MODELING MOVEMENT RELATIONS IN DYNAMIC URBAN SCENES

P. Partsinevelos, P. Agouris, A. Stefanidis
Dept. of Spatial Information Engineering
National Center for Geographic Information and Analysis
University of Maine
5711 Boardman Hall #348
Orono, ME 04469-5711
USA
{ panos, peggy, tony }@spatial.maine.edu

KEY WORDS: Spatio-Temporal, Data models, Image sequence, Image understanding, Topology, Dynamic processes.

ABSTRACT

In this paper we address the problem of analyzing and managing movement in dynamic scenes captured in video datasets. We present an approach to summarize the important information included in video datasets by analyzing the trajectories of objects within them. Trajectories form the basis for the construction of abstract data types upon which similarity of behaviors and expected behavior processes take place. We base our selection of representative time instances on the segmentation of trajectory lines into break points termed "nodes". The nodes are distributed dynamically to capture the information content of regions within the 3-D spatiotemporal space. They are computed through self-organizing maps of neural network processing. Additional nodes are supplementing the procedure and are originated from reasoning and proximity analysis. Topologic relations between moving objects in the scene and dynamic topology of the trajectories are processed in order to include the significant information of movement relations in the product summary. This work provides a novel approach to manage dynamic scene analysis at higher levels of abstraction, and to represent concisely the behavior of moving relations in a scene.

1. INTRODUCTION

Classic representations of a scene through static images tend to be insufficient to capture the nature of dynamic environments. Non-static events are becoming increasingly the focus of geospatial applications. Currently, there is a lack of methods to efficiently describe, model, and analyze positional change (henceforth termed as movement) in dynamic scenes. Modeling movement will enable the differentiation of semantic noise from substantial information, and will provide an important step towards efficient generalization of large amounts of spatially overlapping multi-resolution and multi-temporal data.

In this paper, we address the problem of analyzing and managing complex dynamic scenes depicted in video datasets. Efficient modeling of dynamic environments is an important step towards the analysis and management of large video datasets. The objective of the process described in this paper is the generation of information-rich summaries of video scenes that will describe the spatio-temporal behavior of objects within a depicted scene. These summaries are envisioned as new concise multimedia videos comprising vectors and images that portray the significant parts of the original video dataset.

The development of concise representation schemes is essential for the search, retrieval, interchange, query, and visualization of the information included in video datasets. Efforts towards this direction include attempts to summarize video by selecting discrete frames at standard temporal intervals (e.g. every n seconds). However, such an approach would typically fail to capture and represent the actual content of the original video dataset. Summarization alternatives include the use of image templates, statistical features and histogram based retrieval and processing (Chang et al., 1998). Video summaries have also been proposed, taking into consideration both visual and speech properties and constructing a "skim" video that represents a synopsis of the original video. This "skim" Video is constructed by merging segments of the original video (Smith et al., 1995). Video posters are proposed alternatives to describe story content (Yeung et al., 1997), while (Vasconcelos et al., 1998) has presented approaches to identify different scenes within a video stream by analyzing a variety of properties (e.g. dominant motion). In the trajectory domain, for fixed environments, systems extract and recognize moving objects, and classify the motion. (Medioni, 1998), (Rosales, 1999).

Data modeling is important for the manipulation of extensive and complex data such as video datasets. Use of hierarchical data structures provides higher level of information and leads to computationally less expensive management. Sorting data according to their spatial occupancy through tree structures is a promising data manipulation scheme. (Samet, 1990) (Sellis, 1987). Topological spatial relations support spatial analysis with focus on relations in a higher information level where further processing is accommodated. Related work in topology reasoning includes processes on the 9-intersection model, modeling of gradual changes of topological relationships, combined models for direction and topology etc. (Egenhofer, 1992), (Bruns, 1996). Papadias et al. (1995) introduces minimum bounding rectangles to relate objects and uses R-trees for indexing their relations.

Here we present an approach to summarize video datasets by analyzing the trajectories of objects within them. Our work is based on the identification of nodes in these trajectories as critical points in the video stream. These nodes form a generalization of the trajectory of a moving object within a video stream. The time instances that correspond to these nodes provide the critical frames for a video summary that describes adequately and concisely an object's behavior within a video segment. In doing so, we benefit from substantial advancements in object extraction from digital imagery, and video image processing.

The paper presents a framework for video summarization using this approach. Section 2 offers an overview of our overall approach. In section 3 we present the algorithm for the selection of the nodes in the defined spatiotemporal domain, while abstract data types are introduced in 4 facilitating similarity matching analysis. Section 5 extends the analysis for multiple objects and section 6 presents the framework for additional node selection. Finally, section 7 introduces the topologic approach for scene summarization. Experimental results are used throughout the paper to demonstrate the performance of the designed algorithms.

2. OVERVIEW

In monitoring applications the background usually remains fixed while objects move throughout the scene (e.g. cars moving in a parking lot monitored by a camera atop a nearby building). In such an environment, the crucial elements for video generalization are the ones describing the behaviors of the moving objects in time. We consider the spatio-temporal space of a scene as comprising two *(x,y)* spatial dimensions and one *(t)* temporal dimension. Object movements are identified by tracing objects in this 3-dimensional (x, y, t) space. These trajectories are the basic elements upon which our summarization scheme is based. An outline of our approach is shown in figure 1. It should be noted that the spatial coordinates *(x,y)* are the ones defined by the image space. If we want to translate them into survey coordinates we can use any of the well-known orientation models and relevant pose/rotation parameters of the specific video camera. While we assume a fixed sensor, we could easily handle cases of moving sensors by relating the variable video coordinates to the fixed axes of a suitable mosaic model (e.g. Zhou & Albertz, 1999).

Trajectories form the basis for the construction of abstract data types upon which similarity of behaviors and expected behavior processes take place. We base our selection of representative frames on the segmentation of trajectory lines into break points termed "nodes". The nodes are distributed dynamically to capture the information content of regions within the above mentioned 3-D S-T space. More nodes are assigned where trajectory presents S-T breakpoints, and fewer nodes are assigned to segments where the spatiotemporal behavior of an object is smooth. Node placement is based on concepts of self organizing maps (SOM) from neural network theory. The number of nodes may be selected to control the degree of generalization, similar to the number of nodes in a k-means approach.

Additional nodes are supplementing the SOM ones originated from reasoning and proximity analysis. Topologic relations between moving objects in the scene or topology of the trajectories in time, provides us with the basis to include the significant information of movement relations in the product summary. To facilitate such process octree structures are designed.

The summary of a video is a new shorter video, which includes a base map-image representing the background of the monitored area. Actual video frames are used at node instances, while the behavior of objects between nodes is represented by rapidly evolving vectors (e.g. moving spots or trace lines). The time it takes to bridge nodes in a video summary depends on the desired duration of the video summary. It is equivalent to selecting a tape speed to fast forward between events. This visualization aspect is beyond the focus of this paper, where we focus on node selection for this scenario.

Combined, the above provide a novel approach to manage dynamic scene analysis at higher levels of abstraction, and to visualize concisely the behavior of moving relations in a scene. In doing so we benefit from advancements in computer vision and digital image analysis, and transfer methodologies from these areas into spatiotemporal analysis. The above outlined processes offer a robust and consistent way to describe the content of video datasets, and provide a powerful environment for further analysis. This new abstract environment of data summaries is a first step toward complex scene understanding, behavior comparisons, and information dissemination.

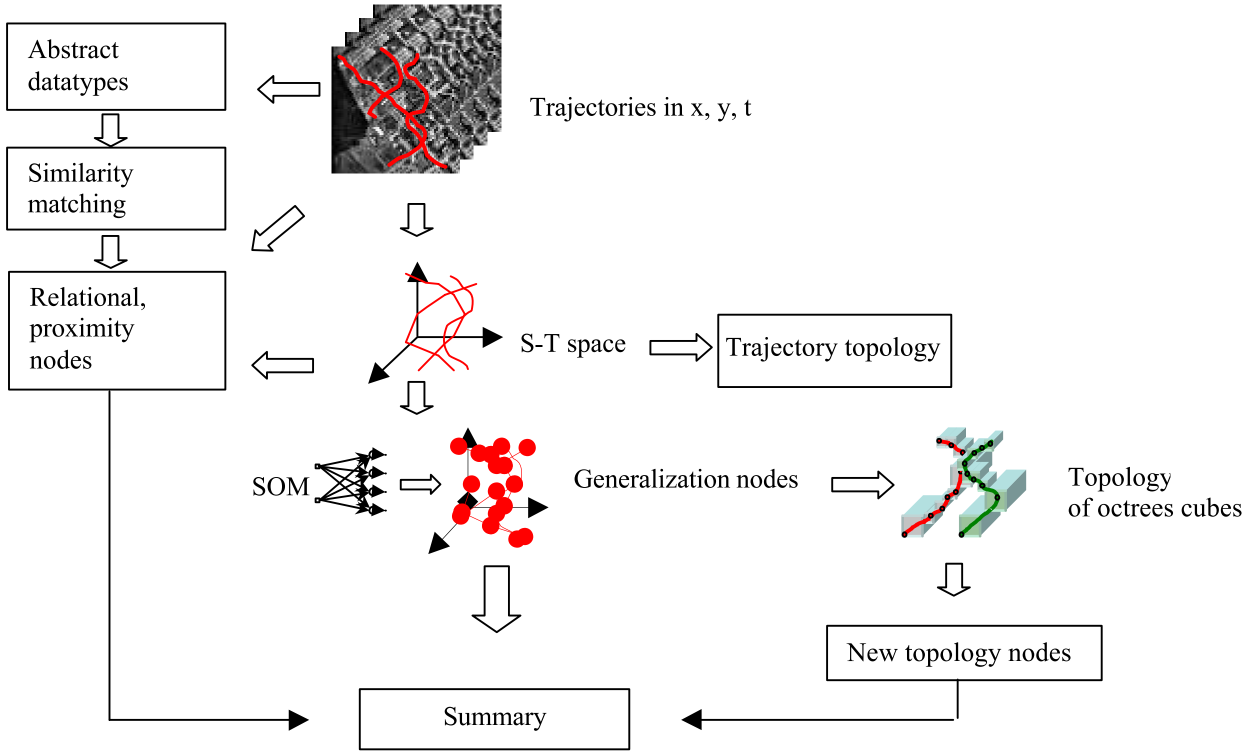


Figure 1. Outline of the proposed approach

3. SINGLE OBJECT TRAJECTORY ANALYSIS

3.1 Spatiotemporal space

Video frames define a spatiotemporal space (S-T cube) where each frame is registered at the time *(t)* of its acquisition. Assuming a fixed camera, the spatial dimensions *(x,y)* of the cube coincide with the image coordinate systems of each individual frame. In that sense, individual frames pile up on top of each other to form the 3-D S-T cube. The movement of an object within this cube is manifested as a set of classified points moving over time. Treating the S-T cube as a near continuous representation of reality, the trajectory of an object defines a linear feature within this 3-D space, by connecting all positions of the same object over time. The trajectory begins at point (x^i_0, y^i_0, t^i_0) and ends at point (x^i_n, y^i_n, t^i_n), where (x^i_0, y^i_0) are the image coordinates of object i at the time t^i_0 that it first appears in the video stream, and (x^i_n, y^i_n) are the corresponding coordinates at the time t^i_n that it moves outside the video stream. At the same spatial and/or temporal increment the information that we are interested is not always similar. We introduce artificial nodes on the S-T trajectories that form the points where change is more eminent. These nodes provide representative placements of the moving objects and their estimation is very critical. The introduction of more (or less) nodes produces summaries of higher (respectively lower) resolution. We examine the node selection problem separately in one and multiple objects.

3.2 Single Object Trajectory Analysis using SOM

The self-organizing map (SOM) algorithm (Kohonen, 1982; Kohonen, 1997) is a nonlinear and nonparametric regression solution to a class of vector quantization problems, which is used as the method for information abstraction. The SOM belongs to a distinct class of artificial neural networks (ANN) characterized by unsupervised and competitive learning. In this iterative clustering technique, cluster centers-nodes are spatially ordered in the network space $\Re_N$ in order to represent the input space $\Re_I$,. The objective of the SOM is to define a mapping from $\Re_I^m$ onto $\Re_N^d$ where $m \geq d$. The goal of competitive learning is to reward each node that optimally satisfies a similarity measure between a given input vector (x, y, t), compared against all other nodes. The single spatiotemporal trajectory is represented by a large number of sequential points in S-T domain and forms the input space. Under this procedure the network space is formed by selection of representative nodes that model abstractly the input space.

3.3 Expected behavior

The S-T traces of moving objects offer a powerful approach to the solution of behavioral tasks, as they represent the attitude of the objects in time. For a specified path of interest, where similar types of objects move, we conclude similarities of behavior. The projection of the S-T traces in (x, y) plane shows only the spatial trajectories of the objects involved in the scene. These trajectories are temporally registered in a relative manner by computing only the temporal difference between their projected S-T nodes. After a sufficient number of similar objects cross the examined section, we register each trajectory in unanimous prespecified temporal instances that occur by averaging the node coordinates of the traces involved. The collection of trajectories and registered nodes on the path of interest, defines a 3-d surface of spatio-temporal behavior in our (x, y ,t) graph. When a new object enters the area of interest, the expected behavior test is applied by investigating whether its trajectory falls within the defined the 3-D surface. Any trace out of the 3-d test curve is considered suspicious of some kind of abnormality, and temporal zooming takes place. In order to actualize the procedure of creating the test environment in a simpler mode, we first measure the distances between all S-T trajectories of the objects that are used to create the expected scheme in time. These distances are projected to the averaged nodes. Then, we expand the spatial occupancy of the node in a disk so that it contains all the points of the object's trajectories in that instant. Therefore, when a new object takes the path of interest we compute its spatial coordinates (x, y) in node time instances ti, and check if they reside inside the defined disk. If not according to the extend and duration of the inconsistency we provide additional nodes in our summary. (fig. 2)

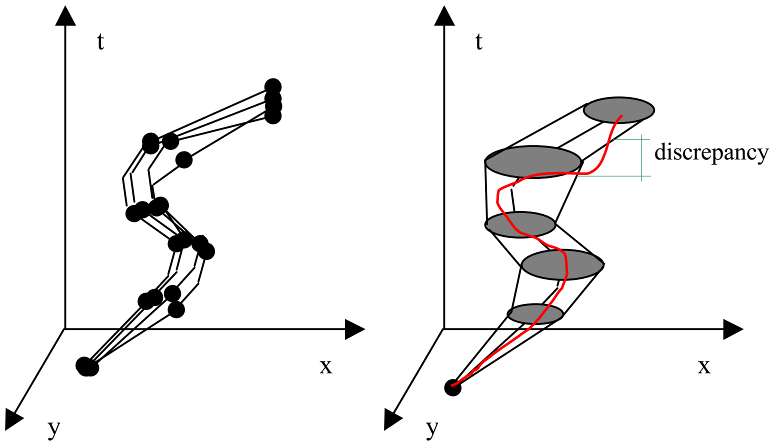


Figure 2. Expected behavior design and testing.

4. ABSTRACT DATATYPES

Incorporating both the spatial and temporal dimensions of information density, we can define scaled abstract descriptive data types that contain different volume of information. In this abstract space, significant information, its dissemination and comparison procedures are simpler to analyze. Each movement descriptor refers to the moving object and is defined by a set of attributes that describe points of interest and their connecting route referenced in time. The connecting route is the base element of the data type upon which spatial and temporal scaling is operated in order to acquire the desired density of information. The spatial component is determined by several techniques of varying complexity. Polynomial geometric representation is one technique, while coordinate definition in every (d) distance is simpler to compute. Registration to the image features or segmented regions applies when we have a classified background with specified regions. Finally, the routes may participate with no more significance than as connections of points of interest with no geometric significance. Points of interest (starting, end, and intermediate points) include selected points where a discrepancy is expected to happen (e.g. stop signs) and decision points where an object has more than one choice for continuing its course. The time stamp is either absolute in every point of interest or relative for each route.

4.1 Scaled similarity matching

Based on the above definition of the movement descriptors, we define scaled similarity matching for objects that move from a start point to an end point. According to the volume of detail we wish to compare their relation with, we divide the comparison of behaviors in the following levels for spatial and temporal zooming.

Spatial zooming

- Actual geometric trajectory: The trajectory is represented by a polynomial of 2^{nd}–3^{rd} order or (x, y) coordinates in smaller or larger distances upon the trajectory. When we are not concerned for the actual trajectory, we replace it with the background road that the trajectory was formed upon. This process requires that we have analyzed the scene of its spatial components, using any of the known methods.
- Directional consideration: The route is represented by topology and direction components between the points of interest.
- Decisional consideration: The actual route remains insignificant, while we focus on the decision that the moving object makes in the predefined decision points.
- In the most abstract space only the start-target points are recorded and processed usually for statistical evaluations.

Temporal zooming. Time is processed under different conditions according to the application. Relative time is considered when similarity matching is performed while absolute time is essential for certain types of spatiotemporal queries. Zooming is accomplished by defining time in values with more or less uncertainty. (fi. temporal distance between two points should be 10 to 15 minutes).

By using SOFM we do not differentiate the temporal and spatial dimensions while by this procedure, comparison is made distinctly in the routes and the referenced time. SOM produces nodes for the formation of summary and we can define a framework under which similarity matching will occur. Using the data types defined above we have more control on the environment of the scene, so that identification of points of interest is more focused and selective by the user.

5. MULTIPLE TRAJECTORY ANALYSIS

The consideration of multiple objects brings forward the need to address two issues. First, we have to select specific time instances for our video summaries using independent nodes from multiple trajectories. Second, we have to consider the introduction of additional nodes when taking into account the proximity of two or more trajectories. One can easily understand that the set of temporal coordinates of the nodes describing the path of object i and these describing the path of another object j may be totally disjoint. According to the density and the dissimilarity of the S-T trajectories and the corresponding nodes, we can follow different strategies for merging a complex scene summary under one time reference:

- An obvious solution is to use the nodes from all S-T trajectories and reference all moving objects to every estimated node. This results in a relatively large summary, depending on the number and behavior of the objects.
- Another solution is to define nodes according to the most demanding moving object and project all other node sets to this dominant set. If the behaviors of scene objects are incompatible then the other objects are not efficiently represented. One way to overcome the overcrowded node collection is to group sets of nodes over a minimal increment ät and identify an average temporal position t_{av} to substitute individual nodes. This allows us to minimize the number of nodes and the complexity of the produced summary.
- Furthermore, by using the SOM we can obtain a "medium" estimation of node selection upon the whole set of moving objects. This gives a summary of the whole scene, which does not explicitly depict behavior information for single objects. On the other hand, it provides a technique to unravel mass behavioral attitudes in the scene. For instance if a police car enters the scene, the majority of the moving cars tend to slow down.

6. ADDITIONAL NODE SELECTION

The construction of the product summary of the original video stream is based on the selected nodes. In addition to the information that each spatiotemporal trajectory carries, more information is inherent in a dynamic scene. When more than one object are present in a scene, their relations define significant regions of interrelation. Therefore, we introduce several procedures that focus on spatial and/or temporal sections that need to be included in the summary or monitored in larger zooming than the SOM analysis provides. According to the dominant features we investigate and provide the following procedures for node addition.

- Geometry nodes: The spatial dimension of the routes included in the video stream, is represented in a similar manner as the S-T trajectories, through a 2-dimensional SOM. The corresponding nodes depict a generalized abstract representation of the route focussing in the regions where geometric complexity is eminent.
- Proximity-Relational nodes: Throughout the scene we are highly interested in the behavior relations between objects, which approximate each other. We can then monitor closer their relation and investigate further reasoning association and dependence. Therefore, we introduce nodes termed as "proximity nodes" when the distance between two or more objects drops under a predefined threshold. This proximity is either computed as the

Euclidean distance between the related objects or as the distance that takes into consideration the actual path between the objects.

- Reasoning nodes: Furthermore, by adding restrictions on the S-T cube, we provide a reasoning framework within which we wish to monitor more closely the behavior of objects. These restrictions are either purely spatial and include regions and paths of interest, or spatiotemporal and include more complex relations like one way roads, acceleration and deceleration limitations etc. In the S-T cube we construct new spatiotemporal regions and when a trajectory passes through these regions, we add mandatory nodes in order to include this information in the summary. These types of nodes are termed as "reasoning" nodes. (Fig. 3)

The supplementary information is described either as additional nodes on the summary or as the time reference when the actual video should be examined in order to further investigate the circumstances.

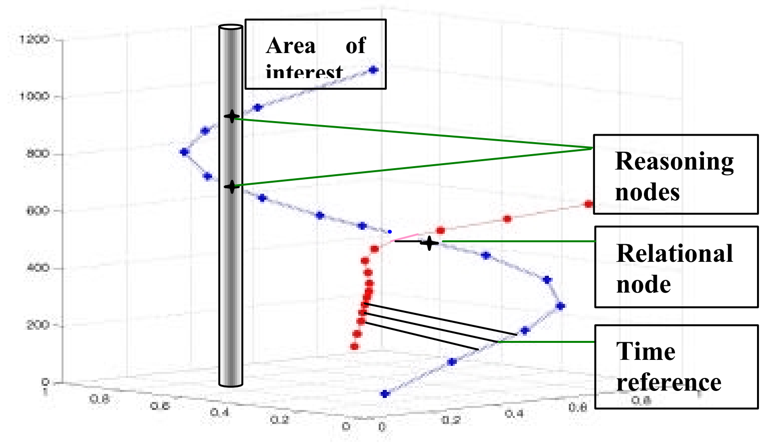


Figure 3. Mandatory nodes for multiple S-T trajectories

7. TOPOLOGY

7.1 General topologic relations

Topologic relations that occur within the scene between objects, are of high importance in order to monitor, model and detect behavioral aspects. Our summary includes the important information that describes movement, based on each object separately and taking into consideration important areas and proximity thresholds. The relative position of the objects and the dynamic change of their topology, is something extremely essential to be embodied in the summary. Towards this focus, we capture the topologic relations by extending the static models into dynamic, in order to include them to the product summary. The elements that we examine in order to provide the basic framework for relations between objects are focusing in relative topology, distance and direction between moving objects. On the first level, the topology relation between two objects is assumed to be always a "disjoint" one as it refers to discrete and solid objects as vehicles. This topology is dynamic as it changes according to the relational movements of the objects. Distances are measured either as minimum straight euclidean distances between the objects or upon the path of movement as spatial trajectories. Therefore, the absolute value of proximity differentiates from the actual spatiotemporal proximity. Proximity is a key component under which the topology relations should or should not be examined. When two objects are relatively far from each other then there is no need for topology checking. Finally, the direction between two objects is computed and is referenced to a cardinal model. In order to process and capture the dynamic nature of the topology, we compute the following relational components between two objects. Change of distance in time, relative speed and change of direction in time. By setting thresholds in the described components, we flag the time instances when their values exceed the thresholds. Dynamic topology is defined as change in relative values or as change in placement upon a predefined framework. A topology framework scheme is introduced and is used for experimental purposes. The changes in topology under the proximity restrictions define a new set of nodes, which are added in our summary model. (fig. 4)

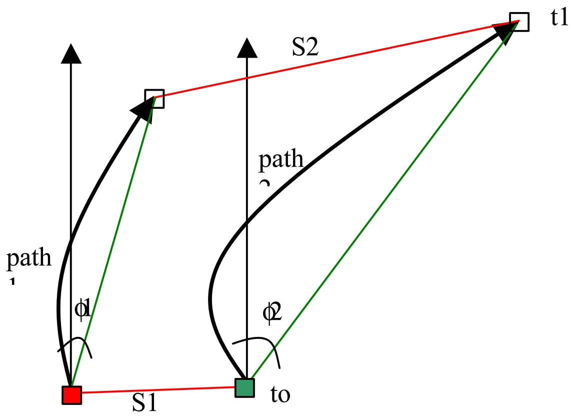


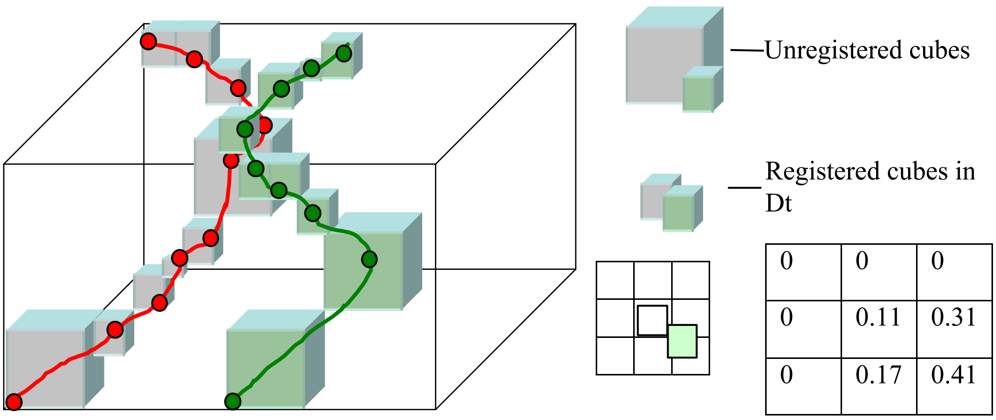
Figure 4. Dynamic topology of moving objects

7.2 Topology of s-t trajectories

By examining the s-t space, we conclude that the trajectories form topologic relations that characterize the objects they represent. In order to simplify the procedure of relating complex curves, we design octrees that model the s-t trajectories into representative minimum bounding cubes. The design is formed upon the generalized SOM node space. According to the octree structure, decomposition process of the data volume is carried out iteratively in a step by step fashion by dividing the space into eight disjoint cubes with the aim of eventually reach a resolution criterion. If any of the cells is homogenous, that is the cell lies entirely inside or outside the object or satisfies the criterion the sub-division stops. If the cell is heterogeneous then it is sub-divided further into eight sub-cells until the prespecified resolution criterion is met. The information on the original video stream is compactly represented and the leaf nodes represent the minimum resolution segment. In the node space the design includes:

- The input data is the set of node points of (x, y, t) space.
- Decomposition of S-T space is based on standard octree decomposition as described above.
- The subdivision is terminated when each cube has at most one node.

In order to examine the relation between two objects we define the base of the comparison either in a predefined time increment or in a time increment equal to the height of the cube. We then focus on the cubes embracing the s-t trajectory segments. Because different trajectories are modeled by different volumes of tree cubes, we register the larger cube by reducing it to the smaller and more demanding one. In this case, the minimum boundary cubes can be related with known topological models and process their change in time through the product matrices. (fig. 5)



0	0	0
0	0.11	0.31
0	0.17	0.41

Figure 5. Octree decomposition cubes for two trajectories, topology pattern and product matrix.

Because the actual trajectory is not used in the process we have a deterioration in the relational analysis which is compensated by the fact that the representative cubes are simpler to implement. Inside the cube the direction of the trajectory can be diverge. Nevertheless, the gain from this modeling is dual, as generalization is accomplished both in the spatial domain and in the temporal dimension. The topology of the s-t trajectories provides valuable information, as it depicts significant changes in the relations of the examined objects and it forms the basis for further processing.

Euclidean distance between the related objects or as the distance that takes into consideration the actual path between the objects.

- Reasoning nodes: Furthermore, by adding restrictions on the S-T cube, we provide a reasoning framework within which we wish to monitor more closely the behavior of objects. These restrictions are either purely spatial and include regions and paths of interest, or spatiotemporal and include more complex relations like one way roads, acceleration and deceleration limitations etc. In the S-T cube we construct new spatiotemporal regions and when a trajectory passes through these regions, we add mandatory nodes in order to include this information in the summary. These types of nodes are termed as "reasoning" nodes. (Fig. 3)

The supplementary information is described either as additional nodes on the summary or as the time reference when the actual video should be examined in order to further investigate the circumstances.

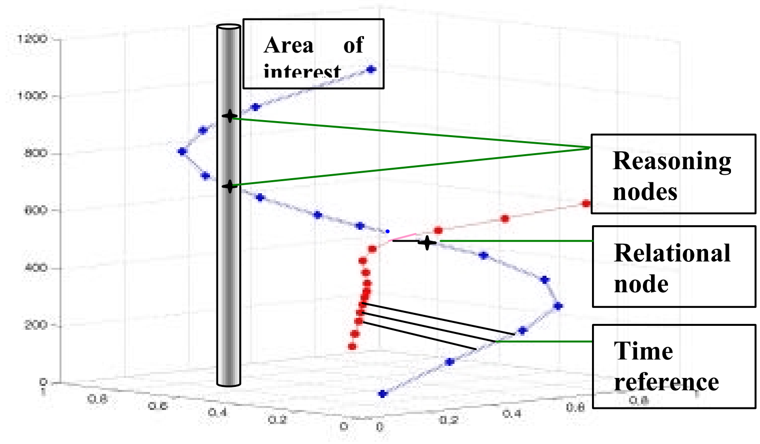


Figure 3. Mandatory nodes for multiple S-T trajectories

7. TOPOLOGY

7.1 General topologic relations

Topologic relations that occur within the scene between objects, are of high importance in order to monitor, model and detect behavioral aspects. Our summary includes the important information that describes movement, based on each object separately and taking into consideration important areas and proximity thresholds. The relative position of the objects and the dynamic change of their topology, is something extremely essential to be embodied in the summary. Towards this focus, we capture the topologic relations by extending the static models into dynamic, in order to include them to the product summary. The elements that we examine in order to provide the basic framework for relations between objects are focusing in relative topology, distance and direction between moving objects. On the first level, the topology relation between two objects is assumed to be always a "disjoint" one as it refers to discrete and solid objects as vehicles. This topology is dynamic as it changes according to the relational movements of the objects. Distances are measured either as minimum straight euclidean distances between the objects or upon the path of movement as spatial trajectories. Therefore, the absolute value of proximity differentiates from the actual spatiotemporal proximity. Proximity is a key component under which the topology relations should or should not be examined. When two objects are relatively far from each other then there is no need for topology checking. Finally, the direction between two objects is computed and is referenced to a cardinal model. In order to process and capture the dynamic nature of the topology, we compute the following relational components between two objects. Change of distance in time, relative speed and change of direction in time. By setting thresholds in the described components, we flag the time instances when their values exceed the thresholds. Dynamic topology is defined as change in relative values or as change in placement upon a predefined framework. A topology framework scheme is introduced and is used for experimental purposes. The changes in topology under the proximity restrictions define a new set of nodes, which are added in our summary model. (fig. 4)

DATABASE GENERALIZATION: CONCEPTS, PROBLEMS, AND OPERATIONS

Wanning PENG
ESRI
380 New York St., Redlands, CA 92373, USA
wpeng@esri.com

KEY WORDS: Database Generalization, Multi-scale Database, Modeling.

ABSTRACT

Generalization in GIS contains two main aspects: a) manipulating the geometric and thematic descriptions of spatial objects and their relationships with respect to certain changes of the underlying application model, and b) mapping/transforming the *digital description* of spatial objects and their relationships into *graphic description,* which is confined to graphic legibility and cartographic principle. The first aspect is called *database generalization* in literature, and the second is referred to as *view generalization*. This paper deals with some important issues of database generalization, including concepts, problems, and operations.

1. INTRODUCTION

Although human experts have not yet been able to sum up, and generalize, the practices of map generalization to develop a " generalization theory" , the issue has been studied by a number of authors. Several conceptual generalization models have been proposed, such as the Ratajski model, Morrison model, Nickerson and Freeman model, McMaster and Shea model, Brassel and Weibel model. In his review of these models, McMaster (1991) identified the Brassel and Weibel model as the best for implementing an expert system.

These models, however, were developed based on the long tradition and practice of multi-scale map production. In recent years, research has been paying more and more attention to model-oriented generalization and database generalization. Examples include Muller 1991; Richardson, 1993; Muller et al., 1995; Weibel, 1995; Peng and; Molenaar, 1996; van Smaalen, 1996; Peng and Tempfli, 1996; Peng, 1997. However, unlike map/view generalization where the objectives and problems are clear and commonly understood, research in database generalization has largely focused on developing solutions for specific problems, neglecting the general picture (Weibel, 1995), in particular, the objectives and scope, the requirements and problems, and the relationship with graphic representation.

Having understood the above problems, this paper attempts to formally define the subject of database generalization, by focusing on the related concepts of geo-data and GIS. This includes defining the objectives of generalization in GIS; defining the general principles of database generalization; identifying and elaborating database generalization problems and operations. Finally, it points out some key issues to be addressed in realizing an automated database generalization.

2. RELEVANT ASPECTS OF GEO-DATA AND GIS

This section sets up the foundation of this research by looking into relevant aspects of geo-data and GIS. The aim is not to carry out a review of general GIS concepts; rather, it provides a discussion on some key aspects that will have important effects on defining the new concept and strategy of generalization.

2.1. Geo-database and Data Modeling

A *geo-database*, or database for convenience, is central to a GIS. It is the digital form of a *geo-spatial model* which is a replica of some portion of the planet earth (Pilouk, 1996). A database is not only a collection of data, but also contains relationships among data elements, and the rules and operations that are used to change the state of the data elements.

While a database is the core of a GIS, the underlying modeling process is the essential step that brings about a meaningful database for an application. This process, known as data modeling (Frank, 1983; Peuquet, 1984), aims at producing *representation schemes* for real world phenomena, that later can be implemented in a computer environment, and be used for building a database. It includes three steps: conceptual data modeling, logical data modeling, and physical data modeling. Among these three processes, only the relevant aspects of conceptual data modeling will be discussed, as the concept of generalization is somewhat independent from the other two processes.

2.1.1. The Need for Conceptual Data Modeling

The real world is complex. It is not possible (and not necessary) for a spatial model to accommodate all the aspects of the reality. Spatial models should always be subject to interpretations of different disciplines for particular applications, and should be constructed at such a complexity level that the modeled phenomena, as well as underlying processes, are meaningful and best understood (Muller et al., 1995; Weibel, 1995). Higher complexity implies the result of more detailed information, but this does not necessarily mean that such would be more adequate for a particular application. Moreover, maintaining such details in a database would lower efficiency and may create difficulties in spatial analysis, decision-making, geometric operation, storage, updating, and maintenance. Hence, before a database can be constructed, one has to determine what aspects of reality are relevant to his/her application(s). This includes specifying types of objects, the relationships among them, and how they should be represented.

2.1.2. Spatial Object and Object Types

A *spatial object* is a real world object that contains both thematic and geometric information, and is normally represented through thematic and geometric descriptions in a GIS.

Objects in a spatial model, that have common patterns of both state and behavior within the framework of an application, may be grouped into classes to form *object types*, and object types in turn may be organized into super classes to form *super-types*, and so on. Object types, together with classification and aggregation hierarchies are important aspects in *semantic data modeling* and play a critical role in defining the concept of generalization in GIS.

2.1.3. Classification Hierarchy

Object types and super-types can then be organized into a hierarchical structure called *classification hierarchy* (Smith and Smith, 1977; Thompson, 1989; Hughes, 1991; Molenaar, 1993). This hierarchical structure reflects a certain aspect of data abstraction. The lower levels in the hierarchy correspond to lower abstraction levels and thus will result in more complex data, including both thematic and spatial aspects, whereas the higher levels correspond to higher abstraction levels, thus will lead to less complex data. In this sense, specifying an object type implies, to a certain extent, determining the abstraction/complexity level of a geo-spatial model. For instance, assuming *Transportation* is a super-type of types *Railway*, *Road*, and *River*, a model that employs the type *Transportation* is usually less complex than another model that employs the types *Railway*, *Road*, and *River*. However, these two models have some *inherent relationship* due to the *IS-A* relationship between object types *Transportation* and *Road* (and *Railway* and *River*). This relationship makes it possible to transform the more complex model to the less complex one, and this *transformation process* is, in fact, one of the major aspects in database generalization.

Because object types at different levels in a classification hierarchy correspond to data of different complexity, changing the object types of an existing model to the ones at a higher level in the same hierarchy, would mean transforming the model from a lower abstraction level to a higher abstraction level. Such a transformation will lead to a generalization process to take place, in order to convert instances of the sub-types to instances of the super-types.

In addition to object types, domains of some attributes of an object type may also be associated to a classification hierarchy, such as the land-use property of object type *Cadastral-parcel*.

2.1.4. Aggregation Hierarchy

Another important structure is the *aggregation hierarchy* (Hughes, 1991; Molenaar, 1993). This structure shows how lower-order object types that belong to different classification hierarchies are combined to form a higher-order object type. For example, object type *Building-block* is a combination of the types *Building* and *Garden*. In other words, *Building* is part of *Building-block*, and so is the *Garden*.

In this article, a higher-order object type in the hierarchy is called *aggregation-type*, whereas an object type that is part of the aggregation-type is called *component-type*. Accordingly, an instance of the aggregation-type is referred to as an *aggregated-object*, and an instance of the component-type is regarded as a *component-object*. An aggregation-type can be the component-type of another (super) aggregation-type.

Similar to the classification hierarchy, this structure also reflects some aspect of data abstraction. The aggregation-types in the hierarchy correspond to higher abstraction levels and thus will result in less complex data, while the component-types correspond to lower abstraction levels and hence will result in more complex data. This implies that replacing the component-types in a model with their aggregation-type will result in transforming the model from a lower abstraction level to a higher abstraction level. Such a transformation may require a generalization process to take place in order to construct instances of the aggregation-type using the existing objects of the component-types.

Determining a right object type for an application can be seen as a process of choosing a proper geographic unit that represents at which abstraction level a geographic structure should be understood (Molenaar, 1996). Choosing an adequate object type, thus a proper complexity, for a GIS application, is comparable to the work of selecting a proper map scale in the analogue environment. However, selecting a proper map scale is often rather confusing in practice, as one can hardly explain why a particular scale was selected for use in solving his/her problem. In fact, in many cases, the user was forced to use what is available from surveying and mapping agencies, not what is more suitable for solving his/her problem.

2.2. Spatial Resolution and Thematic Resolution

A spatial model represents some real-world phenomena at a certain abstraction level. When the model is in the form of a database, this complexity level can be indicated by means of *resolution*. The resolution, together with data quality, serves as a specification for the evaluation and usage of a database.

Three types of resolutions can be perceived concerning objects and a database. They are thematic resolution, spatial resolution, and temporal resolution. Temporal resolution and related aspects are not discussed in this study.

2.2.1. Thematic Resolution

Thematic resolution is a specification that indicates the thematic abstraction level of the objects in a database. It includes five aspects:

- level in which an object type is located in its associated classification hierarchy;
- level in which the associated domain of an attribute of an object type is located in its associated classification hierarchy;
- level in which an object type is located in its associated aggregation hierarchy;
- number of objects contained in an object type;
- number of attributes contained in an object type.

These five aspects, and the number of object types that a database contains, determine the thematic resolution of the database. Thematic resolutions may be ranked, but cannot be measured.

2.2.2. Spatial Resolution

The spatial resolution of an object type is a specification that indicates the spatial abstraction level of the object type in a database. It comprises four aspects: a) type of geometric description, b) minimum object size, c) minimum space between two adjacent objects of the same type, and d) minimum object’ s details. Note that the last three aspects of spatial resolution are different from the three criteria for ensuring legible visualization in map/view generalization.

- *Geometric description type*: in a vector approach, the geometry of a spatial object can be described using 2D area type, 1D line type, or 0D point type (Molenaar, 1991). Higher dimensions correspond to higher spatial resolutions.
- *Minimum Object Size* that a database can contain: minimum size for area objects, or minimum length for line objects. Only objects that are larger or equal to this threshold can be described and contained in the database (Figure 1a). In other words, the database is suitable for applications that are not interested in objects smaller than the threshold. However, this criterion should not be applied to the objects that are described as points in the database, because it would be meaningless.
- *Minimum Space* between two adjacent objects of the same type: minimum space by which a database can distinguish two adjacent objects of the same type. Two adjacent, but geometrically disconnected, objects of the same type become one larger object if the space between them is smaller than this threshold (Figure 1b). This implies that the database is suitable for applications that are not interested in object spacing smaller than the stated threshold. For instance, bus navigation may be not interested in narrow alleys smaller than two meters in width, while motorbike navigation is.
- *Minimum Object's Detail* that a database can contain: local spatial details of an object cannot be described and contained in the database, if smaller than this threshold (Figure 1c). This means that the database cannot provide spatial information of an object at a detail level higher than that indicated by this threshold.

These three aspects of spatial resolution apply to an object type, rather than to the entire database, and may take different values for different object types in the same database (which is also a common practice in data acquisition and traditional map generalization).

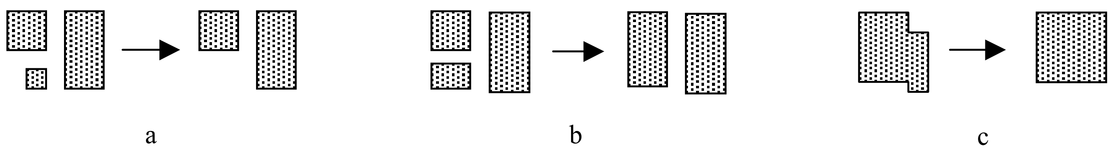


Figure1. Examples of spatial resolution components and their effects.

2.2.3. Relationship between Spatial Resolution and Thematic Resolution

Although these two types of resolutions have no explicit link, they do have an implicit relationship. For example, higher thematic resolution tends to lead to higher spatial complexity (when increasing thematic resolution, geographic units that remain homogeneous tend to become smaller in size). If an application requires a database of higher thematic resolution, then, the spatial resolution of the database should be also higher. If the thematic resolution should be reduced for another application, then probably the spatial resolution will also need to be readjusted to a lower level. For instance, if an application needs to work at the level of *parcel*, then the required spatial resolution is likely higher than that required for another application that works at the level of *district*. Spatial resolution and thematic resolution also have impact on the selection of map/view scales. Map/view scales, on the other hand, reflect different spatial and thematic resolutions (Peng, 1997).

2.4. The Graphic Representation of a Database - Views

A database’ s view is a graphic representational form of the database. It is concerned with the graphics, and thus, is scale-dependent. Database contents, together with communication rules and graphic constraints, determine the appearance of a view, and the view in turn reflects the nature of the database, but should not change the database. This implies that while a change of the database may lead to an update of its associated view(s), the design, processing, and modification of a view should not cause any change over its associated database.

The distinction between a database, and its views, naturally leads to the introduction of the, so-called, *database-objects* and *view-objects*. A database-object is an object presented in a database, and a view-object is the graphic mapping of one or more database-object(s). While its graphic mapping may take various forms, a database-object should not have more than one version within one database.

3. GENERALIZATION AS A DATABASE PROCESS   DATABASE GENERALIZATION

Having understood the concept of a database and the underlying modeling process, and the concept of views and the relationships with the associated database, generalization in GIS can be regarded as a transformation process with the following two objectives (Peng and Molenaar, 1995):

- *to* ***derive*** *a new (digital) database with different (coarser) spatial/thematic/temporal resolutions from existing database(s), for a particular application;*
- *to* ***enhance*** *graphic representations of a database or part thereof, when the output scale cannot accommodate the data set of interest, for visualization purposes.*

While *view generalization* is the process corresponding to the second objective, the first objective relates to the aspect of changing the abstraction/complexity level of a spatial model, and the transformation process due to this change is what we called *database generalization*.

3.1. General Generalization Principles

The following general principles can be defined for database generalization, based on the concepts discussed in section 2. These principles are to be used to define generalization problems and guide development of solutions.

- Database generalization transforms an existing database only if the user has introduced a new conceptual data model, which will lead to a database of lower resolution.
- The underlying conceptual data model, not map scale, determines what object types and which instances of these object types, should be contained in the generalized database.
- The underlying conceptual data model, not graphic constraints, determines the resolution of the target database.
- The same phenomena should be described using the same thematic resolution through out the entire data model.
- Topological constraints are critical and any generalization process should be subject to such constraints. These constraints include: 1) an object must not move across the boundary of another object, and 2) an object must not overlap with another object, in a generalization process.

3.2. Elementary Problems and Operations

This sub-section identifies the problems in database generalization, by examining the processes in thematic and spatial resolution transformation. In order to facilitate the description, the term *adjacency* will be used to describe the adjacency relationships among objects that are geometrically connected to and/or disconnected from each other, and the term *adjoining* is employed to describe the adjacency relationships among objects that are geometrically connected to each other.

3.2.1. Changing Thematic Resolution

Problems concerning changing thematic resolution are related to the five aspects of thematic resolution, as well as the number of object types that a database contains.

- Extracting application-relevant object types: the new conceptual model may omit some of the object types presented in the existing model. In this case, only objects that belong to those types presented in the new model should be selected into the new database. Such a thematic ***selection*** process can be described as follows:

Given a set of object types $T = \{ t_1, t_2, ..., t_n \}$, and a subset $Ts = \{ t_{s1}, t_{s2}, ..., t_{sm} \} \subseteq T$, select only objects that belong to type $t_{si} \in Ts$ ($1 \leq i \leq m$).

- Changing classification level of an object type: when replacing an existing object type with a super-type, objects of the sub-type need to be converted into instances of the super-type. Similarly, if the domain of an attribute of an object type is replaced by the one located at a higher level of the same classification hierarchy, then attribute values need to be reevaluated accordingly. Such a process is called ***universalization***, and is equivalent to the *generalization* operation in semantic data modeling.

- Changing scope: this is a thematic ***simplification*** operation related to " the number of attributes of an object type." Some of the attributes of an existing object type may not be relevant to a new application, thus, may not be specified in the new data model. For example, for an application, object type *Road* may have attributes *number-of-lanes* and *traffic-volume*, whereas for another application these may not be relevant. The process can be defined as follows:

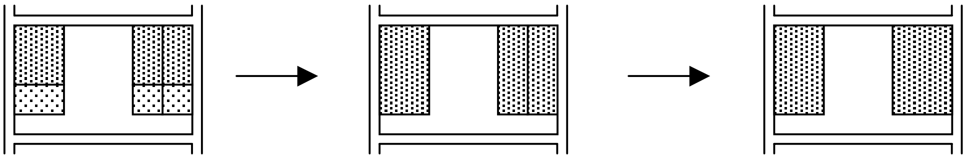
Given a list of attributes $List(Ti) = \{ a_1, a_2, ..., a_n \}$ of object type Ti, remove a subset of attributes $SubList(Ti) = \{ a_{s1}, a_{s2}, ..., a_{sm} \} \subset List(Ti)$, where $m < n$.

- Changing aggregation level: when replacing a sub set of object types with an aggregated-type, existing objects of the component-types need to be aggregated to form aggregated-objects of the aggregation-type, based on their geometric and semantic relationships. The process is similar to the *aggregation* operation in semantic data modeling, and is called ***combination***.

When aggregating the component-objects, spatial relationship among them plays an important role, because only adjoining objects should be aggregated to form an aggregated-object (Figures 2a). In some cases, semantic relationship is also required. For instance, when aggregating farm yards and fields into farms, only the farm yards and fields that belong to the same farmer should be aggregated (Richardson, 1993; Molenaar 1996).

- Reducing object numbers of a specific object type: this problem exists for two reasons. One is that an application may be interested only in objects that have certain property values, e.g., those parcels of which the land-use is *residential*. In this case, a ***selection*** operation is required in order to select a subset of objects having particular geometric and/or thematic properties.

Another reason is related to the spatial boundary of a homogeneous unit. In many cases, an application may require those adjoining objects of the same type to be described as a single object. This requirement often comes after changing the classification/aggregation level of an object type. Two objects of different types, for instance, may become instances of the same type after the process. In many cases it makes sense to replace them with a larger homogeneous object if they are adjoining, not only for the efficiency of the database, but also because the environment has been changed in which the two original objects exist as two separated ones.



a b

Figure 2. An example of combination operation and homogenization. operation

The process that creates a homogeneous object by merging a subset of adjoining objects of the same type is called ***homogenization***. Figure 2b shows an example. An application may introduce extra criteria to further restrict which objects can be merged.

3.2.2. Changing Spatial Resolution

Similar to changing thematic resolution, problems concerning changing spatial resolution are related to the four aspects of spatial resolution. These problems are straightforward.

- Changing geometric description type: a ***collapse*** operation that changes the geometric description of a spatial object from area type to line or point type, or from line type to point type.

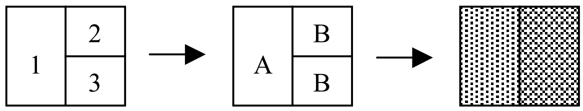
- Filtering out small objects: when the *minimum object size* of the spatial resolution is changed to a larger one, those objects of which the sizes are smaller than the required minimum value, should be deleted. Such a ***deletion*** operation only concerns the geometric property of an object, which is different from the case in map/view generalization where the thematic property of the object is also taken into consideration.

- Merging close objects: changing the *minimum space* of the spatial resolution to a larger one, or some geometric operations, may lead to a situation in which some of the objects are " too close to each other" . In this case, an ***aggregation*** operation is required: if the space between two disconnected, but adjacent, objects is smaller than the required minimum value, aggregate them to form a new object of the same type without moving any of them (Figure 1b). Notice that this aggregation operation, motivated by spatial resolution transformation, is neither the same as homogenization nor the same as combination.

- Filtering out small spatial details: when the *minimum object s detail* of the spatial resolution is changed to a larger one, a geometric ***simplification*** operation may be needed. This process filters out small spatial details of an area or line object if their sizes are smaller than the required value (Figure 1c). Unlike in map/view generalization, this geometric ***simplification*** operation does not consider the thematic property of an object.

3.2.3. Context Transformation

Context transformation, or changing theme, is a ***reclassification*** process aiming at creating instances of a new object type using objects of another type, of which one of the attributes defines the theme of the new type. For instance, object type *Parcel* may include an attribute *land-use*. If *Land-use* is an object type in the new model, it is then possible to construct a land-use unit (instance of *Land-use*) using parcels, assuming that the land-use is homogenous in each parcel.



1, 2, 3: parcels; A, B: land-uses; : land-use unit.

Figure 3. An example of context transformation.

Context transformation, though it is not within the objective of database generalization, usually plays a role in database transformation, and may have a close link with resolution transformation. In the example described above, it is likely that a homogenization operation needs to follow afterwards (Figure 3).

3.3. Modeling Operations for Database Generalization

Generalization operations described in section 3.2 include selection, universalization, simplification, homogenization, combination, collapse, deletion, aggregation, and reclassification. Among them, thematic simplification, reclassification, and universalization apply only to the thematic domain, whereas deletion, geometric simplification, and collapse only apply to the geometric domain. Homogenization, combination, and aggregation require inputs from both thematic and geometric domains. Thematic operations may require geometric operations to follow.

Figure 4 shows how these operations are arranged into an order of execution, in which operations are arranged into groups according to the three generalization tasks discussed in section 3.2. At the group level, context transformation is independent of the other two transformations, and should be conducted first, so that other generalization processes will not affect the result. Thematic resolution transformation, on the other hand, should be executed before spatial resolution transformation. This is because thematic resolution transformation works at the level of object definition, whereas spatial

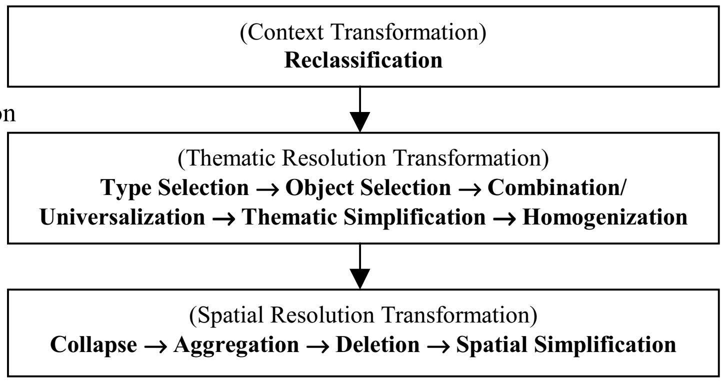


Figure 4. Modeling generalization operations

resolution transformation deals with the spatial property of objects that have been defined.

3.3.1. Operations in Thematic Resolution Transformation

Inside the thematic resolution transformation group, selection is always the first operation. This is obvious because if an object or object type is not selected, then it would be meaningless to apply any other operations to the object or object type. Combination and universalization cannot coexist for the same object type within one database transformation, as one must not replace an object type with a super-type and an aggregation-type simultaneously. However, both operations should be executed before thematic simplification and homogenization, because both of them redefine objects, and thematic simplification and homogenization are suppose to work on the target objects. Homogenization and thematic simplification do not affect each other. However, executing thematic simplification before homogenization may increase the efficiency in a homogenization process.

An important question in this modeling process is, will these operations be invoked multiple times for the same object type? The answer is *no*. There is no need to execute selection or any other operations again after applying a thematic simplification operation, for instance. Each of these operations applies, systematically, consistently, and in the strict sequential order as specified in Figure 4, to all the objects of the same type at one time, and will not generate a new problem that would require an operation to be invoked again. This is very different from map/view generalization, where a generalization problem (e.g., spatial conflict) may require several operations to be involved, and an operation can create a new problem. Notice that each problem in database generalization corresponds to exactly one operation.

The same conclusion also holds for the three transformations, i.e., spatial resolution transformation will not lead to another thematic resolution transformation, and both resolution transformations cannot lead to a situation that requires another context transformation process.

3.3.2. Operations in Spatial Resolution Transformation

Inside the spatial resolution transformation group, collapse is ordered higher than aggregation, deletion, and spatial simplification, because the last three operations depend on the geometric description type of an object. Simplification should be executed after deletion has been conducted. This is because there is no need to simplify an object if it would eventually be eliminated. However, deletion should not be conducted before aggregation is carried out, as a group of adjacent small objects may be aggregated into a single one of which the size is larger than the criterion for deletion.

Like in thematic resolution transformation, each of these operations applies to all the objects of the same type at one time, and does it systematically, consistently, and in the strict sequential order as specified. When following this manner, there will be no cases where the execution of an operation would require another operation to be invoked again, except for spatial simplification. Aggregation, deletion, and spatial simplification cannot cause the geometric description type of some existing objects to be changed to another type. The geometric description type for an object type is defined at design time, and should be applied to all the objects of the same type. Deletion operation removes an object entirely, thus will not create a case that would require aggregation or spatial simplification to follow.

Spatial simplification is the only operation that may cause aggregation or deletion to be repeated. For example, an object may become too small after removing some of its spatial details. However, this does not create a complex situation as in map/view generalization, because these three operations can be executed repeatedly using the same order, until no object need to be processed.

4. DISCUSSION

In an attempt to formally define the subject of database generalization, this paper adopted a top-down approach. Instead of trying to extract and formalize knowledge from products and experiences of manual map generalization, it first studied related concepts of geo-data and GIS, which explained the need for generalization, and set up the foundation for understanding generalization problems within the context of a GIS.

Based on these concepts, it then defined the objectives of generalization in GIS. The objectives, together with the concepts of databases and views, set up a framework for defining the general principles of database generalization, and for identifying and categorizing elementary generalization problems.

The paper continued by elaborating the generalization problems and operations (as solutions to the problems) with respect to the general principles. Finally, after having formally defined the generalization problems and conceptually introducing the solutions, it went one step further by looking into the issue of modeling generalization operations in a database transformation process.

A number of issues need to be addressed in realizing an operational automated database generalization. These include:

- an adequate supporting conceptual data model that provides a description of spatial objects, and the topologic relationships among them. Such a data model is essential in rule translation, spatial analysis, and the implementation of generalization operations.
- a proper mechanism with which users can dynamically specify different requirements.
- algorithms that actually perform spatial analysis and transformations.

ACKNOWLEDGEMENTS

The author is grateful to those who have given comments and suggestions to this research.

REFERENCES

Frank, A. (1983) Data structures for land information systems - semantical, topological, and spatial relations in data of Geo-Sciences (in German), *PhD Thesis*, Swiss Federal Institute of Technology, Zurich, Switzerland.

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

Molenaar, M. (1991) Terrain objects, data structures and query spaces, in *Geo-Informatik*, (ed Schilcher, M.), Siemens-Nixdorf Informationssysteme A.G., Munchen, 1991, pp. 53-70.

Molenaar, M. (1993) Object hierarchies and uncertainty in GIS or why is standardisation so difficult, *GeoInformations-Systeme*, Vol. 6, No. 3, pp. 22-28.

Molenaar, M. (1996) Multi-scale approaches for geo-data, *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXI, Part B3, Vienna, Austria, pp. 542-554.

Muller, J. C. (1991) Generalization of spatial data bases, in *Geographic Information Systems* (eds Maguire, D. J., Goodchild, M. F. and Rhind, D. W.), Vol. 1, Longman Scientific & Technical Ltd, New York, pp. 457-475.

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

Peng, W., Molenaar, M. (1995) An object-oriented approach to automated generalization, *Proceedings of GeoInformatics 95*, International Symposium on RS, GIS & GPS in Sustainable Development and Environmental Monitoring, Hong Kong. pp. 295-304.

Peng, W., Tempfli, K. (1996) An object-oriented design for automated database generalization, *SDH 96*, pp. 4B.15-4B.30.

Peng, W. (1997) Automated Generalization in GIS, *PhD Thesis*, Wageningen Agricultural University (and ITC), The Netherlands, 188p.

Peuquet, D. (1984) A conceptual framework and comparison of spatial data models, *Cartographica*, Vol. 21, No. 4, pp. 66-113.

Pilouk, M. (1996) Integrated modelling for 3D GIS, *PhD Thesis*, Wageningen Agricultural University, The Netherlands, 200p.

Richardson, D. E. (1993) Automated spatial and thematic generalization using a context transformation model, *PhD. Thesis*, R&B Publications, Canada, 149p.

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

van Smaalen, J. W. N. (1996) A hierarchic rule model for geographic information abstraction, *SDH 96*, pp. 4B.31-4B.41.

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