TIC: TERRASCOPE IMAGE CLUSTERING

APPLYING CLUSTERING TECHNIQUES TO IMAGE AGGLOMERATION IN IMAGE RETRIEVAL SYSTEMS

By
Lizvette Malavé

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
In
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2005

Approved by:

Manuel Rodríguez Martínez, Ph.D. Member, Graduate Committee	Date
Pedro I. Rivera Vega, Ph.D. Member, Graduate Committee	Date
Bienvenido Velez, Ph.D. President, Graduate Committee	Date
Silvestre Colón, M.S. Representative of Graduate Studies	Date
Isidoro Couvertier, Ph.D. Chairperson of the Department	Date

Abstract

In this thesis we describe the design of the Image Clustering features associated with 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 through an integrated portal. Individual Terrascope repositories are designed to locally store images retrieved by satellites from the geographical regions accessible to a particular receiving station. Frequent collection of images from the same geographical region yields images that have a high probability of overlapping with one another making the task of manually browsing through the image database difficult.

TerraScope Image **C**lustering (**TIC**) attempts to provide an effective way of managing highly overlapping image data by organizing a relational query result set into a tree of image clusters. Each node in the tree represents a cluster or subset of the parent node with less image overlap. Users browse the result set by traversing the cluster tree guided by meaningful labels associated with the clusters. Each cluster in the tree can be recursively organized into finer clusters according to multiple criteria such as collection date, or sensor type. The cluster tree can be changed dynamically and recursively in order to capture the clustering that better suits the user's information need. The current version of TIC supports several clustering algorithms and was implemented using an interactive movie authoring environment (Flash MX) and XML (e**X**tensible **M**arkup **L**anguage).

Our experiments prove that TIC improves the effectiveness of manipulation and image navigation by providing an organized and structured display of the retrieved images.

Resumen

En esta tesis describimos el diseño de las características de agrupamiento de imagenes asociadas al sistema de Terrascope. TerraScope es un sistema intermedio que fue diseñado para facilitar la colaboración entre sistemas de depósitos de datos geológicos que desean proporcionar sus datos geo-espaciales a través de un portal integrado. Los depósitos individuales de Terrascope se diseñan localmente para almacenar las imágenes tomadas por los satélites de las regiones geográficas accesibles a una estación de recepción particular. La colección frecuente de imágenes de una misma región geográfica aumenta la probabilidad de solape de una imagen contra haciendo la tarea de navegarlas mas difícil.

"TerraScope Image Clustering" (TIC) procura proporcionar una manera eficaz de manejar datos altamente solapados de las imágenes organizando los resultados de la base de datos en un árbol de imágenes. Cada nodo en el árbol representa un sub-grupo o un subconjunto del nodo del padre con menos solape en las imágenes. Los usuarios buscan en el resultado atravesando el árbol dirigidos por las etiquetas significativas asociadas a los nodos. Cada nodo en el árbol se puede organizar recurrentemente en sub-nodos según criterios múltiples tales como fecha de la colección, o tipo del censor. El árbol puede ser cambiado dinámicamente y recurrentemente agrupar según satisfaga la necesidad de información del usuario. La versión actual de TIC apoya varios algoritmos que clasificación y agrupación y fue implementada usando un ambiente de ejecución de películas interactivas (Flash MX) y de XML (Extensible Markup Language).

Nuestros experimentos prueban que TIC mejora la efectividad de la manipulación y navegación de imágenes, desplegando las imágenes en forma organizada y estructurada.

Acknowledgements

I wish to thank my advisor Dr. Bienvenido Vélez for his valuable contribution to this work.

I also wish to thank Dr. Manuel Rodriguez, Dr. Pedro I. Rivera Vega, and Prof. Amaury Cabarcas Alvarez.

I would like to thank the Department of Electrical and Computer Engineering at University of Puerto Rico-Mayagüez, the **NSF Cen**ter for **S**ubsurface **S**ensing and **I**maging **S**ystems (**CenSSIS**) for their financial support and the **T**ropical **C**enter for **E**arth and **S**pace **S**tudies (TCESS).

Finally, I wish to thank my family: my mother Vilma I. Ramos Medina, my father José G. Malavé Torres, and my sister Ailette I. Malavé Ramos for their love and encouragement.

Table of Contents

List of Figures

Chapter One: Introduction

1.1 The Need for TIC

Frequently, scientists around the world need to effectively access information necessary for research. Data communication and sharing among scientists is an everyday task and the internet provides the best way to share these resources.

Many satellite data collector centers are distributed around the world. They provide web access to their databases of satellite images and other geographic useful for the earth. ***Terrascope*** is a new system developed at the University of Puerto Rico-Mayagüez Campus to provide effective and transparent access to these inherently distributed images.

Terrascope follows the client server architecture depicted in Figure 1. The server module is called the ***Search and Retrieval Engine (SRE*)** **[1]** and the client module is called the ***Terrascope Image Navigator (TIN)*** **[2]**. The server (SRE) consists of a set of JAVA servlets modules running inside a web server. These modules implement an abstraction of a single data repository by communicating with multiple TerraScope SRE's peers. The SRE computes the set of results by potentially forwarding queries to other SRE's believed to hold data pertaining to the query specified by the user. Hence, SREs act as servers to TIN clients, and also as clients of other SREs. This type of architecture is often called peer-to-peer. For more details of the SRE and TIN, the reader is referred to [1] and to [2].

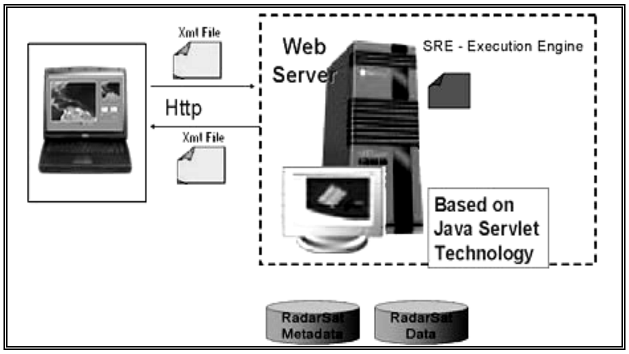
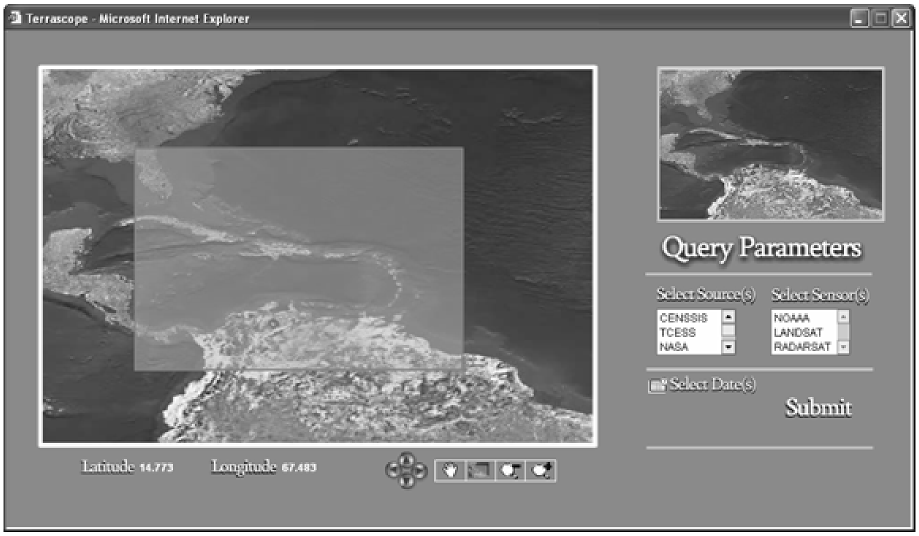


Figure 1 Terrascope Architecture

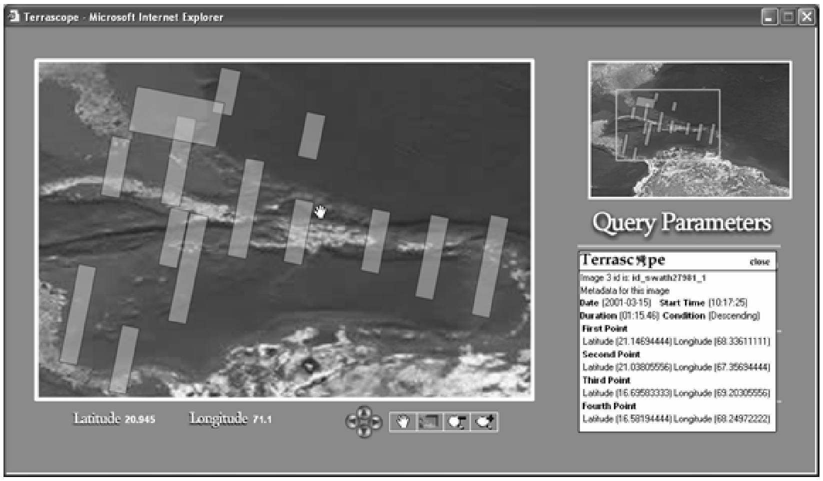
The **TerraScope I**mage **N**avigator (**TIN)** provides the end user graphical interface to access the collection of SREs. It is an image browser that provides ubiquitous and efficient access to distributed information. The **TIN** prototype delivers satellite images with their corresponding metadata, GIS characteristics, and other information to any web browser with a Flash MX player installed. Initially, TIN displays a map showing the overall geographical region covered by the satellite ground stations contributing data to the distributed repository (see Figure 2). The current prototype includes data collected by the Tropical Center for Earth and Space Studies (TCESS), the Center for Subsurface Sensing and Imaging Systems (CenSSIS) both at the University of Puerto Rico Mayaguez Campus and data collected by the Aster sensor from the Southern Urals Region of Rusia. 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) and data collection date.



Latitude 14.773 Longitude 67.483

Figure 2 TIN’s Initial Window

TIN displays polygons of the images found in the database corresponding to the user query and automatically geo-references these images within the query image (see Figure 3). The user can click on the geo representation he/she is interested in and the image will be displayed in the main window. Once an image of interest is selected and displayed, users may easily search the database for images contained within this image (sub-images) or for images overlapping it (see Figure 4). In other words, each browsed image provides a geospatial context from which future exploration may proceed; a feature that we call recursive navigation.

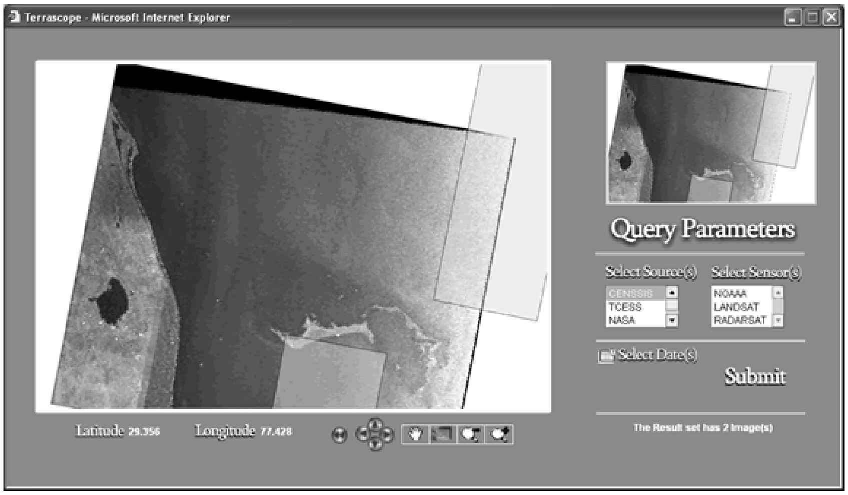


Latitude 20.945 Longitude 71.1

Terrasc pe

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 3 Geo Representation of images retrieved by TIN in response to a query



Latitude 29.356 Longitude 77.428

Select Source(s): CENSSIS, TCESS, NASA

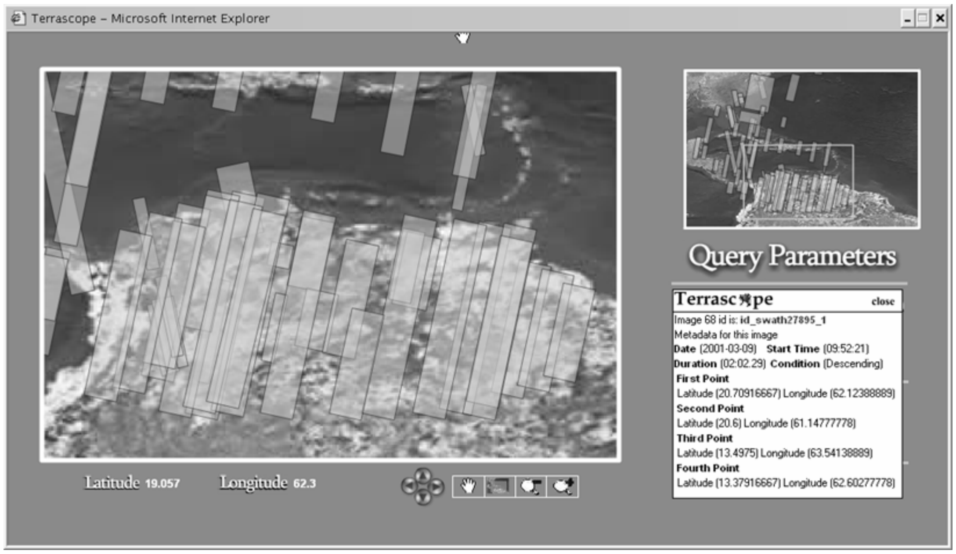
Select Sensor(s): NOAAA, LANDSAT, RADARSAT

The Result set has 2 Image(s)

Figure 4 Recursive Navigation Example in TIN

Previous user studies have evidenced that geo-representation of images helps users visualize the location of images and provide an effective way to present this kind of data. However, when the database consists of multiple highly overlapping images, geo representation of images may worsen the ability of users to find images.

The databases that Terrascope is intended to support contain many images that overlap one another because the satellites that recollect the images are continuously rounding the same geospatial region. The system must support temporal queries across all such overlapping images. As shown in Figure 5, the problem is that the user is unable to access all the data requested when the images overlap and can not request to download the images hidden by other images. In order to access the data the user may be forced to make different queries to retrieve less information per query. Coming up with suitable queries to get rid of unwanted information can be often a tedious and time consuming process.



Latitude 19.057 Longitude 62.3

Image 68 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 5 Example of Image Overlapping in TIN

Another problem confronted by the user is that properties of data cannot be effectively visualized in the result set. For instance, the results in Figure 5 come from different sources and sensors, but this is not easily identified by just looking at the geo-referenced polygons.

TIC exploits image clustering in order to automatically classify the retrieved images into smaller groups and present them to the user in a way that facilitates browsing and finding images of interest.

1.2 Research Objectives

The central hypothesis of our work is that image clustering can be used to improve the effectiveness of image retrieval/browsing systems with similar characteristics to TIN. We test our hypothesis by conducting two types of experiments. The first one is a comparison of several image clustering algorithms and the second is a user study. Our experimental results provide evidence in favor of the effectiveness of TIC to improve the presentation of results and the ability of the users to visualize the properties of data.

1.3 Summary of Contributions

In summary, this research makes the following contributions:

Design one or more algorithms to classify satellite images in order to effectively support browsing.

Redesign the TIN GUI to support effective image clustering-based browsing and searching.

Design various alternative image clustering algorithms

Measure the performance of the various clustering algorithms.

Measure the effectiveness of the new Graphical User Interface (GUI).

1.4 Structure of the Thesis

The reminder of this thesis is organized as follows: Chapter 2 discusses work related to this research; the approach we used to developed the system is explained in Chapter 3; Chapter 4 shows the graphical user interface and the image clustering process in more detail; Chapter 5 discusses the development details of the image clustering algorithms; Chapters 6 shows our experimental methodology and results; and finally, Chapter 7 presents our conclusions and suggests some areas for future work.

Chapter Two: Related Work

Many scientists, research centers and universities have been developing systems to share their image data over the Internet. Some of these systems make geo-representation as TIN and also have classification methods of the results. But most of these systems use static databases or are intended to share the most recently available image over the area. In this section we present some of these systems emphasizing on the way they manage agglomeration of images as well as other factors that affects the design of our Terrascope Image Clustering (TIC) system.

The **Quicklook Swath Browser** is an image browser developed by the Canada Centre for Remote Sensing and tries to bring a solution to this critical issue [5]. This research has some similarities with TIN, both create geo-representation of the images, and use the same mechanism to display the data and metadata to the users, combined text and graphics and use a graphical user interface accessible through web, also both systems provide automatically hyperlinked images. This system does not support the recursive navigation of TIN and also does not support temporal queries; it only keeps the latest images taken over the area. This reduces the quantity of images in the database and result in no overlapping or conglomeration of results.

Remotely Image Navigator of Virtually Hawaii Project [6] 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 supports the recursive image navigation into smaller geographical area where the user may recursively search the database for sub-images contained within the geospatial region covered by this base image. They also provide automatically hyperlinked images. The main difference with TIN consists that it does not have the flexibility of making temporal, or sensor based queries. Instead, Remotely Image Navigator statically predefines a result set of images to be displayed in recursive navigation. This method results in no overlapping or conglomeration of data.

The **USGS Global Visualization Viewer (GloVis)** [7] is a Java based 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 latest images taken over the area. The user can search images of one sensor (ex. VNIR, Landsat 7) at a time. After the result images are presented, the user can navigate through previous images taken over the area, selecting the dates he/she is interested in. For each of the navigations, the Java applet makes a request to their inventory for the images corresponding to the selected area. The main difference between GloVis and TIN is that the later supports requests of different sensors, sources and ranges of dates at the same time. TIN and Terrascope Image Clustering (TIC) presents the available images all at once to the user. Using TIC the user can identify easily the image he/she wants to look.

The **Microsoft® Terraserver** [8] 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. TerraServer contains 3.3 tera-bytes of high resolution United States Geological Survey (USGS) aerial imagery and USGS topographic maps. 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. This research has some similarities with **TIN**. For instance, 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 collection of image data. Terraserver manages the agglomeration and overlapping of images with its program TerraCutter. TerraCutter breaks an image into tiles that then are mosaic into a scene. This computation is feasible for Terraserver because it is done on static data, but it is too much computation to be done dynamically by TIC.

G-Portal is a web portal providing digital library services over geospatial and geo-referenced content found on the World Wide Web (WWW) [9]. **G-Portal** adopts a map-based user interface to visualize and manipulate the distributed geospatial and geo-referenced content. The principal aim of this project is the identification, classification and organization of geospatial and geo-referenced resources on the web, and the provision of digital library services (e.g. searching, visualization) for these types of resources. G-Portal and TIN provide basic search and retrieval services of geospatial data over the Internet; both systems make a geographical representation for each geospatial object. G-Portal uses logical grouping (layers) based in some characteristics of the data (city, rivers, land). This logical grouping is important when the visualization of images becomes large. However, **G-Portal** does the classification using previously manually defined classification rules. These classification rules are predefined in a schema file that specifies the classification attributes. The user must have a detailed knowledge of the database schema to create the rules. TIC is intended to do dynamic categorization and easily adapt to different user needs.

The TIN system borrows some of it automatic clustering ideas from the Inforadar System [10] developed at MIT. Inforadar automatically clusters text documents, and selects categories to organize the result sets. TIN applies some of the same ideas for the realm of satellite imagery.

Some of the systems discussed above exploit some level of static data classification on data, for that reason they are not feasible to work with Terrascope. Other solutions do not support object relational queries, recursive navigation, or the other features that TIN supports. However, object relational queries (ex. temporal, sensor) are an important feature for data comparison and can provide effective access to data. TIC is intended to group data dynamically and recursively and will be completely developed on an interactive movie authoring environment (Flash MX). This makes it accessible via any web browser with a freely available Flash player installed. Such players are available for virtually every major computing platform including portable devices (e.g. PDAs).

Chapter Three: Approach

The design of the ***Terrascope Image Clustering*** is driven by the following goals:

Design various image clustering algorithms.

Redesign the TIN GUI to support effective image clustering-based browsing and searching.

Design and measure the performance of the various clustering algorithms.

Measure the effectiveness of the new Graphical User Interface (GUI).

TIC provides an effective and structured view of the result set. It is developed as a tree that represents the different characteristics of the result set. Each node in the tree represents a cluster or subset of the parent node with less image overlap. TIC provides a graphical user interface that enables effective browsing of highly overlapping image data by organizing a relational query result set into a tree of image clusters.

TIC provides several options to categorize the data by source, sensor, date or minimum overlapping. The user may select any of the options to cluster the result set. Each sub set will become a node in the tree with less image overlap. Users browse the result set by traversing the cluster tree guided by meaningful labels associated with the clusters. Each cluster in the tree can be recursively organized into finer clusters.

The cluster tree can be changed dynamically and recursively in order to capture the clustering that better suits the user’s information need. The current version of TIC supports several clustering algorithms and was implemented using an interactive movie authoring environment (Flash MX) and XML (e**X**tensible **M**arkup **L**anguage).

Chapter Four: A Graphical User Interface to Support Image Browsing

The clustering provided by TIC maximizes the benefits of TIN's image navigation. Initially, TIC displays a map from where the user may select the search area (see Figure 6). The user may use the GUI controls to explore the areas with zooming and panning, or he can use the controls to make queries.

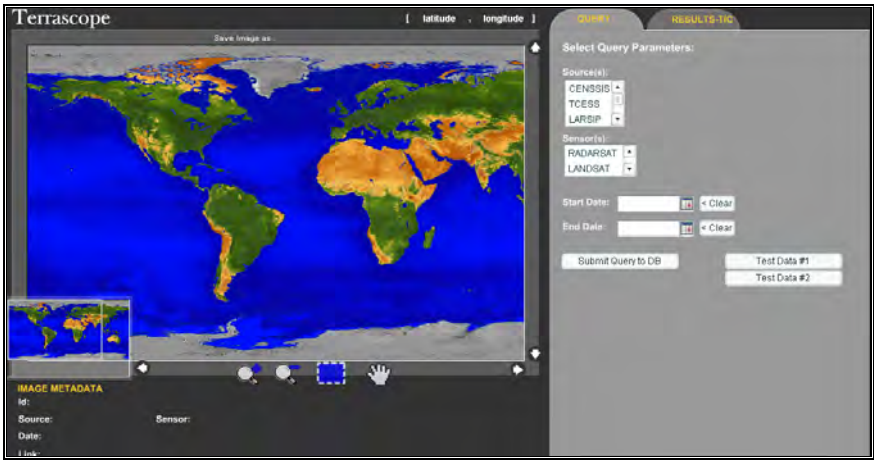


Figure 6 Initial Context Area

The user can make restrict the scope of the query by selecting the area with a rubber band tool. The query can be modified selecting the sensors or sources from where the user needs the data. Also, the user can do temporal queries selecting a specific date or range of dates (see Figure 7)

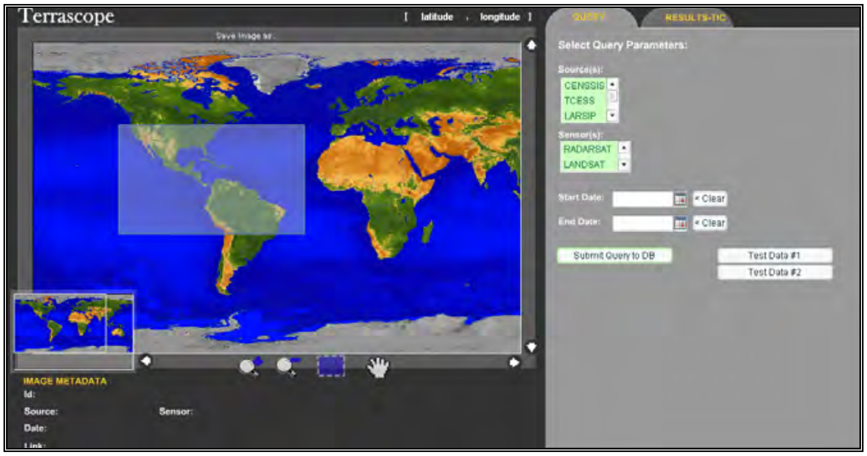
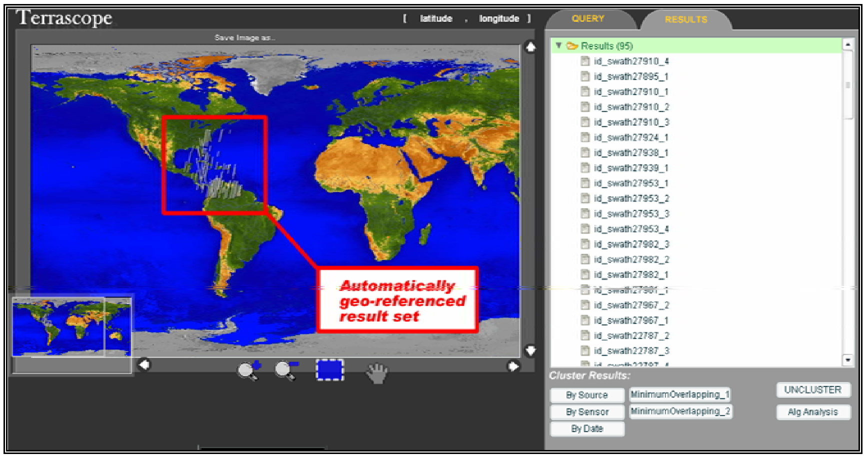


Figure 7 Selecting a search area with the rubber band tool

After submitting the query to the server, TIC parses the XML result message and automatically geo-references the retrieved images into the previous displayed image using their corresponding geographical metadata (see Figure 8). In addition to the visual representation of the result set, a text representation is displayed on a separate panel. Each image in the result set is identified by a unique id number that is extracted from the image metadata. Note that the result set on Figure 8 only contains images which overlap with the selected search area.



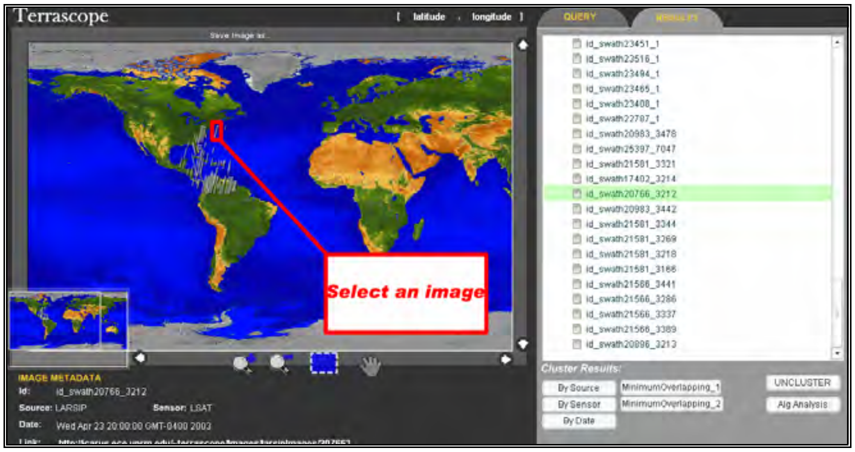
Automatically geo-referenced result set

Results (95)
- id_swath27910_4
- id_swath27895_1
- id_swath27910_1
- id_swath27910_2
- id_swath27910_3
- id_swath27924_1
- id_swath27938_1
- id_swath27939_1
- id_swath27953_1
- id_swath27953_2
- id_swath27953_3
- id_swath27953_4
- id_swath27982_3
- id_swath27982_2
- id_swath27982_1
- id_swath27981_1
- id_swath27967_2
- id_swath27967_1
- id_swath22787_2
- id_swath22787_3
- id_swath22787_4

Figure 8 Geo-referenced and Textual Result Set

The geo represented polygons are rendered as buttons. The buttons and the different panels are synchronized. If the user rolls over a polygon it changes colors and metadata of the corresponding image is displayed on the lower panel. If the user cannot rollover an image because it has other images that overlap, the user may select an image from the list, the system responds by highlighting the geo-representation and displaying its metadata.

From the display in Figure 8 the user may proceed in one of three ways. He/She may submit a different query after realizing that the results were inadequate, he/she may use the GUI controls to explore the areas with zooming and panning, or he/she may click on one of the embedded polygons in order to download the image and navigate into it (recursive navigation) (see Figure 9).



id_swath23451_1
id_swath23516_1
id_swath23494_1
id_swath23465_1
id_swath23408_1
id_swath22787_1
id_swath20983_3478
id_swath25397_7047
id_swath21581_3321
id_swath17402_3214
id_swath20766_3212
id_swath20983_3442
id_swath21581_3344
id_swath21581_3269
id_swath21581_3218
id_swath21581_3166
id_swath21566_3441
id_swath21566_3286
id_swath21566_3337
id_swath21566_3389
id_swath20896_3213

IMAGE METADATA
Id: id_swath20766_3212
Source: LARSIP Sensor: LSAT
Date: Wed Apr 23 20:00:00 GMT-0400 2003
Link: http://icarus.ece.uprm.edu/~terrascope/images/larsipimages/20766]

Figure 9 Image Selection

Recursive navigation facilitates the exploration of smaller regions using higher resolution images. Once an image of interest is downloaded, the user can explore the area by using zooming and panning tools or can save the image into their hard drive. The user can also navigate into the area using GUI controls to make another query. The selected image becomes the new geographical context and the query is automatically restricted to this area. The query can be further refined using the controls, searching by source, sensor, and/or dates.

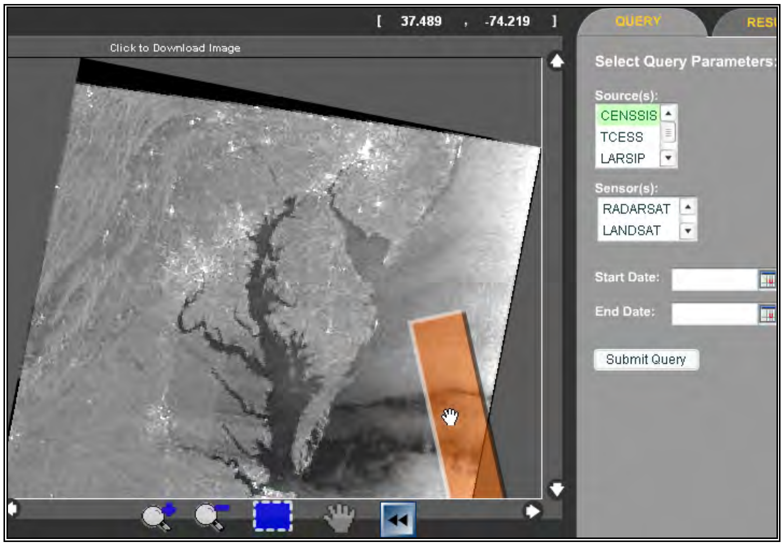


Figure 10 Recursive Navigation

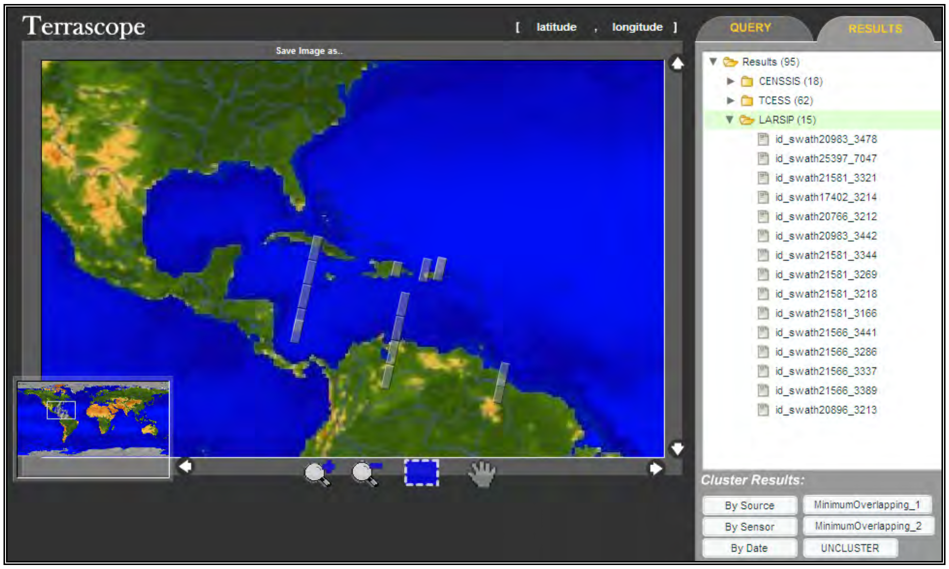
The geo representation is a graphical way to work with some of the characteristics (latitude and longitude points) included in the image metadata but the images have other characteristics such as source, or satellite type that are not easily target by just looking at the result set. Using TIC the user can have a graphical and structured view of all the data by making use of the clustering options that take advantage of the images metadata. This process requires no interaction with the server and no human pre-processing.

TIC's user interface provides several options to categorize data. When the user categorizes images, the list of the result set becomes a tree. Each node gathers images with similar characteristics according to the user request. The following are the options currently supported by TIN to cluster the data:

- By Source: TIC clusters the result set according to the source database that holds the image. Each node corresponds to a source or database that contributes data to the repository. In the example presented in Figure 11 the images are classified by source. The sources that currently contribute data are CENSSIS, TCESS, and LARSIP. The user can select the node labeled LARSIP to display only the images held by this source.
- By Sensor: TIC clusters the result set according to the kind of sensor that was used to collect the image. In the example presented in Figure 12 the images are classified by sensor. The images where taken using the sensors: RSAT and LSAT. For instance, the user can select the node labeled LSAT to display only the images within the result set taken by this type of satellite sensor.
- By Date: TIC clusters the result set by date for the first one to the latest one. Each node label identifies the date of the images contained in the node. In the example presented in Figure 13 the images are classified by date. The user can select the node labeled Sat April 15, 2000 to display only the four images taken that day.
- By Minimum Overlapping: TIC clusters the result set into approximately ten (10) nodes in a way that minimizes overlapping inside each node.

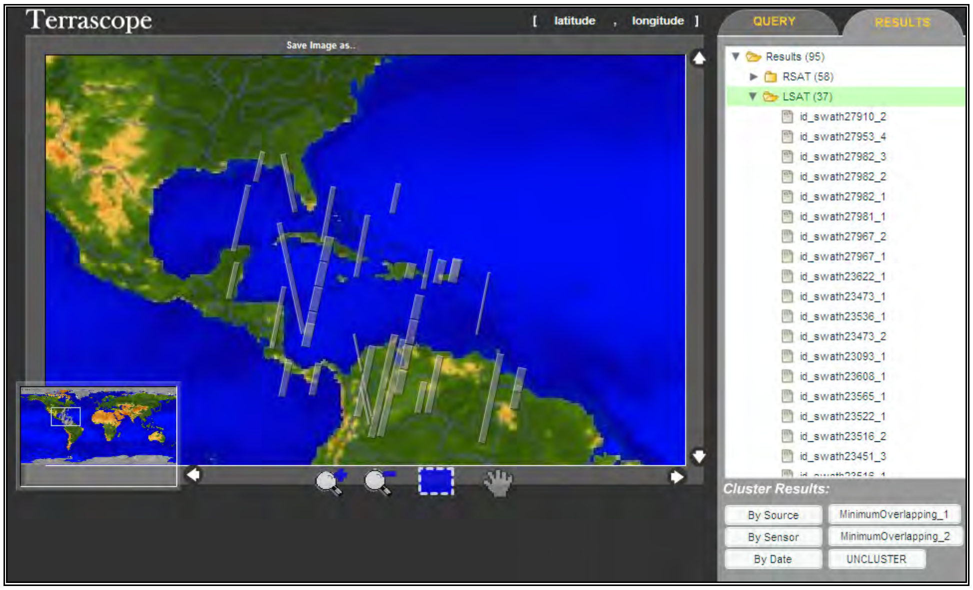
Each branch's label contains the quantity of leafs that it contains.

All clustering options allow users to conveniently expand or collapse nodes to facilitate focusing on the information of interest. Also, at any time the user can click the button labeled UNCLUSTER to start over with the result set list as seen in Figure 8.



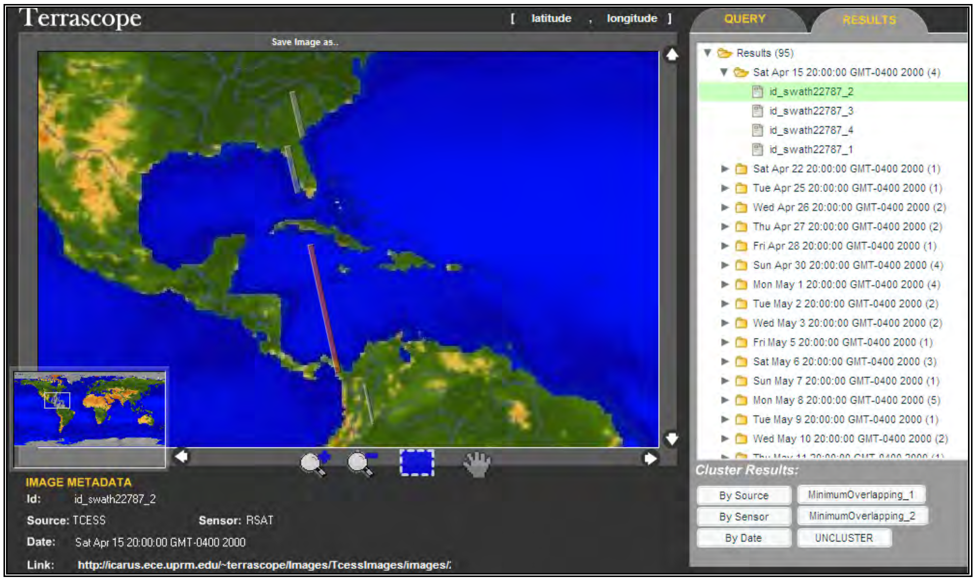
```
▼ Results (95)
    ▶ CENSSIS (18)
    ▶ TCESS (62)
    ▼ LARSIP (15)
        id_swath20983_3478
        id_swath25397_7047
        id_swath21581_3321
        id_swath17402_3214
        id_swath20766_3212
        id_swath20983_3442
        id_swath21581_3344
        id_swath21581_3269
        id_swath21581_3218
        id_swath21581_3166
        id_swath21566_3441
        id_swath21566_3286
        id_swath21566_3337
        id_swath21566_3389
        id_swath20896_3213
```

Figure 11 Image Clustering by Source



Results (95)
RSAT (58)
LSAT (37)
id_swath27910_2
id_swath27953_4
id_swath27982_3
id_swath27982_2
id_swath27982_1
id_swath27981_1
id_swath27967_2
id_swath27967_1
id_swath23622_1
id_swath23473_1
id_swath23536_1
id_swath23473_2
id_swath23093_1
id_swath23608_1
id_swath23565_1
id_swath23522_1
id_swath23516_2
id_swath23451_3
id_swath23516_1

Figure 12 Image Clustering by Sensor



Results (95)
Sat Apr 15 20:00:00 GMT-0400 2000 (4)
id_swath22787_2
id_swath22787_3
id_swath22787_4
id_swath22787_1
Sat Apr 22 20:00:00 GMT-0400 2000 (1)
Tue Apr 25 20:00:00 GMT-0400 2000 (1)
Wed Apr 26 20:00:00 GMT-0400 2000 (2)
Thu Apr 27 20:00:00 GMT-0400 2000 (2)
Fri Apr 28 20:00:00 GMT-0400 2000 (1)
Sun Apr 30 20:00:00 GMT-0400 2000 (4)
Mon May 1 20:00:00 GMT-0400 2000 (4)
Tue May 2 20:00:00 GMT-0400 2000 (2)
Wed May 3 20:00:00 GMT-0400 2000 (2)
Fri May 5 20:00:00 GMT-0400 2000 (1)
Sat May 6 20:00:00 GMT-0400 2000 (3)
Sun May 7 20:00:00 GMT-0400 2000 (1)
Mon May 8 20:00:00 GMT-0400 2000 (5)
Tue May 9 20:00:00 GMT-0400 2000 (1)
Wed May 10 20:00:00 GMT-0400 2000 (2)
Thu May 11 20:00:00 GMT-0400 2000 (1)

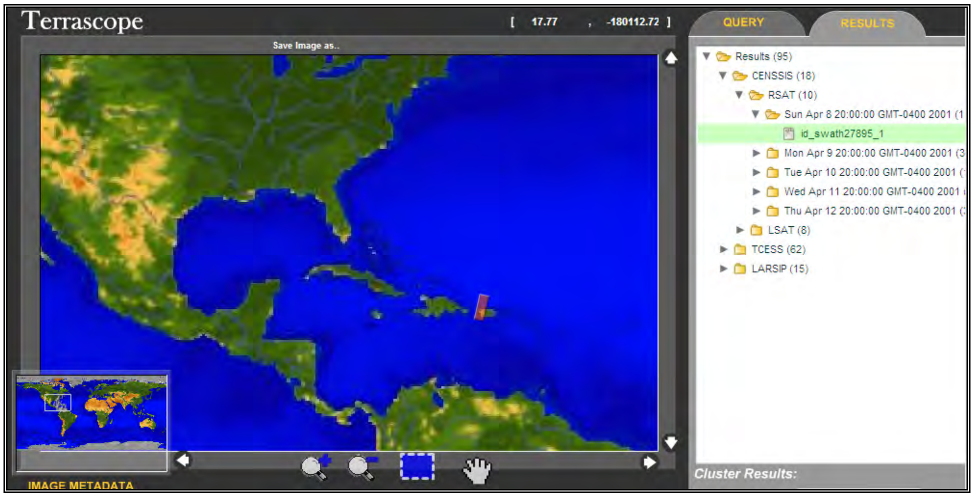
IMAGE METADATA
Id: id_swath22787_2
Source: TCESS Sensor: RSAT
Date: Sat Apr 15 20:00:00 GMT-0400 2000
Link: http://icarus.ece.uprm.edu/~terrascope/Images/TcessImages/images/

Figure 13 Image Clustering by Collection Date

The cluster tree can be changed dynamically and recursively in order to capture the clustering that better suits the user's information need. As shown in Figure 14, the user may select to group images by source (peer) and group again by sensor types within each source group. When the user selects one group (or subgroup), only the images contained in it will be visible in the map; the others will be hidden

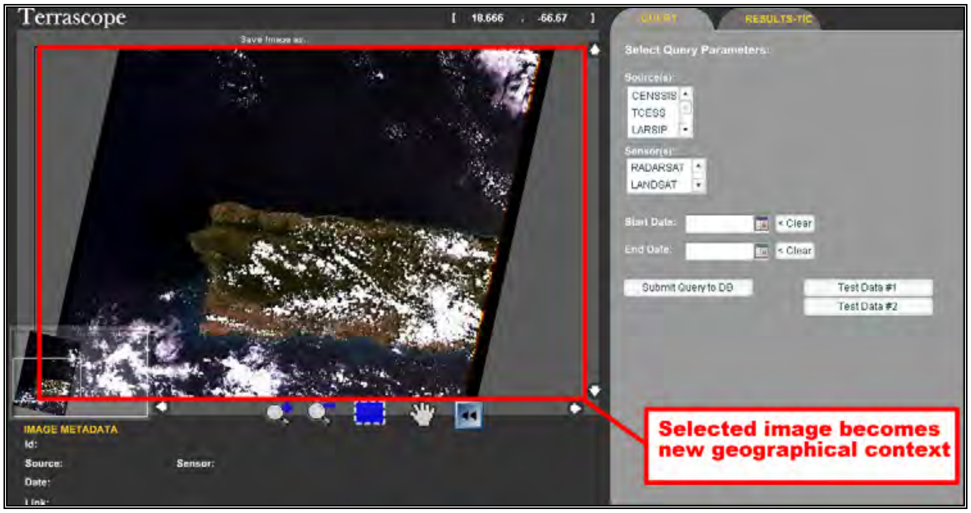
Recursively clustering the result the user can easily answer complex queries that are commonly needed by scientists. For example, consider the following query *"Search all images from any source that took images of America last week"*. Using the query parameters of TIN the user can select the area corresponding to America and the dates corresponding to "last week", submit the query and the result set will be geo represented in the map. This query was suggested to us by a visitor scientist from NASA during a TCESS review board.

After the result set is displayed the user might be interested in getting detailed information of the images. As shown in Figure 14, using TIC the user might easily find out which sources took images of the area and which sensors where used by each source. Also, the user might be interested in knowing which images were taken which day of "last week". To achieve this, the user may use the clustering options available in TIC. The image taken by CENSSIS using a RSAT sensor on Sun Apr 8 2001 can be easily targeted by traversing the tree. If the user clicks on the polygon the image will be displayed and will become the new geographical context as shown in Figure 15.



- Results (95)
  - CENSSIS (18)
    - RSAT (10)
      - Sun Apr 8 20:00:00 GMT-0400 2001 (1
        - id_swath27895_1
      - Mon Apr 9 20:00:00 GMT-0400 2001 (3
      - Tue Apr 10 20:00:00 GMT-0400 2001 (
      - Wed Apr 11 20:00:00 GMT-0400 2001
      - Thu Apr 12 20:00:00 GMT-0400 2001 (
    - LSAT (8)
  - TCESS (62)
  - LARSIP (15)

Figure 14 Recursive Clustering Example



Source(s): CENSSIS, TCESS, LARSIP

Sensor(s): RADARSAT, LANDSAT

Start Date: < Clear

End Date: < Clear

Id:

Source:

Sensor:

Date:

Link:

Figure 15 Selected image becomes new search context

At any point the user can start over the clustering by clicking the button labeled UNCLUSTER to start over with a flat result set list. Also, the user can make a new query at any point and TIC will automatically update the result set.

Chapter Five: Image Clustering Algorithms

In this chapter we describe various clustering algorithms that we have experimented with in order to support TIC graphical user interface. Each algorithm supports clustering of the result set tree to help the user target the desired data. All algorithms alleviate image agglomeration because images grouped in each subgroup have less probability of overlapping. As explained in Chapter 4, images contained within a group can be displayed by clicking on the corresponding node label in the tree.

5.1 Data Structures

Terrascope uses an array structure to hold the images in the result set together with their metadata. Each element in the array represents an image. The attributes of each element are the image's characteristics retrieved from the metadata in the XML message returned by the server. This array structure is used to geo-reference the clickable polygons in the context image.

Terrascope Image Clustering organizes result sets using the structure of a tree. The tree is implemented as a XML object. Each leaf of the tree represents an image and it is represented as a pair of a label and a value. The current version of TIC uses a unique identifier for each image provided by the databases to label the corresponding leaf node. The value of the leaf node is a pointer to the in the result set array. This mapping is shown in Figure 16. Branch nodes are labeled with a meaningful string that describes the data it contains.

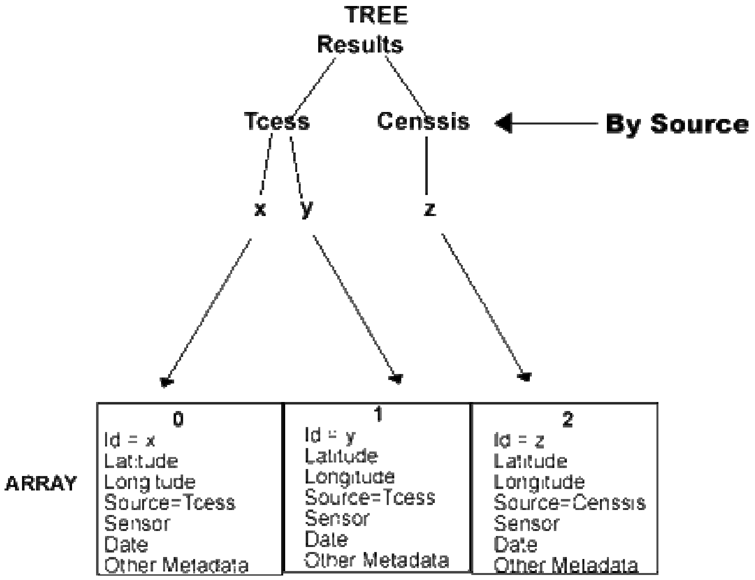


Figure 16 Array-Tree Mapping

We use a tree structure because it can easily support recursive clustering and it makes easier the indexing used by the clustering algorithms when traversing the tree. The data attribute keeps a mapping from each leaf node to the corresponding element in the results array and to the corresponding geo representation object. This mapping is needed to retrieve the images metadata saved in the array and map it to the geo representations used in the clustering algorithms.

5.2 Clustering by Source, Sensor and Dates

TIC currently clusters the result sets according to metadata attributes associated with images such as: source, sensor and date to cluster the images. The algorithm removes leafs of the current tree and according to the required clustering attribute it add them to a descriptive branch in a temporary tree. To access the attribute value we map the data value of the leaf in the tree with the corresponding element in the array (see Figure 16). After all leafs are added to the corresponding branch the labels of the branches are updated to display the quantity of leafs they contain and then GUI's tree is updated. Figure 17 shows the clustering by source algorithm. The clustering by sensor algorithm follows the same idea. The clustering by date differs from the others because it also sorts. Figure 18 demonstrates the clustering by date algorithm.

```
For all branches in current tree
    For all images in branch
        Get image source from Array
        For all source branches in new tree
            If image's source is branch's source
                Add image to branch
```

Figure 17 Date Clustering Algorithm

```
For all images
    Get image’s date from array
        For all date branches in new tree
            If image’s date is less than branch’s
                Add branch before current branch
                Add image to branch
            If image’s date is branch’s
                Add image to branch
```

Figure 18 Source Clustering Algorithm

5.3 Clustering by Minimum Overlapping

When the database consists of multiple highly overlapping images, geo representation of images may worsen the ability of users to find images. We designed several algorithms to address this problem.

The first algorithm that we designed segregates the images into clusters with no overlapping (as shown in Figure 19). The algorithm begins with one cluster and iterates over each image in the results array. The algorithm goes trough the available clusters and add the image in the first available cluster where that image does not overlap with the images previously placed in the cluster. If an image overlaps with at least one image in the cluster, a new cluster is created to add the image in.

```
For all images
    For all branches in new tree
        If image does not overlap other
             Add image to branch
    If image was not added
             Add branch
             Add image to branch
```

Figure 19 No Overlapping Algorithm

The result of running this algorithm was that we often obtained many tiny clusters resulting in a very wide tree. The reason for this is that almost every image overlaps with some other image in the result set. We decided to allow some kind of overlapping in order to narrow the tree. We achieved this by restricting the number of clusters while simultaneously maintaining a balance in the amount of overlap across the different clusters.

The number of clusters should not be a fixed number, because if we cluster a large result set into a small number of clusters it may result in too much overlapping. The number of clusters that we use is dependent of the size of the result set. On The number of clusters allowed is calculated using the following formula:

$$\text{MAX\_CLUSTERS} = 10 \lfloor \log(\text{result set size}) \rfloor$$

We developed two algorithms that cluster the result set into a fixed number of clusters with minimum overlapping. The two algorithms are presented in the next sections.

Algorithm #1 Minimum overlapping by collapsing neighbors

This algorithm (see Figure 21) clusters the result set into clusters with no overlapping. Then the algorithm narrows the tree by searching for clusters that are too small and collapse them with neighboring clusters. In our current prototype “too small” is defined as the number of images that would be held in each cluster if every cluster ended up with the same size.

```
For all images
    For all branches in new tree
        If image does not overlap other
             Add image to branch
    If image overlapped in every branch
             Add branch
             Add image to branch
For all clusters
    While cluster is too small
         Get image from next cluster
```

Figure 20 Minimum Overlapping: Collapsing Neighbors

Algorithm #2 Minimum overlapping by allocating clusters

This algorithm (see Figure 22) allocates the maximum number of clusters. For each image, it counts the overlapping hits in each cluster. In our current prototype, an overlapping hit is when an image overlaps with other image. The image is then assigned to the cluster which where it has the minimum overlapping.

```
Allocate clusters
For all images
   For all branches in new tree
         Find minimum overlapping
   Add image to branch were minimum overlapping
```

Figure 21 Minimum Overlapping: Allocating Clusters

We define:

- N = number of images in the result set
- C = number of clusters with no overlapping
- K = number of clusters with minimum overlapping
- Typically C is greater than K.

The complexity of the minimum overlapping algorithms will be:

- Alg #1 Collapsing Neighbors : N3 + C2
- Alg #2 Allocating Clusters: N3 + K

Algorithm #2 Allocating Clusters is more efficient than Algorithm #1 Collapsing Neighbors.

Chapter Six: Experimental Results

In this chapter we present the results of two types of experiments that we conducted to evaluate the effectiveness of our algorithms. The first type of study is a comparison of the algorithm effectiveness to eliminate overlapping. The second is a user study to measure the effectiveness of TIC's GUI.

6.1 Comparison of Overlapping Algorithms

We compared the effectiveness of the two algorithms: "minimum overlapping by collapsing neighbors" and "minimum overlapping by allocating clusters" in solving the agglomeration of images problem.

Algorithm #1, "minimum overlapping by collapsing neighbors ", clusters the result set into clusters with no overlapping. Then the algorithm narrows the tree, searching for clusters that are too small according to the Maximum number of clusters and tries to fill them with images of the next cluster.

Algorithm #2, "minimum overlapping by allocating clusters", allocates the maximum number of clusters. For each image, it counts the overlapping hits in each cluster. The image is then assigned to the cluster with where it has the minimum overlapping.
Both algorithms create a number of clusters that is dependent of the size of the result set. The number of clusters allowed is calculated using the following formula:

$$\text{MAX\_CLUSTERS} = 10 \lfloor \log(\text{result set size}) \rfloor$$

In order to measure the effectiveness of the algorithms we calculate the maximum of overlapping hits occurred over an image for each cluster. An overlapping hit occurs each time an image is overlapped by another image. For example: If an image in the first cluster was overlapped two times and another image was overlapped three times, the maximum overlapping hits per image in the first cluster is three. We ran the experiments using two different result sets with high overlapping. Figure 22 and Figure 23 demonstrate the results of the experiments. The horizontal axis of the graphs represents the clusters created by the algorithms and the vertical axis represents the maximum overlapping hits over an image in the corresponding cluster.

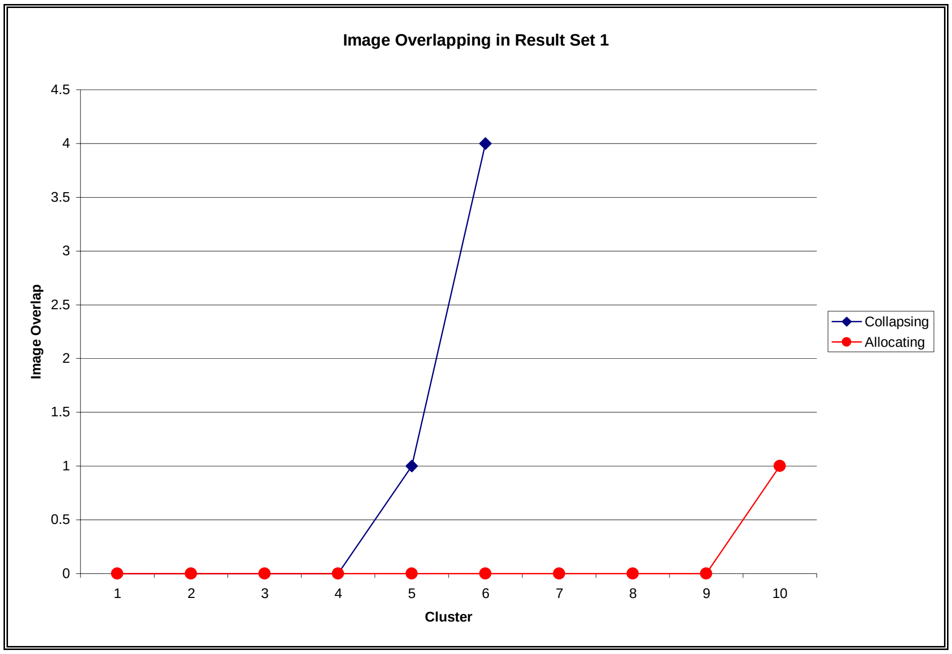


Figure 22 Experiment Results for Result Set 1

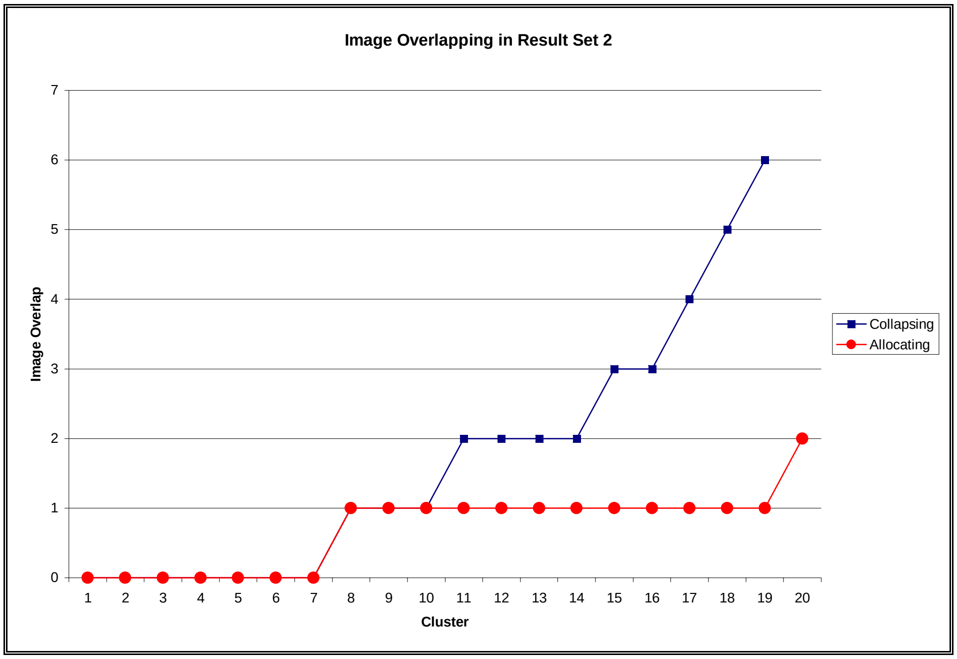


Figure 23 Experiment Results for Result Set 2

Algorithm #2 demonstrated to cluster the results in a better manner minimizing the overlapping occurrences in the result set. Also, as explained in Chapter 5, Algorithm #2 is faster than Algorithm #1.

6.2 User Study

This chapter presents the results of a user study conducted to evaluate the effectiveness of TIC improving the presentation of results and the ability of the users to visualize the properties of data. TIC provides the option to distribute the images in different clusters so the geo representation does not loose usability. Also, TIC provides a way to dynamically and recursively organize the results in a hierarchical way.

The user study was design to measure the benefits and effectiveness that TIC provides to image browsers.

6.2.1 Training

Before beginning the experiment, each participant was educated in the use of image browsers in general. We wanted to be sure each participant had a clear understanding of the assignment they were about to perform. Participants completed some pre-tasks. The participants had no time limit to do the pre-tasks. All the participants were informed that they did not have to proceed with any further tasks until they felt comfortable with the browsing system. The goal of the training was to verify that our subjects obtainded minimal familiarity with TIC.

6.2.2 Subjects Profile

There were 20 participants involved in this experiment; most of them were students at the University of Puerto Rico at Mayagüez, with various backgrounds including Computer Engineering and Electrical Engineering. The participants' ages range between 25-29 years. The majority of users reported to be experts users of personal computers (PC), using them an average of 37 hours per week. We believe that this is a representative sample of the type of user that would use Terrascope.

6.2.3 Methodology

The users were asked to perform various image retrieval tasks designed to evaluate three features of TIC. The three tasks attempted to measure are: the effectiveness of the "minimum overlapping" clustering, effectiveness of TIC helping to answer complex queries that require clustering, and the effectiveness of the recursive clustering of the images. In each task the users were asked to find some data making use of the geo representation alone and making use of the image clustering.

6.2.4 Results

6.2.4.1 Task #1

The users were asked to find an image with much overlapping. They were given the id of the image and were asked to find the metadata and to select the geo representation in order to view the actual image.

80% (16 out of 20) of the users found that it was easier and faster to target the image using TIC's clustering for minimum overlapping than using only the geo-represented polygons.

6.2.4.2 Task #2

The users were asked to find "Which sources took images of South America". To find the answer the user had to use the geo representation alone and had to use the clustering options of TIC. To use TIC the users where asked to cluster the images by Source and then select the clusters and view the according geo representations drawn in the map.

100% (20 out of 20) of the users found easier to find answers to queries using TIC's clustering and structure view of the results.

6.2.4.3 Task #3

The users were asked to make a common query done by scientist: "Which sensor used which source to take the images of South America". To find the answer the user had to use the geo representation alone and had to use the clustering options of TIC. To use TIC the users where asked to cluster the images "by Source" and recursively cluster "by Sensor".

100% (20 out of 20) of the users found easier to find answers to complex queries using TIC's recursive clustering and structure view of the results.

6.2.5 Conclusions

Our experiments prove that TIC improves the effectiveness of manipulation and image navigation by providing an organized and structured display of the retrieved images. Most users preferred to use the clustering methods to target the answer to their queries. All subjects reported that they consider the tools for dynamic clustering and manipulation of the results useful.

Chapter Seven: Conclusions and Future Work

We have presented the design of ***Terrascope Image Clustering***, for the **TerraScope** earth science data middleware system under development by the Advanced Data Management Research Group at the University of Puerto Rico Mayaguez as an alternative to reduce or avoid image cluttering caused by periodic capture of images covering similar geographical areas.

TIC is a convenient platform-independent way to manage high volumes of geospatial data. It was developed using Macromedia Flash MX Professional Actionscript. The module complements the geo representation and recursive navigation of TIN with a structured view of the information. TIC provides a mechanism to dynamically cluster the images. Users can recursively cluster the groups into subgroups. TIC allows users the flexibility to select clustering algorithms to be applied and the order in which they will be applied.

Future efforts of the TerraScope/TIC project will focus on the following goals:

Expanding the diversity of our collection of radar images.

Optimizing the TIC user interface.

Exploring new algorithms to cluster the images.

Bibliography

[1] 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

[2] B. Vélez, A. Cabarcas, L. Malavé: TIN: An Interactive Image Navigator Providing Ubiquitous Access To Distributed Geo-Spatial Data, Proceedings of International Conference on Information Technology: Coding and Computing Las Vegas, Nevada, USA, 2004

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

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

[5] 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**

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

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

[8] 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.
http://terraserver.microsoft.com/default.aspx

[9] 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.
http://www.singaren.net.sg/library/presentations/1aug03_01.pdf

[10] B. Vélez, J. Valiente, Interactive Query Hierarchy Generation Algorithms for Search Result Visualization, Proceedings of the IASTED International Conference Internet and Multimedia Systems and Applications, August 2001.