Multi-Resolution Geospatial Data in Mobile System

Follin Jean-Michel, Bouju Alain, Bertrand Frédéric, Boursier Patrice

Laboratoire Informatique, Image, Interaction, Université de La Rochelle
E-mail:jean-michel.follin@univ-lr.fr

Abstract
Geographic Information Systems (GIS) have to support multi-resolution data in order to represent the real world at different scales. This need is also observed in the emerging field of mobile GIS. In this paper, we propose solutions for the management and visualization of multi-resolution vector data in a mobile spatial information visualization system. We first review the basic priciples of mobile GIS and management of multi-resolution data, and we mention various works that have already been conducted in the areas pertaining to our own research and development works. We then present the client-server architecture and the management approach adopted in the system. Our aim is to reduce the amount of data exchanged between the client and the server. Our solution is based on the use of increments in a multiscale database. In our database architecture, datasets for different predefined scales are precomputed and stored on the server side. Increments correspond to the difference between two datasets with different resolutions and are transmitted in order to increase or decrease the level of detail of information on the client upon request. They allow reusing data which are already present on the client side. They imply to take into consideration the different generalization operators, their mapping configurations and their modifications on objects representations. Finally, we describe our approach for presenting multi-resolution data: an adapted one relying on "intelligent zoom". It consists in measuring the density of data at different resolutions and for different scales, and finding a balanced solution taking account of resolution and scale which respects the well-known "principle of constant density of data".
Keywords
Multi-Resolution Data, Client-Server, Embedded System, Mobile GIS

I. INTRODUCTION

Nowadays, the development of mobile technologies (i.e. communications and devices) allows a mobile user to have access to geographic information from anywhere through web mapping applications. The served data have to be aware of the user's location and circumstances (Krum, et al., 2001), i.e. they have to be context-aware. For instance, detailed information is required in a town center and generalized one is sufficient on a highway.

Our work aims at managing multi-resolution spatial data in a mobile map visualization system (Follin, et al., 2003a). We propose to reuse as much as possible data which are already available on the client side in order to reduce data transfer between client and server. Digital vector maps with different Levels of Detail (LoDs) are used. This "LoD approach" is based on the use of increments which allow to rebuild required LoD representation of an object from the LoD of the same object already available on the client side. In order to adapt data to be displayed to the users needs, we adopt an "intelligent zoom" approach respecting the principle of equal density of data.

In section II, we review the basic priciples of mobile GIS and management of multi-resolution data, and we mention various works that have already been conducted in the areas pertaining to our own research and development works. In section III, we present the framework that we propose for allowing multi-resolution navigation in an embedded spatial information visualization system. It is based on the definition of LoD objects and increments. Finally, in section IV, we propose a methodology for map data visualization. We give the results obtained with some experiments where we adapted visualized LoD data to the scale of representation.

II. MOBILE GIS AND MANAGEMENT OF MULTI-RESOLUTION DATA

A. Basic principles

Two main approaches may be distinguished for the development of Internet or web-based GIS, depending on the distribution of workload between the client and the server, which leads to two different kinds of solutions and systems. In the first case(server side solution), the main part of computation is made on the server, while in the second one(client side solution), data and software are transferred to the client where all processing and computation are done (Huang, et al., 2001). The architecture adopted in our system is rather a server side solution: client is limited to visualization related tasks (see section III.A). A review of the current development of mobile devices and telecommunications, and an analysis of the requirements of the mobile use of geospatial data are carried out in (Nissen, et al., 2003).

The use of an incremental strategy allows reducing the amount of data transferred from server to client by reusing, when possible, already locally available data. Data and transfer models suited to multi-resolution data in an embedded context and preliminary experimentations have been described in

1082-4006/04/1002-106$5.00

(Follin, et al., 2003b).

B. Management of multi-resolution data

Data and transfer models have been presented in (Stockus, et al., 2001) and extended to multi-resolution data through a LoD approach combined with the use of increments in (Follin, et al., 2003a). There exist two main solutions for providing a mobile user with multi-resolution data (Cecconi, et al., 2002) (Figure 1):

- *on-the-fly generalization* (A) (or real-time generalization) where different LoD representations are computed in real-time from a detailed map (G) and transmitted (T) to client upon a query q.
- LoD approach (B) (or multi-representation approach) where different LoD representations are pre-computed and stored on the server side. After a query q, the appropriate LoD is sent to client (T).

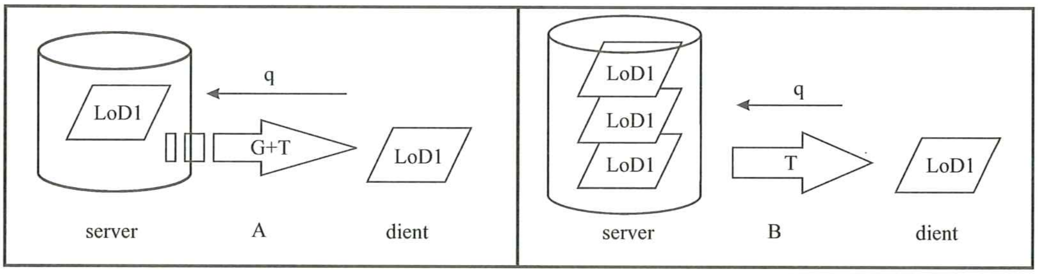

Figure 1. Two approaches for managing multi-resolution data in a mobile context

These two approaches present drawbacks and advantages (Bédard, et al., 2001; Cecconi, 2003). They are resumed in Table 1 (where advantages and drabacks are respectively indicated by + and -).

Table 1. Drawbacks and advantages of the two approaches

	Data	Flexibility	Computational cost	Access to data	Good quality maps
On-the-fly generalization	+	+	-	-	-
LoD Approach	-	-	+	+	+

In a multi-representation approach, less calculation power is required on the server side and access between data levels is faster for client than in on-the-fly generalization. Furthermore better quality maps can be produced in the first approach. Indeed, more complex and computational expensive methods and algorithms can be used in the generalization when this one is accomplished off-line. On the other hand, with on-the-fly generalization data can be provided to user with more flexibility because LoDs are not defined at fixed levels.

For multi-representation, there is a need of existing data and consequently for more storage capacity on the server side. This brings up the problem of keeping database updated for all LoDs: if the base data is changed all the updates must be propagated to the other LoDs.

These two types of approaches are either combined or altered through different solutions in order to propose multi-resolution data in a Web or mobile context (see section II.C). Our approach relies on the notions of LoD objects and increments.

LoD representation and manipulated vertices

A *LoD representation* corresponds to a geographical object version defined for a specific scale interval: its "valid" (or adapted) scale range. As our study focuses only on geometric aspect (and not on attribute), we consider it is a geometric object modelled by one among six two dimensional objects: Point, Polyline or Region for simple (i.e. connected) objects, and MultiPoint, MultiLine and MultiRegion for complex (i.e. not connected) ones. LoD representation of an object o at a level n can be noted o^n.

A *manipulated vertex* is defined for a given transition from level n to either level n+1 or n-1 representation of the same object o. It corresponds to the parts of a vertex V_i^j that are manipulated by geometric operators op_i. V_i^j is the vertex at index i of the LoD j representation of object o. Geometric operators can manipulate either index only, or both index and coordinates of a vertex.

Geometric operators

A *geometric operator* allows to perform changes on objects geometries after a request of transition in the refinement direction (i.e. from LoD n to n-1) or in the generalization one (i.e. from LoD n to n+1). So it is different from a generalization operator which aims at deriving a coarser representation of an object from another by using resolution related parameters. A geometric operator op_i is chosen among set $Op = \{insert, keep, remove, move\}$.

A geometric operator takes into account mapping of data and depends on generalization operators involved in production of less detailed data. So choice of geometric operator is based on observation of the different mapped LoD representations of the same objects. Mapping configuration and generalization operators need to be defined.

Mapping of data and generalization operators
The *mapping configuration* corresponds to the number of mapped LoD representations of same real world entities (when objects are represented at two different LoDs). According to (Bertolotto, et al., 2001; Dettori, et al., 1996), three categories of *generalization operators* can be considered:

- *metric operators*, for handling changes related to simplifications and decreases in size, i.e. affecting the shape of objects,
- *topological operators*, for handling changes in dimension and complexity of objects,
- *semantic operators*, for handling changes related to attributes (which will not be seen here).

Topological changes indirectly imply metric changes but not vice versa. Operators are applied to object geometries. We can call intra-type (resp. inter-type) operator an operator keeping (resp. changing) object's spatial type. The inverse operations of generalization are the *refinement* functions (Bertolotto, et al., 2001). According to mapping configuration, generalization and refinement operators can be seen as different spatial entities mapping procedures (Ai, et al., 2001):

- *1:1* spatial entity mapping procedures map one LoD representation of an object to another LoD representation of the same object,
- *1:n* mapping operators match one LoD representation to a composite entity representing the same real-world object,
- *n:m* mapping procedures can be seen as a *1:1* matching case of cluster object.

The *mapping operators* which are described below concern generalization (1) and refinement (2). They are shown in Figure 2. They are presented in order to give an idea of the variety of representation changes in a multi-resolution context. They correspond to *1:1*, *1:n* and *n:m* geometric operators. For the moment only *1:1* geometric operators are implemented in our system. They take one object as input and send back the same and modified object as output.

Metric operators.
Metric operators are *1:1* mapping intra-type operators.
Simplification operators (A) eliminate details of a polyline or region by selecting ("filtering") a subset of its original points which is considered more representative of its essential shape. Corresponding geometric operators are "*keep*", "*remove*" or "*insert*". The first two operators are used for conservation or removal (referring to generalization), and the third one for addition (referring to refinement) of points to a polyline or a region.

Enhancement operators (B) are used to enlarge objects of all geometries. They include enlargement operators that "enlarge object equally in each direction" and caricature (or exaggeration) operators that "enlarge only some parts of objects" (AGENT, 1999).

Aesthetic refinement operators (C) are applied on an object of any type to improve its visual impression by altering its geometry. They include smoothing operators that "reduce sharp angularity from objects having smooth shapes", and rectification operators that "rectify the geometry from objects which are expected to have a rectangular shape" (AGENT, 1999).

Displacement operators can be applied to two or more distinct objects (external displacement) or to components (points or segments) of a polyline or region (internal displacement). It resolves spatial conflicts (too close or overlapping objects). "In this procedure, the operated object is a spatial entity pair rather than an independent entity" (Ai, et al., 2001). The corresponding geometric operator is "*move*", which uses coordinates shift of a vertex between a LoD *m* and a LoD *n* in order to displace it.

Topological operators.
Contraction (or collapse) operators (D) reduce the dimensionality of objects. They perform changes either from region to polyline or point, or from polyline to point. *Expansion* operators perform inverse transformations. They are *1:1* mapping inter-type operators.

Selection/elimination operators (E) can be applied to object collections (e.g., a network of polylines) to remove objects according to theme, feature type, or conflict type (for example, objects of less importance in Figure 2). Inverse operator of elimination, used in a refinement direction, is *addition* of objects. Elimination and addition are intra-type *n:m* mapping operators. Various methods can be adopted for selection/elimination process: for example Topfer's radix law (see section III.B) or structure-based approach like (Jiang, et al., 2004).

Aggregation operators include (AGENT, 1999)

- *1:n* mapping operators of amalgamation (fusion and merge) and combining,
- and *n:m* mapping operator of typification.

Amalgamation intra-type operators (F) of fusion or merge correspond to fusion of at least two entities (connected in case of fusion, disjoint in case of merge) of same dimension (polylines or regions) in one entity. Object *split* operator corresponds to splitting of one entity into two entities of same dimension.
Combining inter-type operators (G) combine a set of objects of a same geometric type (points or polylines) to one object of higher dimensionality (polyline or region).
Typification operators (H) change a set of discrete objects (points, polylines or regions) in a smaller set of objects with similar structural characteristics. These structuring processes are often the combination of several basic operators (selection, aggregation, displacement, and simplification). They are *n:m* mapping intra-type operator (typified objects in Figure 2 are bends and buildings).
These various generalization operators are used for producing different LoD representations which can then be used for computing increments.

Increments.

Increments can be defined as geometric operators associated to the vertices which are processed between two consecutive LoDs of a single object or dataset. In the case of *1:1* mapping configurations, the increment allowing transition from LoD n to LoD n-1 (resp. LoD n+1) representation of object o can be noted by $Inc(o,n \rightarrow n\text{-}1)$ (resp.$Inc(o,n \rightarrow n+1)$), such that:

$$Inc(o,n \rightarrow n \pm 1) = \{(op_i, V_i^j) \mid op_i \in Op, V_i^j \in \mathrm{V}_{\mathrm{Inc}}\}$$

where o is a Polyline or Region object,$n \rightarrow n \pm 1$ is a LoD transition either from level n to level n-1, or from n to n-1,op_i is a geometrical operator defined in Op (see section II.B Geometric operators),and V_i^j is the part of vertex processed between o^n and o^{n-1} (resp. o^{n+1}).

Increments are used through function r or g to produce a more refined or generalized version of an object o^n:

$$r(o^n, Inc(o,n \rightarrow n-1)) = o^{n-1}$$
$$g(o^n, Inc(o,n \rightarrow n+1)) = o^{n+1}$$

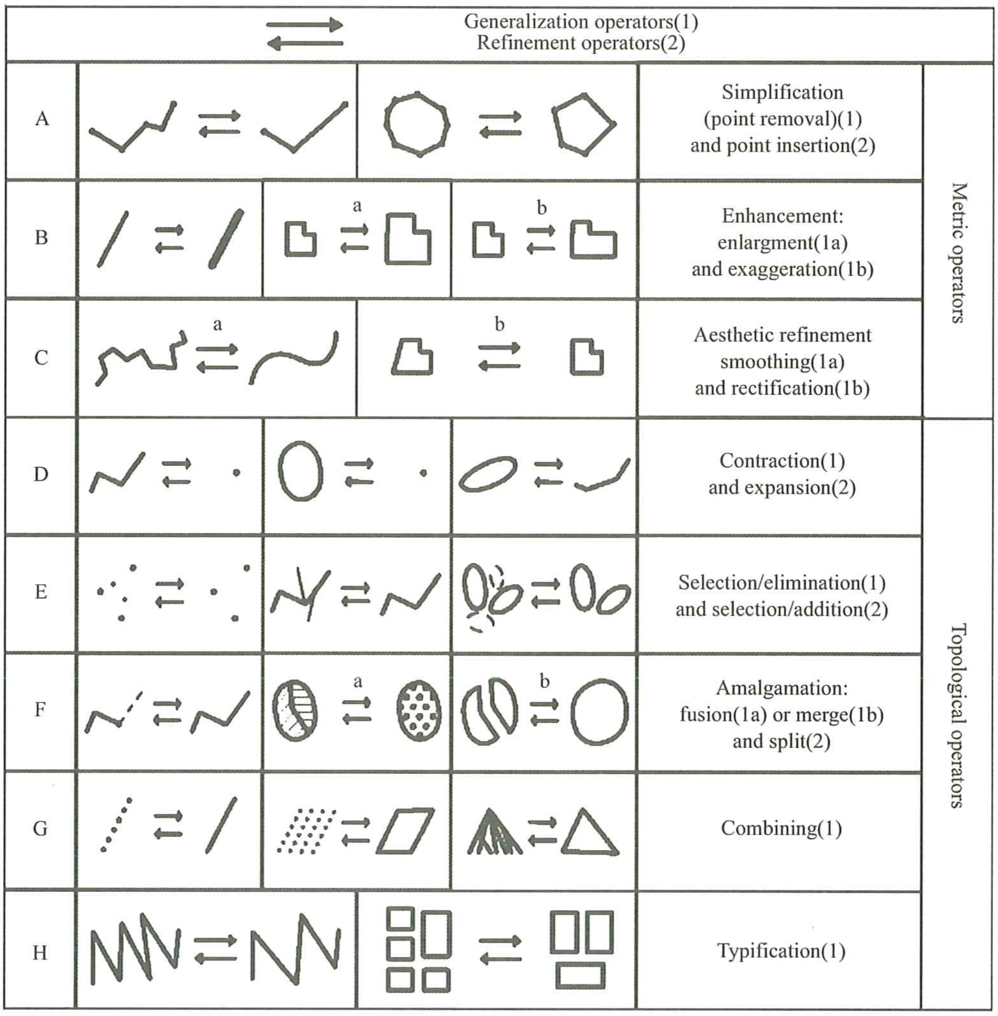

Figure 2. Different generalization and refinement operators

C. Related works

A real-time generalization (RTG) solution is proposed in GiMoDig project for providing mobile user with multi-resolution data (Letho, et al., 2001). It is based on the use of standards, namely the Geographic Markup Language (GML) and Scalable Vector Graphics (SVG). An approach which combines LoD representations for some feature classes and RTG for others is proposed in (Cecconi, 2003).

Another solution for transferring data through a limited bandwidth channel is the progressive transmission of vector data brought up by Bertolotto and Egenhofer (Bertolotto, et al., 2001), Brunner and Sester (Brenner, et al., 2003) and Persson (Persson, 2004). Such streaming approaches are inspired by progressive meshes for the simplification of triangulated surfaces (Hoppe, 1996). An encoding hierarchical structure for multi-resolution sequences is defined in (Bertolotto, et al., 2001): it includes intra-level links connecting different representations of the same entities at different levels. This allows a user to be provided with a more and more detailed map as data are progressively received from the server. In (Brenner, et al., 2003), "Elementary Generalization Operators" (EGO's) are proposed for the simplification of cartographic objects. They aim to provide continuous zoom for small mobile display while avoiding popping effect. Finally, a compressed, streaming and multi-resolution format for geographic vector data is presented in (Persson, 2004): it is called RaveGeo. During the generalization process, the difference between two resolutions (i.e. the increments) is computed and stored. These increments are sent by the server to the client in a streaming way.

D. Presentation and visualization of multi-resolution data

Visualization helps people to gain insight into the system that they study. *Scientific visualization* which is applied to scientific data, often physically based (for example, the human body, the earth or molecules) is commonly distinguished from *information visualization* which is applied to non-physical data (for example, financial data or abstract conceptions). *Geovisualization* integrates approaches from different fields (scientific visualization, information visualization, GIS) "to provide theory, methods and tools for the visual exploration, analysis, synthesis and presentation of geospatial data" (MacEarchen, et al., 2001). *Situational visualization*, introduced by (Krum, et al., 2001), is different from information visualization and scientific visualization because the user is "experiencing the situation firsthand and in real-time". So geovisualization in an embedded context can be defined as the exploration of geospatial data graphically, in order to enhance the user's experience and understanding of the surrounding world by using mobile computing resources.

Different approaches

There are three possible solutions to provide mobile users with maps at different scales on a single screen (Harrie, et al., 2002) (Figure 3):

- the *adapted presentation* approach, where maps at different scales are viewed in the full window and the user switches between them (A)
- the *key-map* approach, where maps at different scales are simultaneously shown: a large-scale map is presented in the main window and a small-scale one is shown in a little window (B)
- the *context+zoom* or *focus* approach, where data at different scales are integrated and presented on the same map: peripheral (i.e. contextual) information is displayed with less detail than information in the vicinity of the user's position (i.e. in the focus) (Frigioni, et al., 2003; Zipf, et al., 2002) (C)

In the latter approach, a selective reduction of information can be obtained by filtering and/or distortion techniques. The distortion oriented technique adopted in (Harrie, et al., 2002) is a variable-scale mapping function. This provides a map with "a circular cap where the scale is homogeneous and beyond which the radial scale constantly decreases to a threshold value". A special type of variable-scale maps is illustrated by *schematic maps* which focus on the rendering and design of navigation map resembling a hand-drawn sketch and depicting a path from one location to another (Agrawala, et al., 2001).

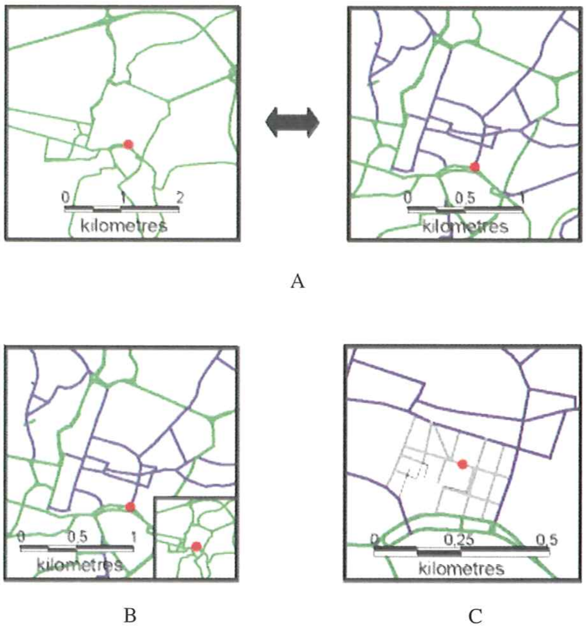

Figure 3. Solutions to provide a mobile user with multi-resolution data

Different kinds of zoom

Two operators are fundamental for the user's graphical navigation: the zoom operator for getting more details and the pan operator for moving the field of vision (Franck, et al., 1994). Zoom is necessary any time the user wishes to change either the visible range of data on the screen or its detail (Timpf, et al., 1997). In an embedded context, scale change can also be chosen based on the user's speed, available resources on the mobile device or communication speed.

Zooming can be done in three different ways.

- The *graphical zoom* is a simple change of scale factor without change of detail (Figure 4).
- The *content zoom* gives more detailed information in the window of interest (Figure 5).
- The *intelligent zoom* mimics the approach of the viewer to an object: when the field of vision becomes smaller, more details about the displayed object appear (Figure 6).

In a client-server framework, it is assumed that the intelligent (or dynamic) zooming is done by requests whereas graphical (also called static) zooming is provided by the client (Nissen, et al., 2003). In our system(presented in section III), we adopt an "intelligent zoom" approach which respects the known principle of "equal information density" (Timpf, et al., 1997).

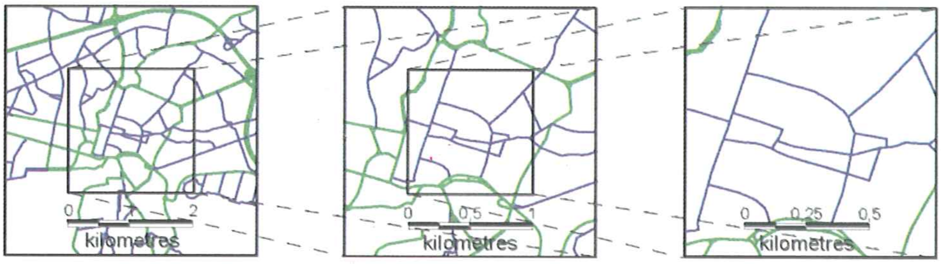

Figure 4. Graphical zoom: simple enlargement of the displayed objects when zooming in

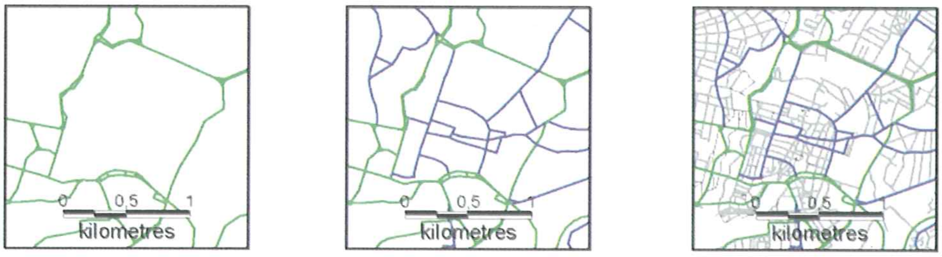

Figure 5. Content zoom: more content information in the same viewing window

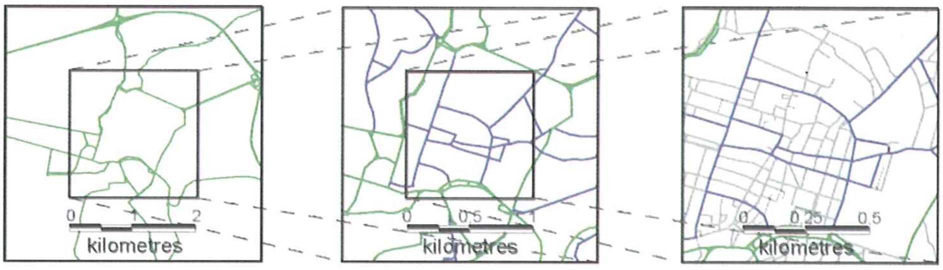

Figure 6. Intelligent zoom: adaptation of LoD data to the representation scale

E. Information density

We aim at finding a balance between the quantity of LoD data and the window size used to display it in order to provide the appropriate level of information to the user. The adequation between the LoD of data and the scale of representation is illustrated by *the principle of constant information density* known in cartography as Topfer's radix law (Franck, et al., 1994). This law posits that the number of objects per display unit should be constant as the user zooms:

$$n_f = n_i \times \sqrt{m_i / m_f}$$

where m_i is the origin scale,
m_f is the destination scale,
n_i is the number of objects at m_i,
n_f is the number of objects at m_f.

More generally, the amount of information should remain constant whatever the map scale is. For (Woodruff, et al., 1998), a "well-formed application" is one that conforms to that principle.

Furthermore, Tufte has proposed *information density metrics* to evaluate the efficiency of representation of quantitative information on a printed page. One of these metrics, the data-ink ratio, has been extended by (Blasio, et al., 2002) to the design of dynamic graphic displays. Data-ink ratio attempts to quantify the efficiency with which the "ink" has been used to present data to the reader by comparing "the amount of ink used for the non-redundant display of data to the total amount of ink in the graphic". The logical equivalent of ink for electronic documents would be active pixels (i.e. pixels which have a value).

Given a set of maps produced (manually or with a computer-assisted tool) at different scales, the *multi-scale tree* algorithm is proposed in (Franck, et al., 1994) to automatically produce different views of the data as the user zooms. These views would have a constant number of active pixels. The "*VIDA environment*" proposed in (Woodruff, et al., 1998) interactively guides users in the construction of applications with constant information density. This visualization environment supports two density metrics, number of objects and number of vertices, but have a general infrastructure for supporting multiple density metrics.

Measures used in *map generalization* (Peter, 2001; Galanda, 2003), for example "consecutive vertex distance", could also be used. The distance between consecutive vertices of a polyline or polygon outline should not be inferior to a minimum visual separability distance. Rather than using this constraint to apply a specific generalization operator, we could use it to determine valid scale intervals for LoD layers. Such graphical density measures can be adopted in a mobile spatial visualization system in order to display the LoD representation adapted for the selected scale.

III. THE SYSTEM DEVELOPED AT ULR-L3I

A. General architecture

The architecture of the system that we propose follows a server-side approach, even if a part of computations can be performed locally. It makes use of wide-spread and standard technologies: Global Positioning System (GPS) for real-time location of mobile user, Personal Digital Assistants (PDAs) or hand-held computers for visualization of spatial information by the client and cellular phone (with communication standards UMTS[1], GPRS[2] or GSM[3]) for communication between client and server. The system that we have developed is an embedded spatial information visualization system that can be divided in two main parts. The client manages data visualization, user requests and communication with the data server. The server manages the data and the access to data sources. A prototype has been implemented as a Java applet that can be executed by a Java-enabled web browser.

Our purpose is to minimize the amount of data exchanged between the client and the server (Stockus, 2002). We have to take into account the actual constraints of mobile devices in relation to desktop computers:

- constraints related to wireless communications: weakness and instability of data transfer,
- constraints of mobile devices: especially calculation power, storage capacity and display size,
- constraints related to the mobile context: real-time perception by user of the spatial objects seen on the map, more distractions.

The following assumptions are made for managing data. All data are centralized and can only be modified on the server. Data transfer is performed as an answer to a client request. As some data are also stored in the client's cache, the reuse of locally available data allows displaying requested information without connection to the server.

[1] UMTS (Universal Mobile Telecommunications System) is the third generation transmission norm in Europe (until 2Mbit/s).

[2] GPRS (General Packet Radio Service) is a high rate technology using the principle of packets transmission (until 144 Kbit/s).

[3] GSM (Global System for Mobile communications) is the second generation mobile phones norm currently used in Europe and some Asian countries.

B. Management of multi-resolution data

LoD of data and incremental strategy

The approach that we have chosen for managing multi-resolution data is a LoD one because it appears to be more interesting for us than the RTG approach (cf. section II.B). Each object is represented by multiple and interconnected representations (one for each resolution where it exists). We suppose that we have several layers of data at various LoDs: in our case, these layers come from only one source by generalization. We consider that all layers are served from a single topologically consistent source. Our solution is close to that of progressive transmission. Increments are computed from two resolutions of data. They allow real-time reconstruction, on the client-side, of object representations in generalization or refinement direction.

Multi-resolution data are managed using an increment based strategy. In a way comparable to (Bertolotto, et al., 2001), increments are considered as the "differences" between two LoDs of a vector dataset representing the same geographic area. This strategy allows to reuse locally available parts of objects during transition between different LoDs, i.e. either during a zoom out (decrease of detail) or during a zoom in (increase of detail). We call "incremental strategy" the "reconstruction" of a complete LoD object by reusing available ones in conjunction with increments.

Our approach presents two main advantages in relation to those of progressive transmission: increments can be used both in a refinement and generalization direction (and not only in a refinement one), and they are not computed during generalization process. Consequently, this solution allows more flexibility and can be applied to already existing data at different resolutions. Given a sequence of gradually simplified representations of the same area, our purpose is to determine scale intervals adapted for each LoD layer. We adopt an approach based on the density of visualized data.

IV. VISUALIZATION: EXPERIMENTATION AND SIMULATION RESULTS

A. Goal and approach

For the purpose of this paper, we consider such devices as PDAs with a typical display size of 4 to 5" and a display resolution from 160×160 to 240×240 pixels (Reichenbacher, 2002). The visualization of multi-resolution data must be adapted to small display constraints. We have chosen to use an adapted presentation (cf. section II.D), where maps at different scales are viewed in the full window and the user switches between them. An adapted presentation appears more interesting for visualizing data on a small screen than a key-map approach and it needs less calculation power than the "context+zoom" one. Furthermore, it seems more suited to our data management strategy because increments are only used during transitions between different levels of detail.

We can use information criteria to adapt data density to the characteristics of a mobile device screen (depending on its size, quality and number of pixels). This level of density should then be kept through scale changes by respecting an adequation between the scale and the LoD. If the criterion for LoD representations O^n displayed on the screen is greater (resp. lower) than a given threshlod value, representations O^{n+1} (resp. O^{n-1}) of the preserved objects should be displayed instead.

B. Dataset

Our dataset is composed of:

- Three LoD layers of a vector map representing the transportation network of the city of La Rochelle. They are only composed of polylines. This set of LoD layers has been produced by generalization in order to be adapted to an incremental management of data. Used generalization operators are selection based on importance of the streets, simplification using the Douglas-Peucker algorithm (Douglas, et al., 1973), and internal displacements for respecting the topological structure of the map.
- Eight different window sizes ranging from 300×300 to 2000×2000 meters and corresponding to different representation scales. The scale of representation is defined by the ratio between the window size used to display a surface and the real size of represented surface. According to a device's display size of 5 inches, the different viewing areas correspond to different scales from about 1/2500 to about 1/15000.
- Three GPS routes (i.e. three sets of coordinates collected at regular time steps with a car equipped with GPS) in La Rochelle city to simulate the real time navigation of a vehicle.

C. Methodology

Two conditions have to be verified in order to determine valid scale ranges: a "good" level of density for the user and respect of the principle of constant information density. The first one implies to take into account "knowledge and internal or cognitive representation of the structure, entities, and relations of space" (Mark, 1993), i.e. spatial cognition, of the mobile user. Two visual indicators have been computed for the three LoD datasets and different scales:

- the number of vertices visible on screen, i.e. the displayed vertices composing polylines which intersect the viewing area (Figure 7)
- the total and normalized length (in meters) of the parts of polylines which are extended into the window (Figure 8).

Displayed length of polylines can be normalized by eliminating scale factor. If l_r is the length of polylines displayed in a 1000× 1000 meters window and W the width of a given displayed area, so the normalized length l_n for this viewed area corresponds to: $l_n = (l_r \times W)/1000$. For instance, a total length of 5000 meters displayed in a 1000×1000 meters window is equivalent to a total length of 1500 meters presented in a 300m ×300m window.

These indicators, close to those used in (Woodruff, et al., 1998), seem revealing the visual density of data. We first chose levels of density for the considered indicators according to the screen characteristics (size and quality): here it is around 50 vertices and around 4000–5000 meters. These values have been chosen because they seem to visually correspond to a "reasonable" amount of data.

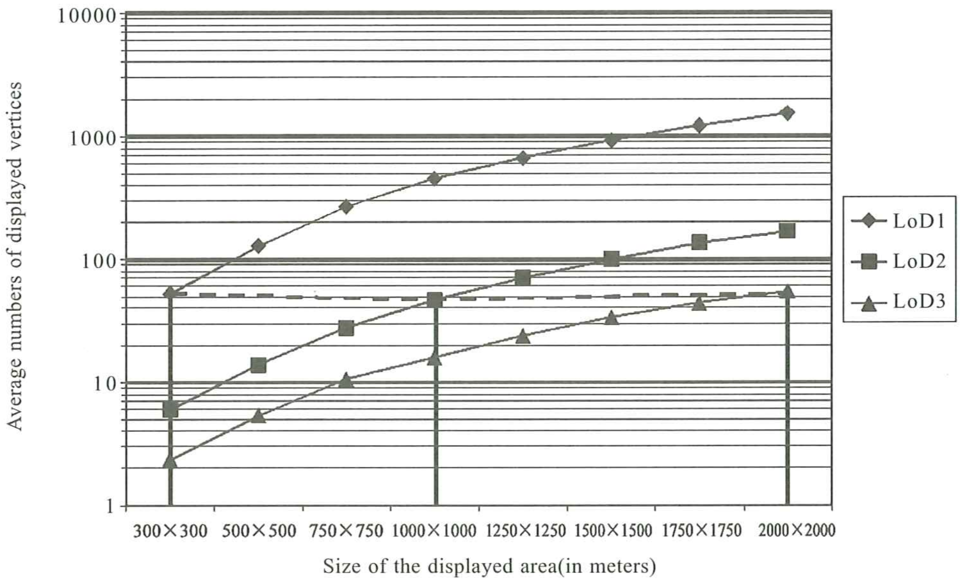

Figure 7. Average numbers of vertices displayed on a screen for different LoD data

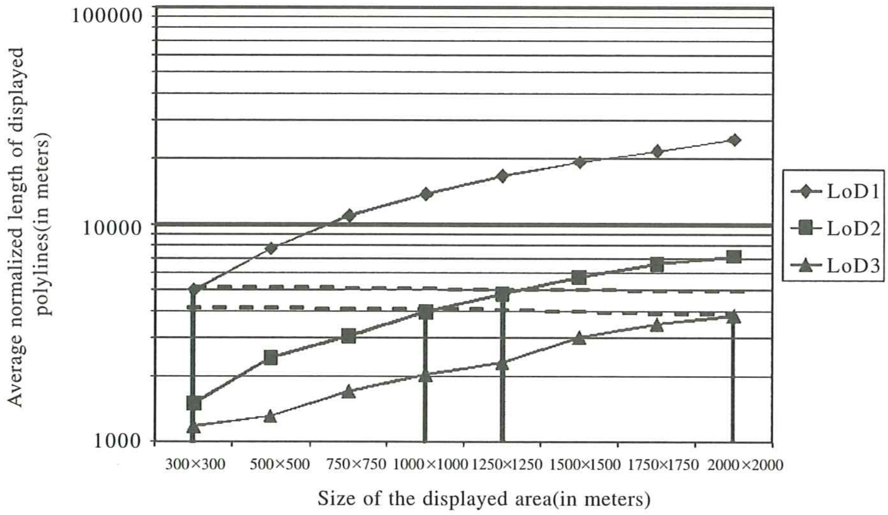

Figure 8. Average lengths of polylines displayed on a screen for different LoD data

Secondly, we draw the curves representing the variation of data density according to the scale: for each LoD of data, we averaged visual indicators for the three GPS routes and each window size (i.e. scale). For example, on average 47.17 vertices and a normalized polylines length of around 4000 meters are viewed on the screen for LoD 2 data in 1000×1000 window (i. e. for a scale of about 1/8000 on a PDA's screen). We can see that the data density logically increases when the scale decreases. Finally we have deduced the scales we had to choose for each LoD of data in order to get a constant information density whatever the scale is.

D. Results

With average numbers ranging from 44 to 53, a similar level of displayed vertices can be observed for each LoD layer for different displayed window sizes (cf. purple dotted line in Figure 7). With an average displayed normalized length varying from 3800 to 5000 meters, a comparable level of density can be found for about the same sizes of displayed areas (300×300, 1000×1000 and 2000×2000 meters). We can consider that appropriate scales are:

- around 1/2500 for LoD 1 layer,
- around 1/8000–1/10000 for LoD 2 layer,
- around 1/15000 for LoD 3 layer.

Appropriate scale intervals for each LoD layer can be centred on these values. An "intelligent zoom" respecting this adequation between LoD and scale is shown in Figure 9. It illustrates zoom out operation of a mobile user. Displayed informations appear to have a "good" level of density by avoiding overloading the map reader with too much data. We notice that problems can occur during a transition from a level i to $i+1$ when the mobile user is located on a street which disappears between these two LoDs. A "context+zoom" approach may be necessary in this case (but topological relations between different LoD objects should be maintained).

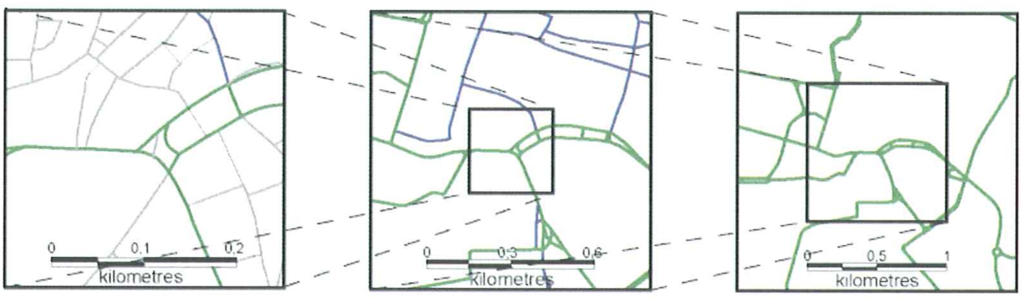

Figure 9. Intelligent zoom with constant information density

V. CONCLUSION AND PERSPECTIVES

In a mobile spatial information visualization system, data at different resolutions have to be provided to the user. In order to adapt these data to the requested scale, we have adopted a LoD based approach with an "intelligent zoom" respecting the principle of constant information density. By measuring density indicators, appropriate scales respecting the above principle have been chosen for each LoD layer. From the data management point of view, we propose an increment based solution and a suitable data transfer model in order to reduce the amount of data transferred between the client and the server.

For future work, spatial cognition of the user could be taken into account: some tests are needed in order to define what a "good" level of density means in an embedded visualization system. Furthermore, the limits of valid scale ranges must be determined more accurately. Other factors could also be considered, in addition to density. For example the colour, pointed out as "the most powerful styling element" in (Nissen, et al., 2003) and used to focus the user's attention in (Zipf, et al., 2002). For incremental management of data, mapping cases other than *1:1* ones should also be taken into consideration. The use of increments is not expected to be always interesting for these different mapping configurations.

ACKNOWLEDGMENT

This research was partially funded by the Communauté d'Agglomérations de La Rochelle, France.

REFERENCES

[1] AGENT (Map Generalization by Multi-Agent Technology). 1999. State of the art and selection of basic algorithms. Technical report, ESPRIT LTR 24 939. http://agent.ign.fr/.

[2] Agrawala M, Stolte C. 2001. Rendering effective route maps: Improving usability through generalization. Proceedings of SIGGRAPH01, Los Angeles (CA, USA), pp. 241–249.

[3] Ai T, Van Oosterom P. 2001. A map generalization model based on algebra mapping transformation. In Proceedings of the Ninth ACM International Symposium on Advances in geographic information systems, New York (NY, USA), pp. 21–27.

[4] Bédard Y, Bernier E, Devillers R. 2001. Automatic generalisation and multiple representations for spatial OLAP. Geo Information Fusion and Revision, Universitée Laval (Quebec, Canada), pp. 9–12. http://sirs.scg.ulaval.ca/yvanbedard/.

[5] Bertolotto M, Egenhofer M J. 2001. Progressive transmission of vector map data over the world wide web. *GeoInformatica*, 5(4): 345–373.

[6] Blasio A J, Bisantz M. 2002. A comparison of the effects of data-ink ratio on performance with dynamic displays in a monitoring task. *International Journal of Industrial Ergonomics*, 30 (2): 89–101.

[7] Brenner C, Sester M. 2003. Continuous generalization for small mobile displays. In International Conference on Next Generation Geospatial Information, Boston (USA), October.

[8] Cecconi A. 2003. Integration of cartographic generalization and multi-scale databases for enhanced web mapping. PhD thesis, University of Zurich.

[9] Cecconi A, Galanda M. 2002. Adaptive zooming in web cartography. Proceedings of SVG Open 2002, Zurich (Switzerland), pp. 78–83. http://www.svgopen.org/papers/2002/.

[10] Dettori G, Puppo E. 1996. How generalization interacts with the topological and metric structure of maps. Proceedings of the 7th International Symposium on Spatial Data Handling, Delft (The Netherlands), pp. 27–38.

[11] Douglas D H, Peucker T K.1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer*, 10 (2): 112–122.

[12] Follin J M, Bouju A, Bertrand F, Boursier P. 2003. Extension to multi-resolution of an embedded spatial information visualization system. Proceedings of 6th AGILE conference on Geographic Information Science, Lyon (France), pp. 149–159.

[13] Follin J M, Bouju A, Bertrand F, Boursier P. 2003. Management of multi-resolution data in a mobile spatial information visualization system. Proceedings of the 3rd International Workshop on Web and Wireless Geographical Information Systems (W2GIS 2003), Roma (Italy), pp. 84–91.

[14] Frank A U, Timpf S. 1994. Multiple representations for cartographic objects in a multi-scale tree - an intelligent graphical zoom. *Computers and Graphics Special Issue on Modelling and Visualization of Spatial Data in GIS*, 18 (6): 823–829.

[15] Frigioni D, Tarantino L. 2003. Multiple zooming in geographic maps. *Data & Knowledge Engineering*, 47(2): 207–236.

[16] Galanda M. 2003. Modelling constraints for polygon generalization. 5th ICA Workshop on Progress in Automated Map Generalization, Paris (France).

[17] Harrie L, Sarjakoski T, Letho L. 2002. A mapping function for variable-scale maps in small-display cartography. *Journal of geospatial engineering*, 4(2): 111–123.

[18] Hoppe H. 1996. Progressive meshes. ACM SIGGRAPH, editor, Proceedings of SIGGRAPH 96, Annual Conference Series, pp. 99–108, New Orleans (USA).

[19] Huang B, Jiang B, Lin H. 2001. An Integration of GIS, Virtual Reality and the Internet for Visualisation, Analysis and Exploration of Spatial Data. *International Journal of Geographical Information Science*, 15(5): 439–456.

[20] Jiang B, Claramunt C. 2004. A Structural Approach to Model Generalisation of an Urban Street Network. *Geòinformatica: an International Journal on Advances of Computer Science for Geographic Information Systems, Kluwer Publishers*, 8 (2): 157–171.

[21] Krum D M, Ribarsky W, Shaw C D, Hodges L F, Faust N. 2001. Situational visualization. Proceedings of the ACM symposium on Virtual reality software and technology, Baniff (Alberta, Canada), pp. 143–150.

[22] Lehto L, Kilpelinen T. 2001. Generalizing XML-encoded spatial data on the web. *Proceedings of 20th International Cartographic Conference*, 4: 2390–2396, Beijing (China).

[23] MacEarchen A M, Kraak M. 2001. Research challenges in geovisualization. *Cartography and Geoinformation Science Journal*, 28(1): 3–12.

[24] Mark D M. 1993. Human Factors in Geographical Information Systems, chapter Human spatial cognition, pp. 51–60. Belhaven Press. http://www.geog.buffalo.edu/ dmark/DMScottchapter.html.

[25] Nissen F, Hvas A, Mnster-Swendsen J, Brodersen L. 2003. Small-display cartography. Public report Deliverable D3.1.1, GiMoDig-project. http://gimodig.fgi.fi/deliverables.php.

[26] Persson J. 2004. Streaming of compressed multi-resolution geographic vector data. Proceedings of 12th International Conference on Geoinformatics–Geospatial Information Research: Bridging the Pacific and Atlantic, Gaevle (Suède), pp. 765–772.

[27] Peter, B. 2001. Measures for the generalization of polygonal maps with categorical data. Fourth ICA Workshop on Progress

in Automated Map Generalization, Beijing (China).

[28] Reichenbacher T. 2002. SVG for adaptive visualisations in mobile situations. Proceedings of SVG Open 2002, Zurich (Switzerland). http://www.svgopen.org/papers/2002/.

[29] Stockus A. 2002. Accés a des bases de données spatiales dans un environnement client-serveur : Application à un système de navigation embarquée. PhD thesis, University of La Rochelle, L3i research laboratory.

[30] Stockus A, Bouju A, Bertrand F, Boursier P. 2001. Accessing to spatial data in mobile environment. In Proceedings of 2nd International Conference on Web Information system, Kyoto (Japan), pp. 414–422.

[31] Timpf S, Devogele T. 1997. New tools for multiple representations. Ottoson, L. Editor, 18th International Cartographic Conference ICC'97, Stockholm (Sweden), pp. 1381–1386. ICA/ACI.

[32] Woodruff A, Landay J A, Stonebraker M. 1998. Constant information density in zoomable interfaces. Proceeding of Advanced Visual Interfaces '98, L'Aquila (Italy), pp. 57–65.

[33] Zipf A, Richter K. 2002. Using focus maps to ease map reading, Developing smart applications for mobile devices. Kunstliche Intelligenz (KI). Special Issue Spatial Cognition, pp. 35–37.