HIERARCHICAL STRUCTURES FOR RULE-BASED INCREMENTAL GENERALISATION

Jan-Henrik HAUNERT, Karl-Heinrich ANDERS, Monika SESTER

Institute of Cartography and Geoinformatics, University of Hannover, Germany
{jan.haunert, karl-heinrich.anders, monika.sester}@ikg.uni-hannover.de

Working Group II/2

KEY WORDS: Generalisation, Update, MRDB, Hierarchies

ABSTRACT

Multiple Representation Databases are structures that link different data sets based on common geometry and/or semantics. They offer the possibility to realize an efficient updating process in case of changes, as only one data set has to be updated and the changes can be propagated to the linked one. There are some prerequisites for the realisation of such a functionality which are elaborated on in the paper. First of all, functional dependencies between the data sets have to be known or made explicit. Secondly, partitioning schemes of space have to be available in order to make sure that local changes can be restricted locally and do not spread over the whole data set.

1 INTRODUCTION

A database which comprises topographic data sets of different scales and explicitly stores correspondences between features of different data sets as links is often referred to as Multiple Representation Database (MRDB). An MRDB can be used for different purposes such as the provision of zoom functionalities for digital maps, the support of complex multiscale analysis functions or the realization of a more efficient updating process. The approach for updating normally consists in a propagation of updates from large to small scale by generalisation methods. The effort for the collection of data is greatly reduced, since updates need to be collected for the largest scale only. The consistency between different data sets is increased.

The updating process is however only efficient, when it can be done automatically and in a way that it runs faster than generalising the whole data set again. Nowadays (model) generalisation techniques are available that can generalise a considerably large area like the whole state of Baden-Württemberg in Germany (36.000 km^2) in only one week (Urbanke, 2005). Although this speed is impressive, it still prevents generalising the whole data set anew, whenever changes occur in the data. Furthermore, this brute force is not required, since updates to the source data set of the MRDB are spatially limited and often only features from certain classes are involved. Because of this, also the influence of an update to the target data set is limited and only some features need to be treated. This approach is generally called incremental generalisation (Kilpeläinen and Sarjakoski, 1995). A prototype system that realizes these principles was shown by Harrie and Hellström (1999).

This vision of an incremental generalisation has several prerequisites: Firstly, in order to automatically determine a new representation in a linked data set, a functional dependency between the data sets has to be known. Thus, in addition to mere links in an MRDB, also a functional correspondence has to be available. Otherwise, having only the links, one could only determine which objects might be affected, but not in which way.

Secondly, in order to restrict the influence of a change locally, special considerations and assumptions are needed that reveal general dependencies among the data sets. Typically, there are major and important objects which serve as a kind of partitioning frame for the others. The street network in topographic data sets is such an example, which builds a tesselation of space. These objects also form boundaries that limit the effect of generalisation functions. Thus, having such boundary objects helps to restrict influences of changes within such a region locally. Such partitioning objects can be given in advance, like the streets in topographic data sets, however, in other data sets, they might not be known a priori. We could show, that such local constraints can also be generated by applying a strict generalisation scheme, namely aggregation, which generates hierarchical connections in the data.

In case of linking arbitrary data sets, these hierarchies forming a partitioning have to be determined from the data itself. In the paper we will present first concepts to determine these hierarchies in the data by examining the behaviour and the relations of objects in scale space: objects that 'survive' in a lot of generalisation levels can be considered as important and potentially serve as partitioning objects.

The presentation is structured as follows: first we give an overview of the two options to generate the linking structure (section 2). Subsequently, the possibility to derive a lower-scale representation using rule-based generalisation is shown (section 3). Then the two before mentioned prerequisites for incremental update are analyzed for both options in section 4. The update scenarios for MRDB's derived from generalisation as well as from data matching are sketched (section 4.1 and 4.2) and the concept for the incremental update for the first case is shown using the example of area aggregation (section 4.3). In order to transfer this approach also to data sets that were acquired separately and are linked by matching processes, we present a generalised approach (section 4.4) and propose mechanisms to determine the partitioning hierarchy (section 4.5). A summary and outlook conclude the paper.

2 MRDB SET-UP

The MRDB approach we have developed is based on the architecture of a federated database system (FDBS) (see Conrad, 1997; Sheth and Larson, 1990). Starting point for a FDBS are several existing databases which should work together to provide a global service, but keep their own local autonomy. Our prototype is based on an Oracle database system and ArcGIS from Esri. A more detailed description of the architecture can be found in (Anders and Haunert, 2004) and (Mantel and Lipeck, 2004).

The developed updating approaches are independent of these technical aspects, but important differences can be found in the two general cases that will be distinguished in this section.

2.1 SET-UP BY GENERALISATION

An MRDB can be set up by applying methods of model generalisation. Normally, the data set with the largest scale is the source and all other data sets are derived consecutively. In this case, correspondences between features result as a byproduct of the generalisation process and can be transferred directly to the link structure of the MRDB. Furthermore, generalisation rules and operators that are used for the MRDB set-up can be documented and provided for later updates. Also the sequence in which generalisation rules and operators are applied as well as the order in which different features are processed can be recorded.

2.2 SET-UP BY MATCHING OF EXISTING DATA SETS

Another possibility for an MRDB set-up is to utilize existing data sets that were collected independently. In this case links can be acquired with automatic matching techniques. Before matching or linking homologous objects of different datasets, correspondences between abstractions need to be found. This process often leads to a global schema or multi-scale schema. Devogele *et al.* (1996) define a multi-scale schema by indroducing scale-transition dependencies between classes from different schemas. For the matching task, attributes, geometric properties and/or topological relationships between features of different data sets can be compared. These three aspects are evaluated sequentially by Cobb *et al.* (1998) for each candidate match to exclude false matches. Walter and Fritsch (1999) propose a matching technique for road datasets based on statistical investigations of geometric properties such as the length or the angle of a line feature. Bruns and Egenhofer (1996) discribe the dissimilarity of spatial scenes by the number of steps, that are needed to transform one representation into the other. For this discretisations of relations like the distance relation are defined.

The major difficulty in the matching task is that data sets are often not consistent, since different cartographers can come to different decisions within a relatively broad scope of interpretation when mapping the real world. Especially topological inconsistencies, which can exist for example between road networks of different data sets are relatively hard to handle. Another problem is, that data sets of different scales are also often collected for slightly different thematic domains, since the derived maps are used for different applications. Thus correspondences between classes in existing databases that are built up as source for these maps can not always be found.
Although matches between features of data sets can be erroneous and incomplete in these cases, the storage of links is beneficial for a later update. The linking is part of the data enrichment process which is often conducted prior to a generalisation. Obviously, the matching and updating task becomes easier with more complete and less ambiguous specifications of the data sets. At best correspondences between classes are explicitly defined.
An example for these conditions can be found in the digital topographic data sets from the German ATKIS project. Consistent specifications are defined for digital landscape models of four different scales, but data sets are collected by different authorities, from different sources and at different updating cycles. For each class of features and each landscape model certain criteria such as area thresholds are clearly defined that need to be fulfilled. However, it is the task of the cartographer to decide how to treat the gaps that result from omitted objects. Also the boundary between different types of land use or vegetation is often rather fuzzy than crisp. Here it is the cartographer's task to find an appropriate representation. Thus, data sets of different scale can be quite inhomogeneous.

3 GENERALISATION

Approaches to the updating problem as well as aspirations from developed methods will be quite different depending on the existence of direct parentage between different data sets. The differences for the two possible cases described in section 2.1 and 2.2 will be discussed in section 4.1 and 4.2. Generally the availability of appropriate generalisation methods is necessary for both MRDB types. In the case of a set-up by matching, generalisation methods need to be defined which overcome the scale differences, when transferring updates between two data sets. In the other case generalisation methods are already needed for the initialization of the MRDB. To manage the generalisation problem, we developed procedures for the aggregation and collapse of areas as well as a rule based system which is used for the triggering of generalisation operators.

3.1 REPRESENTATION OF RULES

In our MRDB approach the rule based system represents the controlling unit for all generalisation and update processes. It enables a flexible behaviour of our system without hard-coded thresholds and workflow of basic generalisation and update operations. A main requirement for the rule based system is that it should be able to store all rules and basic operations in a relational database. Therefore we design a simple inference machine based on the concept of Horn-clause rules, well known from logical programming in Prolog.

The inference machine for our MRDB is managing six tables stored in the database.

- Rule table *(RepModel, OType, RId, Priority, OpId, PSetId)*
- Constraint table *(RepModel, OType, CId, Attribute, RelOp, Threshold, Measure)*
- Conjunction table *(RId, CId)*
- Operation table *(OpId, ReturnType, StoredProcedure)*
- Interface table *(OpId, PNum, PType)*
- Parameter table *(PSetId, PNum, PValue)*

In the rule table all rules triggered by a representation model (*RepModel*) and an object type (*OType*) are stored. For every rule one has to define a rule id (*RId*), a priority (*Priority*), an operation id (*OpId*), and a parameter set id (*PSetId*). The primary key of this table is defined by (RepModel, OType, RId). The rule priority can be used to define the processing sequence of several rules triggered at the same time by (RepModel, OType). The OpId is defining an entry in the operation table which stores the stored procedure (*StoredProcedure*) to be called if the rule fires and the result type (*ReturnType*). The PSetId is used to find all parameters in the parameter table which should be used with the operation. A rule fires if the conjunction of all constraints related to the rule evaluates to true. The relation between rule and constraints is stored in the conjunction table. A constraint defines a boolean relation (*RelOp*) between the value of a certain object attribute (*Attribute*) and a defined threshold (*Threshold*). One has to define for every constraint a constraint id (*CId*). For a flexible scaling handling we are using the column *Measure* in the constraint table. The primary key of the constraint table is (RepModel, OType, CId).

Every operation is uniquely defined by the signature of the operation which is stored in the interface table for semantic tests and for additional information for the user. The signature is defined by the interface id (*OpId*), parameter number (*PNum*), and the parameter type (*PType*). The parameter number defines from left to right the calling sequence of the parameters. For a flexible user definable parameterization of the registered operations we are using the parameter table to store user defined parameter sets. The parameter values are stored in the column (*PValue*).

The simple inference procedure in our MRDB can be described as follows:

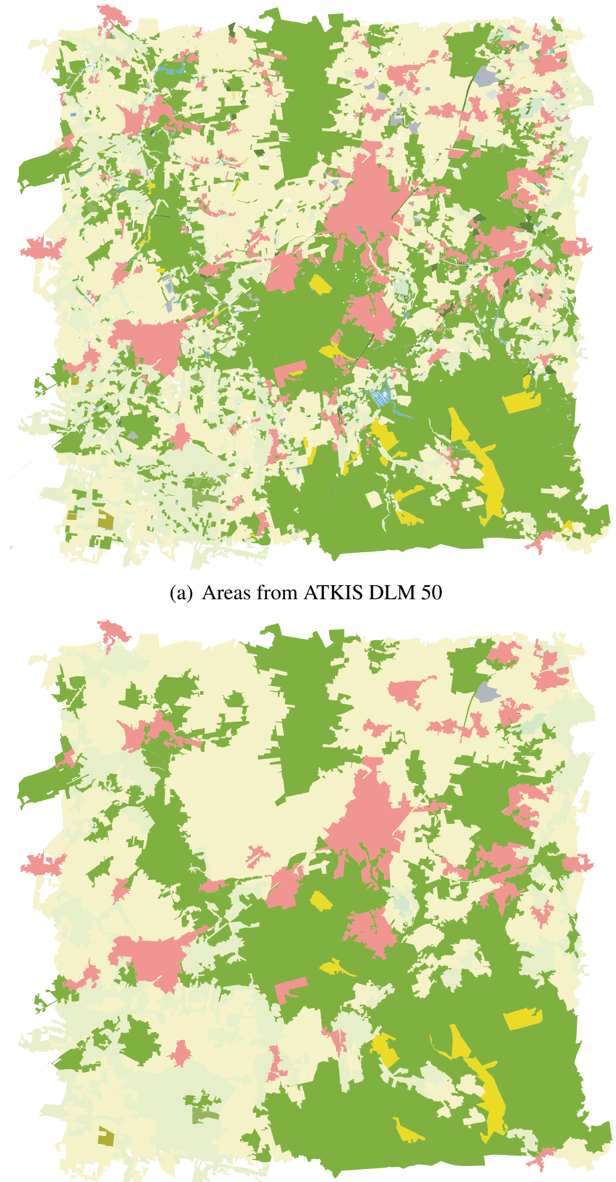
1. Clear selection set Σ
2. Select all changed objects $\rightarrow \Sigma$
3. While Σ is not empty do
4. ...Select and remove an object ω from Σ
5. ...Select all rules triggered by ω
6. ...Process all firing rules and update Σ if needed
7. Done

3.2 AGGREGATION OF AREAS

The area features in the ATKIS data sets constitute a tesselation, which does not allow gaps or overlaps. Thus features which become insignificant when reducing the scale can not simply be omitted. Instead several features need to be aggregated to fulfil the requirements for the target scale. This aim can be achieved with the algorithm used by van Oosterom (1995) for the construction of a Generalised Area Partitioning-tree (GAP-tree). In our implementation the features are selected in ascending order of their areas and are merged to one of their neighbours, until all areas fulfil the requirements of the ATKIS specification. The choice of the neighbour for the merge depends on ranks for transitions between feature classes as well as the areas of the neighbours. The initial situation as well as the results are shown in figure 1. The number of features was reduced and a simpler map was derived. Additionally, the length of the common boundary could be included in the analysis to obtain less intricate geometries.

3.3 COLLAPSE OF AREAS

Sometimes the elimination of an insignificant area by merging with an adjacent feature is not possible without a drastic change of the neighbour's shape. The grey area in figure 2(a) is an example for this. Here, a better solution can be reached by separating the released area into multiple parts and assigning these to different neighbours. This task can be solved using a collapse operator which is based on the Straight Skeleton shown in figure 2(b). Each area that is enclosed by edges of the skeleton will be assigned to the neighbour which shares the incident polygon edge. Figure 2(c) shows the resulting map after this elimination. The released area has been assigned to its neighbours with little changes of their shapes.
A problem which is very similar to the discussed elimination of an area appears if a feature is defined to be represented with different geometry types at different scales. Also here the released area needs to be assigned to its neighbours, but additionally a linear representation must be created for which a centreline of the skeleton can be used. This collapse operator is capable of preserving topological constraints as well as performing partial geometry type changes (Haunert and Sester, 2004).



(a) Areas from ATKIS DLM 50

(b) Aggregated areas according to specifications of ATKIS DLM 250

Figure 1: Aggregation of Areas.

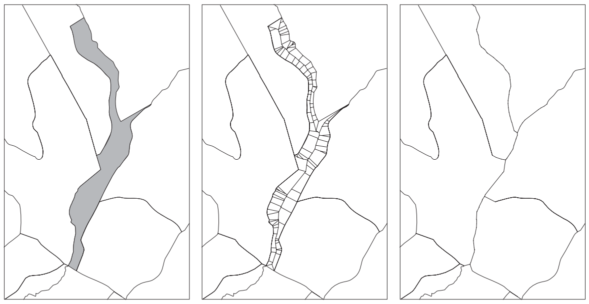
4 INCREMENTAL UPDATE OF MRDB

The objective of incremental update is to introduce changes in only one data set and propagate them to the representations they are linked with. Ideally, this is an automatic process that can be solved locally considering only the affected objects and possibly its neighbours. The propagation, however, presumes that there is a functional dependency between the linked data sets, in particular between the linked objects. In the case that the links have been generated by generalisation, the functional dependency between the two data sets is known – it is the generalisation function, namely:

$$Object_{smallScale} = f(Object_{largeScale})$$

with f being a generalisation function. When changes occur, the known generalisation function has to be applied to generate the derived object(s).

In case of matching independently acquired data sets, however, the links merely indicate a 'connection', but there is not neces-



(a) (b) (c)

Figure 2: Collapse Operator based on Straight Skeleton

sarily a known functional dependency between them. Therefore, it is difficult to derive an action which has to be triggered once one of the related partners is affected by a change.

There are different ways to overcome this problem. First of all, there might be known relationships that can be coded in semantic and geometric transformations. E.g. a data set from traffic navigation contains roads of different classes, whereas a topographic data set also contains roads, however possibly classified according to a different scheme. Knowing a transformation between the data sets represents a functional dependency between them, which then can be applied in case of updates:

$$Object_{dataset_1} \quad = \quad f(Object_{dataset_2})$$

with f being a semantic (and possibly geometric) transformation function (e.g. in case of linking parcel based road descriptions to linear roads from topographic data set, the geometric function corresponds to the collapse operator).

If the functional dependencies are not known, they can be determined using known examples given with links (see Section 4.2). As then the functions are determined using inference or estimation procedures, there will typically be no unique set of functions. Therefore, measures of quality or reliability of the functions are needed, allowing to use these measures to propagate also the quality of the derived object after the transformation:

$$Object_{dataset_1} \quad = \quad f(Object_{dataset_2}) + \sigma$$

In both cases, namely when the links are generated by generalisation functions and using matching, the correspondence function will not only depend on single objects alone, but on the local context they are embedded in, leading to the general relationship:

$$Object_{dataset_1} \quad = \quad f(Object_{dataset_2} | context) + \sigma$$

4.1 OBJECTIVES AND APPROACHES IN CASE OF SET-UP BY GENERALISATION

The general goal of the incremental generalisation is to achieve the same result as the global generalisation of the whole data set. An updating method which satisfies this objective ensures, that the current state of the database is independent of the order in which updates were inserted and it can be restored at any time by performing the global generalisation.

Often changes to features can be triggered directly to corresponding features in the smaller scale by following the links. However, the development of an incremental method is not trivial. In many cases generalisation is not organized by basic operations that have single features as input and produce generalised versions. Normally, the context of a feature or rather its relationships to other features in the neighbourhood is evaluated by generalisation rules and operators. Thus, dependencies between different data sets are generally quite complex. Knowing these dependencies is necessary for the incremental update. Often dependencies conform to certain rules. This occurs for example if features are generalised while regarding the relationships to their direct neighbours. This leads to the first option, namely using the general rules. This presumes, however, that the rules can be applied categorically – and there are no exceptions to them.
In order to also be able to take care of exceptions, the second possibility is to record the whole initial generalisation process and store it in terms of productions (Skogan and Skagestein, 2005).

4.2 OBJECTIVES AND APPROACHES IN CASE OF SET-UP BY MATCHING

In the case of independently collected and matched data sets other objectives need to be postulated for an updating method. Since generalisation methods are not explicitly given in advance, the optimal solution for the incremental update is not well defined. The problem is a more typical generalisation task, which will be solved differently by different cartographers. Here the objective is to resolve all inconsistencies between two data sets that result from an update of the larger scale by altering the smaller one. This can be done within the scope of interpretation that is left by the specifications.
A possible approach is to define new generalisation rules and operators that are evaluated and applied each time an update is performed. Herein the updated features as well as influenced and influencing features need to be evaluated. The search for these features that need to be considered is a specialty of the updating task. The term incremental generalisation can be used also here, since a limited number of features is evaluated.

4.2.1 Utilizing given samples Another difference to the common generalisation problem is that a large quantity of samples can be found in the existing data sets in terms of the links, that result from the matching of both data sets. Exploiting these links greatly supports the analysis of the relationships between features of both data sets. Thus, it can be attempted to reveal the preferences of the cartographer who collected the target data set and to perform the generalisation according to these, instead of defining methods by oneself. Machine Learning techniques (see e.g. Han and Kamber, 2001) can be applied for the extraction of this information. To proceed so, several preliminary fixings need to be defined such as the structural syntax of rules and their possible parameters.

An approach to automate the decision among several alternative generalisation actions is the usage of cost or benefit functions. Often different parameters are involved that need to be weighted. The automatic estimation of weights for these cost functions and the training of parameters like thresholds that are needed for generalisation operators are tasks, which can be solved with given examples.
These techniques are based on fixed structures between input and output parameters and modifications are only possible by adjusting certain settings. The famous learning technique ID3 (Quinlan, 1986) results in an optimal decision tree based on a given set of attributes.

4.2.2 Coping with geometric differences of data sets The cartographer's decision on the exact location of features and their delineations is sometimes intervened by chance. For example the geometry of road networks from different data sets is not exactly the same. Because of this it does not suffice to adapt features to the target scale with generalisation methods. Corrections need to be applied to the original location of transferred features, so that the topological relationships are not lost. As long as topological differences between both data sets do not exist or generalisation methods are capable of resolving these, the geometrical corrections can be performed using rubber sheeting techniques. Also here the access to previously found feature correspondences is beneficial.

Figure 3 shows an approach for the correct transfer of point features from the source data set into the target data set. In the initial situation only the source data set contains towers. Roads, rivers and railways exist in both data sets and feature correspondences are given in advance. The specifications of both data sets do not differ with respect to the representation of towers, thus generalisation is not needed. However, relationships to other features can be lost, if no corrections are applied that cope with geometric differences of both data sets. Such a problem occurs, if a tower happens to be on opposite sides of a road, since an offset between the road centrelines of both data sets exist.

To avoid such problems links are analysed in a first step to find control points that define geometric corrections. In a second step an interpolation function is applied to derive corrections for new features. Now these can be integrated correctly into the target data set. The applied technique allows to define an infinite number of control points on corresponding line segments (Haunert, 2005).

4.3 INCREMENTAL AGGREGATION OF AREAS

The development of a solution for the automatic incremental update of an MRDB is a very extensive task. Even in the case of an MRDB set-up by automatic generalisation, a general solution is not obvious, since generalisation operators can be very complex and interactions between different operators can exist. Both results in complex dependencies between features in the target data set and features in the source data set. Though a system that generally describes these dependencies can not simply be found, certain types of dependencies result from typical generalisation methods. Very often hierarchical approaches are used for generalisation problems, such as the aggregation of areas in section 3.2. Areas are iteratively merged with their neighbours and thus an aggregation hierarchy is constructed which can be regarded as a set of trees, in which each node represents an area of the source data set, an area of the target data set or an area of a provisional result.

Assuming first, that the topology of the trees is fixed, an update like the change of an area's land use could be propagated to the target data set simply by following the edges upwards within the affected tree. Each node that is reached on this path would need to be evaluated anew.

For the presented aggregation algorithm the assumption of a fixed tree structure does not generally hold. Indeed a modification of the source data set can have vast impacts on the topology of the trees, since the selection of a neighbour for a merge depends on properties of features which might have been changed. Thus the aggregation needs to be broken up and generated anew for some features. The influence of an update does not only spread to neighbours of features that where directly affected by an update. Also a chain of influences to neighbours can happen, which means that more and more features need to be comprised in the incremental generalisation process.

With these considerations an incremental aggregation algorithm can be found that produces the same result as the global aggregation. Thus the objective described in section 4.1 is satisfied. The algorithm is illustrated in figure 4 and can be defined as follows.

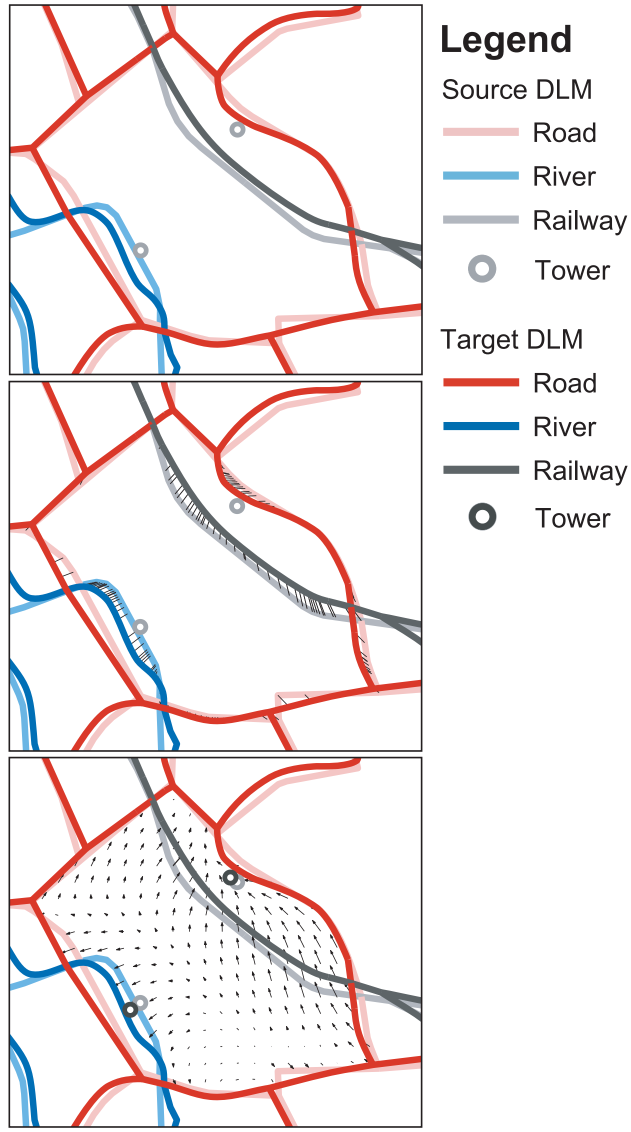


Figure 3: Conflation of two data sets by corresponding network features (roads, railways, rivers) to integrate new features (towers) into the target data set.

Top: Initial situation, towers exist in source data set only.

Center: Definition of corresponding points by network feature correspondences, that are given with links.

Bottom: Rubber Sheeting Transformation to integrate towers into target data set.

Definitions:

- *The region of features in the target data set, which covers all updated features in the source data set is defined as "interior region".*
- *The region which is covered by all features in the target data set that are adjacent to the interior region is defined as "boundary region".*

Process of algorithm:

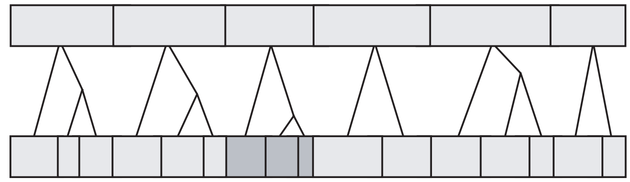
Apply the algorithm for the global aggregation of the source data set only on those areas that are in the interior region or the boundary region. As before the processing sequence of features is defined according to the size of their areas. Also the criteria that define the selection of a neighbour for the merge are adopted.

As long as no change happens to the boundary region, this restriction of the investigation area is valid. Such a change can happen in two cases. The first case is that, due to a change of a feature that is selected during the generalisation process, it is not assigned to the same adjacent feature as before. The second case concerns the other way round: During the construction of the tree a selected feature is not assigned to the same neighbour as before, because a change has happened to one of its neighbours. In both cases the change will be noticeable by a merge of a feature from the interior region with a feature from the boundary region. As mentioned, this can cause further changes to adjacent objects. Hence, the area under examination has to be extended.

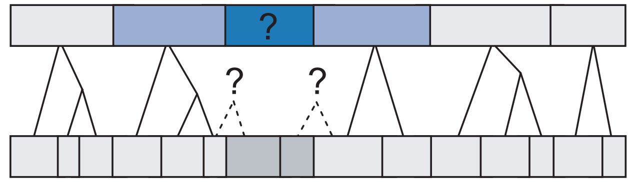
If the separation of the interior region and the boundary region is violated by an area merge:

- *Stop the aggregation.*
- *Extend the interior region on the features in the target data set, which cover the area which was returned by this merge.*
- *Extend the boundary region on the adjacent features.*
- *Perform the area merges within the region which was added to the boundary region, that are defined to happen earlier than the merge which violated the separation of the interior region and boundary region.*
- *Continue with the aggregation of features, that are within the interior region or the boundary region.*

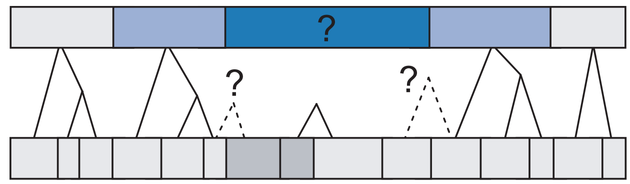
Figure 5 shows an example for the influence of a change. Applying the global aggregation algorithm to the map shown in figure 5(a) results in the map shown in figure 5(c). The sequence of area merges leading to this result is defined by the size of the areas. At first the small forest area is merged with the adjacent forest. Secondly a grassland area becomes merged with an adjacent area of the same vegetation. After the insertion of an update, this feature is treated differently 5(b). Due to a change of its vegetation to forest it is preferred to merge the feature to another neighbour. This small alteration has vast influences on the following processing steps since the two big forest areas become adjacent by this merge. Consequently, the results depicted in figures 5(c) and 5(d) are very different. The change does not only result in different geometries of the corresponding feature and its neighbours. Also the grassland feature on the right side was influenced, though no direct adjacency to the changed feature exists.
In Figure 6 the process of the incremental algorithm is illustrated,



(a) Initial situation: Source data set (bottom) and target data set (top).



(b) Updating of source data set (dark grey areas). Corresponding feature in target data set needs to be generated anew (interior region, dark blue area). Also possible influences on neighbour features need to be tested (boundary region, light blue areas).



(c) Due to a merge of an area from the interior region with an area from the boundary region, the neighbour on the right side becomes involved. The area of investigation is extended.

Figure 4: Expanding the influence of an update within an aggregation hierarchy.

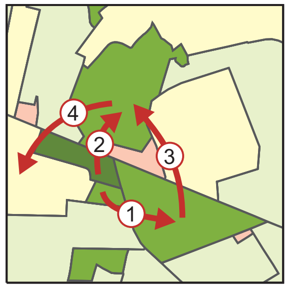
which produces the same result as shown in figure 5(d). The restriction to the initial interior region and boundary region holds until the second merge is processed, which combines both regions (figure 6(a)). Before further merges are processed the regions need to be expanded (figure 6(b)). Figure 7 shows the final stage of both regions. The number of features that needed to be processed was greatly reduced compared to a global processing.
The topology of the areas is not explicitly modelled in the database. Due to the incremental approach, the construction of an adjacency graph was restricted to the areas within the boundary region and was performed progressively. Because of this a great deal of computation time was saved.

The bottom-up construction of the aggregation hierarchy as well as the spread of the influence to neighbour features are examples for the treatment of two different types of relationships. The first example shows that hierarchical or vertical relationships are important for an update. The propagation of updates begins on the level of features from the source data set. Later areas that result from merges which can be seen as superior structures become involved. The second example shows that also horizontal relationships are important. The influence can spread successively to adjacent features of the same level of abstraction.
Though the presented incremental algorithm is designed for a special aggregation method, these general concepts can be transferred to other generalisation cases, since similar approaches are very common to solve different problems. Especially it is aimed to formulate the approach in a more general way, so that it can be adopted to the update of an MRDB that was set up by matching of existing data sets, which will be explained in the next section.

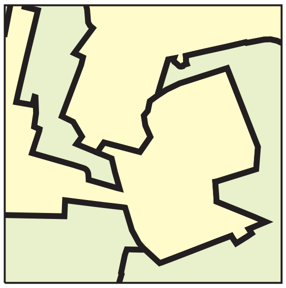
4.4 GENERALISED HIERARCHICAL APPROACH FOR INCREMENTAL UPDATE

The procedure in section 4.3 for the derivation of an influence area can be formulated in a general way. First the area of inves-

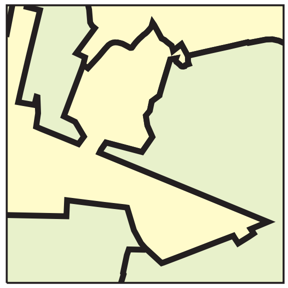
(a) Sequence of merges before update (not all merges shown)



(b) Sequence of merges after update

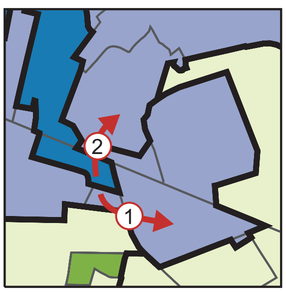


(c) Result of (a)

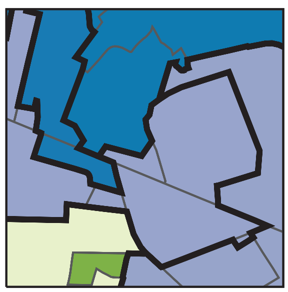


(d) Result of (b)

Figure 5: Influence of change to global algorithm.



(a) Second merge violating separation of interior region and boundary region



(b) Interior region and boundary region after expansion

Figure 6: Incremental algorithm to obtain same result as shown in figure 5(d)

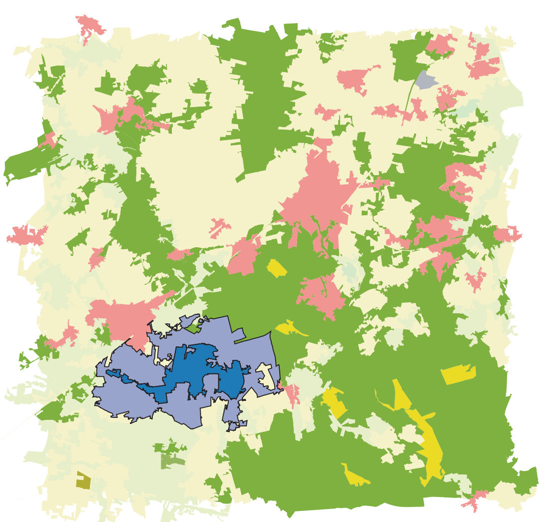


Figure 7: Derived influence regions for the change of a feature's land use atrribute value. Interior region depicted in dark blue, boundary region in light blue.

tigation for the generalisation can be restricted to a local scale, including features that are directly affected by an update. In the following the scope can be extended to neighbours and superior structures, until the final extend of the influence area is reached and all dependent objects in the target data set are considered.

A requirement for this approach is, that the hierarchy of objects is known, so that the triggering of updates can be performed step by step from a local, to a regional and to a global scale. The presented incremental algorithm operates successfully, since this sequence is uniquely defined for the applied aggregation operator.

Transferring the methodology to an MRDB which was set-up by matching of different data sets is only possible if similar hierarchical structures exist, that allow a structuring of the generalisation task into modules of different abstraction levels.

Naturally geographic objects are organised in hierarchical structures (Timpf, 1998). For the purpose of a spatially limited update propagation, the utilization of these hierarchies is aimed. Examples for these hierarchical structures can be found in road networks, which consist of roads that have different priority. Consequently a hierarchical partition of the space exists, that can be used to divide the generalisation into subproblems. This approach is reasonable, since roads naturally have a seperating effect. For example, within generalisation it is often avoided to merge features which are located on opposite sides of a road. Therefore, the investigation area for an update could be restricted first to the smallest enclosing mesh of the road network. If the impact of the update turns out to exceed this extend, a broader investigation area needs to be defined. This could be done either by adding neighbouring meshes or by proceeding with the next superior mesh in the road network.

The priority of road features is often defined by an attribute value expressing its type, which could be side road, main road or highway. In this case the hierarchy in the example is explicitly given. However, it will be more common, that no explicit knowledge about these structures exists.

4.5 DETECTING UNKNOWN HIERARCHICAL STRUCTURES

For the case that explicit knowledge about hierarchies does not exist, analysis techniques can be applied to derive these hierarchical structures. In the example of the road network it can happen that no attribute is defined which expresses the priority of roads. Generally this shortcoming can be overcome, since roads of higher priority have special characteristics. A highway for example distinguishes itself from roads of lower importance, since it connects towns at long distances and on a straight route.

This analysis can be supported by previously found correspondences between representations of features at different scales that are given as links in the MRDB. Generally, the following observations can be made when tracking a feature across different scale levels.

Duration of existence:
Inferior objects are eliminated earlier than superior objects.

Geometric persistence:
Inferior objects are displaced more likely and change their location. Superior objects are often emphasized by enlargement.

Thematic persistence:
An object resulting from a merge will adopt the value for a thematic attribute more likely from the superior element. The thematic features of the inferior element are discarded.

Persistence of relationships:
Topological relationships that have a long persistence indicate

significant structures like superior boundaries between adjacent area features.

All these characteristics support the structuring of the data set. However, the detection of hierarchies is a complex task, since relevant hierarchical structures not only exist between adjacent objects. A sophisticated aggregation method should be able to detect groups of objects that collectively form a superior structure, though no direct neighbourships exist. Areas with the land use "settlement" for example are often not adjacent, due to a separation by smaller structures like parks or traffic areas. The presented aggregation algorithm is not capable of detecting the relationships between these areas, since only direct neighbours are taken into consideration for a merge. The development of a more sophisticated area aggregation method is aimed, which will be approached utilizing a hierarchical parameter free clustering algorithm (Anders, 2004).

This is based on neighbourhood graphs and exploits the natural hierarchy of neighbourhood graphs to detect clusters of different granularity and thus has the advantage, that no thresholds or parameters are needed. Neighbourhood graphs capture proximity between points by connecting "nearby" points with a graph edge. The many possible notions of "nearby" lead to a variety of related graphs. It is easiest to view the graphs as connecting points only when a certain region of space is empty. In our approach we use the following neighbourhood graphs: The Nearest Neighbour Graph (NNG), Minimum Spanning Tree (MST), Relative Neighbourhood Graph (RNG), Gabriel Graph (GG), and the Delaunay Triangulation (DT). The important relationship between these neighbourhood graphs is that they build a hierarchy:

$$\mathsf{NNG} \subseteq \mathsf{MST} \subseteq \mathsf{RNG} \subseteq \mathsf{GG} \subseteq \mathsf{DT} \tag{1}$$

In our approach we use the hierarchical relationship between proximity graphs to represent a local to a global neighbourhood model.

5 CONCLUSION

Often a generalisation function does not exist, which defines direct dependencies between features of different topographic data sets. Thus, the incremental update of an MRDB is generally a rather complex task. The development of an updating method requires the availability of different techniques, such as analysis methods for the detection of the influenced features, appropriate generalisation operators and methods for the overcoming of geometric differences between data sets. Approaches for these problems and their applicability to MRDB resulting from different set-up methods have been discussed.

We propose a hierarchical structuring of the data sets to reduce the complexity of the incremental generalisation problem. Hierarchies which are explicitly given by attributes and class memberships can be utilized, but also analysis techniques can be applied to reveal unknown hierarchical structures. Links that express feature correspondences can be exploited for this task.

ACKNOWLEDGEMENTS

This work is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) and the Bundesamt für Kartographie und Geodäsie, BKG, Frankfurt.

References

Anders, K.-H, 2004. *Parameterfreies hierarchisches Graph-Clustering Verfahren zur Interpretation raumbezogener Daten*. Dissertation, Universität Stuttgart (http://elib.uni-stuttgart.de/opus/volltexte/2004/2024/), Stuttgart. Deutsche Version als PDF-Datei URN: urn:nbn:de:bsz:93-opus-20249 URL: http://elib.uni-stuttgart.de/opus/volltexte/2004/2024/.

Anders, K.-H and J Haunert, 2004. Mrdb to connect 3d city models, cadstral, and topographic data together,. In *Proceedings of 24th Urban Data Management Symposium*, Chioggia (Venice), Italy.

Bruns, T and M Egenhofer, 1996. Similarity of spatial scenes. In M.-J Kraak and M Molenaar (Eds.), *Seventh International Symposium on Spatial Data Handling (SDH '96), Delft, The Netherlands*.

Cobb, M, M Chung, H Foley, F Petry, K Shaw, and H Miller, 1998. A rule-based approach for the conflation of attributed vector data. *GeoInformatica* 2(1), 7–35.

Conrad, S, 1997. *Föderierte Datenbanksysteme : Konzepte der Datenintegration*. Springer-Verlag, Berlin.

Devogele, T, J Trevisan, and L Raynal, 1996. Building a multi-scale database with scale-transition relationships. In M.-J Kraak and M Molenaar (Eds.), *Seventh International Symposium on Spatial Data Handling (SDH '96), Delft, The Netherlands*, Volume I, pp. 6.19–6.33.

Han, J and M Kamber, 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Harrie, L and A.-K Hellström, 1999. A prototype system for propagating updates between cartographic data sets. *The Cartographic Journal* 36(2), 133–140.

Haunert, J.-H, 2005. Link based conflation of geographic datasets. In *Proc. of ICA Workshop on Generalisation and Multiple Representation, 7-8 July 2005, La Coruña*.

Haunert, J.-H and M Sester, 2004. Using the straight skeleton for generalisation in a multiple representation environment. In *Proc. of ICA Workshop on Generalisation and Muliple Representation, 20-21 August 2004, Leicester*.

Kilpeläinen, T and T Sarjakoski, 1995. Incremental generalization for multiple representations of geographic objects. In J. P. L J. C. Muller and R Weibel (Eds.), *GIS and Generalization: Methodology and Practise*, pp. 209–218. Taylor & Francis, London.

Mantel, D and U. W Lipeck, 2004. Matching cartographic objects in spatial databases. In *Proceedings of the XXth Congress of the ISPRS, Istanbul, July 2004, Commission IV Papers*, Number Vol. XXXV, Part B4 in Int.Archives of Photogrammetry,Remote Sensing and Spatial Inf.Sciences, pp. 172–176.

Quinlan, J, 1986. Induction of decision trees. *Machine Learning* 1(1), 81–106.

Sheth, A and J Larson, 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys* 22(3), 183–236.

Skogan, D and G Skagestein, 2005. An automated incremental generalization method. In *Multi-Resolution Geographic Data and Consistency (Dissertation by David Skogan)*. Faculty of Mathematics and Natural Sciences, University of Oslo.

Timpf, S, 1998. *Hierarchical structures in map series*. Ph. D. thesis, Technical University Vienna.

Urbanke, S, 2005. Modellgeneralisierung. personal communication.

van Oosterom, P, 1995. The gap-tree, an approach to 'on-the-fly' map generalization of an area partitioning. In J.-C Müller, J.-P Lagrange, and R Weibel (Eds.), *GIS and Generalization - Methodology and Practice*, Number 1 in GISDATA. Taylor & Francis, London.

Walter, V and D Fritsch, 1999. Matching spatial data sets: a statistical approach. *International Journal of Geographical Information Science* 13(5), 445–473.