data Integration: Data of Similar and Different Scales. Photogrammetrie Fernerkundung Geoinformation (PFG) **6**: 447-457.

Lamy, S., Ruas, A., Demazeu, Y., Jackson, M., Mackaness, W. & Weibel, R. (1999). The Application of Agents in Automated Map Generalization. Proceedings of the 19th International Cartographic Conference of the ICA.

Mackaness, W., Ruas, A., & Sarjakoski, L., Eds. (2007). Generalization of geographic information: cartographic modelling and applications. Oxford, Elsevier.

Richardson, D. & Thomson, R.C. (1999). The 'good-continuation' principle of perceptual organization applied to the generalization of road networks. Proceedings of the 19th International Cartographic Conference of the ICA.

Rodriguez, M. A. & Egenhofer, M. J. (2004). Comparing geospatial entity classes: an asymmetric and context dependent similarity measure. International Journal of Geographical Information Science **18**(3): 229-256.

Schwering, A. (2008). Approaches to Semantic Similarity Measurement for Geo-Spatial Data: A Survey. Transaction in GIS **12**(1): 5-29.

Sester, M. (2005). Optimization approaches for generalization and data abstraction. International Journal of Geographical Information Science **19**(8-9): 871-897.

Staufenbiel, W. (1973). Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung großmaßstäbiger Gebäudedarstellungen, Fachrichtung Vermessungswesen, Universität Hannover.

van Oosterom, P.J.M. (1995). The GAP-tree, an approach to 'on-the-fly' map generalization of an area partitioning. GIS and Generalization - Methodology and Practice, Taylor & Francis.

Ware, J.M. & Jones, C.B. (1998). Conflict Reduction in Map Generalization Using Iterative Improvement. GeoInformatica **2**(4): 383-407.

Yaolin, L., Molenaar, M. & Kraak, K.-J. (2002). Semantic similarity evaluation model in categorical database generalization. Proceedings ISPRS Commission IV Symposium 'Geospatial Theory, Processing and Applications', 9-12 July 2002, Ottawa, Canada. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV (Part 4).