

Web-based Spatial Data Visualization and Analysis Using the New Web Standard

Shangming Qiu¹

¹Department of Computer,
Guangdong AIB Polytechnic College,
Guangzhou, 510507, China

Haosheng Huang²

²Research Group Cartography,
Vienna University of Technology,
Vienna, 1040, Austria

Abstract—Compared to desktop geographic information systems (GIS), current WebGIS are still rather poor in terms of interface and functionality. Much of this discrepancy is due to the limitations of current web standard (HTML4). HTML5 is being developed as the new web standard to replace HTML4. It provides plugin-free graphic supports and enhanced functionality, which make it very promising for the future development of WebGIS. This article introduces three case studies to investigate whether HTML5 can work effectively in developing WebGIS. This is necessary as HTML5 is not designed primarily for geospatial domains. Results of the case studies show that HTML5 can help to develop cross-browser and highly interactive WebGIS with true vector graphic capabilities and enhanced functionality (e.g., spatial analysis).

Keywords—HTML5; WebGIS; spatial data visualization; spatial analysis

I. INTRODUCTION

Recent years have witnessed rapid advances in Web-based geographic information systems (WebGIS), which aim at providing geospatial functionalities (e.g., web mapping and spatial analysis) to users through a web browser. Web technologies have a profound effect on how WebGIS are developed. While the first browsers often acted as a renderer of simple web pages, current web browsers behave like “a runtime environment capable of delivering rich interactive applications across many application domains”[1]. Despite these advances, current WebGIS are still rather poor in terms of interface and functionality, compared to their desktop counterparts [2, 3, 4]. For example, providing cross-platform and highly interactive WebGIS with enhanced functionality (e.g., spatial analysis) is very challenging. This is probably due to the poor functionality support in the current web standard HTML4.

HTML5 is being developed as the next web standard. It aims at improving HTML4 with plugin-free multimedia support, while keeping it consistently understood by computers. It also integrates many application programming interfaces (APIs) for complex web applications. However, since HTML5 is a rather new standard that started in 2009 and is still under development, it has not yet been widely applied in WebGIS.

Based on three case studies, this article aims to investigate the feasibility of using HTML5 to create WebGIS with true vector graphic capabilities, high interactivity and enhanced functionality.

II. THE NEW WEB STANDARD AND RELATED WORK

HTML5 is a response to the observation that current web documents are a mixture of features introduced by various specifications and web browsers, and contain many syntax errors, which can be attributed to the limitations of the current HTML4. HTML5 is being developed to replace HTML4. It introduces some new syntactic elements, such as video, canvas and inline SVG. These new features are designed to make it easy to include multimedia and graphical content on the web in a plugin-free manner. HTML5 also introduces many APIs for complex applications. Beyond that, it has been developed with the consideration of mobile devices. All these features make it a promising technology to create complex, cross-platform, and high-performance web applications.

Current research in WebGIS mainly focuses on spatial data visualization using the canvas element and the web socket of HTML5 [2, 5, 6]. Canvas-based visualization is a pixel-based approach and does not provide a good interactivity. The potential of graphic features of HTML5 in visualizing spatial vector data with high interactivity has not been addressed. Also, it has not yet been considered how HTML5 can be used to provide spatial analysis in WebGIS.

This article will study the feasibility of HTML5 in developing highly interactive WebGIS with true vector graphic capabilities and enhanced functionality. Three case studies will be introduced to address the above concerns. Each of them covers a key aspect of WebGIS.

III. CASE STUDY 1: ONLINE SPATIAL DATA VISUALIZATION

Spatial data can be classified into raster data and vector data. Raster data, such as satellite images, represent the world as a surface divided into a regular grid of cells [7]. In contrast, vector data model the world as a surface littered with recognizable spatial objects, which can be represented as points, lines or polygons [7]. In HTML4, different raster formats (such as JPEG and GIF) are natively supported, which makes it very easy to publish raster data on the Web. However, visualization of vector data is not natively supported in HTML4. In order to visualize vector data on browsers, plugins are often employed, such as Flash and SVG plugins, which leads to compatibility problems.

HTML5 introduces many graphic features, which have high potentials to address the above problems in current

WebGIS. In addition to raster images, it provides canvas and inline SVG to dynamically visualize graphics on browsers. Canvas is an element defined to draw graphics via JavaScript. It is a raster-based visualization. In contrast, SVG defines several elements to draw basic shapes, paths and texts in a vector manner. It also supports interaction with its graphic objects. HTML5 natively supports inline SVG, which makes it easy to visualize vector data on web browsers in a plugin-free manner.

Canvas and inline SVG have high potentials for online spatial data visualization. They both provide methods/elements to draw basic shapes, paths, texts and images, which can be used to visualize different spatial features in spatial data. The support of graphical effects, such as filling, gradient, and filtering, enables visualizing spatial data in a meaningful and aesthetic way. The following case study is designed to compare canvas-based and inline SVG-based spatial data visualization on the Web. The spatial data of Vienna (Austria) are employed, containing a boundary layer, and a land-use image (in GIF format). The boundary layer is encoded in Geography Markup Language (GML), and contains the boundary and other information (such as ID) of each district.

In the canvas-based approach, a canvas is defined in an HTML page, and JavaScript is used to draw each layer. FillText function is used to visualize each district's ID. The functions beginPath, moveTo, lineTo, and closePath are used to draw each district's boundary. As each boundary contains many points, lineTo has to be used for each point. Finally, drawImage function is employed for adding the image. Figure 1(left) shows the graphic interface of the canvas solution.

In the SVG-based solution, we write an XSLT (eXtensible Stylesheet Language Transformations) style sheet to define symbols (polygons and texts) to represent the boundary layer in the GML data. With this, the GML file is then transformed into SVG format. This is done with the XSLT processor, which is natively included in current browsers. The image is then embedded in the SVG file with the image element. All the SVG codes are embedded in an HTML page. Figure 1(right) shows the interface of the SVG solution. It also illustrates the interactivity (e.g., mouse clicks, zooming and panning) of the SVG solution.

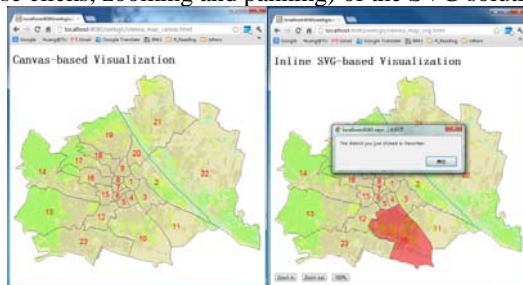


Figure 1. Canvas-based (left) and SVG-based (right) spatial data visualization

Several issues can be learned from this case study. Firstly, in terms of implementation, due to the lack of advanced functions for drawing paths, canvas-based

solution often needs more scripts for visualizing spatial data. In contrast, SVG-based solution can get more supports from other related technologies, which make its implementation rather simpler. Secondly, in terms of visualization quality, canvas provides a raster-based view, while SVG employs a vector-based one, which can be zoomed without quality degradation. Thirdly, in terms of interactivity, SVG-based solution supports users to interact with the spatial objects in the map view. This feature is restricted in canvas.

In summary, both canvas and inline SVG can be used to visualize spatial data on web browsers. However, for providing interactive and high-quality spatial data visualization, inline SVG is preferred to canvas.

IV. CASE STUDY 2: REAL-TIME WEB MAPPING

Currently, with the ubiquity of sensors, large sets of observations about the world are being generated in a nearly real-time manner. Therefore, more and more WebGIS need to provide real-time visualization of spatial data. Before HTML5, several techniques, such as polling and Comet (long polling), have been adopted for showing real-time information. These techniques are often query-driven, where the update request is initiated by browsers. They are often complex to configure, and lead to high network traffic. HTML5 provides web socket to address these problems. Web socket is event-driven, and can be used with JavaScript. With web socket, a bi-directional channel between the browser and server can be set up. An event happening on the server can be captured (such as new data are being stored in the database), and sent to the browser instantly.

In the following, we design a case study to investigate the potentials of web socket in real-time web mapping. Emomap (<http://openemotionmap.org>) is a research project which collects people's emotional ratings in their daily lives. This case study provides a real-time web map of all the ratings, and highlights the latest ratings. We use Apache Tomcat as the web server to implement web socket. A servlet which implements the required web socket functions such as onOpen, onTextMessage, and onClose is developed to deal with user requests, and monitor the update event of the rating database. On the browser side, JavaScript is employed to receive real-time updates, and display them on the web map. Figure 2 shows the screenshot of the real-time map.

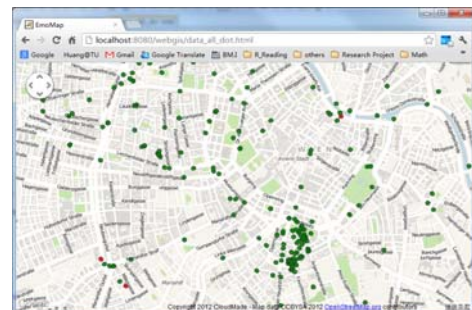


Figure 2. Screenshot of the real-time map. Red dots represent the latest ratings.

Results of this case study show that web socket can be employed for real-time web mapping. Due to its event-driven paradigm, WebGIS employing web socket is easy to implement, and can achieve a very low traffic load.

V. CASE STUDY 3: WEB-BASED SPATIAL ANALYSIS

Spatial analysis plays a key role in GIS [8]. However, current WebGIS mostly provide data visualization only. This is probably due to the poor functionality support in HTML4. HTML5 introduces many APIs for complex applications, such as web SQL database. These APIs have high potentials for providing spatial analysis in WebGIS. In the following, as a case study, we investigate how these APIs can be employed to provide load balancing spatial analysis in WebGIS.

In contrast to implementing spatial analysis solely on the server or browser sides, load balancing spatial analysis executes spatial analysis on either the server or the browser side, depending on the estimated execution costs (i.e., network transmission and computational costs) [4]. When users submit a spatial query (e.g., using a Spatial Extended SQL SESQL), the costs of server-side execution and browser-side execution will be estimated. If the estimated cost of server-side execution is less, this query will be executed on the server side. Otherwise, it will be executed on the browser side. A framework, as depicted in Figure 3, was proposed for load balancing spatial analysis [4]. In this architecture, SESQL compilers are used to carry out SESQL queries on GML or SVG. SVG2GML translator and GML2SVG translator are used for converting between GML and SVG. The cache structure on the browser side records which spatial data are available on the browser. Both decision-makers are employed to decide where to execute the current spatial query [4].

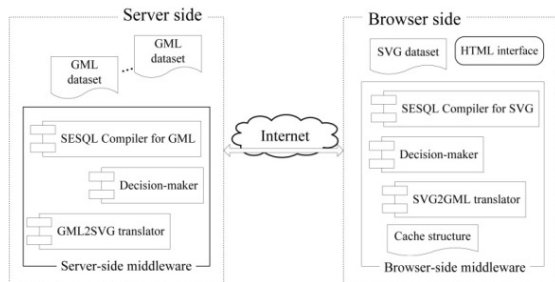


Figure 3. Architecture of the load balancing spatial analysis

For implementing the browser-side middleware, inline SVG is used to visualize spatial data. SESQL compiler is implemented using the SQLite compiler of the web SQL database (<http://www.w3.org/TR/webdatabase/>). Decision-maker and SVG2GML translator are implemented using JavaScript. Web storage (<http://www.w3.org/TR/webstorage>) is used to store the cache structure. The server-side middleware is implemented as Java Servlets. Web socket is used to facilitate the interaction between the server-side and browser side middleware. With this, a WebGIS with load balancing spatial analysis is implemented. Users can access spatial analysis simply with a web browser.

Figure 4 shows the interface of this case study. This screenshot also shows the results of a spatial analysis: “listing all the administrative districts affected by river ‘R1’, and calculating the size of their affected area”. It lists the names of affected districts, and their affected sizes in the list-box at the right-bottom corner. These districts are also highlighted in the map view.

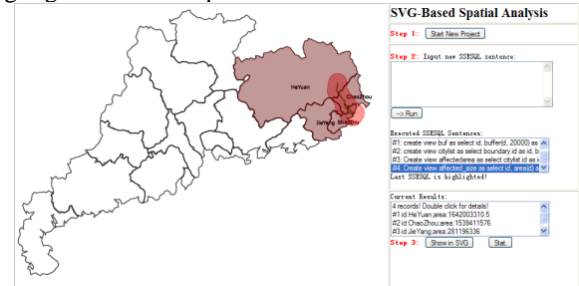


Figure 4. Screenshot of the WebGIS with load balancing spatial analysis.

Results of the case study show that based on HTML5, load balancing spatial analysis can be provided in WebGIS. HTML5 can also be used to implement server-based and browser-based spatial analysis, as both of them are two simple cases of load balancing spatial analysis. In summary, HTML5 is very promising in providing spatial analysis in WebGIS, which will enable users to access spatial analysis functions simply with a web browser.

VI. DISCUSSION AND CONCLUSIONS

In the above case studies, we have shown that HTML5 holds great promise in the future development of WebGIS. It can help to develop cross-browser and highly interactive WebGIS applications with true vector graphic capabilities and enhanced functionality (e.g., spatial analysis). In other words, in terms of interface and functionality, with the help of HTML5, the gap between WebGIS and their desktop counterparts can be narrowed.

However, there are some limitations and issues needing further studied, e.g., copyright issues, large dataset handling and aesthetics in web mapping. Beyond these, as more and more web maps are viewed from mobile devices, the aspects covered in the above studies also need to be investigated for mobile devices.

REFERENCES

- [1] Melamed, T. and Clayton, B. A Comparative Evaluation of HTML5 as a Pervasive Media Platform. In: T. Phan et al., eds. Mobile Computing, Applications, and Services. Heidelberg: Springer, pp. 307–325, 2010.
- [2] Corcoran, P., Mooney, P., Winstanley, P., and Bertolotto, M. Effective Vector Data Transmission and Visualization Using HTML5. In: Proceedings GISRUK 2011. UK, 2011.
- [3] Lin, H. and Huang, B. SQL/SDA: A Query Language for Supporting Spatial Data Analysis and Its Web-Based Implementation. IEEE Transactions on Knowledge and Data Engineering, 13 (4), pp. 671–682, 2001.
- [4] Huang, H., Li, Y., and Gartner, G. A load balancing method to support spatial analysis in XML/GML/SVG-based WebGIS. In: S. Li, S. Dragicevic, and B. Veenendaal, eds. Advances in Web-based GIS,

- Mapping Services and Applications. London: CRC Press, pp. 153–168, 2011.
- [5] Boulos, M., Warren, J., Gong, J., and Yue, P. Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping. *International Journal of Health Geographics*, 9 (1), 14, 2010.
- [6] Shi, B., Chen, N., and Shen, L. Research on web vector representation of thematic map using HTML5. In: FIG Working Week 2012, Italy, 2012.
- [7] ESRI. GIS Dictionary, <http://support.esri.com/en/knowledgebase/Gisdictionary> [Accessed 05 March 2014].
- [8] Goodchild, M.F. and Gopal, S. Modeling error in objects and fields. *The Accuracy Of Spatial Databases*, 1 (4), pp.107–114, 1989.