Streamlined Workflow for Large-Scale Interactive Geographic Visual Analytics

Marc Kramis
University of Konstanz
Department of Computer and Information Science
Box V 519, 78457 Konstanz, Germany
marc.kramis@uni-konstanz.de

Cedric Gabathuler
University of Zurich
Department of Geography
Geographic Information Visualization & Analysis
Winterthurerstr. 190, 8057 Zurich, Switzerland
cedric@geo.uzh.ch

ABSTRACT

Cartography and computer science powerfully attract each other to band together into a melting pot stimulating research in interdisciplinary fields such as Geographic Visual Analytics. Cartography with its rich history of practice and science brings along visualisation knowledge and large data sets. Computer science with its short but vigourous evolution brings along data structures, algorithms and hardware to make the visualisations and large data sets come alive for unprecedented user interaction. However, it turns out that the traditional workflow suffers from noticeable delays and read-only limitations which practically hinder the every-day convenient and fluent interaction with large data sets. We show how to streamline the traditional workflow by eliminating the intermediate data conversion step through switching to a native XML database. We suggest to use a RESTful interface providing scalable temporal read-write access. Finally, we provide preliminary measurements based on our prototype implementation named TreeTank providing both compressed storage and fast SVG delivery.

1. INTRODUCTION

Due to a lot of neurons for visual processing, humans are good at visually identifying patterns. In Geographic Visual Analytics, this is used for knowledge discovery in databases (KDD) or exploratory data analysis (EDA). A challenge of current research in this highly interdisciplinary field is to synthesise information and to derive insights from handling massive, dynamic, ambiguous, and often heterogeneous data sources [5].

The scientific objective is to understand how both individuals and teams carry out analytical reasoning and decision making tasks with complex information and to use this understanding to develop and assess information and communication technologies for this purpose [1]. There are different ways to analyse complex information and a number of different activities in science where such applications are useful. Gahegan [8] presented different approaches and types of methods to handle multivariate data. Thanks to continuous development and use of on-demand geovisualisation tools, it should be possible in the future to propose highly adaptable representations to the current needs of users in an inexpensive way [9, 8].

To provide windows into the complexity of phenomena and processes within complex and linked datasets, MacEachren and Kraak [2] focused research challenges in geovisualisation on four themes: representation, visualisation-computation integration, interfaces, and cognitive or usability issues. Modern cartography deals with complex processes of geospatial information organisation, access, display, and use. Collaboration and interactivity from both the cognitive and the usability perspective are well known research areas [8].

1.1 Problem Statement

Large-scale interactive Geographic Visual Analytics currently faces two major problems related to computer science:

1. *End-to-End Delay:* The traditional three-step workflow imposes a noticeable end-to-end delay between the query issuance and the retrieval of the final data ready for visualisation:
   (a) A SQL query is issued to a spatial database;
   (b) The database returns Standard Open Format;
   (c) The Standard Open Format is converted into SVG.
2. *Read-Only:* The traditional workflow does not consider the aspect of collaboration as the user can not persistently enhance the existing data on-the-fly with individual attributes such as comments or even pictures.

1.2 Contribution

Our contribution is fivefold:

1. *Two-Step Workflow:* We shortcut the traditional workflow by directly storing SVG data in a native XML database. This results in the following concise two steps eliminating the intermediate data conversion step as described with the problem statement:
   (a) An XQuery is issued to a native XML database;
   (b) The database returns SVG.
2. *RESTful Interface:* We introduce a RESTful web interface to Geographic Visual Analytics to cleanly separate client and server in a standardised and scalable way.
3. *Temporal REST:* We extend Geographic Visual Analytics by an inherent temporal dimension which allows to query the current as well as all past revisions of the stored SVG and its related data through the alluded easy-to-use RESTful interface.

4. *Read-Write:* We allow to interactively enhance the already stored SVG with statistical data and new attributes through XQuery Update.

5. *Implementation:* We provide a prototype implementation named TreeTank to estimate the impact of eliminating the traditional intermediate data conversion step.

1.3 Background

Representations such as cartographic maps create links between representation and user interface and map user cognition and geospatial data. A variety of problem solving and data exploration tasks are addressed using cartographic representations. The ongoing technological development changes the representation forms, the spatial data handling, the related information science, the technology communities, and the potential of these representation forms for productive use. Effectiveness also is a theme discussed when using such representations and it is linked with the behaviour of the user interacting with the display. The widespread availability of cartographic maps throughout the Internet leads to increased expectations on how to represent these maps [6].

Established specifications such as the eXtensible Markup Language (XML) as a technique for coding and structuring data should prove beneficial for portraying and interacting with geospatial information and visualisation [6]. Many different approaches (server-side, client-side, hybrid) are available to improve the performance [13]. Each approach has its distinguished impact on data manipulation, map management, user interactivity, and the distribution of server-side or client-side tasks [13]. In addition, scalability and the option for collaboration also vary with each approach.

Scalable Vector Graphics (SVG), an open vector-oriented XML grammar, is suitable to visualise data. Dunfey [10] proposed to use SVG to develop an open architecture for a vector GIS. SVG is a powerful tool and has the potential to visualise data now and in the future. SVG can be used to view vector graphics in a browser. There are plug-ins such as the Adobe SVG Viewer [4] and an increasing number of browsers directly supporting SVG [10]. Batik [11], a Java SVG toolkit, allows to develop applications which use SVG for visualisation. Neumann and Winter [3, 10] proposed to use SVG because it is an ideal vector format for web-based mapping. SVG, however, should be compliant with the OpenGIS Recommendation on the Definition of Coordinate Reference System for a XML grammar [12]. SVG suffers from its lack to store additional attributes. Still, SVG is a desirable tool and the use of a separate XML file for storing additional information is preferred [10].

With Geographic Visual Analytics, data is stored in a spatial database and can be exported as XML in a traditional geographical information system (GIS) file format or as SQL data. Widely used proprietary databases such as ESRI ArcSDE or Oracle Spatial store geospatial information in a binary long data type in an unpublished binary encoding. As such, the SVG document can only be extracted with the help of an SQL query. The traditional approach is to deliver the requested data in a Standard Open Format, e.g., a ESRI Generate File. An intermediate data conversion step is required to generate the SVG used for a flexible and easy-to-use interface. The traditional workflow is depicted in Figure 1 [10].

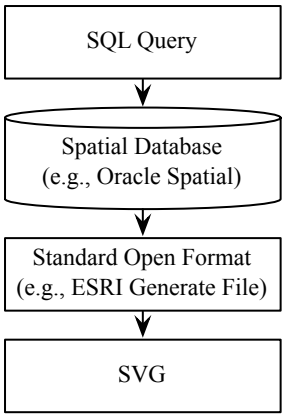

Figure 1: Traditional three-step workflow converting the native output of a spatial database into SVG.

2. FIVE STEPS FORWARD

2.1 Streamlined Two-Step Workflow

We suggest a two-step workflow as depicted in Figure 2:

1. An XQuery is issued to a native XML database;

2. The database returns SVG.

In stark contrast to the traditional three-step workflow, the intermediate data conversion step is eliminated, i.e., there is no need for converting the Standard Open Format such as an ESRI Generate File into SVG. The eliminated intermediate data conversion step both makes heavy use of CPU and IO and mainly contributes to the large end-to-end delay virtually inhibiting interactive Geographic Visual Analytics.

At the hearth of our main contribution lies the switch to a native XML database capable of directly storing and emitting fine-grained XML data. Unlike traditional relational databases, native XML databases do not store the XML data as character large objects and inherently know about the XML structure and XML nodes. The finer granularity allows to answer complex queries and extract the stored XML in a scalable fashion because the parsing and reconstruction process required with character large objects can be omitted. In addition, most state-of-the-art native XML databases support modifications of the stored XML.

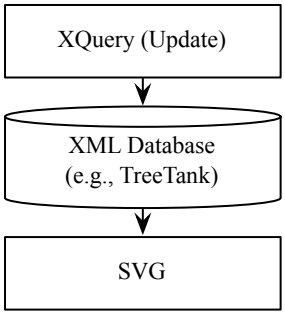

Figure 2: Streamlined two-step workflow directly outputting SVG stored in a native XML database.

Notably, the native XML database must not necessarily store a single large SVG document but may separate the SVG, i.e., cartographic data, from other XML, e.g., statistical or user-provided data. The single XQuery expression issued to the native XML database is responsible to tell the system how to enrich the SVG with additional data, e.g., how to colour different regions of the map due to the population density. The native XML database then executes the query by swiftly searching and combining all required XML fragments to finally return a single SVG. This lays the ground for distributing the underlying native XML database internally while still maintaining a single interface to the upper layers. Consequently, the elimination of the intermediate data conversion step by switching to a native XML database such as TreeTank leads to fundamentally better scalability than traditionally available.

2.2 RESTful Geographic Visual Analytics

Representational state transfer (REST) [7] is a set of network architecture principles which outline how resources are defined and addressed. Practically speaking, REST defines a simple and scalable interface to exchange resources over HTTP. Each resource must be uniquely addressable through hypermedia links meeting a universal syntax. A well-defined and typically small set of HTTP operations specifies how to proceed with the obtained resource. The basic operations are POST to create a resource, GET to read a resource, PUT to update a resource, and DELETE to remove a resource.

The time-tested scalability and unquestioned expressiveness of REST makes it the interface of choice when it comes to handle large-scale SVG data. The clean separation of client and server at the web layer (HTTP) allows both sides to be independently implemented while drawing from state-of-the-art standardised web technologies such as Java, Ruby on Rails, or Adobe Flex. In addition, REST is a bidirectional interface both for querying and modifying the requested resource.

As the specification of the RESTful interface lies in the hands of each service provider, he can both specify the set of available operations and resources. Additionally, the implementation – be it the traditional three-step or the streamlined two-step workflow – can be exchanged without modifying the RESTful interface and breaking existing clients.

A RESTful HTTP request looks like:

```
GET http://{host}/{path-to-resource}?{query}    (1)
```

where {host} is the name of the web server hosting the Geographic Visual Analytics service, {path-to-resource} is the name of the resource to retrieve, and {query} is a query expression used for a detailed specification of the result to return. E.g., a simple request might be:

```
http://localhost/MapOfAmerica?PopulationDensity
                                                (2)
```

to retrieve the SVG representing the map of the United States of America coloured with the population density. Note that the query must not necessarily be an XQuery expression but it may be an implementation-independent expression which is mapped to XQuery (or to a SQL query) on the server side.

2.3 Temporal Geographic Visual Analytics

Figure 3 depicts the evolution of a cartographic map over time. Currently, the user usually retrieves the revision of the map as it was last stored. A simple temporal extension to the RESTful interface empowers the user to retrieve the map as it looked like at any past point in time. This enormously facilitates interactive Geographic Visual Analytics in a temporal fashion. E.g., the evolution of the population density of a whole region can now easily be visualised.

A temporal RESTful HTTP request looks like:

```
.../{path-to-resource}/({point-in-time})?{query}
                                                (3)
```

where ({point-in-time}) denotes a point in time either specified as a revision number or an ISO-compliant date. The parenthesis (and) mark the temporal expression which can easily be extended in the future, e.g., to support time periods in addition to single points in time.

While the temporal dimension comes along with an unimposing but convenient extension to the well-known RESTful interface, it potentially imposes a huge storage and processing overhead if the underlying system does not natively support temporal queries. Our native XML database TreeTank was designed to support temporal queries out-of-the-box and is able to retrieve any past revision at the same cost as the last revision.

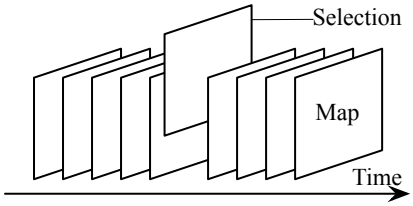

Figure 3: Temporal selection of a cartographic map as it looked liked at a past point in time.

2.4 Interactive Enhancement

Google Maps is an impressive example how users can not only visualise geographic data but also enrich it with new information. Unfortunately, the traditional three-step workflow does not support a feedback channel enabling user-driven enhancements or additional attributes. The complexity lies in the question where to efficiently store the newly added attributes, how to query them, and how to merge them with the existing geographic information.

Our streamlined two-step workflow leaves a single point to store the user enhancements for later retrieval: the native XML database. The user issues an XQuery Update expression which tells the native XML database to insert, change or delete an attribute. The modification immediately becomes visible for reading queries and therefore optimally supports collaborative projects. Taking the RESTful interface into account, the user may also issue a generic query by calling a POST, PUT, or DELETE operation which is transformed into a XQuery Update expression on the server.

To pick up our example about the population densities of a given region, we can now give a complete example involving all mentioned contributions. One can imagine a RESTful service running at the local host. Initially, the service does not contain any data at all. Then, the user successively performs the following actions through the RESTful web interface of the service:

- The SVG-based map is uploaded to the server;

- Attributes are inserted into the SVG to denote several regions by their identifiers;
- The population density data for the geographic region of the SVG-map is uploaded. Each upload contains the data of a specific year;
- More SVG layers, e.g., rivers or streets, are uploaded to the server;
- The user tags several regions on the map with his own data by storing more attributes for each region;

2.5 TreeTank Implementation

TreeTank is a native XML database designed to provide scalable read and write access to XML data. TreeTank concurrently allows multiple read and a single write transaction which creates a new revision with each commit. Furthermore, TreeTank was designed to be secure, easy to maintain, and a well-behaved partner for REST.

Preliminary measurements on a state-of-the-art desktop computer show two significant advantages of TreeTank. First, it compresses the original XML data while storing it in its native data structure. Second, it allows to quickly retrieve the original XML. Table 1 shows the promising preliminary results of both the compression and time measurements for three maps of different sizes. The excellent compression ratio is due to the verbosity of SVG. The time of the data conversion step alone (excluding the time to retrieve the original data from the spatial database) takes much longer than the time required to retrieve the whole SVG from TreeTank.

	SVG Size [MB]	**TreeTank Size** [MB]	**Conversion Step Time** [s]	**TreeTank Time** [s]
Map 1	0.02	0.01	0.38	0.22
Map 2	2.60	0.21	5.52	0.79
Map 3	138.70	11.3	102.38	3.79

Table 1: Preliminary measurements with TreeTank.

3. CONCLUSION

Both cartography and computer science mutually foster their respective research. Interdisciplinary fields such as Geographic Visual Analytics extensively draw from the melting pot of both a comparably experienced and a youthful science. While one needs ever faster tools to interactively visualise large data sets, the other is in need for a catchy use-case.

We identified the data conversion step of the traditional workflow of Geographic Visual Analytics as a major bottleneck imposing large end-to-end delays between the query issuance and the retrieval of the final SVG ready for visualisation. We suggest a new streamlined two-step workflow based on a native XML database promising better scalability and diminished delays while better supporting the interactivity aspect from the end-user perspective. We propose to use a RESTful interface providing scalable temporal read-write access to geographic as well as statistical or individual data. The clean separation of client and server at the HTTP web layer assures back-wards compatibility and better extensibility. We give an example on how to effectively use the RESTful interface and provide preliminary measurements based on our prototype implementation TreeTank. The excellent compression ratio as well as the fast SVG delivery strongly encourage further research.

Future work will include a detailed elaboration and implementation of our ideas to prepare a solid foundation for the research community of Geographic Visual Analytics. While the RESTful interface might look the same for quite some time, it will independently foster the progress on both the client and the server side. Under the hood, we will see distributed native XML databases capable of storing and querying ever larger sets of geographic and statistical data while supporting a multitude of concurrent collaborating users. Five steps towards large-scale interactive Geographic Visual Analytics – a leap forward from the computer science perspective.

4. REFERENCES

[1] A. MacEachren; G. Cai; M. McNeese; R. Sharma; S. Fuhrmann. GeoCollaboration Crisis Management: Designing Technologies to Meet Real-World Needs. In *Proceedings of the 2006 international conference on Digital government research*, volume 151 of *ACM International Conference Proceeding Series*. ACM Press, 2006.

[2] A. MacEachren; M. J. Kraak. Research Challenges in Geovisualization. *Cartography and Geographic Information Science*, 28(1), 2001.

[3] A. Neumann; A. M. Winter. Vector-based Web Cartography: Enabler SVG. `http://www.carto.net/papers/svg`.

[4] Adobe. SVG Zone. `http://www.adobe.com/svg/`.

[5] D. A. Keim; F. Mansmann; J. Schneidewind; H. Ziegler. Challenges in Visual Data Analysis. *Information Visualization (IV 2006)*, 2006.

[6] D. Fairbairn; G. Andrienko; N. Andrienko; G. Buziek; J. Dykes. Representation and its Relationship with Cartographic Visualization: a Research Agenda. *Cartography and Geographic Information Science*, 28(1), 2001.

[7] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[8] M. Gahegan. *Handbook of Geographic Information Science*, chapter Multivariate Geovisualization. Blackwell Publishers, 2007.

[9] M. J. Kraak. The Cartographic Visualization Process: From Presentation to Exploration. *Cartographic Journal*, 35, 1998.

[10] R. I. Dunfey; B. M. Gittings; J. K. Batcheller. Towards an Open Architecture for Vector GIS. *Computers and Geosciences*, 32, 2006.

[11] The Apache XML Graphics Project. Batik SVG Toolkit. `http://xmlgraphics.apache.org/batik/`.

[12] W3C. Coordinate Systems, Transformations and Units. `http://www.w3.org/TR/SVG/coords.html`.

[13] Y. S. Chang; H. D. Park. XML Web Service-based Development Model for Internet GIS Applications. *International Journal of Geographical Information Science*, 20(4), 2006.