TerraFly GeoCloud: An Online Spatial Data Analysis and Visualization System

Mingjin Zhang, Florida International University
Huibo Wang, Florida International University
Yun Lu, Florida International University
Tao Li, Florida International University
Yudong Guang, Florida International University
Chang Liu, Florida International University
Erik Edrosa, Florida International University
Hongtai Li, Florida International University
Naphtali Rishe, Florida International University

With the exponential growth of the usage of web map services, the geo data analysis has become more and more popular. This paper develops an online spatial data analysis and visualization system, TerraFly GeoCloud, which facilitates end users to visualize and analyze spatial data, and to share the analysis results. Built on the TerraFly Geo spatial database, TerraFly GeoCloud is an extra layer running upon the TerraFly map and can efficiently support many different visualization functions and spatial data analysis models. Furthermore, users can create unique URLs to visualize and share the analysis results. TerraFly GeoCloud also enables the MapQL technology to customize map visualization using SQL-like statements. The system is available at http://terrafly.fiu.edu/GeoCloud/.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Data mining, Spatial databases and GIS

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Geospatial analysis, GIS, Visualization, Big Data

1. INTRODUCTION

With the exponential growth of the World Wide Web, there are many domains, such as water management, crime mapping, disease analysis, and real estate, open to Geographic Information System (GIS) applications. The Web can provide a giant amount of information to a multitude of users, making GIS available to a wider range of public users than ever before. Web-based map services are the most important application of modern GIS systems. For example, Google Maps currently has more than 350 million users. There are also a rapidly growing number of geo-enabled applications which utilize web map services on traditional computing platforms as well as the emerging mobile devices.

However, due to the highly complex and dynamic nature of GIS systems, it is quite challenging for end users to quickly understand and analyze the spatial data, and to efficiently share their own data and analysis results to others. First, typical geographic visualization tools are complicated and

This material is based in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, FRP IIP-1230661, SBIR IIP-1058428, SBIR IIP-1026265, SBIR IIP-1058606, SBIR IIP-1127251, SBIR IIP-1127412, SBIR IIP-1118610, SBIR IIP-1230265, SBIR IIP-1256641. Includes material licensed by TerraFly (http://terrafly.com) and the NSF CAKE Center (http://cake.fiu.edu).
Author's addresses: M. Zhang, H. Wang, Y. Lu, T. Li, Y. Guang, E. Edrosa, H. Li, N. Rishe, School of Computing and Information Sciences, Florida International University; 11200 SW 8th Street, Miami, FL, 33199, USA.

DOI : http://dx.doi.org/10.1145/0000000.0000000

fussy with a lot of low-level details, thus they are difficult to use for spatial data analysis. Second, the analysis of large amount spatial data is very resource-consuming. Third, current spatial data visualization tools are not well integrated for map developers and it is difficult for end users to create the map applications on their own spatial datasets.

To address the above challenges, this paper presents TerraFly GeoCloud, an online spatial data analysis and visualization system, which allows end users to easily visualize and analyze various types of spatial data. TerraFly GeoCloud offers the following important features to facilitate the spatial data analysis.

— First, TerraFly GeoCloud can accurately visualize and manipulate point and polygon spatial data with just a few clicks.
— Second, TerraFly GeoCloud employs an analysis engine to support the online analysis of spatial data, and the visualization of the analysis results. Many different spatial analysis functionalities are provided by the analysis engine.
— Third, based on the TerraFly map API, TerraFly GeoCloud offers a MapQL language with SQL-like statements to execute spatial queries, and render maps to visualize the customized query results.

Our TerraFly GeoCloud online spatial data analysis and visualization system is built upon the TerraFly system using TerraFly Maps API and JavaScript TerraFly API add-ons in a high performance cloud Environment. The function modules in the analysis engine are implemented using C and R language and python scripts. Comparing with current GIS applications, our system is more user-friendly and offers better usability in the analysis and visualization of spatial data. The system is available at http://terrafly.fiu.edu/GeoCloud/.

A preliminary version of the work focusing on visualization solutions (e.g., map rendering and spatial data visualization) is published in [Lu et al. 2013a]. In this journal submission, we added many spatial analysis functions and also made the result visualization more interactive. With these changes TerraFly Geocloud became more intelligent and can be applied in many application domains, such as disease analysis, crime analysis, and real estate analysis. We present several application case studies including Florida property analysis and Lung cancer analysis to demonstrate the usefulness of the system.

In summary, the TerraFly GeoCloud system is a type of intelligent decision support system. By leveraging distributed computing, map rendering, visualization technologies, and spatial data mining techniques, TerraFly GeoCloud enables users to perform different types of spatial data analysis tasks for decision support (e.g., gathering and analyzing data, identifying/diagnosing problems, proposing possible actions and strategies, and evaluating the proposed actions and strategies) [Matsatsinis and Siskos 2003]. Analysis functions supported in TerraFly GeoCloud include spatial data visualization, spatial dependency and auto-correlation, spatial data clustering, spatial regression, measuring geographic distribution, spatial interpolation, and customize map visualization. It also leverages rich user interactions to perform data analysis and support human decision intelligently. Two real case studies including Florida property analysis and Lung Cancer analysis using GeoCloud shows how TerraFly GeoCloud helps user perform data analysis and visualization to make decisions. The rest of this paper is organized as follows: Section 2 describes the architecture and the system overview of TerraFly GeoCloud; Section 3 describes the visualization and analysis methods in TerraFly GeoCloud; Section 4 describes the MapQL spatial query language and customized map visualization with MapQL; Section 5 studies the system performance for both on-line and off-line analysis; Section 6 presents the case studies on the online spatial analysis; Section 7 discusses the related work; and finally Section 8 concludes the paper.

2. SYSTEM OVERVIEW

TerraFly GeoCloud is built upon the TerraFly system to support various kinds of online spatial data analysis using TerraFly Maps API and TerraFly API add-ons in a high performance cloud Environ-

ment. We first introduce the TerraFly system and then describe the overall system demonstration of GeoCloud.

2.1. TerraFly

TerraFly is a system for querying and visualizing of geospatial data developed by High Performance Database Research Center (HPDRC) lab in Florida International University (FIU). This TerraFly system serves worldwide web map requests over 125 countries and regions, providing users with customized aerial photography, satellite imagery and various overlays, such as street names, roads, restaurants, services and demographic data [Rishe et al. 2001; Rishe et al. 2005].

TerraFly allows users to virtually fly over enormous geographic information simply via a web browser with a bunch of advanced functionalities and features such as user-friendly geospatial querying interface, map display with user-specific granularity, real-time data suppliers, demographic analysis, annotation, route dissemination via autopilots and API for web sites, etc. TerraFly's server farm ingests geolocates, mosaics, and cross-references 40TB of base map data and user-specific data streams.

2.2. TerraFly GeoCloud

Figure 1 shows the system architecture of TerraFly GeoCloud. Based on the current TerraFly system including the Map API and all sorts of TerraFly data, we developed the TerraFly GeoCloud system to perform online spatial data analysis and visualization. In TerraFly GeoCloud, users can import and visualize various types of spatial data (data with geo-location information) on the TerraFly map, edit the data, perform spatial data analysis, and visualize and share the analysis results to others. Available spatial data sources in TerraFly GeoCloud include but not limited to demographic census, real estate, disaster, hydrology, retail, crime, and disease. In addition, the system supports MapQL, which is a technology to customize map visualization using SQL-like statements.

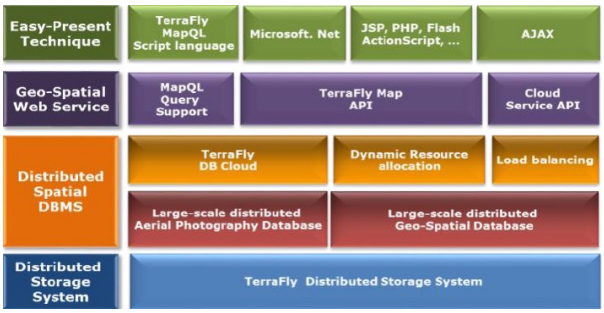


Fig. 1: The Architecture of TerraFly GeoCloud

The spatial data analysis functions provided by TerraFly GeoCloud include spatial data visualization (visualizing the spatial data), spatial dependency and autocorrelation (checking for spatial dependencies), spatial clustering (grouping similar spatial objects),spatial regression, measuring Geographic Distribution and Kriging (geo-statistical estimator for unobserved locations).

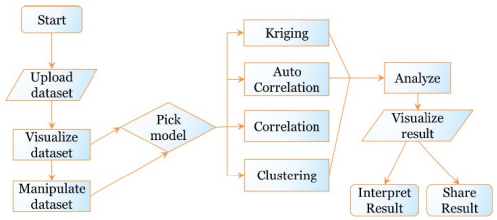
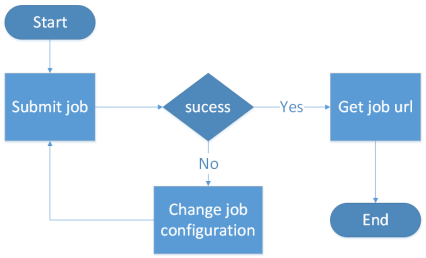
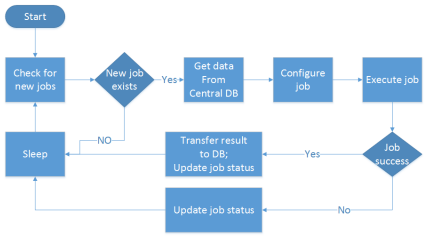


Fig. 2: The Workflow of TerraFly Geocloud

Figure 2 shows the data analysis workflow of the TerraFly GeoCloud system. Users first upload datasets to the system, or view the available datasets in the system. User can upload GeoJson, Shapefile and .asc file. They can then visualize the data sets with customized appearances. By Manipulating the dataset, users can edit the dataset and perform pre-processing (e.g., adding more columns). Followed by pre-processing, users can choose proper spatial analysis functions and perform the analysis. After the analysis, they can visualize the results and also share them with others.



(a) The front-end workflow of offline analysis



(b) The back-end workflow of offline analysis

Fig. 3: The workflow of offline analysis

Geocloud also supports offline analysis, if users want to perform analysis on large data sets. Figure 3 shows the workflow of the offline analysis in TerreFly GeoCloud. The workflow in the front-end is shown in Figure 3a. Users can submit jobs through the GeoCloud website. If the job submission failed, users should change the job configurations. If a job is accepted successfully, the user will receive a URL from which the analysis results can be downloaded. The offline job status can be shown through the URL. Figure 3b shows the back-end workflow of the offline analysis. The system polls the database for new jobs. If a new job exists, first, the system will retrieve data from the central DB. Second, the system will configure the job using the submitted configuration. Third, the system will copy the data to HDFS, send the job to the GeoCloud hadoop platform, and run the hadoop job. If the job is successfully completed, the results will be transferred to the database. After the jobs status being updated, users can download the analysis results through the URL.

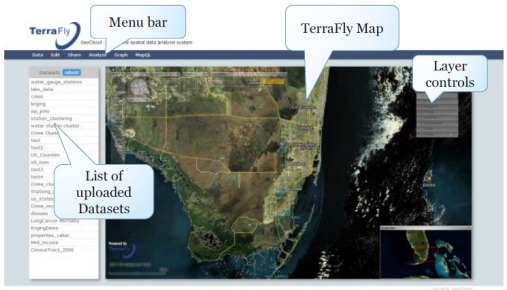


Fig. 4: Interface of TerraFly Geocloud

Figure 4 shows the interface of the TerraFly GeoCloud system. The top bar is the menu of all functions, including Data, analysis, Graph, Share, and MapQL. The left side shows the available datasets, including both the uploaded datasets from the user and the existing datasets in the system. The right map is the main map from TerraFly. This map is composed by TerraFly API, and it includes a detailed base map and diverse overlays which can present different kinds of geographical data.

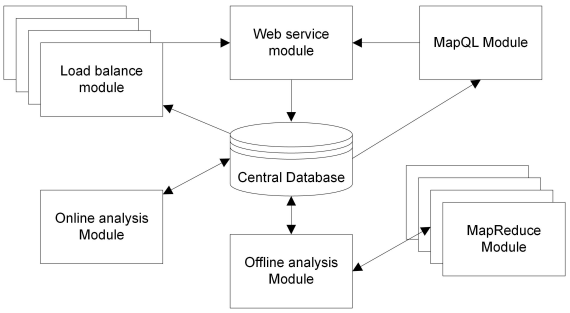


Fig. 5: Modules of the GeoCloud system

Figure 5 shows the main function modules of the GeoCloud system. The center of the system is a central database which holds all the system related data. The central database composed by the sksOpen database, the map file database, and the relational databases such as SQL Server and PostGreSQL. The sksOpen database is a spatial object hybrid index and storage system that includes both an R-Tree spatial index and an inverted text file index, which attained fast retrieval of spatial data even when the matching objects were located far away from one another [Lu et al. 2013b]. The map file database provides the base map for users, and the relational databases are used for storing the uploaded data and the analysis results. The online and offline analysis modules process the analysis tasks and push back the results to the Central Database. The online analysis module processes analysis tasks which can be done at runtime while the offline analysis module employs the MapReduce module to process heavy duty tasks. The load balance module and web service module leverage distributed spatial data visualization with autonomic resource management techniques to provide the on-demand and balanced resource allocation to achieve the QoS (Quality of service).

TerraFly GeoCloud also provides MapQL spatial query and render tools. MapQL supports SQL-like statements to realize the spatial query, and render the map according to users inputs. MapQL tools can help users visualize their own data using a simple statement. This provides users with a

better mechanism to easily visualize geographical data and analysis results. Shown in Figure 5, the MapQL module creates map visualization at runtime based on the MapQL statements.

3. VISUALIZATION AND ANALYSIS METHODS

Many different visualization functions and spatial data analysis models are provided in TerraFly GeoCloud. TerraFly GeoCloud also integrates spatial data mining and data visualization. The spatial data mining results can be easily visualized. In addition, visualization can often be incorporated into the spatial mining process.

3.1. Spatial Data Visualization

Fig. 6: Spatial Data Visualization: Point data and Polygon Data

For spatial data visualization, the system supports both point data and polygon data and users can choose color or color range of data for displaying. As shown in Figure 6, the point data is displayed on left, and the polygen data is displayed on the right. The data labels are shown on the base map as extra layers for point data, and the data polygons are shown on the base map for polygon data. Many different visualization choices are supported for both point data and polygon data. For point data, users can customize different parameters such as the icon style, icon color or color range, and label value. For polygon data, users can customize different parameters including the fill color or color range, fill alpha, line color, line width, line alpha, and label value.

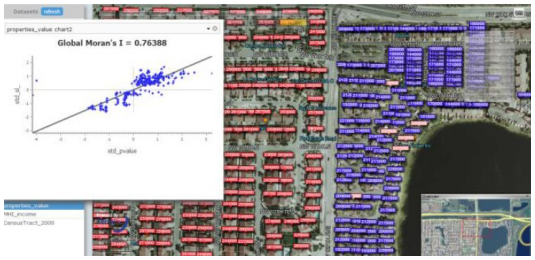
3.2. Spatial Dependency and Auto-Correlation

Spatial dependency is the co-variation of properties within the geographic space: characteristics at proximal locations that appear to be correlated, either positively or negatively. Spatial dependency leads to the spatial autocorrelation problem in statistics [De Knegt et al. 2010]. Spatial autocorrelation is more complex than one-dimensional autocorrelation because spatial correlation is multi-dimensional and multi-directional. The TerraFly GeoCloud system provides auto-correlation analysis tools to discover spatial dependencies in a geographic space, including global and local clusters analysis where Moran's I measure is used [Li et al. 2007]. Formally, Morans I, the slope of the line, estimates the overall global degree of spatial autocorrelation as follows:

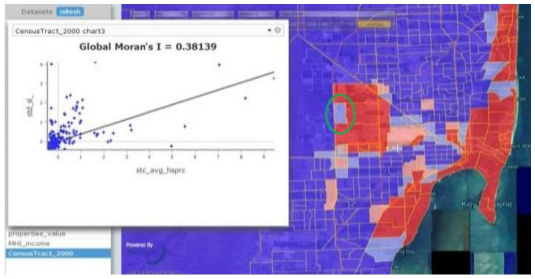
$$I = \frac{n}{\sum_i^n \sum_j^n w_{ij}} * \frac{\sum_i^n \sum_j^n w_{ij}(y_i - \hat{y})(y_i - \hat{y})}{\sum_i^n (y_i - \hat{y})^2}, \tag{1}$$

where w_{ij} is the weight, $w_{ij} = 1$ if locations i and j are adjacent and zero otherwise $w_{ii} = 0$ (a region is not adjacent to itself).y_i and $\hat{y}$ are the variable in the i-th location and the mean of the variable, respectively. n is the total number of observations. Morans I is used to test hypotheses concerning the correlation, ranging between 1.0 and $+1.0$. Morans I measures can be displayed as a checkerboard where a positive Morans I measure indicates the clustering of similar values and a negative Morans I measure indicate dissimilar values. TerraFly GeoCloud provides auto-correlation analysis tools to check for spatial dependencies in a geographic space, including global and local clusters analysis.

Figure 7b shows an example of spatial auto-correlation analysis on the average properties price by zip code data in Miami (polygondata). Each dot here in the scatterplot corresponds to one zip code. The first and third quadrants of the plot represent positive associations (high-high and low-low), while the second and fourth quadrants represent associations (low-high, high-low). For example, the green circle area is in the low-high quadrants. The density of the quadrants represents the dominating local spatial process. The properties in Miami Beach are more expensive, and are in the high-high area. Figure 7a presents the auto-correlation analysis results on the individual properties price in



(a) Properties value in Miami



(b) Average properties price by zip code in Miami

Fig. 7: Spatial Dependency and Auto-Correlation

Miami (point data). Each dot here in the scatterplot corresponds to one property. As the figure shows, the properties near the big lake are cheaper, while the properties along the west are more expensive.

3.3. Spatial Data Clustering

Spatial data clustering algorithms identify clusters, or densely populated regions, according to some distance measures in a large, multidimensional dataset. Several spatial clustering techniques are provided in TerraFly GeoCloud.

K-Means. K-means is an efficient clustering algorithm. K-means partition all the data set in to k cluster. Firstly, the algorithm will randomly find k initial center points. Secondly, finding the nearest center point for each record as its cluster and getting mean value for each cluster as new cluster center. Repeating first and second step until the cluster center doesn't change. In TerraFly GeoCloud system, user can apply k-means algorithm by inputing cluster number.

DBSCAN. The TerraFly GeoCloud system supports the DBSCAN (for density-based spatial clustering of applications with noise) data clustering algorithm [Ester et al. 1996]. DBSCAN is a density-based clustering algorithm and it finds a number of clusters starting from the estimated density distribution of corresponding nodes. DBSCAN requires two parameters as the input: eps (the neighbor size) and minPts (the minimum number of points required to form a cluster). It starts with

an arbitrary starting point that has not been visited so far. This point's neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as a noise point [Ester et al. 1996]. If a point is found to be a dense part of a cluster, its neighborhood is also part of that cluster. Hence, all points that are found within the neighborhood are added. This process continues until the density-connected cluster is completely identified. Then, a new unvisited point is retrieved and processed, leading to the discovery of new cluster or noise points [Bilodeau et al. 2005]. Figure 8a shows an example of DBSCAN clustering on the crime data in Miami. As shown in Figure 8a, each point is an individual crime record marked on the place where the crime happened, and the number displayed in the label is the crime ID. By using the clustering algorithm, the crime records are grouped, and different clusters are represented by different colors on the map.

Cluster Detection. Kulldorff & Nagarwalla(KN)[Kulldorff 1997] provides a method to perform cluster detection. KN method is implemented by scanning all the area using circular zones of variable size. KN method is widely used in spatial epidemiology. The steps of KN method include: (1). Move a circle in space to obtain an infinite number of overlapping circles; (2). Compute LLR (Log Likelihood Ratio) of each circle and sort the LLR; and (3). Get some large LLR then use Monte Carlo method to calculate P-value of them. The Log Likelihood Ratio can be calculated as follow:

$$LLR = max_j(\frac{Y_j}{E_j})^{Y_j}(\frac{Y_+ + Y_j}{Y_+ - E_j})^{Y_+ - Y_J} I(Yj > Ej), \tag{2}$$

where Y_j denotes the observed number of instance in circle area, $Y+$ denotes the number of instance in all the area, E_j denotes the expected number of instance in circle area. Figure 8b shows the result of lung cancer cluster map in Florida. The red points indicate the disease cluster where the unusual disease case happened. The number in the red point is the p-value of each area.[Elliott and Wartenberg 2004]

HotSpot. HotSpot analysis function using Gi* statistic method aims to detect the hot (or cold) cluster which has a high (or a low) Gi* value. Figure 8c shows the result of the hotspot cluster map of lung cancer mortality in Florida. From this map, we can observe that the central part which is covered by red color is a hot cluster and four counties in the south region forms a cold cluster.

Outlier Analysis. Outlier analysis recognizes the outliers whose attributes values are different from their neighbors. In TerraFly GeoCloud, local moran's I map, z-value map, and p-value map are provided.

3.4. Spatial Regression

Regression tools can be used to estimate relationships between attributes.

Linear Regression. TerraFly GeoCloud provides linear regression tools with multiple tests, such as global morans I test. Figure 9a shows the linear regression results between mortality and median house price and median income. It should be noted that global Morans I test indicates that the residual is geo-correlated, and thus linear regression model is not a good fit for this problem.

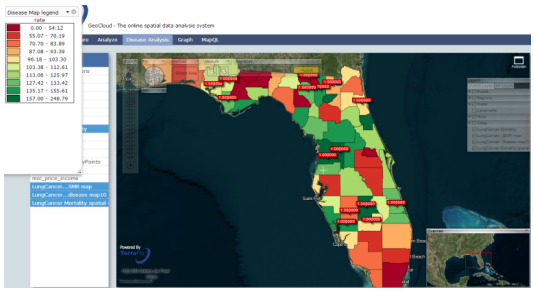
Spatial auto-regression. In spatial auto-regression, a lag model and an error model are provided. The spatial auto-regression lag model can be calculated as follows:

$$Y = \rho Wy + x\beta + \epsilon, \tag{3}$$

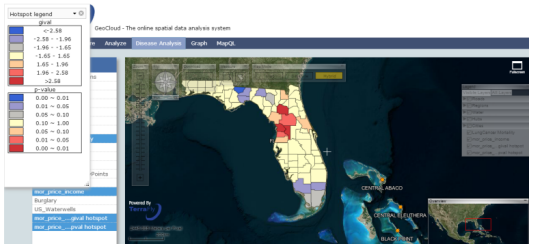
where Y is a dependent variable, W is a matrix of spatial weights, x is an independent variable, β denotes the unknown parameters, and ϵ is an error term.

Figure 9b shows the result of a spatial auto-regression lag model. In this model, multiple test methods are provided for verifiability: Wald test is used to determine whether various parameters can be zero or not; AIC for linear regression and lag model is applied to indicate which model is better; LR test, the Likelihood Ratio diagnostics, is used for testing spatial dependence; and LM test

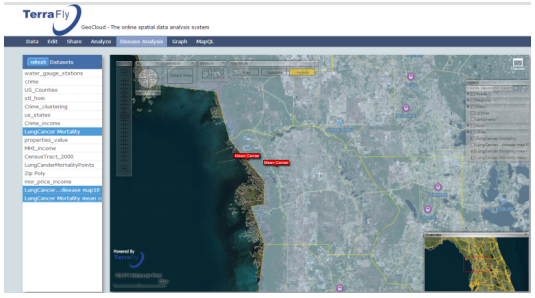
(a) DBSCAN clustering on the crime data in Miami



(b) KN cluster detection on lung cancer in Florida



(c) Hotspot clustering on lung cancer in Florida



(d) Center point and weighted center point

Fig. 8: Spatial Clustering in Geocloud

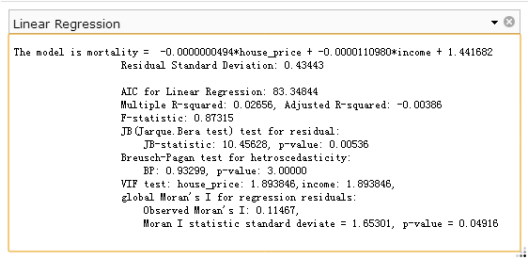
is utilized for evaluating the absence of spatial autocorrelation in lag model residuals [Dubin et al. 1999][Kelejian and Prucha 1998].

3.5. Measuring Geographic Distribution

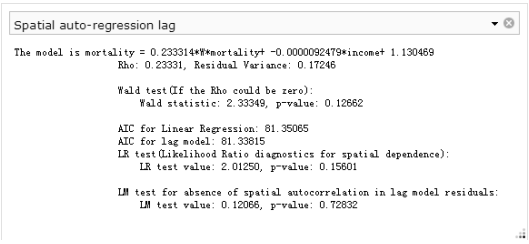
Geographic distribution measurements include mean/median central, standard distance, and distributional trends functions. In our system, a weighted mean central is provided as follow:

$$X = \frac{\Sigma_i w_i x_i}{\Sigma_i w_i}, Y = \frac{\Sigma_i w_i y_i}{\Sigma_i w_i}, \tag{4}$$

where x_i and y_i denote the coordinate of each point (but when the data set is polygonal, x_i and y_i indicate the center of each polygon) and w_i is the weight which corresponds in our system to mortality or incidence. Figure 8d shows these two type of points: one type is the non-weighted center point, and the other type is the lung cancer mortality weighed center point. Besides the center/median point function, TerraFly GeoCloud also includes distributional trends and standard distance.



(a) Linear regression tool on lung cancer in Florida



(b) Spatial auto-regression lag model on lung cancer in Florida

Fig. 9: Spatial Regression in Geocloud

3.6. Spatial Interpolation Method

Kriging is a geo-statistical estimator that infers the value of a random field at an unobserved location (e.g. elevation as a function of geographic coordinates) from samples (see spatial analysis) [Stein 1999] Figure 10 shows an example of Kriging. The data set is the water level from water stations

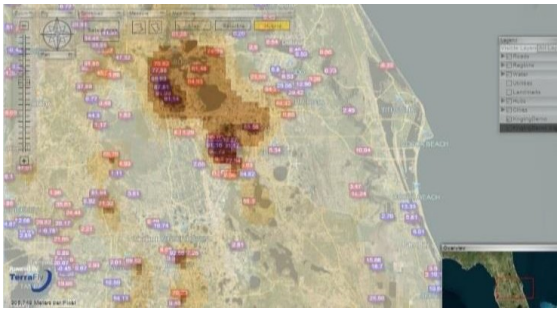


Fig. 10: Kriging data of the water level in Florida

in central Florida. Note that not all the water surfaces are measured by water stations. The Kriging results are estimates of the water levels and are shown by the yellow layer.

4. CUSTOMIZED MAP VISUALIZATION

TerraFly GeoCloud also provides MapQL spatial query and render tools, which supports SQL-like statements to facilitate the spatial query and more importantly, render the map according users requests. This is a better interface than API to facilitate developer and end user to use the TerraFly map as their wish. By using MapQL tools, users can easily create their own maps.

4.1. Introduction and Implementation

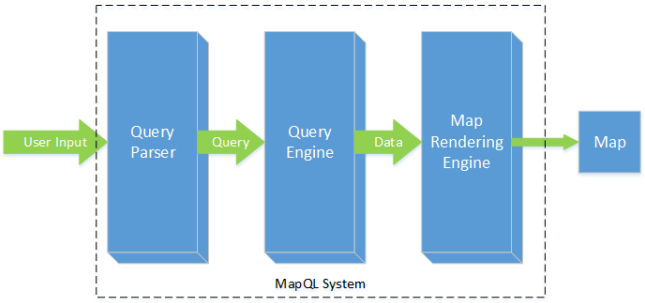


Fig. 11: MapQL System Architecture

MapQL is an extension of GeoSPARQL, which is a standard for representation and querying of geospatial linked data. MapQL defined some new key words that include T_ICON_PATH, T_LABEL, T_LABEL_SIZE, T_FILED_COLOR, T_THICKNESS, T_OPACITY and T_BORDER_COLOR to facilitate customized map visualization. The architecture of MapQL is shown in Figure 11. MapQL contains three modules: Query parser, Query Engine, and Map Rendering Engine. Query Parser checks syntax and semantic correctness of the input query. After passing Query Parser, the query goes to Query Engine where it is committed to the database. The PostGreSQL database, which has a very good support for spatial data indexing and query, is used in the Query Engine module. The returned results from Query Engine will be processed at Map Rendering Engine. Mapnik, a toolkit for making customized map, is used in Map Rendering Engine to create customized maps and put them as a layer on TerraFly map through TerraFly map API. The workflow of MapQL is shown in Figure 12. The input of the whole procedure is MapQL statements, and the output is map visualization rendered by the MapQL engine.

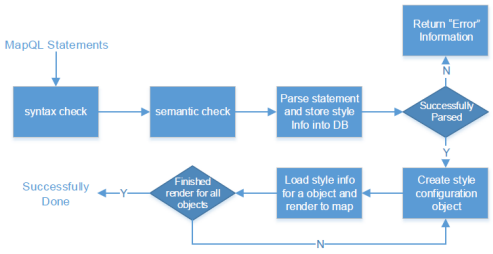


Fig. 12: The workflow of MapQL

Shown in Figure 12, the first step is the syntax check of the statements. The syntax check guarantees that the syntax of an input query conforms to the standard (e.g., the spelling-check of the reserved words). The semantic check ensures that the data source name and metadata which MapQL statements want to visit are correct. After the above two checks, the system will parse the statements and store the parse results including the style information into a spatial database. The style information includes where to render and what to render. After all the style information is stored, the system will create style configuration objects for rendering. The last step is for each object, load the style information from the spatial database and render to the map according to the style information.

We implemented the MapQL tools using C++. For the last step of rendering the objects to the map visualization, we employed the TerraFly map render engine.

4.2. Query Examples

For example, if we want to query the house prices near Florida International University, we use MapQL statements in Figure 13. There are four reserved words in the statements, T_ICON_PATH ,

```
SELECT
    '/var/www/cgi-bin/house.png' AS T_ICON_PATH,
r.price AS T LABEL,
    '15' AS T LABEL SIZE,
r.geo AS GEO
FROM
realtor 20121116 r
WHERE
ST Distance(r.geo,  GeomFromText('POINT(-80.376283 25.757228)')) <
0.03;
```

Fig. 13: Query house prices using MapQL

T_LABEL, T_LABEL_SIZE , and GEO. We use T_ICON_PATH to store the customized icon. Here we choose a local png file as icon. T_LABEL denotes that icon label that will be shown on the map, T_LABEL_SIZE is the pixel size of the label, and GEO is the spatial search geometry. The statements go through the syntax check first. If there is incorrect usage of reserved words or wrong spelling of the syntax, the statements will be corrected or Error information will be sent to the user. For example, if the spelling of select is not correct, Error information will be sent to the user. The semantic check makes sure that the data source name realtor_20121116 and metadata r. price and r.geo are exist and available. After the checks, the system parsed the statements. The SQL part will return corresponding results including the locations and names of nearby objects, the MapQL part will collect the style information including icon path and icon label style. Both of them are stored into a spatial database. The system then created style configuration objects for query results. The last step is rendering all the objects on the map visualizations. The needed style information includes icon picture and label size, and the data information includes label value and location (Lat, Long). Figure 14 shows the result of this query.

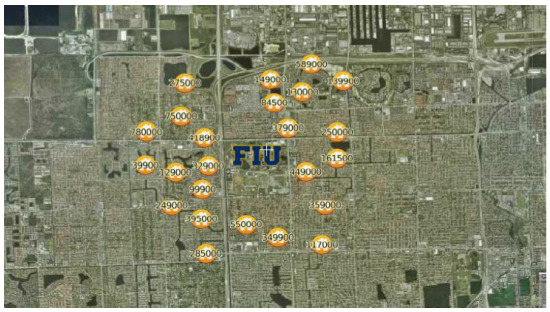


Fig. 14: Result of query house prices using MapQL

In the following, we present several query examples using MapQL statements. Figure 15 shows all the hotels along a certain street within a certain distance and also displays the different stars of the hotels. The MapQL statement for this query is listed below:

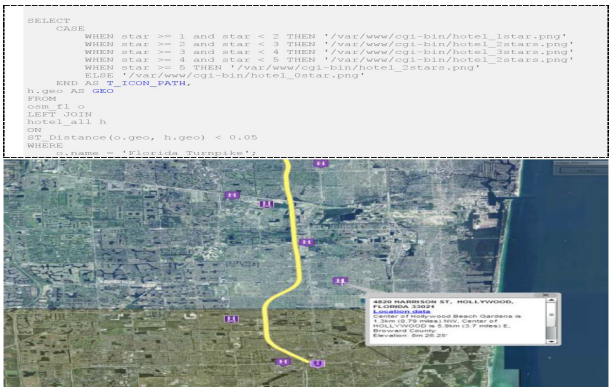


Fig. 15: Query hotel data along the line

Figure 16 shows the traffic of Santiago where the colder the color is, the faster the traffic is; the warmer the color is, the worse the traffic is. The MapQL statement is listed below:

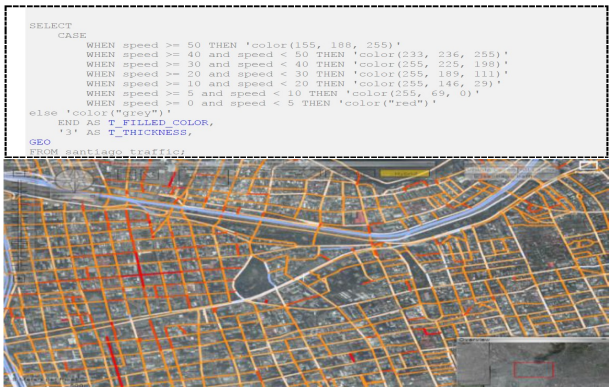


Fig. 16: Query traffic data of Santiago

Figure 17 shows the different average incomes with in different zip codes. In this demo, users can customize the color and style of the map layers, different colors stand for different average incomes. The corresponding MapQL statement is listed below:

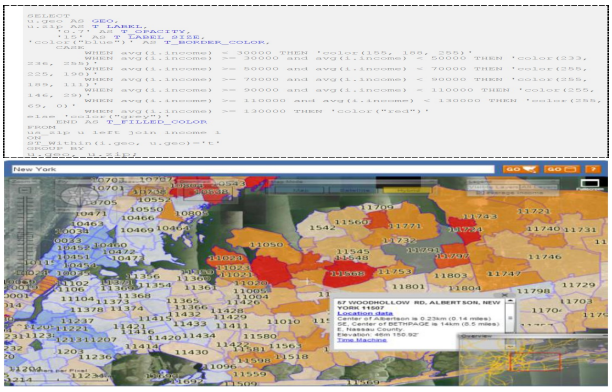


Fig. 17: Query average incomes

All these examples demonstrate that in TerraFly GeoCloud, users can easily create different map applications using simple SQL-like statements.

5. SYSTEM PERFORMANCE

In this section, we evaluate the performance of Terrafly GeoCloud using some example datasets and analysis. Terrafly GeoCloud supports both online analysis and offline analysis. For online analysis, we discuss the performance of correlation analysis; and for offline analysis, we use K-means analysis as an example.

We did not perform performance comparisons with similar products as they typically do not share much about their system design and implementation. Based on the fact that all GeoCloud functions have reasonable running time which facilitated users data analysis, we performed functionality comparisons with products whose functions are available (such as GeoDa and ArcGis [Anselin et al. 2006; Johnston et al. 2001]) in Related Products.

5.1. Online Analysis Performance

The data set used for performance evaluation is Florida_property_value which contains 1,042,281 records, and each record includes longitude, latitude and property value. Figure 18 shows part of the data set on the second zoom level. A yellow point, showing the property value, is used to denote each property.

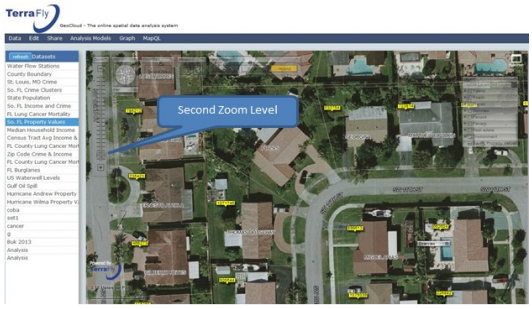


Fig. 18: South Florida Property

In order to provide good user experience, we only show part of the data that can be displayed on user's current screen. When a user zooms out the screen, GeoCloud will load the new data into

the screen and the analysis is then performed on the current displayed data. This guarantees that a user can view the data and obtain the analysis results very quickly. When users want to preform data analysis, most of the time they are more concerned with some local data. For example, if a user wants to buy a property in a certain zip code, and he/she will only care about the property values of his/her interested location. At this time, doing a global analysis is time consuming and unnecessary.

The online analysis performance is related to the zoom level. Here we use the auto-correlation analysis as an example to evaluate the online analysis performance. Figure 19 shows the performance of autocorrelation. The horizontal axis indicates the number of records on each zoom level. The vertical axis denotes the running time. For example, when the user zooms to the third level, there are 52 records showing on the screen, and the autocorrelation analysis needs 0.956s (which includes network communication time, time for analysis, and time for rendering the results on the map) to complete. The time needed for the sixth zoom level is 4 seconds. The sixth zoom level, which contains 1535 data records, is the highest level that all the data can be shown without overlapping. When we zoom to a higher level, too many records are overlapping with each other that makes the results hard to view.

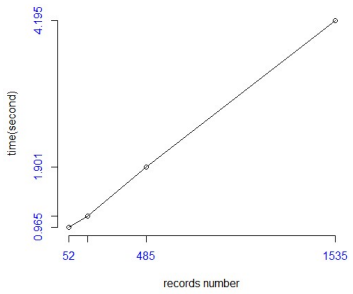


Fig. 19: Auto-Correlation performance

5.2. Offline Analysis Performance

Here we use the K-means clustering method to evaluate offline analysis performance in TerraFly GeoCloud. We apply K-means clustering analysis on Florida_property_value data set. In order to compare the performance of signal machine and hadoop cluster, we duplicate 10 times of the data set, the total number of the records is $19,616,320$.

For the experiment, we set the number of clusters to be 100 and iteration time is 4. The running time for signal machine is 34.83 minutes. Figure 20 shows the running time of hadoop. The vertical axis denotes the running time. The horizontal axis denotes the total task capacity that is the number of cores running parallel, which refer to the total computation power we assigned to the task. When we set total task capacity to 16, the running time of K-means is 7 minutes, so when user wants to perform big data analysis, using Hadoop is more efficient than single machine: when we adding the total task capacity, the performance is increasing, so the running time is decreasing dramatically. Leveraged by the Hadoop platform, we can guarantee the analysis performance by simply adjust total task capacity (computing power).

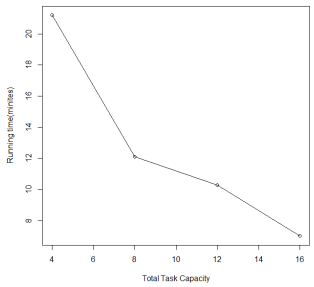


Fig. 20: Parallel K-means performance

6. CASE STUDIES

In this section, we present some case studies on using TerraFly GeoCloud for spatial data analysis and visualization. We use two types of data set, one is Florida property data, the other is Florida Lung cancer mortality to show how to apply Geocloud analysis and visualization function on application domains.

6.1. Florida Property Analysis

As discussed in Section 3.2, we know the results of auto correlation can be shown in a scatter diagram, where the first and third quadrants of the plot represent positive associations, while the second and fourth quadrants represent negative associations. The second quadrant stands for low-high which means the value of the object is low and the values of surrounding objects are high.

A lay user Erik, who has some knowledge about the database and data analysis, wanted to invest a house property in Miami with a good appreciation potential. By using TerraFly GeoCloud, he may obtain some ideas about where to buy. He believes that if a property itself has low price and the surrounding properties have higher values, then the property may have good appreciation potential, and is a good choice for investment. He wants to first identify such properties and then do a field trip with his friends and the realtor agent.

To perform the task, first, Erik checked the average property prices by zip code in Miami which is shown in Figure 7b. He found the green circled area in the low-high quadrants, which means that the average price of properties of this area is lower than the surrounding areas.

So. FL Property Values

id	min_show_level	longitude	latitude	pvalue
9766	1	-80.275203	26.273378	298000
12717	8	-80.142601	26.165103	403000
38997	8	-80.15843	26.340396	381000
44849	8	-80.416315	26.117033	486000
57613	8	-80.42536	26.009357	218000
62752	1	-80.208778	26.184111	149000
73930	8	-80.247377	25.785131	203000
84664	8	-80.11329	26.133841	66000
103612	8	-80.185366	26.101447	189000
106825	8	-80.234288	26.141735	172000
111149	8	-80.31925	26.106385	268000
113091	8	-80.129359	25.835633	1490000

Fig. 21: Sample Data of south_florida_house_price data set

Erik wanted to obtain more insights on the property price in this area. He uploaded a detailed spatial data set named as south_florida_house_price into the TerraFly GeoCloud system.

south_florida_house_price data set contains more than 1 million records and it shows the Geo-location information(coordinates) and price of the property in south Florida. The sample of the data set is shown in Figure 21. He customized the label color range as the properties price changes. And then, he chose different areas in the green circled area in Figure 7b to perform the auto-correlation analysis.

Fig. 22: Properties in Miami

Finally, he found an area shown in Figure 22, where there are some good properties in the low-high quadrants (in yellow circles) with good locations. And one interesting observation is, lots of properties along the road Gratigny Pkwy has lower prices. He was then very excited and wanted to do a query to find all the cheap properties with good appreciation potential along the Gratigny Pkwy. Erik composed the MapQL statements to find out the properties whose distance from the Gratigny Pkwy is less than a threshold and price is lower than the surrounding area, and if the value of the property is between 100,000 to 200,000, using green to denote the property, and if the value between 200,000 and 400,000, using blue to denote the property, and if the value is more than 400,000, using red color to indicate the house.

Fig. 23: MapQL results

The Figure 23 presents the final results of the MapQL statements. Finally, Erik sent the URL of the map visualization out by email, waiting for the response of his friends and the realtor agent.

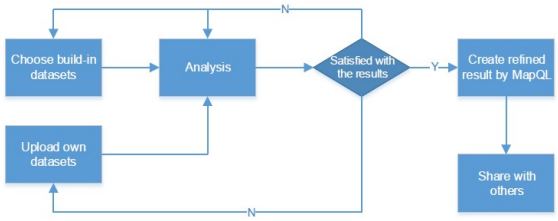


Fig. 24: The flow path of Erik case

Figure 24 illustrates the whole workflow of the case study. In summary, Erik first viewed the system build-in datasets, conducted the data analysis, and then he identified properties of interest. He then composed MapQL statements to create his own map visualization to share with his friends. The case study demonstrates that TerraFly GeoCloud supports the integration of spatial data analysis and visualization and also offers user-friendly mechanisms for customized map visualization.

6.2. Florida Lung Cancer Analysis

In this section we provide an example of how our GeoCloud system can be employed in epidemiologic research. Assume a researcher studies lung cancer in Florida. She can upload and choose the mor_price_income dataset to TerraFly GeoCloud - shown in Figure 25. mor_price_income dataset contains median house price, median income, lung cancer mortality, geometry information and name of each county in Florida.

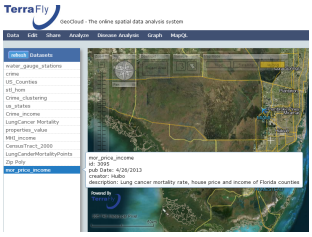


Fig. 25: Datasets in TerraFly GeoCloud

She can then choose the disease analysis button to draw a disease map. In this function, she can choose a legend group number; a disease map is displayed then, as shown in Figure 26.

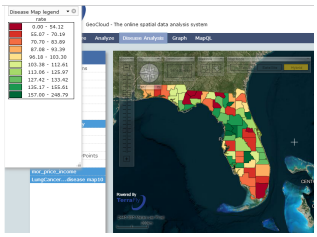


Fig. 26: Lung Cancer disease map

From Figure 26 we observe how this map, with legend at the top left corner, provides a direct summary of the disease data. For lung cancer in Florida, the mortality in the central region is higher and it is lower in the south region. However, the researcher cannot have an accurate analysis result just from this one map. She can further choose the cluster and outlier detection function, which uses Local Morans I to perform further analysis. This analysis function provides three maps: local Morans I map, z-value map, and p-value map. Figure 27 shows the p-value map, from which the researcher can know which counties form a statistically significant cluster and which counties are statistically significant outliers.

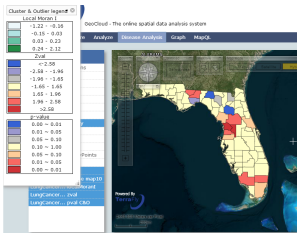


Fig. 27: P-value map of Local Moran I

Now the researcher may want to know what kind of relationship exists between lung cancer mortality and the median income of each county. For this purpose, she can use the median income dataset provided by the TerraFly GeoCloud system, and apply the spatial auto-regression tool. Figure 28 shows the result of this model. From the result, we can observe that when the mortality of surrounding areas increase by 1, the mortality of this county will increase by 0.233, and when the median income in the surrounding area increases by $\$10,000$, the mortality of this county will decrease by 0.09.

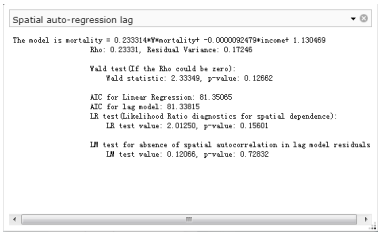


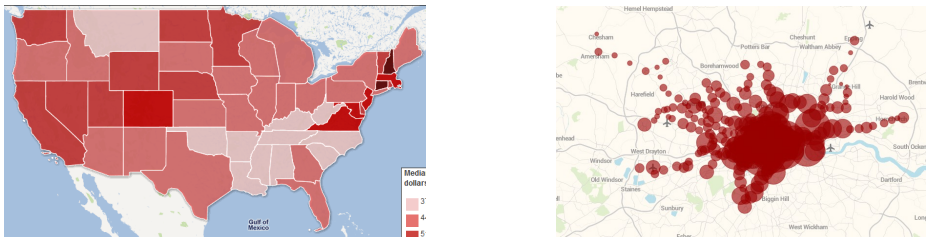
Fig. 28: Spatial auto-regression of lung cancer mortality and median income

7. RELATED WORK AND PRODUCTS

7.1. Spatial Data Visualization

Information visualization (or data visualization) techniques are able to present the data and patterns in a visual form that is intuitive and easily comprehendible, allow users to derive insights from the data, and support user interactions [Zhang and Li 2012; Spence and Press 2000; Li et al. 2010b]. For example, Figure 29a shows the map of Native American population statistics which has the geographic spatial dimensions and several data dimensions. The figure displays both the total population and the population density on a map, and users can easily gain some insights on the data by a glance [Old 2002]. In addition,visualizing spatial data can also help end users interpret and understand spatial data mining results. They can get a better understanding on the discovered patterns.k

Visualizing the objects in geo-spatial data is as important as the data itself. The visualization task becomes more challenging as both the data dimensionality and richness in the object representation increase. In TerraFly GeoCloud, we have devoted lots of effort to address the visualization challenge including the visualization of multi-dimensional data and the flexible user interaction. For spatial data mining to be effective, it is important to include the visualization techniques in the mining process and to generate the discovered patterns for a more comprehensive visual view [Zhang and Li 2012; Rishe et al. 2004].



(a) America Median Income (b) Customized map

Fig. 29: Related work

7.2. Spatial Analysis

Spatial analysis is especially used on geographic data. The difference between spatial analysis and traditional analysis is that spatial analysis methods use spatial information of the data, such as the location, orientation, and adjacent areas. Spatial analysis is widely used in many domains including biology, ecology, epidemiology, ecology, and criminology. There are many kinds of spatial analysis methods which include spatial clustering, spatial autocorrelation, spatial regression, spatial interpolation and spatial distribution measurement [Fotheringham and Rogerson 2013]. TerraFly GeoCloud presents comprehensive spatial analysis methods and result visualization in a more interactive way. User can leverage these methods without programming, and obtain the result visualized on the map with just a few clicks [Bailey et al. 1994].

7.3. Customized Map Visualization

The process of rendering a map generally means taking raw geospatial data and making a visual map from it. Often it applies more specifically to the production of a raster image, or a set of raster tiles, but it can refer to the production of map outputs in vector-based formats. ”3D rendering” is also possible when taking the map data as an input. The ability of rendering maps in new and interesting styles, or highlighting features of special interest, is one of the most exciting aspects in spatial data analysis and visualization.

Customized map visualization have several challenges. First, it takes time to generate a map. User needs to use complicated programs to generate maps from traditional map visualization software tools. Second, it is hard to obtain a really customized map. Some map services can provide some customized views for users. For example, Figure 29b shows a customized map where the adjacent data objects are merged together and are represented using big circles. However, it can not allow users to manipulate the data as there are only few visualization styles are provided.

TerraFly map render engine is a toolkit for rendering maps and is used to render the main map layers. It supports a variety of geospatial data formats and provides flexible styling options for designing many different kinds of maps, and the render speed is fast [Teng et al. 2006; Lu et al. 2014]. TerraFly Geocloud also provides MapQL as a spatial query and map render tool. User can query and visualize the data use a SQL-like statements. Because Geocloud is a web-based online service, user can use MapQL online and get a result in the map directly. This SQL-like statements facilitate users and let them draw the map in their own ways [Lu et al. 2013a].

7.4. Related Products

In the geospatial discipline, web-based GIS services can significantly reduce the data volume and required computing resources at the end-user side [Li et al. 2010a; Fotheringham and Rogerson 2013]. To the best of our knowledge, TerraFly GeoCloud is one of the first systems to study the integration of online visualization of spatial data, data analysis modules and visualization customization language.

Various GIS analysis tools are developed and visualization customization languages have been studied in the literature. ArcGIS is a complete, cloud-based, collaborative content management system for working with geographic information. But systems like ArcGIS and Geoda focus on the content management and share, not online analysis [Johnston et al. 2001; Anselin et al. 2006]. Azavea has many functions such as optimal Location find, Crime analysis, data aggregation and visualization. It is good at visualization, but has very limited analysis functions [Boyer et al. 2011].

Various types of solutions have been studied in the literature to address the problem of visualization of spatial analysis. However, on one hand, good analysis visualization tools like Geoda and ArcGIS do not have online functions. To use them, users have to download and install the software tools, and download the datasets. On the other hand, good online GIS systems like Azavea, SKE, and GISCloud have limited analysis functions. Furthermore, none of above products provides a simple and convenient way like MapQL to let user create their own map visualization [Hearnshaw et al. 1994; Boyer 2010]. The related products are summarized in Table I. Our work is complementary to the existing works and our system also integrates data mining and visualization.

Table I: GIS Analysis & Visualization Products

Name	Website	Product features description	Online tool	Spatial analysis abilities	Spatial visualization abilities
ArcGIS	http://www.esri.com/ software/arcgis/ arcgis-for-desktop	This software provides map creating and multiple analysis functions. But need training.	No	Multiple analysis functions are provided.	Good visualization. But map creating is complicated and need training.
Geoda	http://geodacenter.asu.edu/	User can import map, add layer to do some geodata analysis.	No	Multiple analysis functions, such as statistic map and rate map.	Limited visualization.
ArcGIS Online	http://www.arcgis.com	ArcGIS Online is a complete, cloud-based, collaborative content management system for working with geographic information.	Yes	No online Analysis.	Focus on the content management and share.
Azavea	http://www.azavea.com/ products/	optimal Location find, Crime analsis, data aggregated and visualized	Yes	Very limited analysis functions	Good visualization.
SKE	http://www.skeinc.com/ GeoPortal.html	Spatial data Viewer	Yes	Very limited simple analysis.	Focus on the spatial data viewer.
GISCloud	http://www.giscloud.com	with few analysis (Buffer , Range , Area , Comparison , Hotspot , Coverage , Spatial Selection)	Yes	No spatial analysis function	Focus on geo-data management and share.
GeoIQ	http://www.geoiq.com/ http://geocommons.com/	filtering, buffers, spatial aggregation and predictive	Yes	Very limited and simple analysis: currently provide predictive (Pearsons Correlation).	Focus on GIS, very good visualization and interactive operation.
GeoCloud	http://terrafly.fiu.edu/GeoCloud/	Provide spatial data visualization, spatial dependency and auto-correlation, spatial data clustering, spatial regression, measuring geographic distribution, spatial interpolation and customize map visualization	Yes	Provides multiple spatial analysis function. Easy to use.	Provide good data visualization and interactive operation. Easy to use.

8. CONCLUSION

This paper presents TerraFly GeoCloud, an online spatial data analysis and visualization system, to facilitate end users to visualize and analyze spatial data, and to share the analysis results. TerraFly GeoCloud focuses on building a new intelligent system that allows a general user perform spatial data analysis in a very simple and convenient way. By leveraging distributed computing, visualization and data mining techniques, TerraFly GeoCloud enables users to perform different types of spatial data analysis tasks for decision support. The system is a GIS analysis tool providing software as a service (SaaS). Comparing with traditional desktop software tools, Terrafly GeoCloud is

based on the cloud architecture and users can upload, visualize, analyze, and share the data through browsers with a few clicks. As the application of cloud service is getting widely used, this type of intelligent systems will be more and more popular in the future. About the future works, we will provide better visualization techniques to improve user experience. As user visits increasing, we will add load balance function in the front end through some popular technologies such as Nejx.

sed in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, FRP IIP-1230661, SBIR IIP-1058428, SBIR IIP-1026265, SBIR IIP-1058606, SBIR IIP-1127251, SBIR IIP-1127412, SBIR IIP-1118610, SBIR IIP-1230265, SBIR IIP-1256641. Includes material licensed by TerraFly (http://terrafly.com) and the NSF CAKE Center (http://cake.fiu.edu).

REFERENCES

Luc Anselin. 1995. Local indicators of spatial associationLISA. *Geographical analysis* 27, 2 (1995), 93–115.

Luc Anselin, Ibnu Syabri, and Youngihn Kho. 2006. GeoDa: an introduction to spatial data analysis. *Geographical analysis* 38, 1 (2006), 5–22.

Peter Armitage, Geoffrey Berry, and John Nigel Scott Matthews. 2008. *Statistical methods in medical research*. John Wiley & Sons.

Trevor C Bailey, S Fotheringham, and P Rogerson. 1994. A review of statistical spatial analysis in geographical information systems. *Spatial analysis and GIS* (1994), 13–44.

Michel Bilodeau, Fernand Meyer, Michel Schmitt, and Georges Matheron. 2005. *Space, Structure and Randommess: Contributions in Honor of Georges Matheron in the Field of Geostatistics, Random Sets and Mathematical Morphology*. Springer.

Deborah Boyer. 2010. From internet to iPhone: providing mobile geographic access to Philadelphia's historic photographs and other special collections. *The Reference Librarian* 52, 1-2 (2010), 47–56.

Deborah Boyer, Robert Cheetham, and Mary L Johnson. 2011. Using GIS to Manage Philadelphia's Archival Photographs. *American Archivist* 74, 2 (2011), 652–663.

HJ De Knegt, F Van Langevelde, MB Coughenour, AK Skidmore, WF De Boer, IMA Heitkonig, NM Knox, R Slotow, C Van der Waal, and HHT Prins. 2010. Spatial autocorrelation and the scaling of species-environment relationships. *Ecology* 91, 8 (2010), 2455–2465.

Robin Dubin, R Kelley Pace, and Thomas G Thibodeau. 1999. Spatial autoregression techniques for real estate data. *Journal of Real Estate Literature* 7, 1 (1999), 79–96.

Paul Elliott and Daniel Wartenberg. 2004. Spatial epidemiology: current approaches and future challenges. *Environmental health perspectives* (2004), 998–1006.

Martin Ester, Hans-Peter Kriegel, J Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *KDD*, Vol. 96. 226–231.

Stewart Fotheringham and Peter Rogerson. 2013. *Spatial analysis and GIS*. CRC Press.

Arthur Getis and J Keith Ord. 1992. The analysis of spatial association by use of distance statistics. *Geographical analysis* 24, 3 (1992), 189–206.

Hilary M Hearnshaw, David John Unwin, and others. 1994. *Visualization in geographical information systems.* John Wiley & Sons Ltd.

Kevin Johnston, Jay M Ver Hoef, Konstantin Krivoruchko, and Neil Lucas. 2001. *Using ArcGIS geostatistical analyst*. Vol. 380. Esri Redlands.

Harry H Kelejian and Ingmar R Prucha. 1998. A generalized spatial two-stage least squares procedure for estimating a spatial autoregressive model with autoregressive disturbances. *The Journal of Real Estate Finance and Economics* 17, 1 (1998), 99–121.

Martin Kulldorff. 1997. A spatial scan statistic. *Communications in Statistics-Theory and methods* 26, 6 (1997), 1481–1496.

Martin Kulldorff and Neville Nagarwalla. 1995. Spatial disease clusters: detection and inference. *Statistics in medicine* 14, 8 (1995), 799–810.

Alvin CK Lai, Tracy L Thatcher, and William W Nazaroff. 2000. Inhalation transfer factors for air pollution health risk assessment. *Journal of the Air & Waste Management Association* 50, 9 (2000), 1688–1699.

Hongfei Li, Catherine A Calder, and Noel Cressie. 2007. Beyond Moran's I: testing for spatial dependence based on the spatial autoregressive model. *Geographical Analysis* 39, 4 (2007), 357–375.

Lei Li, Dingding Wang, Chao Shen, and Tao Li. 2010b. Ontology-enriched multi-document summarization in disaster management. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 819–820.

Xiaoyan Li, Liping Di, Weiguo Han, Peisheng Zhao, and Upendra Dadi. 2010a. Sharing geoscience algorithms in a Web service-oriented environment (GRASS GIS example). *Computers & Geosciences* 36, 8 (2010), 1060–1068.

Yun Lu. 2013. Geospatial Data Indexing Analysis and Visualization via Web Services with Autonomic Resource Management. (2013).

Yun Lu, Mingjin Zhang, Tao Li, Yudong Guang, and Naphtali Rishe. 2013a. Online spatial data analysis and visualization system. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*. ACM, 71–78.

Yun Lu, Mingjin Zhang, Shonda Witherspoon, Yelena Yesha, Yaacov Yesha, and Naphtali Rishe. 2013b. SksOpen: Efficient Indexing, Querying, and Visualization of Geo-spatial Big Data. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, Vol. 2. IEEE, 495–500.

Yun Lu, Ming Zhao, Lixi Wang, and Naphtali Rishe. 2014. v-TerraFly: large scale distributed spatial data visualization with autonomic resource management. *Journal Of Big Data* 1, 1 (2014), 4.

Nathan Mantel. 1967. The detection of disease clustering and a generalized regression approach. *Cancer research* 27, 2 Part 1 (1967), 209–220.

Nikolaos Matsatsinis and Yannis Siskos. 2003. *Intelligent support systems for marketing decisions*. Vol. 54. Springer.

Patrick AP Moran. 1950. Notes on continuous stochastic phenomena. *Biometrika* 37, 1-2 (1950), 17–23.

L John Old. 2002. Information Cartography: Using GIS for visualizing non-spatial data. In *Proceedings, ESRI International Users' Conference, San Diego, CA*.

Stan Openshaw, Martin Charlton, Colin Wymer, and Alan Craft. 1987. A mark 1 geographical analysis machine for the automated analysis of point data sets. *International Journal of Geographical Information System* 1, 4 (1987), 335–358.

Naphtali Rishe, Shu-Ching Chen, Nagarajan Prabakar, Mark Allen Weiss, Wei Sun, Andriy Selivonenko, and D Davis-Chu. 2001. TERRAFLY: A High-Performance Web-based Digital Library System for Spatial Data Access.. In *ICDE Demo Sessions*. 17–19.

N Rishe, M Gutierrez, A Selivonenko, and S Graham. 2005. TerraFly: A tool for visualizing and dispensing geospatial data. *Imaging Notes* 20, 2 (2005), 22–23.

Naphtali Rishe, Yanli Sun, Maxim Chekmasov, Andriy Selivonenko, and Scott Graham. 2004. System architecture for 3D terrafly online GIS. In *Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium on*. IEEE, 273–276.

Robert Spence and A Press. 2000. Information visualization. (2000).

Michael L Stein. 1999. *Interpolation of spatial data: some theory for kriging*. Springer.

William Teng, Naphtali Rishe, and Hualan Rui. 2006. Enhancing access and use of NASA satellite data via TerraFly. In *Proceedings of the ASPRS 2006 Annual Conference*.

Jon Wakefield and Paul Elliott. 1999. Issues in the statistical analysis of small area health data. *Statistics in medicine* 18, 17-18 (1999), 2377–2399.

Huan Wang. 2011. A large-scale dynamic vector and raster data visualization geographic information system based on parallel map tiling. (2011).

Huibo Wang, Yun Lu, Yudong Guang, Erik Edrosa, Mingjin Zhang, Raul Camarca, Yelena Yesha, Tajana Lucic, and Naphtali Rishe. 2013. Epidemiological Data Analysis in TerraFly Geo-Spatial Cloud. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, Vol. 2. IEEE, 485–490.

Yi Zhang and Tao Li. 2012. DClusterE: A Framework for Evaluating and Understanding Document Clustering Using Visualization. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 2 (2012), 24.

Weizhong Zhao, Huifang Ma, and Qing He. 2009. Parallel k-means clustering based on mapreduce. In *Cloud Computing*. Springer, 674–679.

Sagit Zolotov, Dafna Ben Yosef, Naphtali D Rishe, Yelena Yesha, and Eddy Karnieli. 2011. Metabolic profiling in personalized medicine: bridging the gap between knowledge and clinical practice in Type 2 diabetes. *Personalized Medicine* 8, 4 (2011), 445–456.