Journal of
Applied Remote Sensing

Case study of visualizing global user download patterns using Google Earth and NASA World Wind

Ziliang Zong
Joshua Job
Xuesong Zhang
Mais Nijim
Xiao Qin

SPIE

Case study of visualizing global user download patterns using Google Earth and NASA World Wind

Ziliang Zong,[a] Joshua Job,[b] Xuesong Zhang,[c] Mais Nijim,[d] and Xiao Qin[e]
[a]Texas State University, Department of Computer Science, 601 University Drive, San Marcos, Texas 78666
zz11@txstate.edu
[b]L3 Communication Inc., 640 North 2200 West, Salt Lake City, Utah 84116
[c]Joint Global Change Research Institute, Pacific Northwest National Laboratory, 5825 University Research Court, Suite 3500, College Park, Maryland 20740
[d]Texas A&M-Kingsville, Department of Electrical Engineering and Computer Science, 700 University Boulevard, Kingsville, Texas 78363
[e]Auburn University, Department of Computer Science and Software Engineering, Auburn, Alabama 36849

Abstract. Geo-visualization is significantly changing the way we view spatial data and discover information. On the one hand, a large number of spatial data are generated every day. On the other hand, these data are not well utilized due to the lack of free and easily used data-visualization tools. This becomes even worse when most of the spatial data remains in the form of plain text such as log files. This paper describes a way of visualizing massive plain-text spatial data at no cost by utilizing Google Earth and NASA World Wind. We illustrate our methods by visualizing over 170,000 global download requests for satellite images maintained by the Earth Resources Observation and Science (EROS) Center of U.S. Geological Survey (USGS). Our visualization results identify the most popular satellite images around the world and discover the global user download patterns. The benefits of this research are: 1. assisting in improving the satellite image downloading services provided by USGS, and 2. providing a proxy for analyzing the "hot spot" areas of research. Most importantly, our methods demonstrate an easy way to geo-visualize massive textual spatial data, which is highly applicable to mining spatially referenced data and information on a wide variety of research domains (e.g., hydrology, agriculture, atmospheric science, natural hazard, and global climate change). © *2012 Society of Photo-Optical Instrumentation Engineers (SPIE).* [DOI: 10.1117/1.JRS.6.061703]

Keywords: geo-visualization; spatial data; satellite images; Google Earth; NASA World Wind.

Paper 12048SS received Feb. 23, 2012; revised manuscript received Aug. 27, 2012; accepted for publication Sep. 18, 2012; published online Oct. 9, 2012.

1 Introduction

Data visualization is playing an increasingly important role in representing and analyzing scientific data. Compared to plain text data, visualized data communicates information much more clearly and effectively. A large amount of scientific data (e.g., the earth satellite image data collected by NASA and USGS) carries geographical information. These data are called geospatial data, and geo-visualization refers to the techniques used to visualize geospatial data.

Geospatial data essentially requires significant time and effort to collect, and it tends to carry information of great value to both academic and commercial entities. Existing data visualization tools are either costly, e.g. ArcGIS extensions,[1] or not applicable to the general scientific research domains, e.g. Amaztype[2] and Flickrtime.[3] Although some powerful and free geo-visualization tools, e.g. Google Earth[4] and NASA World Wind,[5] are capable of generating high-quality visualization results for spatial data, they require specific data input formats, which forms an invisible wall between the raw scientific data (likely in plain-text format) and the final visualization results. This research is motivated by the great need for efficient and cost-effective geo-visualization tools for analyzing scientific spatial data. The primary contributions of this paper include: 1. the design and implementation of mash-up tools for utilizing Google Earth and NASA World Wind to visualize plain-text scientific data, and 2. an illustration of the effectiveness of these methods by visualizing over 170,000 global download requests for satellite images maintained by USGS EROS.

2 Background

2.1 *Related Work*

Nowadays, the lack of data is no longer a problem. Lyman and Varian[6] estimate that in 2002 alone, the world produced 23 exabyte of new stored information. Their study estimates that storage of new information is growing at a rate of more than 30% per year. Instead, the lack of effective tools to analyze enormous amounts of data has become a serious problem. Data visualization has been widely accepted as one of the effective ways to facilitate massive data analysis by detecting the expected and discovering the unexpected information. In 2006, Professor Hans Rosling illustrated a new way of presenting statistical data by combining data visualization techniques, which is capable of turning complex global trends into lively animations.[7]

A large number of interesting applications utilize visual analytics tools and techniques to synthesize information and derive insight from massive and dynamically changed data. For example, Mindmap is able to visualize the most successful websites based upon category, proximity, success, and popularity.[8] Newsmap is an application that visually reflects the constantly changing landscape of the Google News aggregator.[9] Amaztype applies data visualization techniques to collect information from Amazon and present the matched books in an intuitive way to user.[2] A similar idea is being used by Flickrtime, which uses Flickr API to present the uploaded images in real-time.[3] Visualizing the Power Struggle in Wikipedia displays the most popular articles and the most frequent search queries in the heat map.[10] Last year, Chan et al. presented their idea of Visapedia, a visualized version of Wikipedia.[11]

These new techniques and applications have significantly changed the way we view data and web content. Unfortunately, we find it difficult to incorporate them into the analysis of massive scientific data repositories. This is partially because most applications are specifically designed for a unique web application, but a large portion of scientific data is in textual form (e.g. log files), and they are inherently geo-referenced. Therefore, geo-visualization tools that can address visual exploration, analysis, synthesis, and presentation of geospatial data are critical for perceiving complex structures, detecting geospatial patterns, and analyzing relationship and trends in the massive spatial data.[12–15]

An efficient means of developing geo-visualization tools is to use a mash-up technique that allows integrating multiple applications and data sets to create a new application tailored for a specific task.[16,17] Interactive web mapping services, such as Google Earth[4] and NASA World Wind,[5] enable users to exploit geospatially visualized mash-ups[18] that combine local data sets with the data available from external severs.[15] These web mapping services provide APIs that support HTML, Java, and XML and allow developers to create custom applications. In recent years, there has been a quick rise in the application of these mapping services to developing geo-visualization tools. For example, Zhang and Shi[19] presented a web mapping application based on a conference presentation of geographic location using Google Maps. Fisher[20] developed Hotmap, a mash-up visualization tool based on Microsoft's Live Search Maps, to visualize the number of downloads of imagery and understand the users' interaction with the world. Wood et al.[17] demonstrated a geo-visualization mash-up case study using Google Earth for interactive synthesis of encodings generated by MySQL, PHP, and LandSerf. They used this combination of tools to visually explore a mobile directory service log file containing 1.42 million records.

2.2 *NASA and USGS Landsat Project*

The Landsat program[21,22] is a series of earth-observing satellite missions jointly managed by NASA and USGS. Landsat satellites have been collecting information about the earth from space since 1972. In the Landsat 7 project, NASA is responsible for the development and launch of the satellite, while the USGS is responsible for capturing, processing and archiving the satellite images. In September 2008, USGS made Landsat 7 satellite images freely downloadable through the USGS Global Visualization Viewer (http://glovis.usgs.gov/) and the Earth Explorer (http://edcsns17.cr.usgs.gov/EarthExplorer/) websites. These global, high-resolution, and multispectral satellite images enable people to study many aspects of our planet and to evaluate the dynamic changes caused by both natural processes and human practices. Some download requests can be filled within a short period of time, if the selected satellite images were previously requested and cached in the download servers. However, if the requested images were not processed previously, the Distributed Active Archive Center (DAAC) must activate the Landsat 7 Processing System (LPS) and the Landsat 7 Product Generation System (LPGS) to generate the target images, which may take up to several hours or days to complete. Each time a user request is successfully served, the download servers will record according information, which includes USER_REQUEST_ID, LANDSAT_IMAGE_ID, WRS_PATH, WRS_ROW, and REQUEST_TIME etc. Here WRS refers to the Worldwide Reference System of paths and rows, which is a standard rule for documenting satellite images. This information will be written to system log files in the form of plain text. These log files serve as the input for our geo-visualization mash-up tools, which can visually analyze global download patterns.

3 Implementation

In this section, we describe the design methodology and implementation details of our geo-visualization mash-up tools.

3.1 *Mash-Up Methodology*

There are three design philosophies for developing geo-visualization tools. The first philosophy is to design an entirely new tool from scratch. This is obviously difficult and time consuming. The second philosophy is to take advantage of existing geographic information system (GIS) software, e.g. ArcGIS, by implementing extension libraries on the basis of built-in functionalities. This off-the-shelf strategy is capable of providing required geo-visualization ability but often costly because newly developed functionalities are bound to the commercial GIS software. Recent publications[17–20] have indicated that mash-up techniques have considerable potential for rapid development of geo-visualization tools. Mash-up allows easy incorporation of web-oriented technologies (RSS, XML, AJAX, etc.) as well as popular programming languages (Perl, Java, and C++) into existing interactive web mapping services, such as Google Earth[4] and NASA World Wind.[5] Figure 1 illustrates the mash-up approaches used in our geo-visualization tools for Google Earth[4] and NASA World Wind.[5]

3.2 *Data Preprocessing*

The preprocessing module takes the raw data files as input and generates the intermediate data files, which are compatible to Google Earth or NASA World Wind. More specifically, the preprocessing module contains a data conversion component, a data filtering component, and a data aggregation component. The data conversion component converts the row/path information in the raw data to longitude/latitude information, which is acceptable by both Google Earth and NASA World Wind. The data filtering component cleans up the raw data file by filtering invaluable and redundant information. The data aggregation module is able to generate aggregated data sets when precomputation is needed. For example, in order to calculate the number of download requests for each satellite image, we execute the data aggregation process in advance and put the results in the intermediate data file. This can considerably reduce the computational workload for Google Earth and NASA World Wind and thereby substantially improve performance. We made the decision to implement this module using the Perl language because Perl provides effective and powerful file manipulation functions, which are greatly helpful when dealing with massive spatial data in log files.

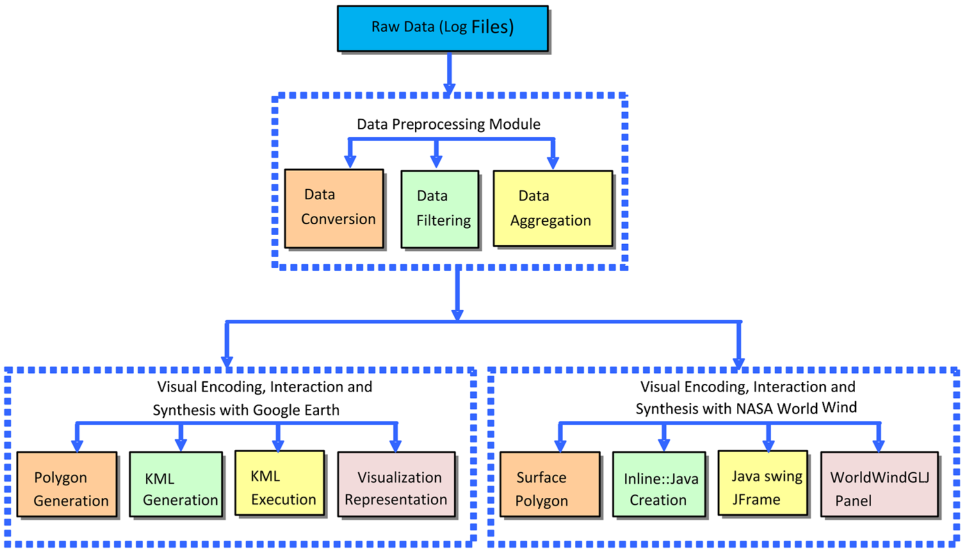


Fig. 1 Mash-up workflow for Google Earth and NASA World Wind.

3.3 *Visual Encoding, Interaction, and Synthesis*

3.3.1 *Google Earth*

Google Earth is a widely used geo-browser that provides an exploratory interface to a rich series of spatial data sets. Google Earth displays information specified by the Keyhole Mark-up Language (KML).[23] KML is an extensible mark-up language (XML) grammar and file format for modeling and storing geographic features such as points, lines, images, and polygons for display in Google Earth.

To geo-visualize the global download requests, the visual encoding, interaction, and synthesis module must be able to position images on the surface of the earth using latitude/longitude, group symbols, and features in a hierarchical folder structure and combine files that contain related spatial information. The key step here is to create the polygons, which can correctly represent the aggregated spatial information. To achieve this in KML, every polygon needs to be surrounded by a <Placemark> element. The <Placemark> element contains the object to be displayed and the metadata associated with it. The <Name> and <PolyStyle> fields are also filled by the scripts accordingly. The biggest challenge when creating polygons stems from the designation of earth locations that are antipodal to the Prime Meridian (± 180 deg). To fix this problem, a polygon is divided along the 180-deg longitude line. Each part of the divided polygon is placed in a separate <Polygon> element. Both of these new elements are placed inside a <MultiGeometry> element. Finally, the <MultiGeometry> element is placed in a <Placemark> element. Figure 2 illustrates the pseudo code of the algorithm to split a simple quadrilateral along 180 degrees longitude. Another problem involved the placement of polygons associated with each path in a separate file with a <Folder> element surrounding the information in the file. Because the globe is divided into 248 rows, we created 248 KML files for each row. Finally, a separate index file is created to point to each row file using a <NetworkLink> element.

3.3.2 *NASA World Wind*

World Wind is another geo-visualization toolkit developed by NASA using the Java SDK. It uses Java OpenGL (JOGL) to render the globe. As such, it can be ported to any platform that supports JOGL. World Wind presented a different set of problems beyond those of Google Earth. The first problem regarded how the pre-processing module, which is implemented using Perl, interfaces with the Java classes of World Wind. To address this problem, the Perl module Inline :: Java was created to connect to the Java code. The second problem is similar to the problem in Google Earth—creating polygons. We defined a new polygon class that inherits from the World Wind

```
polyList //List containing all points of the polygon, in clockwise or counter-clockwise order
eastPoly //List for points of the east polygon
westPoly //List for points of the west polygon
for i = 1 to polyList.length do
    if polyList[i].longitude < 0 then
        eastPoly.append(polyList[i])
    else
        westPoly.append(polyList[i])
    end if
    //Next item in the list(wraps around)
    j = (i + 1) % polyList.length
    if abs(polyList[i].longitude - polyList[i].longitude) > 180 then
        newPoint1
        newPoint2
        //Calculate the latitude of the crossing by using the  formula for a line
        latitude = calcLatitude(polyList[i],polyList[j])
        newPoint1.latitude = -180
        newPoint2.latitude = 180
        newPoint1.latitude = newPoint2.latitude = latitude
        eastPoly.append(newPoint2)
        westPoly.append(newPoint1)
    end if
end for
```

Fig. 2 Algorithm to split a simple quadrilateral along 180 deg longitude.

SurfacePolygon class. This class holds information for a polygon and provides the functions for rendering it onto a bitmap. These bitmaps are then rendered on the globe by a class implementing the Layer interface. The new polygon class contains added functions to allow Perl lists to be used to define a polygon. It also adds hooks into the coordinate and color Perl modules to convert them to the Java classes required by the World Wind API. A Perl wrapper module is implemented to make this conversion easier for the display wrapper. The display class provides a display (a Java swing JFrame) for polygons generated by the Perl scripts. The polygons are displayed on an instance of WorldWindGLJPanel inside an instance of the Layer described below. This display class is wrapped by a Perl module to interface better with the Perl scripts. In addition, the display class gives users the ability to take pictures of the WorldWindGLJPanel.

During the initial test of the interface, we found the third problem—the high number of polygons drawn by the Perl scripts caused an extreme performance loss. This is primarily because World Wind organizes images placed on the globe into layers. Polygons must first be rendered to an image before they can be placed on the globe. The layer responsible for this was rendering polygons to a bitmap every time the layer itself was rendered. This works for a small number of polygons. However, performance degraded substantially when the Perl scripts were processing large number of polygons. To address this problem, the function in the polygon class responsible for rendering polygons is changed from protected to public. This allows the generated images to be kept and reused every time the layer is drawn on the globe, which can greatly improve performance (save 38% of time) because these polygons are only rendered once.

4 Geo-Visualization Results

To evaluate the performance and effectiveness of our mash-up tools, we geo-visualize over 170,000 global download requests provided by USGS. Our geo-visualization tools are able to identify the popular geographic areas around the world within several minutes. In addition, the geo-visualization results facilitate the discovery of global user download patterns.

4.1 *Identify Global "Hot-Spots"*

Figure 3 shows the geo-visualization results generated by Google Earth. The raw data used is the global download requests during October 8, 2008 to March 9, 2009. Our tools are able to mark different areas using different colors based on different number of requests. The background

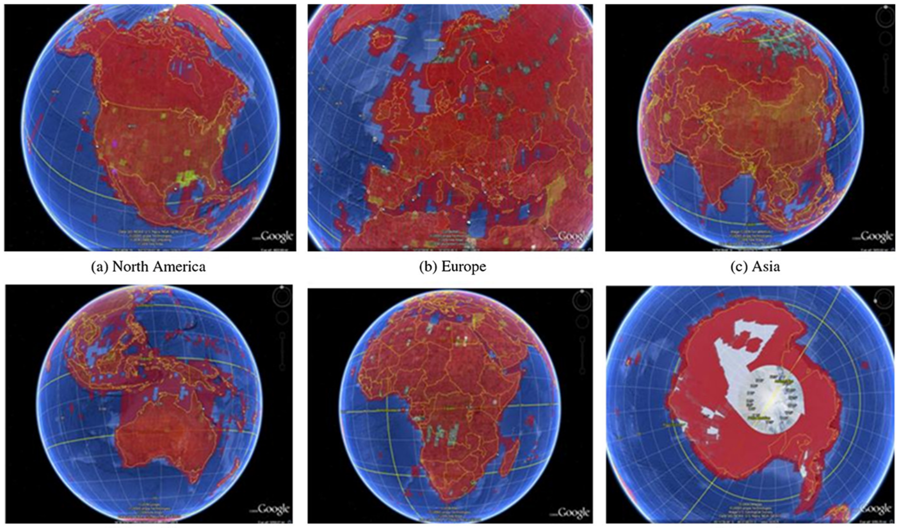

(a) North America (b) Europe (c) Asia (d) Australia (e) Africa (f) Antarctica

Fig. 3 Geo-visualization results of North America, Europe, Asia, Australia, Africa, and Antarctica in Google Earth.

color is light blue, which means there is no request for areas in blue. The red color indicates low number of requests and the bright saturated colors, e.g. yellow and purple, indicate high number of requests. As demonstrated in Fig. 3, requests cover most of all major land masses on the globe. Meanwhile, each continent has its own "hot-spots." We summarize our findings for each continent as follows.

- North America: Most requests for North American images are for the mainland U.S. Popular spots include New York City, the New Orleans area, Central Nevada and California. The areas affected by Hurricane Katrina are popular as well.
- Europe: Most requests for Europe are for Southern Europe with Southern Spain standing out. Number of requests for Israel, Lebanon, and Syria also stays high.
- Asia: Requests for Asia are centered in Southern Mongolia, Northern China, and South Eastern Asia. Popular images include areas in Iran, Kashmir, Bangladesh and Tokyo.
- Africa: Requests for Africa are for the Sahara Desert, coastal regions, and a section including Northern Congo and the Dominican Republic. There are also a high number of requests for desert regions of Southern Africa.

Table 1 Top 10 download areas (October 8, 2008 to March 9, 2009).

Rank	# of Requests
1. Nevada Desert, USA	850
2. Northern Baja California, Mexico	622
3. Southern California, USA	593
4. Central Western Washington, USA	527
5. Southern Peru	397
6. West Central South Dakota, USA	376
7. Washington, DC area, USA	356
8. South Western Louisiana, USA	353
9. Albuquerque, New Mexico, USA	329
10. Kansas City, Kansas, USA	322

- Australia and the Southern Pacific: All major land masses in the area are covered as well as a number of islands. Popular areas include a section of Indonesia, parts of western Queensland, Australia, and a region in central Australia.
- South America: South America appears to be well covered. Most of the popular spots correspond to populated areas. Four hot spots stand out from the Brazilian Rainforest.
- Antarctica: The western portion of Antarctica is well covered with some gaps in the Eastern portion. The edge of what areas covered by the Landsat satellites is very noticeable.

Table 1 lists the top 10 popular areas during October 8, 2008 to March 9, 2009.

4.2 *Discover User Downloading Patterns*

One of the major challenges of the system is how to minimize the service time: the time needed between users sent their requests and the requested images are ready for downloading. As mentioned in Sec. 2.2, the service time may vary from several seconds, when requested images are already in the download servers, to several hours, when requested images are not available in the download servers. Because only a small portion of satellite images can reside in the download

Table 2 Top 10 downloading areas (October 8, 2008 to January 8, 2009).

Rank	# of Requests
1. Nevada Desert, USA	621
2. Baja California, Mexico	608
3. East Central Washington, USA	331
4. Louisiana Coastline, USA	258
5. South Eastern Mississippi, USA	257
6. South Western corner of Louisiana, USA	248
7. South East Mississippi, USA	240
8. Jackson Mississippi, USA	226
9. Coast Near New Orleans, Louisiana, USA	217
10. New Orleans, Louisiana, USA	216

Table 3 Top 10 downloading Areas (January 9, 2009 to March 9, 2009).

Rank	# of Requests
1. Southern California, USA	421
2. Southern Peru	381
3. West Central South Dakota, USA	350
4. East Central Kansas, USA	299
5. East Central Kansas, USA	287
6. Central New Mexico, USA	262
7. West Central New Mexico, USA	258
8. Tampa, Florida, USA	243
9. Southern Coastal Brazil	237
10. Central Nevada, USA	229

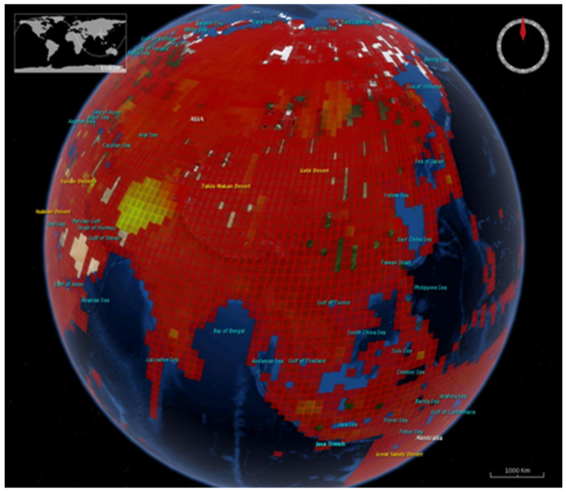


Fig. 4 Bursting requests for Afghanistan during January 2009 to March 2009.

servers, it is critical to cache the most popular satellite images in the download servers in order to shorten average service time. The identified "hot spots" by our geo-visualization tools can help to select qualified candidates for caching. Meanwhile, the user interests evolve over time. For example, we can clearly see the evolved patterns by comparing the different top 10 downloading areas during October 8, 2008 to January 8, 2009 and January 9, 2009 to March 9, 2009, which are shown in Tables 2 and 3, respectively. Figure 4 derived from World Wind shows the bursting requests for satellite images of Afghanistan (the areas in yellow color) after January 9, 2009. Therefore, it is beneficial to regularly visualize the global spatial data and trace the evolving user patterns. Our tools provide an effective way to do this at no cost.

Additionally, we noticed that some users from China are aggressive. The typical character of aggressive users is that they send a large number of requests (e.g., >100) within a short time period. Figure 5 shows that requests from China cover almost all Chinese territory, among which a majority of areas in northern China are requested frequently (marked as light yellow color).

5 Conclusions and Future Work

This paper presents the mash-up techniques for geo-visualizing textual spatial data by utilizing Google Earth and NASA World Wind. To demonstrate the effectiveness of proposed geo-visualization tools, we visualize over 170,000 global download requests for satellite images maintained by the Earth Resources Observation and Science (EROS) Center of U.S. Geological Survey (USGS). The visualization results can be used to identify download "hot spots," discover

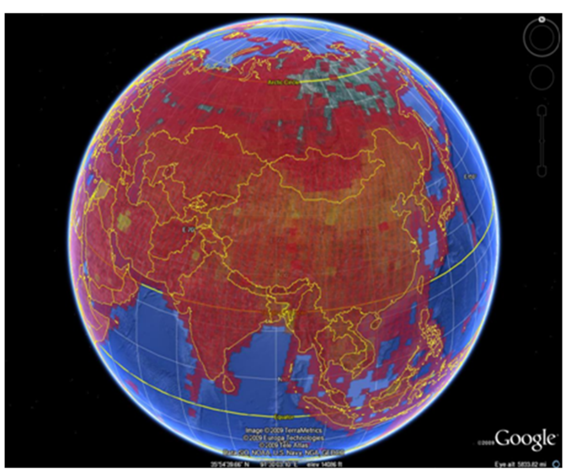


Fig. 5 The download patterns of aggressive users from China.

ever-evolving user download patterns, and help to cache of the most popular satellite images on download servers thereby reducing average service time. The future work of this research should be focused on further improving performance and usability of the proposed mash-up tools. The over-plotting and crowding problems when generating polygons also need to be carefully addressed.

Acknowledgments

The authors sincerely appreciate the comments and feedback from the anonymous reviewers. Their valuable discussions and thoughts have tremendously helped in improving the quality of this paper. The work reported in this paper is supported by the U.S. National Science Foundation under Grants Nos. CNS-1212535, CNS-0915762, CNS-0917137, CNS-1048432, and the Texas State University Library Research Grant. We also gratefully acknowledge the support from the U.S. Geological Survey (USGS) Earth Resources Observation and Science (EROS) Center.

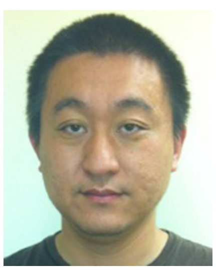
References

1. National Cancer Institute, "Geo-visualization tools development," http://gis.cancer.gov/nci/geo-visualization.html (accessed November 2011).
2. Amaztype, "Visual Amazon Shopping," (2001), http://amaztype.tha.jp/ (accessed November 2011).
3. Yahoo Inc., "Flickr," http://www.hottoast.org/convexstyle/flickrtime/ (accessed November 2011).
4. Google Inc., "Google Earth," (2011), http://earth.google.com/ (accessed November 2011).
5. National Aeronautics and Space Administration (NASA), "World Wind," (2011), http://worldwind.arc.nasa.gov/ (accessed November 2011).
6. P. Lyman and H. R. Varian, "How much information?" (2011), http://www.sims.berkeley.edu/how-much-info (accessed November 2011).
7. H. Rosling, "Show the best stats you have ever seen," (2006), http://www.ted.com/index.php/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen.html (accessed June 2006).
8. O. Reichenstein, "Web Trend Map 2007 Version 2.0," (2011), http://informationarchitects.jp/ia-trendmap-2007v2/ (accessed November 2011).
9. M. Weskamp, "Newsmap," (2011), http://marumushi.com/apps/newsmap/ (accessed November 2011).
10. T. Holloway, "Visualizing the power struggle in Wikipedia," http://abeautifulwww.com/2007/05/20/visualizing-the-power-struggle-in-wikipedia/ (accessed May 2007).
11. B. Chan et al., "Interactive visual exploration of Wikipedia data via search-based integration," *IEEE Trans. Vis. Comp. Graph.* **14**(6), 1213–1220 (2008), http://dx.doi.org/10.1109/TVCG.2008.178.
12. J.Dykes, A. M. MacEachren, and M. J. Kraak, Eds. "Facilitating interaction for geo-visualization," *Exploring Geo-visualization*, pp. 265–291, Elsevier, Amsterdam (2005).
13. B. Jiang and Z. Li, "Geo-visualization: design, enhanced visual tools and applications," *The Carto. J.* **42**(1), 3–4 (2005), http://dx.doi.org/10.1179/000870405X52702.
14. T. Rhyne, A. MacEachren, and J. Dykes, "Guest editors' introduction: exploring geo-visualizatio," *IEEE Trans. Vis. Comp. Graph.* **26**(4), 20–21 (2006), http://dx.doi.org/10.1109/MCG.2006.80.
15. M. J. Kraak, "Beyond geo-visualization," *IEEE Comput. Graph. Appl.* **26**(4), 6–9 (2006), http://dx.doi.org/10.1109/MCG.2006.74.
16. E. Wilde, "Knowledge organization mash-ups," Technical Report TIK Report No. 245, Computer Engineering and Networks Laboratory (TIK), ETH Zürich, http://dret.net/netdret/publications#wi106f (2006).
17. J. Wood et al., "Interactive visual exploration of a large spatio-temporal dataset: reflection on a geo-visualization mash-up," *IEEE Trans. Visual. Comp. Graph.* **13**(6), 1176–1183 (2007), http://dx.doi.org/10.1109/TVCG.2007.70570.
18. T. Austin and A. Bitterer, "Google improves geospatial visualization capabilities for business," (2007), http://www.gartner.com/DisplayDocument?id=510942&ref=g_sitelink (accessed August 2007).

19. J. Y. Zhang and H. Shi, "Geospatial visualization using Google Maps: a case study on conference presenters," *The 2nd International Multisymposium on Computer and Computational Sciences*, pp 472–476. (August 2007).
20. D. Fisher, "Hotmap: looking at geographic attention," *IEEE Trans.Vis. Comp. Graph.* **13**(6), 1184–1191 (2007), http://dx.doi.org/10.1109/TVCG.2007.70561.
21. National Aeronautics and Space Administration (NASA), "Landsat program introduction," (2011), http://landsat.gsfc.nasa.gov/ (accessed November 2011).
22. "Landsat program introduction," U.S. Geological Survey (USGS), (2011), http://landsat.usgs.gov/ (accessed November 2011).
23. Google Inc., "Google KML introduction," (2011), http://code.google.com/apis/kml/ (accessed November 2011).

Ziliang Zong received his PhD in computer science from Auburn University in 2008. He is currently an assistant professor in the Computer Science Department of Texas State University. His research interests include data visualization, parallel programming, and high-performance computing.

Joshua Job received his BS in computer science from South Dakota School of Mines and Technology in 2009. He has worked as a software engineer at L-3 Communications located at the Salt Lake City since his graduation.



Xuesong Zhang received his PhD in hydrology and water sciences from Texas A&M in 2008. He is currently a research scientist at the Joint Global Change Research Institute of Pacific Northwest National Laboratory and University of Maryland. His research interests include sustainability of biofuel, carbon modeling and management, and water management under climate change.

Mais Nijim received her PhD in computer science from New Mexico Institute of Mining and Technology in 2007. She is currently an assistant professor in the Department of Electrical Engineering and Computer Science of Texas A&M-Kingsville. Her research interests include parallel and distributed systems, storage systems, security, and performance evaluation.

Xiao Qin received his PhD in computer science from the University of Nebraska-Lincoln in 2004. He is currently an associate professor of Computer Science at Auburn University. His research interests include parallel and distributed systems, real-time computing, and storage systems. He won the NSF CAREER Award in 2009.