Spatial Object Aggregation Based on Data Structure, Local Triangulation and Hierarchical Analyzing Method

LIU Yaolin Martien Molenaar Menno-Jan Kraak LIU Yanfang

1 Introduction

Aggregation for spatial objects can serve as basic tools for the generation at multiple representations of geo-data within the context of database generalization processes. Aggregation process for spatial objects needs not only analyzing and understanding semantic, geometric characteristics of spatial objects and the relations among them, but also supporting data structures. There is still lack of studies on how data structure supports aggregation process and of the method for semantic analysis in aggregation process though there are several data structures used in the multiple representation of geo-database such as R-tree, B-tree, SDS, etc. The data that are organized by some data structure will be more effective and efficient to be analyzed and measured. The data structures play an important role in identifying and evaluating conflicts and analyzing spatial relations among objects. In this research, we present a data model based on the Formal Data Structure (FDS) and the Local Constrained Delaunay Triangulation\(^\text{[1,2]}\). The semantic relation matrix between different objects and between different object types will govern or guide aggregation process.

The aggregation process for spatial object can be divided into two steps, the semantic analysis of spatial objects and the geometric analysis and operation of spatial objects. The main task of the former step is to establish the semantic similarity matrix by expert knowledge based on the aggregation hierarchy and classification hierarchy as well as the requirements and purposes of database generalization. It is application-dependent. An aggregation hierarchy plays a key role in linking the definition of spatial objects at different scale levels\(^\text{[3,6]}\). It shows how composite objects can be built from elementary objects.
and how these composite objects can be put together to build more complex objects and so on[7].

The main task of the latter step is analyzing the geometric characteristic of spatial objects, the spatial relations among the objects and implementing operation to objects.

2 Elementary object, elementary object type, composite object, composite object type and aggregation hierarchy

Some of concepts should be made clear in aggregation process.

2.1 Elementary object

Elementary object in spatial model is a real object that contains both thematic and geometric information and is normally represented in a database by means of an "object identifier" with associated thematic and geometric data. These elementary objects are divided into three types (point, line and area) based on their geometric attributes. Objects can also be categorized according to their thematic attributes. The definition of elementary object in a database (should be done before a database can be built) depends mainly on the following four factors:

- application discipline
- user context
- aggregation level or scale or resolution
- classification level

On each level, different elementary objects are relevant. Elementary objects at one level may be aggregates of elementary objects at another level.

2.2 Elementary object type

Elementary object type is an abstraction that represents a class of similar elementary objects. This means that the elementary objects in a spatial model that have common pattern of both state and behavior within the framework of an application may be grouped into elementary object types, which may be organized into super object types and so on. An elementary object is an instance of some elementary object type. Elementary object types together with the classification and aggregation hierarchies are important aspects in semantic data modeling and play a critical role in defining the concept of database generalization[3].

2.3 Composite object

Composite object is built from elementary objects that belong to different elementary object types. This means that the elementary objects are the constituents of composite object. Similar to elementary objects, composite objects at one level may be aggregation of composite objects at another level.

2.4 Composite object type

Composite object type is also an abstraction that represents a group of similar composite objects. An instance of the composite object type is referred to a composite object. A composite object type can be the elementary object type of another (super) composite-type. For example, the object type Farm is a combination of the types Yard and Field. In other words, Yard is part of Farm, and so is the Field.

2.5 Aggregation hierarchy

Aggregation hierarchy shows how lower-order object types are combined to form a higher-order object type. A higher-order object type in the hierarchy is composite object type, whereas an object type that is part of the composite-type is elementary object type. This aggregation hierarchy has the following characteristics:

- defining elementary objects,
- composite object types in the hierarchy corresponding to higher abstraction levels,
- elementary object types corresponding to lower abstraction levels,
- specifying the elementary object types of elementary objects building a composite object of this type,
- the upward relationships taking the part of connecting a group of elementary objects with a certain composite object,
- expressing the relationship between a specific composite object and its constituent parts at different levels,
replacing the elementary object types in a model with their composite-type will result in transforming the model from a lower abstraction level to a higher abstraction level.

An aggregation hierarchy has therefore a bottom-up character, in the sense that the elementary objects from the lowest level are combined to compose increasingly composite objects as one ascends in the hierarchy. The definition of aggregation hierarchy is application-dependent and it must be established before aggregation process. The different applications have different aggregation hierarchies, even though these aggregation hierarchies are based on the same data set or thematic classification system. Suppose that we have a land use database in which the contents of the database can meet the requirements of land management at several levels and land evaluation for different land use at several levels. In order to meet different level management and different purpose of land evaluation at different levels, the database needs to be generalized corresponding with management level and evaluation level based on different aggregation hierarchy. The aggregation hierarchy used for land use management is completely different from the aggregation hierarchy for land evaluation. The same database for different land use evaluation at different levels will have different aggregation hierarchy for each one as well.

3 Supporting data model to aggregation

The semantics relations and the spatial relations between objects must be analyzed and geometric operations must be implemented in the aggregation process. A spatial object may be influenced by neighboring objects with no restriction on the distance of separation. No matter how far two objects are separated, one can affect the other as long as they are neighbor. For example:

- If two parcels have the same land use or high similarity and have adjacent relation, then they are aggregated. This involves the geometric operation.
- Parcel A which violates the geometric constraint (too small in area) have adjacent relation with several land parcels. One of this set of land use parcels having the same land use or high similarity as Parcel A should be selected out and aggregated with Parcel A to form a new parcel. This involves the semantic analysis among objects and geometric operation.
- A set of spatial objects have regular pattern in spatial database. First identify them from semantic and spatial aspects, then aggregate them. This involves semantic and spatial analysis and geometric operation.

Before aggregation process, the spatial objects with three components: the distance, the direction and the state of neighbors need to be examined and the semantic relations among the objects need to be analyzed. After aggregation, the geometric and textual properties of spatial objects need to be changed and the spatial relations need to be maintained dynamically. Dynamic maintenance of adjacent relations is a critical issue in aggregation process. All these examination and analysis show that the aggregation need data model to support. If there is no data model supporting these analysis and operations, then a process involving a probably heavy computation and complicated algorithm is necessary to complete the aggregation process. A data model, adequate for automated aggregations, should provide the basis for describing spatial object and the topological relationship among them, through a well defined set of geometric primitives. Since aggregation decision-making relies on both spatial information and thematic information, the model should also indicate how the geometric aspect of a spatial object is linked with its thematic aspects.

The Formal Data Structure (FDS) model for single valued vector maps developed by Molenaar (1989, 1991, 1995) is an object-oriented topological (conceptual) data model. Although the FDS supports a number of elementary topological relationships, it does not support the spatial adjacency relationship among objects that are disconnected from each other. Topological relationships among "disconnected objects" are important to support spatial analysis and geometric operations that in-
volve these kinds of objects.

The constrained Delaunay Triangulation (CDT) can be used for defining the adjacent relation among disconnected objects, conflicting the detection and displacement of spatial objects and finding nearest neighboring object to a given object in generalization\cite{4,8,9}. The CDT can also be used to measure the spatial relations such as measuring disjoint relation, distance relation and direction relation. CDT can be as the geometric primitives of the data model. For category database generalization, CDT is very useful to analyze and measure local spatial relationship but not to organize the whole data set since a simple area object will consist of a lot of triangles that will lead to data redundant and also lead to the difficulty of semantic analysis among objects.

Aggregation process needs not only an analysis process of database concerned from the local to whole but also from whole to the local. The data model should support this analysis process.

We combine the advantage of FDS and CDT to develop a data model which is the integration of FDS and CDT in aggregation process. The data model is shown in Fig. 1. Fig. 2 shows that logical structure of geo-database is organized in database based on this data model.

In structure modeling, the objects in database are represented as an object class, Geo-class, which is super class of point, line and area objects. It can contain some attributes and member function such as Distance ( ) and Direction ( ). The data model consists of:

- three object types, namely point object, line object, area object, classified according to the geometric description of spatial object,
- five geometric data types (geometric primitives), including coordinates, node, edge, triangle and face, the definition of which is based on planar-graph theory at node-arc level,
- a set of links between geometric data types (g-g links), and a set of links between geometric data types and object types (g-f links). It supports a number of elementary topological relationships, including area-area, line-line, point-point, area-line, area-point, and line-point relationships.

Assuming that there are three objects \(O_1, O_2\) and \(O_3\) as shown in Fig. 3, in a constrained Delaunay triangulation \(T\), every boundary vertex of each object \(o_i\) corresponds to a triangle vertex and every boundary edge of each object \(o_i\) serves as a constraining edge. Each object \(o_i\) is defined by a
unique object identifier and refers to the triangles of $T$ which lie within its boundary; each triangle $t_i$ of $T$ is described by a unique triangle identifier, refers to each of its three constituent edges, plus a reference to the object within which it lies; each edge $e_i$ is described by a unique edge identifier and refers to its start and end vertices; and each vertex $v_i$ stores a unique vertex identifier plus $x$ and $y$ coordinate values, plus supplementary topological information in the form of reference to the two triangles to which the edge belongs. If a triangle $t_i$ lies within an object $o_j$, then $t_i$ is said to belong to $o_j$. It has the following properties:

1) Its nodes of triangle must be on the boundary of objects.
2) The edges in a triangle of CDT are divided into two groups:
   - edges belonging to the part of the boundary of the object (named constrained edges)
   - edges not belonging to the part of the boundary of the object.
3) The triangles in CDT are classified as three types:
   - triangles having only one constrained edge in its three edges, seeing Triangles $b, g, i, h$ in Fig. 3, and Tri1EdgeList denoting a set of this kind of triangles,
   - triangles having two constrained edges in its three edges, seeing Triangles $a, f, k$ in Fig. 3, and Tri2EdgeList denoting a set of this kind of triangles,
   - triangles having no constrained edge in its three edges, seeing Triangle $d$ in Fig. 3, and Tri0EdgeList denoting a set of this kind of triangles.
4) The adjacent relationship can be grouped into three types in triangle net:
   - two triangles sharing a common constrained edge, seeing Triangles $i$ and $k$, Triangles $f$ and $d$ in Fig. 3;
   - two triangles sharing a common non-constrained edge, seeing Triangles $c$ and $d$, Triangles $h$ and $k$ in Fig. 3;
   - two triangles sharing a common vertex, seeing Triangles $g$ and $h$ in Fig. 3.
5) If two objects, $o_1$ and $o_2$, share a common node of triangle, they are adjacent.
6) If two objects, $o_1$ and $o_2$, share a common constrained edge of triangle, they are adjacent.
7) If two objects, $o_1$ and $o_2$, share a common triangle between them, they are adjacent, seeing Object $o_3$ and $o_1$ in Fig. 3.

These good properties of the CDT will be used for spatial analysis in aggregation process, such as rendering measurement of distance and direction among objects, maintaining topological links between points, lines and polygons, facilitating easier local proximity relations between the objects. The triangle is used as a basic unit to analyze geomet-
lic characteristics of the object and the proximal relation among the objects in this research.

For functional modeling, the model possesses two function members, Distance( ) and Direction( ). Distance( ) is a function that returns the metric distance from one object to another. Direction( ) returns the bearing from one object to another. As above mentioned, distance and direction are two important parameters on geometric aspects in aggregation process. Apart from Distance( ) and Direction( ), the data model combines aspects of objects-oriented and topologic data model. Point, line, area objects are represented with their geometric and thematic aspects. Their geometric representation contains information about topologic object relationships, whereas their thematic description is structured in object classes that may form generalization hierarchies. Such class hierarchies in combination with the topological object relationships of the data model supports the definition of aggregation hierarchies of objects. These classification and aggregation hierarchies play an important role in linking the definition of spatial objects at several scale levels. They govern and guide the aggregation process.

4 Semantic analysis of aggregation

Whether the two adjacent objects or an adjacent group of objects can be merged or aggregated depends on whether the attributes of the two objects are the same or similar. If their attributes are the same or similar, they can be merged or aggregated. Otherwise they can not.

The closeness among objects and object types can be described by the similarity. The similarity is application-dependent. The larger the value of similarity between objects is, the closer or more similar they are. The similarity among elementary objects, elementary object types, composite object types and the like can be established by expert knowledge based on the aggregation hierarchy and classification hierarchy, requirements and purposes of database generalization. A matrix can be used to represent similarities among objects and object types as shown in Fig. 4. The matrix is symmetric and reflexive, and has the property that $s_{ij}$ is equal to $s_{ji}$ ($s_{ij} = s_{ji}$) and $s_{ii}$ is equal to $s_{ij}$ ($s_{ij} = s_{jj} = 1$) while $s_{ij}$ is a value between 0 and 1. This matrix has the characters of hierarchy and shows the similarity between the objects of the same level and between the objects of different levels. This will provide potential possibility to merge or aggregate the objects of the same level or different levels. This also means that objects of different levels can be kept in a spatial database. The similarity matrix will be used as a look-up table for guiding or governing the aggregation process of spatial objects. An example for computing similarity matrix from classification hierarchy is given as follows:

$$s_{ij} = \begin{cases} 
1 & \text{(same object or its type)} \\
1 - \frac{e_i}{e_{\max}} & \text{(different object or its type)} 
\end{cases}$$

where $e_{\max}$ denotes the maximum link number of edge between different nodes in classification hierarchy, and $e_i$ denotes the link number of edge between different nodes in classification.

where Obj1, Obj2, etc. denote different elementary objects; Subtype 1, Subtype 2, etc. denote different elementary object types; Supertype 1, Supertype 2, etc. denote composite object type; $s_{ij}$ denotes sim-

| SIMILARITY | Obj1 | Obj2 | ... | Subtype 1 | Subtype 2 | ... | Supertype 1 | Supertype 2 | ... |
|------------|------|------|-----|-----------|-----------|-----|-------------|-------------|-----|
| Obj1       | $S_{11}$ | $S_{12}$ | ... | $S_{14}$ | $S_{15}$ | ... | $S_{19}$ | $S_{18}$ | ... |
| Obj2       | $S_{22}$ | $S_{23}$ | ... | $S_{24}$ | $S_{25}$ | ... | $S_{29}$ | $S_{28}$ | ... |
| ...        | ...   | ...   | ... | ...       | ...       | ... | ...         | ...         | ... |
| Subtype 1  | $S_{32}$ | $S_{33}$ | ... | $S_{34}$ | $S_{35}$ | ... | $S_{39}$ | $S_{38}$ | ... |
| Subtype 2  | $S_{42}$ | $S_{43}$ | ... | $S_{44}$ | $S_{45}$ | ... | $S_{49}$ | $S_{48}$ | ... |
| ...        | ...   | ...   | ... | ...       | ...       | ... | ...         | ...         | ... |
| Supertype 1| $S_{52}$ | $S_{53}$ | ... | $S_{54}$ | $S_{55}$ | ... | $S_{59}$ | $S_{58}$ | ... |
| Supertype 2| $S_{62}$ | $S_{63}$ | ... | $S_{64}$ | $S_{65}$ | ... | $S_{69}$ | $S_{68}$ | ... |

Fig. 4 Matrix of similarity
5 Geometric analysis of aggregation

Whether two objects or a group of objects having the same thematic attributes can be merged or aggregated or not depends on whether they are adjacent or the space between them is smaller than the threshold or not. This means that the adjacent objects (or neighbors) must be identified and the space between them must be computed before aggregation operation. Those procedures can be done by analyzing the spatial relationship among the geometric primitives with the aid of CDT.

5.1 Detection of neighboring conflict objects

For effective aggregation process, the number and location of objects that violate the constraints need to be identified rapidly. The neighborhood of the violated objects must be identified in order to aggregate them. We now describe a method for identifying the neighbor set of a particular object based on the data model which we developed. The procedure is primarily concerned with finding triangles which have constrained edge connecting with \( a_i \) (i.e. triangles that share a constrained edge with \( a_i \)) through triangles \( t \) which belong to \( a_i \). Any such a triangle which does not belong to \( a_i \) is added to List, provided it has not already been added. Given an object \( a_i \), it returns a list of the identifiers associated with all triangles connected with \( a_i \) (but not belonging to \( a_i \)) through the adjacent relations with triangles which belong to \( a_i \). The process of finding adjacent triangles of \( a_i \) continues until all triangles \( t \) within \( a_i \) have been processed. The adjacent triangles of \( a_i \) are of interest in that they provide a convenient way of identifying its connectivity and proximal neighboring objects. These neighbors are found by examining each of \( a_i \)'s adjacent triangles in turn. If a particular adjacent triangle \( t_p \) belongs to an object \( a_i \), then \( a_i \) and \( a_j \) are connected. However, \( t_p \) belongs to \( F \), then each of the objects contiguous to its vertices needs to be found. It follows that each of these objects (with the exception of \( a_i \) itself) is proximal to \( a_i \).

The process is described as follows, taking Fig. 4 as an example for illustration.

1) Let \( a_i \) in Fig. 4 is the object in consideration.
2) For \( a_i \), store \( t_i (t_i \in a_i) \) to TriObjectList.
3) For each \( t_i (t_i \in \text{TriObjectList} \land (t_i \in \text{TriEdgeList} \lor t_i \in \text{Tri2EdgeList}))) \), do the following:
   1. Find out the neighbor triangle \( t_j \) which is connected with \( t_i \) through the constrained edge (the boundary of the object \( a_i \)), and \( t_j \) does not belong to \( a_i \).
   2. Store \( t_j \) to TriNeighborList.
4) For \( t_j \in \text{TriNeighborList} \)
   1. Get area neighbor object of \( a_i \) by \( t_j \) with the conception of \( a_i \).
   2. Store it in ObjectNeighborList if it does not exist in ObjectNeighborList.
5) For \( a_i \) and \( a_j \in \text{ObjectNeighborList} \), do the following:
   1. Get a set of triangles between \( a_i \) and \( a_j \) from TriNeighborList, and store them to TempObjectList.
   2. For each triangle \( t_k \in \text{TempObjectList} \), do the following:
      1. Compare \( L \) (the average edge length of \( t_k \)) with the threshold \( d_{\text{threshold}} \).
      2. If \( L > d_{\text{threshold}} \), do nothing, or else store \( a_i \) in ObjectConflictList preparing for aggregation.
   3. Here, note that the average edge length of a triangle represents approximately the distance between two objects.

5.2 Detection of conflict set of objects with regular distribution pattern

How to identify a set of objects with regular spatial or semantic pattern is very important in aggregation process. Although there are many possible patterns that are of interest to aggregation, the following discussion concentrates on the detection of groups of objects which have regular pattern in di-
rection within a larger group. The particular example to be considered concerns the detection of (linear) groups of landuse elementary objects (Fig. 8). Human eyes will detect such a group when:

- The distance between \( o_i \) and neighboring elementary objects in the group are similar and normally less than the distance to the nearest objects outside the group, from any member of the group;
- The objects in the group have normally the same attributes;
- \( o_i \in \text{ObjectConflictList} \) as seed to repeat the above procedure until \( t_j \) does not belong to \( o_i \).

The procedure starts with the object in ObjectConflictList which has been gotten by DetectNeighborObject. The object in ObjectConflictList will be as seed to look for new conflict objects or its neighboring objects within \( d_{\text{threshold}} \) to it. Each object in ObjectConflictList will be processed using DetectNeighborObject method. The new conflict objects will be added in ObjectConflictList if they have not been added before. If the object in ObjectConflictList has been processed, the sign is assigned for it. This means that the object in ObjectConflictList with sign does not need to be processed any more. The objects with sign in ObjectConflictList will be a pattern group of objects if they have the pattern of spatial distribution.

The procedure of detecting a regular group of objects within a larger group is formulated below.

1) For \( o_i \in \text{ObjectConflictList} \), do the following:

   1. Get all area neighboring conflict objects of \( o_i \) using the above processing method if \( o_i \) has not been processed, store them into ObjectConflictList, and store \( o_i \) into ObjectGroupList.
   2. For \( o_j \in \text{ObjectConflictList} \) and \( o_j \neq o_i \), do the following:
      - Assign a sign of the object \( o_j \) that has been processed.
      - Until all objects have been processed in ObjectConflictList.

2) Get a regular group of objects

3) Empty ObjectConflictList

5.3 Extraction of skeleton line

Suppose an area object violating the geometric constraints has similar value in similarity matrix with its neighboring objects, for the area balance of each class after aggregation, the area object must be skeletonized. Each part of area object after skeletonizing the area object is assigned to neighboring objects (Fig. 6). Extraction of skeleton line of double line river is also very useful in river generalization. Here we concentrate on the extraction of the skeleton line of polygonal objects. The extraction of the skeleton line of polygonal objects starts with triangle \( t \) having two constrained edges (seeing Triangle \( a \) in Fig. 3) or triangle \( tt \) having no constrained edge (seeing Triangle \( d \) in Fig. 3), and find out all adjacent triangles \( ttt \) having only one constrained edge through sharing non-constrained edge between two triangles and end with \( t \) or \( tt \), and compute the mid-point of non-constrained edge of \( ttt \), and compute the mid-point of non-constrained edge and center point of triangles \( tt \). The skeleton line will be detected by connecting all special points. The procedure is as follows:

For \( t_{2i} \), \( t_{2i} \in \text{Tri2EdgeList} \) or \( t_{0i} \), \( t_{0i} \in \text{Tri0EdgeList} \), do the following:

- Start with \( t_{2i} \) or \( t_{0i} \), find out all adjacent triangles through sharing non-constrained edge between two triangles and end with \( t_{2i} \) or \( t_{0i} \).
- Compute the mid-point of non-constrained edge of \( t_{2i} \) and \( t_{1i} \), \( t_{1i} \in \text{Tri1EdgeList} \).
- Compute the center point and mid-point of non-constrained edge of \( t_{0i} \).
- Get a sequence of points which describe the skeleton line of the object.
- Connect the opposite points of the non-constrained edges of \( t_{2i} \) with the mid-points of the constrained edge of \( t_{2i} \), and connect the two mid-points of two non-constrained edges of \( t_{1i} \), and connect the mid-points of three mid-points of three non-constrained edges with center point of \( t_{0i} \) respectively.
6 Examples of aggregation operation

The following part gives five different cases of aggregation by the above methods.

Case 1: Assuming that there are five area objects $a_1, a_2, a_3, a_4$ and $a_5$ which belong to different object types as shown in Fig. 5(a). The area of Object $a_3$ is less than the area threshold which each object must have in database. So Object $a_3$ must be aggregated with one of $a_1, a_2, a_4$ and $a_5$. Which one should be selected in $a_1, a_2, a_4$ and $a_5$ to merge with $a_3$ depends on the similarity and relations among $a_3, a_1, a_2, a_4$ and $a_5$. $a_3$ has connectivity relation with $a_2, a_4$ and $a_1$, and has proximal relation with $a_5$. We take similar value between objects from similarity matrix, compare their value size and select largest one for a group of objects having connectivity relation such as $a_1, a_2, a_3, a_4$. The two objects having largest value will be aggregated as shown in Fig. 5(b), (c) and (d). For $a_3$ and $a_5$, the distance between them must be measured first. If the distance is less than the distance threshold which database specifies, the similarity value can be taken from similarity matrix, if the distance is greater than the distance threshold, there is no need to further process $a_3$ and $a_5$.

Case 2: Suppose that there are four objects $a_1, a_2, a_3$ and $a_4$ which belong to different object types as shown in Fig. 6(a). These objects have connectivity relation, $a_2$ is too narrow and should be aggregated with its neighboring objects, $a_2$ has the same similarity value from similarity matrix with the other three objects. So $a_2$ should merge partly with other three objects. First the skeleton line of $a_2$ is created (c.f. Fig. 6(b)). This line will become a part of edges of $a_1, a_3$ and $a_4$. Then the edges of $a_1, a_3$ and $a_4$ and topological relation among them are reorganized (c.f. Fig. 6(c)).

Case 3: Suppose that there are four objects $a_1, a_2, a_3$ and $a_4$ which belong to different object types as shown in Fig. 7(a). $a_3$ contains $a_1$ and $a_2$, and they have inclusion relation. That $a_1$ collapses into $a_3$ or merges with $a_4$ depends on the similarity and distance between $a_1$ and $a_4$. If the distance between the two objects is less than the distance threshold and the similarity value between $a_1$ and $a_4$ is greater than that between $a_1$ and $a_3$, $a_1$ will be merged with $a_4$ (c.f. Fig. 7(b)). Other-
Case 4: Suppose that there are a group of objects which belong to the same object type as shown in Fig. 8(a). These objects have proximal relation (cf. Fig. 8(b)). First a subgroup of objects which violate the distance threshold must be identified using the above method (cf. Fig. 8(c)). Then the conflict objects will be merged (cf. Fig. 8(d), (e)) and at the same time the relationship among these objects will be changed.

Case 5: Suppose that there are a group of objects with proximal relation among them which belong to different object types. First, a subgroup of objects which violate the distance threshold must be identified using the above method. Second, the similarity among these conflict objects in geometry need to be analyzed. Third, these conflict objects will be divided into several groups based on the similarity among these objects. Finally, each group of objects will form a new object and the spatial relations among them will also be changed.

7 Conclusion

The data model which is presented in this paper integrates the advantages of the Formal Data Structure and the Local Constraint Delaunay Triangulation. This data model used in aggregation process avoids involving a probably heavy computation and complicated algorithm in aggregation operation.

The data model for aggregation has a lot of good properties, which allow to measure spatial relations such as disjoint relation, distance and direction relation among objects. The similarity matrix with hierarchical character reflects the semantic relation among spatial object and object type.

References

1. Molenaar M. (1989) Single valued vector maps-a concept in GIS. GeoInformations-Systeme, 2:18-26
2. Molenaar M. (1991) Terrain objects, data structures and query spaces. In Schlicher M., ed. Geo-Informatik, Siemens-Nixdorf Informations-Systeme A.G., 53-70
3. Molenaar M. (1996) Multi-scale approaches for geodata. International Archives of Photogrammetry and Remote Sensing, XXXI(B3)
4 Peng W. (1997) Automated Generalization in GIS: [Ph. D Thesis]. The Netherlands: Wageningen Agricultural University and ITC, 188

5 Richardson D. E. (1993) Automated spatial and thematic generalization using a context transformation model: [Ph. D Thesis]. Canada: R & B Publications, 149

6 van Smaalen J. W. N. (1996) A hierarchic rule model for geographic information abstraction, SDH 96, 4B: 31-4B.

7 Molenaar M. (1993) Object hierarchies and uncertainty in GIS or why is standardisation so difficult. Geoinformations-Systeme, 6(3): 22-28

8 Ware M. J., Jones C. B. (1996) A spatial model for detecting and resolving conflict caused by scale reduction. 7th International Symposium on Spatial Data Handling, Delft, The Netherlands

9 Jones C. B., Ware M. J. (1998) Nearest neighbor search for linear and polygonal objects with constrained triangulations. 8th International Symposium on Spatial Data Handling, Delft, The Netherlands

10 Muller J. C., Weibel R., Lagrange J. P., Salge F. (1995) Generalization: state of the art and issues. In: Muller J. C., Lagrange J. P., Weibel R., eds. GIS and Generalization. Taylor & Francis, 3-17

11 Peuquet D. (1984) A conceptual framework and comparison of spatial data models. Cartographica, 21(4): 22-113

12 Pilouk M. (1996) Integrated modelling for 3D GIS: [Ph. D Thesis]. The Netherlands: Wageningen Agricultural University, 200

13 Bundy G. L., Jones C. B., Furse E. (1995) Holistic generalization of large-scale cartography data. In: Muller J. C., Lagrange J. P., Weibel R., eds. GIS and Generalization. 106-116

14 Smith J. M., Smith D. C. P. (1977) Database abstractions: aggregation and generalization. ACM Transactions on Database System, 2: 105-133

15 Thompson P. J. (1989) Data with semantics: Data models and data management. New York: Van Nostrand Reinhold.

16 McMaster R. B. (1991) Conceptual frameworks for geographical knowledge. In: Buttenfield B. P., McMaster R. B., eds. Map Generalization: Making Rules for Knowledge Representation. United Kingdom: Longman, House Essex, 21-39

17 Muller J. C. (1991). Generalization of spatial databases. In: Maguire D. J., Goodchild M. F., Rhind D. W., eds. Geographic Information Systems. New York: Longman Scientific & Technical Ltd, 1: 457-475

18 Weibel R. (1995) Three essential building blocks for automated generalization. In: Muller J. C., Lagrange J. P., Weibel R., eds. GIS and Generalization. 56-69

Notes to Contributors

Contributions are welcomed on one of the following subjects or in related areas:

- GIS
- Geodynamic
- Physical geo-surveying
- GPS
- Geo-surveying
- Engineering surveying
- RS
- Photogrammetry
- Mapping apparatus
- Cartology
- Graphics

Paper submitted in the electronic text on a diskette should be sent along with two printed copies. The main text should be preceded by the abstract of no more than 300 words, followed by key words. Full references should be listed in the order of the citations in the text under the heading "References", guided by standard publication format. The name of the fund and project series number for articles of funded projects should also be given.

The authors assume sole responsibility for their dissertations. Canned theses are not allowed. Once the paper is submitted, the editors are authorized to make necessary literate processing. Authors who receive no notice in three months after submission may contact us for inquiry.