TIN: An Interactive Image Navigator Providing Ubiquitous Access To Distributed Geo-Spatial Data

Bienvenido Vélez.
bvelez@acm.org

Amaury Cabarcas
amaury@ece.uprm.edu

Lizvette Malavé
lmalave@ece.uprm.edu

Advanced Data Management Group
Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez Campus
Mayaguez, PR 00681-9042

Abstract

*In this paper we describe the design of the **TerraScope** **I**mage **N**avigator (**TIN**), the graphical user interface module of the **TerraScope** system. **TerraScope** is an earth science data middleware system that was designed to facilitate collaboration among a set of data repositories (peers) who wish to provide their geospatial data thru an integrated portal. The base of the system is a distributed database that collects heterogeneous spatial data taken from satellite ground stations. **TIN,** as a component of **TerraScope,** retrieves images and related information from the distributed database and deals with the effective presentation of this kind of data to the user. TIN allows recursive navigation of the image space by dynamically embedding retrieved sub images as spatial hyperlinks inside other retrieved images. The **TIN** prototype was implemented using an interactive movie authoring environment (Flash MX) and XML (e**X**tensible **M**arkup **L**anguage) to communicate queries and retrieve data and metadata from the server. As a result, we expect to port **TIN** to multiple platforms, including portable devices, with relative ease. Measurements using a small image database suggest result set rendering times in the order of a few seconds and response time 97% dominated by server computation.*

1. Introduction

The emergence of multimedia technologies and the possibility of sharing and distributing geospatial satellite data through broadband computer networks have exacerbated the need for geographical information technologies. In one typical scenario, geospatial data is periodically collected at geographically distributed ground stations which often span several technical, administrative and even political domains. Solutions attempting to offer integrated access to distributed data by relying on centralization of data repositories are not often feasible.

The inherently distributed nature of geospatial data should be of no concern to scientists, students or data consumers. The information system should enable and facilitate collaboration among a set of distributed data providers who wish to collaborate to provide an image of a single data repository with minimal loss of their individual autonomy over their data. Based on this principle we are designing and developing the **TerraScope** Earth Science distributed peer-to-peer database middleware system conformed by a Search and Retrieval Engine (**SRE**) [7] and TIN, the graphical user interface (GUI) module presented in this paper.

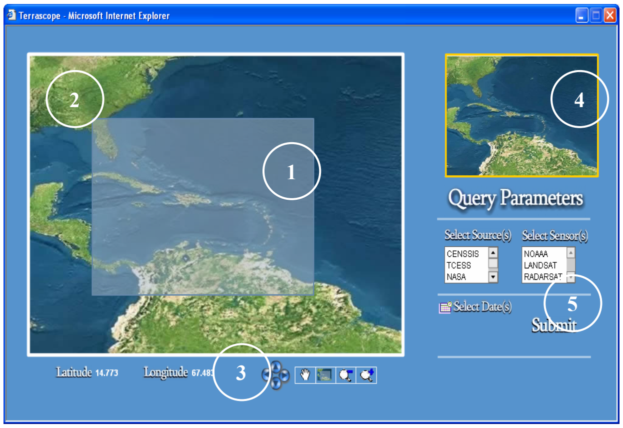


Figure 1. TIN Graphical User Interface

Using Macromedia Flash MX as a development platform, XML [19] and ActionScript [2], we have designed the first **TerraScope** **I**mage **N**avigator (**TIN)** prototype. This image navigator delivers satellite images with their corresponding metadata, GIS characteristics, and other information to any web browser with a Flash MX player installed. As shown in Figure 1, the GUI consists of two image windows (2 and 4), and a number of input controls (5). The main window (2) on the left side displays the currently selected section of the image being browsed. This section can be explored using TIN's zooming and panning controls. Initially, TIN displays a map showing the geographical region covered by the satellite ground stations contributing data to the distributed repository. The current prototype includes data collected by the Tropical Center for Earth and Space Studies (TCESS) and the Center for Subsurface Sensing and Imaging Systems (CenSSIS) both at the University of Puerto Rico Mayaguez Campus. Both of these stations collect imagery from the Caribbean region.

The smaller window (4) on the upper right hand side of the screen, which we call the periscope, continuously outlines the region in the image that the user is examining on the left window. The usefulness of this feature will become move evident when as we continue discussing the user interface in the sections that follow.

Using familiar GUI controls TIN users can restrict the scope of their search to a specific data repository, geographical region, type of satellite sensor (e.g. MODIS, RADARSAT and Landsat 7) or data collection date. Figure 1 shows that a user has selected the region that appears shaded at the center of the image using the rubber band tool. By default the scope of the search is delimited by the geographical region covered by the entire image, but if the user selects a sub region using this tool, the scope of the search is the geospatial area delimited by the selected area. This feature is important especially since TIN is designed to support repositories that continuously collect data.

TIN clients obtain their data from a single ***TerraScope* SRE**. However, in order to satisfy the user request, this SRE may contact other peer SREs holding data matching the criteria specified by the user. Thus, SREs act as servers to TIN clients, and also as clients of other SREs. This type of architecture is often termed peer-to-peer. The reader is referred to [7] for mode details of the Terrascope SRE.

The remainder of this paper is organized as follows: Section 2 describes some related previous research in image retrieval and some projects that have proposed image browsers in the past, Section 3 discusses TIN's architecture. The graphical user interface and the image retrieval process is described in more detail in section 4. In Section 5, we discuss some performance data as well as experimental results from a user study comparing TIN to another system. Section 6 summarizes the lessons learned from using Flash MX as an application development platform. Finally, Section 7 presents our conclusions and suggests some areas for future work.

2. Previous Work

In an effort to amortize the high costs associated with the deployment of the expensive equipment and communications infrastructure required to collect satellite imagery, many scientific research centers and universities have been developing systems to share their image data over the Internet. In this section we compare some of these systems emphasizing the features that distinguish them from TIN.

The Quicklook Swath Browser is an image browser developed by the Canada Centre for Remote Sensing [12]. This system has some similarities with TIN. Both create geo-representations of the images, use similar mechanisms to display textual data and metadata, and use a graphical user interface accessible through web. Also both systems provide automatically hyperlinked images. The Quicklook Swath Browser is Java based while TIN was developed using Macromedia Flash MX. The Quicklook Swath Browser does not support recursive image navigation.

The USGS Global Visualization Viewer (GloVis) [16] is a quick and easy online search and order tool for selected satellite data. GloVis allows user-friendly access to all available images from the Advanced Spaceborn Thermal Emission and Reflection Radiometer ASTER TIR, ASTER VNIR, Landsat 7 (ETM+), and Landsat 4/5 (TM) sensors. Through a graphical user interface, the user can select any area of interest and quickly view all available images within the United States Geological Survey (USGS) inventory for the specified region. The main difference between TerraScope and GloVis is the distributed nature of the TerraScope database. GloVis GUI is Java based while TIN was developed using Macromedia Flash MX. GloVis does not support recursive image navigation.

NASA's Visible Earth System [17] is an image browser that provides a consistently updated, central point of access to the superset of NASA's Earth science-related images, animations, and data visualizations. The Visible Earth System is a web based image browser that does not have a graphical user interface and does not support recursive image navigation.

The Microsoft® Terraserver [15] is one the largest public repositories of high resolution aerial, satellite, and topographic imagery. Terraserver stores its data in a relational database system and makes it available via the Internet from virtually any graphical web browser. Users can zoom and pan across a mosaic of tiles. Users can locate imagery by clicking on a map, entering a city or town name in the "Search TerraServer" form, or entering a U.S. street address. Both Terraserver and TIN, allow similar user actions including submitting a query, and zooming and panning a particular image. While Terraserver was designed for a static set of images, TerraScope was designed to support continuous collections of image data. As a result, Terraserver can work on the assumption that there is a single image for each of the tiles.

Previous image retrieval/navigation systems described in the literature, including SAND [4], G-Portal [3], GeoWorlds [18], IPURE [5], WebSeek [13], ZUI[14] and others [1] [6] share some of the design goals with TIN but lack support for dynamically hyperlinked images and recursive navigation.

Remotely Image Navigator of the Virtually Hawaii Project [10] is a web based tool for finding remote sensing images of Hawaii with a range of spatial resolutions. These images come from a variety of sources, including instruments carried on aircraft, satellites and the Space Shuttle. The Remotely Image Navigator and TIN has several similarities. Both support recursive image navigation and provide automatically hyperlinked images. TIN initially was created to cover the Caribbean region but in a future we want extend it to cover all earth surface.

The Remotely Image Navigator does not integrate basic search and retrieval services.

Summarizing, TIN combines three features that distinguish it from other proposed interactive navigators for accessing online geographical image databases. First, **TIN** has the capability to dynamically generate hyperlinks by embedding automatically geo-referenced sub-images within the images being inspected. These hyperlinks are based on the results returned by queries, and these results may change dynamically as the distributed database changes. Second, **TIN** allows *recursive* navigation. Once the user selects an image she may recursively search the database for sub-images contained within the geospatial region covered by this base image. Third, **TIN** was completely developed on an interactive movie authoring environment (Flash MX) making it accessible from any web browser with a freely available Flash player installed. Flash players are available for virtually every major computing platform including portable devices (e.g. PDAs).

3. TIN Architecture and Implementation

As shown in Figure 2, the TerraScope system follows the typical client-server architecture. The client consists of a Flash MX interactive movie, running inside a web page that can be accessible using any web browser (item 1 in Figure 2). The server consists of a set of JAVA servlets modules running inside a web server (item 2 in Figure 2). These modules implement an abstraction of a single data repository by communicating with multiple TerraScope SRE's peers. The servlets also provide a mechanism for interfacing TerraScope with virtually any available database system. Our current prototype stores data in a POSTGRES [11] relational database.

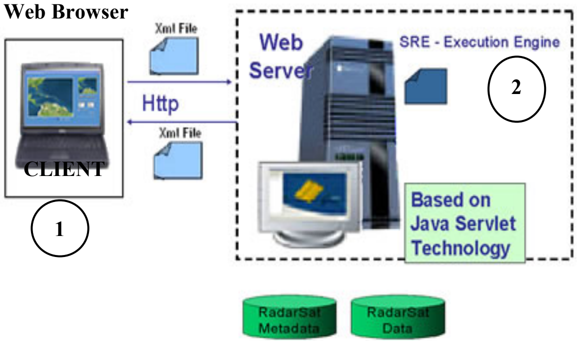


Figure 2 TerraScope Architecture

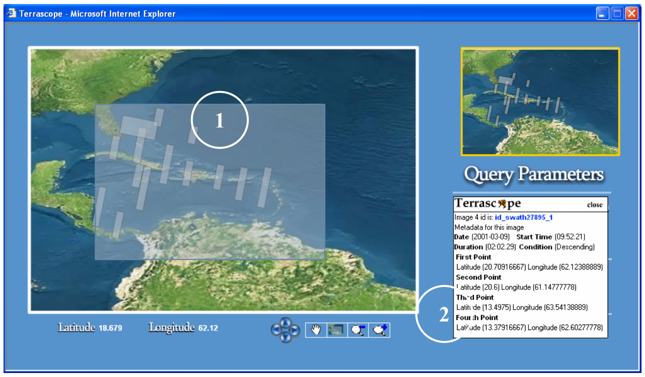
TIN users specify queries by manipulating familiar GUI widgets (e.g. pull-down menus, checkboxes, etc). TIN translates these queries into XML format before sending them to the SRE. The SRE computes the set of results by potentially forwarding queries to other SRE's believed to hold data pertaining to the query. The SRE then packages the results into an XML message similar to the one shown in Figure 3 and returns it to the client. The response includes a list of images and, for each image, the URL where the image is stored, condition, date, time, coordinates of the image, source and any other image specific metadata stored in the database.

```
<? Xml version="1.0"?>
<TerraScope>
 <CENSSIS>
 <id_swath22787_4>
   <bound>
              <latitud0>35.49194444</latitud0>
              <latitud1>35.61194444</latitud1>
              <latitud2>30.68166667</latitud2>
              <latitud3>30.80111111</latitud3>
              <longitud0>82.73694444</longitud0>
              <longitud1>81.62722222</longitud1>
              <longitud2>81.61777778</longitud2>
              <longitud3>80.56666667</longitud3>
  </bound>
  <condition>Ascending</condition>
  <duration_swath>01:21.43</duration_swath>
  <start_date>2000-03-16</start_date>
  <start_time>23:24:09</start_time>
<url>http://icarus.ece.uprm.edu/~TerraScope/images/22787_4.jpg</url>
</id_swath22787_4>
</CENSSIS>
</TerraScope>
```

Figure 3 Example TerraScope response in XML

The XML message in Figure 3 encodes information about one image retrieved from the CenSSIS repository. Images are clustered according to the repository that they originally came from. The user can restrict the set of repositories consulted using the GUI controls.

The client parses the XML message returned by the server and automatically geo-references the retrieved images into the previous displayed image using their corresponding geographical metadata and the metadata previously retrieved for the displayed image. The following section describes how the resulting set of images is visually presented to the user.



Terrascope close

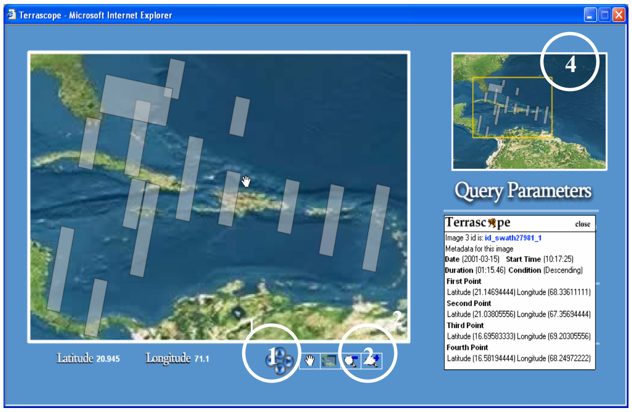
Image 4 id is: id_swath27895_1
Metadata for this image
Date (2001-03-09) **Start Time** (09:52:21)
Duration (02:02.29) **Condition** (Descending)
First Point
Latitude (20.70916667) Longitude (62.12388889)
Second Point
Latitude (20.6) Longitude (61.14777778)
Third Point
Latitude (13.4975) Longitude (63.54138889)
Fourth Point
Latitude (13.37916667) Longitude (62.60277778)

Figure 4. TIN display after submitting a query.

4. TIN Graphical User Interface

Figure 4 displays the set of images resulting from submitting the query specified in Figure 1. After parsing the XML result message file, TIN converts the geographical coordinates (latitude and longitude) of each image to pixel coordinates and embeds a polygonal outline of each image in the in the previously displayed image (item 1 in Figure 4). Each polygonal outline constitutes a spatial hyperlink that the user can click on to navigate into the geographical area covered by the corresponding sub-image. The set of such hyperlinks are creating dynamic and automatically computed according to the result set returned by the user query. The process requires no offline computation and no human pre-processing.

In addition to displaying the polygonal outline, for each retrieved sub-image, TIN displays a pop-up window with the metadata in a text box in the low right corner of the screen (item 2 in Figure 4). Using this pop-up window the user can examine any non-graphical information associated with each sub-image. The pop-up window appears whenever to mouse cursor is moved over one of the polygonal outlines.



Latitude 20.945 Longitude 71.1

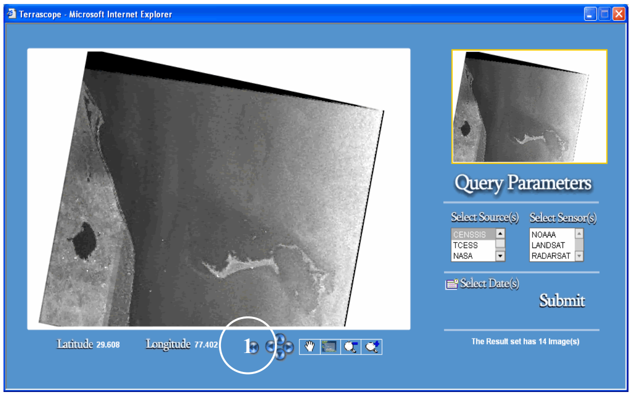
Image 3 id is: id_swath27981_1
Metadata for this image
Date (2001-03-15) **Start Time** (10:17:25)
Duration (01:15.46) **Condition** (Descending)
First Point
Latitude (21.14694444) Longitude (68.33611111)
Second Point
Latitude (21.03805556) Longitude (67.35694444)
Third Point
Latitude (16.69583333) Longitude (69.20305556)
Fourth Point
Latitude (16.58194444) Longitude (68.24972222)

Figure 5. The TIN periscope (item 4) outlines the area that the user is examining in the main window.

From the display in Figure 4 the user may proceed in one of three ways. She/He may submit a different query after realizing that the results were inadequate, he/she may use the GUI controls to explore the areas where retrieved images appear in more detail by zooming and panning, she may select one of the images in order to retrieve it, or she may click on one of the embedded polygons to navigate into the corresponding sub-image.

If a user chooses to explore the current image, she can use panning and zooming controls (items 1 and 2 in Figure 5) to move around the image or get closer to specific areas. The small window on the upper right corner continuously displays an outline (item 4 in Figure 5) indicating the area in the image currently displayed in the main window. The importance of this periscope window is greater when one explores images without landmarks and other aids to maintaining the user's geographically orientation.

As mentioned before, each polygonal outline serves as a hyperlink that the user can click in order to focus her search on a smaller sub-region. This facilitates exploration of smaller regions using higher resolution images whenever such images exist in the database. For instance, let say that upon reaching the point in the search depicted in Figure 4, the user wants to inspect one of the regions covered by one of the sub-images in more detail. To accomplish this, she only needs to click on the outline corresponding to the desired image. TIN responds by retrieving the sub-image from the database and displaying it on both the main window and the periscope window. From this point on, all queries submitted to the database are restricted, by default, by the geographical area covered by the new image. In other words, each browsed image provides a geospatial context from which further exploration may proceed. The user may also use the rubber band tool to indicate a different search context region.



Latitude 29.608 Longitude 77.402

The Result set has 14 Image(s)

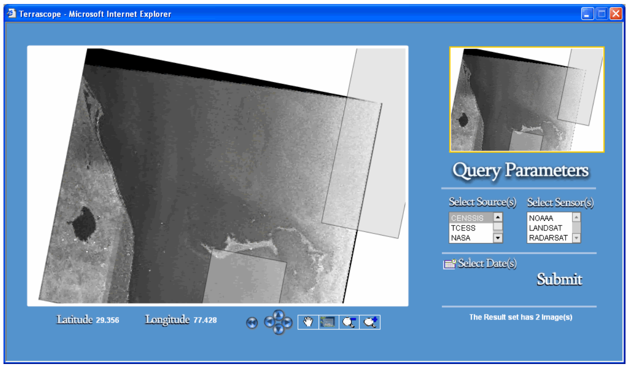
Figure 6. Retrieving a sub image by clicking on its outline.

Figure 6 shows the screen resulting from clicking on one of the polygonal outlines in Figure 4. The new image displays part of the Atlantic coast of the Florida peninsula. Upon examining the sub-image the user may decide that he/she wants to go back to the previous image. This can be accomplished by pressing the back button indicated by item 1 in Figure 6. This action will take the user to back to the screen in Figure 5.

At this time, the last selected sub-image becomes the new *base* image. Any GUI operation that could be performed on the initial image of the Caribbean region (Figure 1) can be performed on any sub-image. In fact, TIN executes exactly the same code in both cases. The initial image is simply a special bootstrapping image that is stored in the database. Future versions of TIN will allow users to configure the client for alternative bootstrapping images according to the geographical area covered by the available peer databases. In Terrascope, this set of databases may change dynamically [7].

The user can either manipulate the base image (e.g. zooming and panning) or conduct a new search for images contained within this base image.

Figure 7 shows the result of a recursive search constrained by the image shown in Figure 6. In the current version of TIN, searches are conducted using the "overlaps" operator available in the Postgres database system. This operator returns all images that overlap with the search region. This is a reason why the result of the search may return images that extend beyond the area covered by the base image. TIN allows users to recursively navigate the geographical space based on the imagery available in the database that satisfies the dynamic constraints placed by its users. To the best of our knowledge this combination of recursive navigation



The Result set has 2 Image(s)

Figure 7 Example of recursive search in TIN

within dynamic image content is a unique feature of TIN. We are currently conducting experiments to assess the usefulness and effectiveness of the feature. Some early results are presented in the next section.

5. Experimental Results

This section presents some measurements of the performance of TIN followed by results form a small pilot user study designed to evaluate its unique combination of features. The results presented below measure TIN's response time as the real time elapsed from the instant the user clicks the submit button to the instant when TIN finishes displaying the results set. This time can be divided in at least three segments: a client segment, a server segment and a communication segment. The client segment includes packing the GUI controls into an XML message, parsing the resulting XML message, and embedding the outlines into the base image. The server segment consist of translating the XML query into one or more SQL queries, submitting the SQL query to the peer Postgres databases, processing of the queries by Postgres, returning the result sets to the SRE, combining the result sets, and packaging the result set in a single XML message. Communication time is the time that the XML messages and SQL queries remain in transit in the network.

The database currently used by TIN includes two peer SRE's which collectively hold 80 images. This is a rather small database compared to what would be expected in real geo-spatial collections. Unfortunately, our experiments have been constrained by the unavailability of a large repository of images. An effort is currently undergoing to assess the performance of TIN on much larger databases.

The graph in Figure 8 shows system latency versus number of images retrieved by the query. The time measurements are taken with a completely offloaded SRE. Result set size varies from ten (10) to eighty (80) images. Each plotted time is averaged over several trials.

As expected, response time increases with result set size at what appears to be a super-linear rate. We are currently consulting the Terrascope SRE team for an explanation of this growth rate. The response time for result sets with less of 40 images are always less than 2 seconds. The total latency is dominated by server and communication time (97% for 80 images, see Figure 8). For all the different result set sizes, TIN time remains in the sub second range. Therefore, TIN by itself appears to run efficiently enough for practical use.

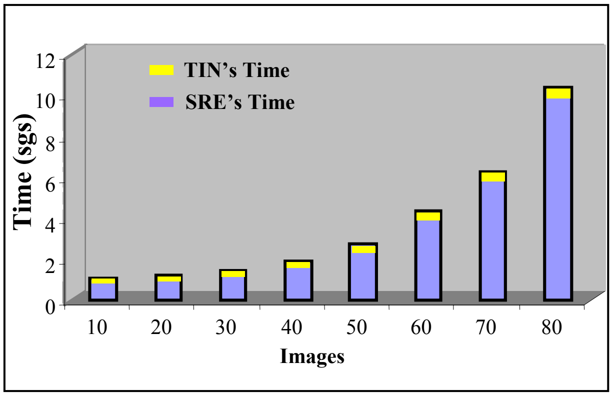


Figure 8. TIN's response time

TIN uses Action Script's [2] built-in methods and properties of the XML object to build, send, load, parse, and otherwise manipulate XML documents. These methods appear to be highly optimized as the latency of the system is dominated by server time.

Table 1: List of systems considered in the design of the BASE system

Name of Image Browser	Web Tool
Tcess Image Browser. Tropical Center for Earth and Space Studies	Java – Html
Quick Look Swath Browser. Canada Centre for Remote Sensing	Java – Html
Visible Earth. NASA	Html
GloVis. USGS Global Visualization Viewer	Java – Html
Virtually Hawaii	Html

5.1 Pilot User Study.

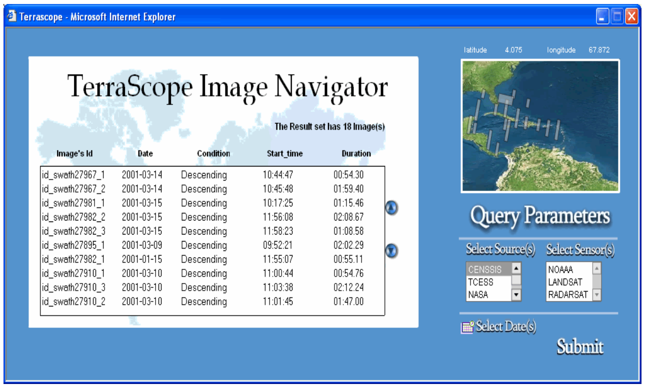
We performed a pilot user study to assess the browsing effectiveness of TIN relative to other existing image browsing/retrieval systems. We adopted the hypothesis that there would be not statistically significant differences in the time it takes to users to locate the targeted images, or in the number of incorrect selections made on a particular browser.

As a basis for comparison we developed a BASE prototype that included the main features of the list of currently available image retrieval systems shown in Table 1. Figure 9 (next page) shows a snapshot of the base system GUI. The main window now presents a list of images retrieved by the query. As the user scrolls this list, the corresponding image is highlighted in the small window on the upper right corner. Both TIN and the BASE access the same image database.

There were 20 participants involved in this experiment; most of them students at the University of Puerto Rico at Mayagüez, with various backgrounds including Computer Engineering and Electrical Engineering. Approximately

63% of the subjects were male and 37% of the subjects were female. Participants' ages were recorded using ranges so they would not feel uncomfortable disclosing their ages. From the data we collected, 79% of the participants were between the ages of 20 and 24, 21% between 25 and 31. 47% of subjects reported they were experts on the WWW and 47% of subjects reported they were intermediate users on the WWW, with the average user browsing 20 hours per week. Users also reported using a personal computer (PC) an average of 37 hours per week.

Each subject conducted two very similar tasks, one on each system. Figure 10 summarizes the user satisfaction scores for the three main questions formulated in the exit survey. These questions asked users to rate each system in a scale from one to five with respect to a particular aspect. For each question we conducted an Excel ANOVA single factor test to determine statistical significance.



The Result set has 18 Image(s)

Image's Id	Date	Condition	Start_time	Duration
id_swath27967_1	2001-03-14	Descending	10:44:47	00:54.30
id_swath27967_2	2001-03-14	Descending	10:45:48	01:59.40
id_swath27981_1	2001-03-15	Descending	10:17:25	01:15.46
id_swath27982_2	2001-03-15	Descending	11:56:08	02:08.67
id_swath27982_3	2001-03-15	Descending	11:58:23	01:08.58
id_swath27895_1	2001-03-09	Descending	09:52:21	02:02.29
id_swath27982_1	2001-01-15	Descending	11:55:07	00:55.11
id_swath27910_1	2001-03-10	Descending	11:00:44	00:54.76
id_swath27910_3	2001-03-10	Descending	11:03:38	02:12.24
id_swath27910_2	2001-03-10	Descending	11:01:45	01:47.00

Figure 9. Snapshot of the BASE Prototype.

Series 1 plots user satisfaction with respect to the way that each system shows the image textual data and metadata. In this case BASE had a slightly higher score but the difference was not statistically significant (ANOVA single factor). Series 2 represents user satisfaction with respect to the way that both systems allow access and manipulation of the retrieved images. This time TIN had the highest user satisfaction and this time the difference was statistically significant (p<.05).

Series 3 represents the user satisfaction with respect to the way that systems combined techniques to show the images, image data and metadata to the users; in this case TIN had the highest user satisfaction. The ANOVA test showed a p-value of .08 meaning that the difference was not statistically significant.

The users also evaluated the usability of the zooming and panning tools, the back button and the recursive queries used in the TIN prototype. All users considered panning and zooming very useful for inspecting and manipulating images with more detail. Also, all users expressed that the back button and the recursive query feature would be useful tools in image navigators to search, inspect and manipulate image of higher resolution.

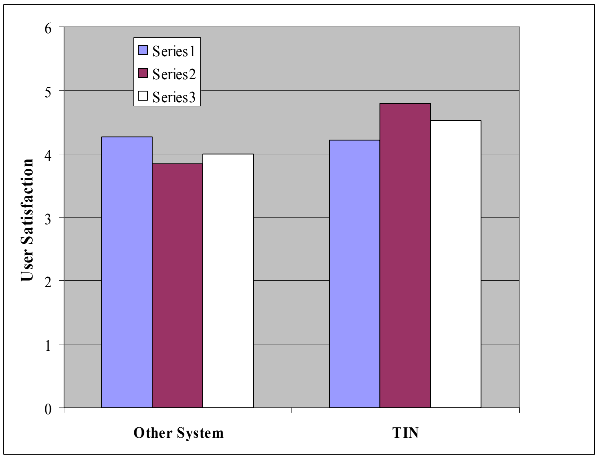


Figure 10. User satisfaction ratings the systems.

6. Lessons from using Flash MX

Flash was originally conceived as an interactive movie authoring tool for dynamic web content development. Some research projects have begun to use Flash, and its scripting language ActionScript, as an application development tool. For instance, Halo [9] implements a technique for visualization of off-screen locations; in this case they designed and implemented a GUI using Flash to test these techniques in a Compaq iPAQ Pocket PC. Some of the advantages of Flash include platform independence, advanced tools for manipulating interactive graphics and animation, and a comprehensive scripting language with many built-in functions for implementing distributed applications over the Internet.

We also experienced some not so positive consequences of the decision to use Flash. Perhaps the main inconvenience resulted from the difficulty of finding the scripting code associated with the many objects comprising the interactive movie. We were forced to resort to Ad hoc solutions that essentially concentrated the code in file modules. Each object action will consist of a function call to the corresponding module. However, this solution caused a significant increase on the edit-compile-execute cycle, since it added more steps to the process we order to load the code into the Flash MX development environment.

Nevertheless, the leverage provided by the Flash MX environment permitted the development of a fully working prototype in less than six (6) man-months. We believe that this time would have dramatically increased had we chosen to implement TIN with the same developers, but with other dynamic content development tools for the web like Java or C#. Our experience developing TIN proves that feature rich and robust client applications can be development using Flash with much less effort an in less time than with more traditional Web development tools. We have learned that Flash MX is a vehicle for delivering non-traditional content and information, within a player that is small enough to fit into PDAs.

7. Conclusions and Future Work

We have presented the design of **TIN**, a novel image navigator for the **TerraScope** earth science data middleware system under development by the Advanced Data Management Research Group at the University of Puerto Rico Mayaguez. **TIN** is designed with the capability to dynamically generating spatial *hyperlinks* connecting sub-images with the images being inspected with no human intervention. Users can geographically constrain their searches to a specific region using visual GUI tools. TIN presents query results graphically by automatically embedding polygonal outlines of retrieved images onto any parent retrieved image.

TIN supports recursive navigation. When the user selects an image he/she may recursively search the database for sub-images contained within the geospatial region covered by this base image. Each retrieved sub-image is dynamically geo-referenced in the base image and becomes a hyperlink that can be used to navigate into smaller geographical regions with higher resolution. TIN was developed with Macromedia Flash MX, Action Script and uses XML to communicate queries and data to and from potentially many content providers. Macromedia Flash MX has resulted in a faster way to create multimedia rich platform-independent Internet applications.

Future efforts of the TerraScope/TIN project will focus on:

- Developing alternative advanced query languages supporting dynamic data composition in the spirit of the image cutters tools proposed in [8]
- Exploring alternative ways to reduce or avoid image clustering caused by periodic capture of images covering similar geographical areas.

8. Acknowledgements

TerraScope was born from a cooperative effort between members of the ADM Group and members of the Earth Science community at UPRM. We obtained seed funding to start ***TerraScope*** thanks to a 1-year combined grant from the NASA Tropical Center for Earth and Space Studies (TCESS) and the NSF Center for Subsurface Sensing and Imaging Systems (CenSSIS).

References

[1] C. Chan, T. Chan, and E. Prakash, A System for Geographical and Spatial Data Exploration on the Internet: Proceeding Series, ACM International Conference, Selected papers from the Pan-Sydney workshop on Visualization - Volume 2. Sydney, Australia 2002.

[2] C. Moock. ActionScript for Flash MX: The Definitive Guide, Second Edition. O'Reilly & Associates, Inc. December 2002.

[3] E. Lim, Goh, D., et al., G-Portal: A Map-based Digital Library for Distributed Geospatial and Georeferenced Resources, Proceeding of the second ACM/IEEE-CS joint conference, International Conference on Digital Libraries. Portland, Oregon, United States, 2002.

[4] E. Tanin, F. Brabec, and H. Samet, Remote Access to large Spatial Databases, Proceedings of the tenth ACM international symposium on Advances in geographic information systems. McLean, Virginia, United States, 2002.

[5] G. Aggarwall, T. V. Ashwin, and S. Ghosal, An Image Retrieval System with Automatic Query Modification: Multimedia, IEEE Transactions Volume: 4, Iss.2 on page(s): 201-214. June 2002.

[6] G. M. Lohman, et al., Remotely-Sensed Geophysical Databases: Experience and Implications for Generalized DBMS, Proceedings of the 1983 ACM SIGMOD International conference on Management of data, San Jose, California, United States, 1983.

[7] M. Rodríguez and E. Coronado, SRE Search and Retrieval Engine of TerraScope Earth Science Information System, Proceedings of IASTED International Conference, Computer Science and Technology, Cancun, Mexico, 2003

[8] M. Rodríguez, P. Rivera, A. Alvear, et al. Database Middleware System Supporting Remote Sensing Analysis over Distributed Sources, Proceedings of IASTED International Conference, Computer Science and Technology, Cancun, Mexico, 2003.

[9] P. Baudisch, and R. Rosenholtz, Halo, a technique for Visualizing Off-Screen Locations, Proceedings of Conference on Human Factors and Computing Systems, Ft. Lauderdale, Florida, United States, 2003.

[10] P. Mouginis-Mark, L. Glaze, P. Flament. Virtually Hawaii by University of Hawaii. **http://satftp.soest.hawaii.edu/space/hawaii/**

[11] PostgreSQL Technical Documentation. **http://techdocs.postgresql.org/**

[12] Quicklook Swath Browser: A web-based tool host by The Canada Centre for Remote Sensing. **http://quicklook.ccrs.nrcan.gc.ca/ql2/en?action=search**

[13] S. Chang, J.R. Smith, M. Beigi, and A. Benitez, Visual Information Retrieval from Large Distributed Online Repositories, Communications of the ACM Volume 40, Issue 12, 1997.

[14] T.A. Combs, and B. Bederson, Does Zooming Improve Image Browsing?, Proceedings of the fourth ACM Conference, International Conference on Digital Libraries, Berkeley, California, United States. 1999.

[15] T. Barclay, J. Gray, and D. Slutz, Microsoft TerraServer: A Spatial Data Warehouse, Proceedings of International Conference on Management of Data and Symposium on Principles of Database Systems, ACM SIGMOD, 2000.

[16] U.S. Geological Survey (USGS) Landsat 7 Image Viewer: **http://glovis.usgs.gov/**

[17] Visible Earth: a searchable directory of images, visualizations, and animations of the earth. **http://visibleearth.nasa.gov/.**

[18] V. Kumar, A. Bugacov, M. Coutinho, and R. Neches, Integrating Geographic Information Systems, Spatial Digital Libraries and Information Spaces for conducting Humanitarian Assistance and Disaster Relief Operations in Urban Environments, Proceedings of the seventh ACM international symposium on Advances in geographic information systems. Kansas City, Missouri, United States 1999.

[19] XML, From the Inside Out. **http://www.xml.com**