GAP-Tree Extensions Based on Skeletons

Tinghua Ai[1] and Peter van Oosterom[2]

Department of Cartography and GIS, Wuhan University, P. R. China,
aith@wuhan.cngb.com[1]
Section GIS technology, Department of Geodesy, Delft University of Technology,
the Netherlands, oosterom@geo.tudelft.nl[2]

Abstract

The Generalized Area Partitioning tree (GAP-tree) is a model that supports on-the-fly generalisation of planar partitions of polygon objects. This paper focuses on solving some limitations of the existing GAP-tree and explores two extensions. The first extension enables the handling of a disjoint polygonal cluster (within the partition) and allows aggregation operations to be performed. The skeleton partitioning model, which is based on the constrained Delaunay triangulation for the polygonal cluster, is used to define the bridge areas between aggregated polygons. The second extension involves the improvement of removing insignificant objects by separating them into parts around the adjusted skeleton and assigning these parts to different neighbours. The adjusted skeleton is defined by the compatibility between the removed object and its neighbours, which considers not only topological relationships but also importance and semantic similarity. This process again uses the Delaunay triangulation. The algorithm is given to construct the extended GAP-tree.
Keywords: on-the-fly generalisation, spatial partitioning, and neighbourhood analysis, GAP-tree

1 Introduction

Multi-scale representation of spatial data is the intent of seamless data navigation, progressive web transfer, self-adaptable visualisation and other applications. Recently it has gained considerable significance in such fields as map generalisation and on-demand mapping. To obtain geographic data representation suitable for multiple resolutions, one has to build hierarchical spatial partitioning and clustering. The Generalized Area Partitioning tree (abbreviated as GAP-tree) developed by Van Oosterom (1991, 1994, 1995) is a spatial data model for representing multiple detail levels of area objects within a planar partition, such as

topography, land use and soil type. It supports display of increased detail during continuous zoom-in rather than just exaggeration of graphic symbols. In the GAP-tree structure, a node in the tree represents a generalised polygon region, which may contain unimportant lower level polygons, and the trace from leaves to root corresponds to polygon selection and simplification in map generalisation. The GAP-tree construction is based on answers to two questions: "Which object is the least important one?" and "Which is the most compatible neighbour of the unimportant polygon?" The first question is resolved based on the object classification and on its attribute and geometric properties such as size. The latter is based on the computation of neighbours and semantic similarity. The neighbourhood in the original GAP-tree does not consider disjoint polygons having an approximate neighbour relationship (with close distance). The original GAP-tree construction work is applicable when regions are connected to each other, which means that the polygon distribution has neither overlaps nor hole areas (or empty space).

This paper extends the GAP-tree data structure for on-the-fly generalisation (continuous zoom-in/out after pre-processing the data) in two ways. First, the neighbourhood is extended to include also polygons at a certain distance. This is used when an unimportant object has to be removed and the neighbour with the highest compatibility value (most compatible neighbour) has to be found. The techniques applied will be spatial searching (using a spatial index) with a buffer surrounding the object and using interpolated constrained Delaunay triangulation (Preparata and Shamos, 1985) to compute distance measures between the two objects. Furthermore, the constrained Delaunay triangulation is used to compute a polygon between the two disjoint neighbours, which will be called the 'neighbour region' of the two objects. This neighbour region is influenced by the importance and the (in)compatibility of the object types belonging to the triangles, which might form the neighbour region. The unimportant object will be merged into the most compatible neighbour and this will include the neighbour region between the two objects.

Second, a different extension has been considered and instead of assigning an unimportant object completely to the neighbour with the highest compatibility value, the object is subdivided in parts and neighbours get assigned pieces based on their compatibility value. This is achieved through an adjusted skeleton of the object that has to be removed. The higher the compatibility value of a neighbour, the more it pushes away the skeleton (and the larger piece of the object will be assigned to this neighbour). An algorithm will be described, that will not produce the 'equal' parts skeleton lines, but rather will give a compatibility value based on weighting. These two developments are an addition to the original GAP-tree and are improvements for features such as (parallel lines) described in earlier papers (Van Putten and Van Oosterom, 1998).

The remaining part of the paper is arranged as follows. In section 2 we present the refined neighbourhood analysis. The aggregation of 'disjoint polygonal cluster' is presented in section 3. Section 4 discusses the extension of another operation amalgamation based on the adjusted skeleton. Section 5 gives the progressive refinement of the extended GAP-tree construction and provides the

example application in land use data generalisation. Finally, some future work is outlined in the conclusion, section 6.

2 Neighbourhood Analysis

The GAP-tree is based on neighbourhood analysis. In spatial cognition, neighbourhood is a general concept influenced by topological, geometric, semantic, and Gestalt nature. If two objects touch and share a common boundary, they are neighbours to each other in an adjoining relationship. Two objects with disjoint topological relationship, but in close proximity, can also be regarded as having an approximate neighbourhood relationship. In this case, the geometric distance plays a major role. In a spatial cluster, in which entities have similar distance to each other, the (near) equal attribute values and the similar distribution properties in Gestalt nature consider the corresponding set of objects to be in the neighbourhood of each other. In the natural world, the distribution of geographic phenomena generally has an uncertainty boundary rather than a sharp one. An example would be soil types or vegetation classes in which there is usually some form of gradation in the real world phenomena as opposed to a finite demarcation often depicted in map form. Although in data representation, we can draw a sharp boundary to separate different categories of phenomena, in relationship judgments we should not rely too much on it. In the original GAP-tree, the neighbourhood judgement is simply based on a topological touch relationship.

Each object in a GIS has a semantic description (at least a classification or feature category) and a spatial description. Selecting all objects from a single feature category results in a sub-set of objects within the planar partition. One polygon object, on the one hand, has adjoining neighbours with objects possibly from other categories. On the other hand, it also have approximate neighbours with the same classification. As far as the adjacency degree is concerned, the former is spatially close but may be semantically far, the latter is semantically close but spatially far. What is the most compatible object in the neighbourhood? In earlier research (Van Oosterom 1991, 1994), the compatibility formula has been introduced to resolve this question, but only for adjoining neighbours. The impact of distance should also be included in the compatibility formula to allow approximate neighbours to participate.

Approximate neighbourhood representation assigns those objects with distance to each other less than the tolerance (buffer zone) into a set called approximate neighbourhood set. Objects with touch relationship can be thought of as special cases in which objects have zero distance to each other. In this sense, the adjoining and the approximate neighbourhood can be unified. For the semantic contribution in neighbourhood identification, we define the following principle: within a neighbourhood cluster, the closer the semantic relationship between object elements, the closer the neighbourhood relationship to each other is. In Fig. 1, an example of a neighbourhood cluster representation is given: object *a* has neighbours *d, c, f, b, h*. When comparing two candidate objects *b* and *f*, we find

that object *b* has the same category as object *a,* so it has closer neighbourhood than object *f* to object *a,* although object *f* touches object *a*. This is just an intuitive judgement and the strict judgement depends on compatibility computation.
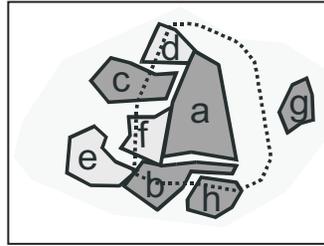

Fig. 1. Neighbourhood cluster of polygon a.

In neighbourhood definition for disjoint objects, another consideration is the context influence (Hernandez and Clementini, 1995). For a large neighbourhood tolerance distance, two objects having less distance are regarded as strong neighbours. However, when the third object appears between them, it is considered that they no longer have a neighbourhood relationship. This implies we cannot simply use the distance metric to find a neighbour, such through application of a buffer operation in commercial GIS software. An additional test has to be designed. The Voronoi diagram of the polygonal cluster is computed and each polygon is extended with its influence region. The result is a partition of space based on the objects with a certain classification. The approximate neighbourhood is based on the topological touch relationship between the partitioning polygons in the Voronoi diagram.

Based on the above analysis, objects are referred to as having neighbourhood relationship if they satisfy one of the following conditions:

1. having topological touch relationship to each other;
2. among objects with a similar classification, their Voronoi cell polygon shares a boundary and the distance between objects is less than some tolerance.

Furthermore, the semantic adjacency and the length of a shared boundary decide the neighbourhood strength. The impact of these parameters has been used in the computation of the most compatible neighbours in the original GAP-tree construction. With the new neighbourhood definition, the shared boundary includes not only boundaries between objects themselves, but also between the touching partitioning polygons in the Voronoi diagram in case of disjoint objects.

According to the definition, in Fig. 1, object *a* has neighbours *b, c, f, d* and this order also represents the neighbourhood strength. Object *g* has the same category but is far away from object *a*, so it cannot act as the neighbor of *a*. Although object *h* has the same category as object *a* and is located within the buffer region of object *a*, the appearance of object *b* makes their corresponding Voronoi diagram partitioning polygons not touch.   Therefore *h* is not the neighbour of object *a*.

3 Aggregation Based on the Skeleton Partitioning Model

In spatial data handling, we can distinguish two combination operations: the **aggregation** for polygon objects with homogeneous semantics, and the **amalgamation** for polygon objects with heterogeneous semantics. The results of aggregation will inherit the attribute of the original ones,, while the result of amalgamation will change the classification of some of the original objects. We will discuss the amalgamation operation in section 4, while in this section will focus on aggregation.

Aggregation is an important operation in the generalisation of data sets such as land use, urban building clusters among other applications. The original GAP-tree focuses on the selection and combining of objects. However, no distinction is made between amalgamation and aggregation. In the GAP-tree structure, the merge operation just happens based on the compatibility with the least important object and the result is a 'parent and child' edge in the GAP-tree. When considering the aggregation of non-adjoining neighbours, we find that the operation implies that a third object (with another classification) is affected.
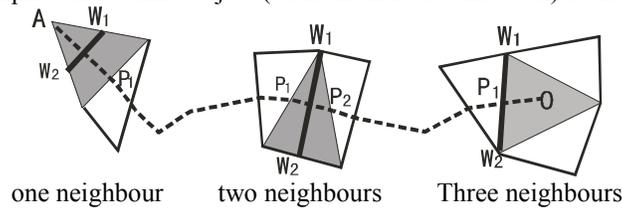

Fig. 2. W_1, W_2 local distance for three triangle types

For a set of objects, from a similar category, we construct the constrained Delaunay triangulation. The triangles connecting different polygons are considered which implies that those triangles within and outside the polygons but with three vertices located in the same polygon will be removed from the triangulation. For the remaining triangles, according to the number of neighbour triangles, the dashed skeleton segments are linked as depicted in Fig. 2 (where P_i is the middle point of the corresponding triangle edge, and O is the weight centre of the triangle). This is done to get the skeleton partitioning model. The result is illustrated in Fig. 3, in which every object polygon is surrounded by one partitioning polygon, and the partitioning skeleton separates the 'empty' space equally between two or more object polygons. If two partitioning polygons touch, the corresponding contained object polygons are possible neighbours, depending on the distance. Next we compute distance measures (minimum, average) between object polygons based on the skeleton edges.

The aggregation of approximate neighbour polygons behaves as filling a certain area between them and changing its classification (a little bit of amalgamation).

We call this area the *neighbour region*. Define a neighbourhood tolerance distance *d*, and then select those triangles whose local distance between the connected object polygons is shorter than *d*. Group these selected triangles into triangle clusters. Each triangle cluster goes across a part of the partitioning skeleton edges and plays the bridge role connecting two or more neighbour objects, see Fig. 3.
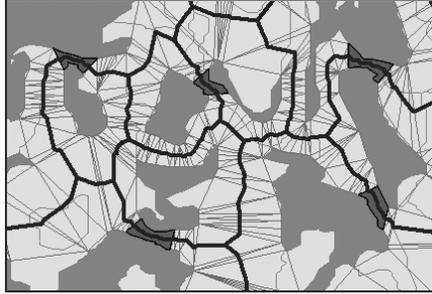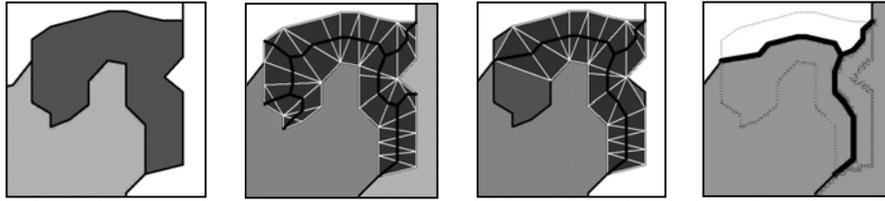

Fig. 3. One category of land use object distribution and *neighbour region* extraction

4 Amalgamation Based on the Adjusted Skeleton

In the iterative process of GAP-tree construction, each step will remove the most insignificant polygon object among the current remaining objects. Therefore, the most compatible neighbour will fill the space. This process can also be understood as the amalgamation operation in the case of two semantically different objects. In the original GAP-tree, this operation is based on topological structure and semantic compatibility computation, and the insignificant object will completely be assigned to one of the neighbours. We cannot always find an obviously strong neighbour to completely replace the insignificant object. From the point of view of error adjustment, separating the insignificant polygon around its skeleton and assigning different parts into different neighbours is better than assigning the whole polygon into one of the neighbours. Bader and Weibel (1997) presented an idea to separate an insignificant object based on the constrained Delaunay triangulation and the skeleton. However, if the insignificant object does not have an obvious lineal stretch distribution, the method cannot guarantee that the terminal point of the skeleton exactly meets one of the intersection points between two neighbours. Adjustment of the skeleton is required. Next we will provide the detailed method of this amalgamation operation from the geometric algorithm.

The basic idea of this method is to generate the skeleton within the insignificant polygon and then create the polygon parts composed of some skeleton edges and one boundary edge. The generated polygon parts are assigned to the neighbour polygon on the other side of the boundary edge. Here the Delaunay triangulation is used again, but this time within the studied polygon. To seamlessly partition the insignificant polygon, we expect that the terminal point of each skeleton edge meets in a node of the boundary edge. This condition includes two aspects: one is that each terminal point of the skeleton edge must exactly be located in one node

of boundary edge, another is that each node of the boundary edge must exactly have one linking skeleton edge. Note that a node is defined as the location where

at least three edges meet. Obviously, directly extracting the skeleton based on all triangles within one polygon cannot guarantee the above conditions, see Fig. 4b. We use the following two steps to adjust the skeleton edges.

<div align="center">(a)   (b)   (c)   (d)</div>

Fig. 4. The generation of skeleton edges within one insignificant polygon, **(a)** original scene, **(b)** original skeleton, **(c)** adjusted skeleton, **(d)** partitioning result

First, all triangles linking the same boundary edge (that is, with all three points on the same boundary of one neighbour polygon) are removed and will be assigned to the corresponding polygon, as shown in Fig. 4c. Based on the remaining set of triangles within the polygon, the new skeleton edge is constructed. We now can guarantee that each terminal point exactly meets in one node, and is not located in the middle of a boundary edge. As illustrated in Fig. 4c, from the point of view of neighbour, the area in which triangles are removed serves as the concave part. Since this part is strongly surrounded by one neighbour polygon, in the partitioning it should not act as a competition region for different neighbours to separate. Second, a skeleton edge begins from a triangle with only one neighbour triangle, which appears in some places of the boundary, which is not smooth.
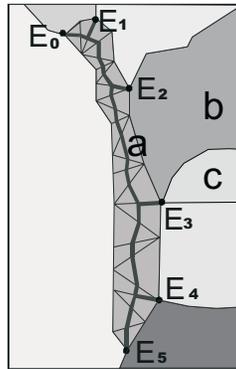

Fig. 5. Adding additional skeleton edges in E_1, E_2, E_3, E_4.

As described above, the skeleton begin point will always be a node. However, in some situations there is no skeleton edge connected to a node (a relative smooth

part), see E_1, E_2, E_3, and E_4 in Fig. 5. In Fig. 5, we can just get one skeleton edge from E_0 to E_5. From nodes without a skeleton edge, search the closest point in the existing skeleton edge along a triangle edge. Some nodes relate to just one triangle edge intersecting with the existing skeleton edge, such as E_1 and E_3. Others may relate to more than one triangle edge, such as E_2 and E_4. Select the shortest connection (half of one triangle edge) and let it act as the added 'skeleton' edge. Note that, if a neighbour wants to receive a part of an insignificant polygon, it must share a common boundary with some length. For example, in Fig. 5, object c shares just one point E_3 with the insignificant object a, it will not merge with the part of object a.

Through these processes for matching boundary nodes and skeleton terminal points, the conditions discussed earlier are fulfilled. Now, select all the boundary edges of the insignificant polygon and all its skeleton edges to organise the polygons. Each boundary edge will make up one polygon together with some skeleton edges. The generated partitioning polygon can now be assigned to the other side neighbour of the boundary edge. This amalgamation maintains the topological consistency, not resulting in gaps or overlapping regions. Fig. 6 shows an experiment example of this amalgamation. In the left, the objects surrounded by the wide dark boundaries are unimportant (small size) and will be removed. The right graphic is the result of subdividing the insignificant objects.
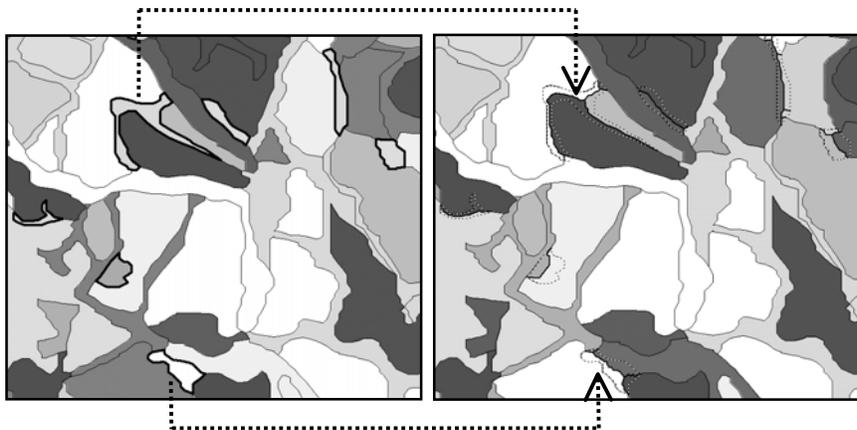

Fig. 6. An experiment of removing insignificant objects based on adjusted skeletons.

The skeleton generation above is based on the assumption that the neighbours have the same influence to divide the insignificant object and the skeleton edge partitions left/right region in an equal way. Further improvement, in future work, can be obtained by considering geometric relationship and semantic similarity. We use function compatibility(a,b) to compute the compatibility between object a (to be removed) and its neighbour object b, taking into consideration:

- The compatibility of the types: *type_comapt(a,b)*;
- The length of the shared edge: *length(a,b)*;
- The importance of *b: Imp(b)*.

The function could be: compatibility(a,b)=type_compat(a,b)*length(a,b)*imp(b).
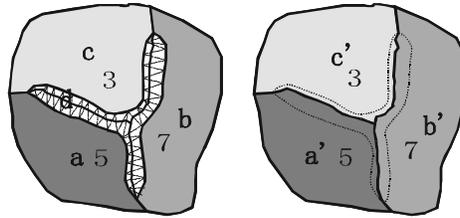

Fig. 7. The generation of adjusted skeleton.

We call this kind of skeleton, the adjusted skeleton, since it considers the compatibility difference between polygons competing for extending their area.
In the geometric algorithm, the adjusted skeleton can be an adjusted version of the normal skeleton. In Fig. 2, the skeleton points P_i need to be computed according to two segment ratios based on the compatibility values, rather than the middle position unless the two compatibility values are the same. For example, in Fig. 7, object *c* has compatibility value 3 and *b* has 7. For the region between object *c* and *b,* 30 percent will be assigned to object *c* and 70 percent will be assigned to object *b*.

5 Progressive Refinement of the GAP-Tree Construction

The aggregation of adjoining neighbours and the improvement of the amalgamation, changes the GAP-tree construction. The basic idea is that once an unimportant object is determined, it has the possibility either to be aggregated with its approximate neighbour or to be amalgamated into adjoining neighbours. The algorithm of the GAP-tree construction, which is an iterative process, is described in Fig. 8.

As usual, an edge in the GAP-tree represents the parent-child relationship between the unimportant polygon (child) and the polygon(s) to which it has been assigned (parent). Fig. 9 shows an example with the iterative process of the GAP-tree construction using both aggregation (P_1 steps) and amalgamation (P_2 steps).

It is possible that one 'neighbour region' lies on top of more than one polygon; see for example the upper-right image and the one below. This is because the 'neighbour region' is computed using a Delaunay triangulation of only the selected polygons, which have similar classification and are within tolerance distance of the least important polygon.
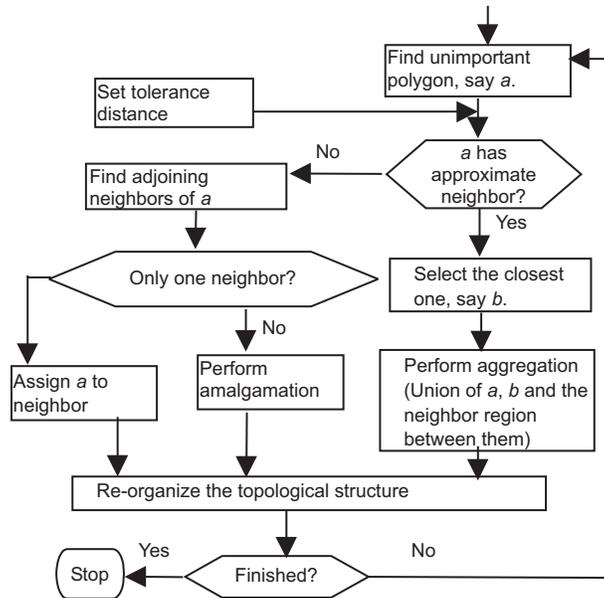
Fig. 8. The progressive construction of the extended GAP-tree.
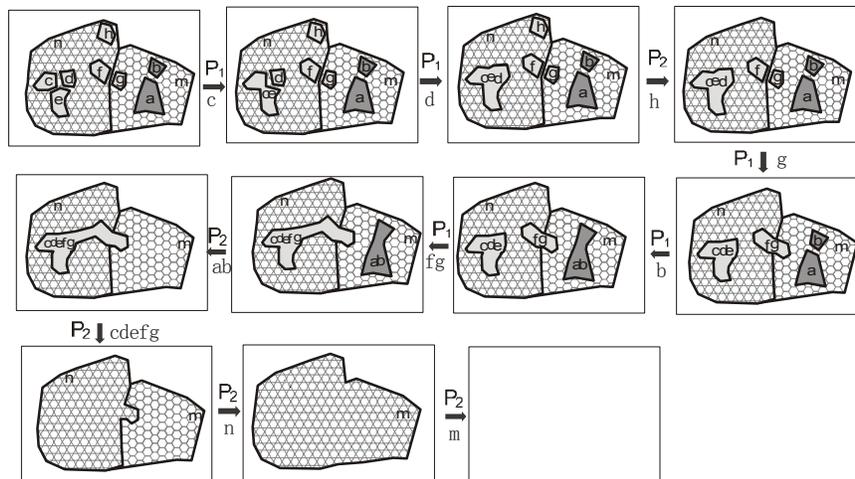

Fig. 9. The scene change during the extended GAP-tree construction.

Using nodes and edges to record the result of two kinds of operation, as shown in Fig. 10, we get a new data structure. As in the amalgamation operation, an object can be split into parts and each is assigned to a different parent, this will destroy the real tree structure of the GAP-'tree'. It has now become a 'directed

acyclic graph', because one child may have several parents (GAP-DAG). In the parent-child edges, it must be stored how big the part of the child object assigned to this parent is (*child-part,* with values between 0 and 1). The original area formula has to be changed into: *area*(object)=*Area*(object-GAP-node)-*sum*(*child-part * Area*(child-node))
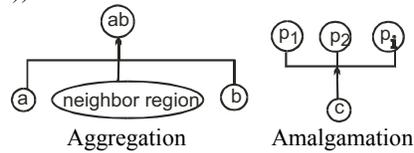

Aggregation   Amalgamation

Fig. 10. Structure for 2 operations

After construction, the GAP-tree is used during on-the-fly generalisation based on the painter's algorithm idea in computer graphics. This means, once an important object is drawn, it will never be drawn again during refinement. Only additional details will be drawn on top of it, each time refining the 'image'. These details include the neighbour regions in aggregation and the divided parts in amalgamation. In this way, it can be compared with the progressive transfer of the raster images on the Internet, which are also displayed from coarse to fine as there is more and more time for sending details (refinements). In the case of joining disjoint neighbours in the GAP-tree construction, which have to be visually disconnected again, when using the GAP-tree, the 'neighbour region' will be drawn on top of this. This is an exception to the rule that the most important (and first drawn object) will never be drawn again. In this case, a small fragment will be 'redrawn'.
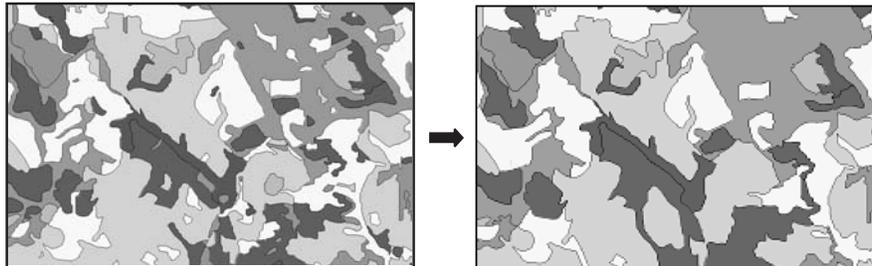

Fig. 11. The generalisation of land use objects based on the extended GAP-tree

The land use map, see Fig. 11, is a typical kind of categorical theme and has the properties covering the whole area with neither gaps nor overlapping areas. Usually it includes many categories in the classification hierarchy. Its generalisation has to consider geometric simplification, semantic hierarchy abstraction and statistical properties maintenance. In our generalization of land use data, we select one category of objects (resulting in a disjoint polygonal cluster) as operated objects, while others are classified as background objects. Fig. 11 gives an experimental illustration. Future research has to determine the proper parameters for compatibility formula computation in real land use data. For the

time being, we just use the normal skeleton (equal partitioning) to separate insignificant objects in this experiment.

6 Conclusion

Based on skeleton handling, this paper extends the GAP-tree construction in two aspects. The first one is the introduction of aggregation of disjoint polygon objects. The second one is the improvement of amalgamation between different category objects. Both extensions use the Delaunay triangulation and skeleton, but in different ways. The idea of two operations can be unified in the GAP-tree. On-the-fly generalisation (continuous zoom-in/out) can be achieved by using the GAP-tree based on painters algorithm. The initial tests show good generalisation results. This strategy is also very suitable to progressively transfer polygonal map in Internet.

In the past other generalisation techniques have been based on the Delauney triangulation, Voronoi diagram or the skeleton (Poorten and Jones, 1999, Jones, Bundy and Ware, 1995, Ware and Jones, 1995, 1997, Ai, Guo and Liu, 2000, Ai and Guo, 2000). This paper has added two more generalisation techniques, which proves again how powerful these structures are.

Further research topics related to GAP-tree use in generalisation include:

a) Improvement of neighbourhood definition, simultaneously considering topological, geometric, semantic and Gestalt nature's impacts. The compatibility formula between two neighbours has been defined, especially how to compute the parameters.
b) Application of this method in a web transferring environment, in a server-client framework, to realise the transfer of vector map in the way from coarse to refinement, just like images are refined during transfer.
c) Uncertainty of both the geometry and the classification of the objects could also be taken into account in the determination of the least important feature and its most compatible neighbours.

References

Ai T, Guo R, Liu Y (2000) A Binary Tree Representation of Bend Hierarchical Structure Based on Gestalt Principles. In: Forer P, Yeh AGO, He J (eds) Proceedings of the 9th International Symposium on Spatial Data Handling, Beijing, pp 2a30-2a43

Ai T, Guo R (2000) A Constrained Delaunay Partitioning of Areal Objects to Support Map Generalization. Journal of Wuhan Technical University of Surveying and Mapping 25(1):35-41 (in Chinese)

Bader M, Weibel R (1997) Detecting and Resolving Size and Proximity Conflicts in the Generalization of Polygonal Maps. In: Ottoson L (eds) Proceedings of the 18th International Cartographic Conference. Stockholm, pp 1525-1532

Hernandez D, Clementini E (1995) Qualitative Distance. In: Proceedings of COSIT'95, Semmering, Austria, pp 45-57

Jones CB, Bundy GL, Ware JM (1995) Map Generalization with a Triangulated Data Structure. Cartography and Geographic Information System 22(4): 317-331

Oosterom P Van (1995) The GAP-tree, An Approach to On-the-Fly Map Generalization of An Area Partitioning. In: Muller J C, Lagrange J P, Weibel R (eds) GIS and Generalization: Methodology and Practice. Taylor & Francis, London, pp 120-13

Oosterom P Van (1994) Reactive Data Structure for Geographic Information Systems. Oxford University Press, Oxford

Oosterom P Van (1991) The Reactive-Tree: A Storage Structure for a Seamless, Scaleless Geographic Database. In: Auto-Carto 10, pp 393-407

Poorten P, Jones C B(1999) Customisable Line Generalization Using Delaunay Triangulation, CD-Rom Proceedings of the 19th ICC, Ottawa, Section 8

Preparata FP, Shamos MI (1985) Computational Geometry An Introduction. Springers-Verlag

Putten J Van, Oosterom P Van (1998) New Results with Generalized Area Partitionings. In: Proceedings of Spatial Data Handling, Ottawa, Canada

Ware JM, Jones CB (1995) A Triangulated Spatial Model for Cartographic Generalization of Areal Objects. In: Proceedings COSIT'95. Semmering, Austria, pp 173-192

Ware JM, Jones CB(1997) A Spatial Model for Detecting (and Resolving) Conflict Caused by Scale Reduction. In: Kraak MJ, and Molenaar M (eds) In: Advance in GIS Research II (7th Int. Symposium on Spatial Data handling), Springer-Verlag