UCRL-TR-218362

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

# Efficient Data Management for Knowledge Discovery in Large-Scale Geospatial Imagery Collections

Chuck Baldwin, Ghaleb Abdulla

January 24, 2006

## Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

# *Efficient Data Management for Knowledge Discovery in Large-Scale Geospatial Imagery Collections*

**Chuck Baldwin and Ghaleb Abdulla**
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

## Abstract

We describe the results of our investigation on supporting ad-hoc and continuous queries over data streams. The major problem we address here is how to identify and utilize metadata for smart caching and to support queries over streaming and archived or historical data.

## Introduction

Interest in geospatial information has seen a dramatic increase over the last several years. Due in part, to the availability of such data, people are becoming more cognizant and comfortable with the presentation of information that has a significant geospatial character. Freely available and convenient tools such as Google Earth [1] have brought geospatial information to a wide audience. Other more specialized products are being created that enable deployment of GIS functionality and provide tools for assembling intelligent geographic information systems. Examples of such tools are ESRI ArcGIS, and the ENVY GIS system. For applications that involve large scale analysis of geospatial data , such as national security or disaster response; managing the shear quantity of information is a major undertaking where performance of these tools becomes an issue. In addition working with real time steaming geospatial data has not been adequately addressed by any particular community. Efficient streaming analysis and hybrid querying of large scale geospatial data requires careful consideration and advancements in the data management, information retrieval, and knowledge discovery communities. An important challenge is keeping up with the data rates without loosing any important information. Data streaming community developed ways and methods to support stream analysis; however, most of these techniques fit well for sensor data that come as a time series

In this context, we have been investigating and developing techniques, algorithms, and tools that will enable analysis of large scale geospatial information. We are primarily addressing the performance problem of streaming and hybrid queries (queries that combines streaming and static archived data) of geospatial imagery. There has been some work of streaming data and hybrid queries. The system proposed in [2] computes and caches predicate result ranges. As long as the queries are within the pre-computed predicates, the system doesn't need to re-compute. However, and over time, the system expands ranges cached so that they are more likely to contain future stream values of expensive functions.

We are looking into a more specific problem which combines streaming images and feature searching. We want to utilize the application characteristics and user access patterns to design smart-caching algorithms. These algorithms will utilize spatial and temporal data indexes over image data.

## Problem Description

### Queries over geospatial[1] image data

Geospatial image data can be stored as Binary Large OBject (BLOB) or as a raw image file; additional information or metadata can be stored associated with each image. Such information can be classified into:

- **Technical and/or engineering:** Information related to the details of the acquisition process, image characteristics, or storage.
- **Derived data:** Information derived from raw image data; for example, features related to color distribution. Extracted or derived data is saved as feature vectors and used for search and retrieval of relevant images and objects' features.
- **Knowledge-based:** Information used to relate images to real world entities such as annotations.

For the time being we don't consider all these sources of metadata, however, we are designing the system while keeping in mind that it should be extensible. Metadata can be useful not only for performance, but also it can be a powerful tool for data fusion and knowledge discovery.

Queries on images can be done statically where the image and the related information are stored in a database[2] and an object or a feature is matched against the database of stored images. However, in certain applications the feature comparison needs to be done in real time where an object is matched against a stream of images. This kind of interaction requires real-time feature extraction which significantly increases the computational requirements and requires the use of Massively Parallel Processing (MPP) computers. In addition to the computation requirements, the I/O becomes a real bottleneck especially when using an MPP with shared storage or if there is a need to communicate information between the computing nodes. The problem gets more complicated when historical data from the archive need to be accessed and compared against the data stream. In addition to using MPP to speed up the computing, efficient data access algorithms should be developed to speed up processing of images.

The way the analyst interacts with the system has also an effect on the proposed algorithms. We assume that the analyst will interact with the system in one of the following ways:

1- Browsing image collection: This includes moving in any direction and zooming in and out.
2- Searching for specific location: The analyst might be interested in locating a specific place using the name of that place. For example a city, a country, or specific homes address.

[1] In this report, image refers to geospatial image (aerial photo, satellite image, etc.)

[2] We refer to RDBM system or any file management system as a database.

3- Searching for a particular object within an image: For example, a building of a specific shape or train tracks. In addition, the analyst might be interested in the location of the retrieved objects.

Understanding user interactions will help in identifying the needed metadata to support the smart caching for the streaming images.

**Geospatial Data and Meta-Data**

The meta-data consists of summary information and/or specific pieces of the data that characterize some important concept or aspect of the image data. It is therefore important to carefully define such summary information to be of maximal use for applications. Fortunately, there are many meta-data standards for various aspects of general imagery and geospatial information [11]. We are looking at two general types of geospatial information in this work. The first type of geospatial information is aerial or satellite imagery of the Earth (or another planetary object), called *raster data*, whose "frames" are, or can be, geo-referenced to a known Coordinate Reference System (CRS). This imagery can be in any prescribed spectrum, numeric type and range, as well as resolution or accuracy. Obviously all of these "high-level" aspects of the imagery should be captured in the meta-data for specific image frames (or sequences). The second type of geospatial information we are concerned with is individual points or sequence of points or other discrete parameter sets, known as *vector data*, whose coordinates are, or can-be, geo-referenced to a known CRS. This type of information can define individual locations on the Earth, a polygonal (open or closed) shape defined by the sequence of points, or some instance of a mathematical curve with field values given by the parameters. The points can have a bounding polygonal area or a known statistical distribution. Assuming that the set of points is large, the meta-data would (possibility) describe a larger and simpler polygonal area that covers all of the individual locations

Minimal sorts of meta-data that is available and consistent for all forms of geospatial information are associated with the spatial-temporal aspects of the observation; such as time and location of the observation. This also associates well with many of the common and useful operations of search and navigation applications along with general GIS graphical user interface (GUI) mechanisms. Narrowing this more, we define and populate a general meta-data structure for the observational data D, with the following information:

1. The temporal component, t, of the observation.
2. The geospatial Coordinate Reference System (CRS) for the associated positions of the observational data; location (x,y) and height h (if present).
3. A set of position information (in the CRS) that describes the maximal extents of the associated observations; $\{(x_i, y_i), h_i\}$ for $i = 1, \ldots, n$.

A useful simplification that can be made is to augment the structure with a rectilinear structure, called a bounding box, which covers or completely encloses the set of positional information. This is easily computed and sometime the exact data that comprises the maximal extents. We denote this structure as:

$$B(\lambda_n, \upsilon_n),$$

where the lower corner of the data extent is denoted by $\lambda_n$ and the upper corner is denoted by $\upsilon_n$. Examining meta-data standards that exist for raster and vector data, this type of information is easily computable, if not already accounted for. The key point is that we require and use at least this minimal set of meta-data at all stages of computation. In many cases the meta-data that can be used from a particular type of geospatial data far exceeds this and can be utilized in further management and query operations. Information concerning the actual values that make up the data can be used to answer queries about the observation itself. These would include things such as the maximum and minimum intensities of a standard raster image. We are also looking into the idea of extending the metadata to support annotated images.

## Results and conclusions:

### System Design and Architecture

Figure 1 shows the high-level view of the user and system interactions. The goals of the system is to provide an efficient querying capability for geospatial information while managing streaming data as well as historical, archived, data.

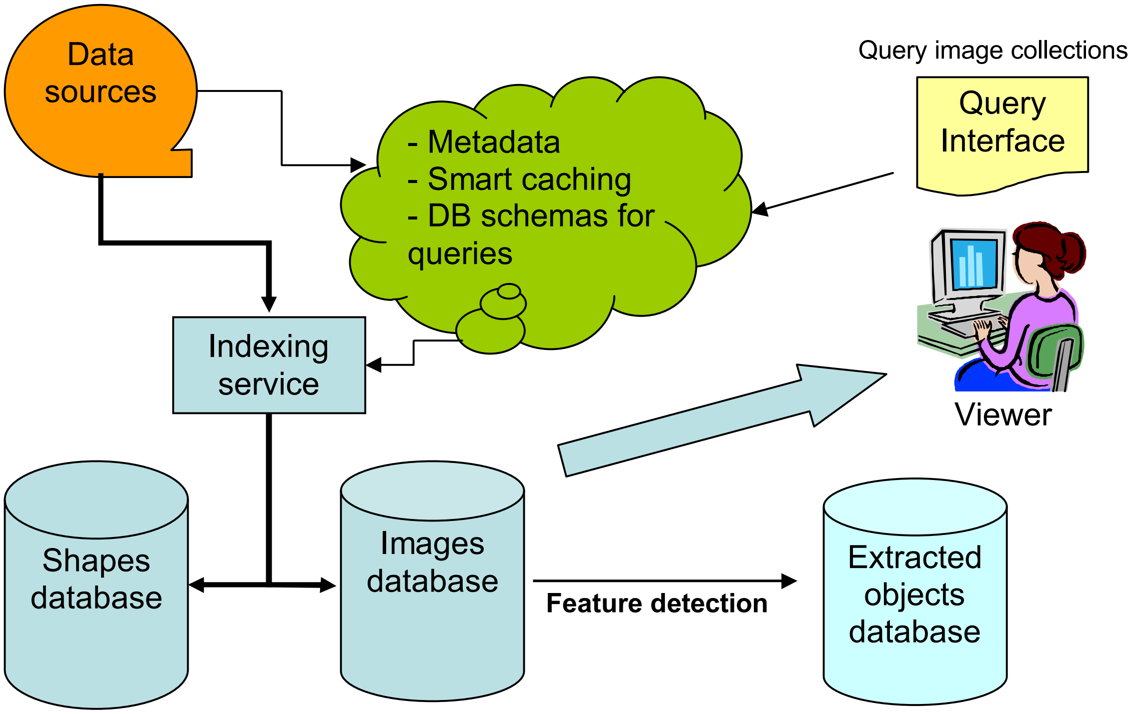

**Figure 1: Caching and metadata will be used to support indexing services and to help answer user queries through an interface that will support data streams.**

Data sources are being input and indexed into the system using extracted (or computed) metadata. In addition to indexing the data (into an archival storage system), metadata is saved to provide a basis for querying by end users. The queries we want to first address by this system are the following three:

1.) Given a location l, in a specified CRS $\rho_l$, retrieve (references to) all images that contains that particular location. The geospatial predicate is $l \in B(\lambda_n, \upsilon_n)$.
2.) Given a rectilinear region $B(a,z)$, in a specified CRS $\rho_B$, retrieve (reference to) all images that intersect this box. The geospatial predicate is $B(a,z) \cap B(\lambda_n, \upsilon_n) \neq \phi$.
3.) Given a rectilinear region $B(a,z)$, in a specified CRS $\rho_B$, retrieve (reference to) all images that are directly adjacent to this box. The geospatial predicate is $B(a,z) \cap B(\lambda_n, \upsilon_n) = \ell$, where $\ell$ is a one-dimensional line (the joining edge).

The vector and box objects referred to here, and used in the predicates, are known as *box calculus* operations and an efficient implementation of such operations are crucial to the overall performance of the queries.

**Conclusions**

In this work we described the results of the research and design efforts for constructing and managing large-scale collections of geospatial data. The problem is examined from the perspective of efficiency, extensible capabilities, and effective user interaction. We have found that the problem is solved by utilizing meta-data; extracting it early and maintaining it through-out the processing pipelines. Basic interaction, through queries, is identified and minimal sets of meta-data are identified. The construction of indexes that will enable the effective and extensible resolution of queries is done through this meta-data. The constructed index marks a static configuration that can be used in further processing. We are designing and implementing a user API for interaction with this index so that future capabilities can be built around its format and information. For example the image browser will be able to access the API to prefetch and cache the images based on the location that is being currently viewed. While this can be easily supported during a regular browsing session, it is more challenging when the analyst works with streaming data or querying the image set based on features. The associated query procedures are done by using the index structures and managing the large-scale interaction with an external data collection. This management has been designed and centers on ideas of caching and pre-fetching of data.

## Future Work

We are planning on integrating our work with the image browser. The image browser will be able to use the index to prefetch images ahead of time to the memory based on the current user activity. The initial API will support the spatial index; however we plan to our index to include spatial and temporal attributes.

## References

[1] http://earth.google.com/

[2] Matthew Denny, Michael J. Franklin: Predicate Result Range Caching for Continuous Queries. SIGMOD Conference 2005: 646-657

[3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In Proceedings of 21st ACM Symposium on Principles of Database Systems (PODS 2002), 2002.

[4] O. Kao and S. Stapel, Case study: Cairo---a distributed image retrieval system for cluster architectures, Distributed multimedia databases: techniques applications, 2002,1-930708-29-7, pp:293—305, Idea Group Publishing.

[5] M. Garofalakis, J. Gehrke, and R. Rastogi. Querying and mining data streams: You only get one look. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 2002.

[6] Alexandre R. J. Francois, A Hybrid Architectural Style for Distributed Parallel Processing of Generic Data Streams, ICSE '04: Proceedings of the 26th International Conference on Software Engineering, 2004, 0-7695-2163-0, pp:367—376, IEEE Computer Society

[7] Philippe Rigaux, Michel Scholl, and Agnes Voisard. Spatial Databases with applications to GIS. Morgan Kaufmann, 2002.

[8] John L. Hennessay and David A. Patterson, Computer Architecture: A Quantitative Approach; 3rd Edition. Morgan Kaufmann, 2002.

[9] Sameer Tyagi, Keiron McCammon, Michael Vorburger, and Heiko Bobzin, Core Java Data Objects. Prentice Hall, 2004.

[10] David Jordan and Craig Russell, Java Data Objects. O'Reilly Press. 2003.

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