Querying Streaming Geospatial Image Data: The GeoStreams Project

Quinn Hart
CalSpace
University of California, Davis
Davis, CA, U.S.A.
qjhart@ucdavis.edu

Michael Gertz
Department of Computer Science
University of California, Davis
Davis, CA, U.S.A.
gertz@cs.ucdavis.edu

Abstract

Data products generated from remotely-sensed, geospatial imagery (RSI) used in emerging areas, such as global climatology, environmental monitoring, land use, and disaster management, require costly and time consuming efforts in processing the data. For the researcher, data is typically fully replicated using file-based approaches, then undergoes multiple processing steps, these steps often being duplicated at many sites. For the provider, data distribution is often tied directly to the data archiving task, focusing on simple, coarse grained offerings. Many RSI instruments transmit data in a continuous or semi-continuous stream, but current techniques in processing do not utilize the stream nature of the imagery. Recent research on continuous querying of data streams offer alternative processing approaches, but typically assume tuple style data objects, relying on traditional relational models as basis for query processing techniques and architectures. Complex types of stream objects, such as multidimensional data sets or raster image data, have not been considered. Our project, GeoStreams*, is a framework to process multiple continuous queries against streaming remotely-sensed geospatial image data. This paper introduces the basic features underlying the* GeoStreams *model. We describe some interesting aspects in processing streaming image data, including optimization and evaluation using specialized index structures.*

Remotely sensed data, in particular satellite imagery, play an important role in many environmental applications and models [10]. Simple, convenient access to remote sensing data has traditionally been a barrier to research and applications. The huge amounts of data generated by the Earth Observing System (EOS) platforms have precipitated a change in this scenario, and access to data products has become substantially easier. New EOS data archives offer fine examples of more transparent data access. However, access to this imagery still largely centers on choosing coarse grained, standard data products for specific regions and times. Applications that study changes in the environmental landscape require frequent, often continuous access to these data, and the temporal discontinuity in these access methods can force complicated preprocessing and synchronization steps between the data provider and the data user.

The sensors themselves, however, follow much more of a streaming paradigm. Data is acquired continuously and transmitted to receiving stations in a continuous manner. Outside the realm of image databases, there have been recent advancements in the more general field of data stream management systems (DSMSs), with new proposed query processing techniques [8] and research applications [1,3,4]. In such systems, data arrives in multiple, continuous, and time-varying data streams and does not take the form of persistent relations. There is clearly a potential benefit in taking techniques developed for DSMSs and adopting them to geospatial Remotely-Sensed Imagery (RSI) data.

The *GeoStreams* project investigates joining these two disciplines. In the *GeoStreams* architecture, researchers will explicitly consider the continuous temporal nature of RSI and formulate queries on these streams. Outputs of these queries continuously feed new RSI data to the researcher. These streams can be fed into applications to allow a continuous source of new input data from a single stream, or saved in more traditional RSI formats. As the functionality of the RSI DSMS increases, more aspects of the applications can be formulated into the queries themselves.

Requirements for the *GeoStreams* architecture include (1) identifying a query syntax that is natural for environmental application developers, as well as concise and unambiguous; (2) development of a core set of operations for RSI access; (3) query optimizations that allow a DSMS systems to tailor their execution plans to the currently active queries; and (4) execution plans that take advantage of the highly organized structure that is a trademark of RSI data. A wider range of interesting activities also include methodologies for continuous client-server data exchange, wire formats for streams of RSI, and investigating costly blocking operations on RSI data like image reprojections that can be

incorporated into a streaming system.

An Overview of the *GeoStreams* architecture is shown in Figure 1. Multiple users connect to the *GeoStreams* server and formulate queries to the system. The system is optimized for continuous queries on the input satellite stream of data. The queries are parsed and validated, then optimized. Optimization includes single and multi-query methods in this model, combining queries to minimize number and size of images that are created and maintained in the *GeoStreams* system. Minimizing the size of images reduces both memory usage and computational burden. Because of the way images can be shared between queries, however, computing query costs can be non-trivial. New queries affect the execution plan for the system, but these changes are made incrementally, because the execution is continuously working on the incoming RSI stream. This stream comes from a stream generation module that reinterprets the raw satellite data into a format more suitable for query processing.

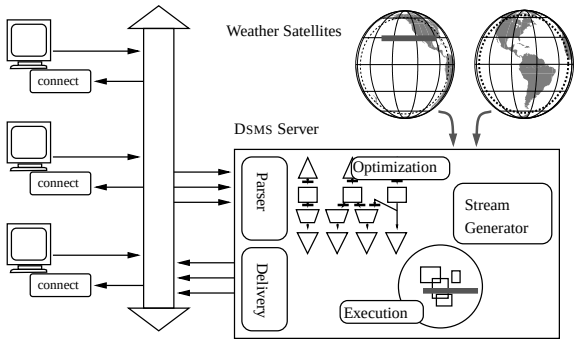


Figure 1. GeoStreams overview

Query execution is highly dependent on the structure of the incoming data. In our model, the RSI data is manipulated one row at a time. This matches the form of the satellite stream and is also convenient for satisfying multiple queries. Query execution ends with operators to return the data to the clients, which require persistent or synchronous connections on both the server and the client.

Our first RSI stream is continuous weather imagery from the National Oceanic and Atmospheric Administration (NOAA) Geostationary Operational Environmental Satellite (GOES) [6]. All data from the GOES satellite is transferred via a format specific these instruments. This continuous data stream transmits at approximately 2.1 Mb/sec. It has two instruments, the Imager and Sounder, which have 5 and 19 spectral channels respectively. GOES scans various sections of the Earth's surface about once every 15-30 minutes in spatial frames. A single frame varies in size from about 100MB to 400MB, depending on the region scanned. The ground resolution of the pixels varies between spectral channels. Data are basically delivered in a line by line manner, as GOES scans the hemisphere from North to South.

Images and image manipulation are based on image algebra [11], which is a rigorous and compact method for describing images, image transformations, and analysis. Initially, the notation for image algebra can be confusing and is kept to a minimum in this paper, although some high points in the context of streaming queries are discussed below. Image algebra is a many-valued algebra that includes *Point Sets*, *Value Sets* and *Images*.

Points Sets are defined in some topological space and correspond to the spatio-temporal location of the individual values in an image. Unlike many image definitions, the *GeoStreams* point sets typically include a temporal dimension. This allows for functional manipulations to be easily described in the algebra. Point sets are denoted with bold capital upright letter and points within a point set are denoted with lower case bold letters, i.e., $\mathbf{y} \in \mathbf{X}$.

Value sets encompass values associated with the points in the point set and are taken from a homogeneous set of operands, typically sets like integers, $\mathbb{Z}$, or real numbers, $\mathbb{R}$, although more complex, multi-valued sets can be defined. Value sets have the usual operations associated with their universal set.

Images are defined in general terms. The notation $\mathbb{F}^{\mathbf{X}}$ describes the set of all functions, $\{f \in \mathbb{F}^{\mathbf{X}} : f \text{ is a function from } \mathbf{X} \text{ to } \mathbb{F}\}$. An image is such a function that maps from a point set $\mathbf{X}$ to the value set $\mathbb{F}$. For an $\mathbb{F}$-valued image, $(\mathbf{a} : \mathbf{X} \to \mathbb{F})$, $\mathbb{F}$ is the possible *range* of the image $\mathbf{a}$ and $\mathbf{X}$ is the *domain* of $\mathbf{a}$.

Another convenient notation for an image $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ is the *data structure representation*, $\mathbf{a} = \{(\mathbf{x}, \mathbf{a}(\mathbf{x})) : \mathbf{x} \in \mathbf{X}\}$. Here the pair $(\mathbf{x}, \mathbf{a}(\mathbf{x}))$ is a *pixel* of the image. The first coordinate $\mathbf{x} \in \mathbf{X}$ is the *pixel location* and the second coordinate $\mathbf{a}(\mathbf{x}) \in \mathbb{F}$ is the *pixel value* at location $\mathbf{x}$.

Image Operations are the basic building blocks for queries to the *GeoStreams* system. These operations include functional operations, image restrictions to specific point sets, spatial transforms on images from one point set to another, and neighborhood operations where multiple pixels from an image are combined to a single value. Figure 2 shows examples of these basic operations.

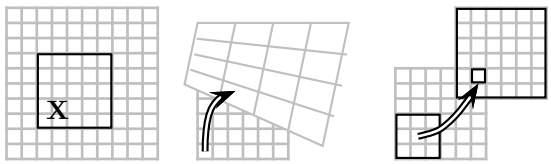
Defined operations on or among images include any operation that operates on the value set $\mathbb{F}$, which induces a natural operation on $\mathbb{F}$-valued images. For example, the addition of two images can be defined as $\mathbf{a} + \mathbf{b} = \{(x, a(x) + b(x)) : x \in \mathbf{X}\}$.

Image restrictions return images restricted to a given point set. In image algebra, if $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$, then the restriction, $\mathbf{a}|_{\mathbf{Z}}$ is defined as $\mathbf{a}|_{\mathbf{Z}} \equiv \mathbf{a} \cap \mathbf{Z} \times \mathbb{F} = \{(x, a(x)) : x \in \mathbf{Z}\}$. Some image models have formulated restrictions as selection operations, $\sigma_{x \in \mathbf{Z}}(a)$. Others formulate this as a spatial join, $a \ltimes_{a.x=Z.x} \mathbf{Z}$. Still others formulate restrictions functionally on an image data type.

Spatial restrictions are possibly the most important of all operations, and flexible methods for defining new point sets need to be included in query formulations. This is especially true in our model where point set restrictions define not only spatial, but also spatio-temporal limits on incoming data streams. Some point set manipulations are easy to represent, but many useful manipulations are more complex. Details of all potential point set manipulations have not been fully investigated, but since point sets are sets, relational algebra could be used as a framework for subset definitions.

Spatial transformations map an image from one point set to another. In general, for any function, f, between two point sets, $f : \mathbf{Y} \rightarrow \mathbf{X}$, and an image $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$; the spatial transform is defined as: $\mathbf{a} \circ f = \{(y, a(f(y))) : y \in \mathbf{Y}\}$.

Spatial transformations are used for magnification, rotation, and other general spatial manipulations. For geolocated imagery, reprojection of data into a new coordinate system is also a geometric transformation.



(a) $\mathbf{a}|_{\mathbf{X}}$ (b) $\mathbf{a} \circ f$ (c) $\mathbf{a} \bigoplus N$

Figure 2. Image Operations

Neighborhood operations allow for multiple pixels from a single image to be combined to create a single pixel value in a new image. Neighborhoods allow for aggregation functions like averaging, edge detection, speckle removal, and other operations. For example, $\mathbf{a} \bigoplus N$, indicates a local summation function where N represents an image template for the operation around local points.

Queries in the *GeoStreams* framework do not build on a variant of SQL syntax, but on something closer to the image algebra representation and on specialized interfaces. For example, consider a query for a normalized difference ratio on two satellite bands, a common type of index for environmental applications. We want to continuously receive this index for a particular region, reprojected to some convenient coordinate system, e.g. UTM. In image algebra, this could be represented as $((\mathbf{a}-\mathbf{b})/(\mathbf{a}+\mathbf{b})) \circ UTM|_X$, where $((\mathbf{a}-\mathbf{b})/(\mathbf{a}+\mathbf{b}))$ represents the index, $\circ UTM$ represents a function mapping from the satellite image to a new coordinate system, and $|_{\mathbf{X}}$ represents a restriction to some spatial extent.

This simple query demonstrates some of the problems in query formulation for a user. There must be methods to create complex spatial transform and restriction criteria, as mentioned before. These problems have been addressed in other research, and a number of workspace and workflow [2] models have been proposed, which are being investigated as a potential platform for describing general queries in *GeoStreams*.

However, in the near term, a query interface based on the OpenGIS Web Map Services (WMS) specification [5] is being developed. This simple interface does not allow for a sophisticated set of user queries, but it does investigate the most basic requirements of serving many spatial restrictions and geometric transformations to many clients. Basically, the interface allows users to specify specific data products, coordinate systems, and spatial extents. Temporal restrictions can also be identified. Queries like the one above could be specified, as long as the index itself is identified as a product in the server. The WMS specification further simplifies query formulation by standardizing and simplifying both spatial transforms and restrictions to a limited but well-defined subset. In general, the WMS specification limits queries to the form, $\mathbf{a} \bigoplus N \circ f|_X$, where $\mathbf{a}$, N, f, and X are specified in a simple standard way.

Query optimization attempts to limit the processing time and/or the amount of memory usage for the DSMS as a whole. In *GeoStreams*, query optimization is primarily concerned with two goals: query rewriting to limit the amount of work done in the system, and exploiting common subsets within the queries active against the image stream.

Consider the previous example, $((\mathbf{a} - \mathbf{b})/(\mathbf{a} + \mathbf{b})) \circ UTM|_{\mathbf{X}}$. This is a natural way to represent the query, but not an efficient computation method. As written, the index and spatial transform are performed on the entire domain of the image, most of which is discarded in the final restriction. Generally, moving restrictions to the front of the query improves efficiency. Restrictions can be reordered over spatial transforms by transforming the restriction point sets as well. For example, given $\mathbf{Y} = \{UTM(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$, the above query can be rewritten as,

$$((\mathbf{a} - \mathbf{b})/(\mathbf{a} + \mathbf{b}))|_{\mathbf{Y}} \circ UTM \quad \text{or}$$
$$((\mathbf{a}|_{\mathbf{Y}} - \mathbf{b}|_{\mathbf{Y}})/(\mathbf{a}|_{\mathbf{Y}} + \mathbf{b}|_{\mathbf{Y}})) \circ UTM$$

Simple heuristics on queries, like those above, work well in the *GeoStreams* architecture, especially in the case were queries are limited in complexity, as they are with the WMS query interface. They also allow for some independence between the single- and multi-query optimization steps.

Once the individual queries are rewritten to optimize their individual execution, the queries are then optimized in a multi-query fashion as well. Optimization here centers around grouping similar query components into a single operation that works simultaneously for a group of queries. In DSMS research this has multiple conceptual definitions, including grouped filters [8] and query indexing [9]. Figure 3

shows a typical query index scheme for a spatial restriction operation, where rather than each query requiring its own restriction operator, a single restriction module has indexed the point sets of a number of active queries. For each continuous user query, a region is associated that describes the restriction for that query. For incoming RSI data, it is determined what data is relevant to what user queries and which queries can share incoming data.

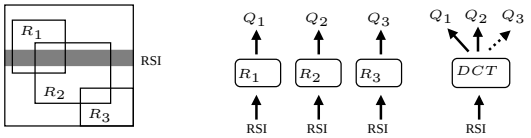


Figure 3. Restrictions on multiple queries

By developing modules for the basic image operations that can take as input a single RSI stream and distribute results to multiple output streams, the complete DSMS in the *GeoStreams* architecture is a number of these operators joined together for a complete system. This allows not only the pipelining of image data to operators to which the data is of interest, but it also facilitates the sharing of image data among queries that have non-disjoint query regions.

Query Execution is tied intrinsically to the query plan developed by the optimizer, and also by the organization of the incoming data stream. Modules are developed for each of the basic image operations, which satisfy multiple queries in a single operation. The modules are linked together for complete query execution. We have discussed how our RSI data stream comes in an ordered row-by-row arrangement. This organization plays an important role in how modules in the query plan are arranged.

Figure 3 shows an example module for processing multiple query image restrictions. For the restriction module, we have proposed the Dynamic Cascade Tree (DCT) [7], a space efficient structure to index query regions that are part of more complex queries against RSI data streams. The spatial trends inherent to most types of streaming RSI data is exploited to build a small index that is especially efficient when the incoming stream data are in close spatial proximity. Queries can be answered very quickly if the next data stream segment has the same result as the previous query and will incrementally update a new result set when the result is different. Based on the information provided by the DCT, incoming data can be pipelined to respective query operators, providing the basis for multiple-query processing models for streaming RSI data.

In Conclusion, we have described some of the basic concepts underlying the plans for a complete *GeoStreams* DSMS architecture for queries on streaming RSI data. We have already demonstrated the effectiveness of some of the basic modules within the system, for example, using the DCT as a method for indexing multiple query restrictions. Work is started on developing a preliminary system using the WMS specification as a basis for web-based access to the DSMS. There are a number of additional issues that can be investigated in this work, including determining the best wire formats for streaming query results, integrating mature publish/subscribe ideas into data delivery of RSI streams, allowing users to start queries in the past while maintaining a streaming paradigm and other issues. Our hope is that the test-bed developed here can be used to investigate these additional issues as well. The project is described at *http://db.cs.ucdavis.edu/geostreams*.

This work is supported by the NSF grant IIS-0326517.

References

[1] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. Aurora: A new model and architecture for data stream management. *The VLDB Journal*, 12(2):120–139, August 2003.

[2] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludaescher, and S. Mock. Kepler: An extensible system for design and execution of scientifi c workfbws. In *16th Intl. Conference on Scientific and Statistical Database Management (SSDBM)*, 2004.

[3] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, R. Motwani, I. Nishizawa, U. Srivastava, D. Thomas, R. Varma, and J. Widom. STREAM: The Stanford stream data manager. *IEEE Data Engineering Bulletin*, 26(1):19–26, March 2003.

[4] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, and M. A. Shah. TelegraphCQ: Continuous datafbw processing for an uncertain world. In *First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*, 2003.

[5] J. de La Beaujardiere. Web map service. OpenGIS Implementation OGC 04-024, Open Geospatial Consortium Inc., Aug 2004.

[6] *GOES I-M DataBook, Revision 1*. Space Systems-Loral, http://rsd.gsfc.nasa.gov/goes/text/goes.databook.html, 1996.

[7] Q. Hart and M. Gertz. Indexing query regions for streaming geospatial data. In *2nd Workshop on Spatio-temporal Database Management, STDBM'04*, 2004.

[8] S. Madden, M. Shah, J. M. Hellerstein, and V. Raman. Continuously adaptive continuous queries over streams. In *Proc. of the 2002 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM Press, 2002.

[9] S. Prabhakar, Y. Xia, D. Kalashnikov, W. Aref, and S. Hambrusch. Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects. *IEEE Trans. on Computers*, 51(10):1124–1140, 2002.

[10] A. Skidmore, editor. *Environmental Modeling with GIS and Remote Sensing*. Taylor & Francis, 2002.

[11] J. N. Wilson and G. X. Ritter. *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press, 2nd edition, 2001.