A Data and Query Model for Streaming Geospatial Image Data*

Michael Gertz, Quinn Hart, Carlos Rueda, Shefali Singhal, and Jie Zhang

Department of Computer Science, University of California, Davis, CA 95616, U.S.A.
{gertz, qjhart, carueda, sssinghal, jiezhang}@ucdavis.edu

Abstract. Most of the recent work on adaptive processing and continuous querying of data streams assume that data objects come in the form of tuples, thus relying on the relational data model and traditional relational operators as basis for query processing techniques. Complex types of objects, such as multidimensional data sets or the vast amounts of raster image data continuously streaming down to Earth from satellites have not been considered.

In this paper, we introduce a data and query model as a comprehensive and practically relevant basis for managing and querying streams of remotely-sensed geospatial image data. Borrowing basic concepts from Image Algebra, we detail a data model that reflects basic properties of such streams of imagery. We present a query model that includes stream restrictions, transforms, and compositions, and provides a sound basis for formulating expressive and practically relevant queries over streams of image data. Finally, we outline how the data and query model is currently realized in a data stream management system for geospatial image data that supports geographic applications.

1 Introduction

Data products generated from remotely-sensed (satellite) imagery and used in emerging applications areas such as global climatology, environmental monitoring, land use, and disaster management currently require costly and time consuming efforts in processing data [12,13,25]. For geographic applications, data is typically replicated using file-based approaches and has to undergo several batch-oriented processing steps before it eventually can be processed to obtain a data product. These processes are often duplicated at many sites for different and even the same type of applications.

Many satellite instruments transmit data in continuous streams to receiving stations. Multi- and hyper-spectral imagery for different wavebands that describe radiometric reflectance from the Earth's surface is typically transmitted in the form of *raster images*. Existing systems for processing the data, however, neither utilize the stream nature of the imagery nor do they expose database like concepts and architectures that provide users and applications with expressive and efficient operators to retrieve and manipulate streams of geospatial image data.

* This work is in part supported by the NSF under award IIS-0326517.

T. Grust et al. (Eds.): EDBT 2006 Workshops, LNCS 4254, pp. 687–699, 2006.

On the other hand, there have recently been considerable advancements in data stream management systems (DSMS), where data arrives in continuous and time-varying data streams and does not take the form of persistent relations [1,5,7]. Most of the proposed techniques, such as adaptive query processing, operator scheduling, and load shedding, exclusively concentrate on simple structured, usually relational data. Query operators and query processing techniques are based on those known and studied in the context of relational databases.

The above observations suggest that there is a strong potential benefit in adopting techniques developed for relational DSMS to the management of streaming remotely-sensed image data. However, the complexity and heterogeneity as well as various non-traditional (compared to relational) operations on geospatial image data pose several challenges. First, remotely-sensed imagery exhibits characteristics of spatio-temporal data. That is, image data taken at a particular point in time describes some properties of a spatial extent on the Earth's surface. In addition, remotely-sensed data is *geo-referenced*, i.e., image data (pixels) can be mapped to locations on the Earth based on some coordinate system. Second, image data is transmitted at a very high rate; well-known satellites such as GOES [8], Landsat [15] or Aqua/Terra [20] each continuously stream about 20-60GB of remotely-sensed image data to receiving stations every day. Third, operators on geospatial image data are more complex than traditional (relational) operators and have to take characteristics of the remotely-sensed data into account, in particular their geographic and stream organization properties.

In this paper, we present a data and query model as basis to formulate (continuous) queries over streaming geospatial image data. Our focus is on the characteristics of remotely-sensed data originating from satellites and used in geographic applications. We formulate a data model that takes both the spatio-temporal and geo-referenced nature of image data into account. We describe three classes of operators: stream restrictions, transforms, and compositions. These allow the formulation of queries to (1) select image data of interest based on its spatio-temporal properties, such as spatial regions of interest and time intervals, (2) perform different types of neighborhood operations and spatial transforms on image data, and (3) combine image data from different streams (corresponding to different spectral channels). We also study some properties of the operators in terms of space and time complexity, as these heavily depend on the organization of the image data in a stream. Overall, our goal is to establish a framework to build a stream management system particularly designed to operate on streaming remotely-sensed data and to stream data products to clients and geographic applications in real-time.

The rest of the paper is organized as follows. In Section 2, we present the data model underlying streaming geospatial image data. Section 3 introduces the query model and presents different types of operators, including stream restrictions, transforms, and compositions. In Section 4, we give an overview of our prototypical query processing infrastructure. After a review of related work in Section 5, we conclude the paper with a summary and outlook in Section 6.

2 Data Model

The primary types of objects in our stream processing framework are *images* and *streams*. Although an image has a fairly intuitive description, such as a (rectangular) set of pixels together with their pixel values, some more formal definitions are necessary to establish a sound framework for describing operations on individual images and in particular streams of images. In the following, we first give some basic definitions adopted from Image Algebra [23] and then extend these definitions to account for fundamental properties of (streaming) geospatial image data.

Roughly speaking, in Image Algebra, an image consists of two things, a *set of points* in some n-dimensional space and a *set of values* associated with points.

Definition 1 (Point Set). *A* point set *is some topological space, consisting of* points *and a* topology *that provides for notions such as distance between two points and neighborhood of a point.*

Typical point sets are discrete subsets of the n-dimensional Euclidean space $\mathbb{R}^n$, together with a discrete topology that provides for a metric space. With points, values from a value set can be associated.

Definition 2 (Value Set). *A* value set $\mathbb{V}$ *is an instance of a homogeneous algebra, that is, a set of values together with a set of operands.*

In Image Algebra there is no application specific semantics associated with points sets, components of points, or point values. For the processing of streaming geospatial image data, however, such a semantics is crucial to build an expressive query model and processing framework. For this, we consider point sets of the form $\mathbf{X} = \mathbf{S} \times \mathbf{T}$, where $\mathbf{S}$ is the *spatial* domain, e.g., $\mathbf{S} = \mathbb{R}^2$ or $\mathbf{S} = \mathbb{R}^3$, and $\mathbf{T}$ is the temporal domain of the point set, typically $\mathbf{T} = \mathbb{R}$.

Each point $\mathbf{x}$ in a point set $\mathbf{X}$ is of the form $\mathbf{x} = \langle s, t \rangle$, where $s \in \mathbf{S}$ describes the *spatial location* of the point, and $t \in \mathbf{T}$ is a *timestamp*. The timestamp t specifies a logical point in time when the value of that point has been obtained. This can be the point in time when the value actually has been measured or the identifier of a satellite scan sector to which the point $\mathbf{x}$ belongs.

Moreover, we only consider point sets $\mathbf{X}$ whose spatial domain is a *regularly-spaced lattice* in $\mathbb{R}^n$, thus providing a *spatial resolution* pertinent to $\mathbf{X}$. For brevity, throughout the paper we will use the term *point lattice* to indicate a point set with the above restrictions. Point lattices exhibit fundamental characteristics of spatio-temporal data and allow for standard vector space and point operations. In particular, they provide a basis for a formal definition of a stream.

Definition 3 (Stream). *Given a point lattice* $\mathbf{X}$ *and a value set* $\mathbb{V}$*, a* $\mathbb{V}$*-valued stream* $\mathbf{G}$ *is a function* $\mathbf{G}\colon \mathbf{X} \to \mathbb{V}$ *that maps points from the point lattice* $\mathbf{X}$ *to values from the value set* $\mathbb{V}$*.*

Let $\mathbb{V}^{\mathbf{X}}$ denote the set of all functions from a point lattice $\mathbf{X}$ to a value set $\mathbb{V}$. Besides the functional notation of a stream $\mathbf{G} \in \mathbb{V}^{\mathbf{X}}$, in the following we also use the set notation $\mathbf{G} = \{(\mathbf{x}, \mathbf{G}(\mathbf{x})) \colon \mathbf{x} \in \mathbf{X}\}$, where $\mathbf{x}$ denotes the *spatio-temporal point location* and $\mathbf{G}(\mathbf{x})$ denotes the *value* at location $\mathbf{x}$.

Definition 4 (Image). *An* image *of a stream* $\mathbf{G}$ *is a subset* $\mathbf{i} \subset \mathbf{G}$ *whose points all have the same timestamp.*

A *raster image* consisting of a rectangular grid of pixels is a typical instance of an image. In a raster image all points (pixels) in the point lattice have the same timestamp value, and point values are taken from, e.g., $\mathbb{Z}$ (for grey-scale images), $\mathbb{Z}^3$ (for color images), or $\mathbb{Z}^n, n > 3$, for multi-spectral images.

To support geographic applications, there is one important property of streams and images that needs to be recognized. With every point lattice $\mathbf{X}$, or more precisely its spatial component $\mathbf{S}$, a *coordinate system* must be associated. A coordinate system, such as latitude/longitude or Universal Transverse Mercator (UTM), provides the basis for mapping points to pairs of numbers and vice versa. As we will see in Section 3, one precondition for applying operations on pairs of image data is that their point lattices are based on the same coordinate system. Note that the spatial resolution of two point lattices still can be different, although they are based on the same coordinate system.

Based on the above notions, we define the concept of streaming geospatial image data, called a GeoStream, as follows.

Definition 5 (GeoStream). *A stream* $\mathbf{G} \in \mathbb{V}^{\mathbf{X}}, \mathbf{X} = \mathbf{S} \times \mathbf{T}$, *is a* GeoStream *if a* coordinate system *is associated with the spatial component* $\mathbf{S}$.

In general, a GeoStream $\mathbf{G}$ is homogeneous in the sense that all points are based on the same coordinate system and that all points have the same spatial resolution. Yet there is no restriction on the shape or orderliness that points in a GeoStream can take. For the definition of operations on one or more GeoStreams, however, it is important to recognize typical point organizations that result from different remote-sensing instruments. Figure 1 illustrates this aspect.

Airborne cameras typically obtain data in an image-by-image fashion, as shown in Fig. 1(a). That is, there are several consecutive frames that cover possibly different spatial regions. Most satellite instruments obtain data in a row-by-row fashion where strips of image data arrive at a time, shown in Fig. 1(b). In this case, a single line of neighboring points constitutes a frame. Some instruments, such as LIDAR [16], have non-uniform point lattice structures, and points are only ordered by time, as shown in the Fig. 1(c).

An important feature of the GeoStreams data model that does not have a counterpart in traditional (relational) stream processing frameworks is that *consecutive points in a GeoStream have a close spatial proximity.* This is true except for the case where the last point of one frame is followed by the first point of a new frame (as shown in in Fig. 1(a)) that covers a different spatial region. In this case there is only a close temporal proximity between points. This feature has a significant impact on how operators are realized on one or more GeoStreams, as we will show in the following section.

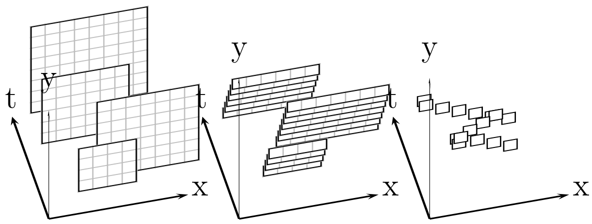

Fig. 1. Point set organization in a GeoStream for different remote-sensing instruments: (a) image-by-image (left); (b) row-by-row (middle); (c) point-by-point (right)

3 Query Model

In the following, we present a set of operators that, together with the concept of GeoStreams, build a basic algebraic model for querying streaming geospatial image data. Our primary focus is on operators that are used in typical geographic applications to derive different data products from geospatial image data. We put less emphasis on image processing operators, such as linear image transforms (e.g., Fourier and Wavelet transforms) or shape and pattern detection (see, e.g., [23]), which are often applied as post-processing steps to individual frames and images rather than streams of geospatial image data.

An important feature of our query algebra is that it is *closed*. That is, the result of applying an operator to one or two GeoStreams is again a GeoStream. This allows the formulation of complex queries over streaming geospatial image data, and it also provides a basis for query optimization techniques, such as query rewriting. In the following section, we present three classes of stream restriction operators, followed by stream transform operators in Section 3.2. Both types of operators operate on a single GeoStream. In Section 3.3, we then discuss stream composition operators. For the operators introduced, we also discuss their cost and practical realization. In Section 3.4, we briefly illustrate formulation techniques for complex queries and query rewriting.

3.1 Stream Restrictions

Assume a GeoStream $\mathbf{G} \in \mathbb{V}^{\mathbf{X}}$, $\mathbf{X} = \mathbf{S} \times \mathbf{T}$. A restriction operator can be thought of as a filter that only selects points from a stream that satisfy a certain condition on the spatial, temporal, or point value component. For a point $\mathbf{x} \in \mathbf{X}$, we denote these components $\mathbf{x}.s$, $\mathbf{x}.t$, and $\mathbf{G}(\mathbf{x})$, respectively.

The most important type of restriction and frequently used operator in typical queries is a *spatial restriction.*

Definition 6 (Spatial Restriction). *Given a point lattice* $\mathbf{R} \subset \mathbf{S}$*, the* spatial restriction *of* $\mathbf{G}$ *to* $\mathbf{R}$*, denoted* $\mathbf{G}_{|\mathbf{R}}$*, is defined as* $\mathbf{G}_{|\mathbf{R}} := \{(\mathbf{x}, \mathbf{G}(\mathbf{x})) \colon \mathbf{x} \in \mathbf{G} \ \wedge \ \mathbf{x}.s \in \mathbf{R}\}$.

Although many remote-sensing instruments may cover large regions, users and applications are often only concerned with particular *regions of interest.* A spatial restriction allows to filter only those incoming point data that are spatially located in the region $\mathbf{R}$ of interest. Conceptually, there are several ways in which $\mathbf{R}$ can be specified: (1) as an enumeration of all x, y value pairs in $\mathbf{R} \subset \mathbf{S}$ (assuming a 2-dimensional space), (2) expressions of a constraint data model, i.e., polynomials on variables x, y [22, Chap. 4], or (3) by specifying two corner points of a rectangle that describes the bounding box of the points of interest. In practice, approach (3) is commonly used in graphical user interfaces.

A temporal restriction operator is defined as follows.

Definition 7 (Temporal Restriction). *Let* $\mathbf{T} \subset \mathbb{R}$ *be a set of timestamps. The* temporal restriction *of* $\mathbf{G}$ *to* $\mathbf{T}$*, denoted* $\mathbf{G}_{|\mathbf{T}}$*, is defined as*

$$\mathbf{G}_{|\mathbf{T}} := \{(\mathbf{x}, \mathbf{G}(\mathbf{x})) \colon \mathbf{x} \in \mathbf{G} \ \wedge \ \mathbf{x}.t \in \mathbf{T}\}$$

For this operator, too, there are several ways in which $\mathbf{T}$ can be described: as a collection of points in time, as an open interval or as a set of (re-occurring) intervals, e.g., if an application requires only data during a specific time period every day. Finally, a value restriction operator, denoted $\mathbf{G}_{|\mathbf{V}}$, over a set $\mathbf{V} \subseteq \mathbb{V}$ of point values is defined as $\mathbf{G}_{|\mathbf{V}} \equiv \{(\mathbf{x}, \mathbf{G}(\mathbf{x})) \colon \mathbf{G}(\mathbf{x}) \in \mathbf{V}\}$.

It is obvious that all three restriction operators can process incoming image data on a point-by-point basis and thus can be evaluated without storage for any intermediate point data. That is, all restriction operators are non-blocking and have constant cost per point, independent of the size of the input stream.

3.2 Stream Transforms

Assume again a GeoStream $\mathbf{G}$ over point lattice $\mathbf{X}$ and value set $\mathbb{V}$. Conceptually, a transform operator maps the point or value set associated with $\mathbf{G}$ to a new point and value set. There are two types of transforms: *value transforms* and *spatial transform.* We start with the simpler one, the value transform.

Definition 8 (Value Transform). *Given a function* $f_{val} \colon \mathbb{V} \to \mathbb{W}$*, with* $\mathbb{V}, \mathbb{W}$ *being value sets, a* value transform*, denoted* $f_{val} \circ \mathbf{G}$*, changes a stream over* $\mathbb{V}^{\mathbf{X}}$ *to a stream over* $\mathbb{W}^{\mathbf{X}}$*, and is defined as* $f_{val} \circ \mathbf{G} := \{(\mathbf{x}, f_{val}(\mathbf{G}(\mathbf{x}))) \colon \mathbf{x} \in \mathbf{X}\}$.

A simple form of a value transform operator is one that transforms color point values with $\mathbb{V} \subset \mathbb{Z}^3$ to gray-scale point values with $\mathbb{W} \subset \mathbb{Z}$. Clearly, such an operator allows for processing on a point-by-point basis. However, not all value transform operators show such a behavior. For example, in order to fully utilize the complete range of values in $\mathbb{V}$, point values can be *scaled.* Typical approaches include linear contrast stretch, histogram equalization, and Gaussian stretch [19]. In order to perform a respective value transform on a point, information about previous point values needs to be maintained, in particular the minimum and maximum point values seen so far. In the context of streaming image data, this is typically done on individual frames of the stream $\mathbf{G}$, and not the complete stream. If a frame has a large number of points, all points of that frame need to be stored before they can be output with new point values. Thus, the cost of a "stretch" transform operator is determined by the size of the largest frame that can occur in $\mathbf{G}$. For most satellites and satellite imagery, such frame sizes are known (e.g., for GOES, the maximum frame size is about 20,840 by 10,820 points for the visible band at $1km$ resolution, requiring approx. 280MB storage).

An important type of operator for processing streaming geospatial image data is a spatial transforms. They allow for magnification (zooming), rotation, and general affine transformations. For geographic point lattices with associated coordinate system, re-projection of points into a new coordinate system is also a transformation.

Definition 9 (Spatial Transform). *Given a function* $f_{spat}\colon \mathbf{Y} \rightarrow 2^{\mathbf{X}}$*, with* $\mathbf{X}, \mathbf{Y}$ *being point lattices. A* spatial transform*, denoted* $\mathbf{G} \circ f_{spat}$*, changes a stream over* $\mathbb{V}^{\mathbf{X}}$ *to a stream over* $\mathbb{V}^{\mathbf{Y}}$*, and is defined as*

$$\mathbf{G} \circ f_{spat} := \{(\mathbf{y}, \mathbf{G}(f_{spat}(\mathbf{y})))\colon \mathbf{y} \in \mathbf{Y}\}.$$

How does a spatial transform operator actually work on a GeoStream? Assume a scenario where image data from $\mathbf{G}$ is coming in and one wants to change the spatial resolution associated with the point lattice $\mathbf{X}$. An operator that increases the spatial resolution would take an incoming point $\mathbf{x}$ and produce a rectangular lattice of $k \times k$ (k being the magnification factor) of points in $\mathbf{Y}$, all with the point value $\mathbf{G}(\mathbf{x})$. No neighboring points for $\mathbf{x}$ are required to accomplish this transform, and thus the spatial transform actually would be of the form $f_{spat}\colon \mathbf{Y} \rightarrow \mathbf{X}$. However, neighboring points are needed in case one wants to decrease the resolution. For a point $\mathbf{x} \in \mathbf{X}$, a rectangular lattice of $k \times k$ neighboring points "surrounding" $\mathbf{x}$ is needed to compute the value $\mathbf{G}(f_{spat}(\mathbf{y}))$ of a point $\mathbf{y} \in \mathbf{Y}$ (see Fig. 2(a)). Thus, the operator has to buffer a sufficient number of points in $\mathbf{X}$ in order to compute the value of a point $\mathbf{y} \in \mathbf{Y}$.

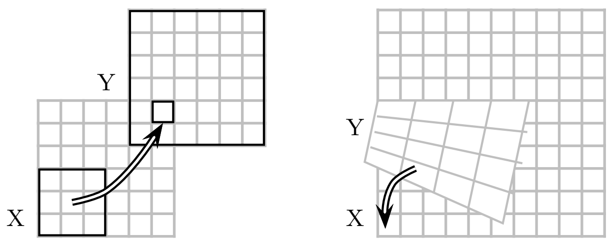

Fig. 2. Spatial transform operators: (a) decreasing the spatial resolution by $\frac{1}{3}$ (left); (b) re-projection to a new coordinate system (right)

Without any knowledge about the point lattice organization in $\mathbf{G}$, such an operator could *potentially block forever,* e.g., when the rightmost lower point in a frame has been received and no suitable neighboring points follow. In an implementation, such a scenario (which in general can occur for any point in a frame) can be avoided by utilizing auxiliary information about the spatial region currently scanned by an instrument and added as metadata to the stream of image data. The operator then can use such metadata and execute appropriate boundary point interpolations.

From a geographic application point of view, an important functionality is to re-project geospatial data from one coordinate system to another one [11,26]. One can think of a re-projection as a mathematical framework that specifies for every point $\mathbf{y} \in \mathbf{Y}$ what points in $\mathbf{X}$ are necessary to compute $\mathbf{y}$ and its point value. Often, the transformation strives for an approximate one-to-one correspondence between points in point lattices $\mathbf{X}$ and $\mathbf{Y}$, and a regular lattice corresponding in size and aspect to the lattice of the original point set $\mathbf{X}$ is overlayed over the spatial extent of the new point lattice. In traditional GIS applications, for a point $\mathbf{y} \in \mathbf{Y}$, either the nearest point in the original point lattice is chosen to supply the point value, or a function is applied to a neighborhood of pixels of $\mathbf{x}$ to provide the point value. Respective functions f_{spat} include linear interpolations or higher-order fitting routines. In general, there is no single best solution, considering the multitude of coordinate systems and types of re-projections.

From a query processing point of view, it is important to note that such types of spatial transform operators may block for a considerable amount of time, as the computation of the value of a point $\mathbf{y} \in \mathbf{Y}$ may require any number of points from $\mathbf{X}$. An implementation of individual operators corresponding to specific spatial transform and re-projections, however, can be again tailored by utilizing metadata about the spatial extent of the current scan sector and the spatial resolution associated with $\mathbf{X}$ and $\mathbf{Y}$.

3.3 Stream Compositions

Geographic applications often derive data products by combining information from different spectral bands. For example, the normalized difference vegetation index (NDVI) describing the health of vegetation combines pixel data from the near-infrared and visible band to compute a single NDVI value for a spatial location. In our model, combining image data from different spectral bands is realized through a generic stream composition operator where each stream represents a single spectral band.

Definition 10 (Stream composition). *Let* $\mathbf{G}_1, \mathbf{G}_2$ *be two streams over a point lattice* $\mathbf{X}$ *and value set* $\mathbb{V}$. *A* stream composition operator γ *over* $\mathbf{G}_1, \mathbf{G}_2$, *denoted* $\mathbf{G}_1 \ \gamma \ \mathbf{G}_2$ *is a binary operator defined as*

$$\mathbf{G}_1 \ \gamma \ \mathbf{G}_2 \equiv \{(\mathbf{x}, \mathbf{G}_1(\mathbf{x}) \ \gamma \ \mathbf{G}_2(\mathbf{x}))\colon \mathbf{x} \in \mathbf{X}\}.$$

Typical stream composition operators include addition, difference, division, supremum and infimum, i.e., $\gamma \in \{+, -, \div, \vee, \wedge\}$. There are several important observations regarding the behavior of these operators. First, although both streams are based on the same point lattice $\mathbf{X}$, it can happen that there is no single point that occurs in both streams. This obviously is the case when the two streams cover different spatial regions.

Second, and more important for practical applications, note that in order to apply γ to two point values $\mathbf{G}_1(\mathbf{x})$ and $\mathbf{G}_2(\mathbf{x})$, the points must match in the spatial dimension and in the timestamp. This has considerable consequences on how timestamping for points is realized and, consequently, how much intermediate image data an operator needs to store in order to output new image data.

For example, assume a satellite that scans a spatial region first for the visible band and then for the near-infrared band. If incoming points are timestamped based on when the points were measured, a stream composition operator would never produce new image data as respective timestamps would never match. That is why in practice, point data is timestamped using *scan-sector identifiers*: a satellite scans a spatial region for different spectral bands, each band resulting in a single GeoStream. A point in each of the imagery obtained for the spectral bands for the scan sector is assigned the scan-sector identifier as timestamp, facilitating the comparison of point data from different bands and streams, respectively.

Finally, although for a single scan, all point data from the different streams have the same timestamp, the space complexity of a stream composition operator depends on the point organization in which the image data is transmitted (see also Fig.1). If the data is transmitted on an image-by-image basis, the operator has to buffer a complete image whereas for a row-by-row organization, it only has to buffer a single row of one stream before it can combine it with a matching row from another stream.

In summary, the realization of a stream composition operator, which conceptually might seem straightforward, very much depends on the scan characteristics of the remote-sensing instrument that generates the image data streams.

3.4 Complex Queries and Query Rewriting

In geographic applications, data products are typically obtained by applying a sequence of operators to imagery. Our query model and the closure property of operators in particular naturally facilitate the formulation of complex queries. However, unlike queries in a relational database context where queries can have complex nested subqueries, continuous queries over streams of remotely-sensed image data typically tend to be less complex; in fact, they are often "sequential".

Due to space constraints we cannot describe all query composition and rewriting techniques for the previously presented operators, but we illustrate a few key aspects of our query model by means of an example. Assume two GeoStreams $\mathbf{G}_1$ and $\mathbf{G}_2$ corresponding to the near-infrared and visible band of a satellite instrument. Consider the following query

$$((f_{val} \circ ((\mathbf{G}_1 - \mathbf{G}_2) \div (\mathbf{G}_2 + \mathbf{G}_1))) \circ f_{UTM})_{|\mathbf{R}}$$

which can be read as follows: (1) compute the NDVI over streams $\mathbf{G}_1, \mathbf{G}$ as stream composition $((\mathbf{G}_1 - \mathbf{G}_2) \div (\mathbf{G}_2 + \mathbf{G}_1))$, (2) perform a value transform f_{val} on the result point lattice, (3) re-project to the UTM coordinate system (f_{UTM}), and finally (4) select only point data for the region $\mathbf{R}$ of interest. Assume $\mathbf{G}_1$ and $\mathbf{G}_2$ are based on a coordinate system C. Rather than performing the composition of *all point data* from the two streams, followed by a value and spatial transform on all the resulting points, the final spatial restriction $_{|\mathbf{R}}$ can be pushed "inwards" and applied first to $\mathbf{G}_1$ and $\mathbf{G}_2$ before any composition.

However, because in the query **R** is based on the UTM coordinate system, **R** needs to be mapped to the coordinate system C. The query optimizer has to identify such rewrites in particular for spatial selections, as these result in the most significantly space and time gains for query evaluation.

4 Query Processing Framework

Some of the operators introduced in Section 3 have been realized in the context of the *GeoStreams* project [6]. In the following, we give a brief overview of the prototypical system being developed in this project to illustrate the data flow and stream processing components.

Remotely-sensed imagery from the GOES satellites [8] is received by the Data Stream Management System (DSMS) server, and the raw data is converted by the stream generator into GeoStream point lattices that have a row-by-row organization. The streams correspond to the different spectral channels that are generated by the imager instrument on-board the GOES satellite. In the DSMS server, a spatial transform operator converts the GeoStream point sets, which come in a satellite specific coordinate system (called GOES Variable Format), into point lattices based on latitude/longitude. Multiple users can connect to the DSMS server and formulate queries over the GOES data streams generated within the DSMS. Users use a Web-based graphical interface to specify spectral channels (streams) of interest, regions of interest, and certain spatial transforms (e.g., zooming). The coordinate system used in this interface is latitude/longitude.

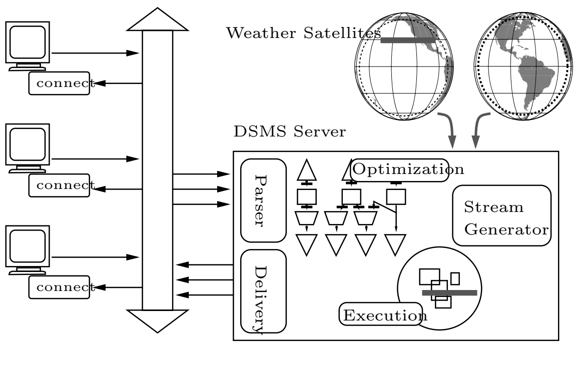

Fig. 3. Architecture of the Stream Management System for Geospatial Image Data

User queries, which are converted by the interface to specialized HTTP requests, are transmitted to the server, parsed, and registered. Multiple queries against a single GeoStream are optimized using a dynamic cascade tree structure [10], which acts as a single spatial restriction operator and efficiently streams only the point data of interest to current continuous queries to subsequent operators. As indicated in Section 3.4, optimizing queries with respect to regions of interest has the greatest benefit. This spatial restriction operator then streams the point data to a specialized stream delivery operator that ships stream results back to clients using the PNG image format. Other operators that are currently being implemented and integrated with the query execution engine include different types of re-projections (using extensions to PROJ.4 [21]) and specialized macro operators that compute specific data products, such as NDVI. Such data products can be directly selected in the user interface, without the need to compose otherwise complex queries.

5 Related Work

Our work borrows several fundamental concepts from Ritter and Wilson's Image Algebra [23], in particular the functional representation of images and some operations on images. Our data and query model provides an extension of these concepts in that we (1) consider streams of image data instead of (static) images and (2) explicitly introduce the notion of spatio-temporal data, which is geo-referenced, and specialized operations on geo-referenced data.

Query processing techniques for multi-dimensional arrays have been studied by Marathe and Salem [18] and Libkin et al. [14]. General types of operations on raster image data, which can be considered as a specialized form of arrays, have been proposed by Baumann [2,3,4]. While these works study frameworks for operations and query processing on array data and raster images, they do not consider the aspect of streaming geospatial image data. The operators are not specific to streaming spatio-temporal data and in particular do not consider the specifics of typical computations performed on remotely-sensed imagery. As we have illustrated in Section 3, knowing about point lattice organizations in a stream of image data and properties of satellite scan sectors can have a significant impact on how image data is processed.

A work closely related to ours is the one by Mokbel et al. [17] on continuous query processing of spatio-temporal data streams in the context of pervasive location-aware computing environments. In their work, they primarily focus on object-based spatio-tempporal data (moving objects), whereas our approach exclusively focuses on field-based spatio-temporal data, in particular satellite imagery and its specific operators.

The large body of work on spatial and spatio-temporal data (for an overview, see, e.g., [22,24]) in general neither considers streaming geospatial image data nor remotely-sensed data.

6 Conclusions and Outlook

Remotely-sensed imagery from the numerous satellites orbiting the Earth provides a great opportunity to develop novel data stream management and processing systems, leveraging techniques and models developed for relational data streams to improve processing streaming geospatial image data. In this paper, we have presented the foundation of a data and query model that allows the formulation and processing of continuous queries over streams of such image data. In particular, we have discussed the specifics of stream restriction, transform, and composition operators with a focus on the properties of remotely-sensed data and processing techniques relevant for typical geographic applications.

We are currently extending the set of operators, with a particular focus on spatial transforms, because they represent the most demanding types of operators in terms of space and time complexity. We are also investigating the full integration of a spatio-temporal aggregate operator for streaming image data. This operator has been proposed in [27], and will provide an important addition to the functionality of our stream management system.

References

1. A. Arasu, S. Babu, J. Widom: The CQL Continuous Query Language: Semantic Foundations and Query Execution. The VLDB Journal 15(2):121–142, 2006.
2. P. Baumann: Management of Multidimensional Discrete Data. The VLDB Journal 3(4):401–444, 1994.
3. P. Baumann: Database Array Algebra for Spatio-Temporal Data and Beyond. In *Next Generation Information Technologies and Systems, 4th International Workshop (NGITS'99)*, LNCS 1649, Springer, 76–93, 1999.
4. P. Baumann, A. Dehmel, P. Furtado, R. Ritsch, N. Widmann: Spatio-Temporal Retrieval with RasDaMan. In *Proceedings of 25th International Conference on Very Large Data Bases*, 746–749, 1999.
5. B. Babcock, S. Babu, M. Datar, R. Motwani, J. Widom: Models and Issues in Data Stream Systems. In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002)*, 1–16, 2002.
6. GeoStreams: Adaptive Query Processing Architectures for Streaming Geospatial Image Data. University of California at Davis, `http://geostreams.ucdavis.edu`.
7. L. Golab, M. T. Özsu: Issues in Data Stream Management. SIGMOD Record 32(2):5–14, 2003.
8. NOAA Satellite and Information Service. Geostationary Satellite Server. http://www.goes.noaa.gov.
9. Q. Hart, M. Gertz: Querying Streaming Geospatial Image Data: The GeoStreams Project. In *17th International Conference on Scientific and Statistical Database Management (SSDBM 2005)*, 2005.
10. Q. Hart, M. Gertz, J. Zhang: Evaluation of a Dynamic Tree Structure for Indexing Query Regions on Streaming Geospatial Data. In *9th International Symposium on Spatial and Temporal Databases (SSTD05)*, LNCS 3633, Springer, 145–162, 2005.
11. J. C. Iliffe: Datums and Map Projections. CRC Press, 2000.
12. J. R. Jensen: Remote Sensing of the Environment: An Earth Resource Perspective. Prentice-Hall, 2000.
13. J. R. Jensen: Introductory Digital Image Processing: A Remote Sensing Perspective (3rd Edition). Prentice-Hall, 2005.
14. L. Libkin, R. Machlin, L. Wong. A Query Language for Multidimensional Arrays: Design, Implementation, and Optimization Techniques. In *Proc. of the 1996 SIGMOD International Conference on Management of Data*, 228–239, 1996.
15. Landsat 7, `http://landsat.gsfc.nasa.gov`.
16. LIDAR Tutorial. `www.ghcc.msfc.nasa.gov/sparcle/sparcle_tutorial.html`.

17. M.F. Mokbel, X. Xiong, M.A. Hammad, W.G. Aref: Continuous Query Processing of Spatio-temporal Data Streams in PLACE. GeoInformatica 9(4):343–365, 2005.
18. A. P. Marathe, K. Salem: Query Processing Techniques for Arrays. The VLDB Journal 11(1):68–91, 2002.
19. P. M. Mather: *Computer Processing of Remotely-Sensed Images*. Wiley, 2004.
20. Moderate Resolution Imaging Spectroradiometer (MODIS). `http://modis.gsfc.nasa.gov`.
21. PROJ.4 - Cartographic Projections Library. `http://proj.maptools.org`.
22. P. Rigaux, M. Scholl, A. Voisard: *Spatial Databases with Application to GIS*. Morgan Kaufmann, 2002.
23. G. X. Ritter, J. N. Wilson: *Handbook of Computer Vision Algorithms in Image Algebra (2nd Edition)*. CRC Press, 2001.
24. S. Shekar, S. Chawla: Spatial Databases: A Tour. Prentice Hall, 2003.
25. A. Skidmore (Editor): *Environmental Modeling with GIS and Remote Sensing*. Taylor & Francis, 2002.
26. J. P. Snyder, W. R. Tobler, O.H. Yang, J.P. Snyder, Q.H. Yang: Map Projection Transformation: Principles and Applications. CRC Press, 2000.
27. J. Zhang, M. Gertz, D. Aksoy: Spatio-Temporal Aggregates over Raster Image Data. In *12th ACM International Workshop on Geographic Information Systems (ACM-GIS 2004)*, 39–46, 2004.