Erschienen in: Enterprise information systems : 12th International Conference, ICEIS 2010, Funchal-Madeira, Portugal, June 8 - 12, 2010; revised selected papers / Joaquim Filipe ... (eds.). - Heidelberg [u.a.] : Springer, 2011. - S. 495-511. - (Lecture Notes in Business Information Processing ; 73). - ISBN 978-3-642-19801-4

Developing Analytical GIS Applications with GEO-SPADE: Three Success Case Studies

Slava Kisilevich[1], Daniel Keim[1], Amit Lasry[2], Leon Bam[2], and Lior Rokach[3]

[1] Department of Computer and Information Science
University of Konstanz, Konstanz, Germany
[2] Department of Information System Engineering
Ben-Gurion University of the Negev, Beer-Sheva, Israel
[3] The Deutsche Telekom Laboratories, Department of Information System Engineering
Ben-Gurion University of the Negev, Beer-Sheva, Israel
{slaks,keim}@dbvis.inf.uni-konstanz.de,
{lasrya,bam}@bgu.ac.il, liorrk@bgu.ac.il

Abstract. Geographic Information Systems (GIS) handle a wide range of problems that involve spatial distribution data from fields as diverse as housing, healthcare, transportation, cartography, criminology, and many others. Selecting the right GIS is determined by the task the user wishes to perform. Some considerations may include the availability of analytical functions, performance, ease of use, interoperability, extensibility, etc. Nowadays, as positioning technology proliferates and affects almost all aspects of our lives, GIS tools have become more and more complex as researchers seek to address and implement, respectively, new spatial tasks and techniques. Due to the rapid growth of GIS in recent years, many provide APIs for extending their software for specific needs. But learning a complex set of API functions, is not feasible in scenarios that demand rapid prototyping like in academic research or when the tool is supposed to be used by non-professionals. In this paper, we present three case studies that are performed on top of a GEO-SPADE, a Google Earth-based .NET-based framework that we developed. Unlike the typical APIs, that may often contain more than hundred or thousand of functions from different logical layers, the GEO-SPADE framework provides only the minimal set of functions for plug-in development, networking, and visualization using Google Earth engine. The extended functionality is provided by the developer and depends on her knowledge of the underlying language. At the same time, the ease of use is achieved by the functionality of Google Earth. The three presented case studies showcase the applicability of the Google Earth-based framework in different domains with different analytical tasks (spatial analysis, spatial exploration, and spatial decision support).

1 Introduction

Geographic Information Systems (GIS) handle a wide range of problems that involve spatial distribution data such as housing, healthcare, transportation, cartography, criminology, and many others. Selecting the right GIS is driven by the task the user wishes to perform [1]. Underlying considerations may include the availability of analytical functions, performance, ease of use, interoperability, extensibility, etc. Nowadays, with the

proliferation of positioning technology and the rapid growth of spatio-temporal data together with a composite structure of non-spatial data, GIS tools have become more and more complex as researchers seek to address and implement new spatial analysis tasks and techniques. Although many contemporary GIS provide APIs for extending their software for specific needs, this involves adapting the GIS to learn a complex set of proprietary API functions, that may often contain more than hundred or thousand of functions from different logical layers, which is not feasible, particularly in scenarios that demand rapid prototyping like in academic research or in resource-constrained projects or when the tool is supposed to be used by non-professionals. Usually, the analyst (or researcher) requires fast implementation of tasks and the decision to extend the GIS application in cases when some functionality is not available, is rarely favorable. Consequently, to achieve the desired results, the geo-processing task is split between multiple GIS applications where each of them partly provides the required functionality.

The issue of proprietary APIs and interoperability is addressed by Open Geospatial Consortium (OGC)[1], which encourages developers to use standard interfaces and specifications when implementing geo-processing. A need for distributed geo-processing systems was addressed in recent publications [2,3,4,5]. Such systems split the workload between client and server, where clients are often Web applications, accessible by multiple users while the server performs all the required computations and delivers the results to the client using Web service technologies. Such architectures have several advantages over old, closed solutions: (1) there is no need to install a Web application; (2) many users can access the application using a Web browser; (3) the computation is performed on the server side, which relieves the user's computer from performing time-consuming operations; and (4) there is no need to locally store the data (the data can be accessed directly by the service and hosted by the service provider).

Scalable, distributed architecture, that conforms to open standards, may solve the problem of interoperability and service reusability. However, the client side still has to be developed according to specific tasks defined in advance. Since any changes in the design or in extending the capabilities should be performed by the body responsible for maintaining the Web application, there is a need for a platform, that supports easy extensibility with minimum knowledge of the underlying API. Such a platform should support many of the general visual data exploration approaches like *brushing, focusing, multiple views, linking*. It should also be able to support (geo)visualization approaches such as direct depiction, visualization of abstract data summaries, and extraction and visualization of computationally extracted patterns. It should also be capable of integrating different geo-related tasks to allow the analyst to quickly generate and test her hypotheses [6]. Of the platforms that combine ease of development and visualization, Google Earth has become popular among researchers dealing with spatial data.

Google Earth has attracted interest ever since it first appeared on the market [7]. Although debates whether Google Earth is a true GIS and what advantages it brings to scientists continue [8,9,10,11], the number of publications in which Google Earth is used for visualization and exploration of geo-related results shows the great interest among researchers in this tool. There are several factors that have made Google Earth

[1] http://www.opengeospatial.org

so successful among geo-browsers: it is freely available; its fast response and real-time exploration of spatial data; its use of satellite imagery; its ease of use and of Keyhole Markup Language (KML)[2], a human readable, XML-based format for geographic visualization. Other attractive factors include Google Earth's ability to implement basic visualization techniques like zooming, panning and tilting, and the availability of many layers of information provided by Google and by user communities. In addition, Google Earth includes a built-in HTML browser to display textual information and allows dynamic content to be streamed from a server in response to changes in a visible frame.

Recent publications demonstrated that Google Earth can be effectively used for data exploration in various areas such as: weather monitoring [12], spatio-temporal data exploration [13], data mining [14], insurance [15], crisis management [16], spatial OLAP [17], and geospatial health [18]. Despite its capabilities, Google Earth is still used as a "secondary" tool for data visualization because of its lack of geo-analytical functionality. Developers who want to integrate Google Earth in their applications usually overcome this geo-analytical deficiency by embedding Google Earth into a Web application using its JavaScript API. "Missing" features can then be implemented using a *Network Link* element, which is part of KML, to point to the URL of the underlying server to request a specific operation. Although the *Network Link* feature is used for creating dynamic content, it allows data to flow in only one direction, from the server to Google Earth in response to the changes in the visual boundaries. Clearly, this feature does not allow more complex operations like passing parameters interactively, running the data mining algorithm on the server, and returning the results for a particular setup of objects being visualized by Google Earth. Other mechanisms for performing such operations are required.

Recently, it became possible to embed Google Earth in desktop applications[3] using the Google Earth Web browser plugin[4] COM interface. This makes it possible to write custom applications around Google Earth in high level languages like c# and to completely control the logic of the application thus bypassing the limitations of the standalone version of Google Earth. This possibility, on the one hand, and the constraints of current GIS systems, on the other, encouraged us to develop a prototype desktop system called GEO-SPADE (**GEO SPA**tiotemporal **D**ata **E**xploration) [19]. The architecture of GEO-SPADE, presented in [19], is based on a thin client paradigm and introduces pluggable components and Service-oriented Architecture (SOA). The framework acts as a bridge between the Google Earth engine and the user defined functionality implemented in a user provided code as a front-end loaded by the framework and the back-end that is completely separated from the client-side (GEO-SPADE) but communicated by means of Web services with the front-end. Unlike the typical APIs, that may often contain more than hundred or thousand of functions from different logical layers, the GEO-SPADE framework provides only the minimal set of functions for plug-in development, networking, and visualization using Google Earth engine. The minimalistic set of API functions and the text-based protocol (KML) for geographical visualization allows for

[2] http://code.google.com/apis/kml/documentation/kmlreference.html
[3] http://code.google.com/p/winforms-geplugin-control-library/
[4] http://earth.google.com/plugin/

rapid prototyping of different tasks, while the ease of use pertinent to Google Earth allows non-professionals to utilize GEO-SPADE.

To highlight its applicability for different analytical tasks, we selected three real-life case studies that were successfully performed by GEO-SPADE. The results of two of the presented case studies were presented at the international conferences. The case studies showcase the applicability of the Google Earth-based framework in various domains with different analytical tasks. The first case study shows how GEO-SPADE is used for analyzing tourist activity [20,21]. In the second case study, GEO-SPADE is implemented as a spatial exploration tool for searching for geo-tagged photos using different search criteria [22]. In the third case study, GEO-SPADE operates as a spatial decision support tool for predicting hotel prices.

2 Related Work

In this section we discuss the applicability of Google Earth in different disciplines and domains as a stand-alone or Web-based tool, and the extensibility of Google Earth functionality.

Google Earth has been used to show weather related information in [12] by producing snapshot updates of the current weather every 120 seconds and utilizing the KML file format to read these updates into the Google Earth. GeoSphereSearch, a context-aware geographic Web search, integrated a Web-based Google Earth to visualize results of a query [23]. [13] demonstrated how visually exploring mobile directory service log files with spatial, temporal and attribute components can be performed using Google Earth in combination with other open source technologies like MySQL, PHP and LandSerf. [13] discusses several aspects, like visual encodings, color and overplotting and shows how Google Earth handles these features. [15] presented the feasibility of exploring catastrophic events and potential loss of information. The tasks that must be performed are outlined. The methods, with which Google Earth can handle such tasks, include: mapping, filtering by attribute, space, time, spatial aggregation, and creation of new views. Bringing geo-processing capabilities and Google Earth together is addressed by [16] to support crisis management scenarios. For this task, "Google Earth Dashboard" was proposed. It combines several technologies: Adobe Flex, Web Map Service (WMS), Web Feature Service (WFS), Web Processing Service (WPS) services and Google Earth as the cornerstone. The potential in combining geo-browsers like Google Earth in analyzing geospatial health sciences is discussed in [18]. The applicability of Google Earth in Online Analytical Processing (OLAP) is demonstrated in [17,24].

A desktop-based exploratory spatial association mining tool integrated with Google Earth for visualization was proposed in [14]. The tool consisted of two visualization views: the first was based on Google Earth for viewing and exploring related phenomena; the second view was developed in-house using Java 3D technology to interact with the association rules produced by the mining algorithm.

Integrating WPS-based services into Google Earth is addressed by [5]. Their approach consists of two parts. In the first part, the WPS client, built on top of the uDig (desktop, Java-based GIS system) exports configured processes into a KML file. In the

second part, the KML file is loaded into Google Earth, which triggers WPS processes using the *Network Link* element. The advantage of the WPS approach is that the exported processes can be consumed by the vast community of people using applications that handle KML formats such as Google Earth or Google Maps. This advantage was addressed in recent publications [12,15]. However, the proposed approach has several disadvantages. Although the approach has only two parts, they are not autonomous and require manual implementation (configuration of the processes using WPS client, creation and export of KML and dissemination of the results). Another aspect is lack of interaction and control over the geo-process exported into KML. As soon as KML is loaded into the application with *Network Link*, the process will be triggered and the results returned.

In general, the client side, which usually involves data visualization and manipulation, still remains more heterogeneous than the server side, since the way the data is presented cannot be enforced by standard specifications and interfaces. Here the choice of an appropriate software is driven by the needs of an expert. As was already mentioned, dozens of GIS applications are available, capable of solving specific spatial processing tasks and of supporting geo-visualization. When these applications are incapable of dealing with the problem, custom solutions and architectures are proposed [25,14,26]. However, the current trend in data visualization and exploration is to use mass-market applications such as Google Earth for visualization, combined with open source technologies for data processing and storage [13,16,17,24].

3 Case Studies

In this section, we present three case studies in which GEO-SPADE is used as an analytical, exploratory and decision support tool. In the first case study presented in Section 3.1, GEO-SPADE is used as analytical tool for analyzing tourist activity using geo-tagged photos. The method was presented in [20] and used as an example of the tool's capabilities in [19]. We extend the case study by discussing the details of data interchange and technologies used in the analysis. The second case study (Section 3.2) discusses the integration of GEO-SPADE in a region exploration scenario using geo-tagged photos sorted according to various criteria. The third case study (Section 3.3) presents an on-going project in which GEO-SPADE acts as a decision support tool for predicting hotel prices.

Before moving to a specific case study, we would like to draw the reader's attention to Table 1. It shows the technologies and tools that were used in each case study. It can be seen that KML is shared between all case studies because it is a format for geographical visualization used by Google Earth. Java API for RESTful Web Services[5] is also used in the three case studies because Java is one of the most popular, general-purpose languages and the language of choice of many Web programmers. However, Windows Communication Foundation (WCF)[6] was used in the first case study along with Java to simplify complex data interchanges between the client-side (GEO-SPADE) written in c# and the server-side. Specifically, class serialization was used to transmit complex

[5] http://jcp.org/aboutJava/communityprocess/mrel/jsr311/index.html
[6] http://msdn.microsoft.com/en-us/netframework/aa663324.aspx

Table 1. Server-side technologies and tools

Technologies	Tourist activity Section 3.1	Region exploration Section 3.2	Hotels Section 3.3
Communication	Java API for RESTful Web Services Windows Communication Foundation	Java API for RESTful Web Services	Java API for RESTful Web Services
Database	PostGis	MySql	Microsoft SQL Server 2008
Data Interchange	KML Class Serialization	KML User-defined	KML
Additional tools	JfreeChart Teiresias [27] DBSCAN [29]		Weka 3.6 API [28] DBSCAN [29]

data structure, and rebuild the native object. Likewise, different databases were used. PostGis[7] was used in the first case study because of its support for spatial queries and free availability. MySql[8] was used in the second case study because of its ease of maintenance. We used Microsoft SQL Server 2008 in the third case study to simulate as close as possible the production environment of the service provider. Thus, the loosely coupled architecture (described in [19]) gives a developer freedom in choosing whatever technology to use to accomplish a specific task.

3.1 Analysis of Tourist Activity

The analysis of people's activity has become a major topic in different domains such as transportation management and tourism [30,31]. The access to a vast amount of geotagged photos taken by people from all over the world provided by photo-sharing sites like Flickr or Panoramio, has enabled the analysis of tourist activity at large scales, raised new research questions, and fostered the development of new analytical methods and applications. In our recent work [20,21], we defined a number of tasks for analyzing attractive areas, people's behavior and movement, and proposed a number of analytical methods.

We outlined six major steps for completing the task of finding travel sequences [20] in an arbitrary region of the world. These steps actively involve the analyst in the process of finding sequence patterns by reviewing the results of every step in the process, changing the parameters and activating the new steps using results from the previous step.

In the first step, presented in Figure 1, the analyst selects the desired area. The visible frame constitutes the boundaries of a region near Funchal, Madeira. The analyst can check the photographic activity in the selected region [21]. Figure 2 shows time-series graph of the number of people who took photos in the selected area from January 2008 to September 2009 aggregated by month. The server receives the boundary of the region and queries the database counting the number of people per month. We used JFreeChart[9] to generate the image of the graph. The image is stored in the temporal directory on the server, while the KML, which is sent to the client includes the URL to the graph. The example of the KML is presented in Listing 1.1.

[7] http://postgis.refractions.net/

[8] http://www.mysql.com/

[9] `http://www.jfree.org/jfreechart/`

```
<Placemark>
<name>Funchal (regional events)</name>
 <description><![CDATA[<table border="1">
  <tr><td>south bound: 32.6445708362354</td></tr>
  <tr><td>north bound: 32.6594099577288</td></tr>
  <tr><td>west bound: -16.922443384542</td></tr>
  <tr><td>east bound: -16.891952135182</td></tr>
  <tr><td>lower time bound: 2008-01-01 00:00:00</td></tr>
  <tr><td>upper time bound: 2009-08-31 00:00:00</td></tr>
  <tr><td>
  <img src="http://192.168.2.100/t1/monthlyEvents.png">
  </td></tr>
  </table>]]>
 </description>
</Placemark>
```

Listing 1.1. Server generated KML that shows photographic activity in the region. The activity graph is stored on the server.

The process of finding travel sequences is divided into two parts [20]. In the first part, every photo location is matched against a database of points of interest (we used the Wikipedia database[10]) and the closest POI is assigned to the photo. This creates clusters in which every photo is assigned to existing points of interest (POIs). In the second part, the remaining unassigned photos are clustered using DBSCAN [29], a density-based clustering algorithm. The results of this part are presented in Figure 3. The left part of the figure shows clusters in which photos were assigned to known POIs (assigned clusters in our terms), while the right part of the figure presents clusters of photos that were not assigned to a POI (unassigned clusters). Listing 1.2 shows part of the KML generated by the server that describes information about unassigned clusters including its symbolic name that begins with "[uc]", identifier number (id), information regarding the number of photos, the number of people in the cluster, and cluster boundaries.

Next, the analyst visually inspects the created clusters and performs the clustering again if needed (for example, when the size of clusters should be changed). She may remove some clusters that are irrelevant or unimportant using either the statistical information of the cluster (number of photos and people in a cluster) or her background knowledge of the area. Figure 4 summarizes this step by presenting a view in which the analyst can select unassigned clusters and artificially create a new POI identification for every selected cluster, either by giving a symbolic identifier or some meaningful name based on the knowledge of the area. Since Google Earth may contain different objects in a view, the system iterates over the objects and selects only those whose names begin with the "[uc]" identifier (see Listing 1.2). The identifiers of unassigned clusters that the analyst decides to add to the list of clusters that are important for the analysis, are sent to the server, which, in turn, queries the database, matches the received identifiers

[10] http://toolserver.org/~kolossos/wp-world/pg-dumps/

```
<Placemark>
 <name>[uc] unassigned cluster id: 23</name>
 <description>
  <![CDATA[<br>owners: 29<br>photos: 32]]>
 </description>
 <Polygon>
  <outerBoundaryIs>
   <LinearRing>
    <coordinates>
      -16.926498,32.639085,0  -16.928043,32.640748,0
      -16.925811,32.642844,0  -16.922807,32.643494,0
      -16.922003,32.642939,0  -16.920404,32.640675,0
      -16.921896,32.639835,0  -16.926498,32.639085,0
    </coordinates>
   </LinearRing>
  </outerBoundaryIs>
 </Polygon>
</Placemark>
```

Listing 1.2. Server generated KML that describes information about unassigned clusters

to the ids stored in the database, adds these clusters to the list of assigned clusters and assigns symbolic names.

In the final step, the analyst generates sequence patterns from the clusters that were assigned to existing and artificial POIs. Figure 5 shows a form in which the analyst can select such parameters as: database properties and the length of generated patterns. We used the Teiresias algorithm [27] for generating sequence patterns. Since the information about the generated sequences is not used directly for visualization, the server does not send the data in the KML format. Instead, we used a class serialization mechanism available in WCF to transfer the complex data structure to the client. The data includes the frequency of the sequence, the identifiers of the clusters in the sequence, the names of points of interest that constitute the travel sequence, the coordinates and centroids of every cluster in a sequence. The inspection of the travel sequences can be performed without referring to the exact location of clusters that are part of the sequence if the names of clusters belong to some existing POIs. However, when the sequence contains a cluster with a generated symbolic name, the analyst has to see the position of the cluster on the map. This is demonstrated by the following case.

The most frequent pattern generated is *Santa Lucia* $\Rightarrow$ *newpoi-3*. While Santa Lucia is a parish in the district of Funchal and may be known to the domain expert, *newpoi-3* is a name assigned by the system to an area in which the Wikipedia database does not contain a known POI. Clearly, this area should be located to give the analyst a hint about the sequence pattern. Sequences can be highlighted by clicking on the sequence pattern. Placemarks are added to the centers of the areas that are part of the selected sequence and the number assigned to them highlights the relative order of the area in

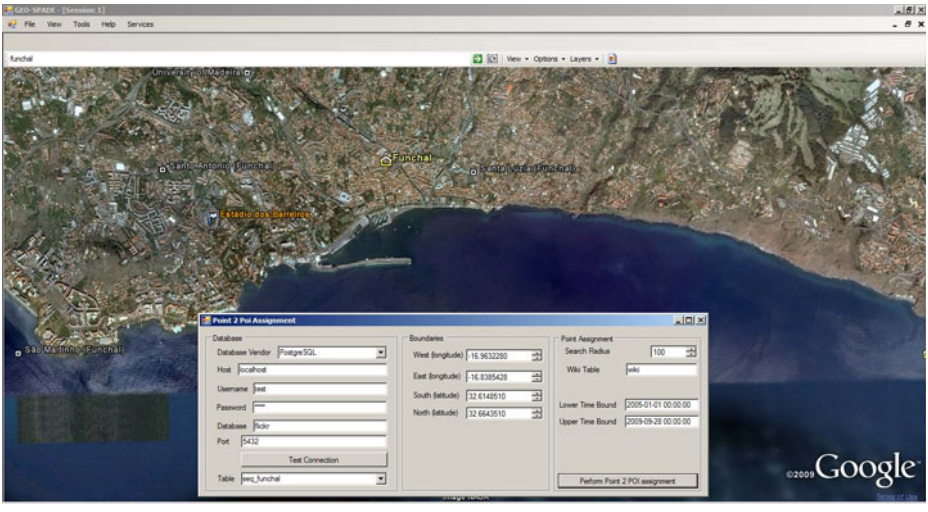

Fig. 1. Selection of the area of interest

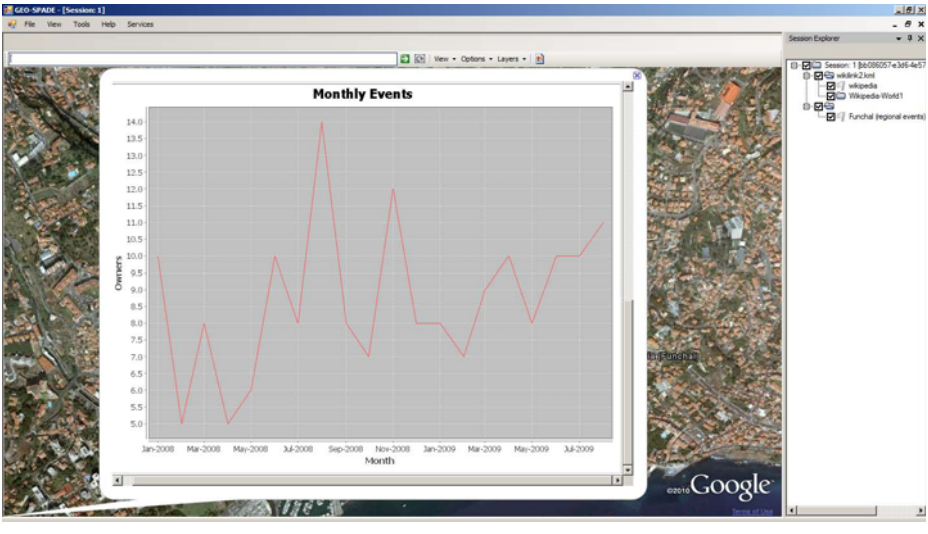

Fig. 2. Monthly photographic activity

the sequence. In the case of *Santa Lucia* ⇒ *newpoi-3*, Santa Lucia will by highlighted with the number one, while *newpoi-3* receives the number two (Figure 6).

3.2 Region Exploration Using Geo-tagged Photos

Photo-sharing web sites like Panoramio or Flickr provide users with the means to explore a specific area by searching for photos taken in that area using different filtering

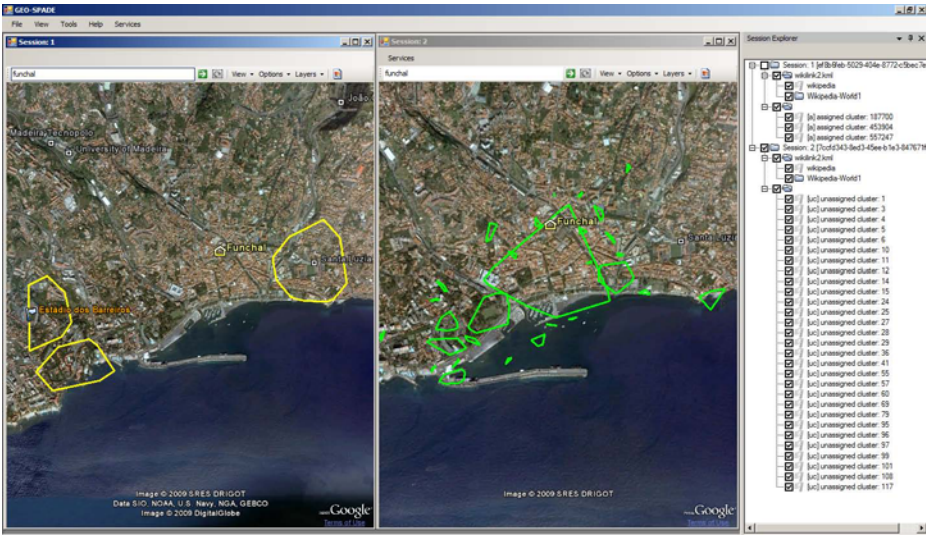

Fig. 3. Multiple views of regions assigned to some existing POI (left) and regions where no POI was found (right)

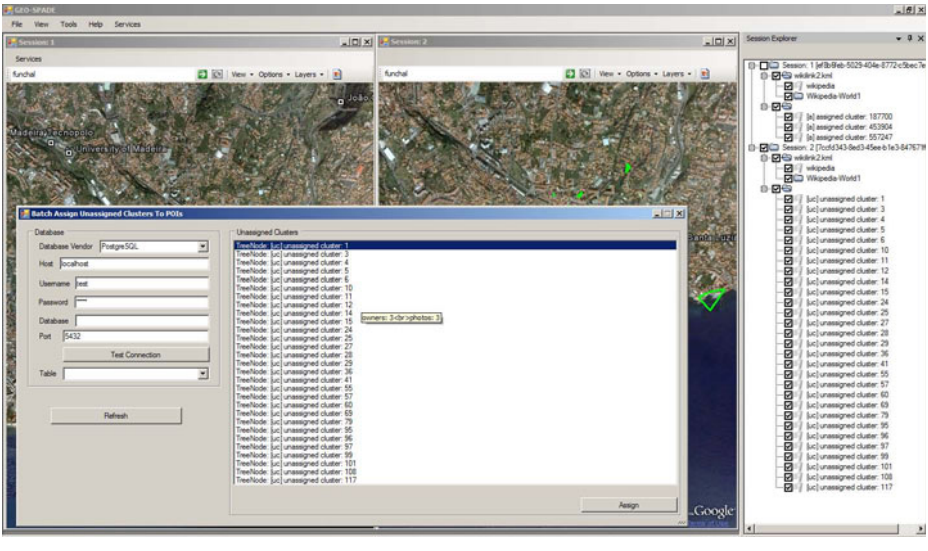

Fig. 4. Selection of unassigned regions and creation of artificial POIs

criteria: keywords, recently uploaded photos, interesting photos defined by the number of comments written or by the number of times a photo was viewed. In [22], we presented a method for analyzing photo comments in terms of user opinions and sentiments that are present in comments. When the text parts that describe opinions

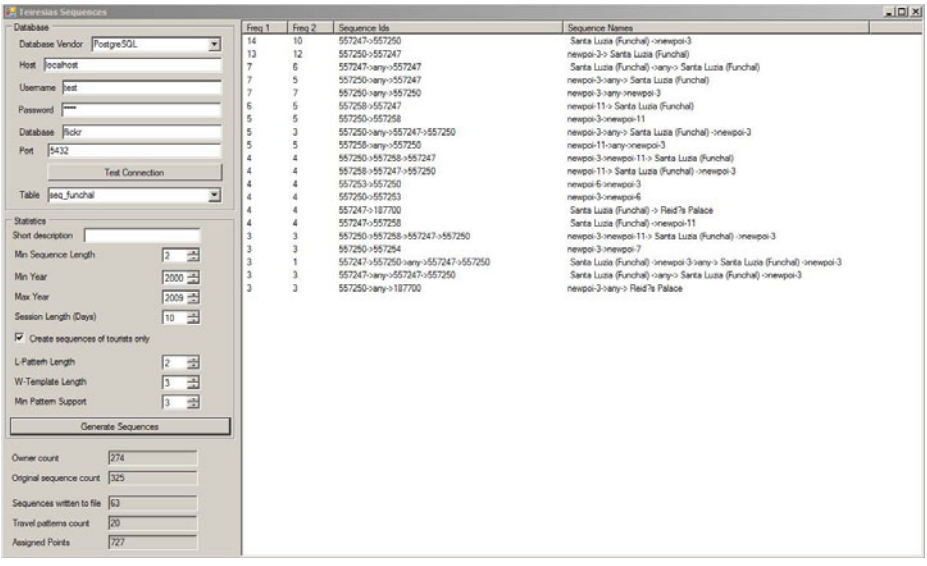

Fig. 5. Sequence patterns

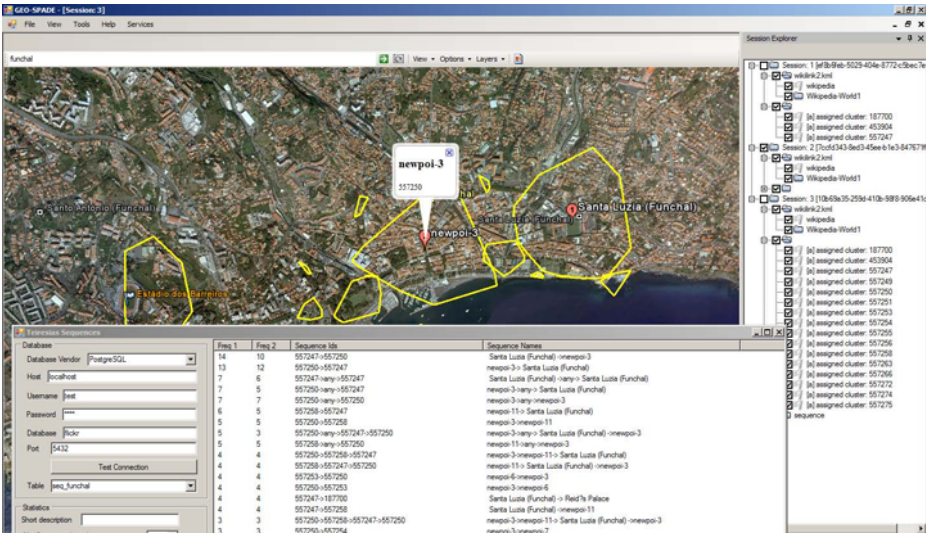

Fig. 6. Combined view of obtained sequence patterns and the map. The regions are highlighted by clicking on the sequences.

(attitude towards the quality of a photo) and/or sentiments (attitudes towards the objects depicted on a photo or mood expressions) are located in the text, the strength of the opinion and/or sentiments can be calculated and mapped to a continuous numerical scale. This allows searching for photos according to the opinion or sentiment scores.

We demonstrate how GEO-SPADE can be used by the user to perform the task of the area exploration using opinion and sentiment scores and show some technical details of the implementation.

The exploration begins by navigating to an area of interest. The control panel shown in Figure 7 is the main control panel for filtering photos according to different criteria (including opinion and sentiment scores). When the focus is on the control panel, the current visual boundaries of the map view are sent to the server. The server connects to the database and fetches the information about all photos found in the area. The response structure consists of key-value pairs separated by ";". An example of the response string is the following:
service=statistics;photos=501;minOpinion=-0.70;maxOpinion=21.30;....
The first key-value parameter denotes the type of the response in such a way that the response handler can delegate the response to the appropriate service handler. The service handler extracts key-value pairs using regular expressions. This allows extraction of the known parameters without breaking the code even if the new parameters were added to the response string without updating the client side. The general statistics are displayed on the control panel, which allows the user to see how many photos are found in the region, what are the minimum and maximum sentiments and opinion scores of these photos and other relevant information.

The next step is to retrieve the photos by applying one of the available filters (opinion, sentiment, etc.) and by restricting the number of photos on the map and on the control panel as well as by selecting the photos that were taken in a specific time period. The server's response is similar to the one described above. However, it contains two parts. The first part is intended for parsing by the control panel and includes the information for N selected photos to be displayed in the control panel and sorted according to the selected filtering criteria (opinion, sentiment). The second part is a KML string that will be delegated to the Google Earth engine. Figure 8 shows an example of the KML response that consists of two representation styles. The left side of Figure 8 shows the map view of a region of Warsaw, Poland with thumbnails of images taken in that area filtered by sentiment scores. The right side of Figure 8, shows the same map view but instead of image thumbnails, the sentiment scores are mapped to colors. Both representations allow the user to quickly explore the area either by observing the image thumbnails or by navigating to a place where higher scores are given to images, which can be seen by looking at the color of the circles. Both ways make it possible to see the image itself and to retrieve more information about it (opinion or sentiment scores, and comments) by clicking on the thumbnail.

3.3 Hotel Price Prediction

HoteLeverage is an ongoing project initiated by Travel Global Systems Inc. (TGS)[11] and carried out in collaboration between TGS, Ben-Gurion University of the Negev (Israel) and Konstanz University (Germany). Serving as a Travel Service Provider (TSP), and the hotel broker between Product Suppliers (PSs) and consumers, TGS provides solutions to travel agencies (B2B) as well as to individual consumers. Accordingly,

[11] http://www.travelholdings.com/brands_tgs.html

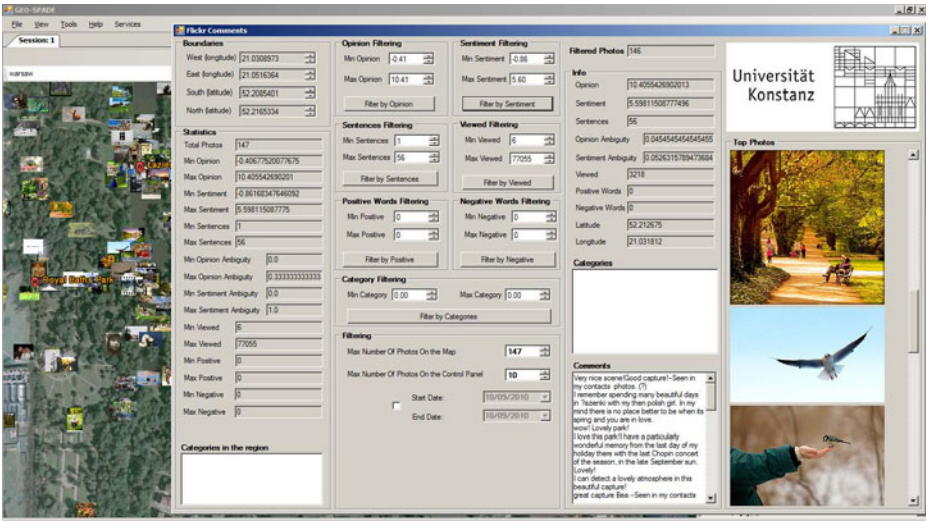

Fig. 7. The control panel for retrieving images according to one of the sorting criteria. The top-N retrieved images are also presented in the control panel. Centering the map view around the top-N photos is implemented by double-clicking the photo.

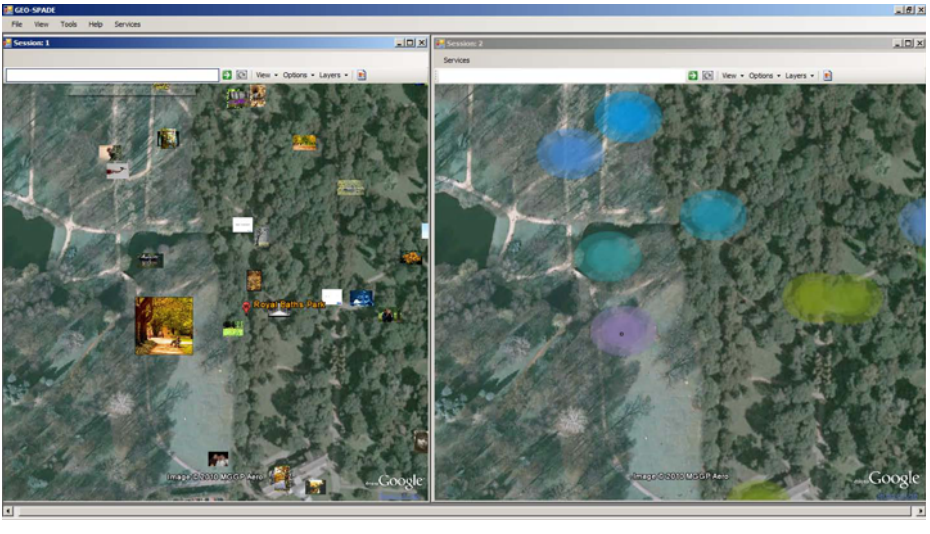

Fig. 8. Two photo representation styles: image thumbnails (left) and color coding according to the image scores of the selected criteria (right)

various PSs offer TGS attractive room prices, which TGS is committed to promoting by means of its Web sites and services. The same PS can provide other travel groups that offer similar services like TGS a lower price for the same room. This situation

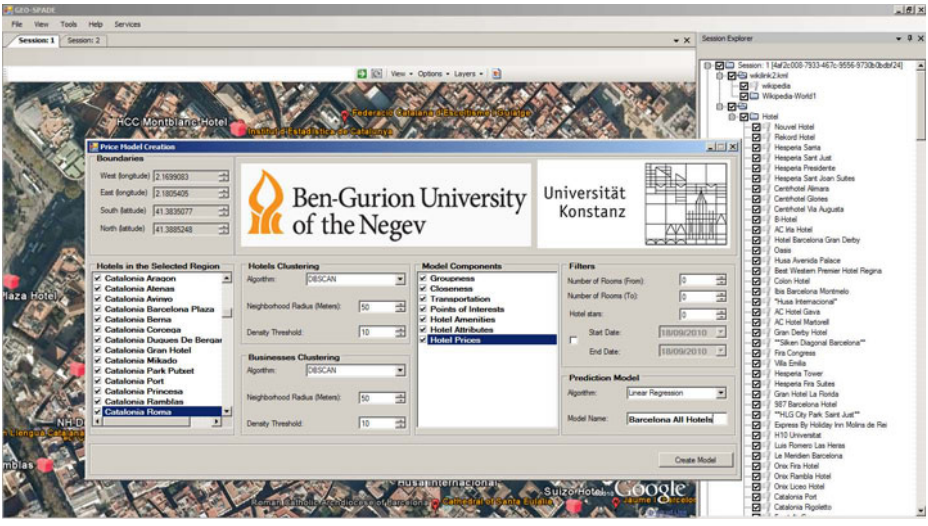

Fig. 9. The control panel for model creation. The panel includes the list of hotels that should be included in the model; clustering algorithms that are applied to hotels and business centers; various components that will be part of the model; filters and the algorithm to be used for model creation.

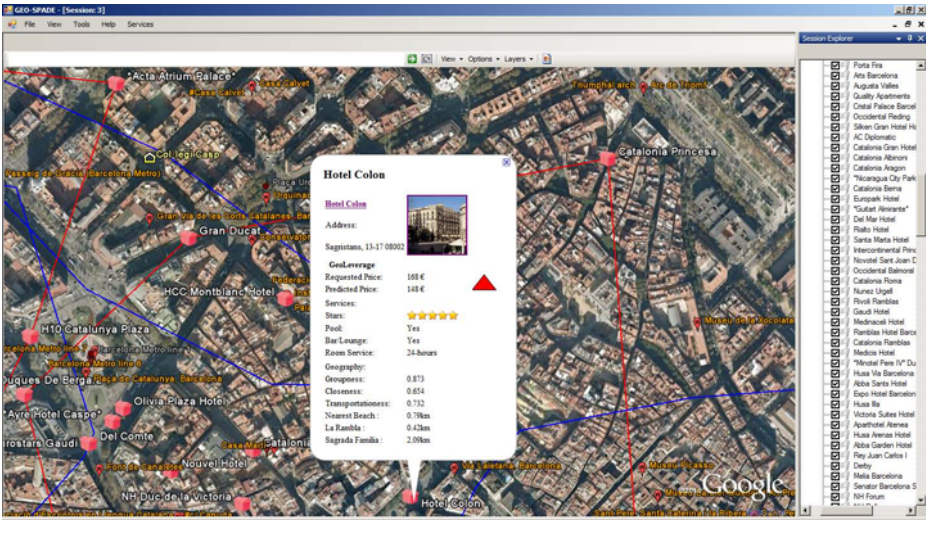

Fig. 10. The map view with hotels in the area of Barcelona, Spain. The requested and predicted prices for the selected hotel are shown along with other relevant information about the hotel. The red triangle indicates that the proposed price is higher than the predicted one.

can lead to considerable loss of revenue. However, searching for the prices of the rival companies is unfeasible due to the high volatility of search space (the hotel prices differ daily and, there is vast amount of PSs and TSPs). Therefore, it was decided to develop a decision support system that can identify hotels within TGS's own database that possess the same characteristics (price changes, closeness to the transportation or points of interest, amenities in the room, etc...) and predict the objective price of the hotel in question. The comparison of the offered and predicted prices can give TGS the "leverage" to negotiate the lower price with a product supplier.

The project consists of three milestones. By the first milestone, Ben-Gurion University students developed and evaluated data mining techniques for price prediction. By the second milestone, the technique was integrated into GEO-SPADE, while by the third milestone the system will be integrated into the TGS infrastructure.

Figure 9 demonstrates one of the control panels available to the domain expert. The domain expert selects the region of interest. The list of hotels located in the region are drawn from the database and presented in the control panel. By default, all the hotels are selected for processing. However, the domain expert can remove hotels from the list. Next, the domain expert selects the appropriate algorithm for hotels and business locations clustering. The hotels and business locations clustering is used for estimating the closeness of a hotel to other hotels and to relevant businesses. Then, the domain expert selects the components that should be included into the model (closeness to points of interest, hotel amenities, hotel attributes, prices, etc...). If needed, filters are selected (hotels with specific number of rooms, hotel category and a specific period). Finally, she selects the algorithm for building prediction model and the name of the model for future reference.

Figure 10 shows the map view with an example of predicted price for a hotel in Barcelona, Spain. It shows the requested and predicted price. The red triangle indicates that the proposed price is higher than the predicted one. In addition, it shows other relevant information about the hotel and the calculated statistics from the model such as distance to the nearest beach or to the main street.

4 Conclusions and Future Work

This paper extends [19] by focusing on real-life scenarios of spatial analysis, exploration and decision support. Specifically, we presented three case studies in which GEO-SPADE is successfully evaluated on problems such as analysis of tourist activity, geo-tagged photo search and hotel price prediction. This is achieved by the architecture that is based on a thin client paradigm, pluggable components and SOA-based architecture in which the core of the functionality is developed separately in any programming language and run on the server, while the results of the computation are transferred using Web services. The results that are targeted for direct visualization are communicated in the KML format and delivered to the Google Earth engine. The results that are required for further processing are communicated to the client-side plug-in component in any format suitable for the developer.

In future research we will concentrate on enhancing the capabilities of the framework and extending the use of the framework by working on various GEO-related problems.

We will also integrate several mapping technologies into GEO-SPADE such as 2D maps from different providers while preserving KML as a primary protocol for data interchange and visualization.

Acknowledgements. This work was partially funded by the German Research Society (DFG) under grant GK-1042 (Re-search Training Group "Explorative Analysis and Visualization of Large Information Spaces"), and by the Priority Program (SPP) 1335 ("Visual Spatio-temporal Pattern Analysis of Movement and Event Data").

References

1. Peters, D.: Building a GIS: System Architecture Design Strategies for Managers. ESRI Press (2008)
2. Friis-Christensen, A., Ostlander, N., Lutz, M., Bernard, L.: Designing service architectures for distributed geoprocessing: Challenges and future directions. Transactions in GIS 11(6), 799 (2007)
3. Diaz, L., Granel, C., Gould, M., Olaya, V.: An open service network for geospatial data processing. In: An Open Service Network for Geospatial Data Processing: Free and Open Source Software for Geospatial (FOSS4G) Conference (2008)
4. Schaeffer, B., Foerster, T.: A client for distributed geo-processing and workflow design. Location Based Services Journal 2(3), 194–210 (2008)
5. Foerster, T., Schaeffer, B., Brauner, J., Jirka, S., Muenster, G.: Integrating ogc web processing services into geospatial mass-market applications. In: International Conference on Advanced Geographic Information Systems & Web Services, pp. 99–103 (2009)
6. Andrienko, G., Andrienko, N., Dykes, J., Fabrikant, S., Wachowicz, M.: Geovisualization of dynamics, movement and change: key issues and developing approaches in visualization research. Information Visualization 7(3), 173–180 (2008)
7. Grossner, K.E.: Is google earth, "digital earth?" - defining a vision. University Consortium of Geographic Information Science, Summer Assembly, Vancouver, WA (2006)
8. Patterson, T.C.: Google Earth as a (not just) geography education tool. Journal of Geography 106(4), 145–152 (2007)
9. Goodchild, M.: The use cases of digital earth. International Journal of Digital Earth 1(1), 31–42 (2008)
10. Sheppard, S., Cizek, P.: The ethics of Google Earth: Crossing thresholds from spatial data to landscape visualisation. Journal of Environmental Management 90(6), 2102–2117 (2009)
11. Farman, J.: Mapping the digital empire: Google Earth and the process of postmodern cartography. New Media & Society 12(6), 869–888 (2010)
12. Smith, T., Lakshmanan, V.: Utilizing google earth as a gis platform for weather applications. In: 22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology, Atlanta, GA, January 29-February 2 (2006)
13. Wood, J., Dykes, J., Slingsby, A., Clarke, K.: Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. IEEE Transactions on Visualization and Computer Graphics 13(6), 1176 (2007)
14. Compieta, P., Di Martino, S., Bertolotto, M., Ferrucci, F., Kechadi, T.: Exploratory spatio-temporal data mining and visualization. Journal of Visual Languages and Computing 18(3), 255–279 (2007)
15. Slingsby, A., Dykes, J., Wood, J., Foote, M., Blom, M.: The visual exploration of insurance data in google earth. In: Proceedings of Geographical Information Systems Research UK (GISRUK), pp. 24–32 (2008)

16. Pezanowski, S., Tomaszewski, B., MacEachren, A.: An open geospatial standards-enabled google earth application to support crisis management. Geomatics Solutions for Disaster Management, 225–238 (2007)
17. Di Martino, S., Bimonte, S., Bertolotto, M., Ferrucci, F.: Integrating google earth within olap tools for multidimensional exploration and analysis of spatial data. In: Proceedings of the 11th International Conference on Enterprise Information Systems, pp. 940–951 (2009)
18. Stensgaard, A., Saarnak, C., Utzinger, J., Vounatsou, P., Simoonga, C., Mushinge, G., Rahbek, C., Møhlenberg, F., Kristensen, T.: Virtual globes and geospatial health: the potential of new tools in the management and control of vector-borne diseases. Geospatial Health 3(2), 127–141 (2009)
19. Kisilevich, S., Keim, D., Rokach, L.: Geo-spade: A generic google earth-based framework for analyzing and exploring spatio-temporal data. In: 12th International Conference on Enterprise Information Systems, pp. 13–20 (2010)
20. Kisilevich, S., Keim, D., Rokach, L.: A novel approach to mining travel sequences using collections of geo-tagged photos. In: The 13th AGILE International Conference on Geographic Information Science, pp. 163–182 (2010)
21. Kisilevich, S., Krstajic, M., Keim, D., Andrienko, N., Andrienko, G.: Event-based analysis of people's activities and behavior using flickr and panoramio geo-tagged photo collections. In: The 14th International Conference Information Visualization, pp. 289–296 (2010)
22. Kisilevich, S., Rohrdantz, C., Keim, D.: "beautiful picture of an ugly place". exploring photo collections using opinion and sentiment analysis of user comments. In: Computational Linguistics & Applications, CLA 2010 (2010)
23. Graupmann, J., Schenkel, R.: GeoSphere-Search: Context-Aware Geographic Web Search. In: 3rd Workshop on Geographic Information Retrieval (2006)
24. Ferraz, V.R.T., Santos, M.T.P.: Globeolap: Improving the geospatial realism in multidimensional analysis environment. In: 12th International Conference on Enterprise Information Systems, pp. 99–107 (2010)
25. Wachowicz, M., Ying, X., Ligtenberg, A., Ur, W.: Land use change explorer: A tool for geographic knowledge discovery. In: Anseling, L., Rey, S.J. (eds.) New Tools for Spatial Data Analysis, Proceedings of the CSISS Specialist Meeting (2002)
26. Lundblad, P., Eurenius, O., Heldring, T.: Interactive visualization of weather and ship data. In: Proceedings of the 13th International Conference Information Visualisation, Washington, DC, USA, pp. 379–386. IEEE Computer Society, Los Alamitos (2009)
27. Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. Bioinformatics-Oxford 14(1), 55–67 (1998)
28. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
29. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. KDD, vol. 96, pp. 226–231 (1996)
30. Girardin, F., Fiore, F.D., Ratti, C., Blat, J.: Leveraging explicitly disclosed location information to understand tourist dynamics: a case study. Jouranl of Location Based Services 2(1), 41–56 (2008)
31. Andrienko, G., Andrienko, N., Bak, P., Kisilevich, S., Keim, D.: Analysis of community-contributed space-and time-referenced data (example of panoramio photos). In: GIS 2009: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 540–541 (2009)