A Framework for Moving Sensor Data Query and Retrieval of Dynamic Atmospheric Events

Shen-Shyang Ho[1], Wenqing Tang[1], W. Timothy Liu[1], and Markus Schneider[2]

[1] Jet Propulsion Laboratory, California Institute of Technology,
Pasadena CA 91109, USA,
{sho, Wenqing.Tang, w.t.liu@jpl.nasa.gov},
[2] Department of Computer & Information Science & Engineering,
University of Florida, Gainesville, FL 32611, USA
mschneid@cise.ufl.edu

Abstract. One challenge in Earth science research is the accurate and efficient ad-hoc query and retrieval of Earth science satellite sensor data based on user-defined criteria to study and analyze atmospheric events such as tropical cyclones. The problem can be formulated as a spatio-temporal join query to identify the spatio-temporal location where moving sensor objects and dynamic atmospheric event objects intersect, either precisely or within a user-defined proximity. In this paper, we describe an efficient query and retrieval framework to handle the problem of identifying the spatio-temporal intersecting positions for satellite sensor data retrieval. We demonstrate the effectiveness of our proposed framework using sensor measurements from QuikSCAT (wind field measurement) and TRMM (precipitation vertical profile measurements) satellites, and the trajectories of the tropical cyclones occurring in the North Atlantic Ocean in 2009.

Key words: data retrieval, satellite data, atmospheric events, spatio-temporal join

1 Introduction

The Earth Observing System Data and Information System (EOSDIS)[3] is a comprehensive data and information system which archives, manages, and distributes Earth science data from the EOS spacecrafts (a.k.a. satellite sensors) [1]. A challenge of EOSDIS is how to "help users find the data that they need and how to get it to them" [2]. The Warehouse Inventory Search Tool (WIST)[4]

This work was partially carried out at the Jet Propulsion Laboratory, California Institute of Technology and was funded by the National Aeronautics and Space Adminstration (NASA) Advanced Information Systems Technology (AIST) Program under grant number AIST-08-0081.

[3] http://esdis.eosdis.nasa.gov

[4] https://wist.echo.nasa.gov/~wist/api/imswelcome/

is the primary search and order tool for Earth Science data sets for EOSDIS. It allows users to browse and retrieve satellite measurements based on user-defined spatial and temporal conditions. This type of data query and retrieval is known in the Earth science community as data "subseting". One important use of the retrieved satellite sensor data is the improvement of weather forecasting such as the use of QuikSCAT wind measurements to accurately depict the initial conditions of air and sea states for tropical cyclone forecast model [3].

In the mid-nineties, there was an ambitious project to develop a "flexible, extensible, and seamless SCF [Scientific Computing Facilities] for scientific data analysis, knowledge discovery, visualization, and collaboration" called the Open Architecture Scientific Information System (OASIS) to support EOSDIS based on the Common Object Request Broker Architecture (CORBA) [4]. The OASIS was not embraced by the scientific community which could have been the result of serious technical, complexity, and security issues related to CORBA [5].

Currently, there is still a lack of capabilities that support flexible data retrieval in the EOSDIS. One non-existent capability is the accurate and efficient ad-hoc query and retrieval of Earth science satellite sensor data for dynamic atmospheric events such as tropical cyclones based on ad-hoc user-defined criteria and event trajectories. In this paper, we describe a fast data query and retrieval framework based on a spatio-temporal partitioning scheme driven by the partitioning of the moving satellite trajectory so that the positions which the satellite trajectory and an atmospheric event trajectory intersect, either precisely or within close proximity, are used for satellite data retrieval. We demonstrate the feasibility of our framework on the tropical cyclone event which is a "non-frontal synoptic scale low-pressure system over tropical or sub-tropical waters with organized convection and definite cyclonic surface wind circulation"[5]. Experimental results are used to show the effectiveness of our proposed framework using sensor measurements from QuikSCAT (wind field measurement) and TRMM (precipitation vertical profile measurements) satellites, and the tropical cyclones occurring in the North Atlantic Ocean in 2009.

From published scientific journal papers [6–10], one observes that such a capability is extremely important to scientists who retrieve specific sensor data of specific atmospheric events for statistical analysis. Some query examples derived from these published scientific papers that require search, retrieval, and analysis of satellite data containing cyclone features, are listed below:

1. Retrieve TRMM precipitation data for tropical cyclones that attained tropical storm intensity or higher over western North Pacific and the South China Sea between longitudes $100^o E$ and 180^o. 138 sensor datasets from 61 tropical cyclones retrieved [6].
2. Retrieve TRMM precipitation data for tropical cyclones from December 1997 to December 2003. 3703 sensor datasets from 563 tropical cyclones retrieved [7].

[5] `http://www.aoml.noaa.gov/hrd/tcfaq/A1.html`

3. Retrieve QuikSCAT wind data for tropical cyclones in western North Pacific from September 1999 to December 2004 which formed west of $160^o E$ and south of $26^o N$. Datasets containing 124 tropical cyclones retrieved [8].

Our problem is fundamentally different from previous research to discover and track cyclones from either sea-level pressure fields [11] or from heterogeneous satellite data [12]. For our problem, the cyclone tracks are known. Our main contribution is an efficient framework that enables the retrieval of satellite data based on known cyclone tracks, an approach to fuse two databases with widely different characteristics.

The paper is organized as follows. In Section 2, we briefly review previous research and systems developed for satellite data query and retrieval, in particular, for the tropical cyclone events. In Section 3, the satellite sensor data query and retrieval problem is defined. In Section 4, the satellite data and tropical cyclone event trajectory data are briefly described. In Section 5, the satellite sensor trajectory data partitioning scheme and partition search algorithm are described in detail. In Section 6, the satellite data retrieval algorithm is described in detail. In Section 7, experimental results are presented to demonstrate the feasibility of our proposed framework for both QuikSCAT and TRMM satellite sensor data. Some visualizations of the retrieved satellite data sets from a queried hurricane trajectory are also shown.

2 Related Work

Existing state-of-the-art publicly available web-based tropical cyclone data and information portals[6] [7], data archives [8], and forecast services [9] provide excellent visualizations and information of tropical cyclones and satellite sensor measurements.

However, comfortable data access (e.g., ad-hoc data retrieval for specific weather events) is not provided, and users only have limited, simple, and hard-coded query and request capabilities. Examples of such queries are:

1. Provide specific satellite data of a specified region at a specific date and time. [EOSDIS]
2. Provide the static dataset for a specific tropical cyclone event. [Physical Oceanography DAAC: Hurricane/Typhoon Tracker]

Users are not able to perform own queries to retrieve satellite data based on arbitrary trajectory information and retrieval parameters.

[6] Navy/NRL Tropical Cyclone. `http://www.nrlmry.navy.mil/tc_pages/tc_home.html`
[7] NASA GSFC Hurricane Portal. `http://daac.gsfc.nasa.gov/hurricane/`
[8] Physical Oceanography DAAC Hurricane/Typhoon Tracker. `http://podaac.jpl.nasa.gov/hurricanes/`
[9] NOAA National Hurricane Center. `http://www.nhc.noaa.gov/pastall.shtml`

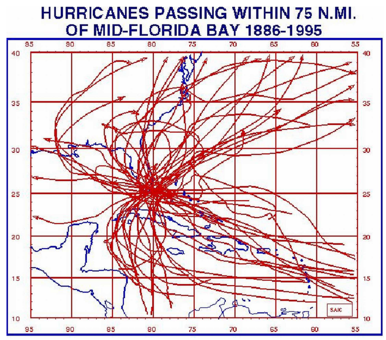

Fig. 1. Visualization of the output from the query "Find all hurricanes from 1886 to 1996 within 75 nautical miles of mid-Florida Bay" (http://www.aoml.noaa.gov/hrd/Storm_pages/fl_track_red.html).

Many spatio-temporal access methods (for indexing historical spatio-temporal data) have been developed [13, 14] (and references therein) to support certain query types or to support efficiently as many query types as possible. Some of the common query types are

1. Selection: Find all objects within a specific region and/or during a specific time interval.
2. Join: Find all objects that are spatially close during a specific time interval.
3. Nearest Neighbor: Find the k-closest objects with respect to a specific region and/or time interval.

These queries are of interest to scientists studying tropical cyclones An example of a selection query is "Find all hurricanes [10] from 1886 to 1996 within 75 nautical miles of mid-Florida Bay" and its output shown in Fig. 1.

In this paper, we are, however, interested in exploring the intersection of two different object classes (hurricanes and satellite trajectories) by a query such as "Find the spatial region(s) R and time interval(s) I such that the hurricane path is either in the satellite sensor scanning region or within some user-defined distance outside the boundary of the satellite sensor scanning region" and then to use its output for data retrieval.

3 Problem Definition

Consider the set of satellite sensors, $O_s = \{O_{s1}, O_{s2}, \ldots, O_{sk}\}$, and the set of atmospheric events, $O_c = \{O_{c1}, O_{c2}, \ldots, O_{cm}\}$ such as the set of tropical cyclones. In particular, the query and retrieval problem of interest is "Find all *unique* satellite sensor measurements from O_{si} that is at most x kilometers from the tropical

[10] Hurricanes are tropical cyclones with sustained surface wind intensity equal or more than 119km/h.

Characteristics	O_s	O_c
Temporal Length	long (several years)	short (unlikely to be several months)
Temporal Resolution	fine grain (order of 10^{-1} seconds)	course grain (several hours)
Motion Speed	high (a full orbit is about 100 minutes)	low
Representation	line segments	points (can be extended to a region)
Spatial Position	continuous motion (orbiting; not geostationary satellites)	unlikely to be stationary, but possible
Data Updates/ Modifications	No delete; most current	No delete; historical, most current

Table 1. Differences in the characteristics between the satellite sensor objects and the tropical cyclone objects

cyclone path P of O_{cj} and in the time interval I." It can be generalized to "Find all *unique* satellite sensor measurements from satellite $O_{s1}, \ldots, O_{sk}$ that are at most x kilometers from the tropical cyclone paths $p_1, \ldots, p_m$ in region R at time interval I." This is closely related to the *spatio-temporal join* which retrieves all pairs of objects $< o_1, o_2 >$ with $o_1 \in O_s$ and $o_2 \in O_c$, $|o_1(t_q) - o_2(t_q)| \leq d$ where t_q is a time-stamp and d is an upper-bound threshold. Our problem goes further by querying for the positions and time instances where and when the join condition is satisfied. This condition is likely to be satisfied at multiple positions and time instances. An orbiting satellite sensor trajectory consists of many years of continuous spatio-temporal information. Hence, one needs to construct an efficient partitioning scheme to handle the lengthy data sequence. We construct the partitions by treating time as another dimension for a satellite sensor object. The tropical cyclone objects are stored in an index structure since there are some fundamental differences between the two object types. The differences in the characteristics between the two object types are shown in Table 1. For selection and nearest neighbor queries for objects in O_c, one can use an index structure such as TB-tree [15] or SEB-tree [16].

Let S be the spatial bound (latitude[min, max], longitude[min, max]) and T be the temporal bound time(start, end). Queries that return sensor objects and their intersecting spatio-temporal information such as

$$O_q = \{o_{si} \in O_s | O_s \cap_{ST} O_s \neq \emptyset \text{ within spatial bound } S \text{ and temporal bound } T\}$$
$$TS = \{(t, s) | t \in T, s \in S \text{ and } O_s \cap_{ST} O_s \neq \emptyset \text{ within spatial bound } S \text{ and temporal bound } T\}$$

Field	matrix size	Unit	Minimum	Maximum
wvc_lat	[nrow, ncol]	degree	-90.00	90.00
wvc_lon	[nrow, ncol]	degree E	0.00	359.99
selected speed	[nrow, ncol]	meter per second	0.00	50.00
selected direction	[nrow, ncol]	degree from North	0.00	359.99
wvc_row_time	[nrow]	Coordinated Universal Time (UTC)	1993-001 T00:00.000	2009-365 T23:59:59.999

Table 2. Relevant QuikSCAT data fields. nrow: number of rows; ncol: number of columns.

Field	Structure	Size
Scan Time	Table	$9\ bytes \times nscan$
Geo-location	Array	$2 \times npixel \times nscan$

Table 3. Relevant TRMM spatio-temporal data field. nscan: number of rows in the data matrix; npixel: number of column in the data matrix.

are not the focus of this paper as the intersections ($\cap_{ST}$[11]) of satellite sensor trajectories *alone* are not useful information for atmospheric, ocean, and weather event research. One is interested in

$$\begin{aligned} O_q &= \{o_{si} \in O_s | O_c \cap_{ST} O_s \neq \emptyset \text{ within spatial bound } S \text{ and temporal bound } T\} \\ TS &= \{(t,s) | t \in T, s \in S \text{ and } O_c \cap_{ST} O_s \neq \emptyset \text{ within spatial bound } S \text{ and} \\ &\quad \text{temporal bound } T\} \end{aligned} \tag{1}$$

The first one is a "Which" query such as a selection or nearest neighbor query. The latter one is a query which determines the positions and time instances where and when the trajectories of the objects in the two sets intersect, either precisely or within a certain proximity. In this paper, we focus on the latter query which can be derived from the first one and its outputs are applicable to our satellite data retrieval problem.

4 Data Description

In this paper, we use the Level 2B QuikSCAT wind field swath data and the Level 2A12 TRMM precipitation swath data stored in hierarchical data format (HDF)[12] to demonstrate the feasibility and efficiency of the partitioning scheme and the data retrieval framework. In Section 4.1, we give a brief description of the satellite data. In Section 4.2, we give a brief description of the tropical cyclone trajectories.

[11] $\cap_{ST}$ denotes the operation that returns the set of elements from the bigger set (usually O_c, if the two sets are different) when the trajectories of objects in O_c and O_s intersect. The simplest case is when $|O_c| = |O_s| = 1$.

[12] `http://www.hdfgroup.org/`

Name	Format	Description
Year	2-byte integer	4-digit year
Month	1-byte integer	The month of the Year
Day of Month	1-byte integer	The day of the Month
Hour	1-byte integer	The hour (UTC) of the Day
Minute	1-byte integer	The minute of the Hour
Second	1-byte integer	The second of the minute
Day of Year	2-byte integer	The day of the Year

Table 4. Scan Time

Name	Minimum	Maximum
Latitude	-90.00	90.00
Longitude	-179.99	180.00

Table 5. Geo-location. Off-Earth is represented by -9999.9

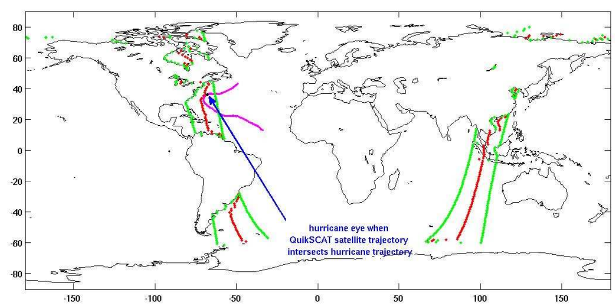

Fig. 2. One QuikSCAT swath intersecting path of Hurricane Irene in 2005.

4.1 Satellite Data

QuikSCAT. One QuikSCAT satellite full polar orbiting revolution takes about 101 minutes. The Level 2B data are grouped by rows of wind vector cells (WVC) which are squares of dimension 25 km or 12.5 km. A complete coverage of the earth circumference requires 1624 WVC rows at 25 km spatial resolution, and 3248 rows at 12.5 km spatial resolution. The width of the swath is 1800 km which amounts to seventy-two 25 km WVCs or one hundred and forty-four 12.5 km WVCs.

There are 25 fields in the data structure for Level 2B data [17]. We are, however, only interested in the latitude, longitude, time, and the most likely wind speed and direction for the WVCs. The fields that we are interested in are summarized in Table 2 and used in Algorithm 1 and 2. The QuikSCAT Level 2B data is obtained from the JPL Physical Oceanography DAAC (PO.DAAC) FTP

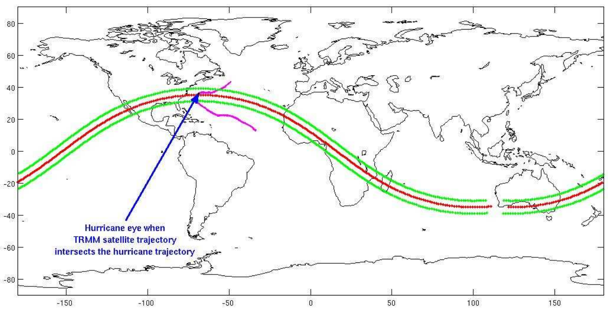

Fig. 3. One TRMM swath intersecting path of Hurricane Irene in 2005

server[13]. In Fig. 2, the two outer curves are the boundaries of the satellite observations and the middle curve represents the median of the observation boundaries. The median approximates the satellite trajectory when sensor takes measurements above the ocean. However, it is impossible to estimate the QuikSCAT satellite trajectory accurately from the Level 2B data when the satellite is above or near land due to the satellite sensor measurement constraints above or near land.

TRMM. The Tropical Rainfall Measurement Mission (TRMM) is a joint mission between NASA and the Japan Aerospace Exploration Agency (JAXA) designed to monitor and study tropical rainfall. TRMM satellite orbits between 35 degrees north and 35 degrees south of the equator. It takes measurements between 50 degrees north and 50 degrees south of the equator. All TRMM products are archived and distributed to the public by the Goddard Distributed Active Archive Center (GES DISC DAAC).[14]

For TRMM, we use the Level 2A12 data product, "TMI Profiling" which contains vertical hydrometeor profiles on a pixel by pixel basis. For each pixel, cloud liquid water, precipitation water, cloud ice water, precipitation ice, and latent heating are given at 14 vertical layers [18]. The TRMM Level 2A12 data is obtained from the Goddard Earth Sciences and Information Services Center[15].

There are 15 fields in the SDS (Science Data Set) in the TRMM Level 2A12 HDF data file. We use the scan time and geo-location shown in Table 3 to estimate the satellite motion. These fields are summarized in Table 4 and 5. Fig. 3 shows a TRMM swath intersecting path of Hurricane Irene in 2005. One notes that TRMM satellite takes measurements over land unlike the QuikSCAT

[13] `ftp://podaac.jpl.nasa.gov/ocean_wind/quikscat/L2B12/data`
[14] `http://disc.sci.gsfc.nasa.gov/`
[15] `ftp://disc2.nascom.nasa.gov/ftp/data/s4pa/TRMM_L2/TRMM_2A12`

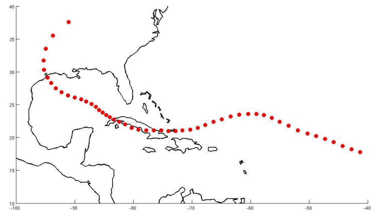

Fig. 4. Cyclone track for Hurricane Ike 2008 from NHC best track data.

satellite. Hence, the median of the observation boundaries approximates the TRMM satellite trajectory well.

4.2 Tropical Cyclone Event Trajectory

A trajectory is the path a moving object follows through space and time. Consider a time-stamped d dimensional data sequence defining a trajectory Tr as follows.

$$Tr = \langle (t_1, \boldsymbol{x}_1), \ldots, (t_i, \boldsymbol{x}_i), \ldots, (t_N, \boldsymbol{x}_N) \rangle$$

where N is the length of the data sequence Tr, $t_1 < \cdots < t_i < \cdots < t_N$ are the timestamps, and the vector $\boldsymbol{x}_i$ containing spatial information can have cardinality $d = 1, 2$, or 3; A tropical cyclone trajectory is described by (i) spatial attributes (latitude and longitude), and (ii) temporal attributes (year, day, time). Fig. 4 shows the trajectory of Hurricane Ike 2008 based on National Hurricane Center (NHC) best track information.

Historical tropical cyclone trajectories are obtained from the NOAA Coastal Services Center (Atlantic and North Eastern Pacific)[16]. The eleven tropical cyclone trajectories in North Atlantic Ocean in 2009 are used in our experiments.

5 Satellite Sensor Object Partitioning Scheme

Since (i) the temporal resolution of the satellite observations is relatively high, and (ii) the satellite orbiting speed is also relatively high compared to the atmospheric event objects, a large amount of data is generated in a relatively short time. Hence, one partition tree structure is used for each satellite object. The partitioning scheme is based on the time segmentation of the satellite trajectory, defined by the boundaries of the satellite sensor measurements.

We use the QuikSCAT satellite swath data (see Table 2) as an example to illustrate the partitioning scheme for satellite sensor objects. One notes that satellite measurements and their positions in an arbitrary row i in a data matrix

[16] `http://csc-s-maps-q.csc.noaa.gov/hurricanes/`

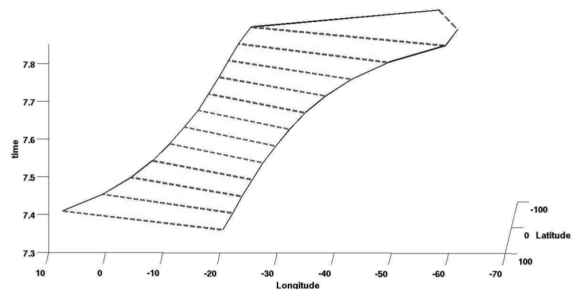

Fig. 5. A segment of the QuikSCAT satellite data swath divided into partitions.

(e.g., selected speed[i, 1:ncol]) have a fix timestamp in QuikSCAT, TRMM, and other satellites. Hence, the satellite sensor object at a fix time instance t (e.g., wvc_row_time[i]) can be represented by a spatial line segment or curve defined by the latitude and longitude values in row i in the position matrices (e.g., wvc_lat[i,1:ncol] and wvc_lon[i,1:ncol]).

The QuikSCAT satellite data swath is divided into partitions such that each partition is a spatial region within a time interval defined by a fixed n number of consecutive wvc_row_time elements (see Fig. 5). These partitions form the leave nodes in the partition tree. Each partition time interval varies slightly due to the non-uniform measurement sampling. Each leave node (partition) contains (i) the temporal information consisting of the start of the time interval wvc_row_time[i] = t_s, and the end of the time interval wvc_row_time[$i+n-1$] = t_e, and (ii) the spatial information for a swath data partition defined by the first and last non-zero elements in wvc_lat[i,1:ncol] and wvc_lon[i,1:ncol] at t_s, and wvc_lat[$i+n-1$,1:ncol] and wvc_lon[$i+n-1$,1:ncol] at t_e. In other words, a leave node partition is a quadrilateral region defined by the four corners of the data swath partition approximating the data swath partition. One notes that as n increases, some measurements in a data swath partition nearer to one of the swath data boundaries fall outside the leave node partition. If n is too high, one may fail to identify the data swath partitions that intersect a tropical cyclone trajectory.

The partition tree structure for a satellite sensor object is shown in Fig. 6. Revolution numbers (Rev. No) are unique incremental numbers tagging the orbits. *p_id*s are unique numbers tagging the partitions shown in Fig. 5. For the QuikSCAT satellite object, there are either 365 or 366 Julian days each year, 14 unique revolution numbers per day, and each swath defined by a revolution number is divided into segments containing n consecutive time instances. For the TRMM satellite object, the only difference is that there is either 15 or 16 unique revolution numbers per day.

Algorithm 1 is used to search the partition tree structure (for QuikSCAT swath data) for partitions that intersect a path defined by two consecutive trajectory points and the user-defined radius R in degree. In Lines 2 to 3, spatio-

```
Root Node:                          [Year, Y]
                                        |
Non-Leave Nodes:            [Year, Y; Julian Day, D]
                                        |
Non-Leave Nodes:    [Year, Y; Julian Day, D; Rev. Number, Rn]
                                        |
Non-Leave Nodes:       [Rev. No., Rn; time: s_time, e_time]
                                        |
Leave Nodes: [p_id; Rev. No., Rn; Partition, P; time: Ps_time, Pe_time]
```

Fig. 6. Partition tree scheme for the moving satellite trajectory.

temporal points between the two consecutive trajectory points and their corresponding circumference points are computed. For each interpolated spatio-temporal point, the partition tree structure is searched to locate the Revolution number Rn which the spatio-temporal point may be in (Line 5). When a Rn is located, the partitions which may contain the interpolated point will be searched (Lines 8 to 14). If the interpolated point and its circumference points are found in a partition, I and TS are updated (Lines 10 to 13). I contains information related to the start time instances and the end time instances of the spatio-temporal partitions that the interpolated points and their circumference points intersect. The goal of Line 20 is to locate the earliest start time and the latest end time from I and also the start row number RI_s and the end row number RI_e in the swath Rn. TS is the set defined in (1). Algorithm 1 can be generalized to other satellite sensor data.

6 Retrieval Algorithm

Next, we describe the algorithm that retrieves all satellite measurements within a specified radius R from TS defined in (1). In practice, we want a unique set of retrieved satellite sensor measurements, $\mathcal{M} = \{M_1, \ldots, M_s\}$ from the satellite sensor data set S such that

$$M_i \bigcap M_j = \emptyset, i \neq j, \forall i, j \in \{1, \ldots, s\} \tag{2}$$

with each M_i defined by $M_i = \{m | m \in S, |m - tp_i| < R\}$ and represented by a unique $tp_i \in TS$ and a user-defined radius R. However, one is likely to match more than one (interpolated) trajectory point $tp_i \in TS$ to a specific satellite measurement partition. This may result in $M_i \bigcap M_j \neq \emptyset$ with M_i corresponding to tp_i and M_j corresponding to tp_j, $tp_i \neq tp_j$, and $i \neq j$. One needs to identify the best time interpolated trajectory position $\hat{\boldsymbol{x}}$ that corresponds to the satellite measurement set $M_{\hat{\boldsymbol{x}}}$ such that (2) is satisfied. We compute the best time interpolated trajectory position $\hat{\boldsymbol{x}}$ as follows.

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in X}{\operatorname{argmin}} \{\bar{\boldsymbol{s}} - \boldsymbol{x}\} \text{ for } \bar{\boldsymbol{s}} \in T \tag{3}$$

Input: Two consecutive trajectory points, $(t_s, \boldsymbol{x}_s)$ and $(t_e, \boldsymbol{x}_e)$; Radius, R (in degree)
Output: RI_s, RI_e, Rn, TS
$TS := \{\}; I := \{\}$;
Generate a set of interpolated points, $P = \{p_1, \ldots, p_k\}$, using $(t_s, \boldsymbol{x}_s)$ and $(t_e, \boldsymbol{x}_e)$;
Generate a set of circumference points, C_i for each $p_i \in P$ based on R;
for interpolated point $p_i = (t_i, \boldsymbol{x}_i)$ **do**
  Identity Rn and leave-node partitions within time interval T =[s_time, e_time] in the partition tree based on t_i;
  if $Rn \neq \emptyset$ **then**
    $C_i := C_i \bigcup \{p_i\}$;
    for partition, Q_j in T of Rn **do**
      $I_i := \{Q_j | Q_j \bigcap C_i \neq \emptyset\}$;
      if $I_i \neq \emptyset$ **then**
        $I := I \bigcup I_i$;
        $TS := TS \bigcup \{p_i\}$;
      end if
    end for
  end if
end for
if $I = \emptyset$ **then**
  $RI_s := RI_e := Rn := \emptyset$;
end if
Use I to identify the start row number, RI_s and end row number, RI_e for wvc_lat, wvc_lon, and wvc_row_time in Rn.

Algorithm 1: Partition tree search to locate the tropical cyclone event in satellite swath data.

Input: Rn, RI_s, RI_e, R (in degree), TS.
Output: Point Sets: PS_{ws}, PS_{wd}
Retrieve QuikSCAT HDF Data with Rev. No., Rn;
$p_y := \{\text{wvc\_lat}[i, 1 : ncol],\ i \in [RI_s, RI_e]\}$;
$p_x := \{\text{wvc\_lon}[i, 1 : ncol],\ i \in [RI_s, RI_e]\}$;
$ws := \{\text{selected\_speed}[i, 1 : ncol],\ i \in [RI_s, RI_e]\}$;
$wd := \{\text{selected\_direction}[i, 1 : ncol],\ i \in [RI_s, RI_e]\}$;
Compute $\hat{\boldsymbol{x}}$ using (3) OR Cyclone Eye Locator (see Algorithm 3);
for $p(j) := (p_y(j), p_x(j))$ **do**
  $dist(j) := \| p(j) - \hat{\boldsymbol{x}} \|_2$;
end for
$PS_{ws} := \{ws(k) \mid dist(k) < R\}$;
$PS_{wd} := \{wd(k) \mid dist(k) < R\}$;

Algorithm 2: Retrieval algorithm for wind direction and speed measurements.

where

$$\begin{aligned} T &= \left\{ \boldsymbol{s} \mid \boldsymbol{s} = \frac{\boldsymbol{x}_e - \boldsymbol{x}_s}{t_e - t_s}(t - t_s) + \boldsymbol{x}_s \text{ for } t \in [t_s, t_e] \right\}, \\ X &= \{\boldsymbol{x} = (x_1, x_2) \mid x_1 = \text{wvc\_lat}[i, j], x_2 = \text{wvc\_lon}[i, j], \\ &\qquad \forall (i, j), j = 1, \cdots, ncol \text{ and wvc\_row\_time}(i) \in [t_s, t_e]\}, \end{aligned} \tag{4}$$

Input: QuikSCAT L2B Data with rev. no, Rn; $(t_s, \boldsymbol{x}_s)$ and $(t_e, \boldsymbol{x}_e)$.
Output: S, Cyclone Eye
1: Subset the L2B data based on $(t_s, \boldsymbol{x}_s)$ and $(t_e, \boldsymbol{x}_e)$;
2: Grid the L2B subseted data;
3: **for** Pixel i from the gridded L2B subseted data **do**
4: Compute the normal vector $\hat{n}_i$ to the direction vector $\hat{d}_i$;
5: Calculate which 8-neighbors $\hat{n}_i$ is pointing;
6: Update the neighbor count N_k of the pixel k $\hat{n}_i$ is pointing;
7: Update l_k, list of neighbor pixels, pointing at k ;
8: **end for**
9: $MaxNeighbor := \max_{1 \leq k \leq m} N_k$;
10: $VC := \{i \mid N_i \geq MaxNeighbor - 1\}$;
11: **for** $j \in VC$ **do**
12: $root := j$;
13: $Count[j] := SizeOfSpanningTree(root, l_{root})$;
14: **end for**
15: $S := \arg\max_{j \in VC} Count[j]$;

Algorithm 3: Cyclone eye locator.

such that $(t_s, \boldsymbol{x}_s)$ and $(t_e, \boldsymbol{x}_e)$ are two consecutive (interpolated) trajectory points and $M_s \bigcap M_e \neq \emptyset$.

Algorithm 2 retrieves wind direction and speed measurement sets from the QuikSCAT HDF data files based on user-defined radius and outputs from Algorithm 1. Assuming that Rn is a single revolution number, a single HDF data file is retrieved (Line 1). To improve the accuracy of a tropical cyclone eye position $\hat{\boldsymbol{x}}$ and data retrieval, a cyclone eye locator algorithm (see Algorithm 3 [19]) based on the vortex feature of a tropical cyclone can be used instead of computing $\hat{\boldsymbol{x}}$ using (3) (Line 6). Then, the distances between all the spatial points in the partition located using outputs from Algorithm 1 and $\hat{\boldsymbol{x}}$ are computed (Line 7-9). The point sets containing wind speed and direction measurements within the user-defined radius, R, are created (Lines 10 to 11). Algorithm 2 can be generalized to data retrieval for any satellite HDF file.

A simple query and retrieval system for QuikSCAT L2B swath data is shown in Fig. 7. The user inputs consist of arbitrary trajectory information and the retrieval parameter R. First, Algorithm 1 searches the partition tree structure. The retrieval parameter R and the outputs from Algorithm 1 are then used by Algorithm 2 to retrieve the satellite data for analysis.

One notes that using Algorithm 3 increases the computation cost of Algorithm 2. Algorithm 3 works as follows. First, the satellite data is gridded (Line 2). Then, one computes the normal vector to the wind direction at each gridded pixel and compute the number of pixels pointing to each gridded pixels (Lines 3 to 8). The most likely cyclone eye position is the one which creates the largest spanning tree from among the pixels (Lines 11 to 14) with the largest number of pixels pointing to them (Line 15). For TRMM measurements, one can use the characteristics discussed in [6] to improve the cyclone eye positions.

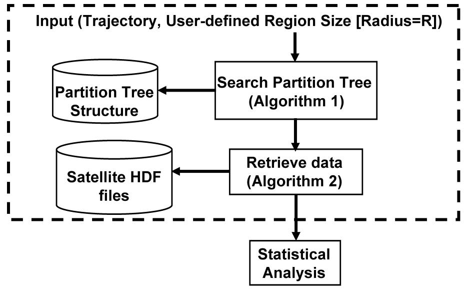

Fig. 7. QuikSCAT data search and retrieval system.

7 Experimental Results

In our experiments, we used QuikSCAT L2B data and TRMM 2A12 data from Day 182 (July 1) to 325 (November 21) in 2009. There are 2623 uncompressed QuikSCAT L2B HDF data files (32.1M each and a total size of 84.2G) and 2245 uncompressed TRMM 2A12 HDF data files (98.4M each and a total size of 220.9G). All eleven tropical cyclones occurring in North Atlantic Ocean in 2009 are used in the experiments.

First, we look at the effect of partition size to the partition tree search time, data retrieval time, and the number of vectors returned to users. The number of consecutive instances, n, in the time interval for each partition is varied from 1 to 200 with $R = 1$. When $n = 1$, it represents the standard approach where each row in a measurement position matrix has to scan through to decide whether there is an intersection between a tropical cyclone trajectory and a satellite object. As n increases, the partition size increases and the number of partitions decreases. One observes from Fig. 8 that as n increases, the mean search time (MST) drops exponentially and stabilizes after $n = 50$ for both the QuikSCAT and TRMM satellite data. The number of vectors returned to the user query decreases as n increases. We pointed out earlier in Section 5 that a bigger n value decreases the accurate approximation of the data swath partition which in turns affects the data retrieval accuracy by failing to return the data vectors based on the tropical cyclone trajectory. While the number of HDF files (NRtr) that need to be opened in Algorithm 2 remains almost the same, the mean retrieval time (MRT) increases due to the decrease in the number of returned vectors (NRtn). The mean retrieval time for TRMM data is much higher than QuikSCAT data as much more data is retrieved for each HDF data file.

Next, we query for QuikSCAT and TRMM measurements for all the North Atlantic Ocean tropical cyclones in 2009 with user-defined radius $R = 1, 2,$ and 3. The data query and retrieval performance statistics for each tropical cyclone for QuikSCAT and TRMM data are presented in Table 6 and 7, respectively. Based on results shown in Fig 8, we set n to 25 for QuikSCAT and 50 for TRMM satellite data so that it has identical returned vectors as standard approach

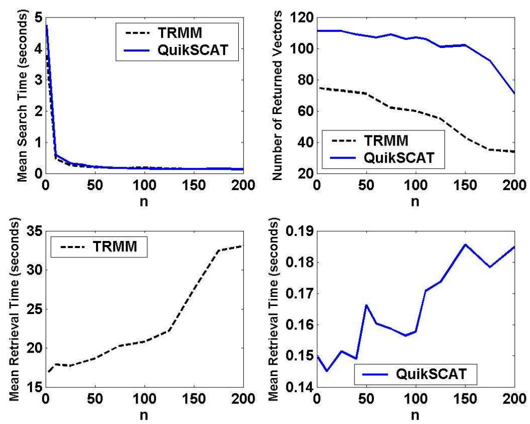

Fig. 8. Effect of partition size, n, on the mean search time (MST), number of vectors returned (NRtn) and mean retrieval time (MRT) for QuikSCAT and TRMM satellite data.

($n = 1$). Some observations from the experimental results in Table 6 and 7 are as follows.

1. The mean search time (MST) for each tropical cyclone for a particular satellite is similar. The MST for TRMM satellite is lower than QuikSCAT satellite as there are more leave nodes (partitions) for each QuikSCAT non-leave node (revolution number).
2. User-defined radius does not affect the MST as the number of sampling points is fixed. It only affects the number of HDF files opened and measurement vectors returned to user, which in turn affects the total retrieval time (TRT).
3. The total retrieval time (TRT) is related to (i) the number of HDF opened (Nrtr), (ii) the number of data features retrieved from the HDF files and their matrix size, and (iii) the number of measurement vectors to be returned to user. For each retrieved QuikSCAT data file, four 2-D matrices, namely speed, direction, latitude, and longitude, are retrieved . For each retrieved TRMM data file, six 3-D (fourteen data points for each spatial location) matrices for five features, and the spatial location (two data points for spatial location) are retrieved. Hence, TRMM data retrieval takes longer time.
4. Even though Hurricane Grace consists of thirty-seven segments, only six to seven TRMM measurement vectors are returned compared to the twenty QuikSCAT measurement vectors since Hurricane Grace occurred close to Europe which is near and beyond the edge of TRMM orbiting latitude.
5. 01L occurred near and beyond the edge of TRMM orbiting latitude. QuikSCAT also did not registered any measurement when three degrees from its eye. Hence, no related data is retrieved or returned.

		Radius = 1					Radius = 2					Radius = 3				
Name	NS	TST	MST	NRtr	NRtn	TRT	TST	MST	NRtr	NRtn	TRT	TST	MST	NRtr	NRtn	TRT
01L	13	3.27	0.27	0	0	0.00	3.19	0.27	0	0	0.00	3.22	0.27	0	0	0.00
Ana	31	9.39	0.31	11	11	1.01	9.38	0.31	12	12	1.11	9.44	0.31	14	14	1.43
Bill	49	14.82	0.31	21	21	1.90	14.79	0.31	21	21	1.91	14.91	0.31	23	23	2.19
Claudette	5	1.22	0.30	2	2	0.30	1.22	0.31	2	2	0.18	1.22	0.31	2	2	0.18
Danny	22	6.60	0.31	8	8	0.72	6.58	0.31	8	8	0.72	6.63	0.32	9	9	1.28
Erika	32	9.67	0.31	11	11	0.97	9.70	0.31	12	11	1.10	9.73	0.31	14	13	1.35
Fred	34	10.43	0.32	14	14	1.22	10.49	0.31	15	15	1.37	10.48	0.32	16	16	1.64
08L	8	2.31	0.33	4	4	0.35	2.33	0.33	4	4	0.37	2.34	0.33	4	4	0.37
Grace	37	11.57	0.32	20	20	1.84	11.67	0.32	20	20	1.94	11.69	0.33	20	20	1.91
Henri	21	6.71	0.34	9	8	1.67	6.75	0.34	10	10	0.90	6.76	0.34	12	11	1.35
Ida	29	9.29	0.33	13	12	1.74	9.32	0.33	15	15	1.43	9.28	0.33	15	15	1.53

Table 6. Query and Retrieval Performance: QuikSCAT data query and retrieval for North Atlantic tropical cyclones in 2009.

		Radius = 1					Radius = 2					Radius = 3				
Name	NS	TST	MST	NRtr	NRtn	TRT	TST	MST	NRtr	NRtn	TRT	TST	MST	NRtr	NRtn	TRT
01L	13	1.76	0.15	0	0	0.00	1.75	0.15	0	0	0.00	1.74	0.14	0	0	0.00
Ana	31	5.86	0.20	24	13	230.20	5.87	0.20	24	15	225.32	5.89	0.20	25	18	236.71
Bill	49	8.93	0.19	26	14	214.87	8.87	0.18	27	20	212.05	8.87	0.18	27	21	211.55
Claudette	5	0.75	0.19	2	2	13.32	0.74	0.18	2	2	13.23	0.74	0.19	2	2	14.41
Danny	22	3.99	0.19	9	6	81.61	3.97	0.20	9	8	81.77	3.94	0.19	9	8	81.15
Erika	32	5.98	0.19	22	10	186.13	6.00	0.20	22	13	176.35	5.99	0.19	22	17	180.55
Fred	34	6.40	0.19	22	6	197.97	6.35	0.20	22	9	195.58	6.43	0.19	22	11	200.66
08L	8	1.43	0.20	5	2	63.88	1.46	0.21	5	3	62.47	1.48	0.21	5	3	63.82
Grace	37	6.59	0.18	9	6	58.47	6.71	0.19	11	6	69.76	6.71	0.19	14	7	93.15
Henri	21	4.08	0.20	11	7	148.06	4.20	0.21	11	8	156.24	4.13	0.21	11	8	157.68
Ida	29	5.52	0.20	14	5	123.21	5.59	0.20	14	9	126.56	5.50	0.20	14	11	126.71

Table 7. Query and Retrieval Performance: TRMM data query and retrieval for North Atlantic tropical cyclones in 2009.

Abbreviation	Definition
NS	Number of line segments = Number of trajectory points - 1
TST	Total time for partition tree searching (seconds)
MST	Mean search time for each segment= $\frac{\text{TST}}{\text{NS}}$ (seconds)
NRtr	Number of HDF files opened.
NRtn	Number of sensor measurement vectors returned to user
TRT	Total time for data retrieval (seconds)

Table 8. Abbreviation Definitions

Fig. 9 shows examples of retrieved QuikSCAT partitions and their output vectors for Hurricane Bill in 2009 when the user-defined radius R is 3. The top row shows the retrieved QuikSCAT data partitions for Hurricane Bill at three time instances. The bottom row shows the output vectors at the three time instances. One notes that the satellite sensor captured the hurricane partially at the later two time instances. In the right column, the example shows a case when the interpolated hurricane eye location is beyond the sensor data boundaries.

Fig. 10 shows a fully developed hurricane. The middle image shows the output vector based on the interpolated hurricane eye location at 15.33N, 310.83E which is slightly East of the hurricane eye. The right image shows the output vector using the eye position computed from Algorithm 3. The hurricane eye is at 15.20N, 310.40E. The right image appears to be the more accurate output vector. Again, one notes that the retrieval time (TST) increases with the application of a cyclone eye location algorithm such as Algorithm 3.

8 Conclusions and Future Work

In this paper, we describe an efficient framework to handle ad-hoc query and retrieval of satellite sensor data for dynamic atmospheric events such as tropical cyclones based on ad-hoc user-defined criteria. This approach provides Earth

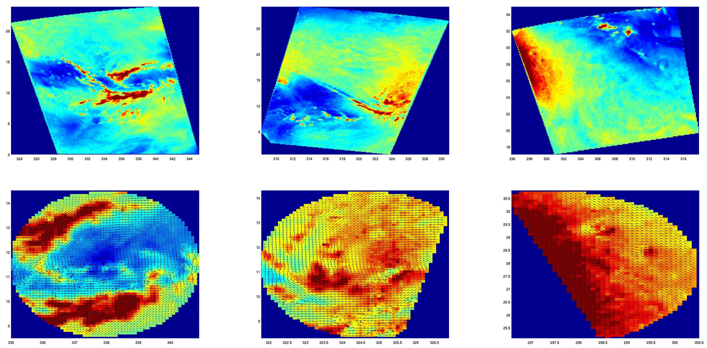

Fig. 9. QuikSCAT examples of retrieved partition and output vector when $R = 3$ for Hurricane Bill in 2009.

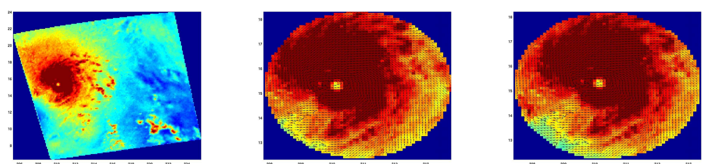

Fig. 10. A fully developed Hurricane Bill from QuikSCAT measurements. Left: Retrieved Partition; Middle: Output vector with $R = 3$ using interpolated eye location; Right: Output vector with $R = 3$ using Algorithm 3.

science researchers the capability to retrieve and manipulate satellite data to study dynamic atmospheric events. Future work include (i) integrating the current framework into a moving objects database for both satellite sensor objects and dynamic atmospheric (also earth and ocean) event objects, and (ii) the design and implementation of a spatio-temporal query language that enables users to pose ad-hoc satellite data retrieval queries (see query examples in Section 1). One also foresees the possibility of integrating our query and retrieval framework into a scientific workflow system to support flexible scientific analysis.

References

1. Esfandiari, M., Ramapriyan, H., Behnke, J, and Sofinowski, E.: Earth observing system (EOS) data and information system (EOSDIS) - evolution update and future. *IEEE Inter. Geoscience and Remote Sensing Symposium*, pp. 4005–4008 (2007)
2. Behnkre, J., Watts, T. .H., Kobler, B., Lowe, D., Fox, S., and Meyer, R.: EOSDIS petabyte archives: tenth anniversary. *Proc. 22nd IEEE/13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST' 05)*, pp. 81–93 (2005)

3. Yueh, S. H., Stiles, B. W., and Liu, W. T.: QuikSCAT Wind Retrievals for Tropical Cyclones, IEEE Transactions on Geoscience and Remote Sensing. Vol. 41, No. 11, pp. 2616-2628 (2003)
4. Mesrobian, E., Muntz, R., Shek, E. C., Nittel, S., Rouche, M., Kriguer, M., and Fabbrocino, F.: OASIS: An EOSDIS science computing facility. Proc. SPIE, vol. 2820, pp. 284–298 (1996)
5. Henning, M.: The Rise and Fall of CORBA. ACM Queue, vol. 4, no. 5, pp. 28–34 (2006)
6. Kodama, Y. M. and Yamada, T.: Detectability and Configuration of Tropical Cyclone Eyes over the Western North Pacific in TRMM PR and IR Observations. Monthly Atmospheric Review, vol. 133, pp. 2213–2226 (2005)
7. Yokoyama, C. and Takayabu, Y. N.: A Statistical Study on Rain Characteristics of Tropical Cyclones using TRMM Satellite Data. Monthly Atmospheric Review, vol. 136, pp. 3848–3862 (2008)
8. Lee C. S., Cheung, K. W., Hui, S. N., and Elsberry, R. L.: Mesoscale Features Associated with Tropical Cyclone Formations in the Western North Pacific. Monthly Atmospheric Review, vol. 136, pp. 2006–2022 (2008)
9. Rodgers, E. B. and Pierce, H. F.: A Satellite Observational Study of Precipitation Characteristics in Western North Pacific Tropical Cyclones. Journal of Applied Meteorology, vol. 34, pp. 2587–2599 (1995)
10. McTaggart-Cowan, R., Deane, G. D., Bosart, L. F., Davis, C. A., and Galarneau T. J. Jr.: Climatology of Tropical Cyclogenesis in the North Atlantic (1948-2004). Monthly Weather Review, vol. 136, pp. 1284–1304 (2008)
11. Shek E. C., Muntz, R., Mesrobian, E.: Extensible Parallel Query Processing for Exploratory Geoscientific Data Mining. Data Min. Knowl. Discov., vol. 5, no. 4, pp. 277-304 (2001)
12. Ho, S.-S. and Talukder, A.: Automated Cyclone Discovery and Tracking using Knowledge Sharing in Multiple Heterogeneous Satellite Data. Proc. 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 928–936 (2008)
13. Guting, R. H. and Schneider, M.: Moving Objects Databases. Morgan Kaufmann (2005)
14. Mokbel, M. F., Ghanem, T. M., and Aref, W. G.: Spatio-Temporal Access Methods. *Bullentin of the IEEE Computer Society Technical Committee on Data Engineering*, Vol. 26, No. 2, pp. 40–49 (2003)
15. Pfoser, D., Jensen, C. S., and Theodoridis, Y.: Novel Approaches to the Indexing of Moving Object Trajectories. Proc. 26th Int. Conf. on Very Large Databases, pp. 395–403 (2000)
16. Song, Z. and Roussopoulos, N.: SEB-tree: An Approach to Index Continuously Moving Objects. Proc. 4th Int. Conf. on Mobile Data Management, pp. 340–344 (2003)
17. Lungu, T. and et. al.: QuikSCAT Science Data Product User's Manual, Version 3.0, D-18053-Rev A (2006)
18. Tropical Rainfall Measuring Mission Science Data and Information System, Vol. 4: File Specifications for TRMM Products - Level 2 and Level 3, Release 6.09 (2007)
19. Ho, S.-S. and Talukder, A.: Utilizing Spatio-Temporal Text Information for Cyclone Eye Annotation in Satellite Data. *Proc. IJCAI Workshop on Cross-media Information Access and Mining*, pp. 25–32 (2009)