Toward Web Mapping with Vector Data

Julien Gaffuri

European Commission - Joint Research Centre
Via Enrico Fermi 2749, 21027 Ispra, Italy
julien.gaffuri@gmail.com

Abstract. Improving the use of vector data in web mapping is often shown as an important challenge. Such shift from raster to vector web maps would open web mapping and GIS to new innovations and new practices. The main obstacle is a performance issue: Vector web maps in nowadays web mapping environments are usually too slow and not usable. Existing techniques for vector web mapping cannot solve alone the performance issue. This article describes a unified framework where some of these techniques are integrated in order to build efficient vector web mapping clients and servers. This framework is composed of the following elements: Specific formats for vector data and symbology, vector tiling, spatial index services, and generalization for multi-scale data. A prototype based on this framework has been implemented and has shown satisfying results. Some principles for future standards to support the development of vector web mapping are given.

Keywords: Web mapping, standard, spatial data infrastructure, geoportal, vector data, vector tiling, generalization, spatial index.

1 Introduction

An always increasing part of the maps we use every day are digital maps published on the Internet. If the first web maps were simple static images, web maps have progressively been considered as special images displayed within specific viewers. In such viewers, specific cartographic tools are available to explore the geographical space by panning and zooming in and out. Data layers from different servers can also be selected to be overlayed. The Internet has deeply changed the way maps are nowadays designed and used [1].

However, it seems the limit of existing web mapping technologies has been reached. To open a next level of interactivity and improve the user experience of web maps, it may be necessary to change the approach web maps are made with. This next step could be to enable a direct interaction of the user with the map objects. This interaction is not possible nowadays because web maps are, for a huge majority of them, based on raster data. Like paper maps, these maps are just images of objects the user can only see and not touch and manipulate.

The solution to go further in web mapping interactivity is to fully open web mapping to vector data. Vector data are nowadays mainly used in web mapping to build static raster maps to be published on a server. Developing a new web

N. Xiao et al. (Eds.): GIScience 2012, LNCS 7478, pp. 87–101, 2012.

mapping architecture to enable the publication and on the fly display of vector data would be an important step toward a new generation of web mapping applications.

In this paper, some benefits and challenges of shifting from raster to vector web mapping are presented. A state of the art of existing techniques for vector web mapping is given. Then, a framework unifying some of these techniques to make vector web mapping feasible is proposed. This framework integrates the following elements: Vector tiling, spatial indexing, multi-scale data and generalization. Finally, requirements for future vector web mapping standards are given.

2 Benefits and Challenges of Vector Web-Mapping

Improving the use of vector data in web mapping is often shown as the next challenge of web mapping [2,3,4]. Such change would allow, for example, unlocking the development of the following applications:

- A user could easily retrieve thematic and semantic information for each map object, like in a traditional GIS software. This information could be displayed in a specific window or a tool-tip. The user may have access not only to the primary attributes of the object but also to a wider set of external data linked to this object. This feature is especially important for augmented reality applications [5].
- Using the object geometries, some simple geoprocesses could be performed on the client side, like for example, computing the length of a road or the area of a parcel. To go further, more complex geoprocesses may be run on more advanced clients, which may open the gate to the fusion of web mapping and web GIS.
- Because the objects are rendered on the fly on the client, vector web mapping would allow an improved map content personalization. This personalization could be done at the layer level (the user may define his own style for a full data layer) and also at the object level (the user could make an important object bigger and display it with a different style).
- Many advanced digital cartography methods such as graphic generalization [6], label placement [7], legend customization [8,9], etc. may be introduced in web mapping clients. These modules may ensure the data rendering follows some basic cartographic principles. They may be loaded dynamically depending on the user needs.
- A true integration of data coming from different servers would become possible. Instead of having "lazy mash-ups", where data layers of different servers are simply overlaid on top of each other, "smart mash-ups" could be developed. In such mash-ups, explicit relations between the objects may be computed and used for specific purposes. For example, a map showing pizzerias close to metro stations could be built from the integration and analysis of restaurant and public transport data layers.
- The interaction between data users and data providers may be improved. The users' feedback on the data could be more explicit: Instead of specifying

only the location of an error described in a free text field, the user may submit a full and more precise update of the data. He could easily modify, add and delete objects. He could also capture new object geometries and snap them on existing geometries. This feature could support the development of collaborative maps and VGI [10].
- By opening web mapping to vector data, it would not be necessary anymore to pre-process and cache raster tiles. This may shorten the publication of updates of existing data. This is especially important to encourage the mapping of "live" data, like sensor data or geoRSS data. Nowadays, spatial dynamics are usually shown on videos records - only rarely raw live data are accessible to be displayed.
- The on the fly rendering of vector data by the client makes possible the development of new innovative cartographic visualization techniques (see for example figure 1), especially dynamic visualizations with moving and changing objects. Depending on the specific context of the user, and the nature of the data he wants to display, suitable cartographic visualization techniques may be developed.
- and finally, vector web mapping may certainly open many other advanced and innovative applications we cannot imagine yet [11].

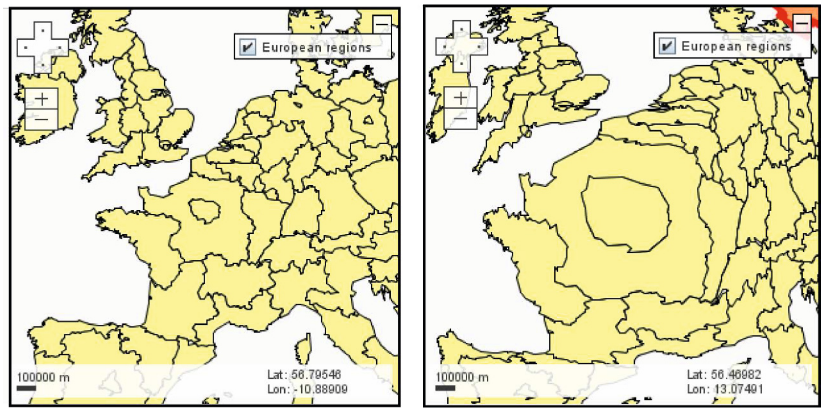

Fig. 1. Magnified view as described by [12]. On the right image, the map is magnified at the center of the view. A deformation of the vector data is computed on the fly when the user pans. See on-line demo: `http://www.opencarto.goldzoneweb.info/index.php?id=european-regions`.

The main obstacle to the development of vector web mapping is performance. Web maps must be fast maps, and existing web maps based on vector data usually do not meet the minimal requirements in terms of display speed. For this reason, the approach based on the publication of pre-computed raster maps has been preferred until now. However, taking into account that:

- client device memory, processing and connection capacities are always improving,
- and digital mapping methods of vector data, like generalization, are nowadays mature,

web mapping with vector data is becoming an acceptable approach. There are already emerging practices for web mapping systems based on vector data. If initiatives exist to make the shift to vector web mapping, it is not so common and nowadays, a huge majority of the map used on the Internet are raster maps. Rarely, vector data layers composed of usually few markers are displayed on top of raster maps.

One reason of this under-utilization of vector data is the lack of well-established, standardized and integrated approach to support efficient vector web mapping. Existing framework have been mainly developed for raster data and do not take into account the specific requirements of vector data. However, approaches exist to improve vector web mapping.

3 Approaches for Vector Web Mapping

The predominant approach to use vector data in web mapping is to extend existing raster clients to vector data. The client usually downloads vector data and displays it on top of raster images. The well-known limit of this approach is the long time usually necessary to transfer, decode and render the vector data. Furthermore, the final map is often not even legible because too dense for the map scale (see for example figure 2). As a result, the user waits a long time before an illegible map is displayed, and the application often becomes slow.

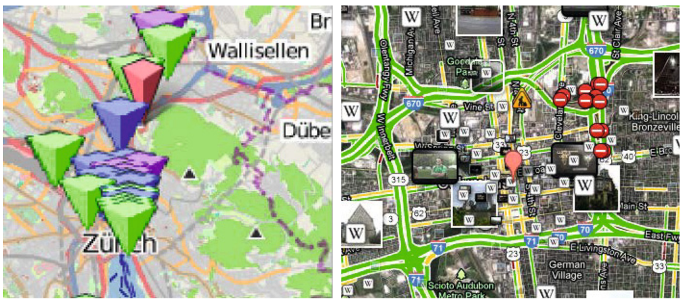

Fig. 2. Examples of existing web maps based on vector data

Approaches exist to improve the performance of web vector maps:

Use of specific data formats. The transfer duration is improved by the use of small and compressed formats for vector data and their symbology. There are many formats for vector data, most of them used in GIS softwares. A significant part of them is based on XML [13,14] like KML, SVG, GML and SLD. XML based formats are efficient for spatial data exchange, but usually too verbose for a fast transmission, as required for vector web mapping. Some formats have been developed specifically for this purpose, like the GeoJSON format. Some vector formats allow to describe the object properties either as a list of (key,value), following the GIS practice, either embedded within HTML code, like in KML. File compression also helps making the files smaller (like zip compression for KMZ files, and several JSON compressions for GeoJSON). Beside vector data formats, style formats allow to describe how vector data are rendered. In some vector formats like SVG and KML, the styles are encoded within the data file. Some other formats like geoCSS, GSS and SLD allow an independent encoding of the data and their associated styles.

Vector tiling. Existing vector web mapping applications often load a full file containing vector data the user will never see, because outside of its current view. Vector tiling [15,16,17] allows to ensure only the data within the user's view are requested and loaded by the client. The principle is to decompose the vector dataset into different parts, each of them corresponding to vector data contained within a tile (see figure 3). In the case vector objects belong to several tiles, these objects are cut into pieces and each piece is assigned to the corresponding tile. Only the tiles are published on the server (usually one file per tile) and the client requests, caches and renders only the suitable tiles depending on its view and zoom level. Useless data outside of the view are not retrieved, which allows a performance improvement. A drawback of this method is the necessity to reassemble the objects on the client side. Compared to raster tiling, vector tiling is relatively new in web mapping and not well established yet.

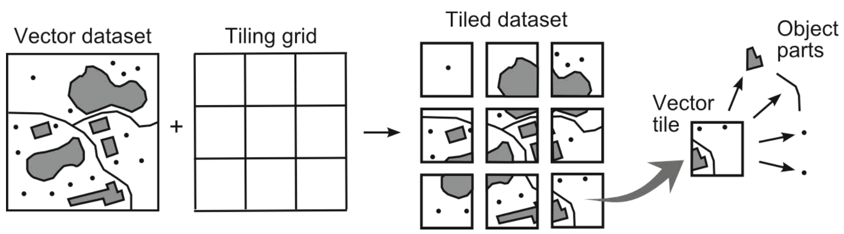

Fig. 3. Principle of vector tiling

Multi-scale data and generalization. The performance problem in vector web mapping is often due to the use of too detailed vector data. Indeed, such data are cumbersome to transfer, load and render, and may also not be legible as shown on figure 2. The solution is to provide to the client vector data with a level of detail suitable with the chosen zoom level. When the zoom level changes, new vector data with a suitable level of detail for this zoom level are requested, cached and rendered. For this purpose, a multi-scale vector database is required on the server. A multi-scale database provides different representations of a region with different levels of details. Such multi-scale database can be produced automatically by generalization. Generalization has been identified as one of the key elements to make vector web mapping possible [18,19]. Its automation is known as a challenging issue, and has been the topic of many research for years [20,21]. Generalization is nowadays well formalized and operational techniques are used to automate many data and map production processes. In web-mapping, mainly the Ramer-Douglass-Peucker filtering algorithm and some clustering algorithms are used. Richer generalization methods exist [20,21] but have, surprisingly, not been adopted in web mapping.

Progressive transmission. Progressive transmission and streaming methods exist for many kind of data, like images [22]. Specific methods have been developed for vector data [23,24,25,26,27,28,29]. The principle of these methods is to load progressively the points composing the object geometries, and display the loaded data continuously, before the full transmission is complete. As a result, the data are displayed starting with a simplified view progressively enriched with additional details. Progressive transmission do not contribute to solve the performance problem. It improves the user experience but is not the prior aspect to focus on to unlock the use of vector data in web mapping. A progressive loading of the data may also be obtained using asynchronous queries to the server for each vector object.

None of the previous approaches allows to solve alone the performance problem – an efficient vector web mapping demands to use several together. In the next section we propose a framework that integrates some of these approaches and may help to progress toward vector web mapping.

4 An Integrated Framework for Vector Web Mapping

4.1 The Relevant "Data Slice"

The performance issue can be solved by serving **only** the relevant data to the user's client. For this purpose, we propose to extract and serve the relevant "data slice" in the location-LoD space, as shown on figure 4. This relevant data slice depends on the selected position in the geographical space, and the selected zoom level [28]. Data outside of the view, and more detailed than what the zoom level requires are useless. Vector web mapping servers should send only these extracted data to the clients. This requirement illustrates that scale has to be considered as a full dimension of geographical information, like the three spatial dimensions and the temporal dimension [30]. For the selected zoom level, data with a relevant level of detail have to be provided. Furthermore, taking into account that:

– the viewer screen size (usually) do not change,
– and according to the equal information density law [31], the information density has to be constant whatever the zoom level,

all data slices should have comparable sizes, whatever the position and the zoom level. Consequently, depending on the client capacity (network bandwidth, memory, processing, screen size), a threshold data slice size should be defined, and the data slice provided by the server should not exceed this threshold size. The only way to ensure all data slices do not exceed this threshold size is to simplify the data by generalisation. The performance is controlled by the level of generalisation of the vector data: If the client faces performance issues, it means the vector data have not been simplified/generalised enough.

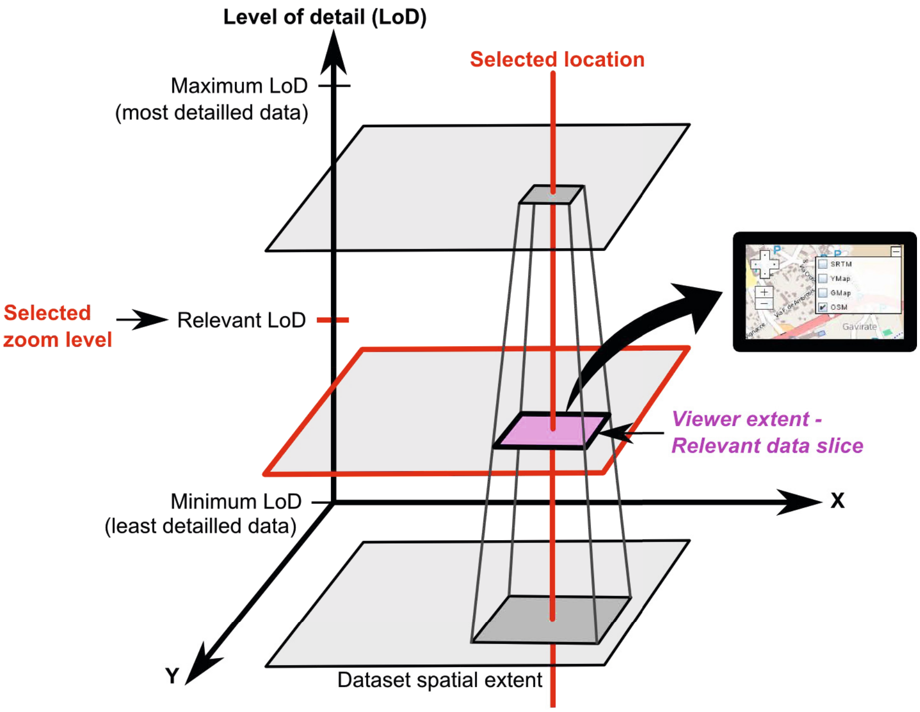

Fig. 4. The relevant information corresponds to the selected spatial extent, for the selected zoom level

In order to improve the performance of vector web mapping, it is necessary to extract the relevant data on both dimensions: Location and level of detail (LoD).

4.2 Location-Based Data Extraction

To extract the relevant data according to the view location, vector tiling and spatial indexing are used.

Vector Tiling. Vector tiling allows an efficient extraction of the relevant data according to the view location. Vector tiles are pre-computed on the server and identified according to their position in the tiling grid. Traditional tiling grids used for raster tiles may be applied also for vector tiles. The client needs the capability to retrieve vector tiles according to the view location, like in raster tiling. The following elements are also necessary:

– Vector objects are identified. The objects are built by merging their pieces belonging to different tiles and having a same identifier. The vector format used should include the possibility to identify the objects.
– The client is able to compute a union algorithm to build the object geometries from the union of their pieces. This union algorithm is however more simple than a generic union algorithm, taking into account that the geometries to union do not overlap and only touch each other along the grid lines. For linear geometries, this union is a simple concatenation of vertice lists. This union algorithm may be improved by including a code to each piece, that show from which side of the tile the original geometry has been cut. Further work may be undertaken to design such specific union algorithm for vector tiling.
– The client is able to cache vector data. Like for raster data, this caching improves the efficiency, even if it requires some memory capacities. For vector caching, two caches are required: For the tiles and for the vector objects.
– Object geometries and attributes are retrieved separately. Indeed, it is not necessary to retrieve the object attribute values for each object piece. A separation of both geometrical and semantic data enables to retrieve the object attribute values only once and improve the performance.

Spatial Indexing. Spatial indexing is a well-known technique in GIS to improve the location-based retrieval of vector objects. We propose to introduce spatial index services to improve the vector web mapping performance. Such service has the following characteristics:

– The spatial index structure is known by the client. We propose to use a quad-tree spatial index build on the same structure as the vector tiling grid.
– The spatial index service has the capability to provide:
  - References to the objects contained within a specified index cell,
  - an individual object from its reference.

The vector data retrieval is performed in two steps: First the client computes the relevant index cells depending on the view. If some of these cells have not been cached yet, the client sends a query to the spatial index service to retrieve the references to the objects the cells intersect. Then, the client retrieves the object it has not cached yet - because an object may be referenced in several index cells, it may be already have been retrieved. As for vector tiling, two caches are needed: For the index cells and for the vector objects.

Vector Tiling or Spatial Indexing? Vector tiling and spatial indexing are two different strategies to do the same thing: Retrieve vector objects efficiently according to their location. None of these strategies is better. In the first one, the objects have to be reassembled on the client side. In the second one, two steps are required, and the whole object geometry is retrieved even if only one small part is within the view. The most suitable strategy depends on the kind of vector data: **Vector tiling is suitable for large and non compact object layers** (like for example contour lines, routes, GPS traces, etc.), while **spatial indexing for layers composed of small and compact objects** (like point objects, small areas, etc.). In case a data layer is composed of heterogeneous objects, it may be possible to split it into two layers of large and small objects and use the relevant strategy for each sub-layer. In order to improve the architecture of the system (servers and clients), it is pertinent to use the same grid structure for both vector tiling and spatial indexing (a quad-tree). In that way, the same client cache structure for tiles and vector objects may be used for both strategies.

4.3 LoD-Based Data Extraction

Multi-scale data produced by generalization allow relevant data to be extracted according to the zoom level. It is necessary to synchronize the zoom level with the relevant level of detail (LoD) so that simplified enough data are transfered from the server to the client. Pre-computed multi-scale data should follow the equal information density law [31]: Whatever the visualization scale, the information density should be constant and remain below a threshold. This threshold is both a legibility and performance threshold: It ensures the map is simple enough to be legible, and, in a web context, it also ensures there is no performance issue according to the system capacities (bandwidth, memory, data processing and rendering) – Generalization should be used to ensure vector tiles size is low enough to be transfered and rendered by the client in a satisfying time.

Nowadays, only few simplistic generalization techniques are used in web mapping. In [6], we propose an architecture to improve the use of existing generalization techniques in web mapping. In this architecture, the generalization is shared between the server and the client:

– Multi-scale vector data are computed and stored on the server using model generalization. Model generalization (also called conceptual or semantic generalization) allows a level of detail reduction of the data. Object representing detailed concept are usually aggregated into objects representing more generalized concepts (See figure 5). The geometric level of detail is also reduced according to a target resolution of the data.

– Graphic generalization is performed on the fly and progressively on the client while loading and rendering the vector data. Graphic generalization transforms the map objects to ensure legibility constraints are satisfied. For example, too small objects are enlarged, and too close objects are deformed or displaced (See figure 6). Opening web mapping to vector data makes possible the development of clients with graphic generalization capabilities as described in [32].

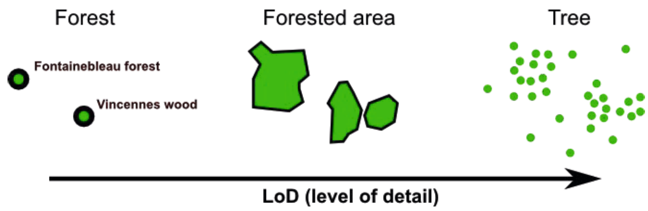

Fig. 5. Model generalization: Forests, forested areas, and trees. Three concepts representing the same reality for different semantic levels of details.

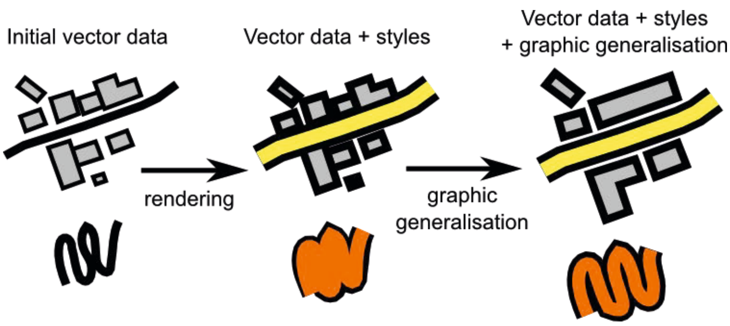

Fig. 6. Graphic generalization (Figure from [6])

5 Experiments

The presented framework has been implemented as part of the OpenCarto project [33]. This project aims at providing a software platform to expose advanced spatial data visualisation techniques on the web using vector data. It includes various modules for spatial data import, a component for multi-scale mapping composed of a generic multi-scale data model and generalisation algorithms, some components for vector tiling and spatial indexing, and a vector web mapping client as described in [32].

The prototype has been tested on two kinds of datasets (See figure 7): A dense dataset of small objects (world airports represented as points), and a dataset of large objects (relief contour lines). For both datasets, one generalised data layer has been produced for each zoom layer – the standard mercator zoom levels from

4 to 15 have been tested. The small objects dataset has been generalised using clustering, displacement and filtering algorithms (See figure 7 left); The large objects dataset has been generalised using selection based on contour interval and filtering algorithms (See figure 7 right). These datasets have then been transformed into a hierarchy of 256*256px GeoJSON vector tiles and published using the *http://myurl.org/z/x/y.json* standardised URLs pattern. No spatial indexing service has been tested yet.

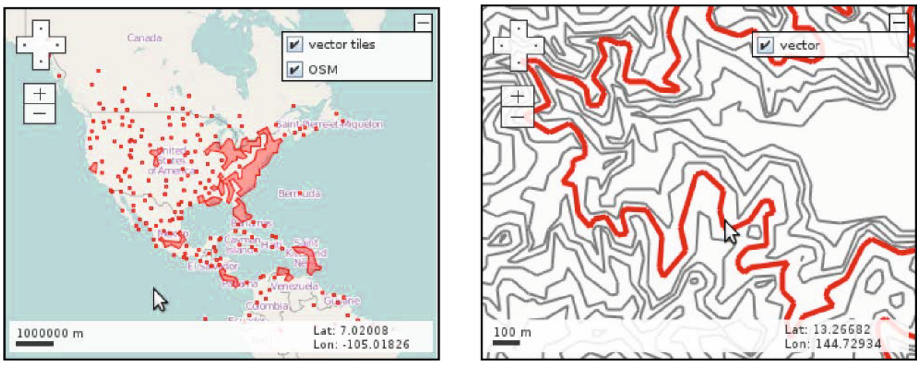

Fig. 7. Test case: Airport point data (left) and relief data as contour lines (right)

At the end of the tile preparation process, the size of the tile repository is 34MB for dataset 1 and 22MB for dataset 2. Without generalisation, these sizes are respectively 242MB and 1.16GB. Without generalisation, the tile size distribution is rather heterogeneous and some tiles have a size of 101MB for dataset 2. With generalisation, the tile size distribution is homogeneous and do not exceed 110KB – this maximum size can be controlled by the generalisation level. For a basic 'spatial exploration' from one point to another and from one zoom level to another, the performence can be measured by the data amount to transit from the server to the client: Because the screen size do not change, the number of vector tiles requested do not change, and because the tile size is low thanks to generalisation, the performence is significantly improved. The spatial exploration is smooth whatever the location and zoom level. No performence issue

Table 1. Comparison raster – vector

Raster	**Vector**
Resolution	Level of detail
Image pyramid	Multi-scale database
Resampling	Generalization
Raster tiling	Vector tiling / spatial indexing
Raster progressive transmission	Vector progressive transmission

is encountered anymore, mainly thanks to the integrated use of the techniques presented in section 4.

6 Discussion and Conclusion

In this article, the potential benefits and challenges of vector web mapping have been presented. A framework integrating some existing techniques (vector tiling, spatial indexing, multi scale data and generalization) has been proposed, implemented and tested.

A future challenge would be to improve the integration of vector and raster web mapping techniques. Table 1 proposes analogies between raster and vector techniques. Unified data structures and services may be designed to progressively erase the boundary between vector and raster approaches. In a same way that the feature and coverage views can be integrated in GIS [34], vector and raster approaches may be merged in web mapping. A phenomena that appears as objects at some scales and as coverages at other scales may be represented using either raster or vector web mapping services depending on the zoom level.

Furthermore, in order to support the development of vector web mapping and ensure a minimal interoperability between vector servers and clients, specific standards may be proposed. Open formats for vector data and associated style formats are required. The requirements for such format according to the proposed framework would be:

- To allow thin representations of geometry and attributes. The JSON grammar used in GeoJSON format is certainly a pertinent candidate. Standard file compressions may also be used.
- To allow a separation between geometrical and attribute data. This requirement may improve vector tiling performence.
- To allow a separation between object and style description. This separation would enable the reuse of on-line data with personalized styles and, in the same way, the reuse of on-line styles on other data. It would make possible the development of vector style servers, beside vector data servers.
- To allow the definition of dynamic styling behaviors. The way an object displays should not be static - it should depend on its context.
- To allow the definition of object behaviors according different interface events.

A second standardization field may be protocols for client/server communication. Most of the existing international standards (like the ISO and OGC standards WMS, WFS, GML and SLD) have been designed mainly for download services and do not take into account the specific requirements of vector web mapping. Specific services, such as the *Complex Vector Web Service Protocol*, may emerge. The spatial indexing service we have proposed may also be subject to standardization – it may be designed as an extension of WFS.

Furthermore, it may be useful to improve the way the LoD/scale dimension is handled in existing vector data formats. In the same way geographical objects have a spatial and temporal extent, they also have a “scale extent” as formalized in [30]. This scale extent is a scale range for which the spatial object exists. It would make easier the publication of vector objects on the Internet and their use by vector web mapping applications. The definition of scale extents for vector object exist in KML (with the “lod” element), in SLD (for layers), and also in SVG [35]. Structures and formats to represent multi-scale objects would also be required. For the same reason that there are coordinate reference systems for space, it may be needed to define *scale reference systems*. Such scale reference system would define which zoom levels are supported by the multi-scale vector database – it may be continuous (an object scale extent would be a scale interval) or discrete (an object scale extent would be a set of zoom levels).

Finally, taking into account the high diversity of geographical data available on the Internet, it is necessary to provide generic model and graphic generalization patterns to be adaptable to a wide set of geographical objects. Generic model generalization patterns such as heat maps, cluster hierarchies, multi-scale networks and multi-scale contour maps may be developed in the future.

References

1. Peterson, M.P. (ed.): International Perspectives on Maps and the Internet. Lecture Notes in Geoinformation and Cartography. Springer (2008)
2. Esri: Comparing Vector and Raster Mapping for Internet Applications, An ESRI White Paper. Technical report, ESRI (August 2006)
3. Antoniou, V., Morley, J.: Web Mapping and WebGIS: do we actually need to use SVG? In: SVG Open 2008: 6th International Conference on Scalable Vector Graphics (August 2008)
4. Perron, J.: The future of web mapping (November 2009), `http://www.nsimtech.com/the-future-of-web-mapping/`
5. Ghadirian, P., Bishop, I.D.: Integration of augmented reality and GIS: A new approach to realistic landscape visualisation. Landscape and Urban Planning 86(3-4), 226–232 (2008)
6. Gaffuri, J.: Improving Web Mapping with Generalization. Cartographica: The International Journal for Geographic Information and Geovisualization 46(2), 83–91 (2011)
7. Zhang, Q., Harrie, L.: Placing Text and Icon Labels Simultaneously: A Real-Time Method. Cartography and Geographic Information Science 33(1), 53–64 (2006)
8. Christophe, S.: Creative Colours Specification Based on Knowledge (COLorLEGend system). The Cartographic Journal, 138–145 (May 2011)
9. Bucher, B., Buard, E., Jolivet, L., Ruas, A.: The Need for Web Legend Services. In: Ware, J.M., Taylor, G.E. (eds.) W2GIS 2007. LNCS, vol. 4857, pp. 44–60. Springer, Heidelberg (2007)
10. Goodchild, M.F.: Citizens as sensors: the world of volunteered geography. GeoJournal 69(4), 211–221 (2007)
11. Craglia, M., de Bie, K., Jackson, D., Pesaresi, M., Remetey-Fülöpp, G., Wang, C., Annoni, A., Bian, L., Campbell, F., Ehlers, M., van Genderen, J., Goodchild, M., Guo, H., Lewis, A., Simpson, R., Skidmore, A., Woodgate, P.: Digital Earth 2020: towards the vision for the next decade. International Journal of Digital Earth 5(1), 4–21 (2011)

12. Harrie, L., Sarjakoski, T., Lehto, L.: A variable-scale map for small-display cartography. In: Joint International Symposium on GeoSpatial Theory, Processing and Applications, ISPRS/Commission IV, SDH 2002, Ottawa, Canada (July 2002)
13. Badard, T., Richard, D.: Using XML for the exchange of updating information between geographical information systems. Computers, Environment and Urban Systems 25(1), 17–31 (2001)
14. Spanaki, M., Antoniou, B., Tsoulos, L.: Web Mapping and XML Technologies: A Close Relationship. In: 7th AGILE Conference on Geographic Information Science (April 2004)
15. Antoniou, V., Morley, J., Haklay, M(M.): Tiled Vectors: A Method for Vector Transmission over the Web. In: Carswell, J.D., Fotheringham, A.S., McArdle, G. (eds.) W2GIS 2009. LNCS, vol. 5886, pp. 56–71. Springer, Heidelberg (2009)
16. Langfeld, D., Kunze, R., Vornberger, O.: SVG Web Mapping. Four-dimensional visualization of time- and geobased data. In: SVGOpen 2008 (2008)
17. On Building Pyramid of Geographic Vector Data for OpenStreetMap. In: The XII congress of the International Society for Photogrammetry and Remote Sensing (ISPRS) (August 2012)
18. Jones, C.B., Mark Ware, J.: Map generalization in the Web age. International Journal of Geographical Information Science 19(8-9), 859–870 (2005)
19. Cecconi, A., Galanda, M.: Adaptive Zooming in Web Cartography. Computer Graphics Forum 21(4), 787–799 (2002)
20. Mackaness, W.A., Ruas, A., Sarjakoski, L.T. (eds.): Generalisation of Geographic Information, Cartographic Modelling and Applications. International Cartographic Association Series. Elsevier B.V., Oxford (2007)
21. Ica: Website of the ICA commission on generalisation and multiple representation, `http://generalisation.icaci.org/`
22. Rauschenbach, U., Schumann, H.: Demand-driven image transmission with levels of detail and regions of interest. Computers and Graphics, 857–866 (December 1999)
23. Bertolotto, M., Egenhofer, M.J.: Progressive Transmission of Vector Map Data over the World Wide Web. Geoinformatica 5(4), 345–373 (2001)
24. Liu, H., Yang, W., Yu, S., Chen, Y.: Progressive Vector Data Transmission Based on Overall Effects. In: NCM 2008: Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management, pp. 604–608. IEEE Computer Society, Washington, DC (2008)
25. Yang, B., Weibel, R.: Editorial: Some thoughts on progressive transmission of spatial datasets in the web environment. Computers & Geosciences 35(11), 2175–2176 (2009)
26. Yang, B., Li, Q.: Efficient compression of vector data map based on a clustering model. Geo-Spatial Information Science 12(1), 13–17 (2009)
27. Ai, B., Ai, T., Tang, X.: Progressive transmission of vector map on the web. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 37(pt. B2) (2008)
28. Corcoran, P., Mooney, P., Bertolotto, M., Winstanley, A.: View- and Scale-Based Progressive Transmission of Vector Data. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2011, Part II. LNCS, vol. 6783, pp. 51–62. Springer, Heidelberg (2011)
29. Ramos, J.A., Esperanca, C., Clua, E.W.: A progressive vector map browser for the web. Journal of the Brazilian Computer Society 15, 35–48 (2009)

30. van Oosterom, P., Stoter, J.: 5D Data Modelling: Full Integration of 2D/3D Space, Time and Scale Dimensions. In: Fabrikant, S.I., Reichenbacher, T., van Kreveld, M., Schlieder, C. (eds.) GIScience 2010. LNCS, vol. 6292, pp. 310–324. Springer, Heidelberg (2010)
31. Töpfer, F., Pillewitzer, W.: The principes of selection. Cartographic Journal 3, 10–16 (1966)
32. Gaffuri, J.: Generalisation on the web: Towards 'scale-aware' web mapping clients. In: Harrie, L., Sandgren, U. (eds.) Workshop on Web Cartography. Lantmäteriet & Lund University (May 2011)
33. OpenCarto project, `http://www.opencarto.goldzoneweb.info/`
34. Cova, T.J., Goodchild, M.F.: Extending geographical representation to include fields of spatial objects. International Journal of Geographical Information Science, 509–532 (September 2002)
35. Campin, B.: Use of vector and raster tiles for middle-size Scalable Vector Graphics' mapping applications. In: SVGOpen 2005 (August 2005)