Animating Maps: Visual Analytics meets Geoweb 2.0

Piyush Yadav[1], Shailesh Deshpande[1], Raja Sengupta[2]

[1]Tata Research Development and Design Centre, Pune (India)
Email: {piyush.yadav1, shailesh.deshpande}@tcs.com

[2]Department of Geography, McGill University, Montreal (Canada)
Email: raja.sengupta@mcgill.ca

Abstract

Sound visualization techniques for spatio-temporal data have potential to reveal interesting insights from GIS data. Visual analytics – hence - has captured attention of GIS researchers in recent past. Further, developments in free tools such as Google Map API, Open Streep Maps provide easy access to geospatial data that can be leveraged by Visual Analytics. In this paper we propose such a system which utilizes free GIS APIs to visualize spatio-temporal data effectively. Our application allows user to create time slide bar control to connect time and position of various GIS objects on the map and further displays them in animated mode. The control can handle vector and raster data with equal ease. Because of the effective visualization it is very easy to understand some complex spatio-temporal patterns as exemplified in the paper.

Keywords: Visual Analytics, Google Map API, Spatio –Temporal data, Geospatial data.

1. Introduction

Spatio-temporal information plays a vital role in understanding many complex geographic processes. Visualization techniques often neglect temporal properties of the data and thus undermine its importance (Andrienko *et al*., 2003). Existing visualization techniques need to be augmented further to generate meaningful insights from complex spatio-temporal datasets of large size. Dynamic visualization of the information is one of such key agenda for Geovisualization research (MacEachren and Jan-Kraak, 2001). The field of Visual Analytics (Andrienko *et al*., 2007, 2010) proposes visualization of spatially-explicit events that are temporally interconnected. Such events, when synchronized and displayed on a map can potentially reveal interesting facts. Further, the arrival of the Geoweb 2.0 attracts large amount of citizen participation in the process of map development (Haklay *et al*., 2008). Citizen’s novice to geo-spatial technology such as GIS can easily access and share the spatial information with Google Maps, Open Street Maps *etc.* Combining Visual Analytics with a Geoweb 2.0 interface provides a sophisticated mechanism to understand spatio-temporal data for variety of geographic processes.

This preliminary research extends the notion of Visual Analytics by developing a fully customizable, bar-type time legend (Harrower and Fabrikant, 2008) for various map API’s like Google Maps, Open Street Maps *etc*. This gives users the ability to represent spatio-temporal information by associating various GIS objects on map, and then synchronizing them with time legend to visualize changing patterns. For example, changing migration patterns of animals (*e.g.*, migratory birds) is a complex spatio-temporal phenomenon. Analysing this data using just tables and/or graphs would not provide complete insights. Integration of this time space data on a map, and further its visualization would help in better understanding. Such visualization shows a potential to generate new insights quickly, which would otherwise be very difficult to generate from other data formats. Added benefit is visually stimulating experience where user can view various events through a slider by controlling the relevancy of information. Since map API's itself contains a lot of pre-existing information, by combining this information with other geospatial database, a new form of visual analytics can be achieved.

2. System Design

The application described in this paper provides various tools for drawing, and animating information on Google and Open Street Maps using various API's of these maps. The application is enabled to handle both vector and raster data. It facilitates the user to create user specific maps either by providing the input to the application or by connecting it to any spatio-temporal database. These two application modes are explained below:

2.1 Mode of operation

Automatic Mode: In this mode, user can import the data by connecting to external spatio-temporal database (Fig. 1). Initially, the user provides the credential of database to connect with it. The server handles the further process by executing various queries to fetch the spatio-temporal information from the database. This mode handles both vector and raster data. The vector objects like line, circle, polyline *etc*. can be drawn using map API's. So, they are directly stored in the vector table of the application database. On the other hand raster data is stored in application database in two ways depending on the nature of external spatio-temporal database. If database table consists of series of raster images then they are directly stored in the raster table of application database. If the database consists of spatial information in form of pixel values then a python process is applied to convert these pixels to images and then stored in the application database. All the above data is stored with its temporal information.

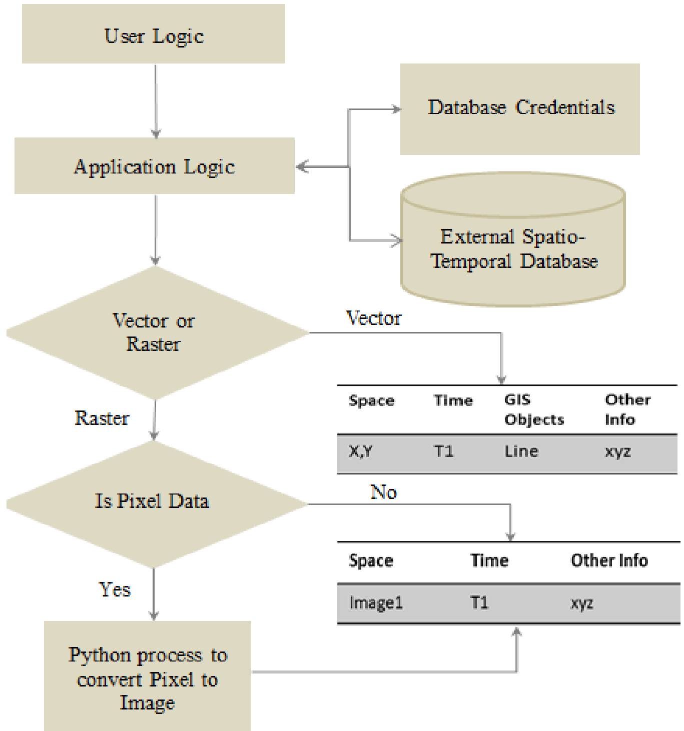

Figure 1. Data Fetching Process from External Data Source

Fig.2 explains the process of visualisation of the stored data. For vector data, the system would create map/GIS objects such as point, line, polyline for visualization automatically using map API's. User can specify additional graphical properties such as size, shape, colour for the attributes as and when required. The application will automatically create controls and integrate the data with map and slider, creating a dynamic slider-map visualisation. Similarly, slider controls are associated with each raster images defining its timestamp. Thus with each hover on slider a query is executed and raster image gets changed. The time lag of changing images have been kept minimal, thus giving notion of variations happening in same image.

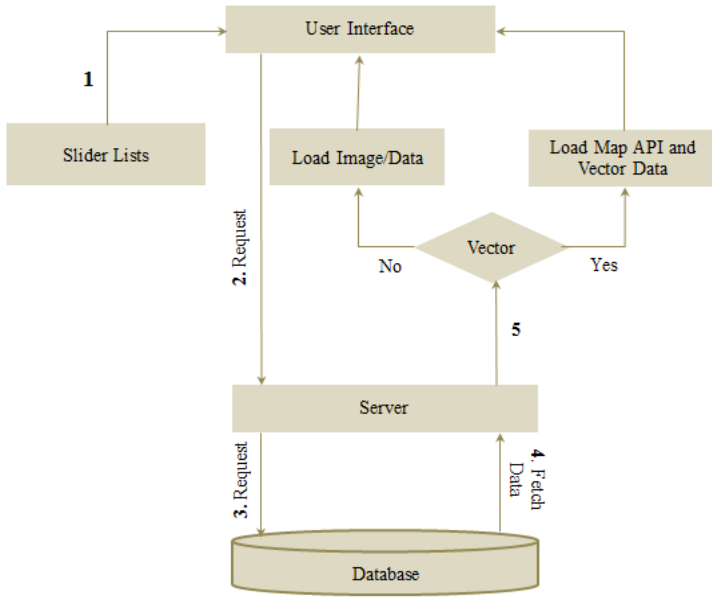

Figure 2. Data Retrieval Process from Application Database for Slider

User Customization Mode: In this mode, user can add a geographic feature (line, polygon, circle, marker, polyline *etc.*) on the map by clicking on it, and associating temporal information (timeline) with it. Further, different graphical properties can be assigned to the vector layer according to the importance of events, and geographical features: user can assign different colours or can add marker to the features as per the design. Thus, the digitized map can be saved with slider and can be accessed for future reference.

2.2 Architecture

The application is broadly divided into four parts (Fig. 3):

User Interface: HTML and JavaScript acts as a front end to the system. The interface provides various controls like import data, addition and deletion of slider, saving of map with slider *etc.* The user can create a customize slider-map by using these controls. The maps can be saved with slider and can be accessed by clicking on the saved link provided in the interface. The database connectivity for external spatio-temporal database is done by providing various credentials which is then pushed to server.

Server: The apache webserver is used for running the application. The server handles the JSON data: store and retrieve it from database on Request Response call. As shown in fig. 1 on request server (application logic) also connects to the external spatio-temporal database, fetch the data accordingly and store it into the application database. Various python processes have been written to handle the raster data. These processes convert the pixel information into images and associate the temporal information with it as fetched from the external spatio-temporal database.

Map API: The application presently uses Google Maps API v3 and Open Street Maps to create events on map. Various features of map API (marker, polyline, circle, marker *etc.*) have been used to create overlay on maps.

Database: SQL database stores the data of the customized maps. The customized data is in the tag-based format just like xml. This tag based data is converted to JSON while interacting with the map API. The database handles the vector and raster data differently. The vector table handles various GIS objects like line , circle *etc.* associating its temporal and other information in different columns. Similarly raster table handles the raster images with its relevant temporal information.

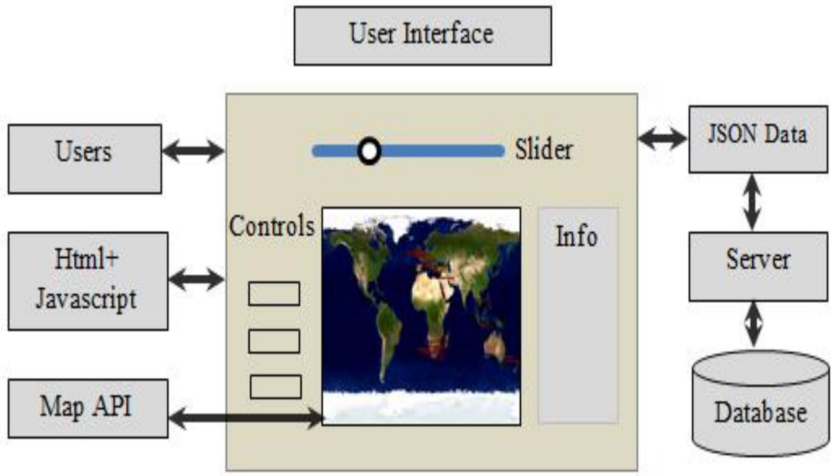

Figure 3. Architecture

3. Results

We demonstrate the discussed application with following working example: how Vasco Da Gama reached India from Portugal. In this example, each segment of the voyage is recorded as a separate element with a timestamp, which is then displayed on the bar-type legend (Fig. 4). This can be then used for visual analytics by picking a particular date (for example, Jan 1) and visualizing the information in as a spatio-temporal sequence of this voyage. In the future, one can imagine adding other objects (such as storms) that could have affected the voyage itself.

Figure 4. Visualizing the voyage of Vasco Da Gama

Fig 5. Visualizes how Ebola virus has spread to various African countries. In the first map has been shown that initially Ebola started spreading from three coastal African countries *i.e.* Guinea, Liberia and Sierra Leone but with the extent of time it has also spread to Nigeria, Mali *etc*. Thus, these type of analysis can be easily done that how disease has spread spatially with respect to time.

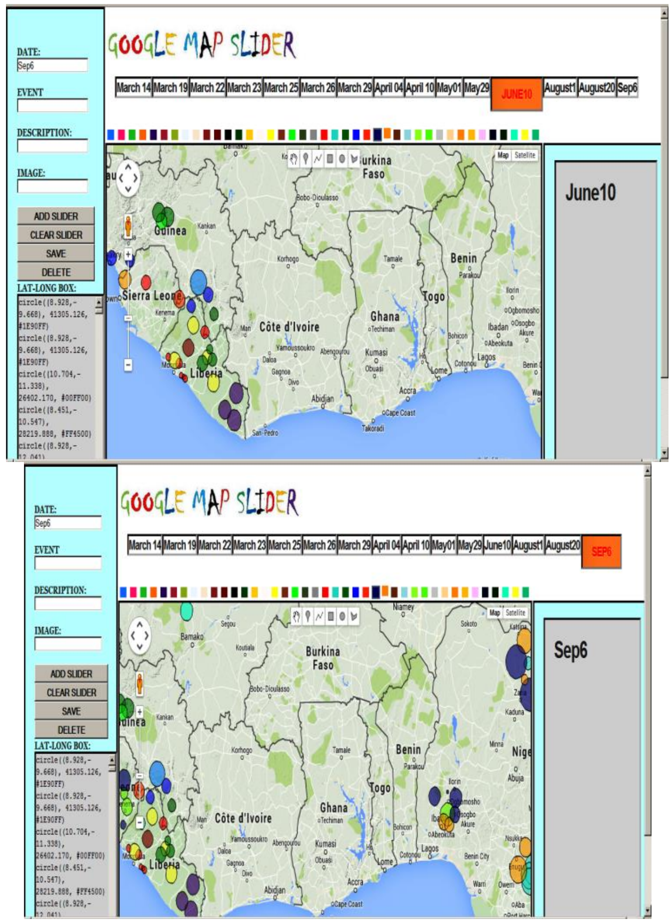

Figure 5. Visualizing the spread of Ebola virus in Africa

4. Conclusion

In this paper we presented a spatio-temporal bar type slider for the maps. The application gives full-fledged control to users to develop their own slider map view with various customization options. The users can save the map and can use it for future reference. The application has tried to solve the classic spatio-temporal visualization problem by integrating both the data at same time giving broader view of information. Thus, it provides for a more visually stimulating experience where user can view graphical representation of various events through a navigational slider, controlling the relevancy of information. Further, different graphical properties can be assigned to the digitized structure that allows for their importance can be reflected on the screen.

5. References

Andrienko, G et al., 2010. Space, Time and Visual Analytics. *International Journal of Geographic Information Science* 24(10): 1577-1600

Haklay, M et al., 2008. Web Mapping 2.0: *The NeoGeography of the GeoWeb. Geography Compass* 3:2011-2039.

Harrower, M & Fabrikant S, 2008. '*The Role of Map Animation for Geographic Visualization*' in Geographic Visualization: Concepts, Tools and Applications (Dodge et al., Ed):Wiley, UK 49-65.

MacEachren, A & Jan-Kraak M, 2001. Research Challenges in Geovisualization. *Cartography and Geographic Information Science* 28(1): 3-12.

Demšar, Urška, and Kirsi Virrantaus. "Space–time density of trajectories: exploring spatio-temporal patterns in movement data." *International Journal of Geographical Information Science* 24, no. 10 (2010): 1527-1542.

Andrienko, Gennady, et al. "Geovisual analytics for spatial decision support: Setting the research agenda." *International Journal of Geographical Information Science* 21.8 (2007): 839-857.

Andrienko, Natalia, Gennady Andrienko, and Peter Gatalsky. "Exploratory spatio-temporal visualization: an analytical review." *Journal of Visual Languages & Computing* 14.6 (2003): 503-541.

Google Map javascript API , https://developers.google.com/maps/web/