7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO
C. Bauzer Medeiros (ed.), November 1999.

Progressive Vector Transmission

Michela Bertolotto Max J. Egenhofer
National Center for Geographic Information and Analysis
Department of Spatial Information Science & Engineering, University of Maine, Orono, ME, 04469, USA
{michelab,max}@spatial.maine.edu

ABSTRACT

Progressive transmission of raster images over the World-Wide Web has been successfully applied to provide the user with coarser versions of the data before downloading a complete image. On the other hand, in the vector domain progressive transmission is challenging. Increasing the level of detail of a vector dataset does not simply imply adding pixels to it. In this paper, we discuss issues related to the progressive transmission of vector maps. We also describe a model for multiple representations of maps that can be transmitted progressively.

Keywords

Progressive transmission over the Web, multiple representations, levels of detail.

1. INTRODUCTION

Interleaving used for raster data transmission over the World-Wide Web corresponds to the transmission of 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 image is then progressively completed by adding new pixels. This process is particularly useful when trying to access remote data through a slow communication link (e.g., a modem) or when datasets are particularly large. So far, implementations have focused on progressive transmission of raster data through the Web.

The transmission of vector data is generally done by means of a one-step long process. 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. The user attempting to download a vector file needs to wait for the fully detailed version without having the possibility to start working with a coarser version of the data stored in a smaller file. We use the term detail to mean both the amount of entities contained in the file and the information stored about such entities, for instance, their geometry. Therefore, for example, changes of detail in a representation 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).

Our intent is to define a framework for progressive vector data transmission. This problem can be seen as part of a more general research area: the creation of summaries to help the user query very large databases. A type of summaries commonly used for both vector and raster datasets is metadata in the form of textual descriptions. However, in some applications, summaries are needed not only to convey the idea of what is contained in the database, but also as work datasets themselves. For example, during remote access, temporary versions of the fully detailed dataset may be used to perform some preliminary analysis or manipulation. Metadata are only a descriptive kind of summary and, therefore, cannot be used as substitute working datasets. For this purpose, subsetting is a common technique: a meaningful sample of the data is provided instead of the whole set. For instance, in digital image archives, thumbnail images are generated to convey the suitability of high resolution images.

In the vector domain, a method for providing subset-based summaries generates multiple representations of a dataset, each corresponding to a different level of detail. For instance, summaries of digital vector maps are created in cartographic generalization by extracting less detailed vector maps from a fully detailed map [7]. The topic of this paper is the creation of this kind of summaries for vector data to be used as preliminary work datasets during progressive transmission over the Web. Ideally, the summaries would be generated on-line by fulfilling the user's requirements. However, this is a non-trivial problem in the case of vector datasets.

In the remainder of this paper, we discuss the challenges related to progressive vector transmission (Section 2). An example of a model for multiple representations for vector maps to be stored on the server site is presented in Section 3. Other examples could be provided using models defined for multiple representations of triangular meshes [1][4][6]. Section 4 is dedicated to transmission strategies. In Section 5 vertical links are added to the model described in Section 3 in order to facilitate the reconstruction of a complete representation corresponding to an intermediate level of detail. Finally, an example of a multiple representation sequence with four levels of detail is provided in Section 6, followed by our conclusions in Section 7.

2. CHALLENGES IN PROGRESSIVE VECTOR TRANSMISSION

Progressive transmission of vector data is currently needed when very large datasets must be accessed remotely or when data must be transmitted across slow communication links (e.g., a modem). Communication lines (e.g., fiber optic cable) are getting faster and fairly soon we can expect to be able to download large amounts of data without long waits. On the other hand, the current technology still does not adequately support wireless communication. Mobile computing will be one of the major topics of the 21st century research. However, transmission of large amounts of data across mobile devices will still be an impediment in the foreseeable future. In this setting, progressive vector transmission will be very useful. For instance, geographers who want to access a vector map while performing a data collection directly in the field, will certainly benefit from the prompt availability of a coarse version of the map on which they will perform initial operations while waiting for the fully detailed file to be accessible.

Progressive transmission of raster data is relatively simple, since just adding pixels to an incomplete image generates a more refined version of it. On the other hand, progressive vector transmission is challenging, because the creation of a suitable representation at finer detail is a far more complex process than just adding some entities to a vector dataset. Likewise random subsampling of entities in vector format does not generate a consistent representation at coarser detail. To maintain consistency (e.g., preserving constraints on overlappings) between maps at different detail in cartographic generalization [5], some spatial entities must be added while others must be deleted or replaced. For example, if at a lower level of detail only the map of the major road network is stored (Figure 1a) and at a higher level minor roads must also be added to the map (Figure 1b), there will be problems of intersection and overlapping. This may imply the addition of an intersection point as well as the replacement of a whole road with two separate branches of road (e.g., “road to the left of the intersection” and “road to the right of the intersection”).

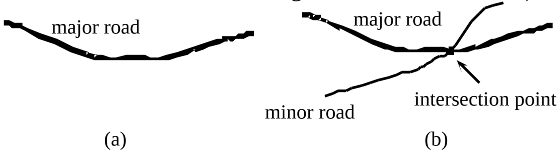

Figure 1: Intersection between roads at different levels of detail.

A way to enable the progressive transmission of vector files is to pre-compute a sequence of consistent representations at lower levels of detail on the server site from the fully detailed representation and to transmit them in order of increasing detail (Figure 2).

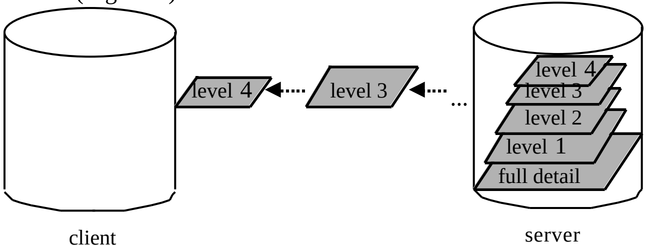

Figure 2: Representations at different levels of detail are transmitted to the client in order of increasing detail.

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. Therefore, both time and disk space can be saved.

An interesting property of progressive transmission of raster data is the fact that only the increments (i.e., sets of pixels) are transmitted and added to the currently displayed image 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. However, increments between two consecutive levels of detail 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 into 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.

The progressive vector transmission process presents a two-sided problem. On one side, the sender needs methods for building, manipulating, and transmitting a sequence of representations at different levels of detail. On the other side, the receiver must be provided with a set of operations for visualizing as well as updating and integrating the transmitted levels. To enable progressive vector transmission, the following functionalities are required:

- *on the server site*: a preprocessing task must be performed in order to build a sequence of representations at different levels of detail, as well as a progressive transmission technique 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.

3. MULTIPLE REPRESENTATIONS

In this section we describe a method for defining a sequence of vector map representations at different levels of detail that can be transmitted progressively. We define a *map* in vector format as an overlayed set of points, simple lines and (possibly multiply connected) regions described by collections of 0-, 1-, and 2-cells, i.e., cell complexes, where semantic information can be attached in the form of attributes of the cells. The method we describe was initially presented in [8] and then further formalized in [2] and [3], where a set of generalization operators have been shown to be minimal and sufficient for defining consistent transformations of maps. Only topological changes are possible by applying such operators: metric and semantic changes are not being taken into account. The operators defined are:

- **Line contraction**: contraction of an open line (including its endpoints) to a point (Figure 3a).

- **Region contraction**: contraction of a simply connected region (with its boundary) to a point (Figure 3b).
- **Region thinning**: a region (and its bounding lines) is reduced to a line (Figure 3c).
- **Line merge**: fusion of two lines sharing an endpoint into a single line (Figure 3d).
- **Region merge**: fusion of two regions sharing a boundary line into a single region (Figure 3e).
- **Point abstraction**: elimination of an isolated point inside a region (Figure 3f).
- **Line abstraction**: elimination of a line inside a region (Figure 3g).

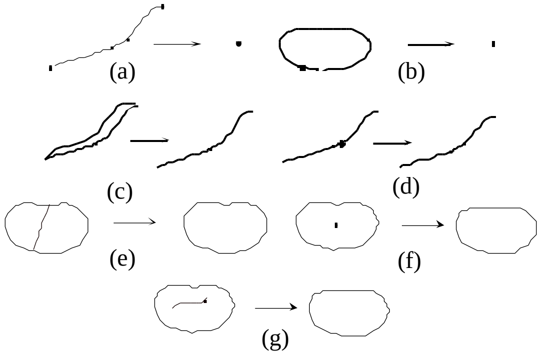

Figure 3: Generalization operators.

These operators have been formalized by means of functions between the cell complexes describing the original and the resulting map [2]. Contractions and thinning correspond to a decrease of dimension for a group of entities, merge processes group two entities of the same dimension into a single one, while abstractions 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.

In the context of map generalization [7][9], an important problem is guaranteeing that the resulting representation be consistent with the source fully detailed dataset [5]. 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.

In the case of vector data transmission between two remote sites, the preservation of consistency is essential, because the client has no way of checking the consistency. This should be done on the server site, i.e., the datasets that are being sent over should be consistent (either implicitly by construction or by means of *a posteriori* checks). When the previously described operators are used, this problem does not arise because it has been shown that they represent a minimal and sufficient set of functions that generate by composition only consistent transformations for modifying the level of detail of a map [2][3]. In particular, they allow to generate all map transformations that preserve the boundary of entities and whose inverse image preserves connectivity (thus, they do not allow to model aggregation of non-connected entities). Spatial relations are consistently transformed. Therefore, intrinsically consistent multiple representation sequences are built on the basis of such operators.

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, such that

$$f_1 : M_0 \rightarrow M_1, f_2 : M_1 \rightarrow M_2, \ldots, f_k : M_{k-1} \rightarrow M_k$$

and $f = f_k \circ f_{k-1} \circ \ldots \circ f_2 \circ f_1 : M_0 \rightarrow M_k$ 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, i.e., it is a multiple representation sequence.

This model is built on the basis of operators that perform only topological changes. However, in real implementations also changes in the geometric shape are commonly required, such as simplifications of lines. Therefore, this model should be extended to include this kind of operations.

An important issue relates to the efficient storing and transmission of such sequence. Entities that are preserved throughout the levels should be stored only once in the sequence instead of redundantly encoding them at each level. A straightforward way to avoid redundancy is storing the entire dataset 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 the currently displayed level, once the transmission has been completed, to use, manipulate, and query it. For this purpose, new algorithms are required for extracting a representation at an intermediate level in the sequence because the client can decide to stop the transmission at any level. 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 only vector data 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 [8]. 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* (Figure 4).

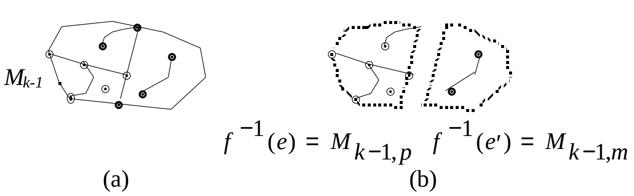

Figure 4: (a) Two levels of a multiple representation sequence, and (b) an example of sequence encoding, where preserved entities are not represented at the more refined level.

4. TRANSMISSION OF DIFFERENT LEVELS OF DETAIL

In this section we discuss the process of transmitting a multiple representation sequence between two remote sites. 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.* The process starts with the server transmitting the representation corresponding to the first level of detail (i.e., the least detailed). Let *t* be a binary operation 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).* Upon completion of the transmission of the first representation, 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), …, t(mr(d,k),1).*

In the encoding structure for the model described in section 3, in order to avoid unnecessary repetitions for storage saving purposes, 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.* This model saves space on the server site and speeds up the transmission to the client site. On the server site the coarsest representation *mr((d,k),k)* is completely stored, while only portions of the subsequent representations are stored in the subsequent levels. Such portions are the transmitted packages. 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 the point of view of the reconstruction of the dataset, *ad hoc* integration techniques must be developed. This issue is discussed in the following section.

5. ADDING VERTICAL 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 *vertical* or *intra-level* links.

Keeping vertical links facilitates the reconstruction of a representation at an intermediate 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. 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 vertical links allows for efficient browsing across the levels without having to query entire levels. A query can be 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 vertical 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 4, links between entities in consecutive levels are provided by the transformation functions defining the sequence. A function f_i maps a more detailed map M_{i-1} onto a less detailed map M_i. Such transformation functions perform *generalizations* of maps. Here we want to deal with the inverse operation, i.e., the *refinement* of maps. The encoding model we consider stores the complete set of entities for the coarsest representation and only the refinement of each entity as well as newly introduced entities in subsequent levels. Such a model has been defined on the basis of inverse images of entities through the transformation functions defining the sequence. The model enriched with vertical links has a tree-like structure. 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.

We recall that transformation functions are obtained as compositions of atomic operators. Thus, we consider inverse images of entities through atomic operators. In [2], the inverse image of an entity in the co-domain of minimal 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 region merge operator), $\{r,p\}$ (i.e., the domain of point abstraction operator), $\{r,l\}$ (i.e., the domain of line abstraction operator).

The inverse image of a line can only be one of the following sets: $\{p,l,l'\}$ (i.e., the domain of line merge operator), $\{l,l',r\}$ (i.e., the domain of region thinning operator). Finally, the inverse image of a point can only be one of the following sets: $\{r,l,p\}$ (i.e., the domain of region contraction operator), $\{l,p,p'\}$ (i.e., the domain of line contraction operator).

Inverse (i.e., refinement) operations can thus be easily defined (see [3]). 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 already mentioned, keeping vertical links facilitates the reconstruction of the dataset corresponding to an intermediate level on the client site. Such a dataset is not stored explicitly on the server either. Integration operations are needed to reconstruct it 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.

For the hierarchical model described in this section, a representation corresponding to an intermediate level can be obtained by means of a visit of the tree. The visit will include all transmitted nodes up to the desired level: entities contained in a given node will be deleted and replaced by the set of entities corresponding to the child of such node. Spatial relations between two entities e and e' at level i can also be reconstructed on the basis of spatial relations between entities in the inverse image of e and e' at preceding levels. For this purpose, in addition to the increments, the transmitted package must include also vertical links. In the following section, a multiple representation sequence is described with examples of operations that need to be performed on the server and client sites.

6. AN EXAMPLE

In this section, we describe an example of a multiple representation sequence with four levels of detail (Figure 5). 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 segments 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, generalization operations must be applied to the map at higher level of detail (level 1 in Figure 5). For example, to obtain level 2, the following operations are applied:

- region thinning of *major road* and *bridge,* and
- line simplification of the lines bounding *paper mill.*

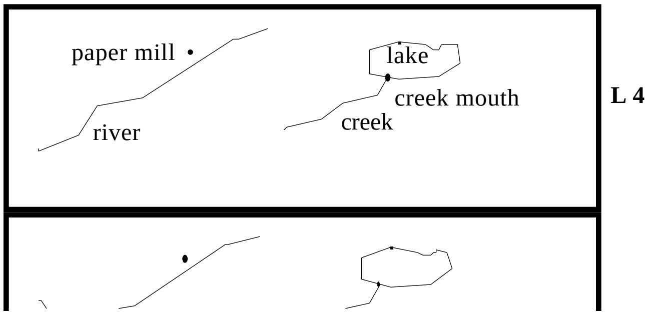

Figure 5. A multiple representation sequence with four levels.

Figure 6 illustrates the structure described in section 3 for the efficient encoding of the multiple representation sequence of Figure 5. The set of entities in level i (for $1 \le i \le 3$) is 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. 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.*

The above structure could be stored on the server site and used for progressive transmission. The client site would receive the increments for each transmitted level. Since only level 4 is completely represented, the reconstruction of each intermediate level is required on the client site. Thus, for example, to reconstruct level 3 from level 4 and the increment for level 3, 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;
- the complete set of spatial relations for level 3 can be 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 (stored in the increments).

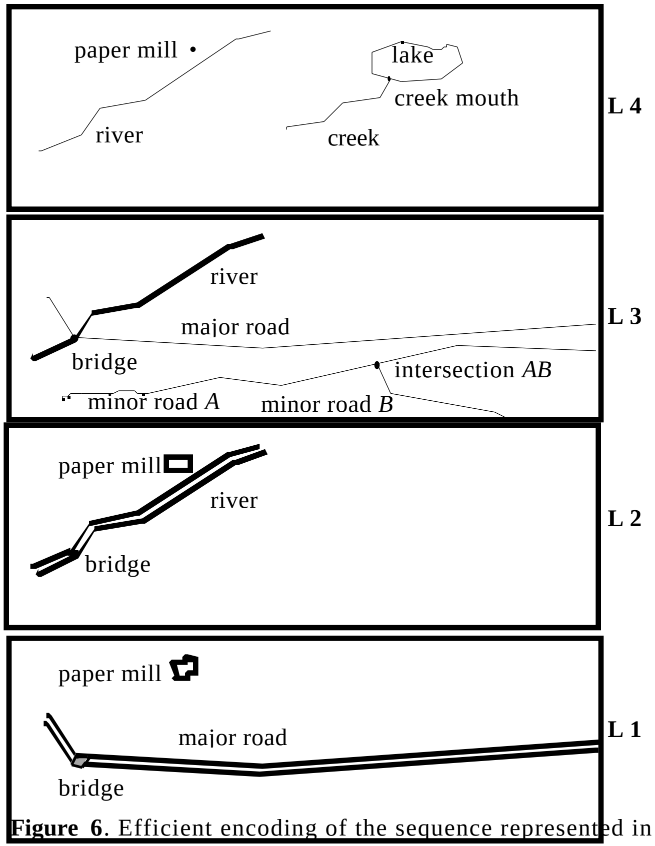

Figure 6. Efficient encoding of the sequence represented in Figure 5.

7. CONCLUSIONS

This paper discusses the need for progressive vector transmission and its inherent challenges. Extracting a consistent representation at a lower level of detail from a vector dataset is a complex and time-consuming operation that cannot be performed on-line during progressive transmission. A solution is to pre-compute 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. As an example, we have described the model for plane map representation proposed in [8]. The model has been shown to be intrinsically consistent [2]. Our future goal is to define other 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. Other technical issues such as the definition of rules to automatically generate multiple representations and the incorporation of semantics will be taken into account in the design of a prototype implementation.

During query processing, multiple representation sequences benefit from the addition of vertical links between different representations of the same entity at different levels of detail. Within this setting, a further development of our current research is the investigation of how to transmit vertical links to efficiently perform spatial queries at different levels of detail.

8. ACKNOWLEDGMENTS

Michela Bertolotto's work is supported by the Italian National Research Council under grant number 106701-00/97/10003. Max Egenhofer's work is partially supported by the National Science Foundation under NSF grant numbers SBR-9700465, IRI-9613646, BDI-9723873, 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; Bangor Hydro-Electric Co.; and a Massive Digital Data Systems contract sponsored by the Advanced Research and Development Committee of the Community Management Staff.

9. REFERENCES

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

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

[3] Bertolotto, M., Dettori, G., and Puppo, E. A Combinatorial Framework for Managing Levels of Detail in Geographic Databases, 1999 (in preparation).

[4] De Floriani, L., Magillo, P.,and Puppo, E. Efficient Implementation of Multi-Triangulations.Proceedings IEEE Visualization 98, Research Triangle Park, NC, USA, 1998.

[5] Egenhofer, M., Clementini, E., Di Felice, P. Evaluating inconsistencies among multiple representations. International Journal of Geographical Information Systems, 8, 2, 1994, 129-142.

[6] Junger, B., and Snoeyink, J., Importance Measures for TIN Simplification by Parallel Decimation. Proceedings Spatial Data Handling '98, Vancouver, Canada, 1998, 637-646.

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

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

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