TerraFly GeoCloud: Online Spatial Data Analysis System

Yun Lu, Mingjin Zhang, Tao Li, Erik Edrosa, Chang Liu, Naphtali Rishe
School of Computing and Information Sciences
Florida International University, Miami, Florida, 33199, USA
{yun,zhangm,taoli,eedro001,chang,rishen}@cs.fiu.edu

ABSTRACT
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 System, TerraFly GeoCloud, which facilitates the end user 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 TerraFly map supporting many different visualization functions and spatial data analysis models. TerraFly GeoCloud also enables the MapQL technology to create maps using SQL-like statements. The system is available at http://131.94.133.223/.

1. INTRODUCTION
TerraFly GeoCloud is built upon the TerraFly system to support various kinds of online spatial data analysis using TerraFly Maps API and JavaScript TerraFly API add-ons in a high performance cloud Environment. We first introduce the TerraFly system and then present the details on TerraFly GeoCloud.

1.1 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[1].

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 application programming interface (API) for web sites, etc. TerraFly's server farm ingests geo-locates, cleanses, mosaics, and cross-references 40TB of base map data and user-specific data streams [1].

1.2 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. TerraFly GeoCloud can import and display various kinds of spatial data (data with geo-location information) on the TerraFly map, edit the data, perform spatial data analysis, 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 application supports MapQL, which is a technology to create maps using SQL-like statements. TerraFly GeoCloud is available at http://131.94.133.223/.

The analysis functions provided by TerraFly GeoCloud include spatial data visualization (visualizing the spatial data), spatial dependency and autocorrelation (checking for spatial

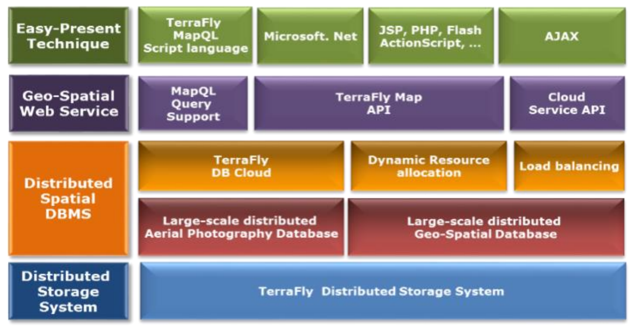

Figure 1: The Architecture of TerraFly GeoCloud

dependencies), spatial clustering (grouping similar spatial objects), and Kriging (geo-statistical estimator for unobserved locations). 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. They can then *visualize the data sets* with customized appearances. By *Manipulate 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 are also able to share them with others.

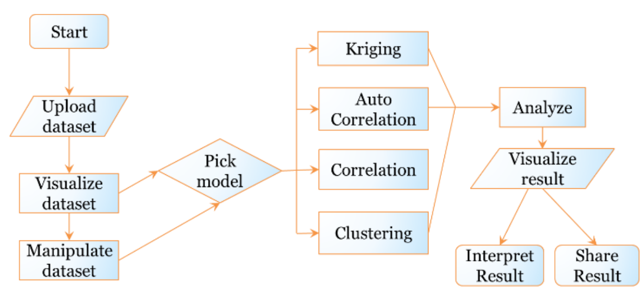

Figure 2: The Workflow of TerraFly GeoCloud

2. Spatial Data Analysis

2.1 Spatial Data Visualization
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 3, the point data is displayed on left, and the polygen data is displayed on the right.The data labels will be showed on the base map as extra layers for point data, and the data polygons will be showed on the base map for polygon data.

2.2 Spatial dependency and Auto-Correlation
Spatial dependency is the co-variation of properties within geographic space: characteristics at proximal locations appear to be correlated, either positively or negatively. Spatial dependency leads to the spatial autocorrelation problem in statistics [2].

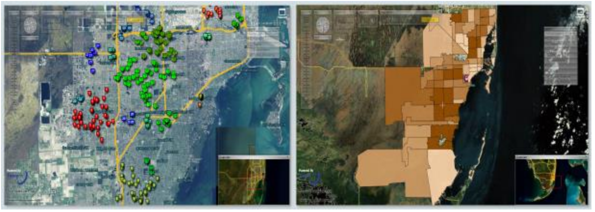

Figure 3: Spatial Data Visualization: Left subfigure: Point Data; Right subfigure: Polygon Data

Spatial autocorrelation is more complex than one-dimensional autocorrelation because spatial correlation is multi-dimensional (i.e. 2 or 3 dimensions of space) and multi-directional. The TerraFly GeoCloud system provides auto-correlation analysis tools to check for discovering spatial dependency in a geographic space, including global and local clusters analysis where Moran's I measure is used[3].Moran's I, the slope of the line, estimates the overall global degree of spatial autocorrelation as follows:

$$I = \frac{n}{\sum_i^n \sum_j^n w_{ij}} \times \frac{\sum_i^n \sum_j^n w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\sum_i^n (y_j - \bar{y})^2}$$

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 $\bar{y}$ are the variable in the ith location and the mean of the variable, respectively. n is the total number of observations. Moran's Iis used to test hypotheses concerning the correlation, ranging between –1.0 and +1.0.

Figure 4: Average properties price by zip code in Miami

Figure 4 shows an example of spatial auto-correlation analysis on the average properties price by zip code data in Miami (polygon data). The first and third quadrants of the plot represent positive association (high-high and low-low), while the second and fourth negative (high-low, low-high). The density of the quadrants represents the dominating local spatial process. The properties in Miami Beach are more expensive, and in the high-high area. Figure 5 shows auto-correlation analysis on the individual properties price in Miami (point data). As the figure shows, the properties near the highway are cheaper, while the properties along the lake are more expensive.

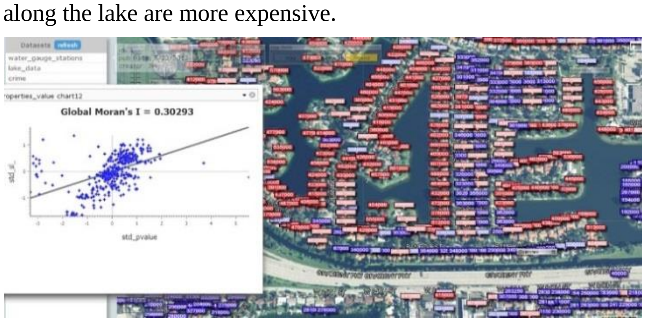

Figure 5: Properties value in Miami

2.3 Spatial Data Clustering

The TerraFly GeoCloud system supports the DBSCAN data clustering algorithm [4]. Figure 6 shows an example of DBSCAN clustering on the crime data in Miami. As shown in Figure 6, 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.

Figure 6: DBSCAN clustering on the crime data in Miami

2.4 Kriging

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) [5]. Figure 7 shows an example of Kriging. The data set is the water level from water stations 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.

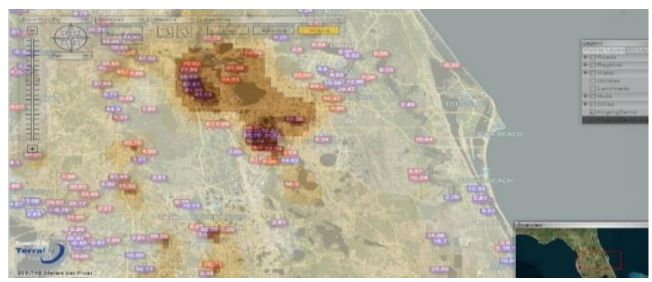

Figure 7: Kriging data of the water level in Florida

3. MapQL Spatial Query and Render tools

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. By using MapQL tools, users can easily create their own maps.

Figure 8 shows all the open-house within a certain distance of FIU, and the MapQL statement for this query is listed below. Please be noticed that the unit of the distance function in all the demos is Lat-Long.

```
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;
```

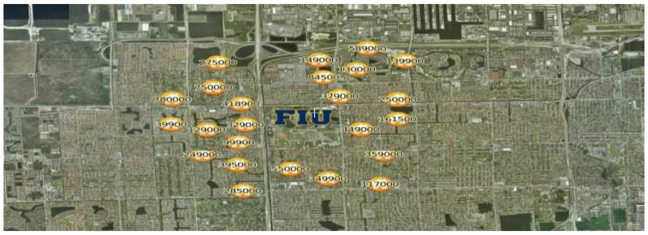

Figure 8: Query data near the point

Figure 9 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:

```
SELECT
    CASE
        WHEN star >= 1 and star < 2 THEN '/var/www/cgi-bin/hotel_1star.png'
        WHEN star >= 2 and star < 3 THEN '/var/www/cgi-bin/hotel_2stars.png'
        WHEN star >= 3 and star < 4 THEN '/var/www/cgi-bin/hotel_3stars.png'
        WHEN star >= 4 and star < 5 THEN '/var/www/cgi-bin/hotel_2stars.png'
        WHEN star >= 5 THEN '/var/www/cgi-bin/hotel_2stars.png'
        ELSE '/var/www/cgi-bin/hotel_0star.png'
    END AS T_ICON_PATH,
h.geo AS GEO
FROM
osm_fl o
LEFT JOIN
hotel_all h
ON
ST_Distance(o.geo, h.geo) < 0.05
WHERE
    o.name = 'Florida Turnpike';
```

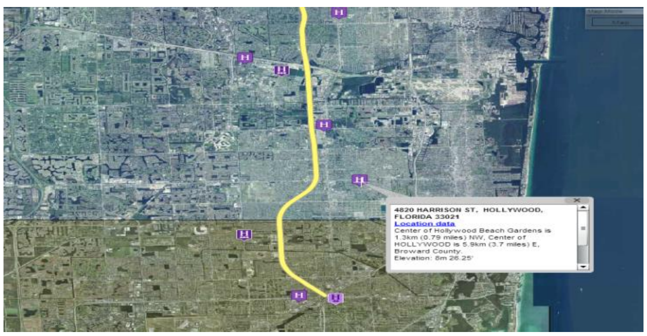

Figure 9: Query data along the line

Figure 10 shows the traffic of Santiago where the colder the color is, the faster the traffic is, the warmer the color is, and the worse the traffic is. The MapQL statement is listed below:

```
SELECT
    CASE
        WHEN speed >= 50 THEN 'color(155, 188, 255)'
        WHEN speed >= 40 and speed < 50 THEN 'color(233, 236, 255)'
        WHEN speed >= 30 and speed < 40 THEN 'color(255, 225, 198)'
        WHEN speed >= 20 and speed < 30 THEN 'color(255, 189, 111)'
        WHEN speed >= 10 and speed < 20 THEN 'color(255, 146, 29)'
        WHEN speed >= 5 and speed < 10 THEN 'color(255, 69, 0)'
        WHEN speed >= 0 and speed < 5 THEN 'color("red")'
else 'color("grey")'
    END AS T_FILLED_COLOR,
    '3' AS T_THICKNESS,
GEO
FROM santiago_traffic;
```

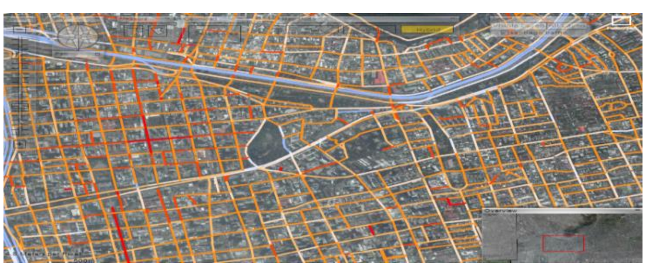

Figure 10: Traffic of Santiago

Figure 11 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 color stand for different average incomes. And the MapQL statement is listed below:

```
SELECT
u.geo AS GEO,
u.zip AS T_LABEL,
    '0.7' AS T_OPACITY,
    '15' AS T_LABEL_SIZE,
'color("blue")' AS T_BORDER_COLOR,
    CASE
        WHEN avg(i.income) < 30000 THEN 'color(155, 188, 255)'
        WHEN avg(i.income) >= 30000 and avg(i.income) < 50000 THEN 'color(233,
236, 255)'
        WHEN avg(i.income) >= 50000 and avg(i.income) < 70000 THEN 'color(255,
225, 198)'
        WHEN avg(i.income) >= 70000 and avg(i.income) < 90000 THEN 'color(255,
189, 111)'
        WHEN avg(i.income) >= 90000 and avg(i.income) < 110000 THEN 'color(255,
146, 29)'
        WHEN avg(i.income) >= 110000 and avg(i.income) < 130000 THEN 'color(255,
69, 0)'
        WHEN avg(i.income) >= 130000 THEN 'color("red")'
else 'color("grey")'
    END AS T_FILLED_COLOR
FROM
us_zip u left join income i
ON
ST_Within(i.geo, u.geo)='t'
GROUP BY
u.geo, u.zip;
```

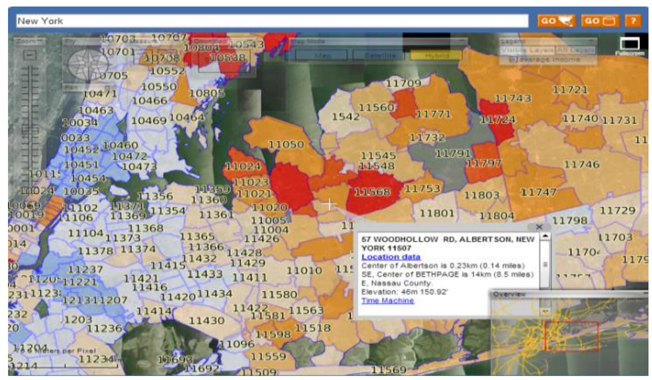

Figure 11: Income at New York

All these examples demonstrate that in TerraFly GeoCloud, users can easily create different map applications using simple SQL-like statements. More demo of MapQL result, please go to http://131.94.133.236/index.htm

4. REFERENCES

[1] Rishe, N., Chen, S. C., Prabakar, N., Weiss, M. A., Sun, W., Selivonenko, A., & Davis-Chu, D. (2001, April). TerraFly: A high-performance web-based digital library system for spatial data access. In The 17th IEEE International Conference on Data Engineering (ICDE), Heidelberg, Germany (pp. 17-19).

[2] De Knegt, H. J., Van Langevelde, F., Coughenour, M. B., Skidmore, A. K., De Boer, W. F., Heitk önig, I. M. A., ... &Prins, H. H. T. (2010). Spatial autocorrelation and the scaling of species-environment relationships. Ecology, 91(8), 2455-2465.

[3] Li,Hongfei; Calder, Catherine A, "Beyond Moran's I: Testing for Spatial Dependence Based on the Spatial Autoregressive Model". Geographical AnalysisCressie, Noel (2007).

[4] Ester, M., Kriegel, H. P., Sander, J., &Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. ACM SIGKDD.

[5] Stein, M. L. (1999). Interpolation of spatial data: some theory for kriging. Springer Verlag.