BUILDING AN INTEGRATED WEB-BASED ENVIRONMENT FOR EXPLORATORY SPATIOTEMPORAL DATA ANALYSIS

Bing She [a, *], Xinyan Zhu [a], Weidong Xiao [b]

[a] Liesmars, Wuhan University, 430079, China - coolnanjizhou@163.com, geozxy@263.net
[b] Science and technology on information system engineering laboratory, National University of Defense Technology, Changsha 410073, China - wilsonshaw@vip.sina.com

Commission IV, WG IV/5

KEY WORDS: GIS, Web based, Distributed, Services, Integration, Analysis

ABSTRACT:

Before conducting domain-specific modelling, one task faced by researchers is trying to effectively explore the data. There has been much research done to put spatiotemporal data analysis functions online. However, there is still a gap between the spatial data and the analysis procedures. Manually matching the data sources and the geoprocessing services would be a troublesome job for normal users. Furthermore, geoprocessing services published online are still limited and mostly target only static spatial data. This paper investigates an integrated web-based environment, which incorporates several open source software packages including PySAL, R, JTS, and customized implementations for exploratory functions such as the gravity model and the center-of-gravity model. Users can navigate spatial data and perform spatial analysis directly on them, without knowing the complexities of data transformation and integration. Both data sources and the analytic functions can be dynamically added. The environment released users from the great burden of intermediate data processing unrelated to their work.

1. INTRODUCTION

Location and time are the innate characters that exist in almost all datasets collected. Scientists from both natural science and social science are trying to explain the spatiotemporal phenomenon, using GIS tools and the spatial analysis functions embedded to help model the data. For example, spatial analysis methods are applied to identify the distribution of social problems like alcohol outlets (Ellawaya, Macdonalda et al., 2010). In hazard and risk assessment, researchers use GIS to create risk maps of different categories (Castelli and Scavia, 2008). Before conducting these domain-specific modelling, users have to effectively explore the data to identify possible relationships and patterns. The description and exploration of the dataset are commonly known as exploratory data analysis (EDA).

EDA is a collection of techniques for summarizing properties, detecting patterns and identifying outliers in data. When related to spatial or spatiotemporal dataset, it is termed as exploratory spatial data analysis (ESDA) and exploratory spatiotemporal data analysis (ESTDA). A wide range of ESDA tools have been implemented in many free GIS packages, including GeoDa, GeoVista studio; some of the functionalities are also implemented in commercial software such as ArcGis. The ESTDA functions have recently been made available in a number of software packages, like Geoviz, the STARS open source project, etc. (de Smith, Goodchild et al., 2011). However, these software tools are mostly available in desktop version.

Providing spatiotemporal data analysis functions over the web has many advantages over desktop software. It can easily adopt the server-computing technologies like cloud computing, and thus are capable of dealing with large spatiotemporal datasets. Besides, the application is more accessible to ordinary users over the web. There has been much research done to put spatiotemporal data analysis functions online. For example, Java applets are used to put exploratory spatial data analysis on the web (Anselin, Kim et al., 2004). Spatial weight creation is available online through web services (Rey, Anselin et al., 2009). In recent years, the OGC WPS specification and its related applications subjects has become an area of active study (Yue, Gong et al., 2010), many open-source WPS framework solutions are available nowadays, such as Python Web Processing Service (PyWPS, 2012), ZOO (ZOO, 2012), WPSint (WPSint 2012), etc. These solutions give great capacity for general geoprocesing tasks as well as customizing services for specific analysis.

However, there is still a gap between the spatial data and the analysis procedures and services. Although there have now existed convenient approaches for the retrieval of geospatial resources, the geoprocessing services are often separately published. Most of current web-based analysis tools or services require users to acquire and integrate the data first and then upload the data for analysis, but matching the data sources and the geoprocessing services would be a troublesome and time-consuming job for normal users. It would be great that users can access and analysis spatiotemporal data online as if they were using desktop software, without any intermediate steps for data processing.

This paper investigates an integrated web-based environment, where users can navigate the data and perform spatial analysis directly on them at the same time. Both data sources and the analytic functions can be dynamically added. The environment is incorporated into a web-based socio-economic application

* Corresponding author.

named China Geo-Explorer (She, Zhu et al., 2010). The environment released users from the great burden of intermediate data processing unrelated to their work.

The remainder of this paper is organized as follows. Section 2 discusses various spatiotemporal data analysis methods and the technologies used. Section 3 investigates the architecture of the integrated environment, and then gives a typical algorithm flow of the analysis procedure. Section 4 covers the experimental results of the environment incorporated into China Geo-Explorer. Concluding remarks are presented in Section 5.

2. EXPLORATORY SPATIOTEMPORAL DATA ANALYSIS

2.1 EDA Methods

EDA provides various descriptive statistics such as the mean and median. There have been many plots developed for visual display, including histogram, box plot, scatter plot and multi-plot. We employed StatGL (Data Numerica Institute, 2011), a highly interactive graphical library specifically designed for statistical visualization, to draw these plots. Linked to the map display, these plots provide users intuitive feelings of the data.

2.2 ESDA methods

Spatial autocorrelation is the driving force of ESDA analysis. Substantial amount of work have been done both in theoretical side and interactive visual methods, one of the most popular spatial statistics is the LISA statistics (Anselin, 1995). The interaction ways for geovisulazation in ESDA includes scaling, rotation, querying, brushing, browsing and effective navigational tools (Koua and Kraak, 2004). Gravity model and center-of-gravity model are commonly used in social science, and can also be categorized into the ESDA toolset. Gravity model is often used in trade analysis, while center-of-gravity model provides an intuitive way for indentifying the geographical center for selected economic variable.

For ESDA analysis, several open source software tools are linked into our environment. The first one is PySAL, an open source cross-platform library of spatial analysis functions written in Python (Rey and Anselin, 2010). PySAL contains many useful basic functions for exploratory spatial analysis, and the libarary is keep expanding. PySAL can be easily incorporate into application development. We use PySAL to compute the Moran's I statistics and spatial weight matrices. Because PySAL is still under developments, we have also implemented a set of other statistics in C++ as supplements, including the Global Geary's C statistics, and the Local Geary's C and Local G statistic. The C++ implementations are based on existing open source solutions in R. R is a software environment for statistical computing and also graphical display. We adopt C++ mainly for validating our integrated environment. The R version implementations can be linked to our environment as well.

The second library is JTS Topology Suite, which conforms to the OpenGIS "Simple Features Specification for SQL" and contains a rich collection of functions for spatial operations. Built on the functions of JTS, we implemented the gravity model and the center-of-gravity models.

2.3 ESTDA methods

Any method that tries to describe the spatiotemporal structure or provides dynamic interaction visualization can be broadly recognized as an ESTDA method. Present work includes scan statistics (Kulldorff and Nagarwalla, 1995), spatiotemporal autocorrelation (Hardisty and Klippel, 2010), etc., and there now exists a number of ways to effectively explore the spatiotemporal data.

In this paper, we implement the ESTDA analysis module as a direct extension of the ESDA module, for example, for the gravity model in ESDA, there is a spatiotemporal version gravity model correspondingly. The server side computes statistics for each time point and wrapped the result into a HashMap data structure, with the key denoting the time point and the value denoting statistics calculated for that time point. The client side is responsible to dynamically display the result through animation.

2.4 Spatial Regression Methods

Besides exploratory analysis, some experiments are also taken in this study to incorporate spatial modelling methods. We choose spatial analysis function in R to achieve the study. There are many packages now incorporated into R for spatial analysis. In this study, we use the spdep package to fit two basic models in spatial regression, the spatial lag model and spatial error model.

3. INTEGRATED WEB-BASED ENVIRONMENT

3.1 Server Side Design

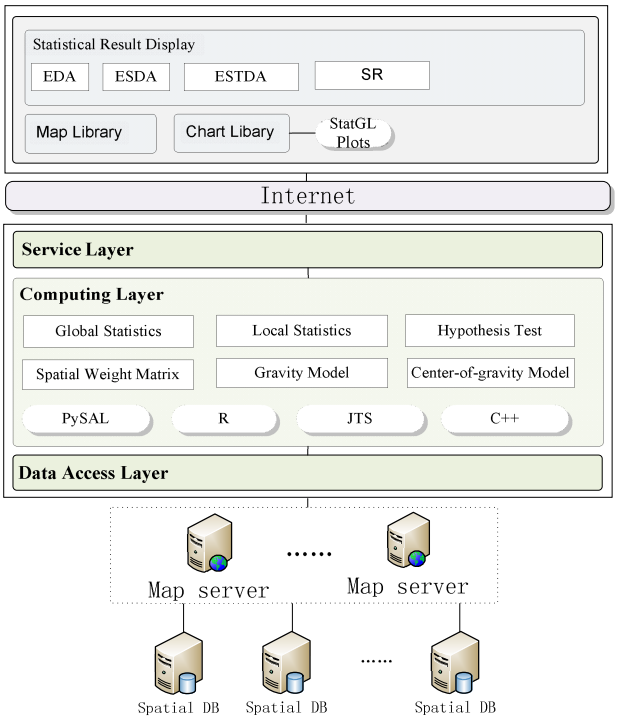

Figure 1. Architecture of the integrated environment

Figure 1 shows the architecture of the integrated web-based environment. The environment is incorporated into the China

Geo-Explorer (CGE), which employs a typical Browser/Server (B/S) structure. The server side is responsible for data storage and computing, and the client side takes the job of interaction and display of map and analysis process. SR in Figure 1 stands for spatial regression.

As shown in Figure 1, the server side of the integrated environment consists of three layers, the service layer, the data access layer and the computing layer.

3.1.1 Data Access Layer: The bottom layer is the data access layer which links to OGC-compliant servers. A Java open source mapping server, GeoServer, is employed in CGE. The data access layer is scalable to support multiple mapping servers, and automatically integrate data from various sources. Currently the data in CGE consist of mainly social-economic census data which are configured by system administrators, which are updated regularly. The environment also supports Shapefile upload by users. Our future work includes supporting user specifying the WFS address for customized data analysis.

The Web Map Service (WMS) is employed in the client side for users to quickly navigate the data and target the regions of their interests. The Web Feature Service (WFS) is used both in the client side for selection and the server side for spatial analysis.

An exception here is that for time-series data, although the new version WFS 2.0 (ISO 19142) has defined temporal filters, Geoserver has no support for it yet. Therefore for the time being, we use the Java Database Connectivity (JDBC) techniques to connect to the database system to get the time-series data, the spatial data are requested normally through WFS queries.

3.1.2 Computing Layer: The computing layer is the central layer of the environment. It first acquires data from the data access layer. Various kinds of spatial analysis methods discussed in Section 2 are then performed, using open source spatial analysis libraries as well as our own customized implementations.

Figure 2 shows the basic class diagram of the computing layer. Four packages are listed at the bottom of the figure. The basic entities are implemented in Java. We wrap a PreparedGeometry object from JTS into the Feature object. The TemporalFeature object is used for spatiotemporal data analysis, which extends the Feature class.

We abstract the data analysis process out, and define a combination of StatStrategy and TestStrategy classes for each analysis method. Libraries written in different languages are then wrapped into the extensions of these two basic classes. TestStrategy can be set to NULL if the analysis method doesn't require a test, such as the gravity model. This design is commonly known as strategy pattern (Gamma et al. 1994), where the algorithms for specific analysis method are implemented in related packages. The four input-output classes (StatInput, StatOutput, TestInput, and TestOutput) are built for encapsulating basic input and output for each analysis method. Implementations that extend these classes exist in related packages.

The PySAL library is written in Python. There are several ways to let Python and Java talk:

- Web service: Most standardized and extensible way, preferable if resources are placed separately.
- Executable Files: Invoke python scripts by opening a new process. This approach is stable and flexible, yet requires a lot of code written.
- JEPP: A JNI-style open source library, providing serialization support, relies on versions of both Java and Python.
- Jython: A reimplementation of Python upon Java. But PySAL relies on other libraries like numpy, scipy, which doesn't have counterparts in Jython for now.

CGE currently use executable files for communication with PySAL for its ease of use.

C++ can be easily linked to Java through the Java Native Interface (JNI). For R, we use Rserve, a TCP/IP server which allows convenient communication with Java and other languages.

All analysis method discussed in Section 2 can be divided into two parts: computation and visualization. Depending on the complexity of the method, the computation tasks are split between the client side and the server side. Computations that relate closely to analysis functions such as a simulation algorithm scale better in an server environment; while computations used for graphic display are better implemented in client side, avoiding unnecessary network transmission.

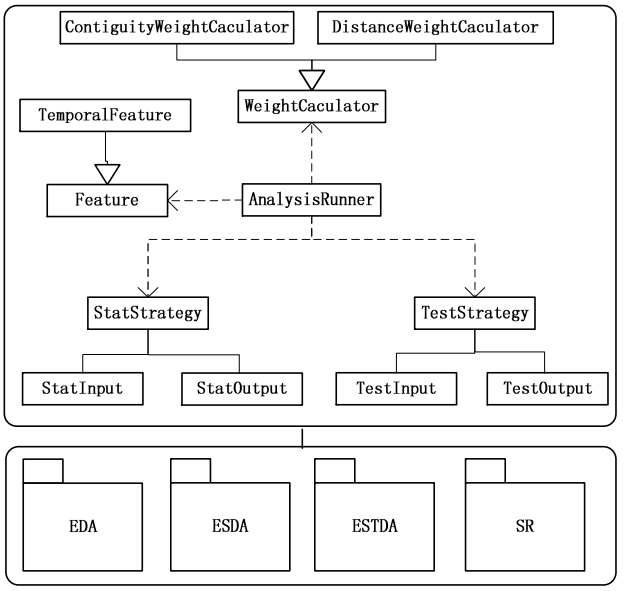

Figure 2. Basic class diagram of the computing layer

3.1.3 Service Layer: The standard WPS services could be used here to ensure interoperability. In this paper, we adopted an alternative way using Blazeds, a server-based Java remoting and web messaging technology. Blazeds is used for its convenience of data serialization and dynamic form construction, thus make users conduct spatial analysis in a more convenient way.

Suppose we want to calculate the global Moran'I coefficient of a selection area, we need to select the method to create a weight matrix and the hypothesis testing methods, but the parameters for different kinds of matrixes and hypothesis testing methods are varying. Concerning usability, it's desirable to incorporate these variations in one dynamic form displayed at the client side, such that when user change the type weight creation method from a combo box, the form can automatically changed

according to the method description. It is possible to achieve this dynamic form by writing multiple WPS services and combine them through chaining. The parameters are wrapped in a HashMap data structure. The server provides an interface for client to query the metadata of different analysis methods. The metadata of an analysis method contains a list of parameters and their validation information. The metadata is used in the client side to dynamically generate forms for user selection. Within this structure, both data sources and the analytic function can be dynamically added into the integrated environment.

The Blazeds combined with the configuration file is more straightforward in our experiment environment since we adopt Flash and Java as our development platform. In the future, we will wrap the analysis methods through WPS services so that third-party clients can communicate with the server in a more standard and unified way.

3.2 Client Side Design

Visualization is a key part in all exploratory analysis methods. The interactions should be responsive enough to meet the efficiency need. Rich Internet Application (RIA) is becoming increasingly popular for deploying such highly interactive contents. There exist many RIA solutions now, including Adobe Flash, Java applets, and Microsoft Silverlight. We chose Flash in our implementation for its high performance and the easiness to integrate with a Java server.

3.3 Algorithm Flow

In the server side design, the key objects of the integrated environment have been illustrated and the relationships to other libraries for spatial analysis have been built. In order to demonstrate this design, an algorithm flow is given in Figure 3 to illustrate the Moran's I computation using PySAL. The algorithm flows for other analysis methods are similar.

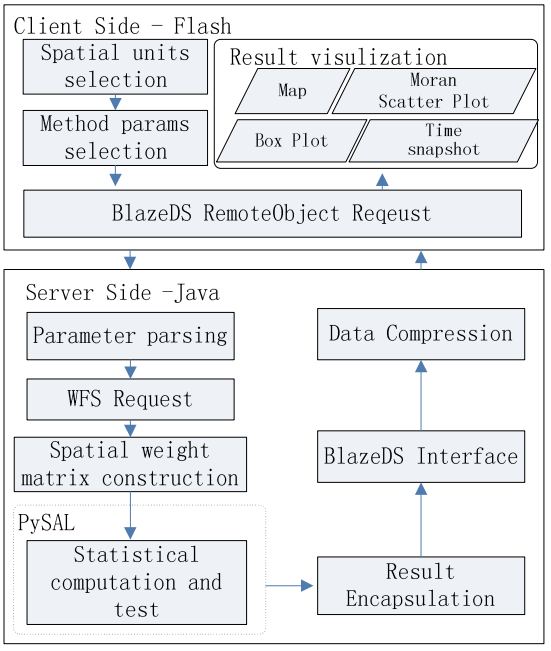

Figure 3. Flowchart of Moran's I computation

Step 1: Users interacts with the flash client, and selects the features that they find interested. Depending on the data selected, the flash client will then display methods correspondingly for the user. For example, if the selected data is temporal data, the system will display the ESTDA methods instead of ESDA methods. Users will then select method parameters and submit the request to the server.

Step 2: The server side will first parse the parameters. A set of WFS requests are then constructed and sent to corresponding map server. The received data are then wrapped into the Feature objects.

Step 3: The Java side will first generate a spatial weight matrix, write a temporary ESRI Shapefile, and invoke the Python scripts that execute PySAL functions. PySAL will perform statistical computation and hypothesis test for Moran's I.

Step 4: The calculations are wrapped into a Java HashMap data structure. After serialization, it is passed to the client side. Data will be compressed if the size exceeds a predefined limit.

Step 5: The client side parses the results and displays them in the Moran scatter plot, box plot and map. A time snapshot will be displayed if multiple time points are selected.

4. EXPERIMENTAL RESULTS

Our test environment is deployed on a server workstation with 2.49 GHz Intel Xeon E5420 CPU and 4GB RAM running Windows Server 2003. The applications are wrapped into Tomcat 6.0, a popular open source web server and container. We accessed the application in a personal computer with 2.66GHz Inter Core2 Duo CPU and 2GB memory running Windows XP, and we used Google chrome browser in the following experiment.

Following the algorithm flow described in Section 3.3, the first step is to select the spatial units by drawing shapes on the map. The system will then construct a WFS query for the shape. Figure 4 illustrates the selection process. The user selects all counties in three provinces in central China.

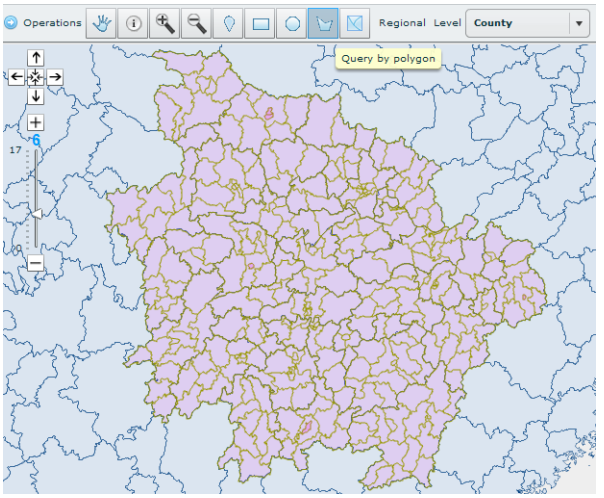

Figure 4. Spatial unit selection

After selection, the data are ready for analysis. User can directly go to spatial methods analysis panel. For the data selected in Figure 4, the system will recognize it as spatial data, thus can only read methods related to spatial data, and then pop up a tree of method names as shown in Figure 5, after user selects a method in the tree, the parameters will be dynamically added at the bottom.

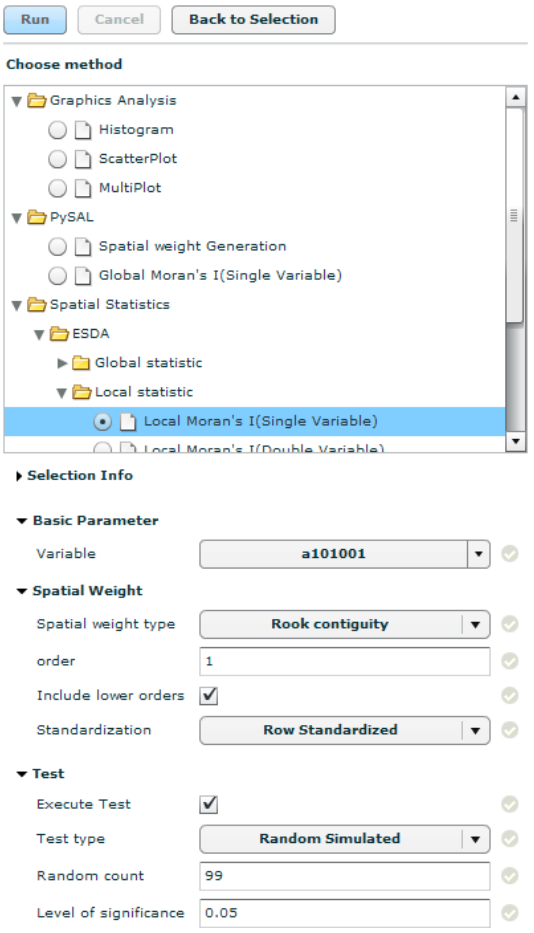

Figure 5. Analysis method and parameter selection

Figure 6 shows a possible analysis result. The histogram shows the distribution of a given variable. The multi-plot can be used to check relations of multiple variables. The Moran scatter plot on the right is used to explore the spatial autocorrelation.

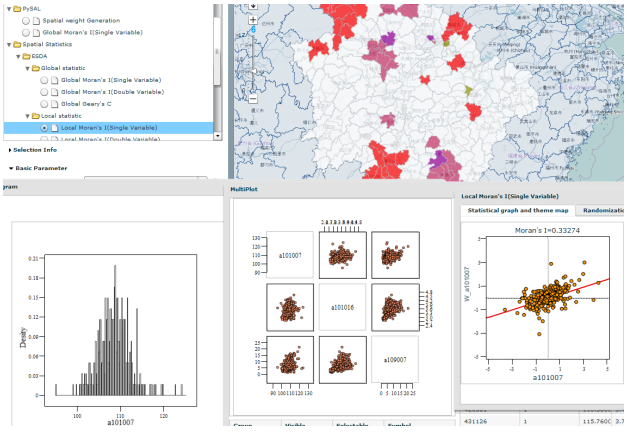

Figure 6. EDA and ESDA plots with the pattern map

Figure 7 shows the analysis results of a gravity model for total investment in fixed assets in twelve provinces in northern China from 2000 to 2008. On the gravity map, a circle stands for a total gravity value within one district, and a line stands for a gravity value between two districts. The gravity intensity is coloured in different sizes of circles and lines. The arrow of the line stands for the attraction flow direction. The "snapshot control" supports animation with a flexible switch to a specific time point. The animation effect is smooth during transitions, which is built on the Tween facilities provided in the Flex framework.

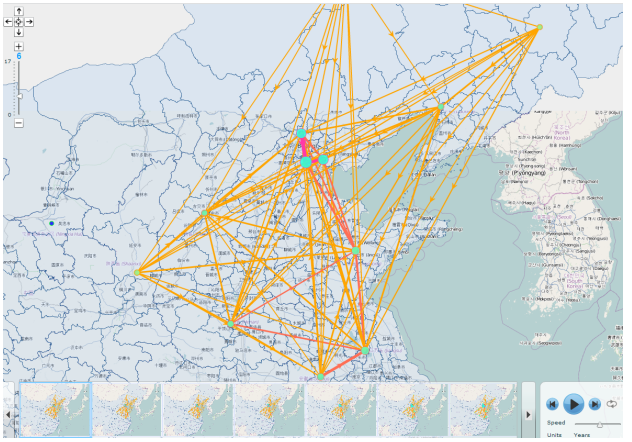

Figure 7. Spatiotemporal gravity model visualization

Table 1 gives performance evaluation for some of the analysis methods in our integrated environment. It shows that the analysis can be done in reasonable time for moderate size dataset, thus suitable for online data analysis. The measurements, however, can only seen as a coarse estimate, because the actual cost depends on many dynamic factors, such as complexities of selected spatial objects, parameter selection of the analysis method, drawing speed of the client browser, etc.. In our evaluation, the total running time is equal to the time when the client completes drawing the analysis result minus the time when client sends the request to the server. The experiment was taken under a local area network, thus the time didn't take into account the communication cost over the internet. We used China counties as the input layer in the evaluation.

Analysis method	Total Time consumed (second) Grouped by count of selected spatial units			
	50	**100**	**200**	**500**
Scatter plot	0.265	0.36	0.406	1.078
Spatial weight matrix	1.359	1.797	2.547	4.954
Global Moran's I	2.594	3.109	3.344	5.657
Centre-of-gravity model	0.297	0.578	0.86	1.515

Table 1. Performance evaluation for some analysis methods

5. CONCLUSION

The gap between the spatial data and the analysis procedures prevents users from effectively conducting spatial analysis. This paper investigates an integrated web-based framework, which integrates multiple data sources and several open source software tools for spatiotemporal exploratory data analysis.

Users can navigate spatial data and perform spatial analysis directly on them, without performing the time-consuming job of data acquisition and integration. The environment released users from the great burden of intermediate data processing unrelated to their work. The power of web 2.0 techniques such as Flash can be seamlessly integrated with server side computing environment to provide reliable and responsive spatial applications. Some future directions are given below.

- Provide the analysis methods as standard WPS services to ensure interoperability.
- Cloud computing technologies are emerging rapidly. There are many mature solutions provided by Amazon, Google, etc. The analysis functions can be more easily scalable to large spatiotemporal dataset in a Cloud platform.
- More advanced and visually compelling analytic methods can be integrated in this framework, like space-time cubes. Optimizations strategies need to be investigated for each visualization method to deal with large data set.
- The current framework provides a fusion mechanism for various computing environment. Other functions in PySAL, R and analysis functions in other software packages can be integrated into this framework.

6. REFERENCE

Anselin, L., 1995. Local Indicators of Spatial Association-LISA. *Geographical Analysis*, 27(2), pp. 93-115.

Anselin, L., Y. W. Kim, and I. Syabri, 2004. Web-based analytical tools for the exploration of spatial data. *Journal of Geographical Systems*, 6, pp. 197-218

Castelli, M. and C. Scavia, 2008. A multidisciplinary methodology for hazard and risk assessment of rock avalanches. *Rock Mechanics and Rock Engineering*, 41, pp. 3-36.

Data Numerica Institute, 2011. StatGL Graphical Library: A foundation for online graphical data analysis. Data Numerica Institute, Bellevue. http://datanumerica.com/StatGL.html(20 Dec. 2011)

de Smith, M. J., M. F. Goodchild and P. A. Longley, 2011. *Geospatial analysis: A comprehensive guide to principles, techniques and software tools*. The Winchelsea Press, Winchelsea.
http://www.spatialanalysisonline.com/output/topics/ExploratorySpatialDataAnalysis.html (20 Dec. 2011)

Ellawaya, A., L. Macdonalda, A. Forsyth and S. Macintyre, 2010. The socio-spatial distribution of alcohol outlets in Glasgow city. *Health & Place*, 16, pp 167-172.

Gamma, E., R. Helm, R. Johnson and J. Vlissides, 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman, Reading, pp. 315-325.

Hardisty, F. and A. Klippel, 2010. Analysing spatio-temporal autocorrelation with LISTA-Viz. *International Journal of Geographical Information Science*, 24(10), pp. 1515-1526.

Koua, E. L. and M.-J. Kraak, 2004. A Usability Framework for the Design and Evaluation of an Exploratory Geovisualization Environment. In: Proceedings of the Eighth International Conference on Information Visualisation, Washington, DC, USA, pp. 153-158.

Kulldorff, M. and N. Nagarwalla, 1995. Spatial Disease Clusters: Detection and Inference. *Statistics in Medcine*, 14, pp. 799-810.

PyWPS, 2012. Python web processing service. WWW document, http://pywps.wald.intevation.org (2 April. 2012).

Rey, S. J., L. Anselin and M. Hwang, 2009. Manipulation of Spatial Weights Using Web Services. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, New York, NY, USA, pp. 72-80.

Rey, S. J. and L. Anselin, 2010. PySAL: A Python Library of Spatial Analytical Methods. In *M. Fischer and A. Getis (eds.) Handbook of Applied Spatial Analysis: Software Tools, Methods and Applications*. Springer, Berlin, pp. 175-193

She, B., X.-y. Zhu and S.-m. Bao, 2010. Spatial data integration and analysis with spatial intelligence. In: 18th International Conference on Geoinformatics, Geoinformatics 2010 Conference, Beijing, China, pp. 1-6.

WPSint, 2012. Spring plug-in for OGC Web Processing Service (WPS) interface. WWW document, http://wpsint.tigris.org (2 April. 2012).

Yue, P., J.-y. Gong, L.-p. Di, J. Yuan, L.-z. Sun, Z.-h. Sun and Q. Wang, 2010. GeoPW: Laying Blocks for the Geospatial Processing Web. *Transactions in GIS*, 14, pp. 755-772.

ZOO, 2012. ZOO Web Processing Service. WWW document, http://www.zoo-project.org (2 April. 2012).

7. ACKNOWLEDGEMENTS

The research is founded by National Science & Technology Pillar Program (2012BAH35B03), National High Technology Research and Development Program of China (863 Program) (No. 2011AA010500), the Fundamental Research Funds for the Central Universities, and Open Research Fund of LIESMARS. The authors would like to thank the editor and reviewers for their thorough review and valuable comments. The authors would also like to thank Dr. Edward C. Chao for providing the StatGL library and his valuable support.