AUTOMATIC DERIVATION OF LAND-USE FROM TOPOGRAPHIC DATA

Frank Thiemann [a], Monika Sester [a], Joachim Bobrich [b]
[a] Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Appelstraße 9a, 30167 Hannover, Germany {frank.thiemann, monika.sester}@ikg.uni-hannover.de
[b] Federal Agency for Cartography and Geodesy, Richard-Strauss-Allee 11, 60598 Frankfurt am Main, Germany, joachim.bobrich@bkg.bund.de

KEY WORDS: Generalization, Aggregation, Processing, Large Vector Data, CORINE Land Cover

ABSTRACT:
The paper presents an approach for the reclassification and generalization of land-use information from topographic information. Based on a given transformation matrix describing the transition from topographic data to land-use data, a semantic and geometry based generalization of too small features for the target scale is performed. The challenges of the problem are as follows: (1) identification and reclassification of heterogeneous feature classes by local interpretation, (2) presence of concave, narrow or very elongated features, (3) processing of very large data sets. The approach is composed of several steps consisting of aggregation, feature partitioning, identification of mixed feature classes and simplification of feature outlines.
The workflow will be presented with examples for generating CORINE Land Cover (CLC) features from German Authoritative Topographic Cartographic Information System (ATKIS) data for the whole are of Germany. The results will be discussed in detail, including runtimes as well as dependency of the result on the parameter setting.

1. INTRODUCTION

1.1 Project Background

The European Environment Agency collects the Coordinated Information on the European Environment (CORINE) Land Cover (CLC) data set to monitor the land-use changes in the European Union. The member nations have to deliver this data every few years. Traditionally this data set was derived from remote sensing data. However, the classification of land-use from satellite images in shorter time intervals becomes more cost intensive.

Therefore in Germany the federal mapping agency (BKG) investigates an approach of deriving the land cover data from topographic information. The BKG collects the digital topographic landscape models (ATKIS Base DLM) from all federal states. The topographic base data contains up-to-date land-use information. But there are some differences between ATKIS and CLC.

1.2 CORINE Land Cover (CLC)

CORINE Land Cover is a polygon data set in the form of a tessellation: polygons do not overlap and cover the whole area without gaps. The scale is 1:100000. Each polygon has a minimum area of 25 hectare. There are no adjacent polygons with the same land-use class as these have to be merged.
Land cover is classified hierarchically into 46 classes in three levels, for which a three digit numerical code is used. The first and second level groups are:

1. artificial (urban, industrial, mine)
2. agricultural (arable, permanent, pasture, heterogeneous)
3. forest and semi-natural (forest, shrub, open)
4. wetland (inland, coastal)
5. water (inland, marine)

CLC has a detailed thematic granularity concerning vegetation objects. In the agricultural group, there are also some aggregated classes for heterogeneous agricultural land-use. Such areas are composed of small areas of different agricultural land-use, e.g. class 242 which is composed of alternating agricultural uses (classes 2xx).

1.3 ATKIS Base DLM

The *Base Digital Landscape Model* (DLM) of the *Authoritative Topographic Cartographic Information System* (ATKIS) is Germany's large scale topographic landscape model. It contains polygon and also poly-line and point data. The scale is approx. 1:10000. The minimum area for polygons is one hectare. The data set is organized in thematic layers, which can also overlap. The land cover information is spread among these different layers.

Each object has a four digit class code[1] and different attributes consisting of a three character key and a four digit alphanumeric attribute. The classes are also organized hierarchically in three levels. The seven first levels groups are:

1. presentation
2. residential
3. traffic (street, railway, airway, waterway)
4. vegetation
5. water
6. relief
7. areas (administrative, geographic, protective, danger)

Table 1 gives a summarized comparison of the two data sets.

Data set	ATKIS Base DLM	CORINE LC
scale	1:10000	1:100000
source	aerial images, cadastre	satellite images
min. area size	1 ha	25 ha
topology	overlaps, gaps e.g. between the divided carriageways	tessellation
feature classes	90 relevant (155 with attributes)	44 (37 in Germany)
agricultural feature classes	5 relevant (9 with attributes)	11 (6 in Germany) 4 (2) heterogeneous classes

Table 1: Comparison of ATKIS and CLC

[1] In the new AAA model, which is currently being introduced, there is a 5-digit object code.

1.4 Automatic derivation of CLC from DLM

The aim of the project is the automated derivation of CLC data from ATKIS. This derivation can be considered as generalization process, as there it requires both thematic selection and reclassification, and geometric operations due to the reduction in scale. Therefore, the whole workflow consists of two main parts. The first part is a model transformation and consists of the extraction, reclassification and topological correction of the data. The second part, the generalization part, which will be described in more detail in this paper, is the aggregation and simplification for the smaller scale.

The first part consists of the following steps: after the extraction of the relevant features from the DLM the topological problems like overlaps and gaps area solved automatically using appropriate algorithms. The reclassification is done using a translation table which takes the ATKIS classes and their attributes into account. In the cases where a unique translation is not possible, a semi-automatic classification from remote sensing data is used. The derived model is called DLM-DE LC.

In the second part the high level information from the DLM-DE is generalized to the small scale of 1:100000 of the CLC. For that purpose a sequence of generalization operations is used. The operators are dissolve, aggregate, split, simplify and a mixed-class filter.

1.5 Main Challenges

One of the main challenges of the project is the huge amount of data. The DLM-DE contains ten million polygons. Each polygon consists in average of thirty points, so one has to deal with 300 million points, which is more than a standard PC can store in the main memory. Therefore a partitioning concept is needed that allows processing the data sequentially or in parallel. Fast algorithms and efficient data structures reduce the required time.

Another challenge is the aggregation of agricultural heterogeneous used areas to a group of 24x-classes in the case that a special mixture of land-uses occurs. The difficulty is to separate these areas from homogeneous as well as from other heterogeneous classes.

2. RELATED WORK

CORINE Land Cover (Büttner et al. 2006) is being derived by the European States (Geoff et al. 2007). The Federal Agency of Cartography and Geodesy attempts to link the topographic data base with the land-use data. To this end, transformation rules between CLC and ATKIS have been established (Arnold 2009). As described above, the approach uses different generalization and interpretation steps. The current state of the art in generalization is described in Mackaness et al. (2007). The major generalization step needed for the generalization of land-use classes is aggregation. The classical approach for area aggregation was given by Oosterom (1995), the so-called GAP-tree (Generalized Area Partitioning). In a region-growing fashion areas that are too small are merged with neighboring areas until they satisfy the size constraint. The selection of the neighbor to merge with depends on different criteria, mainly geometric and semantic constraints, e.g. similarity of object classes or length of common boundary. This approach is implemented in different software solutions (e.g. Podrenek, 2002). Although the method yields areas of required minimum size, there are some drawbacks: a local determination of the most compatible object class can lead to a high amount of class changes in the whole data set. Also, objects can only survive the generalization process, if they have compatible neighbors. The method by Haunert (2008) is able to overcome these drawbacks. He is also able to introduce additional constraints e.g. that the form of the resulting objects should be compact. The solution of the problem has been achieved using an exact approach based on mixed-integer programming (Gomory, 1958), as well as a heuristic approach using simulated annealing (Kirkpatrick 1983). However, the computational effort for this global optimization approach is very high.

Collapse of polygon features corresponds to the skeleton operation, which can be realized using different ways. A simple method is based on triangulation; another is medial axis or straight skeleton (Haunert & Sester, 2008).

The identification of mixed classes is an interpretation problem. Whereas interpretation is predominant in image understanding where the task is to extract meaningful objects from a collection of pixels (Lillesand & Kiefer, 1999), also in GIS-data interpretation is needed, even when the geo-data are already interpreted. E.g. in our case although the polygons are semantically annotated with land-use classes, however, we are looking for a higher level structure in the data which evolves from a spatial arrangement of polygons. Interpretation can be achieved using pattern recognition and model based approaches (Heinzle & Anders, 2007).

3. APPROACH

3.1 Data and index structures

Efficient algorithms demand for efficient data and search structures. For topology depending operations a topologic data structure is essential. For spatial searching a spatial index structure is needed; furthermore, also structures for one-dimensional indexing are used.

In the project the we use a extended Doubly Connected Edge List (DCEL) as topologic structure and grids (two-dimensional hashing) as spatial index.

3.1.1 Extended DCEL

The doubly connected edge list (DCEL) is a data structure for polygonal meshes. It is a kind of boundary representation. The topological elements (and their geometric correspondence) are faces (polygon), edges (lines) and nodes (points). All topologic relations (adjacencies and incidences) are expressed by explicit links (see Figure 1).. For efficient iteration over all nodes or edges of a face or all incident edges of a node the edges are split into a pair of two directed half-edges. Each half-edge links its origin (starting point), its twin, the previous and next half-edge and the incident face. The node contains the geometric information and a link to one of the incident half-edges. The face contains a link to a half-edge from the outer loop and if the polygon has holes also, a half-edge from each inner loop respectively.

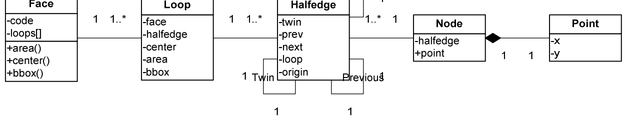

Figure 1: UML Diagram of the extended DCEL

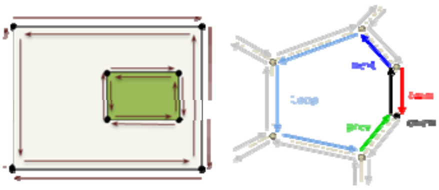

Figure 2: Left: A Face with an inner face in DCEL Right: Topological relations of a half-edge

For reasons of object orientated modeling loops were placed between the faces and half-edges, as one can often find in 3D data-structures (e.g. ACIS). The loop represents a closed ring of half-edges. This ring can be an inner or outer border of a face (see Figure 2). Algorithms for calculation the area (the area of inner loops is negative) or the centroid are implemented as member function of the loop. Because of the linear time complexity the values will be stored for each loop. For efficient spatial operations also the bounding box of the loop is stored. The land-use code is attached to faces.

3.1.2 GRID

As spatial index for nodes, edges and faces we use a simple two dimensional hashing. We put a regular grid over the whole area. Each cell of this grid contains a list of all included points and all intersecting edges and faces, respectively. This simple structure can be used, because of the approximately equally distributed geometric features.

For the DLM-DE a grid width of 100 m for points and edges (<10 features per cell) and 1000 m for faces (40 faces per cell) leads to nearly optimal speed. Experiments with a KD-tree for the points lead to similar results.

3.2 Topological cleaning

Before starting the generalization process, the data have to be imported into the topological structure. In this step we also look for topological or semantic errors. Each polygon is check for a valid CLC class. Small sliver polygons with a size under a threshold of e.g. 1 m will be rejected. A snapping with a distance of 1 cm is done for each inserted point. With a point in polygon test and a test for segment intersection overlapping polygons are detected and also rejected. Holes in the tessellation can be easily found by building loops of the half-edges which not belong to any face. Loops with a positive orientation are holes in the data set. The largest loop with a negative orientation is the outer border of the loaded data.

3.3 Generalization operators

3.3.1 Dissolve

The dissolve operator merges adjacent faces of the same class. For this purpose the edges which separate such faces will be removed and new loops are built. Besides the obvious cases which reduce the number of loops, there are also cases which generate new inner loops (see Figure 3).

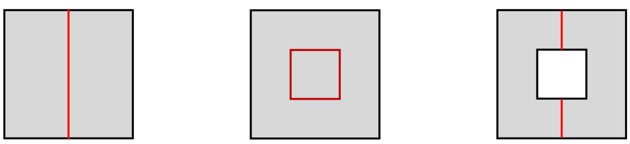

Figure 3: Beside the obvious cases (left and middle) of a merge, where the number of loops is reduced there are also cases which produce new inner loops (right).

3.3.2 Aggregate

The aggregation step aims at guarantying the minimum size of all faces. The aggregation operator in our case uses a simple greedy algorithm. It starts with the smallest face and merges it to a compatible neighbor. This fast algorithm is able to process the data set sequentially. However, in some cases it may lead to unexpected results, as shown in Figure 4. This is due to the fact that the decisions are only taken locally and not globally.

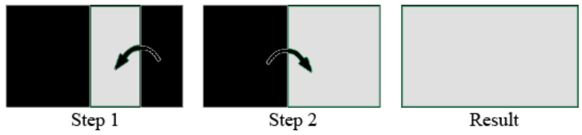

Figure 4: The sequential aggregation can lead to an unexpected result: The black area is dominating the source data set, but after aggregation the result is grey (according to Haunert (2008))

There are different options to determine compatible neighbors. The criterion can be:

- the semantic compatibility (semantic distance),
- the geometric compactness
- or a combination of both.

CLC	111	112	121	122	123	124	131	132	133	141	142	211	212	213	221	222	223	231	241	242	243	244
111	0	1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	2	2	3	4
112	1	0	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	2	2	3	4
121	3	3	0	1	1	1	2	2	2	4	4	6	6	6	6	6	6	6	5	5	6	7
122	2	2	1	0	1	1	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4
123	3	3	1	1	0	1	2	2	2	4	4	5	5	5	5	5	5	5	5	5	5	5
124	3	3	1	1	1	0	4	4	4	2	2	6	6	6	6	6	6	5	6	5	6	6
131	3	3	2	2	3	3	0	1	1	4	4	7	7	7	7	7	7	7	7	7	7	7
132	3	3	2	2	3	3	3	0	1	4	4	7	7	7	7	7	7	7	7	7	7	7
133	1	1	1	1	1	1	2	2	0	2	2	3	3	3	3	3	3	3	3	3	3	3
141	3	2	3	3	3	3	3	3	3	0	1	7	7	7	7	7	7	7	7	7	5	5
142	3	2	3	3	3	3	3	3	3	1	0	5	5	5	5	5	5	5	5	5	5	5
211	5	5	5	5	5	5	5	5	5	5	5	0	1	1	4	4	4	3	2	2	2	2

Figure 5: Small extract of the CLC priority matrix

The **semantically** nearest partner can be found using a priority matrix. We use the matrix from the CLC technical guide (Bossard, Feranec & Otahel 2000) (Figure 5). The priority values are from an ordinal scale, so their differences and their values in different lines should not be compared. The matrix is not symmetric, as there may be different ranks when going from one object to another than vice versa (e.g. settlement -> vegetation). Priority value zero is used if both faces have the same class. The higher the priority value, the higher is the semantic distance. Therefore the neighbor with the lowest priority value is chosen.

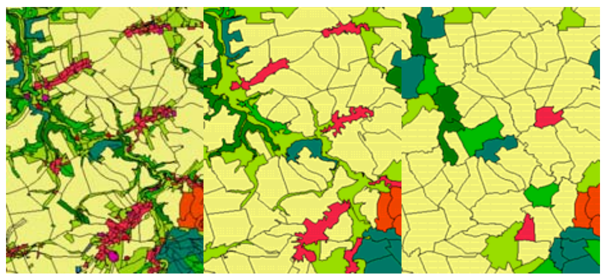

Figure 6: (left to right) Original situation, the result of the semantic and geometric aggregation.

As **geometric** criterion the length of the common edge is used. This leads to compact forms. Compactness can be measured as the ratio of area and perimeter. A shorter perimeter leads to better compactness. So the maximum edge length has to be reduced to achieve a better compactness.

The effects of using the criteria separate are shown in a real example in Figure 6. The semantic criterion leads to non-compact forms, whereas the geometric criterion is more compact but leads to a large amount of class change.

The combination of both criteria allows merging of semantically more distant objects, if the resulting form is more compact. This leads to Formula 1.

$$distance(A,B) = \frac{b^{priority}}{length} \quad (1)$$

The formula means that a b-times longer shared edge allows a neighbor with the next worse priority. The base b allows to weight between compactness and semantic proximity. A value of b=1 leads to only compact results, a high value of b leads to semantically optimal results. Using the priority values is not quite correct; it is only a simple approximation for the semantic distance.

Another application of the aggregation operation is a special kind of dissolve that stops at defined area size. It merges small faces of the same class to bigger compact faces using the geometric aggregation with the condition that only adjacent faces of the same class are considered.

3.3.3 Split

In addition to the criterion of minimal area size also the extent of the polygon is limited to a minimum distance. That demands for a collapse operator to remove slim, elongated polygons and narrow parts. The collapse algorithm by Haunert & Sester (2008) requires buffer and skeleton operations that are time consuming. Therefore - as faster alternative - a combination of splitting such polygons and merging the resulting parts with a geometric aggregation to other neighbors is used.

Figure 7: The operator splits the polygon at narrow parts if there is a higher order node or a concave node. An existing node is preferred if it is close to the orthogonal projection.

The split operator cuts faces at narrow internal parts. First, the concave or higher order node with the smallest distance to a non-adjacent edge is calculated. A new node will be inserted at the orthogonal projection if there is no existing node nearby. An edge is inserted if it fulfills the conditions being inside and not intersecting other edges. Else the next suited node is chosen (see Figure 7). After the split operator the aggregate operator merges too small pieces to other adjacent faces.

3.3.4 24x-Filter

In CORINE land-cover there is a group of classes which stands for heterogeneous land-uses. The classes 242 and 243 are relevant for Germany. Class 242 (complex cultivation pattern) is used for a mixture of small parcels with different cultures. Class 243 is used for land that is principally occupied by agriculture, with significant areas of natural vegetation.

Heterogeneous classes are not included in the ATKIS schema. To form these 24x-classes an operator for detecting heterogeneous land-use is needed. The properties of these classes are that smaller areas with different, mostly agricultural land-use alternate within the minimum area size (actually 25 ha in CLC). For the recognition of class 242 only the agriculture areas (2xx) are relevant. For 243 also forest, semi- and natural areas (3xx, 4xx) and lakes (512) have to be taken into account.
The algorithm calculates some neighborhood statistics for each face. All adjacent faces within a distance of the centroid smaller than a given radius and with an area size smaller than the target size are collected by a deep search in the topological structure. The fraction of the area of the majority class and the summarized fractions of agricultural areas (2xx) and (semi-)natural areas (3xx, 4xx, 512) are calculated. In the case the majority class dominates (>75%) then the majority class becomes the new class of the polygon. Otherwise there is a check, if it is a heterogeneous area or only a border region of larger homogeneous areas.

For that purpose the length of the borders between the relevant classes is summarized and weighted with the considered area. A heterogeneous area is characterized by a high border length, as there is a high number of alternating areas. To distinguish between 242 and 243 the percentage of (semi-natural) areas has to be significant (>25%).

3.3.5 Simplify

The simplify-operator removes redundant points from the loops. A point is redundant, if the geometric without using this point is lower than an epsilon and if the topology do not change.

We implemented the algorithm of Douglas & Peucker (1973) with an extension for closed loops and a topology check. The algorithm is running over all loops, between each pair of adjacent topological nodes (degree > 2). If the loop contains no topological nodes, the first one is chosen. The algorithm tries like Douglas-Peuker to use the direct line between the two end nodes and searches for the farthest point of the original line to this new line. The first extension is for the case, that both end points are the same nodes. Then the point to point distance is used instead of point to straight line distance. If the distance of the farthest point is larger than the epsilon-value then the point is inserted in the new line and the algorithm processes both parts recursively. If the distance is smaller than epsilon the Douglas-Peucker algorithm would remove all points between the end nodes. Here the second extension is done to checks the topology. All points in the bounding-box spanned by the two nodes are checked for switching the side of the line. If a point switches the side, the farthest point is inserted to the line (i.e. treating it as if it were too far).

3.4 Process chain

In this section the use of the introduced operators and their orchestration in the process change is shown. The workflow for a target size of 25 ha is as follows:

1. import and data cleaning
2. fill holes
3. dissolve faces < 25 ha
4. split faces < 50 m
5. aggregate faces < 1 ha geometrically (base 1.2)
6. reclassify faces with 24x-filter (radius 282 m)

7. aggregate faces < 5 ha weighted (base 2)
8. aggregate faces < 25 ha semantically
9. simplify polygons (tolerance 20 m)
10. dissolve all

During the import step (1) semantic and topology is checked. Small topologic errors are resolved by a snapping. The hole-fill step (2), searches for all outer loops and fills gaps with dummy objects. These objects will be merged to other objects in the later steps.

A first dissolve step (3) merges all faces with an adjacent face of the same CLC class which are smaller than the target size (25 ha). The dissolve is limited to 25 ha to prevent polygons from being too large (e.g. rivers that may extend over the whole data set). This step leads to many very non-compact polygons. To be able to remove them later, the following split-step (4) cuts them at narrow internal parts (smaller than 50 m = 0.5 mm in the map). Afterwards an aggregation (5) merges all faces smaller than the source area size of 1 ha to geometrically fitting neighbors.

The proximity analysis of the 24x-filter step (6) reclassifies agricultural or natural polygons smaller than 25 ha in a given surrounding as heterogeneous (24x class).
The next step aggregates all polygons to the target size of 25 ha. First we start with a geometric/semantically weighted aggregation (7) to get more compact forms, second only the semantic criterion is used (8) to prevent large semantic changes of large areas.

The simplify step (9) smoothes the polygon outlines by reducing the number of nodes. As geometric error tolerance 20 m (0.2 mm in the map) is used. The finishing dissolve step (10) removes all remaining edges between faces of same class.

4. RESULTS

4.1 Runtime and memory

The implemented algorithms are fast but require a lot of memory. Data and index structures need up to 160 Bytes per point on a 32 bit machine. With 6 GB free main memory on a 64 bit computer we were able to process up to 30 million points at once, which corresponds to the tenth part of Germany.

The run-time was tested with a 32 bit 2.66 GHz Intel Core 2 processor with a balanced system of RAM, hard disk and processor (windows performance index 5.5). The whole generalization sequence for a 45 x 45 km data set takes less than two minutes. The most time expensive parts of the process are the I/O-operations which take more than 75% of the computing time. We are able to read 100000 points per second from shape files while building the topology. The time of the writing process depends on the disk cache. In the worst case it is the same as for reading.

The time of the operations highly depends on the data. The most expensive one is the split operation that is quadratic with the number of points per polygon. At the introduced position in the process chain the split operation takes the same time as the reading process.

The other operations are ten and more times faster than I/O operations. The aggregation operator processes one million points per second. The line generalization with 0.7 million points per second is a bit slower, but it works on the reduced data set at the end of the generalization process.

4.2 Semantic and geometric correctness

To evaluate the semantic and geometric correctness we did some statistics comparing input, result and a CLC 2006 reference data set, which was derived from remote sensing data.

Data set	DLM-DE	Result	CLC 2006
Polygons	91324	1341	878
Points per Polygon	24	104	77
Area per Polygon	2.3 ha	155 ha	238 ha
Perimeter per Polygon	0.6 km	9.4 km	10.1 km
Avg. Compactness	50%	24%	33%

Table 2: Statistic of the test data set Dresden (45 x 45 km)

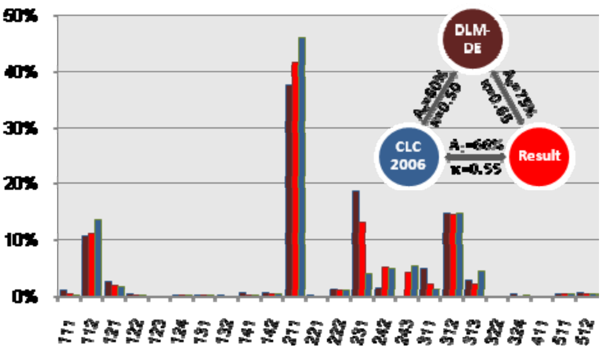

Figure 8: Percentage of area for each CLC class (bars) and percentage of match (A_0) and κ-values for the Dresden data set.

Figure 9 shows the input data (DLM-DE), our result and the CLC 2006 of the test area Dresden. The statistics in Figure 8 verifies that our result matches with DLM-DE (75%) better than the reference data set (60%). This is not surprising as for CLC 2006 different data sources were used. Because of the removing of the small faces our generalization result is a bit more similar to CLC 2006 (66%) than CLC 2006 to our input.

Table 2 shows, that our polygons are only a bit smaller and more complex and less compact than the CLC 2006 polygons. The percentage of the CLC classes is similar in all data sets (Figure 9). There are some significant differences between the DLM-DE and CLC 2006 within the classes 211/234 (arable/grass land) and also between 311/313 (broad-leaved/mixed forest) and 111/112 (continuous/discontinuous urban fabric). We assume that it comes from different interpretations. The percentages in our generated data set are mostly in the middle. The heterogeneous classes 242 and 243 are not included in the input data. Our generalization generates a similar fraction of these classes. However, the automatically generated areas are mostly not at the same location as in the manually generated reference data set. We argue though, that this is the result of an interpretation process, where different human interpreters would also yield slightly different results.

Input (DLM-DE) and the result match with 75%. This means that 25% of the area changes its class during generalization process. This is not an error; it is an unavoidable effect of the generalization. The κ-values 0.5-0.65 which stand for a moderate up to substantial agreement should also not be interpreted as bad results, because it is not a comparison with

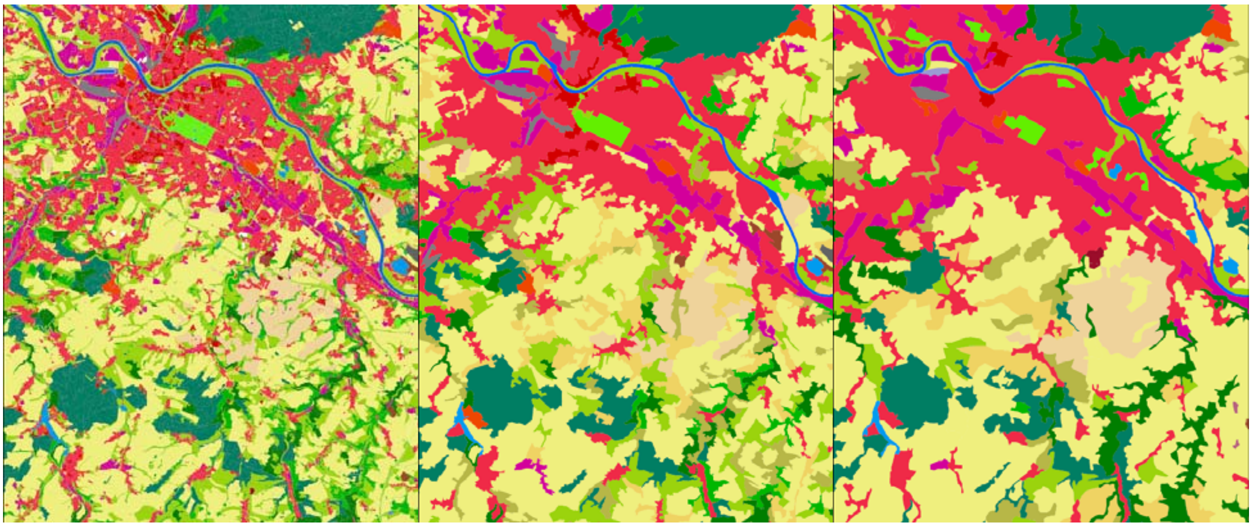

Figure 9: Extract (20 x 25 km) of test data set Dresden from left to right: input DLM-DE, our result and CLC 2006 as reference.

the real truth, or with a defined valid generalization, respectively.

5. OUTLOOK ON FUTURE WORK

With non-high end PC's it is not possible to process the whole of Germany at once. Theoretically it is possible to process all data sequentially. But most operators need a spatial environment for each polygon. Reading polygons and environments object by object leads to a high I/O-traffic, which is the bottleneck of the algorithms. This communication time would even get worse in a database implementation. Therefore we currently try some partitioning concepts which allow working on bigger areas than single polygons. The partitioning may also allow for a parallel processing of the data. However, the borders of the partitions should have only a very small effect on the generalization result.

Because of the aggregation algorithm there may be a chaotic effect - small changes causes a big changes of the result. We want to study and quantify these effects. Also the problem of the heterogeneous classes should be studied to find out, if it is possible to get more certain results.

Our next project aim is to derive land cover changes from the historic versions of CLC. The problem is to divide pseudo from the real changes. In account, that it is not possible to get real 5 ha changes from a 25 ha data set; we think about getting and aggregating the changes from the high resolution DLM-DE.

6. REFERENCES

Arnold, S., 2009. Digital Landscape Model DLM-DE – Deriving Land Cover Information by Intergration of Topographic Reference Data with Remote Sensing Data. *Proceedings of the ISPRS Workshop on High-Resolution Earth Imaging for Geospatial Information,* Hannover.

Bossard, M., Feranec, J. & Otahel, J., 2000. EEA CORINE Land Cover Technical Guide – Addendum 2000. – Technical Report No. 40, Kopenhagen.

Büttner, G., Feranec, G. & Jaffrain, G., 2006. EEA CORINE Land Cover Nomenclature Illustrated Guide – Addendum 2006. – European Environment Agency.

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 (1973) 112-122.

Geoff, B. et al. 2007. UK Land Cover Map Production Through the Generalisation of OS MasterMap®. *The Cartographic Journal*, 44 (3). 276-283.

Gomory, R., 1958. Outline of an algorithm for integer solutions to linear programms, *Bulletin of the American Mathematical Society*, 64(5), 274-278.

Haunert, J.-H., 2008. Aggregation in Map Generalization by Combinatorial Optimization, Vol. Heft 626 of *Reihe C*, Deutschen Geodätische Kommission, München.

Haunert, J.-H. & Sester, M., 2008. Area collapse and road centerlines based on straight skeletons, *GeoInformatica*, vol. 12, no. 2, p. 169-191, 2008.

Heinzle, F. & Anders, K.-H., 2007. Characterising Space via Pattern Recognition Techniques: Identifying Patterns in Road Networks, in: W. Mackaness, A. Ruas & L.T. Sarjakoski, eds, *Generalization of geographic information: cartographic modelling and applications*, Elsevier, Oxford, pp. 233-253.

Kirkpatrick, S., Gelatt, C. D. Jr., & Vecchi, M. P., 1983. Optimization by Simulated Annealing. In: *Science 220* (4598), 671. 13 May 1983.

Lillesand, T. M. & Kiefer, R. W., 1999. *Remote Sensing and Image Interpretation*, 4th edn, John Wiley &\& Sons.

Mackaness, W., A., Ruas, A. & Sarjakoski, L.T., 2007. *Generalisation of Geographic Information - Cartographic Modelling and Applications*, Elsevier Applied Science.

Pondrenk. M, 2002. Aufbau des DLM50 aus dem Basis-DLM und Ableitung der DTK50 – Lösungsansatz in Niedersachsen. In: *Kartographische Schriften, Band 6, Kartographie als Baustein moderner Kommunikation*, S.126-130, Bonn.

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 & R. Weibel, eds, *GIS and Generalization - Methodology and Practice*, Taylor & Francis, pp. 120-132.