Progressive Transmission of Vector Map Data over the World Wide Web*

Michela Bertolotto and Max J. Egenhofer

National Center for Geographic Information and Analysis
Department of Spatial Information Science and Engineering
University of Maine
Orono, ME 04469-5711, USA
{michelab, max}@spatial.maine.edu

Abstract

Within distributed computing environments, access to very large geospatial datasets often suffers from slow or unreliable network connections. To allow users to start working with a partially delivered dataset, progressive transmission methods are a viable solution. While incremental and progressive methods have been applied successfully to the transmission of raster images over the World Wide Web, and, in the form of prototypes, of triangular meshes, the transmission of vector map datasets has lacked a similar attention. This paper introduces a solution to the progressive transmission of vector map data that allows users to apply analytical GIS methods to partially transmitted data sets. The architecture follows a client-server model with multiple map representations at the server side, and a thin client that compiles transmitted increments into a topologically consistent format. This paper describes the concepts, develops an architecture, and discusses implementation concerns.

* Michela Bertolotto's work was partially supported by CNR grant number 106701-00/97/10003. Max Egenhofer's work is partially supported by grants from the National Science Foundation under grant numbers IRI-9613646, SBR-9700465, BDI-9723873, EIA-9876707, and IIS-9970123; the National Imagery and Mapping Agency under grant number NMA202-97-1-1023; the National Institute of Environmental Health Sciences, NIH, under grant number 1 R 01 ES09816-01; and a contract wtih Lockheed Martin M&DS.

1. Introduction

The Internet creates a unique environment for sharing geospatial data. Data providers make their datasets available, much like posting their textual web pages, so that users can use the World Wide Web to download the data for viewing, analysis, or manipulation. While such a setting provides new opportunities for public domain as well as commercial use of geospatial datasets, it also generates new problems such as the availability of vast amounts of data stored in huge unstructured databases (repositories) that need to be searched and downloaded. Therefore, users need better tools to find the datasets they are interested in, be it through improved spatial metadata or spatial summaries (Flewelling and Egenhofer 1999).

Metadata in the form of textual summaries are not suitable when summaries of spatial data are needed not only to convey the idea of what is contained in the database, but also as working datasets themselves. For example, during remote access, temporary versions of the fully detailed dataset may be used to perform some preliminary analysis or manipulation. For this purpose, spatial summaries can be generated by means of subsetting techniques: a meaningful sample of the data is provided instead of the whole set (Flewelling 1997). For instance, in digital image archives, thumbnail images are generated to convey the suitability of high resolution images. Similarly, videos are summarized as video clips (Wactlar *et al*., 1999). Map generalization generates spatial summaries of digital vector maps using cartographic principles (Buttenfield and McMaster 1991; McMaster and Shea 1992; Müller *et al.* 1995; Robinson *et al.* 1984).

Once users have found a particular dataset, they face the problem of transferring the data in a timely fashion from a server to the client, where they intend to analyze and manipulate the data. Since geospatial datasets are typically very large, this process can require long waits. Actually, many available datasets have reached such a fine level of resolution (often even over-detailed with respect to users' needs) that their downloading through slow communication links and mobile devices has become prohibitive. Although communication lines (e.g., fiber optic cable) are getting faster, the current technology does not adequately support wireless communication and transmission of large amounts of data across mobile devices will still be an impediment in the foreseeable future. To overcome such technical impediments, progressive transmission is a viable solution.

Progressive transmission of raster images over the World Wide Web has been successfully applied to provide the user with temporary versions of the data before downloading a complete image. For example, interleaving techniques transmit a sequence of versions of an image. A coarser representation obtained by subsampling the pixels of the image to be displayed on the receiver's screen is sent so that the user can start working without having to wait for the whole image to be downloaded. The initial version is then progressively completed by adding new pixels. Compression mechanisms (e.g., JPEG, wavelets) are also employed to generate complete versions of an image at lower resolution that can be used in progressive raster transmission.

So far, implementations have focused on progressive transmission of raster data through the Web. The transmission of vector data[1], with the exception of the particular case of data in the form of triangular meshes, is generally done by means of a one-step long process. The user attempting to download a vector file needs to wait for the complete version without having the possibility to start working with a coarser version of the data stored in a smaller file. However, for example, geographers and surveyors might want to access a vector map through a mobile device, while performing a data collection directly in the field. Instead of waiting for the complete file to be accessible, they would certainly benefit from the prompt availability of a coarser version of the map on which they can perform initial operations.

In the past, privatization of map data has limited the need to investigate this problem. As we are heading towards increasing availability and global sharing of data, the need for efficient strategies to deliver vector data over the Internet is becoming more pressing. Currently, progressive transmission of vector map data is still a challenging topic due to the intrinsic complexity of map generalization. In this paper, we propose a solution to this problem, based on a client-server architecture. A sequence of coarser representations of each vector map is pre-computed on the basis of sound cartographic principles, stored on the server, and transmitted progressively to the client. This way, not only we generate spatial summaries of vector maps but we also provide users with temporary working datasets while the complete dataset is being downloaded.

The remainder of the paper is organized as follows. In Section 2 we review existing methods for progressive transmission of geospatial data over the WWW. As our focus is on vector map data, models for vector map representation are also reviewed in this section. In Section 3 we outline the challenging issues related to the progressive transmission of vector map data. In Section 4, we propose a possible solution to

[1] Data in vector format refers to a set of spatial entities in the form of points, lines, and polygons that are related through spatial relations.

progressive vector data transmission based on a distributed architecture, and on the pre-computation and storage of multiple map representations on the server site. Section 5 and 6 describe a model for multiple map representations and its efficient encoding. An example of a multiple representation sequence with four levels of detail is provided in Section 7. Finally, implementation concerns are discussed in Section 8, followed by our conclusions in Section 9.

2. Related Work

In this section we review existing methods adopted for progressive transmission of geospatial data over the WWW. The first approaches have been developed for raster data and include interleaving techniques and other mechanisms based on image compression (Section 2.1). The great success and application of image compression mechanisms for progressive raster transmission over the WWW is due to their effectiveness. In fact, they provide good compression with low loss of information. Furthermore, they are relatively easy to implement. However, they do not take into account aspects such as the preservation of topological relations between objects, which is an essential property if actual object manipulation is required.

Recently, the attention has also been revolved to the progressive transmission of vector data (Section 2.2). Within this framework, several compression mechanisms have been defined and implemented for the particular case of triangular meshes, while the investigation for the case of more general vector data is still at its start.

Finally, as our focus is on vector map data, in Section 2.3 we review models for vector map representation.

2.1 Raster Data Transmission

Originally, progressive raster transmission has been implemented by means of interleaving techniques. The simplest method randomly extracts subsets of pixels from the image being downloaded and incrementally completes the image by adding pixels. The implicit row/column ordering of images can also be exploited to choose the subset of pixels more uniformly (e.g., one every five pixels in each row/column, etc.). If hierarchical structures such as quadtrees (Samet 1990a and 1990b) are used to partition the image, more pixels can be extracted from those parts of the image that contain greater concentration of detail.

More sophisticated methods for progressive raster transmission are based on image compression techniques (see, for example, Rauschenbach and Schumann (1999), and Srinivas *et al.* (1999)). Probably the most used image compression method is the JPEG (Joint Photographic Experts Group[2]) format (see Gonzales and Woods (1993) for a detailed description). The JPEG method is a transform-based compression method that decomposes the original data before compression. As images in JPEG format are segmented into rectangular sub-blocks and each block is transformed independently, at high compression ratios they take on unnatural blocky artifacts.

Newer techniques are based on wavelet decompositions (Morlet and Grossman 1984). Wavelet methods are also transform-based but they act on the entire image. The result is a very robust digital representation of a picture, which maintains its natural look even at (relatively) high compression ratios (see Davis and Nosratinia (1998), for an overview). Wavelets have a tremendous advantage for progressive transmission because, in addition to superior overall compression efficiency, they naturally represent image data as a hierarchy of resolution features and its inverse at each level provides subsampled versions of the original image. Consequently, progressive transmission is a natural reconstructive mode for a wavelet-based compression algorithm (see, for example, Rauschenbach and Schumann (1999)).

The mathematics behind wavelets is closely related to Fourier transforms (Titchmarsh 1948; Cooley and Tukey 1965). As Fourier transforms are based on trigonometric functions (i.e., non-local smooth curves which stretch into infinity), they do not well represent sharp spikes and chopped signals.

A lot of work on image compression also relates to fractal theory (Barnsley 1989). A fractal is a geometrical figure whose local features resemble its global characteristics (self-similar property). The difficulty in compressing an image using fractals is in finding a small number of affine transformations to generate the image and in finding subparts of the input image that have self-similarity properties (Fisher (ed.) 1995; Davis 1996). More recently, hybrid compression methods, e.g., combining fractals and wavelets, have been defined (Zhao and Yuan 1996; Davis 1998) to improve the performance of fractal-based compression mechanisms.

[2] A joint body of the Consultative Committee of the International Telephone and Telegraph (CCITT) and International Standardization Organization (ISO).

When transform-based methods are used to compress an image, during transmission over the WWW, instead of the image, the function coefficients are transmitted, and the image is subsequently reconstructed by inverting the transformation. Both transmission and reconstruction are very efficient, as they can be performed simply by means of a single pass through each color band of the image.

The purpose of raster data exchanging over the WWW is very often only visual. As visual meaning can be extracted from images at very low resolution, techniques for progressive raster transmission are usually very efficient. Even when measurements need be performed on the images (e.g., for photogrammetric purposes), common compression mechanisms such as JPEG have proved to be effective (Kern and Carswell 1994). However, sometimes objects need to be directly handled and manipulated. In this case a vector representation is required and preservation of topological and metric properties is a major issue. This topic is discussed in the next section.

2.2 Vector Data Transmission

Part of the vector data exchanged over the WWW comes in the form of triangular meshes. For example, triangulations are used for digital terrain modeling and for real objects surface reconstruction and rendering (Bajaj *et al.* 1999b; De Floriani *et al.* 1998).

Several compression methods for triangular meshes have been defined in the literature (see De Floriani and Puppo (1995) for a survey). These methods are either based on optimal point decimation techniques or they exploit combinatorial properties of triangulations for efficient encoding. Among them, the progressive mesh scheme proposed by Hoppe (1996), represents a fundamental milestone for progressive transmission. However, this method does not have a satisfactory performance as, like many other traditional methods (Eck *et al.* 1995; Zorin *et al.* 1997), it is topology preserving. This limits the level of simplification. More recent methods achieve higher compression by slightly modifying the topology of the input mesh (Bajaj *et al.* 1999a; 1999b).

Juenger and Snoeyink (1998a and 1998b) have defined a parallel point decimation technique for TIN simplification that allows progressive transmission and rendering of TIN data (check http://www.cs.ubc.ca/spider/snoeyink/terrain/Demo.html for a demo). Finally, De Floriani and Puppo (1995) proposed a general framework for multiresolution hierarchical mesh representation. Such a model has a very high storage cost. A more efficient encoding structure has been described in De Floriani *et al.* (1998) to allow progressive transmission. Although these progressive transmission techniques are effective and they have already produced several prototypes (Bajaj *et al.* 1999b; Juenger and Snoeyink 1998a), they can only be applied to data represented in the form of triangular meshes.

As our focus is on vector map data, in the following we describe into details a model for progressive transmission of data in the form of sets of line features (e.g., files storing the hydrography of a given geographic area) (Buttenfield 1999). Such a model is based on the pre-computation of multiple coarser representations. The building procedure consists of the following three steps.

1. *Separation of themes*: different thematic layers (e.g., hydrography, and transportation) are stored in separate files.
2. *Determination of an ordering of features within a theme*: an ordering of the features is established for transmission purposes.
3. *Hierarchical subdivision of vectors within each feature*: each line (represented as a set of arcs) is iteratively subdivided using the Douglas-Peucker algorithm (Douglas and Peucker 1973) and stored in a hierarchical strip tree (Ballard 1981).

A pointer-based encoding structure for the tree is provided and used to visit the tree during progressive transmission and for reconstruction of each level.

One drawback of this method is obviously represented by the pre-processing phase. However, this approach allows progressive real-time analysis of generalized data that might not be possible if on-line simplification was applied on demand. This issue is also strictly related to the well known multiple versus single representation debate. The choice of what representations need to be stored is both application- and data-dependent.

A second drawback of this method relates to the fact that The Douglas-Peucker algorithm, although adopted in many commercial GIS packages, does not guarantee preservation of topological consistency. This problem must be solved if the purpose of progressive transmission is actual manipulation and querying of the exchanged file. A solution consists of performing *a posteriori* checks on the model (see also Saalfeld (1999)). This step, of course, increases pre-processing time.

Finally, this method is applicable only for linear features, while it does not handle intrinsically 2- (or higher) dimensional entities. Only line simplification operations can be performed on the transmitted vector files, while other basic operations (e.g., selection, aggregation, etc.) are commonly used in cartographic generalization (McMaster and Shea 1992) to extract coarser versions of a map. However, a critical issue relates to the fact that a complete formalization of such operations is still lacking as they are traditionally performed manually and subjectively by expert cartographers. This topic is discussed in further detail in the Section 3, where the challenges of progressive transmission of vector map data are presented.

2.3 Vector Map Representation

In this section we focus our attention on the definition of representations for vector maps. First of all, we discuss methods for encoding vector data (Section 2.3.1). Among them, we choose the overlayed approach and describe the mathematical models that are most commonly used for this case, i.e., cell complexes (Section 2.3.2).

2.3.1 Vector Data Encoding

Vector datasets consist of sets of spatial entities in the form of points, lines and polygons that are related through spatial relations (e.g., topological, metric, and direction relations). Two different approaches can be used for storing a vector file. In one case, the file is composed of different thematic layers such as the hydrography of a country, the set of administrative regions of a country, a road network (de Hoop *et al*. 1993). Each layer covers the same area but is composed of a given set of entities sharing the same semantics. Entities belonging to different layers can spatially intersect each other. Thus, the file is composed of a generic collection of entities organized in layers with no particular constraint on overlapping. As an example, in Figure 1, we can detect two different layers: the layer of major road network and the layer of minor road network. Entities belonging to the two layers overlap; however, intersection points are not necessarily recorded (unless they are points of interest in both layers).

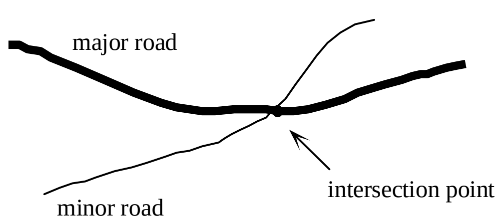

Figure 1: Roads belonging to different layers.

In the second case, the file is composed of a single layer and entities having different semantic and thematic meaning coexist in it. In this case, it is assumed that the original composing set of entities has been overlayed, i.e., there are no intersections between pairs of entities stored in the file.

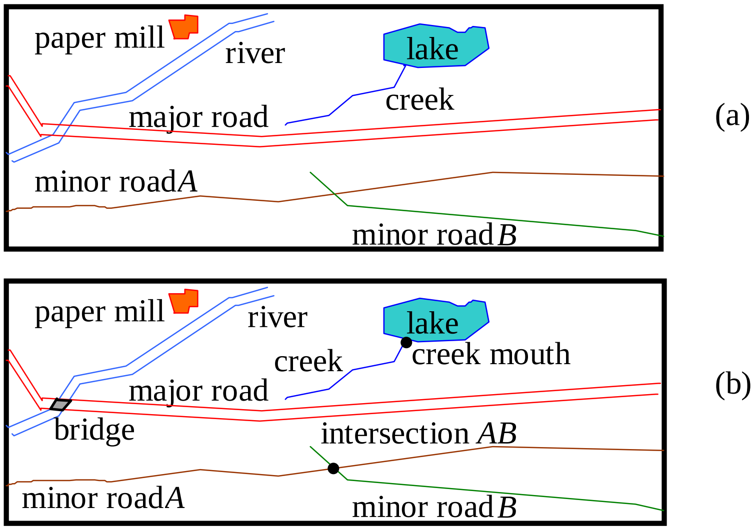

Figure 2: (a) A collection of spatial entities with the following pairwise intersections: *minor road A* and *minor road B*; *creek* and (the boundary of) *lake*; *river* and *major road*; (b) an overlay algorithm has been applied to the set of entities in (a).

For example, in Figure 2(a) a map is shown where intersections between entities occur. In order to obtain an overlayed set, some entities must be added to the vector file storing the map, namely the intersection point between *minor road A* and *minor road B* (*intersection AB*), the intersection point between *creek* and *lake* (*creek mouth*), and the intersection region between *river* and *major road* (*bridge*). Such a file, shown in Figure 2(b), can be generated by applying an overlay algorithm to the set of entities in Figure 2(a) (Bentley and Ottman 1979, Chazelle and Edelsbrunner 1992).

Overlay operations are always required when a query involving entities whose thematic information is stored over two or more different layers must be performed. For example, given a layer representing the hydrography of Italy and a layer describing the set of Italian regions, we may want to perform queries such as, “Which regions are traversed by the river Po?” Map overlay is a complex operation and it is computationally expensive. Thus, if the purpose of the vector file utilization is querying, it is more convenient to keep an already overlayed set of entities instead of performing an overlay operation every time a query is formulated. In the case of progressive transmission, if the client requires the downloading for query purposes, he more likely expects to be provided with a ready-to-use file instead of having to perform expensive preprocessing operations on it, such as map overlay.

On the other side, keeping separated thematic layers in the same file prevents segmentation of entities. For instance, each road in Figure 1 is fragmented into two branches after the detection of the intersection point, if the overlayed approach is adopted. This splitting is not necessary if two distinct layers, one for major roads and one for minor roads, are maintained. Using this second approach, a smaller amount of entities is stored. Furthermore, a higher degree of granularity in the transmission can be obtained. With reference to the above example, the major road network and the minor road network can be extracted from the file and transmitted separately, even if they are included in the same level of detail. In conclusion, the adoption of either the overlayed or the separated layers approach depends on whether the priority is mainly querying or visualization, respectively.

In this paper, we consider vector files composed of a single layer of overlayed entities composing a map. In this case, the most commonly used mathematical model is the cell complex. We describe this concept in the next section.

2.3.2 Cell-Complexes

Cell complexes are commonly used as underlying mathematical models for vector maps (Bertolotto *et al.* 1994; Egenhofer *et al.* 1989; Frank and Kuhn 1986; Güting and Schneider 1995; Worboys 1992) as they provide a sound theoretical basis for the formalization of spatial entities and relations contained in a map.

More formally, we define a *map* as an overlayed set of points, simple lines and (possibly multiply connected) regions. A (Euclidean) cell complex is defined as a collection of cells that are homeomorphic to disks. Each cell has a dimension and its boundary is composed of cells of lower dimension. Pairs of cells in the complex are either completely disjoint or they meet along their boundary. A complex is said to be *n-dimensinal* if *n* is the maximum dimension of its cells.

The correspondence between entities in a map and cells in a two-dimensional cell complex is straightforward: cells in a cell complex describe entities of a map. Points are represented as 0-cells, lines as 1-cells, and regions as 2-cells. The interior and boundary properties of cells in a cell complex model topological relations among entities in a map (see Bertolotto (1998) for a more detailed description).

Several data structures for cell complexes have been defined in the computational geometry and GIS literature (De Floriani *et al.* 1993; Frank and Kuhn 1986; Preparata and Shamos 1985). In particular, topological data structures must encode both entities and their mutual topological relations (Egenhofer and Franzosa 1991).

3. Challenges in Progressive Vector Map Data Transmission

As shown in the previous section, progressive transmission of raster data is relatively simple. For example, if the purpose of the transmission is visualization, just adding pixels to an incomplete image generates a more refined version of it. Available compression mechanisms are also very effective.

When the purpose of the transmission is object manipulation, a vector representation is required. Except for the particular case of triangular meshes for which effective compression methods have been defined (see previous section), progressive vector transmission is still challenging. Our focus is on vector map data, for which effective compression mechanisms are currently lacking. Indeed, the set of spatial objects that need to be represented in a map depends on its level of detail. In this context, the term detail refers both to the amount of entities contained in the file and to the information stored about such entities, e.g., their geometry (Goodchild and Proctor 1997). Therefore, changes of detail can be caused by the addition or elimination of some entities, as well as by the refinement or coarsening of the representation of existing entities (e.g., change of dimension or shape).

Decreasing the level of detail of a vector map is a complex and time-consuming process called map generalization whose automation and on-line application are still open problems. In fact, it relies on cartographic principles whose complete formalization is still lacking (see Weibel and Dutton (1999) for a survey). Automated solutions have only been proposed for sub-problems and are based on heuristic methods (e.g., the Douglas-Peucker algorithm (1973)). Therefore, the process still requires interaction between semi-automatic solutions and expert cartographers.

A critical issue in map generalization is preservation of consistency (e.g., constraints on overlappings) while decreasing the level of detail (Robinson *et al.* 1984; McMaster and Shea 1992; Müller *et al.* 1995; Weibel and Dutton 1999). For example, random subsampling of objects to be eliminated or to be simplified does not usually generate a consistent representation at coarser detail. In Figure 3 and 4, examples are shown where random subsampling generates topological inconsistencies across different representations.

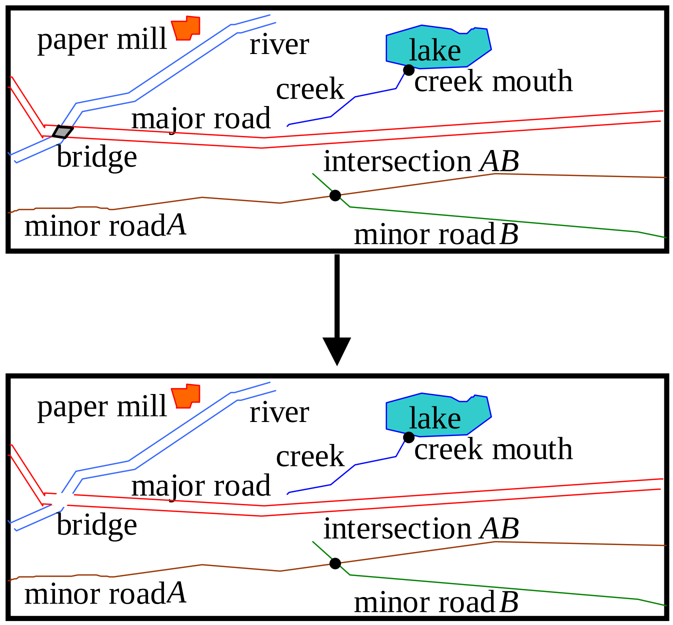

Figure 3: Random elimination of some entities (e.g., *bridge*) results in topologically incorrect representations: the two branches of *major road* and *river* have incomplete boundaries.

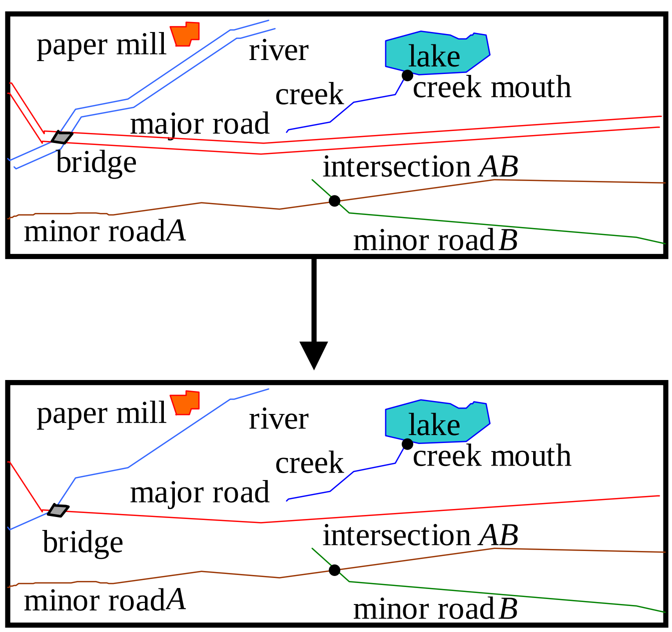

Figure 4: Selection of a random subset of entities, namely, *major road* and *river* (but not *bridge*), to coarsen their representation generates a topologically incorrect representation: topological relations between *major road* and *bridge* (and between *river* and *bridge*) are not consistent across levels.

Consistency is an essential property for the usability of data (e.g., for analysis and querying purposes). Therefore, *ad hoc* techniques and operators need to be applied. Unfortunately, the majority of proposed methods do not intrinsically provide consistency and thus *a posteriori* checks are required in order to adjust the result when some inconsistency has been introduced.

During progressive transmission, consistent datasets need to be transmitted to the client site, where there is no way to discover inconsistencies. Thus, consistency must be checked for on the server site either implicitly by adopting generalization operators that preserve it or by means of *a posteriori checks* on the generalized datasets.

In the following section we propose a solution to progressive vector map data transmission.

4. A Client-Server Architecture for Progressive Vector Transmission

In this section we present a distributed client-server architecture that enables progressive transmission of vector data. This work extends our previous research on progressive vector transmission presented in Bertolotto and Egenhofer (1999). The solution we propose meets the architectural requirements for the design of contemporary information systems, involving both heterogeneity of data and processes. In fact, as different tasks are better performed by different system components, distributed environments represent the natural direction for which future generation systems are heading. This direction is in accordance with the exponential network growth and the global connectivity reached in the last few years. As a consequence, and thanks to the technological improvements recently obtained, some of the old systems and, in general, newly developed systems, are being redesigned and embedded into distributed environments (see, for example, Hofmann (1999); Kramer *et al.* (1997); Kumar *et al.* (1999)).

To enable the progressive transmission of vector files, we pre-compute and store a sequence of consistent representations at lower levels of detail on the server site. Such representations are transmitted in order of increasing detail to the client upon request (Figure 5).

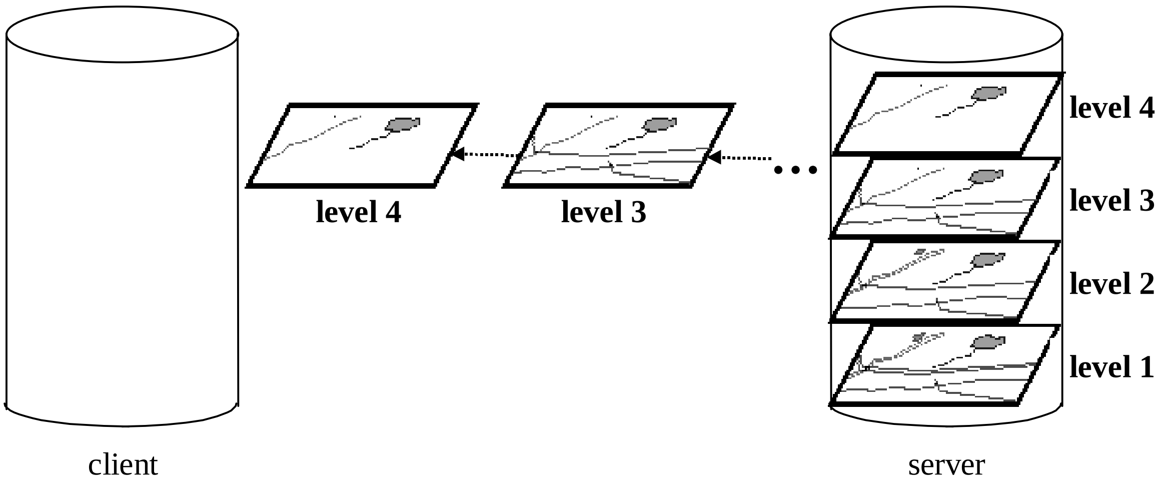

Figure 5: Representations at different levels of detail are maintained on the server and transmitted to the client in order of increasing detail.

Besides providing temporary versions of the data on which preliminary operations can be performed, in this way, the user can realize during the process that the detail of the currently displayed representation is good enough for her purpose and so she can decide to interrupt the downloading of more detailed representations (stored in larger files). Therefore, both time and disk space can be saved.

An interesting property of some techniques for progressive transmission of raster data (e.g., interleaving techniques) is the fact that they only transmit increments (i.e., sets of new pixels) at each step. Such increments are added to the currently displayed image to improve its resolution without requiring the transmission or downloading of another complete image file.

Since vector files can be very large, it would be useful to speed up the transmission in a similar way to reduce network traffic. However, increments between two consecutive levels of detail of a vector dataset can be complex sets of entities that are added to the level at finer detail to refine the representation of a set of entities at the lower level. The integration of such increments with the currently downloaded or displayed representation is a non-trivial task if consistency between different representations must be preserved.

From a purely graphical point of view, a major requirement for spatial analysis consists of providing a method for combining the results of several queries into a unique graphical representation. Therefore, the system should allow the addition or elimination of representation layers from the current display. However, in the case of progressive transmission, the user may be interested in more than a visual inspection of the data. She may want to analyze and work with a consistent dataset. A vector file is usually composed not only of a set of points, lines, and polygons, but also of a set of spatial relations linking such entities. Thus, suitable overlaying and integration techniques must be applied not only at the graphical level, but also at the data level to include the computation of spatial relations between newly introduced entities and preserved entities (see Section 8.2).

Our solution for progressive vector map transmission relies on a distributed client-server architecture, with a server provided with methods for building, manipulating, and transmitting a sequence of map representations at different levels of detail; and a thin client provided with a set of operations for visualizing as well as updating and integrating the transmitted levels.

In summary, the following functionalities are required:

- *on the server site*: a preprocessing task must be performed in order to build a sequence of map representations at different levels of detail, as well as a progressive transmission technique must be developed to send the different levels one at a time; and
- *on the client site*: a mechanism must be built in order to have different graphic layers so that the displayed representation is complete at each step (corresponding to the transmission of a given level of detail), and an integration algorithm must be developed and implemented for reconstructing the dataset corresponding to the displayed representation.

Thus, the major issues involved in the process are:

- the definition of a model for consistent multiple map representations;
- the definition of a transmission technique;
- the development of a graphic interface and a dynamic visualization tool for displaying different layers corresponding to different levels of detail (during transmission); and
- the development of algorithms to compile transmitted increments into a topologically consistent format for the reconstruction of intermediate levels of detail.

In this paper, we deal with modeling aspects involved in progressive vector map transmission, disregarding the mainly graphical issues. In particular, in next section we describe a model for multiple map representations.

5. Multiple Map Representations for Progressive Transmission

In this section we describe a model that was initially presented in Puppo and Dettori (1995) and then further formalized in Bertolotto (1998). The model consists of a sequence of map representations of a given area, each corresponding to a different level of detail. Such sequence can be stored on the server and sent progressively to the client upon request (as described in Section 4). In Bertolotto (1998) a set of generalization operators have been defined and shown to be minimal and sufficient to generate consistent transformations of maps. Only topological changes (e.g., changes of dimension, complexity, etc.) are possible by applying such operators: metric and semantic changes are not being taken into account. An example of topological changes in a map is shown in Figure 6.

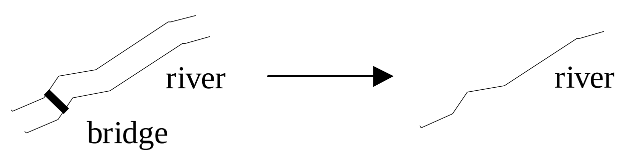

Figure 6: Topological changes between two different levels of detail in a map: at higher level (left) the river is represented as a region split into two branches by a bridge (represented as a line); at lower level (right) the bridge is not represented and the set of entities representing the river are reduced to a single line.

The set of operators defined is the following:

- **Line contraction**: contraction of an open line (including its endpoints) to a point (Figure 7(a)).
- **Region contraction**: contraction of a simply connected region (with its boundary) to a point (Figure 7(b)).
- **Region thinning**: a region (and its bounding lines) is reduced to a line (Figure 7(c)).
- **Line merge**: fusion of two lines sharing an endpoint into a single line (Figure 7(d)).
- **Region merge**: fusion of two regions sharing a boundary line into a single region (Figure 7(e)).
- **Point abstraction**: elimination of an isolated point inside a region (Figure 7(f)).
- **Line abstraction**: elimination of a line inside a region (Figure 7(g)).

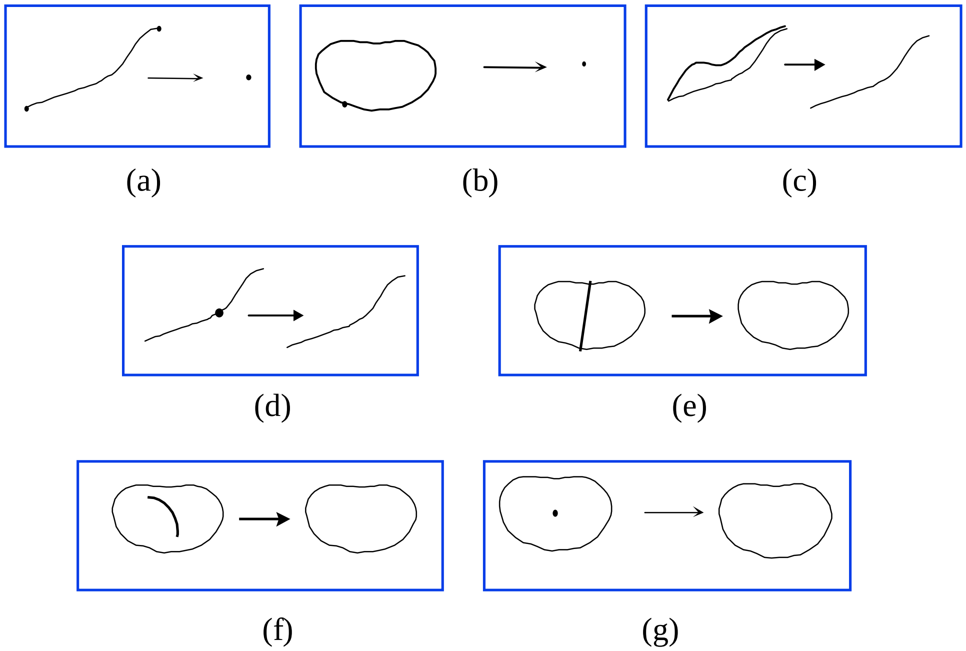

Figure 7: Generalization operators.

These operators have been formalized by means of functions between the cell complexes describing the original and the resulting map (Bertolotto 1998).

For example, in Figure 7(a), if l is the line on the left, p' and p'' are its endpoints, and p is the point on the right, the **line contraction** operator is formalized by the function $l_c : C \to C'$, where C and C' are cell complexes, such that:

$$l_c(p') = l_c(p'') = l_c(l) = p \tag{1}$$

and

$$\forall e \in C, e \neq p', p'', l, \;\; l_c(e) = e \tag{2}$$

Equation (2) states that all entities in C other than l, p' and p'' are preserved. Furthermore, constraints on mappings of topological relations must be fulfilled. For instance, all lines in the map having an endpoint in either p' or p'' are transformed into lines of C' having an endpoint in p. A complete description of such constraints can be found in Bertolotto (1998).

Contractions and thinning operators correspond to a decrease of dimension for a group of entities, merge operators group two entities of the same dimension into a single one, while abstraction operators correspond to the elimination of some lower-dimensional entity from a region. These operators are called atomic, because they perform minimal changes, i.e., they modify minimal sets of entities and preserve the others. By composing functions corresponding to such operators, complex map transformations can be defined.

As outilined in Section 3, in the case of progressive transmission between two remote sites, the preservation of consistency is a major issue, and it must be guaranteed by the server as the client has no way to control it. When the previously described operators are used, consistency is automatically guaranteed. In fact it has been shown that they represent a minimal and sufficient set of functions that generate (by composition) only consistent transformations for decreasing the level of detail of a map (Bertolotto 1998). In particular, they allow to generate all map transformations that preserve the boundary of entities and whose inverse image preserves connectivity (i.e., they do not allow to model aggregation of non-connected entities[3]). Topological relations are thus consistently transformed. Therefore, all multiple representation sequences built on the basis of such operators are intrinsically consistent.

The model for a multiple representation sequence can be formalized as follows: let M_0 be the map at higher level of detail (stored on the server) and let $f_1, f_2, \ldots, f_k$ be an ordered sequence of map transformations, i.e., a sequence of functions obtained as compositions of atomic operators; let $f_1 : M_0 \to M_1, f_2 : M_1 \to M_2, \ldots, f_k : M_{k-1} \to M_k$ be such that

$$f = f_k \circ f_{k-1} \circ \ldots \circ f_2 \circ f_1 : M_0 \to M_k$$

is a transformation of maps. The sequence $M_0, \ldots, M_k$ is composed of maps corresponding to less and less detailed representations of the same area, and it is called a multiple representation sequence (see Figure 8 for an example).

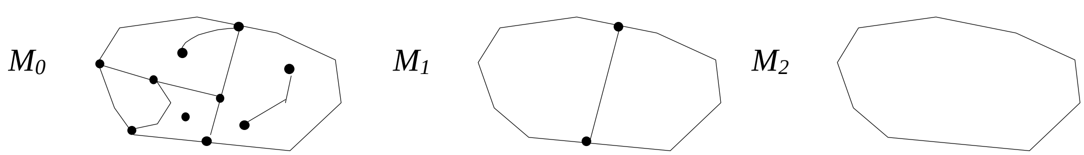

Figure 8: A multiple representation sequence with three levels of detail.

[3] The formalization of aggregation of disjoint entities is traditionally a difficult issue, as this operation might generate semantic and topological inconsistency. A tentative solution generates a fictitious “shell” containing the entities to be aggregated. However, this method has the drawback of forcing an arbitrary partition (Dettori and Puppo 1996).

6. Encodings for Efficient Transmission

The efficiency of the model presented in Section 5 can be improved by defining encodings that avoid redundancy between levels and by adding to it intra-level links to allow navigation and browsing across levels. These issues are discussed in Section 6.1 and Section 6.2.

6.1 Data Structure

An important issue relates to the efficient storing of multiple map representations. Some entities are preserved throughout the levels. Such entities should be stored only once in the sequence instead of redundantly encoding them at each level. If such representations are to be transmitted through the Web, reducing the amount of information to be exchanged is highly useful. A straightforward way to avoid redundancy is storing the entire set of entities only for the coarsest level, while all subsequent levels just record newly introduced entities and more refined representations of entities present in previous levels (i.e., the *increments*). This way transmission is also sped up; however, by not storing the complete set of entities corresponding to each transmitted level, the problem arises of defining algorithms for reconstructing the vector file corresponding to an intermediate level, once the transmission has been terminated, to use, manipulate, and query it. The problem lies in the computation of the spatial relations between newly introduced entities and preserved entities. This problem does not occur if the purpose of the transmission is just visualization. In such a case, only techniques for overlaying graphical layers are necessary.

In the following we describe a structure for encoding a sequence $M_0,\ldots,M_k$ of multiple representations in which entities are stored only once to avoid unnecessary duplications. Such a structure is defined inductively on the basis of the inverse image of entities inside each map (Puppo and Dettori 1995). More formally, for $1 \le i \le k$, given a submap $M_{i,j}$ of a map M_i, we consider each entity *e* of $M_{i,j}$ such that its inverse image through f_i contains more than one entity in M_{i-1}, i.e., *e* is not a preserved entity, its representation is more detailed in M_{i-1}. $M_{i-1,p} = f_i^{-1}(e)$ is a submap of M_{i-1}. The set of submaps of M_{i-1} corresponding to all inverse images of entities in M_i that contain more than one entity form level *i-1*. In this way, the map at the coarsest level (i.e., M_k) is completely stored, while only increments are stored for intermediate levels. In Figure 9 we show an example of this encoding structure for the multiple representation sequence of Figure 8. Note that entities that are preserved across different levels are represented only at the coarsest level in which they appear (they are drawn as dashed in the intermediate levels in which they are not stored explicitly).

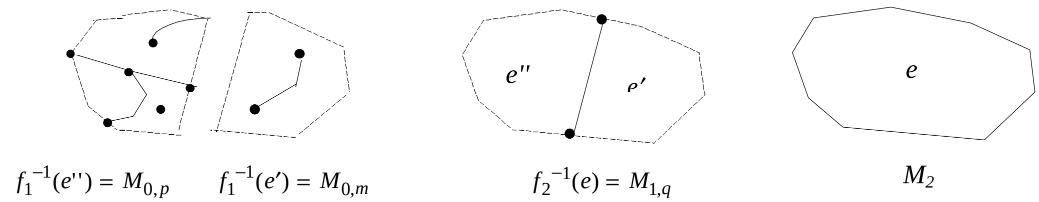

Figure 9: An example of encoding structure for the multiple representation sequence of Figure 8.

6.2 Adding Intra-Level Links

In this section we enrich a multiple representation sequence with links that connect different representations of the same entities at different levels. We call such links *intra-level* links or *vertical* links, since they provide the sequence with a hierarchical structure.

Keeping intra-level links facilitates the reconstruction of a representation at a given level of detail and it allows us to know which are the entities that are being modified at subsequent levels and which ones are preserved.An example of intra-level links joining different representations of a river and a bridge at different levels in a multiple representation sequence is shown in Figure 10.

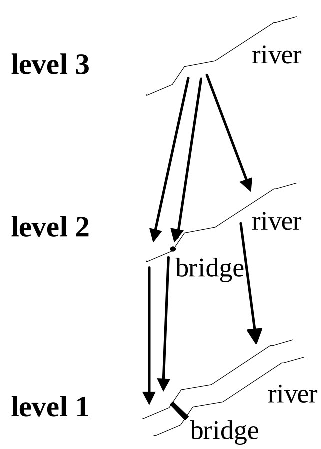

Figure 10: Intra-level links joining different representations of a river and a bridge in a multiple map representation sequence.

Intra-level links also allow for hierarchical spatial reasoning. In particular, in a querying environment, the user can be interested in retrieving information at different detail.

Although a lower level of detail can be sufficient for processing a given query, sometimes a more detailed answer is required. Maintaining intra-level links allows for efficient navigation and browsing across the levels without having to query entire levels. A query is initially performed at the lowest level and the result can be evaluated according to a given user-defined criteria. If the outcome of the evaluation is satisfactory, there is no need to query against a more detailed dataset. Otherwise, the intra-level links are followed in order to find a satisfactory answer. Usually only links between entities that belong to consecutive levels are kept. By combining them, also links between non-consecutive levels can be obtained.

For the model described in Section 5, links between entities in consecutive levels are formalized by means of the transformation functions defining the sequence (obtained as compositions of atomic operators). Such functions perform *generalizations* of maps. The inverse operation, i.e., the *refinement* of maps, can be formalized by "inverting" such functions. Although these functions are not injective, in Bertolotto (1998), the inverse image of an entity in the co-domain of atomic generalization functions between cell complexes representing maps has been characterized in such a way that an entity can be refined through a small number of operations. Let p, l, and r denote a point, a line and a region, respectively. It has been shown that the inverse image of a region can only be one of the following sets:

- $\{r, r', l\}$ (i.e., the domain of operator **region merge**),
- $\{r, p\}$ (i.e., the domain of operator **point abstraction**),
- $\{r, l\}$ (i.e., the domain of operator **line abstraction**).

The inverse image of a line can only be one of the following sets:

- $\{p, l, l'\}$ (i.e., the domain of operator **line merge**),
- $\{l, l', r\}$ (i.e., the domain of operator **region thinning**).

Finally, the inverse image of a point can only be one of the following sets:

- $\{r, l, p\}$ (i.e., the domain of operator **region contraction**),
- $\{l, p, p'\}$ (i.e., the domain of operator **line contraction**).

Therefore, we can define the inverse of generalization operators (defined in Section 5) as follows:

- **Point-region expansion**: expansion of a point into a region (with its boundary)

$$p \to \{r, l, p'\} \qquad (3)$$

- **Point-line expansion**: expansion of a point into a line (with its endpoints)

$$p \to \{l, p', p''\} \qquad (4)$$

- **Line-region expansion**: expansion of a line into a region (with its boundary)

$$l \to \{r, l', l''\} \qquad (5)$$

- **Line split**: splitting of a line into two lines sharing an endpoint

$$l \to \{l', l'', p\} \qquad (6)$$

- **Region split**: splitting of a region into two regions sharing a bounding line

$$r \to \{r', r'', l\} \qquad (7)$$

- **Point insertion**: insertion of an isolated point inside a region

$$r \to \{r', p\} \qquad (8)$$

- **Line insertion**: insertion of a line inside a region

$$r \to \{r', l\} \qquad (9)$$

Basically, the inverse operations of contractions and thinning are expansions of an entity into a set of entities including some higher dimensional entity. The inverse of merging is splitting an entity into two entities of the same dimension, while the inverse of abstracting an entity from a region is the insertion of an entity inside a region.

As for generalization operators, these refinement operators only perform minimal changes. Thus, for example, in the **line split** operator, endpoints of line l are mapped onto the corresponding endpoints of lines l' and l'' (Figure 11).

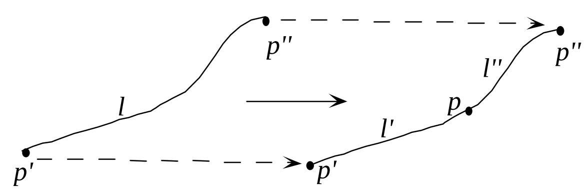

Figure 11: Endpoints of line l are preserved by line split.

On the basis of *refinement* operators, i.e., by considering inverse images of entities, we enrich the model defined in Section 5 and encoded by means of the structure defined in Section 6.1 with intra-level links. The resulting model has a tree-like organization. The root is the map at the coarsest level of detail M_k. Each submap $M_{i-1,p}$ that corresponds to the inverse image of some entity *e* belonging to a submap $M_{i,j}$ at level *i* in the sequence is stored as a node and is called a *child* of $M_{i,j}$. An arc between $M_{i,j}$ and $M_{i-1,p}$ is established and labeled *e*. The child relationship represents the *refinement* of a submap, while the parent relationship represents *generalization* of a submap.

In Figure 12 the example of Figure 9 has been enriched with intra-level links.

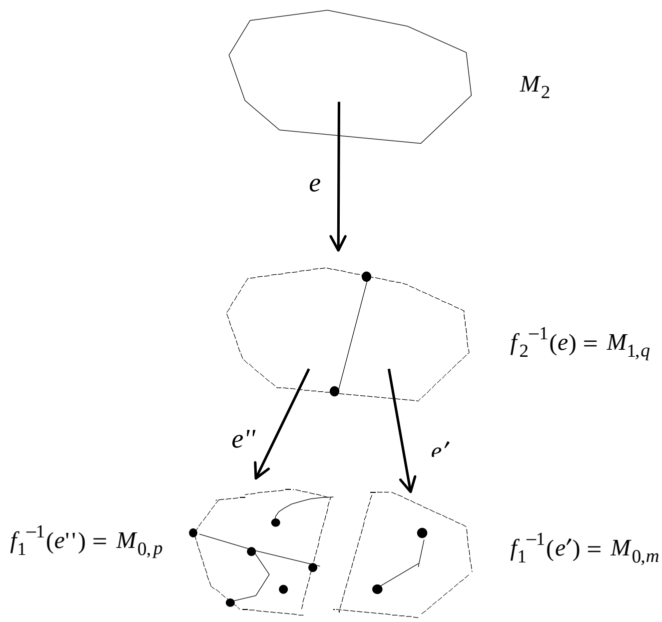

Figure 12: Adding intra-level links results in a tree-like structure. Arcs are labeled with the corresponding refined entities.

In the following section, an example of a multiple representation sequence and its efficient encoding is described.

7. An Example

In this section, we describe an example of a multiple representation sequence with four levels of detail (Figure 13). The coarsest level (level 4) contains the hydrographic map of a given geographic area. This level includes the following set of entities *{river, creek, paper mill, lake, creek mouth}*, such that *river* and *creek* are represented by lines, *paper mill* and *creek mouth* (i.e., the intersection between *creek* and *lake*) are points, and *lake* is a region. In level 3, the road network covering the same area is also represented: *major road, minor road A, minor road B, intersection AB,* and *bridge* are added. The first three entities are represented by lines. *Intersection AB* is the intersection point between *minor road A* and *minor road B,* while *bridge* is the intersection point between *major road* and *river*. Therefore, *minor road A, major road* and *river* are represented at this level as lines split into two segments. Level 2 corresponds to a refinement in the representation of *paper mill,* that is now a region, *river,* for which the thickness is shown (in both the branches composing it), and *bridge,* that is now a line. All other entities are preserved, i.e., their representation at this level does not change. Finally, the fully detailed map at level 1 refines the shape of *paper mill* and shows the thickness of the two segments of line that comprise *major road,* thus implying a transformation of *bridge* from line to region.

In order to build such a sequence of representations to be stored on the server site, we apply the generalization operators described in Section 5. For example, to obtain level 3 from level 2, the following operators are applied:

- **region thinning** of *river,*
- **line contraction** of *bridge,* and
- **region contraction** of *paper mill.*

Figure 14 illustrates the structure described in Section 6.1 for the efficient encoding of the multiple representation sequence of Figure 13. The set of entities in level i (for $1 \leq i \leq 3$) corresponds to the increment with respect to level $i+1$. Entities that are present at both level $i+1$ and i, and whose representation is refined at level i, are highlighted at level i. An example is provided by *river,* that at level 4 is represented by a single line, while at level 3 is composed of two segments joined by *bridge.*

Level 4 is completely represented, i.e., all entities together with their spatial relations are stored in the encoding structure. For instance, the information that *creek mouth* is an endpoint of *creek* is maintained. The set of spatial relations that are explicitly stored depends on the particular data structure adopted to encode each map in vector format (e.g., the extended *DCEL* structure defined in De Floriani *et al.* (1993)). Among the entities present at level 3, only *river* (fragmented into two segments) and the newly introduced entities *(major road, bridge, minor road A, minor road B,* and *intersection AB)* are encoded, as well as their spatial relations. At level 2, the only changes are the refinement of *paper mill, river* and *bridge.* Only these entities and their relations are represented. Similarly, at level 1, the encoding structure includes just *paper mill, bridge* and the two segments composing *major road.* Intra-level links are also maintained between different representations of the same entities at consecutive levels.

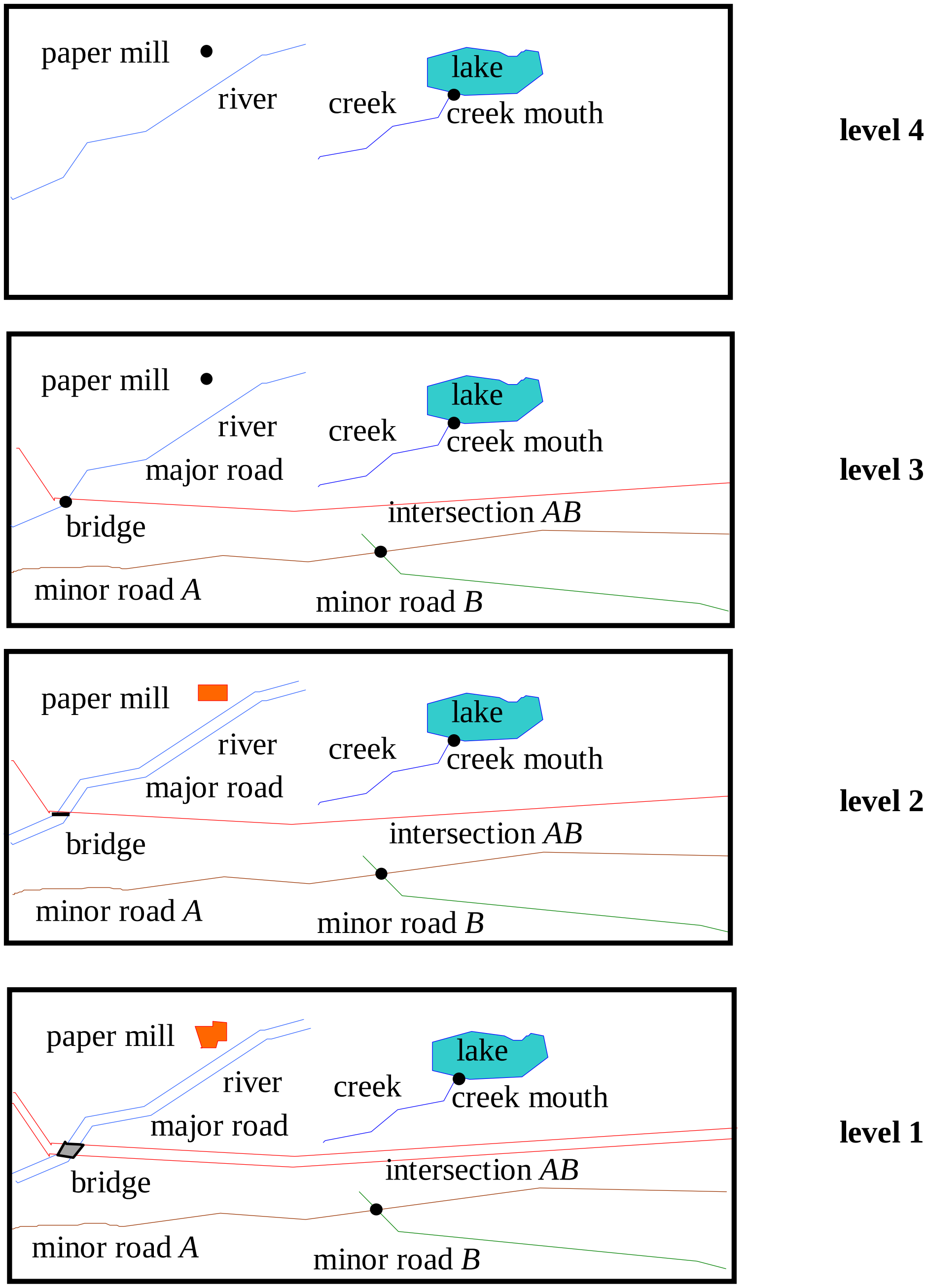

Figure 13: A multiple map representation sequence with four levels of detail.

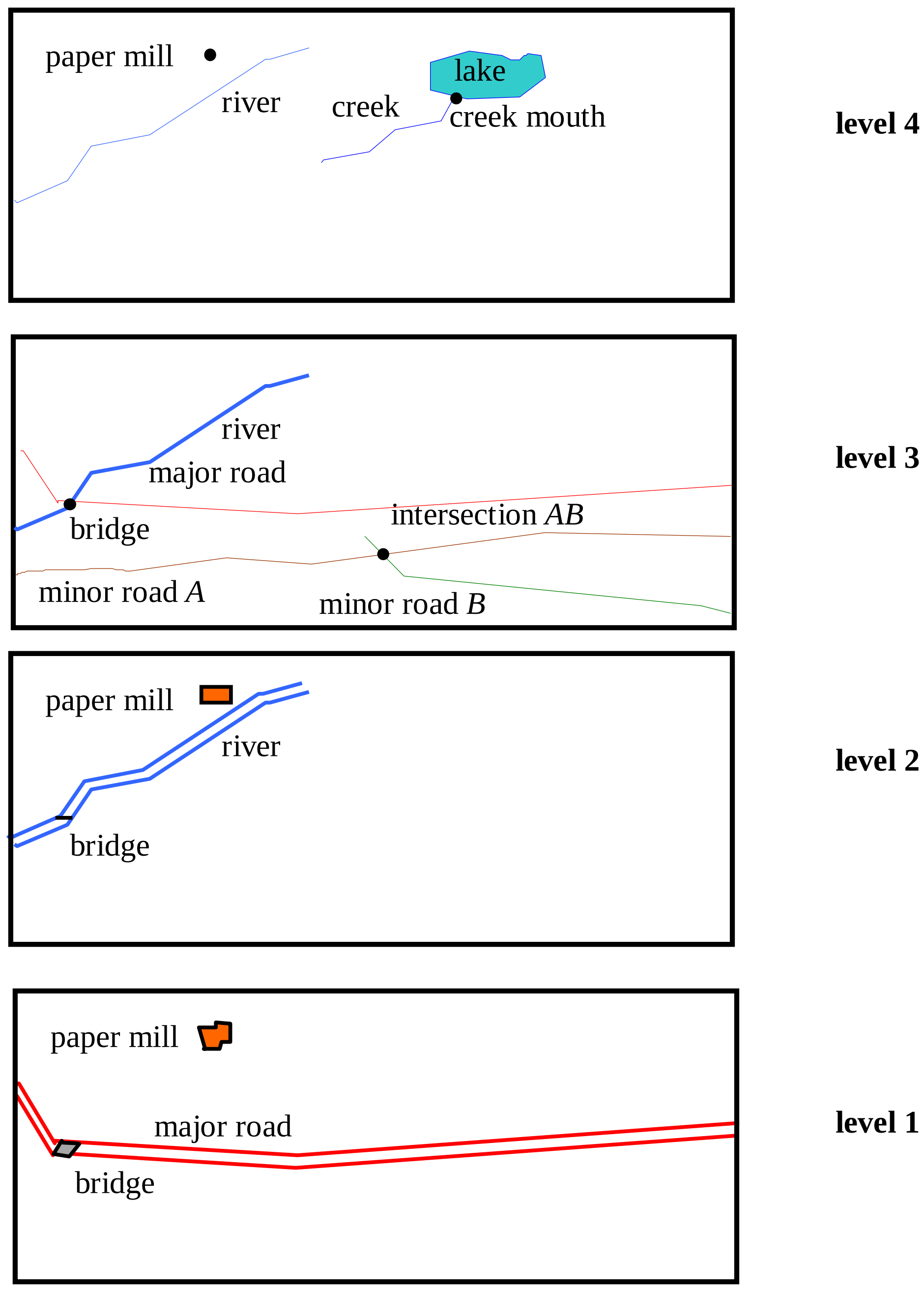

Figure 14: Efficient encoding of the sequence represented in Figure 13.

8. Discussion and Implementation Issues

In Section 5 we have described a model for multiple map representations that can be used for progressive transmission of map data over the WWW. The model is built on the basis of a set of operators that perform topological changes on a map. Although such operators have the interesting property of preserving topological consistency, they only modify the topology of the map. However, in real implementations also changes in the geometric shape of objects are commonly required, such as simplifications of lines (Douglas and Peucker 1973). We are currently investigating the possibility of extending our model to include this kind of operations. Changes depending on the semantic of objects should also be allowed. The incorporation of semantic aspects is another goal of our current research.

A model for generating multiple map representations based on the Douglas-Peucker algorithm and used for progressive transmission over the WWW has been described in Section 2.2 (Buttenfield 1999). However, such a model has the drawback of not guaranteeing the preservation of consistency. Furthermore, it only deals with line simplification operations and does not allow other kinds of commonly used map generalization operations (McMaster and Shea 1992).

Finally, as underlined in Section 2, a well known question related to the generation of multiple map representations is the following: "How many and what are the levels of detail to be generated and stored?" The answer to this question strongly depends on the application and on the particular task. For an implementation of our model, we plan to analyze typical queries from different application fields to detect what the best levels of detail are to provide satisfactory answers. Also, as underlined by Buttenfield (1989), it is possible to detect the range of resolution at which some geographic processes become evident. Such considerations will help decide what levels of detail need to be stored on the server site.

In the following sections we discuss in more detail issues related to the implementation of the model for progressive transmission proposed in this paper. In particular, we consider the data transmission technique adopted (Section 8.1) and the reconstruction of datasets corresponding to intermediate levels of detail downloaded on the client site (Section 8.2).

8.1 Transmission of Different Levels of Detail

In the solution we propose, the process of transmitting a multiple representation sequence between two remote sites presents a few critical points. We assume to have a dataset *d* on the server and a sequence of multiple representations of *d* corresponding to *k* levels of detail, such that the higher the level, the less detailed the representation. We denote such sequence by *mr(d,k)*. We consider a transmission model in which the user is attempting to download progressively the dataset *d*. Each level of detail corresponds to a separate package. As described in Section 4, the process starts with the server transmitting the representation corresponding to the first level of detail (i.e., the coarsest). We can formalize such process by means of a binary operation *t* that takes a sequence *mr(d,k)* and an integer *i*, with $1 \leq i \leq k$, and performs the transmission of the representation corresponding to level *i* in the sequence. Thus, the first step in the transmission process can be denoted by

$$t(mr(d,k),k) \tag{10}$$

Upon completion of the transmission of the first package, the server continues the transmission of the subsequent representations in order of increasing detail until the user decides to stop the process. We indicate this sequence of steps with

$$t(mr(d,k),k-1), \ldots, t(mr(d,k),1) \tag{11}$$

In order to avoid unnecessary repetitions for storage saving purposes, in the encoding structure for the model described in Section 6.1, given a representation corresponding to a level *i* $(0<i<k)$ of detail in the sequence, only the entities that have been modified are stored at level *i-1*. Not only this model saves space on the server site, but it also speeds up the transmission to the client site, i.e., it minimizes network traffic. On the server site the coarsest representation *mr((d,k),k)* is completely stored, while only portions of the subsequent representations (increments) are stored in the subsequent levels. Thus, the size of the transmitted packages is reduced. On the client site, a new buffer is used to store the increments to be integrated with the previously transmitted representation. From a graphical point of view, such a buffer can be used as a graphic layer to be superimposed on the previous one in order to completely display the representation at the new level of detail. From a more operational point of view, *ad hoc* integration techniques must be developed (such as for reconstruction of the topology of the dataset, discussed in next section).

A further way to increase the granularity of the transmission is to send subparts of the increments that are topologically disjoint (corresponding to nodes in the tree) as separate packages. This way, topological consistency is still preserved and even smaller packages are transmitted. Also, the user would have the possibility to decide to increase the detail of only some subparts of the map (i.e., to decide to download only the subtrees corresponding to the interesting areas), thus generating a representation at "variable" detail (the so-called *selective transmission*). Entities belonging to different levels of detail would coexist in the representation reconstructed on the client site. We are currently investigating this interesting aspect.

An important issue concerns the transmission of intra-level links. As discussed in Section 6.2, during query processing, multiple representation sequences benefit from the addition of intra-level links between different representations of the same entity at different levels of detail. A query is first performed at the coarsest level; if the result is unsatisfactory, intra-level links can be followed to find a better answer. In this way we do not need to query the entire following level. Furthermore, as discussed in next section, intra-level links facilitate the reconstruction of the intermediate levels. Thus, such links must be encapsulated in the transmitted packages, in the form of pointers, in addition to the new entities to be displayed in the overlayed buffers. Each transmitted entity maintains a reference to (the ID of) its parent in the previous level.

8.2 Reconstruction of a Dataset

Once the user at the client site decides to stop the transmission process, because the level of detail of the currently displayed representation is good enough, the dataset corresponding to that intermediate representation must be reconstructed. Such a dataset is not stored explicitly (i.e., as an independent representation level) on the server either. Thus, on the client site, integration operations are needed to compile a consistent representation at a given level of detail from the collection of displayed layers. Operations for deleting entities in the previous layers as well as for adding new sets of entities (belonging to newer layers) to substitute them are needed on the client site. Furthermore, spatial relations between newly introduced entities and preserved entities must be computed. This operation can be very time consuming.

For the hierarchical model we consider, i.e., the sequence of multiple representations described in Sections 5 and 6.1, enriched with vertical links (Section 6.2), a complete representation corresponding to a given level of detail can be obtained by means of a visit of the tree. The visit includes all nodes up to the desired level: entities contained in a given node are deleted and replaced by the set of entities corresponding to the child of such node (i.e., its refinement). The visit of the tree can be performed on the client site as vertical links are transmitted together with increments (i.e., nodes in the tree) in the form of pointers (as discussed in the previous section).

Due to the intrinsic consistency properties of the model (in terms of preservation of topology), topological relations between two entities e and e' at level i can be reconstructed on the basis of relations between entities that are *parents* of e and e' at the preceding level.

As an example, we consider the sequence of multiple map representations shown in Figure 13. Such a sequence is stored on the server and transmitted progressively. For each transmitted level, the client receives only the increments and the intra-level links. In this example, only level 4 is completely represented. Therefore, intermediate levels must be reconstructed on the client site. For instance, to reconstruct level 3 from level 4 and the increment between levels 4 and 3 (including: the refinement of *river*; the new entities *bridge, major road, minor road A, minor road B,* and *intersection AB*; and the set of spatial relations among them), the following operations must be performed:

- the line representing *river* at level 4 must be substituted by the two segments that share *bridge* as an endpoint;
- *bridge, major road, minor road A, minor road B,* and *intersection AB* must be added as new entities; and
- the complete set of spatial relations for level 3 are reconstructed in a straightforward way as union of the set of relations between preserved entities (stored at level 4) and the set of relations between newly introduced entities (included in the increment package).

9. Conclusions

This paper discusses the need for progressive vector transmission, by which a user is provided with coarser versions of the data while she is downloading a large vector file from a remote site. This process presents inherent challenges. Extracting a consistent representation at a lower level of detail from a vector map dataset is a complex and time-consuming operation that cannot be performed on-line during progressive transmission. A solution is to pre-compute, on the basis of sound cartographic principles, a sequence of multiple representations of the data to be stored on the server site. Each representation, corresponding to a different level of detail, is transmitted separately. This solution has the obvious drawback of requiring preprocessing. However, multiple representations can be utilized to define structured digital libraries of summaries of geospatial data. Such organization enables efficient access and manipulation of such data over the WWW.

Our contribution is a solution to vector map data transmission, based on a distributed architecture. A model generating multiple map representations (Puppo and Dettori 1995; Bertolotto 1998) that can be used for progressive transmission is described. The model is built by applying a set of generalization operators that perform atomic topological changes on vector maps. The defined operators guarantee the preservation of consistency (Bertolotto 1998), an essential property for the usability of generalized data. Our future goal is to extend such a model to include metric and semantic aspects as well as to develop alternative models and to compare them in terms of storage and transmission costs. The development of efficient encoding strategies plays a central role in this framework.

In a more general setting, we are planning the design of a heterogeneous and distributed geographic information system integrating data in different formats (e.g., raster, vector, and textual) and retrieved from different sources. The system will include distributed Internet archives in the form of digital libraries of generalized geospatial data for fast and efficient use. Different formats of the same data will be automatically chosen by the system to perform different tasks on the basis of efficiency. For example, rasterized (scanned) versions of maps will be provided to the user for mainly visual purposes while the corresponding vector versions will be progressively transmitted if actual object manipulation is required. Several aspects will be integrated in the data including spatial, temporal and semantic aspects to allow a wide range of operations and analysis.

Another essential requirement of contemporary information systems is related to the concept of interoperability, i.e., the capability of autonomous systems to exchange data and to handle processing requests by means of a common understanding of processes and requests (Sondheim *et al.* 1999). This can be achieved if common representation models are used (UCGIS 1996). In GIS, data modeling plays a central role in the framework of interoperability, as agreement at the representation level is critical for exchanging geospatial data. Therefore, the design and implementation of our comprehensive Internet-based archive, will meet interoperability and standardization requirements as defined by the OpenGIS Consortium (OGC 1996a and 1996b).

10. References

C.L. Bajaj, V. Pascucci, and G. Zhuang, 1999a, "Single resolution compression of arbitrary triangular meshes with properties", *IEEE Data Compression Conference*, pp. 247-256.

C.L. Bajaj, V. Pascucci, and G. Zhuang, 1999b, "Progressive Compression and Transmission of Arbitrary Triangular Meshes", *Technical Report*, Department of Computer Science, University of Texas at Austin, Austin, TX. http://king.ticam.utexas.edu/CCV/papers/cpm-all.pdf

D. H. Ballard, 1981, "Strip Trees: A Hierarchical Representation for Curves", *Communications of the ACM*, 24(5), pp. 310-321.

M. Barnsley, 1989, *Fractals Everywhere*, Academic Press, San Diego.

J. L. Bentley and T. A. Ottman, 1979, "Algorithms for reporting and counting geometric intersections", *IEEE Transactions on Computers*, 28(9), pp. 643-647.

M. Bertolotto, L. De Floriani, and E. Puppo, 1994, "Multiresolution topological maps", in M. Molenaar, S. De Hoop (eds.), *Advanced Geographic Data Modelling - Spatial Data Modelling and Query Languages for 2D and 3D Applications*, Netherland Geodetic Commission, Publications on Geodesy - New Series, 40, pp. 179-190.

M. Bertolotto, 1998, "Geometric Modeling of Spatial Entities at Multiple Levels of Resolution", *Ph.D. Thesis*, Department of Computer and Information Sciences, University of Genova, Italy.

M. Bertolotto and M. Egenhofer, 1999, "Progressive Vector Transmission", *Proceedings ACMGIS99*, Kansas City, MO, pp. 152-157.

B. Buttenfield, 1989, "Scale-Dependence and Self-Similarity in Cartographic Lines", *Cartographica*, 26(1), pp.79-100.

B. Buttenfield, 1999, "Progressive Transmission of Vector Data on the Internet: A Cartographic Solution", *Proceedings 18th International Cartographic Conference*, Ottawa, Canada.

B. Buttenfield and R. McMaster, 1991, *Map Generalization: Making Rules for Knowledge Representation*, Longman, London.

B. Chazelle and H. Edelsbrunner, 1992, "An optimal algorithm for intersecting line segments in the plane", *Journal of the Association for Computing Machinery,* 39(1) pp. 1-54.

J. W. Cooley and J. W. Tukey, 1965, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Math. of Comput.*, 19, pp. 297-301.

G. Davis and A. Nosratinia, 1998, "Wavelet-Based Image Coding: An Overview", *Applied and Computational Control, Signals, and Circuits*, Vol. 1, No. 1.

G. Davis, 1998, "A Wavelet-based Analysis of Fractal Image Compression", *IEEE Transactions on Image Processing*, 7(2), pp. 141-154.

G. Davis, 1996, "Implicit Image Models in Fractal Image Compression", *SPIE Conference on Wavelet Applications in Signal and Image Processing IV*, Denver, TX.

L. De Floriani and E. Puppo, 1995, "Hierarchical Triangulation for Multiresolution Surface Description", *ACM Transaction on Graphics,* 14(4), pp. 363-411.

L. De Floriani P. Magillo and E. Puppo, 1998, "Efficient Implementation of Multi-Triangulations" *Proceedings IEEE Visualization '98*, Research Triangle Park, NC, USA, pp. 43-50.

L. De Floriani, P. Marzano, E. Puppo, 1993, "Spatial Queries and Data Models", in A.U. Frank, I. Campari (eds.), *Spatial Information Theory – A theoretical basis for GIS,* Lecture Notes in Computer Science N. 716, Springer-Verlag, pp. 113-138.

G. Dettori and E. Puppo, 1996, "How generalization interacts with the topological and geometric structure of maps", *Proceedings Spatial Data Handling '96*, Delft, The Netherlands.

D. H. Douglas and T.K. Peucker, 1973, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", *The Canadian Cartographer,* 10, 2, pp. 112-122.

M. Eck, T. DeRose, T. Duchamp, and H. Hoppe, 1995, “Multiresolution analysis of arbitrary meshes”, *Computer Graphics* (*Proceedings SIGGRAPH ’95)*, pp. 173-182.

M. Egenhofer, 1994, “Spatial SQL: A query and presentation language”, *IEEE Transactions on Knowledge and Data Engineering*, 6, pp. 86-95.

M. Egenhofer, A. Frank, and J. P. Jackson, 1989, “A topological model for spatial databases”, *Proceedings SSD89*, Lecture Notes in Computer Science N. 409, pp. 271-286.

M. Egenhofer and R. Franzosa, 1991, “Point-set topological spatial relations”, *International Journal of Geographic Information Systems*, 5(2), pp. 161-174.

Y. Fisher (ed.), 1995, *Fractal Image Compression: Theory and Application to Digital Images*, Springer Verlag, New York.

D. Flewelling, 1997, “Comparing Subsets from Digital Spatial Archives: Point Set Similarity”, *Ph.D. Thesis*, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME.

D. Flewelling and M. Egenhofer, 1999, “Using Digital Spatial Archives Effectively”, *International Journal of Geographical Information Science*, 13, 8, pp. 1-8.

A. Frank and W. Kuhn, 1986, “Cell graph: a provable correct method for the storage of geometry”, *Proceedings Spatial Data Handling ‘86*, Seattle, WA.

R. C. Gonzales and R. E. Woods, 1993, *Digital Image Processing*, Addison-Wesley.

M. F. Goodchild and J. Proctor, 1997, “Scale in Digital Geographic World”, *Geographical & Environmental Modelling*, 1, 1, pp. 5-23.

R. Güting and M. Schneider, 1995, “Realm-Based Spatial Data Types: the Rose Algebra”, *VLDB Journal*, 4, pp. 243-286.

C. Hofmann, 1999, “A multi tier framework for accessing distributed, heterogeneous spatial data in a federation based EIS”, *Proceedings ACMGIS99*, Kansas City, MO, pp. 140-145.

S. de Hoop, P. van Oosterom, and M. Molenaar, 1993, "Topological Querying of Multiple Map Layers", in A. U. Frank, I. Campari (eds.), *Spatial Information Theory - A theoretical basis for GIS*, Lecture Notes in Computer Science N. 716, Springer-Verlag, pp. 139-157.

H. Hoppe, 1996, "Progressive Meshes", *Proceedings SIGGRAPH96*, pp. 99-108.

B. Junger and J. Snoeyink, 1998a, "Importance Measures for TIN Simplification by parallel decimation", *Proceedings Spatial Data Handling '98*, Vancouver, Canada, pp. 637-646.

B. Junger and J. Snoeyink, 1998b, "Selecting Independent Vertices For Terrain Simplification", *Proceedings WSCG '98*, Plzen, Czech Republic.

P. Kern and J. D. Carswell, 1994, "An Investigation into the Use of JPEG Image Compression for Digital Photogrammetry: Does the Compression of Images Affect Measurement Accuracy", *Proceedings EGIS94*, Paris, France.

R. Kramer, R. Nikolai, A. Koschel, C. Rolker, P. Lockermann, A. Keitel, R. Legat, and K. Zirm, 1997, "WWW-UDK: A Web-Based Environmental Metainformation System", *ACM SIGMOD Record*, 26(1).

V. Kumar, A. Bugacov, M. Countinho, and R. Neches, 1999, "Integrating Geographic Information Systems, Spatial Digital Libraries and Information Spaces for Conducting Humanitarian Assistance and Disaster Relief Operations in Urban Environments", *Proceedings ACMGIS99*, Kansas City, MO, pp. 146-151.

R.B. McMaster and K.S. Shea, 1992, *Generalization in Digital Cartography*, Washington DC, Association of American Geographers.

J. Morlet and A. Grossman, 1984, "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape" - *Siam. J. Math, Anal.*, 15(4), pp. 723 –736.

J.C. Müller, R. Weibel, J.P. Lagrange, and F. Salgé, 1995, "Generalization: State of the Art and Issues", in GIS and Generalization: Methodology and Practice, J.C. Müller, J.P. Lagrange, R. Weibel (eds.), Taylor and Francis, pp. 3-7.

OGC (OpenGIS Consortium), 1996a, *The OpenGIS guide, introduction to interoperable geoprocessing, Part 1*, http://www.ogis.org/guide/guide1.html.

OGC (OpenGIS Consortium), 1996b, *The OpenGIS abstract specification*, http://www.opengis.org/public/abstract.html.

F. Preparata and M. Shamos, 1985, *Computational Geometry: An Introduction*, Springer-Verlag.

E. Puppo and G. Dettori, 1995, "Towards a formal model for multiresolution spatial maps", in M. J. Egenhofer, J. R. Herring (eds.), *Advances in Spatial Databases,* Lecture Notes in Computer Science N. 951, Springer Verlag, pp. 152-169.

U. Rauschenbach, and H. Schumann, 1999, "Demand-driven Image Transmission with Levels of Detail and Regions of Interest", *Computers and Graphics*, 23(6), pp. 857-866, 1999.

A.H. Robinson, R.D. Sale, J.L. Morrison, and P.C. Muehrcke, 1984, *Elements of Cartography*, John Wiley, New York, 5th edition.

A. Saalfeld, 1999, "Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm", *Cartography and Geographic Information Science,* 26(1), pp. 7-18.

H. Samet,1990a, *The Design and Analysis of Spatial Data Structures,* Addison-Wesley, Reading, MA.

H. Samet,1990b, *Applications of Spatial Data Structures,* Addison-Wesley, Reading, MA.

M. Sondheim, K. Gardels, and K. Buehler, 1999, "GIS Interoperability", in *Geographical Information Systems: Principles, Techniques, Management and Applications,* P. Longley, M.F. Goodchild, D.J. Maguire and D.W.Rhind (eds.), Second Edition, Cambridge, GeoInformation International, pp. 347-358.

B. S. Srinivas, R. Ladner, M. Azizoglu, and E. A. Riskin, 1999, "Progressive Transmission of Images using MAP Detection over Channels with Memory", *IEEE Transactions on Image Processing*, 8(4), pp. 462-75.

E. C. Titchmarsh, 1948, *Introduction to the Theory of Fourier Intervals,* Oxford University Press, New York.

UCGIS (University Consortium for Geographic Information Science), 1996, *The UCGIS research agenda: interoperability*, http://www.ucgis.org.

M. van Kreveld, 1996, "Digital elevation models and TIN algorithms", *Notes for CISM Advanced School on Algorithmic Foundations of GIS.*

H. Wactlar, M. Christel, Y. Gong, and A. Hauptmann, 1999, "Lessons Learned from Building a Terabyte Digital Video Library", *IEEE Computer*, 32 (2): pp. 66-73.

R. Weibel and G. Dutton, 1999, "Generalising spatial data and dealing with multiple representations", in *Geographical Information Systems: Principles, Techniques, Management and Applications*, P. Longley, M.F. Goodchild, D.J. Maguire and D.W.Rhind (eds.), Second Edition, Cambridge, GeoInformation International, pp. 125-155.

M. Worboys, 1992, "A generic model for planar geographic objects", *International Journal of Geographic Information Systems*, 6(5), pp. 353-372.

Y. Zhao and B. Yuan, 1996, "A hybrid image compression scheme combining block-based fractal coding and DCT", *Signal Processing: Image Communication*, 8(2), pp. 73-78.

D. Zorin, P. Shroder, and W. Sweldens, 1997, "Interactive multiresolution mesh editing", *Computer Graphics* (*Proceedings SIGGRAPH '97)*, pp. 259-268.