# Moving Objects Database Technology for Ad-Hoc Querying and Satellite Data Retrieval of Dynamic Atmospheric Events

Markus Schneider[1], Shen-Shyang Ho[2], Tao Chen[1], Arif Khan[1],
Ganesh Viswanathan[1], Wenqing Tang[2], W. Timothy Liu[2]

[1]Department of Computer &
Information Science & Engineering
University of Florida
Gainesville, FL 32611

[2]Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

***Abstract*—Existing state-of-the-art and web-based weather event information portals, data archives, and forecast services provide excellent subsetting and visualizations of weather events and satellite sensor measurements. However, users only obtain limited, simple, and hard-coded query, retrieval, and analysis capabilities from these sources. One non-existent but desirable capability is the accurate and efficient ad-hoc querying of the trajectories of dynamic atmospheric events (e.g., tropical cyclones, hurricanes) as well as the efficient retrieval of Earth science satellite sensor data for these events based on ad-hoc, user defined criteria and the event trajectories.**

**In this paper, we describe our current work and progress in the development of a sophisticated framework based on the moving objects database technology for ad-hoc querying and retrieval of atmospheric events and their associated satellite measurements. Such a capability is extremely important to scientists who process sensor data of atmospheric events for statistical analysis and scientific investigation.**

## I. INTRODUCTION

The main objective of our project is to provide the NASA workforce with previously unavailable data(base) management, analysis, and query capabilities that will advance the research and understanding of dynamic weather events and be based on weather data derived from the NASA mission sensor measurements. Our focus is especially on tropical cyclone weather events. The proposed core contribution for achieving this objective is based on the concept of *moving objects* describing the evolution of spatial objects (geometries) over time (like planes, hurricanes, cars, whales) and includes the design and implementation of a *moving objects database technology* providing four main key components:

1) the *Moving Objects Software Library* (*MOSL*), which is a software library that provides a representation of moving objects, enables the execution of operations on them, and can be integrated into extensible databases,
2) the *Spatio-Temporal Query Language* (*STQL*), which enables decision makers and application scientists to pose ad hoc database queries for spatio-temporal analysis tasks in a comfortable manner and to obtain immediate response,
3) a *Moving Objects Database* (*MOD*), which stores spatiotemporal and thematic weather data from different web sites in a uniform and consistent manner, and
4) a *Satellite Data Retrieval* (*SDR*) component, which takes trajectories provided by moving objects as input and enables scientists to retrieve and access pertaining satellite data for analysis purposes.

Our technology supports ad-hoc spatio-temporal queries such as those listed below:

- 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 100E and 180.
- Retrieve TRMM precipitation data for tropical cyclones from December 1997 to December 2003.
- Retrieve QuikSCAT 25km resolution wind vector and 50km resolution global wind vector for tropical cyclones in western North Pacific which formed west of 160E and south of 26N from September 1999 to December 2004.
- Retrieve QuikSCAT wind vectors and TRMM precipitation data for all tropical cyclones with translation speed faster than 7 meter/second [1].

The first three queries are derived from published journal papers [2], [3], [4] that require search, retrieval, and analysis of satellite data containing cyclone features. The fundamental problem is that the retrieval results have been obtained by troublesome, time-consuming, and manual search without software and database support.

The outline of this paper is as follows. Section II reviews related work. In Section III, we outline the application and system objectives of our project. In Section IV, we describe the main components of our system and their implementation. Finally, in Section V, we draw some conclusions and provide an outlook to our future work.

## II. RELATED WORK

We discuss the related literature from an application standpoint (earth science applications) in Section II-A and from a system standpoint (moving objects databases) in Section II-B.

### A. Data and Information Systems for Earth Science Applications

The Earth Observing System Data and Information System (EOSDIS)[1] is a comprehensive data and information system which archives, manages, and distributes Earth science data from the EOS spacecrafts (a.k.a. satellite sensors) [5]. A challenge of EOSDIS is to "help users find the data that they need and to get it to them" [6]. The Warehouse Inventory Search Tool (WIST)[2] 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. However, there is still a lack of capabilities that support flexible data query and retrieval of Earth science satellite sensor data for dynamic atmospheric events in the EOSDIS.

There are many existing state-of-the-art publicly available web-based tropical cyclone data and information portals[3] [4], data archives [5], and forecast services [6] provide excellent visualizations and information of tropical cyclones and satellite sensor measurements. Yet, 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:

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

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) [7]. 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 [8].

### B. Moving Objects Databases

The general idea of moving objects databases [9], [10], [11], [12], [13] is that we would like to be able to represent moving entities in databases and ask queries about them. Moving entities could be people, animals, vehicles such as cars, trucks, air planes, ships, etc. For these examples, usually only the time-dependent position in space is relevant, not the extent, hence we can characterize them as *moving points*. The trajectory of such point-entities over time can be described by *moving lines*. However, there are also moving entities with an extent, for example, hurricanes, forest fires, oil spills, armies, epidemic diseases, and so forth. These can be characterized as *moving regions*.

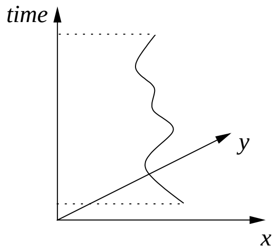


Fig. 1. Example of a moving point object

Specifically, the goal of our research on *moving objects databases* is to extend database technology so that any kind of moving entity can be represented in a database and all questions about such movements can be formulated in a precise yet simple way using the featured query languages. These enhanced systems are called *moving objects databases* [12] and are full-fledged DBMSs with additional support for the management of spatial and moving objects data. The fundamental improvement here is the inclusion of *spatiotemporal data types* into the DBMS type system as standard attribute types together with a collection of comprehensive operations and predicates. In [9], the semantics of *spatiotemporal data types* and *spatiotemporal operations* for *moving points* (type *mpoint*), *moving lines* (type *mline*), and *moving regions* (type *mregion*) are formally defined at the level of an abstract model. This serves as a formal specification for implementing types and operations. Figure 1 shows an example of a moving point. A formal model for *spatiotemporal predicates* as time-varying topological predicates is presented in [10]. A discrete model is proposed in [11], [13] which provides finite representations, as well as data structures, for all the types of the abstract model. The efficient implementation of algorithms for a rather large set of comprehensive operations defined in the abstract model is discussed in [14]. It offers a blueprint for implementing a moving objects extension package in extensible databases.

## III. OBJECTIVES

We discuss the objectives of our project from an application standpoint in Section III-A and from a system standpoint in Section III-B.

### A. Application Objectives

Currently, there is still a lack of capabilities that support ad-hoc satellite data query and retrieval in the EOSDIS. In particular, the accurate and efficient ad-hoc query and retrieval of Earth science satellite sensor data for dynamic atmospheric events such as tropical cyclones is currently not available.

The main application objectives of our technology are as follows.

---

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

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

[3] Navy/NRL Tropical Cyclone. http://www.nrlmry.navy.mil/tc_pages/tc_home.html

[4] NASA GSFC Hurricane Portal. http://daac.gsfc.nasa.gov/hurricane/

[5] Physical Oceanography DAAC Hurricane/Typhoon Tracker. http://podaac.jpl.nasa.gov/hurricanes/

[6] NOAA National Hurricane Center. http://www.nhc.noaa.gov/pastall.shtml

1) *To equip scientists with ad-hoc dynamic event query and data retrieval tools.* Such tools allow scientists to perform flexible query and then to retrieve satellite data (see examples in Section I) for statistical analysis.
2) *To support scientific analysis of retrieved satellite measurements.* Novel and standard 1-D and 2-D statistical composite analysis tools will be developed to assist scientists in justifying scientific hypotheses and to support knowledge discovery.

The application objectives listed above are relevant to NASA missions and support existing NASA Earth Observing missions such as Cloudsat, Calipso, AIRS, MODIS, CERES, QuikSCAT, and TRMM that measure dynamic processes. The objectives will also directly support proposed missions such as (i) Extended Ocean Vector Wind Mission (XOVWM) ([15], pp. 100-103), (ii) Precipitation and All-Weather Temperature and Humidity (PATH) ([15], pp. 124-126), and (iii) Three-dimensional Tropospheric Winds from Space-based Lidar (3D-Winds) ([15], pp. 137-139).

The focus of this paper is the first application objective. Our solution approach to this objective can be easily incorporated into current information systems as well as systems supporting future missions since the core moving objects database technology is mission-independent. This will reduce implementation cost related to future missions while providing an efficient and uniform data basis for weather information and a flexible query tool for retrieval and analysis. In Section IV, we describe our solution approach to achieve this objective.

### B. System Objectives

The system objectives describe the goals of the project from a system and software architecture standpoint and include three components, namely (i) a Moving Objects Database (MOD), (ii) the Moving Objects Software Library (MOSL), and (iii) the Spatiotemporal Query Language (STQL).

From an application perspective, the *moving objects database* (*MOD*) that we are going to create is supposed to keep tropical cyclone and hurricane data provided by public sources and web sites in a centralized repository and make them available for querying to decision makers and application scientists. From a system perspective, it is a full-fledged database with additional support for spatial and spatiotemporal data in its data model and query language. In order to be able to add new functionality, we have the important requirement that a database system (DBMS) must be *extensible*.

From an application perspective, the *Moving Objects Software Library* (*MOSL*) provides the functionality in terms of spatiotemporal data types, operations, and predicates that can be deployed by decision makers and scientists in ad hoc queries and in database application programs (written in C++, Java, Matlab, etc.) to retrieve and derive tropical cyclone data. For that purpose, from a system perspective, MOSL provides a system of *spatiotemporal data types* together with a large number of *spatiotemporal operations* (e.g., *Intersection*, *Union*, *Difference*) and *spatiotemporal predicates* (e.g., *Inside*, *Meet*, *Disjoint*, *Overlaps*; *Enters*, *Leaves*, *Crosses*, *Bypasses*). A detailed description of the moving objects database technology can be found in [12]. It leads to many further scientific publications on this topic. In particular, MOSL provides *historical* spatiotemporal data types named *hmpoint*, *hmline*, and *hmregion* which can be integrated as attribute data types into extensible databases in a database-independent and application-neutral manner.

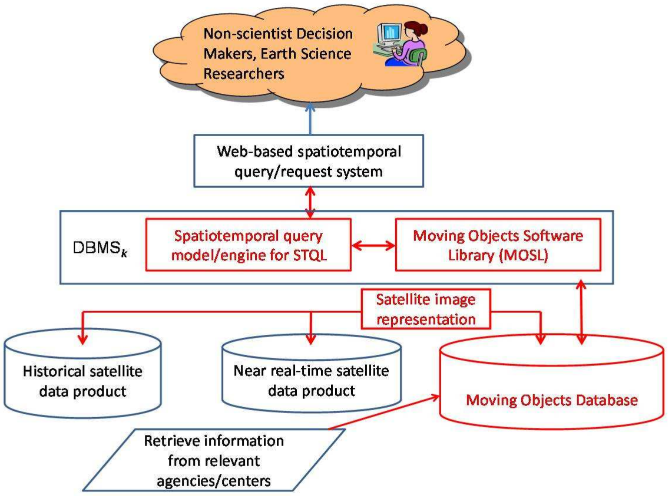


Fig. 2. The planned system architecture

From an application perspective, the *Spatiotemporal Query Language* (*STQL*) provides the communication interface between the moving objects database for tropical cyclone data and the decision maker and/or scientist. This textual language enables users to pose ad hoc spatiotemporal queries on moving objects in general and on tropical cyclone data in particular and to obtain immediate response. Further, it is supposed to retrieve satellite data based on user queries.

All three components are to be integrated into extensible database systems. This leads to the planned system architecture shown in Figure 2. The components in red indicate the new components and their embedding into available DBMS.

## IV. SYSTEM DESIGN AND IMPLEMENTATION

In this section, we describe our main technical findings. Section IV-A focuses on the design and implementation steps for the three system components MOSL, STQL, and MOD. Section IV-B deals with the progress made on the satellite data retrieval (SDR) component.

### A. System Components for Querying Tropical Cyclone Trajectories

For being able to create a database for moving objects, we have in a first step explored several available commercial and public domain database systems in order to find out whether they provide the needed and appropriate extensibility features and are thus appropriate for our purposes. The needed features comprise the availability of a data type for BLOBs (Binary Large Objects) for representing values/objects of arbitrary

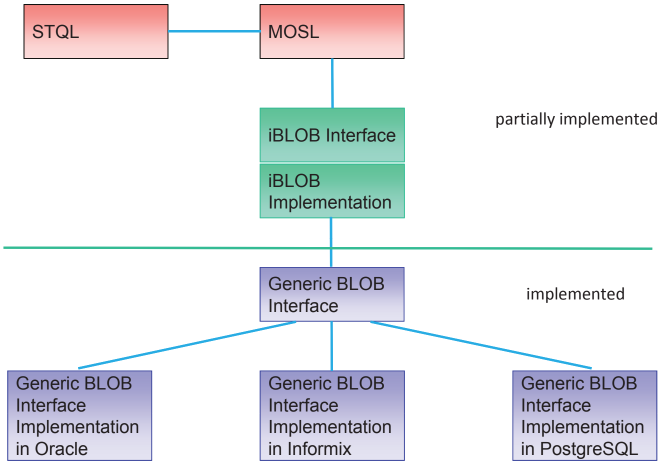
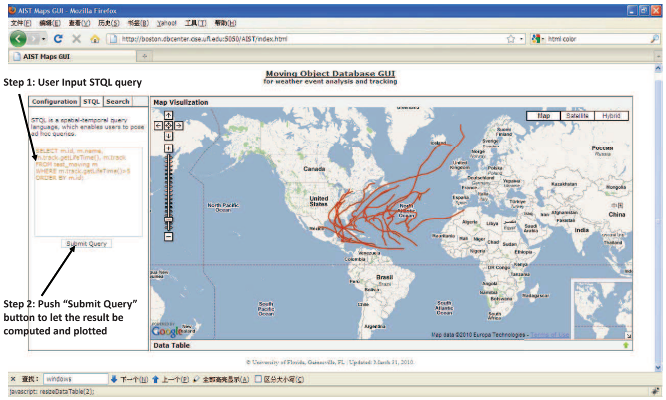


Fig. 3. Software architecture of MOSL, STQL, iBLOBs, and the generic BLOB interface



Moving Object Database GUI
for weather event analysis and tracking

Step 1: User Input STQL query

STQL is a spatial-temporal query language, which enables users to pose ad hoc queries.

```
SELECT m.id, m.name,
m.track.getLifeTime(), m.track
FROM test_moving m
WHERE m.track.getLifeTime()>5
ORDER BY m.id;
```

Step 2: Push “Submit Query” button to let the result be computed and plotted

Map data ©2010 Europa Technologies - Terms of Use

© University of Florida, Gainesville, FL | Updated: March 31, 2010.

```
javascript: resizeDataTable(2);
```

Fig. 4. STQL query tool and visualization tool

length and the availability of extensibility concepts like *user-defined data types* (*UDTs*) and *user-defined functions* (*UDFs*) for the specification and external binding of spatiotemporal data types and functions implemented in MOSL to the DBMS. The explored commercial database systems have been Oracle, DB2, Informix, and Sybase, and the explored public domain database systems have been PostgreSQL and MySQL. It turns out that only Oracle, Informix, and PostgreSQL fulfil our requirements. Hence, we have taken these three systems as a basis of our work in order to demonstrate later the database independence of MOSL and STQL.

The BLOB data type is a built-in data type in SQL, which is the standard query language in the database world, and represents arbitrarily long, finite byte strings in the magnitude of gigabytes and even terrabytes. It is used for representing *complex application objects* of large and/or varying length (e.g., spatial, image, DNA, video, multimedia objects). Since the three considered DBMS have different BLOB interfaces and implementations, we have created a *generic BLOB interface* that provides the least common denominator of all BLOB interfaces and defines the minimum BLOB functionality needed. An additional abstraction step makes the generic blob interface and thus all components built on top of it database system independent. The generic BLOB interface has three database specific implementations (Figure 3).

A fundamental problem of the BLOB data type is that it can only represent byte sequences and that it only offers byte operations like byte read and byte write. That is, the structure of complex application objects is not visible and not accessible but lost in a BLOB object. Hence, BLOBs have to be loaded completely into main memory in order to enable access to components of application objects again. This can become very expensive in join operations which are the central operations in databases to combine different data sets. Therefore, we have designed and implemented a new data type called *iBLOB* (*Intelligent Binary Large Object*) (Figure 3). It is implemented on top of the generic BLOB interface and enables *random read access* to the *conceptual components* of a complex application object without the need to understand their meaning. Further, it supports *random updates* of application objects. Updates of large and complex application objects are a rather neglected topic in the database literature. With this approach, it is not necessary to load a complex application object completely into main memory but only the components of interest. An iBLOB object consists of two main parts. Its *structure index* represents the hierarchical structure of the application object. Its *sequence index* maintains the logical sequential order of components stored in the structure index and supports updates.

Our design and implementation of the Moving Objects Software Library (MOSL) (Figure 3) is still at the beginning. At the moment, MOSL provides a first implementation of the spatiotemporal data type *hmpoint* for *historical moving point objects* together with a few spatiotemporal operations on them. For example, it is possible to determine the lifetime of a moving point object. We have managed to create the framework or skeleton of MOSL into which many new types, operations, and predicates will be integrated in the future.

We have begun to extend the standard database query language SQL by spatiotemporal operations and predicates to the spatiotemporal query language STQL (Figure 3) that allows the execution of spatiotemporal queries. Also here the development is still at the beginning and will be extended in the future.

For illustration and visualization purposes we have constructed a simple query tool for posing STQL queries as well as a web-based and Google map-based visualization tool for showing the results of STQL queries (trajectories) with Google maps as the background (Figure 4). The advantage of this approach is that it provides the full functionality of Google maps that many users are familiar with.

### *B. System Components for Retrieving Satellite Data for Queried Tropical Cyclones*

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

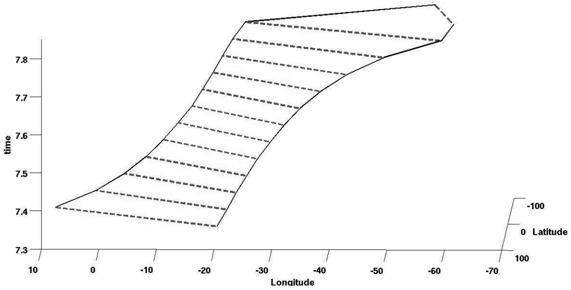


Fig. 5. A satellite data swath segment divided into partitions [16].

interest is “Find all *unique* satellite sensor measurements from $O_{si}$ that is at most $x$ kilometers from the tropical 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.

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. Furthermore, 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. Moreover, 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. The satellite data swath is divided into partitions such that each partition is a spatial region within a time interval defined by a fixed number of consecutive time instances (see Figure 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 and the end of the time interval, and (ii) the spatial information for a swath data partition. In short, a leave node partition is a quadrilateral region defined by the four corners of the data swath partition approximating the data swath partition.

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

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

with each $M_i$ defined by $M_i = \{m | m \in S, |m - tp_i| < R\}$ and represented by a unique center point and a user-defined radius $R$. However, one is likely to match more than one (interpolated) trajectory point to a specific satellite measurement partition. One needs to identify the best time interpolated trajectory position $\hat{\vec{x}}$ that corresponds to the satellite measurement set $M_{\hat{\vec{x}}}$ such that (1) is satisfied. We compute the best time interpolated trajectory position to represent the cyclone eye location based on the minimum distance between satellite measurement spatial locations and the cyclone trajectory at the closest time intersection between the satellite and the cyclone.

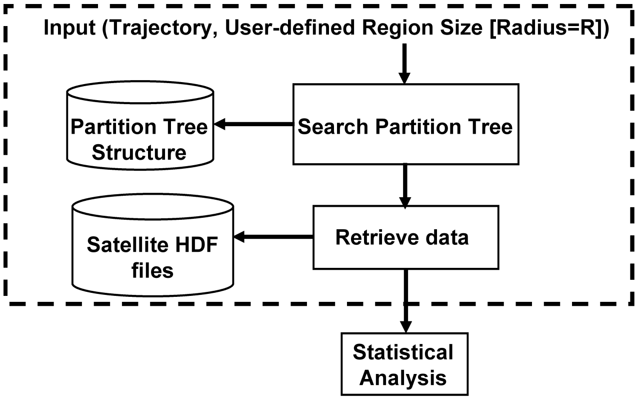


Fig. 6. Satellite Data Retrieval (SDR) Component [16].

The satellite data retrieval (SDR) component is shown in Figure 6. The user inputs consist of arbitrary trajectory information and the retrieval parameter $R$. First, the partition tree structure is searched based on the user inputs. Then, the retrieval parameter $R$ and the search results are used to retrieve the satellite data for analysis. Readers can refer to [16] for details on the SDR component.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we describe a preliminary software framework for the whole system architecture to be implemented based on the moving objects database technology for ad-hoc querying and retrieval of atmospheric events and their associated satellite measurements. Future work include additional functionalities for all system components and to continue the development of the software library MOSL as well as the design of the query language STQL. Further objectives are to migrate and integrate global trajectory data of the period from 2000 to 2009 into our moving objects database, the exploration of different system component integration alternatives, the integration of SDR into the database component, and to apply our integrated technology on real scientific end-user applications.

## VI. ACKNOWLEDGMENTS

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

# REFERENCES

[1] W. Tang and W. T. Liu, “Dependence of Hurricane Asymmetry and Intensification on Translation Speed Revealed by a Decade of QuikSCAT Measurements,” in *NASA Ocean Vector Wind Science Team Meeting, Boulder, Colorado*, 2009.

[2] Y. M. Kodama and T. Yamada, “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.

[3] C. S. Lee, K. W. Cheung, S. N. Hui, and R. L. Elsberry, “Mesoscale Features Associated with Tropical Cyclone Formations in the Western North Pacific,” *Monthly Atmospheric Review*, vol. 136, pp. 2006–2022, 2008.

[4] C. Yokoyama and Y. N. Takayabu, “A Statistical Study on Rain Characteristics of Tropical Cyclones using TRMM Satellite Data,” *Monthly Atmospheric Review*, vol. 136, pp. 3848–3862, 2008.

[5] M. Esfandiari, H. Ramapriyan, J. Behnke, and E. Sofinowski, “Earth Observing System (EOS) Data and Information System (EOSDIS) – Evolution Update and Future,” in *IEEE Int. Geoscience and Remote Sensing Symposium*, 2007, pp. 4005–4008.

[6] J. Behnkre, T. H. Watts, B. Kobler, D. Lowe, S. Fox, and R. Meyer, “EOSDIS Petabyte Archives: Tenth Anniversary,” in *22nd IEEE/13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST’ 05)*, 2005, pp. 81–93.

[7] E. Mesrobian, R. Muntz, E. C. Shek, S. Nittel, M. Rouche, M. Kriguer, and F. Fabbrocino, “OASIS: An EOSDIS Science Computing Facility,” in *Proc. of SPIE*, vol. 2820, 1996, pp. 284–298.

[8] M. Henning, “The Rise and Fall of CORBA,” *ACM Queue*, vol. 4, no. 5, pp. 28–34, 2006.

[9] M. Erwig, R. H. Güting, M. Schneider, and M. Vazirgiannis, “Spatio-temporal Data Types: an Approach To Modeling and Querying Moving Objects in Databases,” *GeoInformatica*, vol. 3, no. 3, pp. 269–296, 1999.

[10] M. Erwig and M. Schneider, “Spatio-Temporal Predicates,” vol. 14, no. 4, pp. 1–42, 2002.

[11] L. Forlizzi, R. Güting, E. Nardelli, and M. Schneider, “A Data Model and Data Structures for Moving Objects Databases,” in *ACM SIGMOD Int. Conf. on Management of Data*, 2000, pp. 319–330.

[12] R. Güting and M. Schneider, *Moving Objects Databases*. Morgan Kaufmann Publications, 2005.

[13] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis, “A Foundation for Representing and Querying Moving Objects,” *ACM Trans. on Database Systems (TODS)*, vol. 25, no. 1, pp. 1–42, 2000.

[14] J. A. C. Lema, L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider, “Algorithms for Moving Objects Databases,” *Computer Journal*, vol. 46, no. 6, pp. 680–712, 2003.

[15] Committee on Earth Science and Applications from Space: A Community Assessment and Strategy for the Future, National Research Council, *Earth Science and Applications from Space: National Imperatives for the Next Decade and Beyond*. The National Academies Press, 2007, freely downloadable from http://www.nap.edu/catalog/11820.html.

[16] S.-S. Ho, W. Tang, W. T. Liu, and M. Schneider, “A Framework for Moving Sensor Data Query and Retrieval of Dynamic Atmospheric Events,” in *22nd Int. Conf. on Scientific and Statistical Database Management (SSDBM)*, ser. Lecture Notes in Computer Science, vol. 6187. Springer, 2010, pp. 96–113.