5-2011

Event Discovery and Classification in Space-Time Series: A Case Study for Storms

Avinash Rude

EVENT DISCOVERY AND CLASSIFICATION IN SPACE-TIME SERIES: A CASE STUDY FOR STORMS

By

Avinash Rude

B. Tech, National Institute of Technology, Hamirpur, India

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

(in Spatial Information Science and Engineering)

The Graduate School

The University of Maine

May, 2011

Advisory Committee:

Kate Beard-Tisdale, Professor of Spatial Information Science and Engineering, Advisor

Silvia Nittel, Associate Professor of Spatial Information Science and Engineering

Neal R. Pettigrew, Professor of School of Marine Sciences

THESIS ACCEPTANCE STATEMENT

On behalf of the Graduate Committee for Avinash Rude, I affirm that this manuscript is the final and accepted thesis. Signatures of all committee members are on file with the Graduate School at the University of Maine, 42 Stodder Hall, Orono, Maine.

Dr. Kate Beard-Tisdale, Date
Professor of Spatial Information Science and Engineering

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my mentor and advisor Kate Beard-Tisdale for her kindness, support and patience throughout this work. She gave me opportunity and walked with me on this journey with reassuring faith in my abilities. Without her indispensible advice and expertise, this thesis would never have completed. I would also like to thank the advising committee members: Silvia Nittel for her encouragement of my development as a student and Neal R. Pettigrew for this critical domain expertise and insightful discussions of applications of this research in his field of research.

I would like to express my appreciation to my professors and friends in the Department of Spatial Information Science and Engineering for their help and their genuine interest in the success of my work. Special thanks to Matt Dube for helping with formal representation of the algorithms. I am grateful to my friends in the department especially Jake Emerson, Stacy Doore, Kripa Joshi, Monoj Raja and Susan Elston for their excellent company, feedback and ideas.

I like to acknowledge the support from the National Science Foundation (grant number 0429644), without with my graduate studies would not have been possible.

Finally, I would like to thank my mother for her kindness and oversight and my sister for her benevolent presence and support of my endeavors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIBRARY RIGHTS STATEMENT

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at The University of Maine, I agree that the Library shall make it freely available for inspection. I further agree that permission for “fair use” copying of this thesis for scholarly purposes may be granted by the Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature:

Date:

EVENT DISCOVERY AND CLASSIFICATION IN SPACE-TIME SERIES: A CASE STUDY FOR STORMS

By Avinash Rude

Thesis Advisor: Dr. Kate Beard-Tisdale

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Science
(in Spatial Information Science and Engineering)
May, 2011

Recent advancement in sensor technology has enabled the deployment of wireless sensors for surveillance and monitoring of phenomenon in diverse domains such as environment and health. Data generated by these sensors are typically high-dimensional and therefore difficult to analyze and comprehend. Additionally, high level phenomenon that humans commonly recognize, such as storms, fire, traffic jams are often complex and multivariate which individual univariate sensors are incapable of detecting. This thesis describes the Event Oriented approach, which addresses these challenges by providing a way to reduce dimensionality of space-time series and a way to integrate multivariate data over space and/or time for the purpose of detecting and exploring high level events.

The proposed Event Oriented approach is implemented using space-time series data from the Gulf of Maine Ocean Observation System (GOMOOS). GOMOOS is a long standing network of wireless sensors in the Gulf of Maine monitoring the high energy ocean environment. As a case study, high level storm events are detected and classified using the Event Oriented approach. A domain-independent ontology for detecting high level

composite events called a General Composite Event Ontology is presented and used as a basis of the Storm Event Ontology. Primitive events are detected from univariate sensors and assembled into Composite Storm Events using the Storm Event Ontology. To evaluate the effectiveness of the Event Oriented approach, the resulting candidate storm events are compared with an independent historic Storm Events Database from the National Climatic Data Center (NCDC) indicating that the Event Oriented approach detected about 92% of the storms recorded by the NCDC.

The Event Oriented approach facilitates classification of high level composite event. In the case study, candidate storms were classified based on their spatial progression and profile. Since ontological knowledge is used for constructing high level event ontology, detection of candidate high level events could help refine existing ontological knowledge about them.

In summary, this thesis demonstrates the Event Oriented approach to reduce dimensionality in complex space-time series sensor data and the facility to integrate ime series data over space for detecting high level phenomenon.

Chapter 1

INTRODUCTION

Time series are a common form of data sequences found in signal processing, econometrics, mathematical finance, and environmental and health monitoring. With the advent of numerous and widely deployed sensor monitoring systems and particularly wireless sensor networks (WSN), time series are becoming increasingly common and with the added characteristic that they are spatially distributed. Wireless Sensor Networks are generating large and unprecedented volumes of spatio-temporal data at fine temporal granularities. Space-time series refers to time series having spatial and temporal components. For example, each node in a sensor network typically generates a localized view of space in the form of time series with different locations. The space-time series data can provide benefits for scientific investigation of phenomena but also create new challenges. Generating large volumes of data presents the problem of converting data into meaningful and understandable information, which can effectively contribute to scientific investigation, problem solving, and decision making. Spatio-temporal data is typically high-dimensional and can be difficult to analyze and comprehend. New approaches are needed to cope with these growing volumes of data and convert them into understandable information. This thesis presents an Event Oriented (EO) approach that seeks to make large volumes of times series data more understandable by abstracting the time series data to events and making events the primary unit of analysis.

Standard time series analysis methods address the detection of patterns in time series where patterns consist of an identifiable set of systematic components and random noise or error. The systematic components can include trend, seasonal, and cyclical components and "classic" methods that decompose time series and *Census* methods have been around since the 1920s (Makridakis, Wheelwright, & McGee, Forecasting, 1983; Makridakis & Wheelwright, Forecasting methods for management, 1989). The EO approach differs from the traditional time series methods because the focus is more on the transient signals in the times series than on the systematic signals. The transient signals, assuming they can be distinguished from noise, are a potential phenomenon of interest which has not been routinely addressed.

Various approaches particularly in the field of Data Mining and Knowledge Discovery (DMKD) have also focused on converting high dimensional, complex data such as time series into understandable information (Ultsch, A method for temporal knowledge conversion, 1999; Hoppner, Discovery of core episodes from sequences, 2002). Knowledge Discovery (KDD) has been loosely described as 'methods and techniques for making sense of data' (Fayyad, Piatetsky, & Padhraic, 1996), and so by definition, results of KDD are expected to be 'more compact..., more abstract..., or more useful and understandable'. The EO approach is a form of KDD, as resulting events have the characteristics of more compact, more abstract, and potentially more useful and understandable forms of information.

We demonstrate the workings of the EO approach in the context of detecting high-level storm events from sensor data streams collected by the Gulf of Maine Ocean Observing

System. This chapter contains the overall outline for the work, motivation, problem statement, definition of terms, and research objectives.

1.1 Motivation

Recent advancements in sensor technology have made sensors suitable for applications involving monitoring, detection, and surveillance. Sensor devices monitor the environment by producing a measurable response to changes in the physical surroundings. Monitoring of multiple physical quantities, i.e., parameters in the environment, produce multivariate data. Use of WSNs and their spatio-temporal aspects thus add dimensionality to the data. Sensors are deployed mainly in two ways: 1) far from actual phenomenon e.g., remote sensing, and 2) very close or embedded in the monitored phenomenon (Intanagonwiwat, Govindan, & Estrin, 2000; Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002). In both cases, sensors monitor an activity phenomena or identify target occurrences within some spatial setting.

Data collected over time using sensors are most often represented as a time series. Mining time series for interesting occurrences has been a major area of research for several years across many disciplines (Allan, Papka, & Lavrenko, 1998; Guralnik & Srivastava, 1999; Padmanabhan & Tuzhilin, 1996). Most data mining techniques focus on analyzing time series with the goal of predicting future interesting occurrences in univariate (Weigend & Gershenfeld, 1994; Fawcett & Provost, 1999) and multivariate time series (Hoppner, Learning dependencies in multivariate time series, 2002; Morchen, Time series feature extraction for data mining using DWT and DFT, 2003). However, a gap exists between high-level cognitive events which humans can recognize and the

events detectable from sensor time series. In this work, we will refer to the events that humans can cognitively identify as *high-level* events, whereas the *low-level* or primitive events refer to constructs closer to the parent sensor time series with one or two levels of processing or aggregation. Low level events are referred to as primitive events in the reminder of this thesis and these terms are developed further in subsequent chapters.

High-level phenomena that humans commonly recognize as events are often complex multivariate phenomena that individual univariate sensors are incapable of detecting. Humans, through their accumulated knowledge, are adept at recognizing events such as storms, fires, disease outbreaks or traffic congestion. In general, individual sensors will not be able to detect such high-level and multivariate events as they evolve through time and space. However, they can detect components (spatial, temporal or thematic parts) of these events, and with sufficient domain knowledge, information from separate univariate sensors can be integrated spatially and temporally to identify high-level events of interest. Knowledge discovery methods described in the literature address some aspects of this problem by using techniques such as time series segmentation or time series clustering and classification. Few, however, address the problem of considering multiple time series over space. Moreover, these types of data mining methods are typically directed towards discovery of unknown patterns in the data. Therefore, existing methods in the literature are not yet sufficient to effectively integrate and synthesize multivariate data for identification and characterization of high-level events.

Claramunt and Theriault (Claramunt & Theriault, 1995) have suggested that a temporal GIS needs to be capable in ways that support monitoring and analysis of successive states of spatial entities. The EO approach provides a model for such successive states through

primitive events and assembly of these into composite events. The proposed EO approach provides a method for reducing dimensionality in data while providing building blocks of observable system states and facilitating integration of temporal states over space. The EO approach is illustrated using GOMOOS data to detect storm events as an example of high-level events. A storm event is a meteorological event that humans cognitively recognize using their knowledge and senses. To attain the goals of the EO approach, this thesis takes a two tier approach. First, simple patterns or states are identified within individual time series in a manner similar to many other time series data mining and feature detection methods. The second step identifies and assembles high-level events from the primitive events based on specification of a high-level event ontology. A high-level event ontology is a conceptual representation of the high-level event that specifies its structure in terms of primitive events.

1.2 Problem Statement

In general, the problem addressed by this thesis is a spatio-temporal data mining problem. The specific objective is to identify and characterize a high-level event such as a storm from a set of sensor time series distributed in space. A typical characteristic of this setting is that temporal resolution is high and spatial resolution is relatively coarse. The problem has analogies to spatial feature extraction from images and temporal feature extraction from time series. The feature of interest in this context is an event with spatial properties (e.g. spatial extent) that is evolving over time. The setting includes sets of sensor nodes distributed in space with each observing one or more *parameter*s on regular time intervals. Any one sensor node location may see partial evidence of the high-level event

but not a complete picture. Two approaches might be considered for this problem. One is to monitor each time series separately and check for high-level event signals at each location. The other is to combine and process a collection of signals synoptically. Several methods exist to monitor time series and detect events, some of which are parallel and some consensus based, but as Neill (Neill, 2009), indicates, these methods do not account for the spatial context of the event detection problem. The time series either have no relevant spatial location (e.g., financial) or are processed independently of space (e.g., industrial process time series).

Machine learning approaches have used one or more time series from a single location but do not typically address the discovery of spatio-temporal features over time series from multiple locations. Spatio-temporal scan statistics detect events by searching for spatio–temporal clustering of a single type of event e.g., an Emergency Department visit (Kulldorff, 1997; Neill, 2009). The approach developed in this thesis is to first define sets of domain independent primitive events which describe basic states or changes in the state of a parameter and serve as building blocks for assembling high-level events. We assume that *a posteriori* knowledge exists for the high-level event, that it has characteristics that are detectable using available sensors, and that its constituent components are expressible in an ontology. Ontologies, were chosen because of their ability to formaly express concepts, relations and rules of the high-level event.

We chose to detect storms because *a posteriori* knowledge about temporal patterns and the role of low pressure system in formation of storms is well documented within the domain of meteorology. For example, the National Weather Service glossary defines a cyclone as large scale circulation of winds around a central region of low atmospheric

pressure. It has been well understood by meteorologists that formation of a low pressure point within the atmospheric system causes wind flow resulting in storms. Since barometric pressure is a vital indicator of storm dynamics, we call it a marker parameter. In this thesis, the term marker parameter indicates a sensor-readable parameter that is sensitive to the advent, progress and termination of a high-level event.

Given a set of multivariate sensor nodes generating time series that cover a region, the steps involved in the EO approach are as follows.

1. Specify an ontology for the high-level event in terms of composition of primitive events
2. Detect constituent primitive events from univariate time series using abstraction functions.
3. Assemble constituent primitive events into composite events according to event-event relationships specified by the high-level event ontology.

The resulting high-level composite events could be compared to an independent source, thereby evaluating the effectiveness of the approach. The high-level composite events could be further processed to classify them in various ways, e.g., on the basis of constituent primitive events and their characteristics. After detecting storms, presented as a case-study in the later part of the thesis, they are classified into several classes.

Ideally, an outcome of this approach is new ontological knowledge about the high-level event creating a feedback loop. Figure 1.1 shows such a feedback loop in which available ontological knowledge about the high-level event may be refined by using the abstractions of sensor data in the EO approach as evidence.

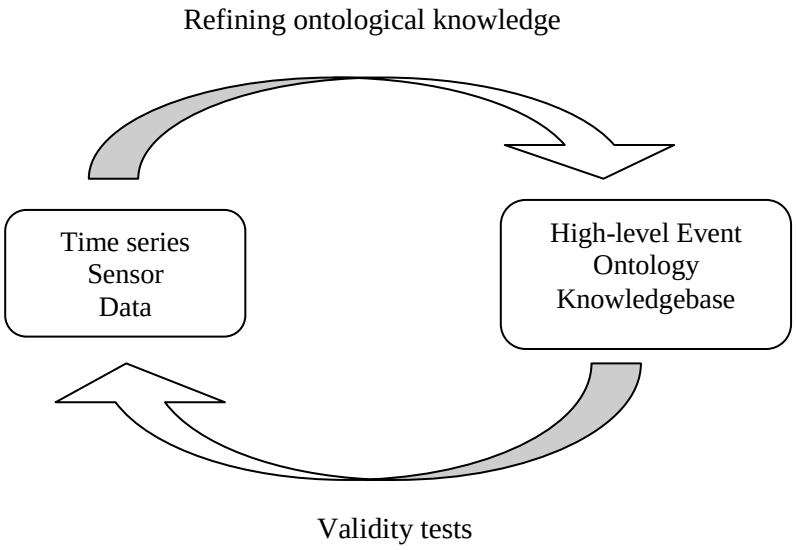


Figure 1.1 Refinement of posterior event knowledge expressed in an ontology through new empirical data obtained from sensors

1.3 Data Formats, Terms and Definitions

To set the scene for this approach, this section introduces key terms and concepts that will be used throughout the thesis. To describe abstraction of high-level concepts from time series data, this thesis makes use of Shahar's temporal abstraction framework (Shahar, 1997).

1.3.1 Time Series

According to Tufte, "The time-series plot is the most frequently used form of graphic design. With one dimension marching along to the regular rhythm of seconds, minutes,

hours, days, weeks, months, years, or millennia, the natural ordering of the time scale gives this design a strength and efficiency of interpretation found in no other graphic arrangement.*"* (Tufte, 1983)

The Engineering Statistics Handbook describes time series as an ordered sequence of values of a *parameter* at equally spaced time intervals. A parameter is a measurable aspect of a phenomenon e.g., wind speed, barometric pressure, air temperature etc., obtainable from a sensor. In this thesis, a time series is the output from a sensor observing a parameter in a specified spatial setting at regular time interval where each observed value of the parameter is associated with a time stamp. A sensor platform is assumed to have a fixed location and thus the generated time series are associated with fixed three dimensional locations. Multiple time series may be collected on the same parameter at different locations in the same time frame. For example Figure 1.2 shows several time series plots of the parameter, barometric pressure, obtained from moored ocean buoys deployed at different locations. Each buoy generates a different time series for the parameter based on a common time interval. Additionally multiple time series may be collected on different parameter at the same locations in the same time frame (e.g. air temperature and wind speed and barometric pressure are each collected at the same location).

There are two types of time series: discrete and continuous. A time series is said to be 'discrete' when observations are taken only at specific times and 'continuous' when observations are continuous in time. This thesis deals exclusively with discrete time series. The interval on which an observation is made defines the granularity of the time series. Integration of time series with multiple granularities is beyond the scope of this

thesis. This thesis assumes time series of the same granularity although this is not a constraint on the approach. From previous literature on time series analysis, two main purposes can be identified: (i) understanding and modeling the stochastic mechanisms that give rise to the observed series and (ii) predicting the future values of a series based on the history and other related series or factors. We assume that the time series is stochastic; that is the future may only be partially dependent on past behavior. The scope of this thesis is limited to explaining and describing behavior of the observed stochastic mechanisms; through detection and classification of events embedded in the time series.

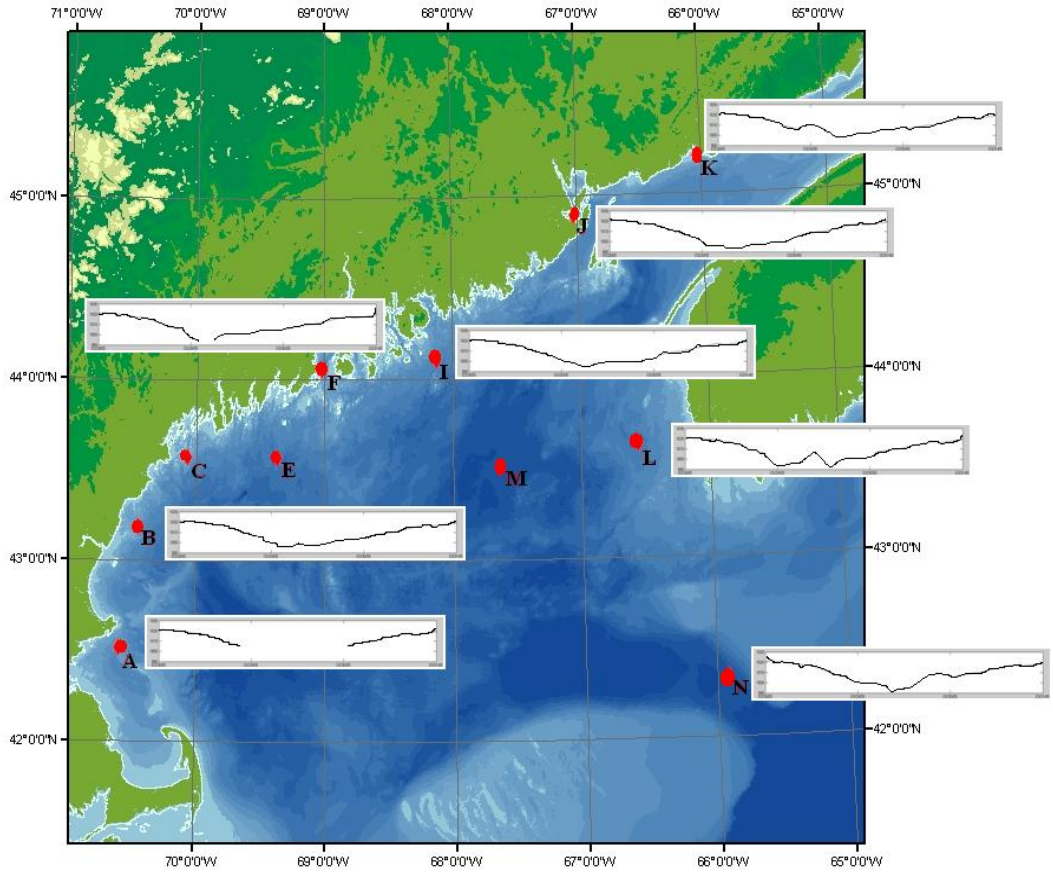


Figure 1.2 Time series plots of barometric air pressure at certain buoy locations between date-time interval [03-28-2005 00:00] to [03-31-2005 00:00]

1.3.2 Events

In this thesis, events are basic spatio-temporal entities and the basis for the EO approach. It is therefore important to note that this definition of event may differ from other definitions found in the literature such as Chakravarthy where events correspond to database operations (Chakravarthy, Krishnaprasad, Anwar, & Kim, 1994); in Grenon and Smith where events are purely instantaneous temporal entities (Grenon & Smith, 2004) or Shahar (Shahar, 1997) who defines an event purely in terms of external volitional action. The Basic Formal Ontology (BFO), a widely accepted theory of the basic structures of reality, endorses a view of the world containing Occurrents and Processes (Bittner & Smith, 2003). According to this view, Occurrents are bound in time whereas Processes persist (perdure) in time. The BFO describes events as entities which exhaust themselves in single instances of time.

In this thesis, events can be instantaneous or have duration. Therefore, they are conceptually similar to Occurrents in the BFO. We consider two distinct sets of events: high-level events called composite events and low-level events called primitive events. A primitive event is an abstraction from a univariate time series indicating a change in one parameter as observed by a time series. A ‘rise in wind speed’ is an example of a primitive event as it represents a change in a single parameter. High-level events may have two conceptualizations; one as a gestalt view in which the whole pattern of a physical, biological, or psychological phenomena is so integrated as to constitute a functional unit with properties not derivable by summation of its parts, versus a partitive view in which the high-level event is derived by composition from component parts and its individual factors are understood as contributing to an understanding of the

phenomena. From a partitive view, high-level events such as storms, forest fires, traffic congestion etc., can be seen as assemblies of primitive events. In this thesis, a composite event is assumed to be formed from several primitive events of the same type and parameter or more commonly from primitive events of different types and parameters.

1.3.3 Primitive Events

Primitive events are univariate entities and the first level of abstraction from sensor time series. A primitive event can be seen as a qualitatively significant change in the behavior of a dynamic phenomenon (Guralnik & Srivastava, 1999) or as representative of a particular state of a phenomenon. In this thesis, a primitive event represents an observable state of a *parameter*, where a *parameter* has several possible observable states. Time series are converted into these states S; where each such state represents a property that holds true during an temporal interval [t1, t2] defined by a beginning and ending time stamp. Shahar describes the process of abstraction of time series in terms of *abstraction functions*, unary or multi-argument functions from one or more *parameters* into a sequence of abstract states (Shahar, 1997). These abstract states correspond to primitive events as defined in this thesis. Primitive events as outputs of *abstraction functions*, are characterized by *abstraction types* e.g. value state, gradient, or rate. For example, a primitive event, obtained from a time series on the parameter barometric pressure might be: *significant barometric pressure fall below 15 millibars*. This primitive event has a value state: *lower than 15 millibars*, a gradient: *fall*, referring to the sign of the derivative of the parameter's value and a rate: *significant* as set by a threshold.

Different methods corresponding to Shahar's abstraction functions such as wavelet analysis, change point detection (Basseville & Nikiforov, 1993; Bakshi, 1999), and time series segmentation (Keogh, Chu, Hart, & Pazzani, 2003) can be used for identifying qualitatively significant change from time series data (Beard, Deese, & Pettigrew, 2007). We use statistically-derived or user defined thresholds to abstract time series into primitive events. Fuzzy thresholds may also be used to extract primitive events with fuzzy boundaries to accommodate uncertainty in primitive event definitions. The abstraction function or primitive event detection process determines the start and end time of the primitive event and additionally may generate statistical information such as an average magnitude for a primitive event. The location of the primitive event is inherited as the location of the sensor platform from which the generating time series was obtained. Primitive event detection and storage are presented in Chapters Three and Four.

1.3.4 Composite Events

In this thesis, a composite event is an aggregate of primitive events. Primitive events constituting a composite event are organized into temporally structured initiating, body and terminating event sets based on available ontological knowledge about the high-level event. Therefore, composite events are assembled from primitive events using *a posteriori* domain knowledge about a high-level event. In the context of this thesis composite events are derived indirectly from univariate or multivariate, multi-location sensor data streams through primitive events. For example, primitive events may be observed at multiple locations and aggregated to represent a spatially extensive composite event in a common timeframe. Composite events are composed from one or

more types of primitive events. For example, a typical storm is known to have barometric pressure drop as an 'initiating primitive event' followed by a set of other primitive events which may involve combinations of change in wind direction, variation in wind speed or wind gusts and followed by a 'terminating primitive event' such as barometric pressure recovery. Based on the available knowledge about the high-level event, a composite event ontology specifies how a composite event is formed from primitive events. This process will be illustrated using storms as the example high-level composite event.

1.3.5 Composite Event Validation

To assess the EO approach, high-level composite events detected using the EO approach are compared to an independent data source. In our case study, we use the National Climatic Data Center (NCDC) storm events database, to validate the storm events discovered through the EO approach. Type I and II error information is stored with each composite event record and can be utilized during further processing such as event classification.

1.3.6 Composite Event Classification

Composite event classification quantifies the degree of similarity and difference between discovered composite events based on a set of classification criteria. Classification of composite events may be based on various themes: magnitude, temporal and spatial sequencing of a primitive event type or a combination thereof. For the storm event detection case study, we use classification based on barometric pressure recovery

characteristics along with spatial sequencing and uniformity in spatial behavior among primitive events to summarize and classify storms.

1.4 Objectives, Scope and Hypothesis

The objective of this thesis is to propose, describe and implement a data abstraction approach named as the EO approach. The EO approach differs from existing approaches because: (1) it facilitates data integration from low-level sensor time series extracts to high-level occurrences, and (2) facilitates extraction of transient or non-systematic components from time series. The scope of the work is limited to proposing and implementing the EO approach, followed by validation of results using an independent data source.

The hypothesis tested in this thesis is: *High-level, spatio-temporal occurrences can be detected using low-level sensor measurements.*

As an outcome of the stated hypothesis, we might be able to answer questions such as:

- To what degree can high-level composite events be detected by combining univariate primitive events based on a posteriori knowledge of the composite event?
- To what degree can spatial location of primitive events be used to infer spatial properties of composite events? Thus, how well can the spatial extent and movement of a high-level and multivariate occurrence be determined from univariate spatially distributed primitive events?

- What new information can be derived from detection and classification of primitive events?

1.5 Organization of Remaining Chapters

The following chapters are organized to describe the EO approach in detail. The next chapter provides background and review of similar approaches and supporting literature. Chapter Three provides a detailed specification for primitive events through a primitive event ontology and the specification of composite events through a composite event ontology. A case study for storm detection using Gulf of Maine Ocean Observation System data is presented in Chapter Four and Five along with a description of the implementation and results. Classification methods for composite events based on primitive event characteristics are presented in Chapter Six. Chapter Seven provides conclusions and describes future work.

Chapter 2

LITERATURE REVIEW

This chapter reviews areas of research related to the Event Oriented approach. The work derives from various fields such as time series analysis and data mining, temporal abstractions, and anomaly detection. As event detection and composition are key components of the approach, this chapter reviews related literature on these topics. Literature particularly relevant to event detection, classification and validation are presented. Related approaches to similar problems are also summarized. We start with the general domain of our work, followed by related work on temporal abstractions, other event based approaches, and general methods for primitive event detection and event composition.

2.1 Data Mining and Knowledge Discovery

The approach utilized in this thesis can generally be categorized under the field of Data Mining and Knowledge Discovery (DMKD). The terms data mining and knowledge discovery are not mutually exclusive terms and at times have been used as synonyms. The topic Knowledge Discovery and Databases (KDD) provides an apt description of this work. On an abstract level, KDD can loosely be described as 'development of methods and techniques for making sense of data'. Thus, by definition, results of KDD have to be 'more compact..., more abstract.., or more useful and understandable. Data mining is defined as the application of specific algorithms for extracting patterns from data. We adopt the above stated definitions over all the other definitions available in literature.

Knowledge discovery has also been defined as *'mining of previously unknown rules... '* (Morchen & Ultsch, Optimizing time series discretization for knowledge discovery, 2005), but this thesis uses considers knowledge discovery in a broader context. Our objective is to develop an approach to detect a high-level occurrence using time series data. *A posteriori* knowledge about the relationships between time series parameters that form a high-level occurrence is assumed to be available. Validation of the detected high-level occurrence by comparison with an independent source will lead to revision or addition to the *a posteriori* knowledge about the occurrence.

Detection of interesting patterns from data has been one of the standard problems of the data mining community. Data mining typically requires transformation of the data to new representational forms that can simplify pattern detection and detection of patterns at different scales. Many time series based data mining methods reduce time series to a few important features. These features can be the coefficients of Discrete Fast Fourier (DFF) transforms, Discrete Wavelet transforms (DWT), or principal components Analysis (PCA). In the context of the Event Oriented approach in this thesis, the features extracted from time series are primitive events. Some approaches to primitive event detection will be mentioned in section 2.3.

Once features have been isolated, various classification and clustering methods are typically applied. Classification methods include regression trees, decision trees, and clustering methods include K-means, etc. In this thesis the primitive events form building blocks that can be composed into high-level forms of events. Related work on event composition is described in Section 2.4.

2.2 The Event Oriented Approach

The thesis uses what we call an Event Oriented approach. Earlier work has used similar terms and this section described other event approaches and how the thesis research relates to these approaches.

The term 'event' has been used with different meanings in computing, mathematics, data mining, philosophy, and other domains. In computing, an event is usually referred to a software message indicating that something has happened, such as a mouse click or keystroke. In probability theory, the term event may mean one element from a set of outcomes. In philosophy, several theories exist about events. Jaegwon Kim proposed a 'Property-Exemplification Account of Events' and theorized that events are structures of three things: object(s), a property, and time or a time interval (Kim, 1969). Lewis (Lewis, 1973) theorized that events are merely spatiotemporal regions and properties (i.e. membership of a class). He defines an event as 'a class of spatiotemporal regions, both this worldly and otherworldly'. The only problem with this definition is that it only tells us what an event could be, but does not define a unique event.

Nagel (Nagel, 1979) describes events as follows:

In formal language, an event Y at the time T is caused by a preceding event X, if and only if Y is deducible from X with the aid of the laws L_T known at the time T.

…all that is important here is the recognition insisted upon by Hume that natural events (e.g., explosions, cell division, etc.) which are causally related are logically independent of one another. In natural sciences, events have a formal structure of a deductive

argument, in which the explicandum is a logically necessary consequence of the explanatory premises […]

These statements highlight the ambiguity associated with events as described in the literature. A simple definition of an event can be broadly defined as 'a segment of time at a given location that is conceived by an observer as having a beginning and an end'. Quine (Quine, 1985) describes events as units that can be localized in space and time, broken into sub-parts, and arranged in a taxonomical hierarchy. Peuquet (Peuquet, 2001) describes an event as a change in some location(s) or object(s). Chen and Jiang (Chen & Jiang, 1998) define an event as an application-driven concept that supports a cognitive interpretation of a significant pattern of change. Another definition is provided by Guralnik and Srivastava (Guralnik & Srivastava, 1999) as 'a qualitatively significant change in the behavior of some dynamic phenomenon'.

Zacks and Tversky (Zacks & Tversky, 2001) describe the generalized definition and structure of events and describe the object-oriented treatment of events. Their view supports the concepts of partology, taxonomy and causality with respect to events. This thesis takes an object-oriented approach to events, similar to Worboys and Hornsbys (Worboys & Hornsby, 2004) except from an object oriented perspective they distinguish two types of primary classes: objects and events.

A common theme among many of these definitions is that events are associated with change and localized in space and time. In this work, we consider events as change units. Primitive events are change units extracted from space-time data by a number of methods (regression models, Fourier analysis, wavelets) and typically with statistically similar change properties (e.g. a linearly increasing trend, convexly decreasing trend, a change in

direction). Primitive events represent low-level change units obtainable from individual sensor data streams (e.g. single univariate time series). Recent work in the Wireless Sensor Network community takes a similar view of events (Kapitanova & Son, 2009; Jiao, Son, & Stankovic, 2005; Li, Lin, Son, Stankovic, & Wei, 2004; Yin & Gaber, 2008). *Composite events are aggregates of primitive events from single or multiple sensor data streams* (Beard, Deese, & Pettigrew, 2007).

We define events as spatio-temporal entities related to quantifiable change units as suggested by Guralnik and Srivastava (Guralnik & Srivastava, 1999). Primitive events are statistically derived entities from temporal data stream(s), which usually have a spatiotemporal component.

2.2.1 Events as Objects

Humans use multiple sources of information in perceiving events, namely, partonomic relations, and perceptual event boundaries. Research by Zacks and Tversky (Zacks & Tversky, 2001) shows that humans use objective features of object-actor motion, perceptual causal properties, statistical patterns of occurrence and goal relations for indentifying events. Our choice of approach of identifying primitive events and partonomically constructing composite events closely follows human perception to understanding event occurrence.

Events can be regarded as objects as suggested by Zacks and Tversky (Zacks & Tversky, 2001) and Quine (Quine, 1985). Objects have boundaries in space (Michotte, 1963) and time. For example, the object pen takes up space, which can be perceptually identified.

The event of 'picking up a pen' has a start and an end time along with an associated place giving a spatiotemporal dimension to the event. In their work, Worboys and Hornsby (Worboys & Hornsby, 2004) define the GEM model which models events and objects in an information system on the basis of their structural similarities, while noting their differences.

As humans are known to perceive events by causal relationships, we briefly discuss the concepts of partonomy and taxonomy of events. This supports the rationale for taking the EO approach to multivariate time series integration. Partonomy captures the hierarchical relationship between parts and subparts (Miller & Johnson-Laird, 1976; Tversky & Hemenway, 1984). Partonomical relationships give rise to distinctive spatial configurations that can be useful in categorizing objects. Events, according to Rosch (Rosch, 1978), may also be classified on the basis of their shape.

Another common form of hierarchical structure is taxonomical structure, based on 'kind-of' relationship. The kind-of or is-A relationship exemplified by the statement, 'Eagle is a kind of bird,' creates a taxonomical hierarchy between objects (or events). Experimental results have demonstrated that humans perceive events as partonomically organized (Barker & Wring, 1954) as well as being capable of hierarchical organization. In psychological experiments; given an activity, people show good agreement on what constitutes a scene within an activity (Bower, Black, & Turner, 1979). Further, when people are presented with subordinate-level actions, people tend to make inferences up to the scene level. However, when presented with information at scene level, they are relatively unlikely to make downward inferences to the subordinate level (Abbott, Black, & Smith, 1985). This previous experimental work supports the EO approach to time

series abstraction and integration e.g., construction of composite events from primitive events, without explicitly supporting the reverse.

The important advantages of an object-event paradigm are:

- Events become distinct information objects directly available for query and analysis.
- Events founded on a common data model can facilitate integration, better qualitative information retention during abstraction and are suitable to be analyzed in contrast to available abstraction methods for disparate information arising directly from sensor data streams.
- Event level objects provide a closer match with scientific models.

2.3 Primitive Event Detection

This section presents literature related to discovery of primitive events from time series data. First, the section presents methods of time series representation and pre-processing before describing primitive event detection methods. Second, methods for identifying subsequences of interest from a time series sequence are presented.

Representation and pre-processing of time series data has been a challenging problem because of the difficulties in direct manipulation of continuous, high dimensional time series. Antunes and Oliveira (Antunes & Oliveira, 2001) presents a survey of methods for pre-processing and representing time series before data mining e.g., primitive event detection can be undertaken. Four types of time series representation are discussed: (1) Time-domain Continuous representation (involving minimal transformation of time

series), (2) Transformation based representation (involving transformation of time series from time to another domain), (3) Discretization based representation (involving translation of time series into sequence of alphabetic symbols) and (4) Generative models (involving use of statistical or deterministic models to obtain data). In this thesis, only Time-domain Continuous representation and Discretization based representations are used. The Time-domain continuous representation of time series is the only representation used for primitive event detection. Discretization based representation of time series is used for composite event detection and the construct Spatial Progression String (SPS), which are introduced in later chapters.

Primitive event detection is similar to the problem of discovery of subsequences within a large sequence based on constraints. A common method of finding constraint qualifying subsequences from a sequence is to use a sliding window to traverse the length of the sequence to find qualifying subsequences within the large sequence (Faloutsos, Ranganathan, & Manolopoulos, 1994). Shahar (Shahar, 1997) describes the use of abstraction functions e.g., thresholds, to abstract time series into parameter intervals e.g., primitive events. Ultsch (Ultsch, Unification-based Temporal Grammar, 2004) uses the term *succession* to refer to a primitive event like construct, which mainly serves as a qualitative descriptor of the time series.

2.4 Composite Events

In the literature, many approaches to time series data mining focus on compression of univariate time series into a few temporal features. There are a few works namely Guimaraes and Ultsch, (Guimaraes & Ultsch, 1999); Morchen and Ultsch (Morchen & Ultsch, Discovering temporal knowledge in multivariate time series, 2005); Höppner (Hoppner, Learning dependencies in multivariate time series, 2002) which focus on mining multivariate data. Although these methods work with multivariate data, Guimaraes and Ultsch (Guimaraes & Ultsch, 1999) focus on generating understandable linguistic descriptions of complex multivariate patterns; Morchen and Ultsch (Morchen & Ultsch, Discovering temporal knowledge in multivariate time series, 2005) focus on generating semiotic descriptions of multivariate patterns; Höppner (Hoppner, Discovery of core episodes from sequences, 2002) focuses on finding previously unknown but frequently occurring dependencies in multivariate data. However, most of this and other previous work focuses on searching previously unknown but frequently occurring patterns and rules. Most rule generation approaches search for rules which describe an unknown pattern, predicting a predefined event (Povinelli, 2001; Agrawal, Psaila, Wimmers, & Zait, 1995).

The proposed EO approach supports integration based on knowledge about causality of events, allowing us to take a top-down/ bottom-up approach to event composition. Causality is the key feature for definition of composite event structure. Physical causality is governed by the phenomenon of amplification of motion, supporting the composite event construction approach (Michotte, 1963). More information on Generative models

can be found in Antunes and Oliveira (Antunes & Oliveira, 2001). Some of the closely related approaches on event composition are discussed further.

The Unification-based Temporal Grammar (UTG) is a rule language developed especially for the description of patterns in multivariate time series (Ultsch, Unification-based Temporal Grammar, 2004). UTG uses first order logic and offers a hierarchical description of temporal concepts. It presents an abstraction hierarchy that starts with primitive patterns extracted from raw data, followed by Successions, Events, Sequences and the final rules called Temporal Patterns, as illustrated in Figure 2.1. At each level of abstraction, the grammar consists of semiotic triples: unique symbol, grammatical rule and user-defined labels.

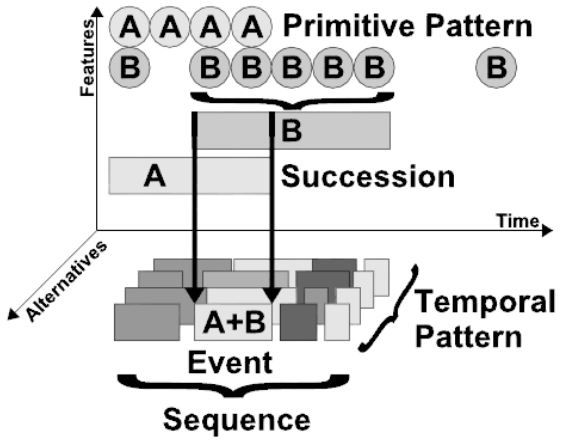


Figure 2.1 Abstraction levels in Unification-based Temporal Grammar (UTG) (*Source:* (Ultsch, Unification-based Temporal Grammar, 2004))

As a first step of UTG, a time series is abstracted into symbol labels to form 'primitive patterns'. A primitive pattern is the assignment of a single point in time to one of a number of the possible states (Ultsch, Unification-based Temporal Grammar, 2004).

Such abstraction is limiting because as the user must is forced to assign symbols for each time step for a qualitative classification of time series. The increase in symbol classes increases the complexity for in further abstraction. In UTG, 'successions' introduce the temporal concepts of duration and persistence. Successions are derived from similar adjacent primitive patterns. The disadvantage of this method is that when missing data of short duration is present between successions, two successions will be shown instead of one. UTG uses the term 'Events' for representing the temporal concept of synchronicity in sequences. If two or more successions occur simultaneously, they form an Event. An Event in UTG is therefore, a univariate symbolic Event series formed from multivariate successions. Since multivariate successions are unified into one symbol representing an Event, each succession has the same weight within the combined event symbol. This is a disadvantage in comparison to EO approach which allows one or more parameters to be 'marker parameters' having different or maximum weight. Unlike the EO approach, UTG approach makes isolating patterns based on trends difficult. The interested reader can refer to Ultsch (Ultsch, Unification-based Temporal Grammar, 2004) for further study of UTG.

Another related method of composite event construction was described by Höppner (Hoppner, Discovery of core episodes from sequences, 2002). Höppner's approach segments time series into sequences of labeled intervals. Magnitude or other quantitative descriptors of a segment are lost in Höppner's method during abstraction. The labels denote qualitative only aspects of the signal in the respective intervals as shown in Figure 2.2. These sequences of labeled intervals are used to discover rules such that premise and

conclusion consist of temporal patterns. The rule discovery method used by Höppner is between a sequence of two intervals as shown in Figure 2.3.

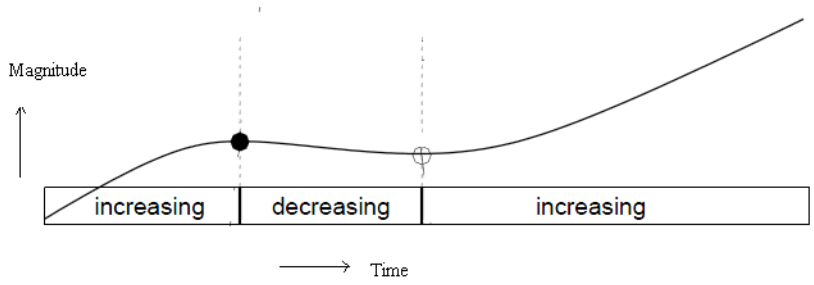


Figure 2.2 Conversion of time series into qualitative segments called *state intervals*.

(Source: (Hoppner, Knowledge discovery from sequential data, 2003))

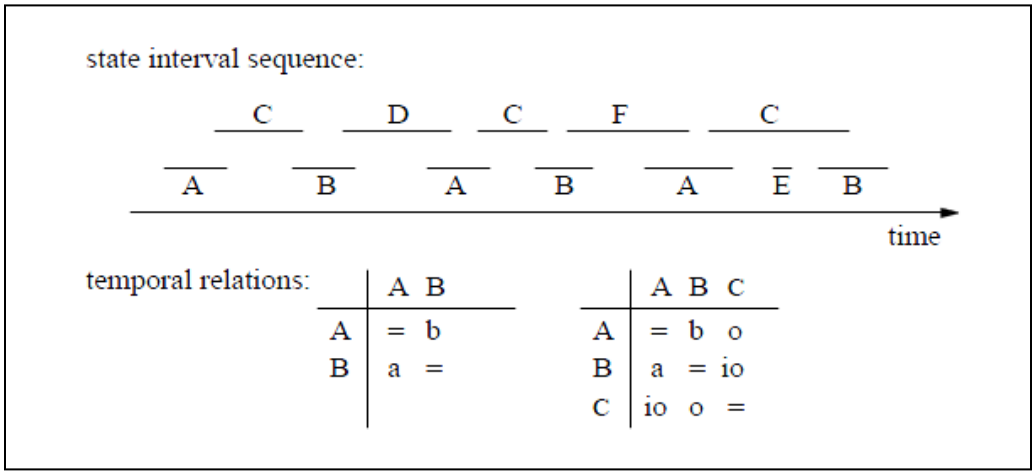


Figure 2.3 Rule discovery between labeled interval sequences

using Relationship Matrix.

(Source: (Hoppner & Klawonn, Finding informative rules in interval sequences, 2002))

A limitation in Höppner's method is that it depends completely on Allen's intervals relationships (Allen & Ferguson, 1994) for rule discovery in subsequences abstracted from time series. As discussed in Morchen and Ultsch (Morchen & Ultsch, Efficient

mining of understandable patterns from multivariate interval time series, 2007), patterns from noisy interval data expressed in Allen's interval relations are not robust, unambiguous, or easily comprehensible. This view is supported by Batal et. al. (Batal, Sacchi, Bellazzi, & Hauskrecht, 2009) who only use *before* and *overlaps* relationships due to imprecision in the event extraction. In comparison to Höppner's method, the main advantage of the EO approach is the ability to construct composite events based on the duration of overlap, instead of using the interval relationships. When applied to the storm events detection case study, Höppner's method could not be used to much advantage since inter-event temporal overlaps are more important than inter-event relationships. For example, the number of hours in temporal overlap between events 'significant fall in barometric pressure' and 'high wind speed' was more important than other Allen's relationship between them such as whether one started or terminated the other. Therefore, the EO approach is not strictly reliant on temporal relationships but employs additional semantic relationships as specified by a domain specific ontology.

2.4.1 Ontologies for Composite Events

The term 'ontology' has different interpretations and meanings across several fields of study that has changed over time. Aristotle first attempted a complete ontology of reality stating it as 'all species of being qua being and the attributes which belong to it qua being' (Ross, 1924). The modern day Oxford English Dictionary describes ontology as 'science or study of being'. Our definition of ontology is adopted from Audi (Audi, 1995) which states: ontology is a study of explaining reality by breaking it down into concepts, relations and rules. We share our view of events as expressed by Allen and Fergusson

(Allen & Ferguson, 1994), where '...events are primarily linguistic or cognitive in nature'. According to this view, the world does not really contain events. Rather, events are the way by which agents classify certain useful and relevant patterns of change. This view additionally supports our view that events can be regarded as objects and therefore, can be expressed in ontology.

Ontologies have been used as a means of knowledge sharing across disciplines and improving interoperability among different geographic databases (Smith & Mark, 1998; Fonseca, Egenhofer, Agouris, & Camara, 2002). In the domain of information systems, ontologies have been described as dynamic, object-oriented structures that can be navigated (Fonseca, Egenhofer, Agouris, & Camara, 2002). Gruber (Gruber, 1991) described ontology as an explicit specification of a conceptualization. Ontology-driven information systems have been shown to act as system integrator, independent of the model of representation used in Fonseca et al. (Fonseca, Egenhofer, Agouris, & Camara, 2002). Sowa (Sowa, 1999) provided a domain-specific, user-dependant view of ontology as 'the method to extract a catalogue of things or entities (E) that exist in a domain (D) from the perspective of a person who uses a certain language (L) to describe it'.

There are several types of event ontologies available on the web. Each differs in its definition of events with temporal extents ranging from instantaneous, having duration or both. An upper-level event ontology developed by Center for Digital Music, University of London takes a purely linguistic and cognitive view stating that: event is 'an arbitrary classification of a space-time region by a cognitive agent'. Another available ontology, the Semantic Web for Earth and Environmental Terminology (SWEET) provides an

upper ontology for Earth system science in OWL language. SWEET contains an ‘Event’ class which is an ‘Occurrence’ and a ‘Temporal entity’.

The domain for the Storm event ontology is restricted to atmospheric events that are detectable using wireless sensors and represented in time series. The Storm ontology is domain specific and low-level. It will be used to specify the structure of initialization, continuance and termination of a high-level event.

2.5 Storm Detection

This thesis uses storms as a case study event type. This section describes related work on storm detection that is pertinent to our approach. Detection of severe climatic events from multivariate data has been a subject of much interest to climatologists since the early 1980s. Many methods initially tried to sort weather maps into a discrete number of weather types based on analysis of the atmospheric parameters (Muller, 1977; Brazel & Nickling, 1986). These attempts mostly dealt with reducing dimensionality of the dataset statistically into discrete classes. Notably, Davis and Rogers (Davis & Rogers, 1992) attempt developing a synoptic climatology for severe storms using Principal Component Analysis (PCA) with 21 years of multivariate weather data. PCA was used to reduce the high dimensionality into a smaller and manageable dataset of entirely uncorrelated parameters. The resulting orthogonal parameters are then clustered into homogenous groups so that each cluster represents a distinct meteorological situation, thus identifying storms.

Although as demonstrated in Davis and Rogers (Davis & Rogers, 1992), PCA is capable of identifying clusters of multivariate data which show statistically similar properties, there is no way to search complex events based on available ontological knowledge about them using PCA. For example, in the composite event ‘forest fire’, ontological knowledge such as the initiation by primitive event ‘spark’ and ‘rise in air temperature’ followed by ‘smoke’ and termination by ‘drop in air temperature’ and ‘drop in smoke’ cannot be included in searching for the composite event in PCA.

2.6 Summary

This chapter discussed literature and background of the EO approach. The concept of events including primitive and composite event detection and its relationship to other works is presented. Existing high-level ontologies were briefly discussed, and provide background for the presentation of ontologies in the next chapter.

Chapter 3

PRIMITIVE AND COMPOSITE EVENT ONTOLOGIES

Swartout et al. (Swartout, Patil, Knight, & Russ, 1997) describes an ontology as a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base i.e., objects, concepts or entities. This chapter describes ontologies for primitive and composite events along with a storm ontology. As introduced in Chapter One, primitive events are the first level of abstraction of time series, whereas composite events are assemblies of primitive events. The primitive event ontology presented in Section 3.1 explicitly states and describes the concepts in abstracting primitive events from time series data. These primitive events are domain-independent building blocks for composite events. Challenges imposed by missing data and uncertainties introduced during primitive event detection are also presented. Section 3.2 presents the general composite event ontology, which can be applied for assembling high-level composite events in many domains. For detecting storms from time series data, Section 3.3 presents the Storm Event Ontology, which is a specialized high-level composite event ontology specific to the domain of meteorology.

An ‘Event ontology’, from Raimond and Abdallah (Raimond & Abdallah, 2007) illustrated in Figure 3.1 provides the basic specification for an event.

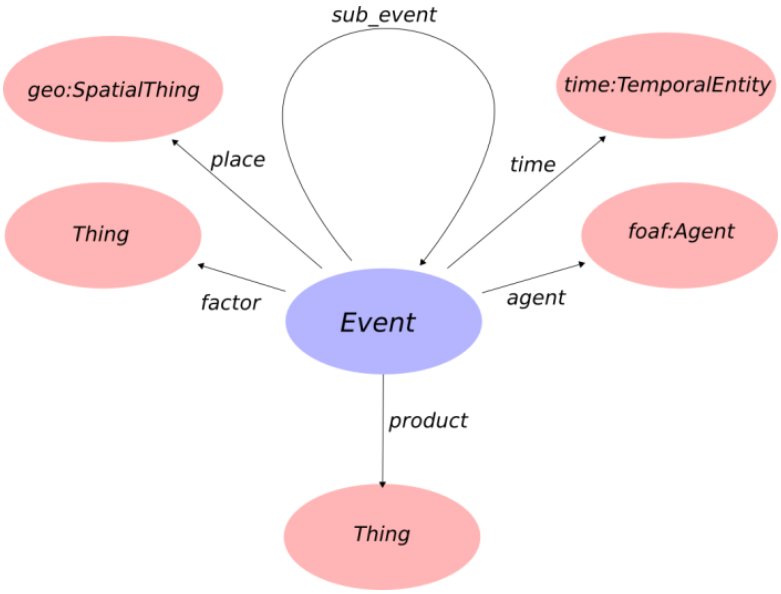


Figure 3.1 Event ontology

(Source: (Raimond & Abdallah, 2007)*)*

As shown in Figure 3.1, an event may have a location, time, active agents, factors and products. This concept of an event aligns with the Basic Formal Ontology (BFO) described by Grenon and Smith (Grenon & Smith, 2004). Both primitive and composite event ontologies developed in this thesis align with the basic spatial and temporal constructs of these event ontologies.

We use the ontology editor and knowledge acquisition system Protégé-OWL to develop the ontology for primitive and composite events. OWL or Web Ontology Language is a standard knowledge representation language for authoring ontologies. Protégé supports the creation, visualization and manipulation of ontologies in formats such as Resource Description Framework (RDF), the Web Ontology Language (OWL) and Extensible Markup Language (XML).

3.1 Primitive Event Ontology

The main purpose of the primitive event ontology is to enable common understanding of the structure of information and to explicitly state the concept of a primitive event. Additionally, it also describes pertinent relationships and sub-concepts of primitive event. Since primitive events are temporal abstractions from time series, Section 3.1.1 sets up definitions for key temporal concepts. Section 3.1.2 describes time series concepts before abstraction, followed by Section 3.1.3 which describes AbstractionFunctions as the concept for detection of primitive events from time series.

3.1.1 Temporal Concepts

This thesis borrows several time concepts from Shahar's (Shahar, 1997) Knowledge Based Temporal Abstraction (KBTA) framework and from OWL-Time. OWL-Time is an ontology that provides vocabulary for expressing facts about topological relations among instants and intervals, together with information about durations and date-time (Pan & Hobbs, 2005).

KBTA defines time stamps as structures (e.g., dates) that can be mapped by a time-standardization function into an integer amount of any element from a set of predefined temporal granularity units (G_i). Temporal granularity units are standard (e.g. minutes, days, hours) or domain defined (e.g. tidal cycle) units of time. A time measure is a finite negative or positive integer amount expressed in a G_i unit (e.g. 20 minutes, 3 days). According to Shahar, a domain must have a time granularity G_0 corresponding to the

finest granularity (e.g. seconds) into which integer amounts of other granularity units can be mapped.

OWL-Time's date-time corresponds to KBTA's timestamp. A zero-point is a time stamp which is grounded in each domain to different absolute 'real-word' time points (e.g. the beginning date of a sensor deployment). KBTA defines a time-interval as an ordered pair of time stamps representing the interval's start and end points. An interval may have more than one duration description given in different temporal granularity units (e.g. year, month, day, hours, minutes or seconds). All primitive events have an associated time interval given by time stamps which define the beginning and end of the time interval. OWL-Time uses Allen's calculus of binary interval relations to represent the qualitative temporal relationships between time intervals (Allen & Ferguson, 1994).

As pointed out in Section 2.4, only two relationships—'before' and 'overlaps'—from the Allen's interval relationships are used. This is due to the imprecision in primitive event start and end times and the small probability of such end times coinciding.

3.1.2 Primitive Event Ontology Concepts

This section describes concepts that play a role in describing a primitive event. As mentioned before, primitive events are obtained from time series.

As shown in Figure 3.2, the essential classes of the primitive event ontology are: Parameter, Value, MeasurementUnit, MeasurementScale, TimeSeries, TimeStamp, AbstractionFunction, Threshold, AbstractionType and PrimitiveEvent. All the concepts except PrimitiveEvent, Threshold and Event are as defined by (Shahar, 1997). We build

on these existing concepts to suit our purpose of specifying the domain independent primitive event ontology.

A Parameter is defined as a measurable aspect of a describable state of the world (e.g. salinity) and has properties that include MeasurementUnit, a domain of Values, and MeasurementScale (Shahar, 1997). The domain of Values can be symbolic or numeric. MeasurementUnits are the basic units of measure (e.g., meter for length, second for time etc.) or derived units (e.g., cubic meters for volume, degrees for angles etc.). The MeasurementScale defines the level of measurement (e.g., nominal, ordinal, interval or ratio) used in observing a parameter. A nominal MeasurementScale is one which applies categories (e.g., North, South, East, West) to represent direction.

The TimeSeries is a series of observations taken on a parameter over a period of time. Each observed value of the parameter is associated with a TimeStamp. Thus a time series is an ordered sequence of tuples (Value, TimeStamp) ordered according to the timestamp values. Another important property of a time series, in the context of this thesis is the location at which it is observed. A time series denoted as TS is indexed by its parameter (p) and location (x) and is represented as TS^{p}_{x}.

A primitive event is a subsequence of a time series for which a particular property of the parameter holds. In the primitive event ontology, the primitive event is connected to its source time series and by extension, to the time series parameter and location.

An AbstractionFunction is a function that converts a time series into a sequence of primitive events. There are many different possible subsequences to consider that could form primitive events depending on the interests of a researcher and thus a number of possible AbstractionFunctions. Simple AbstractionFunctions apply appropriate user

defined thresholds, for example to obtain subsequences that exceed or fall below some parameter value. Different types of thresholds that apply to the parameter value directly (e.g., State threshold) or to the derivative of the value (e.g. Gradient threshold and Rate threshold) create different AbstractionTypes. An AbstractionType is one of several possible abstract states generated by applying an AbstractionFunction to a time series. AbstractionTypes are represented by an alphabet, A of symbols, one for each possible state of a parameter. An AbstractionType can be a value type corresponding to classification of a parameter's value (e.g. high, medium, low) or a trend type corresponding to the derivative of the parameter's value. A trend type can have subtypes—gradient and rate—that correspond respectively to the sign and magnitude of the derivative of the parameter.

Thus a primitive event and an associated AbstractionType are the result of applying an AbstractionFunction to a time series. Shahar's 'abstract parameters' correspond to the primitive events in this work.

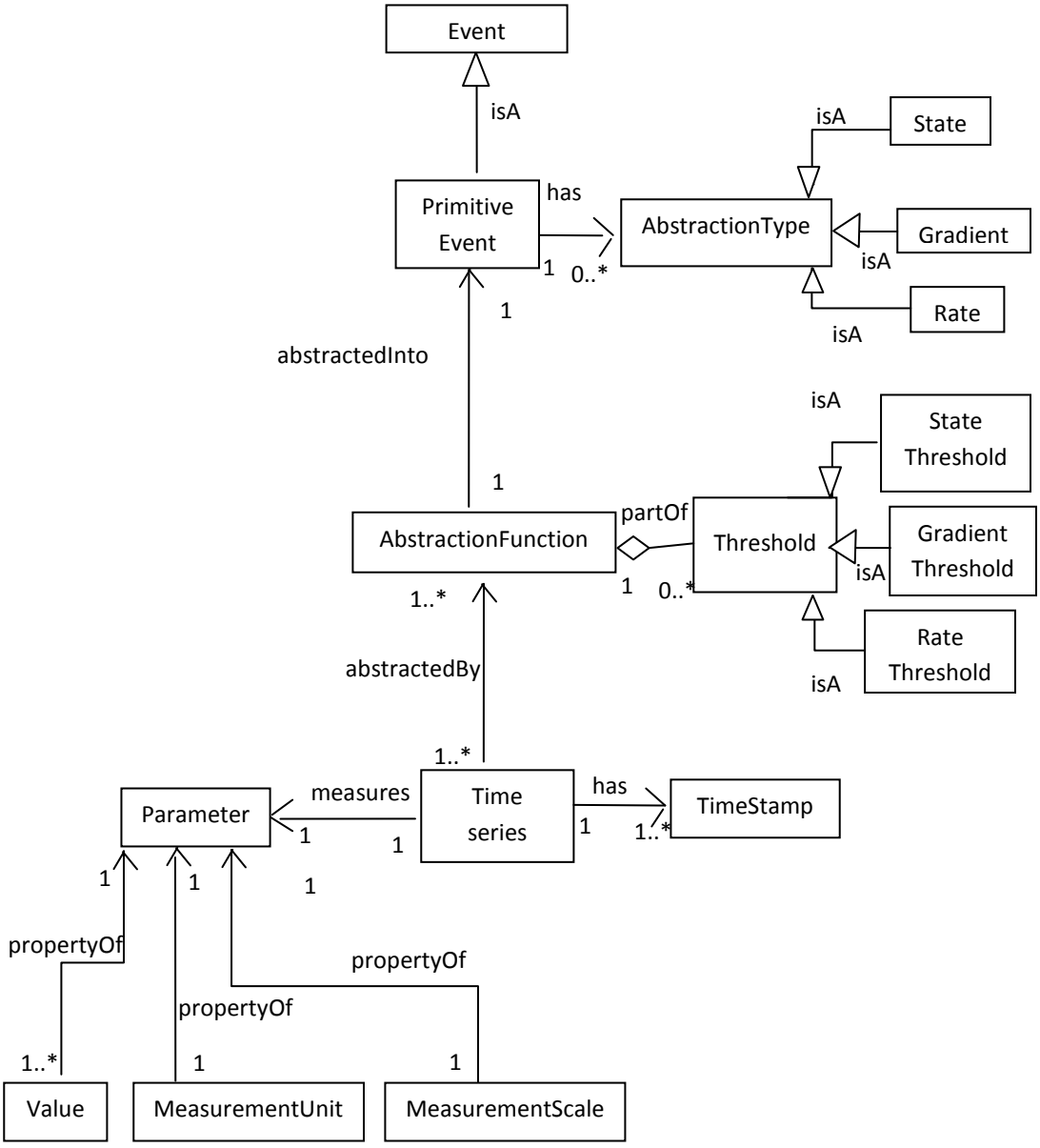


Figure 3.2 Primitive event ontology

3.1.3 Primitive Event Detection Using AbstractionFunction

As specified in the primitive event ontology, primitive events are generated by applying an AbstractionFunction to a time series. We denote a time series collected on a parameter at several locations in space by TS^{p}_{x} indexed by parameter (p) and location (x). When an AbstractionFunction is applied, it is applied to all time series for the parameter and all locations at which the parameter is measured. The AbstractionFunction establishes the time interval (begin time, end time) of a primitive event as well as the AbstractionType as described above. A primitive event and its time interval are denoted as PE[b,e]. All primitive events obtained by one AbstractionFunction share a parameter and AbstractionType denoted by two upper case letters representing the parameter (e.g. WS for Wind speed or BP for barometric pressure) and a symbol from some alphabet A that defines AbstractionTypes (e.g. A={rise, fall, steady}). When referring to a specific type of primitive event, we use this parameter_AbstractionType combination (e.g BP_fall to designate a falling barometric pressure primitive event). An instance of a primitive event of type given by parameter_AbstractionType has a time interval specific to a location and is denoted by $PE[b_x,e_x]$. The remainder of this section describes a set of AbstractionFunctions used in this thesis to obtain primitive events.

Any of the four types of time series representations mentioned in Section 2.3 could be used for primitive event detection but in this thesis, only time domain continuous is used for primitive event detection. Time domain continuous representation of time series is suitable for threshold conditions such as ‘less-than’, ‘more than’ or within threshold. Use of thresholds to abstract time series into primitive events using AbstractionFunctions is described in detail in the following sections.

Throughout the thesis, we employ a standard notation for addressing a particular row of a matrix (e.g., structure matrix shown in Figure 4.3). We use the notation '[matrix_name].[row_name]'. The 'row_name' corresponds to a predefined row label in the matrix and is usually a two letter label. For example, TS.jd refers to the row corresponding to the Julian dates of the matrices stored with the label 'jd'. Similarly, other 'row names' of TS correspond to other parameters detected by the sensor, such as 'at' for the row storing air temperature and so forth. In case where the row of the matrix is arbitrary, we use the convention '.xx' to denote the arbitrary row. The primitive event PE[b,e] can be formally represented by the following expression, hereby denoted as Expression 3.1:

The 'condition' used for identifying the primitive event depends on the AbstractionFunction used and will be discussed next.

3.1.3.1 Threshold Based Abstraction Functions for Primitive Event Detection

This section describes three types of threshold conditions: State, Gradient and Rate, which are used by the *AbstractionFunction* for detecting primitive events from time series data. Descriptions of each threshold type, their concept and formal expression are

presented. This section describes fuzzy thresholds but use of thresholds in this thesis is limited to ‘hard’ thresholds only.

3.1.3.1.1 State Threshold

State thresholds are numerically explicit bounds applied to the value of the time series parameter. State thresholds can either be statistically-derived or manually set to include certain specific characteristics thought to be important to the user. Manual threshold setting may be performed in case of poor data quality or for making thresholds suitably conservative. A state threshold applies a constraint on the ‘state’ of the parameter and yields primitive event types such as ‘high wind speed’, ‘cold air temperature’, ‘north-east wind direction’, ‘high wave height’. This section discusses the AbstractionFunctions using the three sub-types of state thresholds i.e., line, band and fuzzy state thresholds.

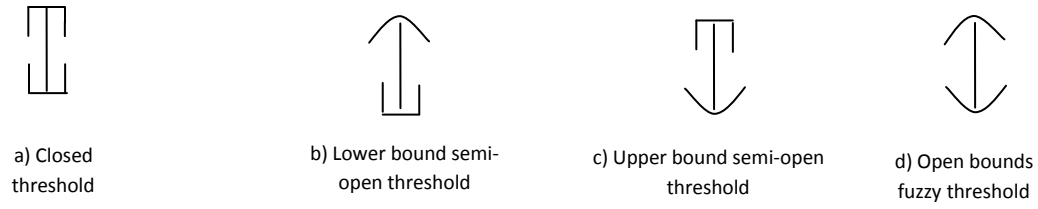


Figure 3.3 State threshold types in primitive event detection

3.1.3.1.1.1 Line State Threshold

A line state threshold is a simple AbstractionFunction that generates value abstraction types such as ‘greater than’ or ‘less than’. It can be an upper or lower bound semi-open threshold as shown in Figure 3.3 (b) and (c). Upper bound semi-open thresholds specify an upper bound and no lower bound. Lower bound semi-open thresholds specify a lower bound and no upper bound. For the primitive event PE[b,e] extracted from the time series TS_x^p (written simply as TS) the ‘condition’ used in the Expression 3.1 for:

Above line state threshold is $TS.xx[n] \geq h$

Below line state threshold is $TS.xx[n] \leq h$

where,

n= index of the array, h=threshold value.

For these two cases, the alphabet for AbstractionType is logically A={low for $p \leq h$ and high for $p \geq h$}.

3.1.3.1.1.2 Band State Threshold

A band threshold is a combination of two line state thresholds: *greater-than* and *less-than*. Types of band conditions could be: outside a band or inside a band. The band threshold is conceptually similar to the closed threshold shown in Figure 3.3 (a). Closed thresholds are bounded by both sides and represent conditions like ‘more than x but less than y’. They capture sequences that fall within specific ranges of parameter values, for

example, ‘a South-East wind direction’ primitive event has wind direction observations between 112.5 and157.5 degrees.

An *outside a band* threshold bound by magnitudes h_u (denoting upper threshold in band) and h_l (denoting lower threshold in band) is given by:

$$TS.xx[n] \leq h_l$$

$$\wedge$$

$$TS.xx[n] \geq h_u$$

Similarly, the condition for *inside a band* event threshold is given by:

$$h_l \leq TS.xx[n] \leq h_u$$

3.1.3.1.1.3 Fuzzy State Threshold

Use of a unit value, hard threshold creates a crisp boundary that ignores the ‘borderline’ cases that just miss qualifying a threshold. The EO approach can address the uncertainty in primitive event detection by use of fuzzy thresholds. Fuzzy thresholds are conceptually similar to the open bound threshold as shown in Figure 3.3 (d). The open bounded fuzzy threshold condition involves fuzzy thresholds. Every observation is included in the definition of the fuzzy event because each observation is given a membership function with respect to the threshold. Thus a hard threshold is also required on the membership function of the fuzzy boundary.

Uncertainty in fuzzy primitive events is discussed in Section 3.1.4., but implementing fuzzy thresholds is beyond the scope of the thesis.

3.1.3.1.2 Gradient Threshold

The gradient threshold is an AbstractionFunction that generates a trend AbstractionType primitive event corresponding to Shahar's (Shahar, 1997) *gradient* abstraction type which indicates the sign of the derivative of the parameter's value. Primitive event detection using constraints on gradient involves the transformation of time series into first difference sequences, denoted by *D.xx*. From first difference sequences, we extract primitive events with particular change characteristics such as change in direction or magnitude. Examples of primitive events extracted using first difference derivative are 'barometric pressure fall' (or rise), 'fall in wind speed', 'cold spike in water temperature', or 'change in wind direction'.

Formal expressions for extracting trend AbstractionType primitive events are:

$$PE[b,e]=$$

The "condition" for *Rising* primitive event is given by:

$$D.xx[n] \geq 0$$

and *Falling* primitive event is given by:

$$D.xx[n] \leq 0$$

where, $D.xx[n]=TS.xx[n+1]-TS.xx[n]$

and $|D.xx|+1=|TS.xx|$

3.1.3.1.3 Rate Threshold

The test condition for a rate threshold is the magnitude of the derivative of the parameter's value. For example, the AbstractionFunction to detect primitive event 'significant barometric pressure fall' requires a gradient threshold to detect a fall gradient, followed by a rate threshold to test the 'significance' of the condition. If a fall of more than 10 barometric pressure units was considered to be 'significant', then it is the rate threshold. The rate of gradient is calculated over the duration of the interval as follows:

$$S = (M_{max} - \quad) / (\text{length } PE[b,e])$$

where,

b and e= begin and end time of primitive event PE[b,e]

Duration of PE[b,e]= (e-b+1), if 1 hour is the granularity

M_{max} and M_{min} = Maximum and minimum magnitude for event PE[b,e]respectively

S = Gradient of event PE[b,e].

The rate of the gradient within the interval of the primitive event can be compared against the user-defined or statistical rate threshold to determine if the event meets the rate threshold.

3.1.3.1.4 Combination of State, Gradient and Rate Thresholds

In this thesis, generally a combination of state, gradient and rate thresholds is used to detect primitive events. These threshold conditions are generally applied in steps to get the desired primitive event. For example, a 'significantly falling low barometric pressure event' first involves application of a state threshold to test the 'low' condition followed by the gradient threshold to test the 'fall' condition, followed by the rate threshold to test the 'significant' condition.

3.1.4 Uncertainty and Fuzzy Primitive Events

Vagueness is the presence of border line cases (Russell, 1923). Vagueness in primitive events can be dealt with by use of fuzzy logic, which provides a method to assign a fuzzy membership function to time series observations according to their position with respect to the hard threshold. We use fuzzy set theory to introduce fuzzy thresholds for detection of fuzzy primitive events. Fuzzy thresholds assign a continuum of 'grades of membership' between the interval [0, 1] to every observation in a time series. Thus, an observation nearer the threshold gets a higher grade of membership (Zadeh, 1965). Observations clearly meeting the threshold are given a grade of membership of 1. As we move farther from the threshold, the grade of membership decreases (See Figure 3.4). Thus, each observation has a grade of membership with respect to its distance from the threshold. The user can assign a 'non-trivial threshold' on the grade of membership, assigning the level at which the grade of membership be considered non-trivial. In general, a 'non-trivial threshold' of 0.4 is used.

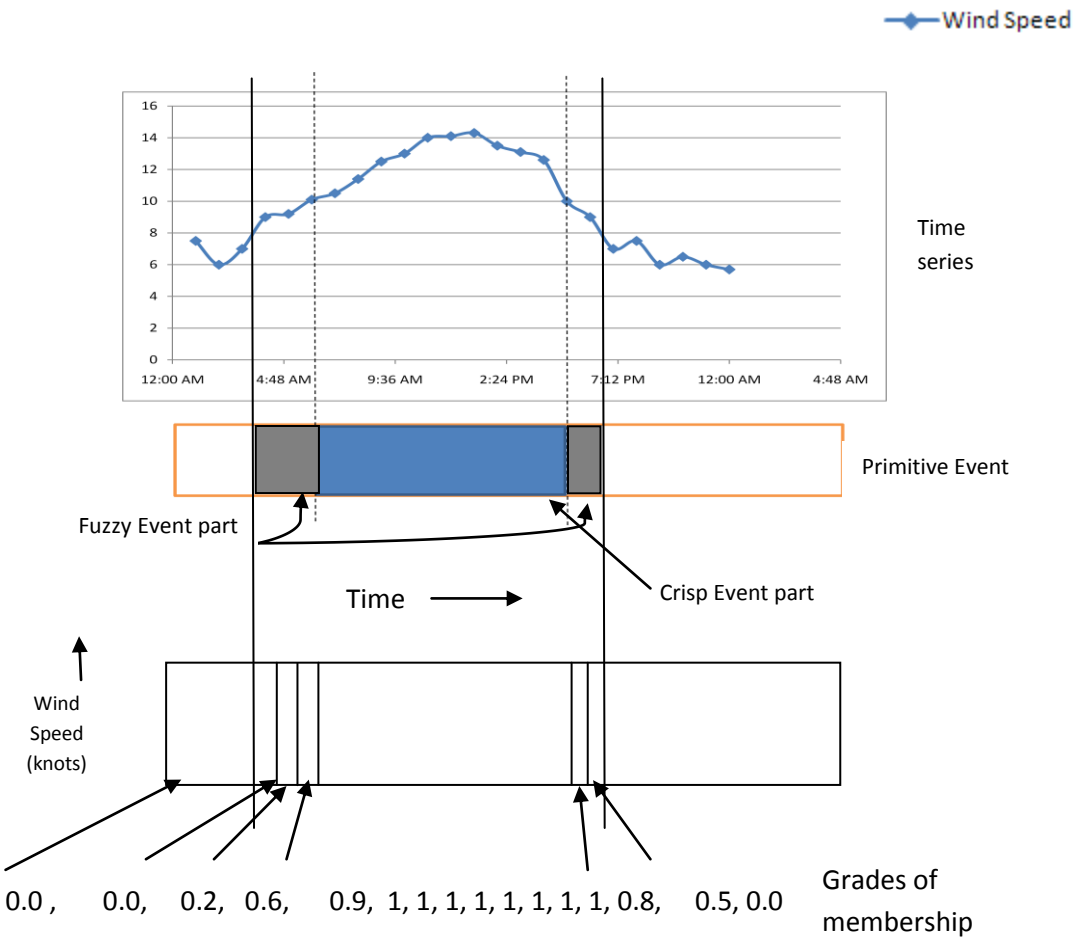


Figure 3.4 Assignments of degree of membership while definition of fuzzy events

Events extracted using fuzzy thresholds are called fuzzy events, because they have a fuzzy boundary defined by membership values.

3.1.5 Missing Data Events

Missing data is a common problem in sensor systems. Missing data may indicate several states of data, such as 'sensor failure', 'don't know', 'refused', 'unintelligible', 'noisy' etc. (Schafer & Graham, 2002). There have been several attempts to reduce missing data values through efficient system design (Stann & Heidemann, 2003). However, missing data is a frequent problem in sensor systems, as sensors may be subjected to harsh conditions in high energy systems and design limitations. Some statistical treatments are not possible in the presence of missing data. Missing data is integral while changing the granularity of data e.g. when data is grouped, aggregated, rounded, censored, truncated or processed for noise (Heitjan & Rubin, 1991). Rubin proved that while making inferences like sampling distribution, direct likelihood or Bayesian inferences about the data, 'it is appropriate to ignore the process that causes missing data, if the missing data are 'missing at random'' (Rubin, 1976).

Rubin defined data to be *missing at random* if: 'for each possible value of the parameter p, the conditional probability of the observed pattern of missing data, given the missing data and the value of the observed data, is the same for all possible values of the missing data'.

Most sensor data cannot be guaranteed to be '*missing at random*'. In the event approach, missing data sequences are treated like other primitive events. They correspond to a subset of a time series with start and end time stamps denoting the beginning and end of the missing data sequence. Like any other primitive event, they have an associated parameter and a location that corresponds to the sensor location. The next section

describes how the primitive events form building blocks for assembling composite events in various domains.

3.2 General Composite Event Ontology

A composite event is a temporally ordered sequence of primitive events. A general composite event ontology describes the structural organization of a composite event in terms of primitive events. The general composite event ontology is centered on the notion that any high-level composite event has three components: initiating, body, and terminating components made up of primitive events, as shown in Figure 3.5. This structure can be used to describe a wide range of high-level events such as storms, rain events, snowfall, flooding, forest fires, or traffic jams. The common theme among these high-level events is that they are spatio-temporal events which have distinct low-level initiating and terminating behaviors that are sensor-detectable. Some domain knowledge about the high-level event, in terms of initiating and terminating behavior of low-level parameters is assumed to be available. For example, extreme weather events i.e., storm, rain or snowfall, are typically initiated by a significant fall in barometric pressure and terminated by a barometric pressure recovery. River flooding events may have 'rapidly increasing water-levels' exceeding a threshold as an initiating primitive event and 'recovery to normal water-level' as a terminating event. A high-level event 'forest fire' may have an initiating low-level event 'spark' or 'rise in air temperature/smoke' followed by 'recovery to normal air temperature/smoke' as a terminating event. Similarly a high-level event 'traffic jam' may have an initiating low-level event such as a 'rise in exhaust'

or ‘stalled traffic’ and terminating events such as ‘reduction in exhaust/normal traffic speed’.

The composite event ontology has two key classes: primitive event and composite event. The class composite event has properties: hasInitiatingCondition, hasBodyCondition, hasTerminatingCondition. A first requirement is that the primitive events that initialize and terminate a composite event must be disjoint. More than one type of primitive event may initiate or terminate a composite event. The required ordering of primitive events in a composition are specified using Allen’s intervals (Allen J. F., 1984).

As an event, a composite event has start and end timestamps. These are determined by the timestamps of the initiating and terminating primitive events. The location of a composite event is more complex as it is composed from the spatial locations of the constituent primitive events.

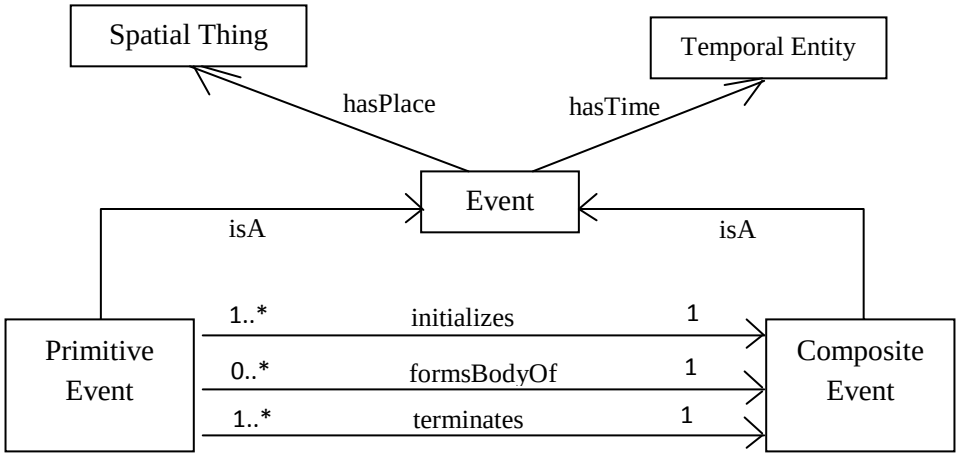


Figure 3.5 General composite event ontology

3.2.1 Composite Event Assembly

Composite event assembly starts with identifying the domain knowledge about the high-level composite event. This knowledge may be present in the *a priori* or *a posteriori* form. The conceptual flow diagram shown in Figure 3.6 presents the process of assembling composite events from primitive events using ontologies. The primitive event ontology (described in Section 3.1.2) defines a primitive event as an abstraction from a time series. The general composite event ontology provides the general structure by which primitive events are assembled to form composite events. Its specifies the primitive events that initiate, form the body of and terminate a composite event. This general composite ontology can then be specialized to a domain-specific high-level ontology. In this thesis, the Storm ontology is an example of a domain-specific, high-level composite event ontology. The storm event ontology specifies specific types of primitive events that initiate, form the body of and terminate a storm event.

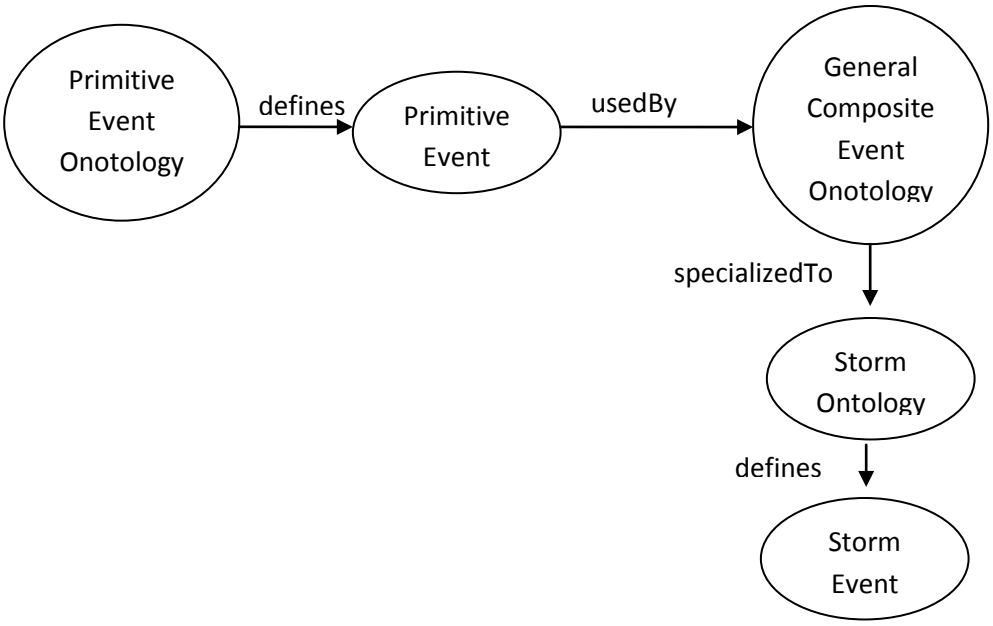


Figure 3.6 Conceptual flow for composite event assembly

The general sequence of steps for composite event assembly is described as follows:

1. Identify available *a priori* or *a posteriori* domain knowledge about the high-level composite event.

2. Identify marker parameters and conditions for a composite event. Identify temporal ordering for primitive events. This will enable us to identify *initializing* and *terminating* primitive events.

3. Use ontology to explicitly state relationship between initializing, body and terminating primitive events to form a composite event.

4. Algorithm for implementing assembly of primitive event into high-level composite event.

A marker parameter refers to a sensor measured parameter that signals the onset of a high-level event. An initiating primitive event is one obtained from the marker parameter time series that captures the relevant initiating behavior of the parameter (e.g. rapidly falling traffic speed/volume for a traffic jam event). An initiating primitive event set may consist of a single element for a case where there is a single primitive event at a single location that initiates the composite event. More often the initiating set will be a spatial set, a set of initiating primitive events from the set of observation or sensor locations. In general we would expect this initiating set to be temporally clustered or occurring within some short time lag of each other. Characteristics of a particular type of composite event will dictate the temporal pattern for the initiating set. We denote the initiating primitive event set as $PE_I^p[b_x,e_x]$ where p refers to one or more marker parameters and x indicates

the 1,…m locations and b_x, and e_x refer respectively to the begin time and end time for an initiating primitive event at location x.

The marker parameter that indicates the initiating primitive event set often serves as the marker parameter for the terminating primitive event set. In other words, the recovery of the marker parameter to a normal or steady state condition typically signals a termination of the composite event. As in the initiating case, the terminating primitive event set may consist of a single element or it can be a spatial set composed of the terminating primitive events from the set of observation locations. We denote the terminating primitive event set as $PE_T^p[b_x,e_x]$ where p refers to one or more terminating marker parameters and x indicates the 1,…m locations and b_x, and e_x refer respectively to the begin time and end time for a terminating primitive event at location x.

The temporal duration of a composite event is determined from the start and end times of the initiating and terminating primitive event sets. We denote composite event as CE[bc,ec] where bc and ec indicate begin and end times for the composite event and bc and ec are functions of the time intervals of the initiating and terminating primitive event sets:

$$bc=\min(PE_I[bx])$$

$$ec=\max(PE_T[ex])$$

Body primitive events are any set of primitive events on parameters of interest that occur during the composite event interval, CE[bc,ec].

A formal expression for the temporal ordering of primitive events as *initiating*, *body* and *terminating* subsets of events is given as follows:

$I[_{i,\ m}]$, $B[_{n,o}]$, $T[_{p,j}] \subseteq C[i,j]$:

$I[_{i,m}]$, $B[_{n,o}]$, $T[_{p,j}]$ }: I_{im} B_{no} B_{no} =

where,

I_{im} = Set of primitive event/s that *initiate* a composite event. Start and end time of the set is represented by 'i' and 'm' respectively.

B_{no} = Set of primitive event/s that form *body* of composite event. Start and end time of the set is represented by 'n' and 'o' respectively.

T_{pj} = Set of primitive event/s that *terminate* a composite event. Start and end time of the set is represented by 'p' and 'j' respectively.

P_k = Primitive event set with 'k' events

Allen's interval relationships are abstracted into temporal concepts, as described by Morchen (Morchen, Time Series Knowledge Mining, 2006). As discussed in Section 2.4, amongst the thirteen of Allen's interval relationships, only *before* and *overlaps* were implemented for storm detection in this thesis. Using these two temporal relationships, the ontology explicitly states the event-event relationship among the primitive events. As an example, given two events A and B with intervals [b1 e1] and [b2 e2] where, b1, b2 are start times and e1, e2 are end times of event A and B respectively:

A *before B* iif e1<=b2

A *overlaps* B iif b1 <= b2 and e1 > s2

The restriction on the length of gap between e1 and b2 is imposed by a threshold *gapTh* (detailed later in Step 6 of Algorithm 5.1).

3.3 Storm Event Ontology

This section describes specialization of the general composite event ontology for a specific type of high-level event, the Storm Event, as presented in Figure 3.7.

The main *competency questions* (Gruninger & Fox, 1995) that we aim to answer using this ontology are:

- What are the requirements for identifying storm event using time series data?
- What is the relationship between primitive and composite events in storm events?
- Does a storm have spatial, temporal or spatio-temporal aspects? How are these dimensions related to each other?

Related domain-level ontologies for the concept of storm events include SWEET (Semantic Web for Earth and Environmental Terminology) which comprehensively covers the earth and environmental domain. It contains a *Storm* class, which is a child of *Precipitation* class. Some interesting subclasses of class *Storm* are *HailStorm*, *IceStorm, LocalStorm, Monsoon, NortheastStorm, Squall, Thunderstorm* and *Tornado*. Interestingly, some sibling classes of class *Storm* were *Drizzle, FreezingRain, Hail, Mist, Rainfall, Sleet* and class *Snowfall.* Although the SWEET ontology is comprehensive in terms of stating the class hierarchies amongst the *MetereologicalPhenomenon* class, it does not serve our purpose of explicitly characterizing a storm by its primitive event parts.

The domain of the storm event ontology is restricted to detection of storm events using sensor generated time series which monitor atmospheric parameters. The key initiating and terminating primitive events are extracted from a *marker parameter*. For storm detection, barometric pressure is chosen as the marker parameter, because of its sensitivity to disturbance in the atmosphere and the relationship of low pressure dynamics to storm formation. The scope of this thesis is limited to implementing storm detection using barometric pressure as the single marker parameter. The primitive event 'significant barometric pressure fall' event i.e., BP_Fall initiates the Storm event, whereas the 'significant barometric pressure rise' event i.e. BP_Rise terminates it. The logical assumption in the ontology is that the primitive event that 'initializes' occurs before the primitive event that 'terminates' the composite event.

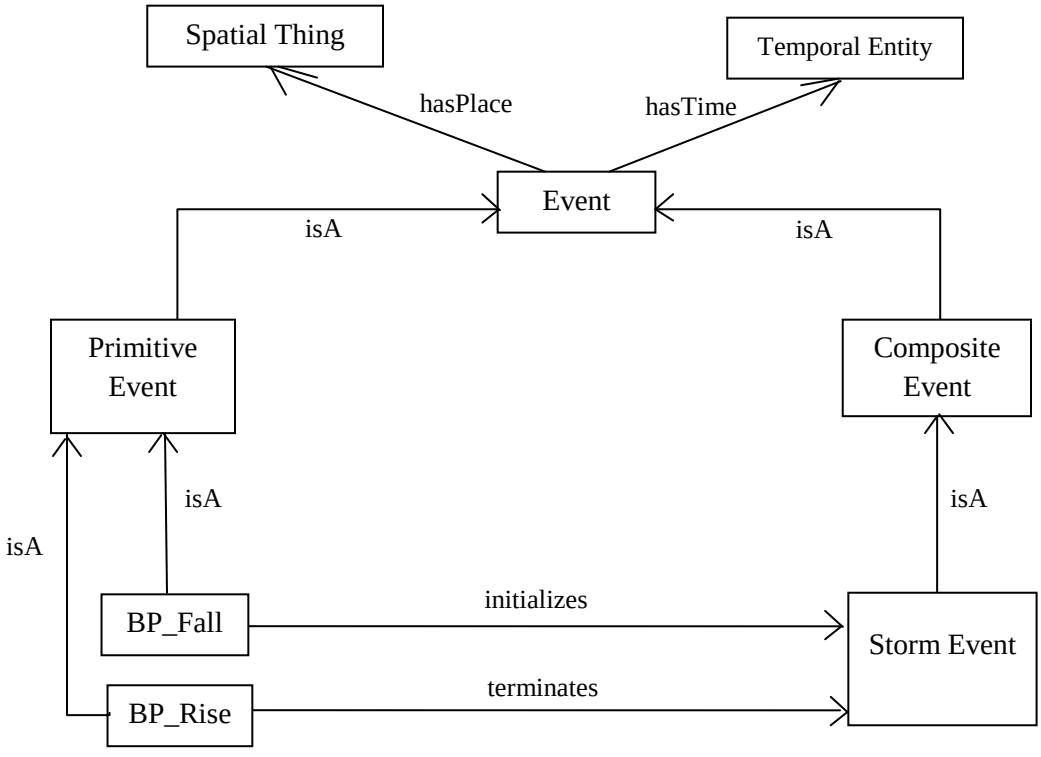


Figure 3.7 Storm ontology

3.4 Summary

This chapter presented the general process of data abstraction from time series to primitive events to composite events. The primitive event ontology specifies the concept of primitive event as used in this thesis and how primitive events are obtained from time series through AbstractionFunctions. Types of thresholds and ways to deal with uncertainty due to hard thresholds were discussed. A general composite event ontology for detecting a wide range of high-level events in several domains was presented followed by the storm event ontology specific to the domain of meteorology.

Chapter 4

PRIMITIVE EVENT DETECTION FROM GOMOOS TIME SERIES DATA

Based on the primitive event ontology and detection methodology as described in Chapter Three, this chapter describes the application of the primitive event detection methods to sensor time series data collected from the Gulf of Maine Ocean Observation System (GOMOOS). This chapter presents descriptions of the GOMOOS system, the time series datasets and several data quality issues. The selected parameters and the primitive event detection methods are geared toward capture of storms in the form of high-level composite event.

4.1 Gulf of Maine Ocean Observation System

The Gulf of Maine Ocean Observation System (GOMOOS) is a regional ocean observation system utilizing a network of buoys deployed in the Gulf of Maine (GOM) to obtain sustained, year-round and real-time observations of the ocean environment. The GOM covers approximately 94,000 km^2 in area and has ocean-depths ranging from 4 to 500 meters. The GOMOOS buoy system is capable of accommodating on the order of 100 surface and subsurface sensors. Sensors are deployed on moored buoys and each has a data logger. Figure 4.1 shows the extent of the Gulf of Maine and the location of the moored buoys. Each buoy carries sensors at multiple depths. The data logger collects, and on a regular schedule (usually hourly), transmits the data measurements via cellular telephone, iridium phone, or through NOAA's Geo-stationary Satellite Server (GOES)

satellite system. Data processing and preliminary quality control checks are performed by the Physical Oceanography Group (PHOG) at the University of Maine and data are then transferred to the GOMOOS website (GOMOOS, 2002; Wallinga, Pettigrew, & Irish, 2003).

4.2 Description of the GOMOOS Dataset

Sensor networks are deployed in the real world for measurement, detection and surveillance applications (Bonnet, Gehrke, & Seshadri, 2001). Sensors transform physical phenomena such as heat, light, sound, pressure, magnetism or motion into measurements using signal processing functions. Oceanic buoys carrying sensors form a fixed sensor network in GOM. This network generates data which are archived in a time series database. Each time series is associated with a sensor, location, parameter and depth. Table 4.1 shows a list of atmospheric, oceanographic and spatial parameters that sensors on each buoy record. The frequency of data collection i.e. temporal granularity varies by parameter but is similar (or processed to be uniform) for a parameter across all buoys. This thesis uses only atmospheric parameters: air temperature, wind speed, barometric pressure and wind direction which are collected by instruments placed at 3 m and 4 m above the sea surface. Other parameters such as wave height could be additional indicators of a storm event but were not used for this study.

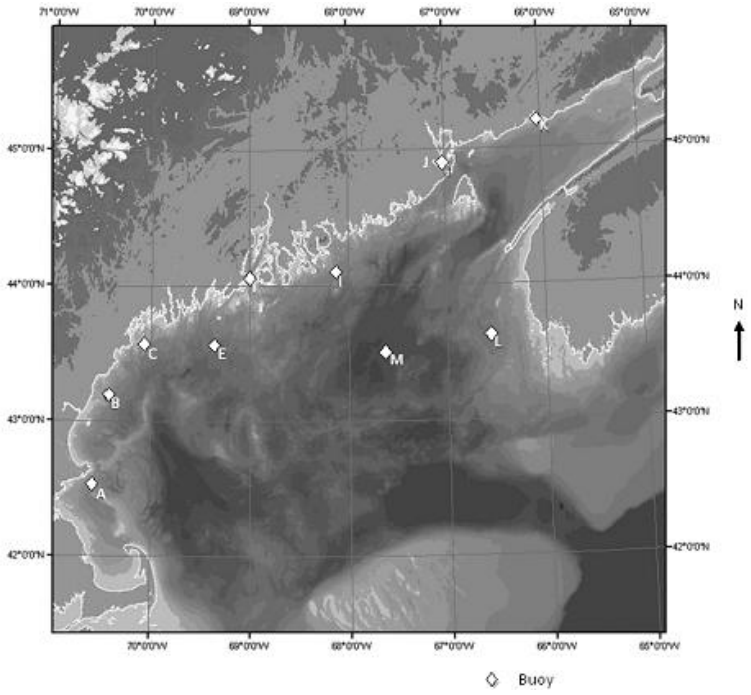


Figure 4.1 Position of buoys in the Gulf of Maine

Atmospheric parameters	Oceanic parameters	Spatial parameters
Air temperature, visibility, wind direction, wind speed, wind stress, wind gust, barometric pressure and pressure tendency	*Water:* Water temperatures, salinity, sigmaT, dissolved oxygen, percent oxygen, oxygen saturation, conductivity, density, transmissivity. *Current:* Current speed, current direction, *Waves:* Dominant wave period, sign wave.	*Buoy:* latitude, longitude, sensor depth location

Table 4.1: List of parameters measured by GOMOOS sensor network

4.2.1 Physical Data Architecture

Data collection methods and quality of the data are important to understand in preparing the data for processing and data mining. This section describes the process of data collection and storage in MATLAB structures, which facilitates event detection.

4.2.1.1 Data Collection

Data collected by sensors deployed on buoys are telemetered to a computer system maintained by PHOG at the University of Maine, Orono (Pettigrew, Roesler, Neville, & Deese, 2008). The computer system appends the incoming data to a file specific to the buoy that initiated the data transmission. These files are then parsed into constituent data streams according to the instruments that produced them. Processing algorithms are then run on raw data streams, and the data parameters are appended to time series NetCDF files, which are used to update the time series database. Figure 4.2 shows the overall process of data collection and processing.

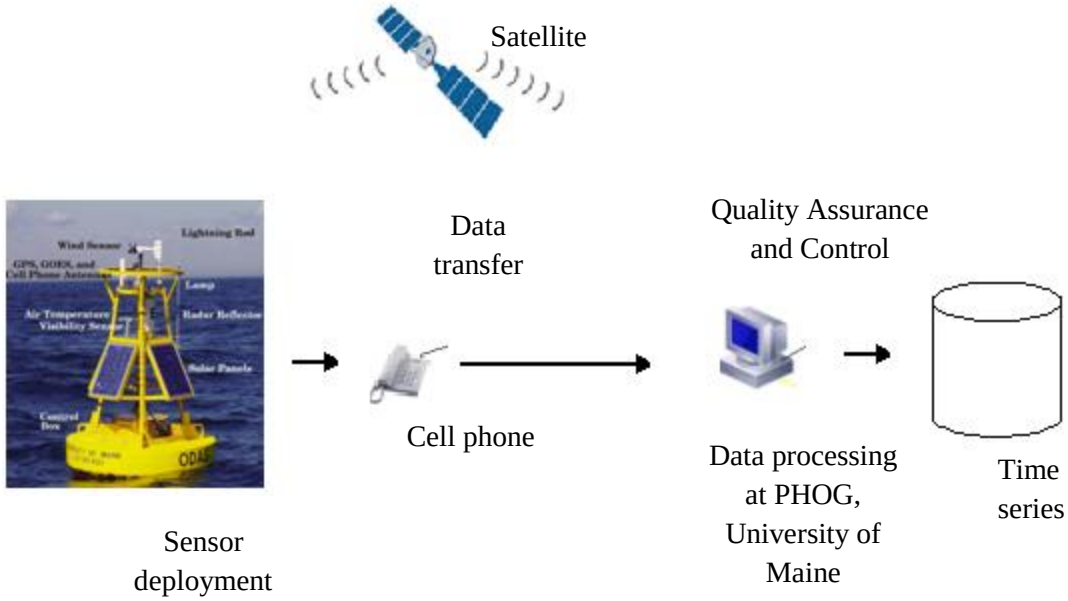


Figure 4.2 Data collection and pre-processing

The moored buoys are each referred to by a location/deployment code (e.g. A01, B01, C02). For this study, time series data were obtained for 10 buoys: A, B, C, E, F, I , J, L, M and N. Parameters selected for each buoy included barometric pressure, air temperature, wind speed and wind direction. Thus, there were 4 time series collected for each of ten locations for a total of 40 time series. For this study the full deployment time series was subset to cover 32 months between the years 2004 and 2007. The temporal granularity for each of the selected parameters is one hour. The time stamps on these time series were recorded initially as yyyy/mm/dd: hh in GMT and converted to Julian dates.

The units used for expressing values of parameter barometric pressure is millibars (mb), wind speed is meters per second (m/s), air temperature is degree Celsius and wind direction is degrees from North. Each of the parameters has an associated quality code

with values: 0b, 1b, 2b, 3b indicating respectively "quality_good, out_of_range, sensor_nonfunctional, questionable".

4.2.1.2 Data Structures in Matlab®

Data structures are a way of storing data to facilitate use and efficient processing. To maintain and manage the temporal indexing, the Matlab® programming language was used to create structure arrays for each time series. Matlab® structures store the time series by buoy location and sub-domain (i.e., *Air*, *Water*, *Current*, *Waves*), according to the following format:

buoy_subDomain {'field1', values1, 'field2', values2,…….}

For example,

a01_air {'jd', jd_array1, 'at', at_array, 'bp', 'bp_array',…}, a01air is a structure that can be visualized as shown in Figure 4.3.

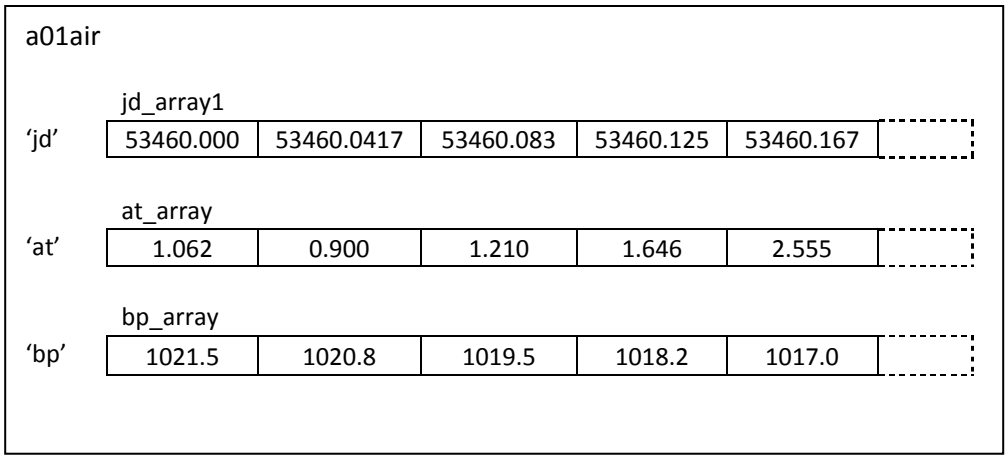


Figure 4.3 Visualization of a structure

The structure *a01air* stores time series belonging to buoy *A01* for the sub-domain *Air* which corresponds to the times series data on air temperature and barometric pressure collected by meteorological sensors. The time series are stored in one-dimensional arrays, the first representing Julian day values referenced by the field name *'jd'* and stored in *jd_array1*. The array *jd_array1* stores time stamps shared commonly by corresponding values in the array *at_array* for air temperature and the array *bp_array* for barometric pressure. Therefore, the lengths of arrays *jd_array1*, *at_array* and *bp_array* are same. Sensors measuring a parameter at the same location but different depths are stored by separate field names.

4.2.1.3 Primitive Event Detection Process Flow Diagram

Figure 4.4 shows the process flow diagram for detecting primitive events from sensor data. Matlab® scripts were used to extract crisp and fuzzy primitive events from Matlab® structures. Matlab® was chosen due to its computational strengths, ability to handle high-dimensional data and robust support for structures. Primitive events extracted from the time series can either be stored in the primitive events database or in simple Matlab® two-dimensional arrays. Primitive event files, stored as two-dimensional arrays, support further processing for composite event assembly using algorithms and through clustering and filtering. Matlab® structures are used for intermediate results (e.g. first difference, smoothed time series). The values stored in a structure can be numeric or symbolic.

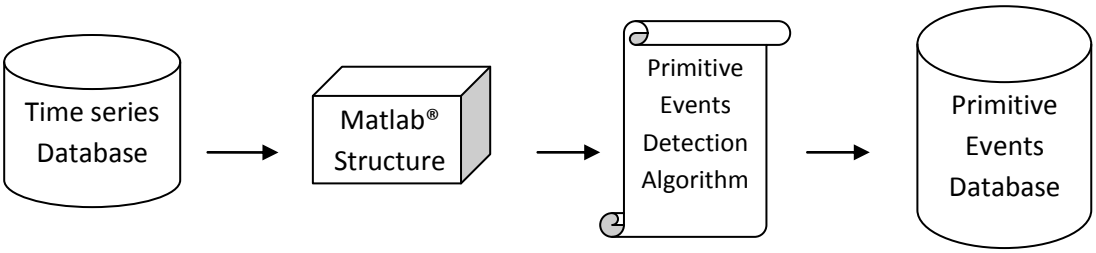


Figure 4.4 Process flow diagram for primitive event detection

4.2.2 Data Quality of GOMOOS Dataset

We evaluate the GOMOOS dataset for data quality as suggested by Pipino et al. (Pipino, Lee, & Wang, 2002). The GOMOOS dataset is accessible online for academic use under the Open Source license. Data is available from the year 2001 to present. As with any similar long term project, there have been practical and operational difficulties that need to be considered when using the data for knowledge discovery.

Missing values are present within the data during several intervals of time for various reasons, which range from faulty sensors, noise, or removal during preprocessing. Varying temporal granularities occur over the period of deployment either due to updating of sensor capabilities or removal of some buoys. Parameters are recorded at different temporal resolution depending upon the type of parameter. For example, wind direction is measured every 15 minute. We account for differences in temporal granularity in data by abstraction of time series with granularity less than an hour into one-hour granularity by averaging. One hour is the common coarsest level of granularity for most parameters. The data format and representation of parameters depends on parameter type, but is consistent across all buoys by parameter type. Every observation is

recorded with a consistent corresponding time stamp across all buoys. The data are generally assumed to be free of instrumental errors. Noise is removed during data preprocessing. Parameter values undergo range checks and the accuracy of the dataset post-processing is considered to be good and logically consistent. The relevancy of the collected GOMOOS data to the objective of storm discovery is high, as GOMOOS measures the following atmospheric parameters that are highly relevant to storm discovery: barometric pressure, wind speed, wind gust, wind direction, wave height, and current speed. Data collected is considered *timely* and *current*. GOMOOS dataset has much value since it provides unprecedented oceanic observations of the Gulf of Maine, enabling researchers to study ocean systems at fine temporal scales. The completeness of the dataset is evaluated in the next section on missing values.

4.2.2.1 Missing Data and Chosen Timeframe in GOMOOS Dataset

As mentioned in Section 3.1.5, missing data is a common problem in sensor surveillance systems. The performance of statistical analysis and inference depends largely on both the amount and pattern of missing data, which affects the quality of the resulting products. The number of missing values varies by parameter, sensor, buoy and is due to a number of reasons, including bad weather, sensor settings, service lags and so forth. The GOMOOS system collects data in a highly dynamic, high energy ocean environment, and as such, missing data in such a harsh environment is quite common. Moreover, not all buoys and sensors were deployed at the same time. To minimize missing values due to different buoy deployment periods, a common timeframe across all buoys and parameters was chosen for all further analysis and implementation of the EO approach. The common

timeframe is: 01-Oct-2004 22:00 to 04-Jul-2007 00:00 and will hence forth, be referred to as the ‘chosen time frame’. The chosen timeframe included observed data for ten buoys and the duration of the timeframe, i.e., 32 months, was considered sufficient for storm detection.

Figure 4.5 shows a bar chart comparing missing and non-missing observations at each buoy for each of the parameters: barometric pressure (BP), wind speed (WS), air temperature (AT) and wind direction (WD). It can be seen that the presence of missing observations varies by parameter. Parameter AT shows the least number of missing observations, followed by the storm marker parameter BP. The parameters WS and WD tend to have more missing observations, which may be due to various reasons ranging from limitations of the sensors or behavior of the parameter. Since the average percentage of missing observations for the marker parameter BP across all buoys was found to be 7.91%, we consider the marker parameter data quality to be good.

Figure 4.6 shows a comparative plot for buoy locations showing non-missing observations of parameters BP, AT, WS and WD. The x-axis shows time, whereas the y-axis shows buoy labels. Each buoy has four corresponding line-plots indicating the parameters in different colors: BP in blue, AT in green, WS in red, and WD in black color. Discontinuity in the line-plots indicates presence of missing data for that parameter. We observed that generally WS and WD sensors record missing observations at the same times. The high number of missing observations of wind speed and wind direction parameters, as seen in Figure 4.5, is attributed to service related issues and not due to faulty sensor observations.

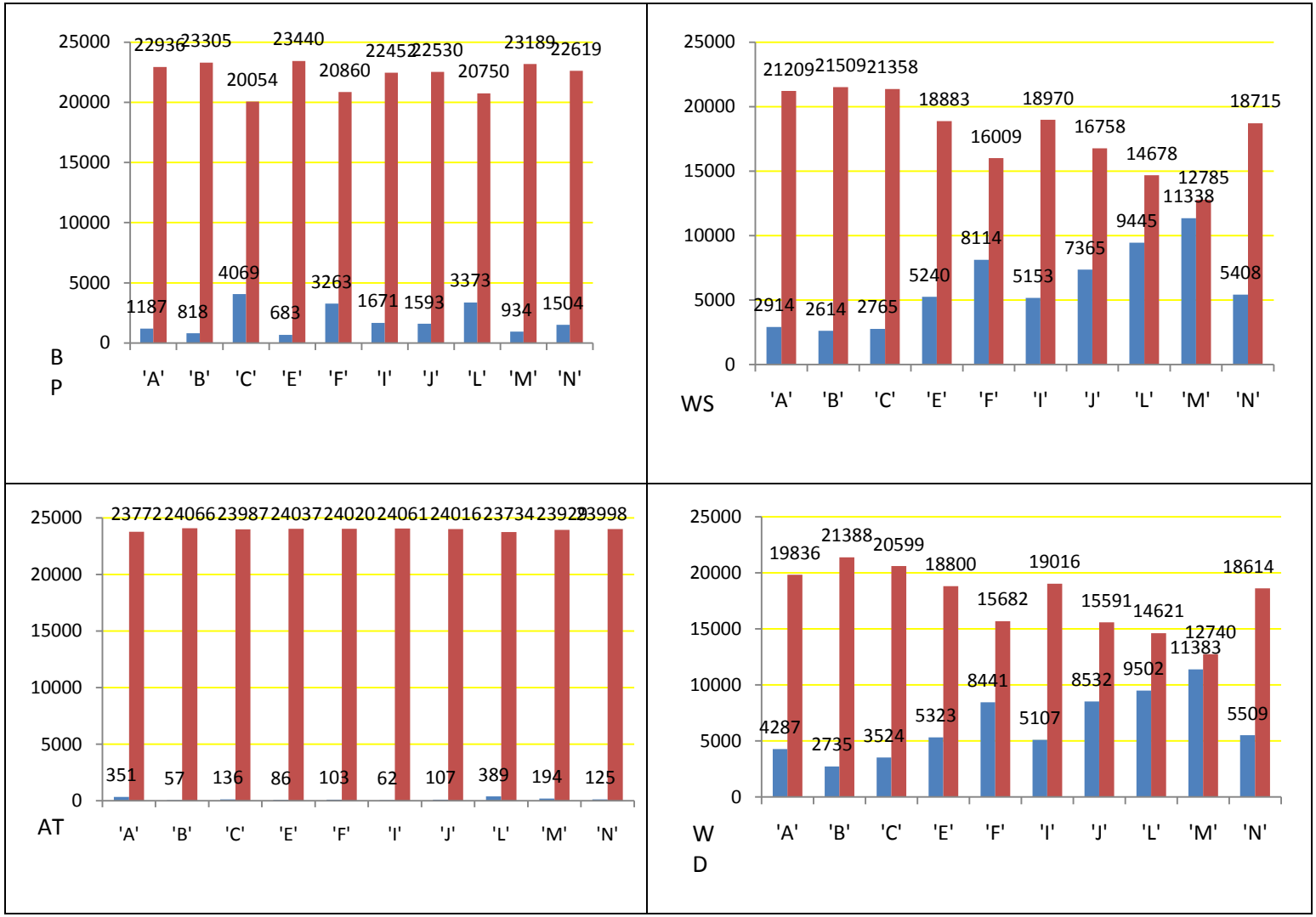


Figure 4.5 Comparison of missing and non-missing observations
for the chosen time frame [01-Oct-2004 22:00 to 04-Jul-2007 00:00]

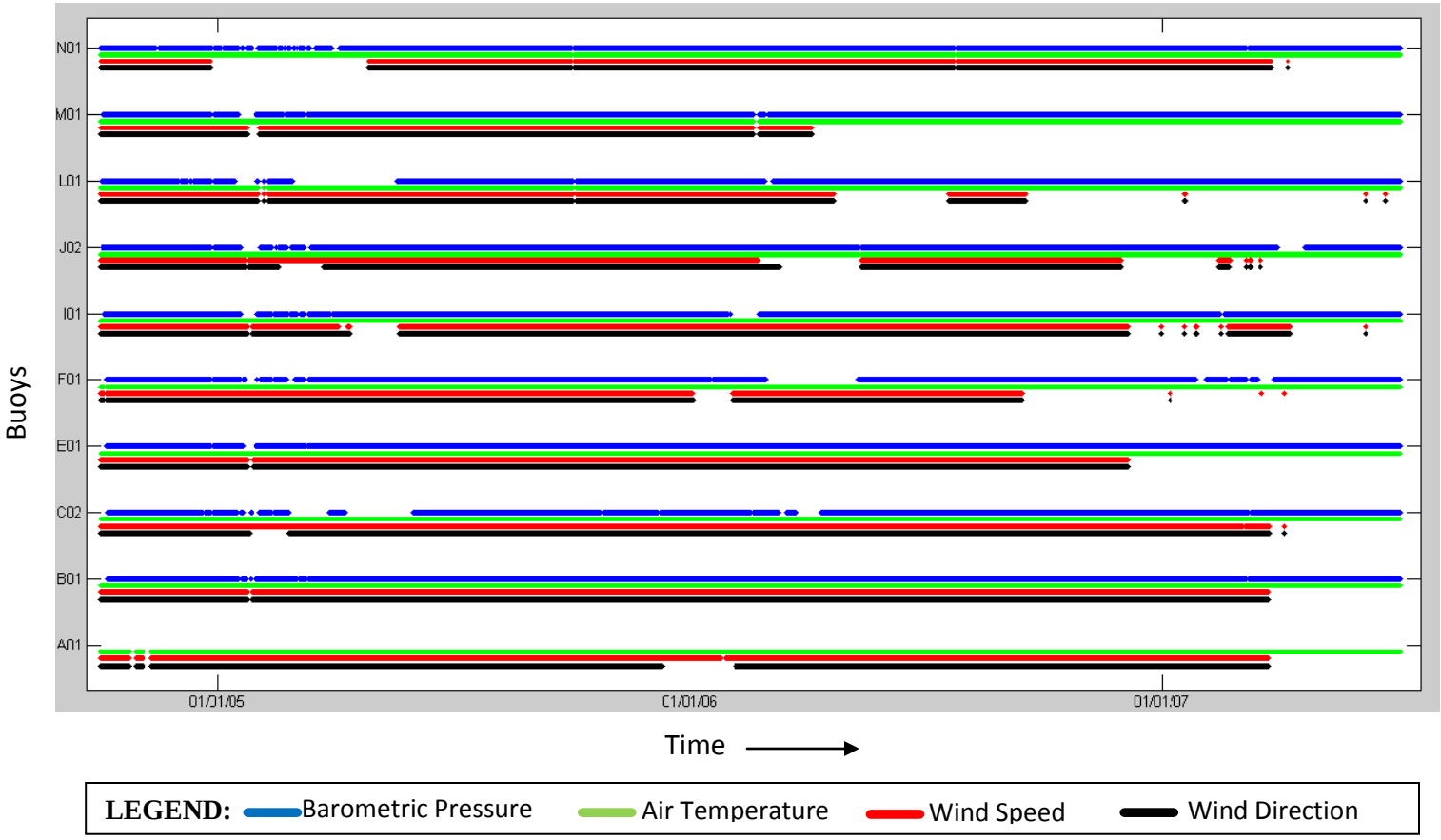


Figure 4.6 Comparative buoy data plot for parameters (discontinuity in lines indicates missing data)

4.3 Primitive Event Detection Method

In this thesis, primitive event detection only uses time series in Time-domain Continuous representation (See Section 2.3 for details). This representation requires minimal transformation from its time-stamped-values form. Primitive event detection using gradient thresholds requires a first difference transform, however this does not qualify as a Transformation-based representation because the first difference array still maintains time stamps and the time domain of the parent time series. The following sections discuss both the algorithms that implement primitive event detection from the time series data using various threshold types, and the results obtained.

4.3.1 Global Thresholds in the GOMOOS Dataset

Two methods were used to determine thresholds: statistical and user-defined. Statistically derived thresholds require much less decision-making on the part of the user. However, since thresholds are statistically derived, they are data dependant and sensitive to noise in the data, such as outliers. User-defined thresholds are data independent and may be determined with the assistance of data statistics. We combine statistically-derived and user-defined thresholds for primitive event extraction from the marker parameter BP in the ‘chosen timeframe’.

Table 4.2 presents the mean and standard deviations for the parameters barometric pressure, wind speed, wind gust, air temperature and wind direction. The term ‘global average’ refers to an average of the time series for all available buoys. For the line threshold abstraction function applied to barometric pressure, we use a threshold of 1015

mb units (half standard deviation below the mean, as shown in Table 4.2). There are seasonal differences in the means of these parameters which need to be taken into consideration in setting thresholds in some cases of event detection.

Figure 4.7 illustrates a primitive event detected using a combination of line state and gradient thresholds. The abstracted primitive events are referred to as ‘BP_Fall’ events and are stored in the primitive events database as a record as shown in Figure 4.8, along with the metadata.

	Statistic	A01	B01	C02	D01	E01	F01	I01	J02	K01	L01	M01	N01	Global Average
BP	Mean	1014.6	1014.5	1014.6	--	1013.9	1014.0	1014.3	1014.3	--	1013.8	1014	1014.5	1014.21
	Std. Dev.	8.7	8.8	8.9	--	9.1	9.1	9.1	9.3	--	9.0	9.16	8.8	9.02
WS	Mean	5.85	5.64	5.48	3.8	5.64	5.11	5.75	4.5	4.95	6.3	6.65	6.33	5.5
	Std. Dev.	3.10	3.10	3.1	2.3	3.3	3.04	3.49	2.9	3.13	3.65	3.5	3.37	3.16
WG	Mean	7.32	7.07	6.84	5.16	7.04	6.42	7.18	6.16	6.61	7.74	8.2	7.94	6.97
	Std. Dev.	3.85	3.81	3.74	3.19	4.04	3.68	4.33	3.82	3.92	4.46	4.37	4.19	3.95
AT	Mean	9.29	8.28	7.93	4.98	7.79	6.97	6.59	6.28	6.48	6.68	8.4	8.4	7.34
	Std. Dev.	7.66	7.78	7.86	7.82	7.25	7.61	6.24	7.59	7.69	5.87	7.01	6.22	7.22

Table 4.2 Statistics on atmospheric parameters across all buoys for calculation of global averages

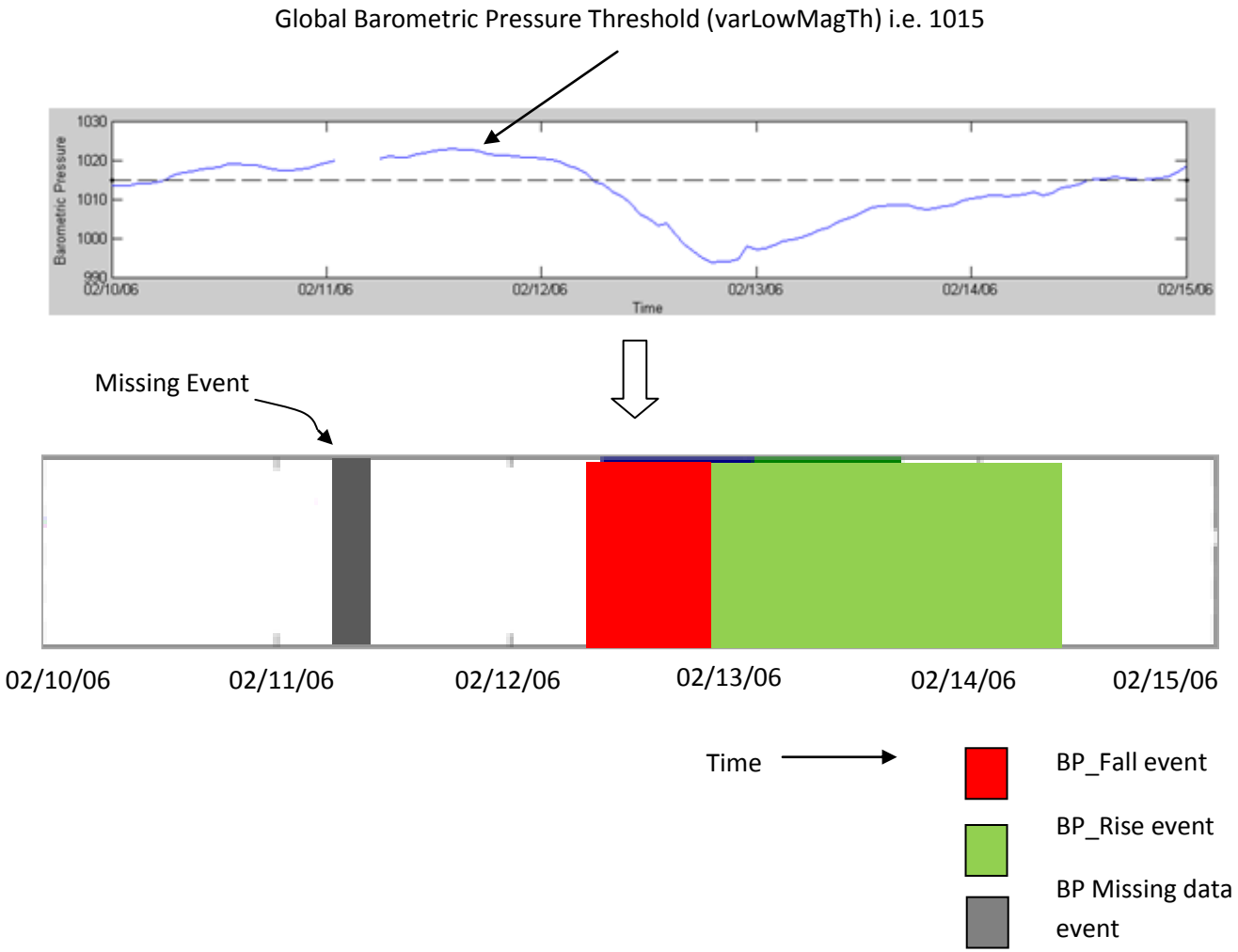


Figure 4.7 Visualization of primitive event detection using combined line and gradient threshold

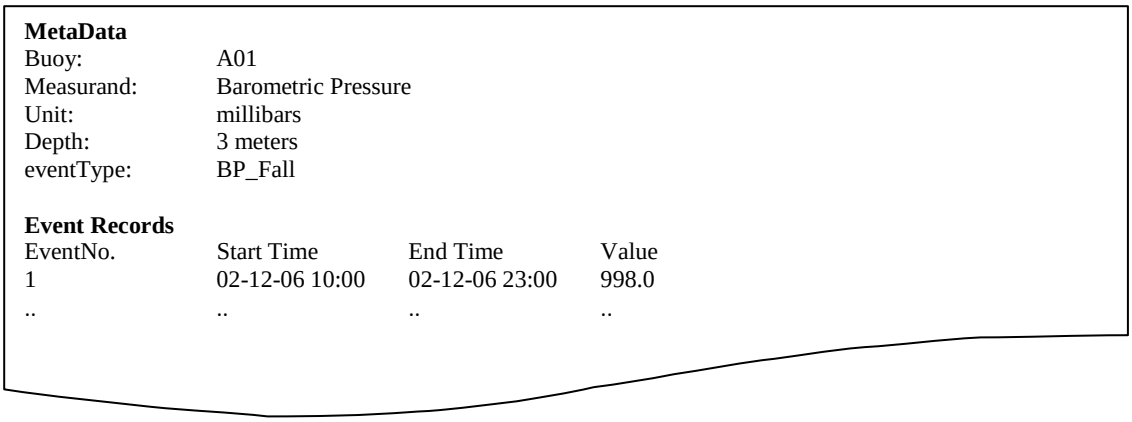


Figure 4.8 Primitive event data stored in a file along with metadata

Events	Transform	Temporal Filter	Thresholds		
			State	Gradient	Rate
'Significant BP fall' primitive event	First difference	Min event duration 6 hour; Min events disjoin 3 hours	1015	fall	4 mb
'Significant BP rise' primitive event	First difference	Min event duration 6 hour; Min events disjoin 3 hours	1015	rise	3 mb

Table 4.3 Threshold criteria for primitive event detection in GOMOOS

4.3.2 Primitive Event Detection Algorithm

This section presents the implementation of the primitive event detection approach discussed in Section 3.1. As stated in the storm event ontology (see Section 3.3), a storm

event is initiated by the ‘significant BP fall’ and terminated by the ‘significant BP rise’ primitive events. These primitive events are described next.

The ‘significant BP fall and BP rise’ primitive events are abstracted subsequences of time series which satisfy state, gradient, rate and temporal thresholds. The rate and temporal thresholds used in detecting primitive events vary. Table 4.3 summaries thresholds and filters used for barometric pressure primitive events. The rate threshold for a ‘significant BP fall’ is 4 mb and 3 mb over the primitive event duration, to qualify as a ‘significant BP rise’ primitive event. The minimum temporal filters as indicated in Table 4.3 are 6 hours for ‘significant BP fall’ and 3 hours for ‘significant BP rise’. This is because recovery in BP tends to occur at a lower rate as compared to the fall. These temporal threshold values were determined from visual evaluation of the time series data. The least temporal distance between two storm candidates was set to 30 hours.

The primitive event detection algorithm processes data in four steps. Inputs to the algorithm (see Figure 4.9) are time series in the form of a one-dimensional time stamped array called varArray. varArray corresponds to the array described in section 4.2.1.2 (e.g. air01). The thresholds include a value threshold (varLowMagTh), rate threshold (rateTh), and temporal filters (minDurTh and minEvent_Event_Dur) examples of which appear in Table 4.3. The varLowMagTh represents the threshold value below which a parameter value is considered to be ‘low’. The threshold rateTh represents the overall gradient, that is, the fall or rise within the total duration of the primitive event. This is calculated as the total range difference within the primitive event. The temporal filter minDurTh represents the minimum duration of a primitive event. A primitive event with duration less than this threshold is considered too short to qualify. For example, if the value of the threshold

minDurTh was 3 hours, then a primitive event qualifying on all other thresholds but lasting only two hours would be ignored. The second temporal filter, minEvent_Event_Dur, represents the minimum temporal gap between two qualifying primitive events of the same type that determines whether they are considered as one or as separate primitive events. If two qualifying primitive events have a temporal gap less than this threshold, they are considered as one primitive event.

The expected output of the algorithm is a two dimensional array with start and end times (or corresponding indexes) for intervals in which parameter values meet the thresholds specified in the inputs.

The first step of the algorithm loops though the varArray to check if the gradient condition is met using first difference of adjacent values. If the condition is met, the index (time stamp) is stored in a new array (named indexGradient). The second step applies the state threshold (1015 mb for BP state threshold, as noted in Figure 4.7) condition on the indexGradient array to create another new array of qualifying indexes (named indexGradient_LowMagTh). The third step loops through indexGradient_LowMagTh and picks start and end points for candidate primitive events meeting temporal separation constraints. In the last step, the algorithm applies the temporal filters minDurTh on the candidate primitive events detected in step 3 to generate the final set of primitive events as the output result.

Results of the implementation of this algorithm for detecting ‘significant BP fall and rise’ events using thresholds as specified in Table 4.3 are presented in the next section.

```
Input:   varArray, varLowMagTh, rateTh, minEvent_Event_Dur, minDurTh

Output: eventMatrix (i.e, BP_Fall if gradientFeature is Fall; BP_Rise
          if gradientFeature is Rise)
```

```
Step 1) Extract indexes with desired 'gradientFeature' in varArray
indexGradient←empty
firstdiffArray← first difference transform (varArray.bp)
FOR each j in firstdiffArray
        IF firstdiffArray(j) > 'fallgradient' THEN
//Comment: where, fallgradient ←0 to find negative firstdiffArray values
//for 'fall' and positive //firstdiffArray values for 'rise'
                Add j to indexGradient
        END IF
END LOOP

Step 2) Apply line threshold on value of parameter i.e., varLowMagTh
indexGradient_LowMagTh←empty
FOR each j in indexGradient
        IF varArray.bp(j) <= varLowMagTh THEN
                Add j to indexGradient_LowMagTh
        END IF
END LOOP

Step 3) Detect primitive events from indexGradient_LowMagTh
//Comment: Detect Start Points
EventCount←1
EventStart←empty
Add indexGradient_LowMagTh(1) to EventStart
FOR each k in indexGradient_LowMagTh
        IF indexGradient_LowMagTh(k+1)-indexGradient_LowMagTh(k)>
minEvent_Event_Dur THEN

                EventCount ← EventCount + 1
                Add indexGradient_LowMagTh(k+1) to EventStart
        END IF
END LOOP
//Comment: Detect End Points
EventCount←0
EventEnd←empty
FOR each k in indexGradient_LowMagTh
        IF indexGradient_LowMagTh(k+1)-indexGradient_LowMagTh(k)>
minEvent_Event_Dur THEN          EventCount←EventCount + 1
```

Figure 4.9 Pseudo code for gradient type primitive event detection

Figure 4.9 continued

```
Add indexGradient_LowMagTh(k) to EventEnd
        END IF
END LOOP

Step 4) Filter out events less than duration 'minDurTh' and rate less than
        'rateTh' units
//Comment: Check for length (eventStart) = length (eventEnd)
countMatxIndex←0
startTime←empty
endTime←empty

FOR i 1 to length (eventStart)
        IF eventEnd(i) – eventStart (i) > minDurTh THEN
                qualifyIndex= eventStart(i): eventEnd(i)
                IF RANGE (varArray(qualifyIndex) >= rateTh THEN
                        countMatxIndex ←countMatxIndex + 1
                        startTime(countMatxIndex) ←eventStart(i)
                        endTime(countMatxIndex) ←eventEnd(i)
                END IF
        END IF
END LOOP
VerticalConcatenate startTime, endTime INTO ARRAY eventMatrix
RETURN eventMatrix
```

4.4 Results of Primitive Event Detection

Table 4.4 summarizes the results of the primitive event detection using the algorithm and thresholds as indicated in Figure 4.9 and Table 4.3 for each event-type.

A comparison of the detected primitive event numbers with the plot in Figure 4.5 showing missing and non-missing observations indicates that for parameter BP, primitive events detected at a buoy depend on the number of non-missing observations at that buoy. C02 has the most number of missing observations and also the least number of 'significant BP fall or rise' events in comparison to all other buoys. Buoy E01 detected

the highest number of ‘significant BP fall’ and ‘significant BP rise’ events and also had the least number of missing observations. Therefore, it appears that the number of BP primitive events detected is closely tied to data quality of the BP parameter. Notice that the numbers of ‘significant BP rise’ events are higher than ‘significant BP fall’. This is most likely due to the difference in threshold values.

Buoy N01 has the least number of cold spike events in air temperature. However, this is not due to missing data. One possibility for the low number of cold spike primitive events may be due to the spatial location of the buoy N01 in the farthest South-East or its location as farthest out in the sea (see Figure 4.1). However, the mean for AT observations at buoy N01 is not significantly different from other buoys such as L01 and M01.

Another interesting finding is that the number of high wind speed primitive events is highest for J02, closely followed by F01. This is not due to longer data records (low number of missing values) at these buoys. The data statistics in Table 4.2 do not indicate higher mean WS values in buoys J02 and F01 in comparison to other buoys. The high number of primitive events therefore is not attributable to any clear reason.

Higher numbers of ‘sustained NE wind direction’ primitive events were found at Buoys F01, B01, I01 and A01 respectively. Interestingly, buoy F01 has more missing WS observations compared to other buoy locations. Buoy B01 does not have many missing observations. Buoys A01 and I01 have about the same number of missing observations. It appears that the number of primitive events of event type ‘sustained wind direction’ does not show much correlation with the number of missing observations.

Parameter\Buoy	A01	B01	C02	E01	F01	I01	J02	L01	M01	N01
Number of 'Significant Barometric Pressure- Fall' events	162	165	151	173	164	168	169	158	171	163
Number of 'Significant Barometric Pressure- Rise' events	199	207	176	216	196	208	204	188	217	201
Number of 'Air Temperature - Cold Spike (below -4°C)' events	104	134	139	141	211	155	205	80	97	26
Number of 'High Wind Speed (above 10 m/s)' events	55	52	90	72	139	83	192	29	20	17
Number of 'Sustained Wind Direction (North-East i.e., 0-100 degree from magnetic north)' events	567	623	511	539	737	563	486	263	334	197

Table 4.4 Results of primitive events detected

In conclusion, the number of primitive events detected for the marker parameter BP shows good correlation to the number of non-missing observations. The number of primitive events, 'Cold Spike in AT', 'High WS' and 'sustained NE wind direction', did not show correlation with the presence of non-missing values. There could be seasonal or other reasons associated with the numbers of primitive events that were found in these parameters. However, an analysis of the discrepancy is beyond the scope of this thesis.

4.5 Summary

This chapter presented primitive event detection using the GOMOOS moored buoy time series for meteorological parameters. First, information on the GOMOOS system and data limitations was presented. Description of the data and data structures for event detection was presented, followed by data quality considerations taken into account. Further, this chapter presented the algorithm for primitive event detection along with the reasoning behind the choice of particular threshold values. Lastly, results of primitive event detection were presented and discussed in Section 4.4.

The next chapter presents implementation of composite event assembly for discovering storm events using primitive events in the GOMOOS dataset.

Chapter 5

STORM COMPOSITE EVENT ASSEMBLY FROM GOMOOS DATA PRIMITIVE EVENTS

This chapter describes composite event assembly from primitive events. General methods of composite event assembly were mentioned in Chapter Three. An algorithm for assembly of candidate storms by integrating primitive events from the GOMOOS dataset is presented here. The chapter concludes with a validation of candidate storms by comparison to an independent data source, in this case the NCDC storm events database.

5.1 Composite Event Assembly

Primitive events are the building blocks which are assembled to form a composite event. Detection of primitive events from time series was presented in Section 3.1. The key to high-level composite event assembly is the discovery of initiating and terminating conditions, as specified in the composite event ontology. The Storm Event ontology, presented in Section 3.3, specifies the initiating and terminating primitive events for the high-level composite storm event to be changes in the marker parameter, barometric pressure. Allen's temporal relationships *before* and *overlaps* were used to order initiating and terminating primitive event sets for candidate storm events. The algorithm implementing the process of composite storm event assembly from primitive events is discussed next.

5.2 Algorithm for Composite Event Assembly

Wireless sensor networks contain many nodes and GOMOOS, while not a wireless sensor network, has several deployed buoy locations where time series are collected on various parameters. The detected primitive events indicate types of change in a parameter as observed in time series from individual locations (buoy or node). The assembly of a composite event needs to consider the spatial arrangement and temporal order of initiating primitive event detected at these locations. For the example, in high-level storm events, we would expect the initiating primitive events to occur nearly simultaneously or in some spatio-temporal order across the buoy locations. Thus the first step in the assembly is to use spatio-temporal clustering on primitive events. Primitive events are spatio-temporally clustered using a string construct called a *Spatial Progression String* (SPS). The SPS is a string of time-stamped, comma-separated substrings made up of symbols representing the location and some qualitative property of the primitive event (typically the AbstractionType). The SPS construct is domain-independent and could be applied to the assembly of other types of high-level events in sensor network settings. The SPS construct works well with data which is spatially sparse but temporally dense. In sensor networks, if the sensor node locations are more than 26 (in case of letters of an alphabet representing sensor node location) the SPS construct will become complex and therefore difficult to manage.

In constructing the storm SPS for the GOMOOS case-study, the string symbols are letters of the alphabet representing the buoy location at which the primitive event was observed and the gradient abstraction type of that primitive event. Since GOMOOS names buoys by letters A01, B01, C02 etc., these letters were adopted. Gradient abstraction type ‘fall’

is represented by an upper case buoy letter, whereas 'rise' is represented by the lower case buoy letter. For example, a 'fall' primitive event extracted from buoy A01 would be represented in SPS as the letter 'A', whereas a rise primitive event from the same location is represented as 'a'. These letters are called *BuoyTags*, and in this context but could be considered location tags in a more general sensor network setting. The *BuoyTags* are symbols that represent both a gradient primitive event and a buoy location at the particular time of observation. Generation of the SPS facilitates spatio-temporal clustering of primitive events and serves as an intermediate step to identifying candidate storms. Once the candidate storms are identified, they can be further classified using the SPS. This process of storm event detection is presented next.

As specified in the Storm Event (SE) ontology in Section 3.3, the primitive event type '*Significant barometric pressure fall*' initiates a candidate storm event and the primitive event type '*Significant barometric pressure rise*' terminates a candidate storm event. These primitive events are extracted as described in Section 4.3. For storm detection, these two primitive event types are loaded into two-dimensional date-string arrays, *BP_Fall* and *BP_Rise* respectively. These two arrays then serve as input to an algorithm that creates the SPS, as outlined in Figure 5.1. The format of the primitive event arrays *BP_Fall* and *BP_Rise* is: [Starttime, Endtime, BuoyTag], where 'Starttime' and 'Endtime' are time stamps when the primitive event starts and ends. The example array tuple, [01-15-2004 00:00, 02-16-2004 00:00, A] indicates a 'fall' primitive event starting at 01-15-2004 00:00 and ending at 02-16-2004 00:00, if month day shouldn't second number be 01-16-2004 observed at buoy *A01*. Alternatively, a *BuoyTag* of c in the example would indicate a 'rise' in the parameter, indicated by the lower case, with the

letter c indicating buoy location C02. A tuple from either the *BP_Fall* or *BP_Rise* array (e.g., [01-15-2004 00:00, 02-16-2004 00:00, A]) thus captures the primitive event type, time of occurrence, and location.

To identify composite events spread over several locations but clustered in time, we create the SPS from these arrays which contain the spatio-temporal information for a candidate storm. A candidate storm appears as a temporal cluster in the SPS and these clusters contain the spatio-temporal information for a candidate storm.

Figure 5.1 illustrates the generation of SPS from the BP_Fall and BP_Rise arrays. The two matrices are concatenated vertically to form a one-dimensional array of SPS values which are the appended set of *BuoyTags* ordered by temporal indexes. A temporal cluster in the resulting SPS indicates the gradient of the barometric pressure parameter over all available locations, and as each BuoyTag has a unique time stamp, their combination equals the total duration of the candidate storm.

As illustrated in Figure 5.1, the start and end indexes of temporal clusters in the SPS indicate the start and end timestamps of storm candidates. The SPS cells corresponding to indexes 1 and 2 are empty. Index 3 which contains a BuoyTag (in this case indicating an initiating primitive event) signals the start of the temporal cluster. So index 3 is chosen as the start index. Similarly index 14 contains the last BuoyTag (last meaning it is followed by a blank cell at index 15). Thus, index 14 is chosen as the end index for this example temporal cluster in the SPS cell array.

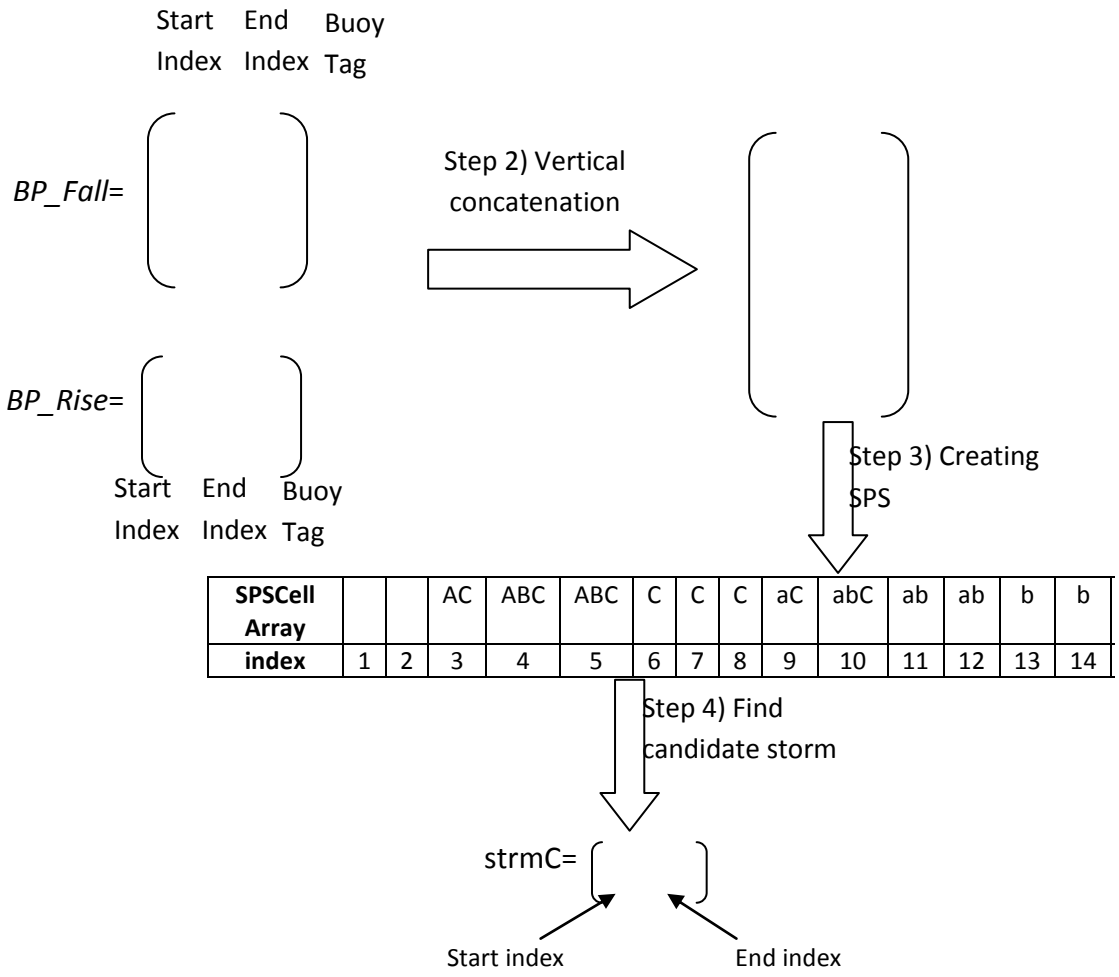


Figure 5.1 Visual illustration of SPS formation and identification of candidate storms

A flowchart explaining the algorithm used to create the SPS is presented in Figure 5.2. The inputs to the algorithm are *BP_Fall* and *BP_Rise* arrays which are vertically concatenated to form a new matrix, BP_Matx. The one dimensional time stamped cell array 'SPS' is initialized to the length of the time period of the *chosen timeframe* (i.e., [01-Oct-2004 22:00 to 04-Jul-2007 00:00]) in hours. A loop is run over the total duration of

the time period to check and append *BuoyTags* to the corresponding SPS cell strings. The output of the algorithm in Figure 5.2 is the SPS cell array.

INPUTS: Arrays of primitive initiating and terminating events: BP_Fall, BP_Rise, tfs=[01-Oct-2004 22:00] – temporal index for start of the chosen timeframe, tfe=[04-Jul-2007 00:00]- temporal index for end of the chosen time frame
OUTPUT: SPS in MATLAB cell array format

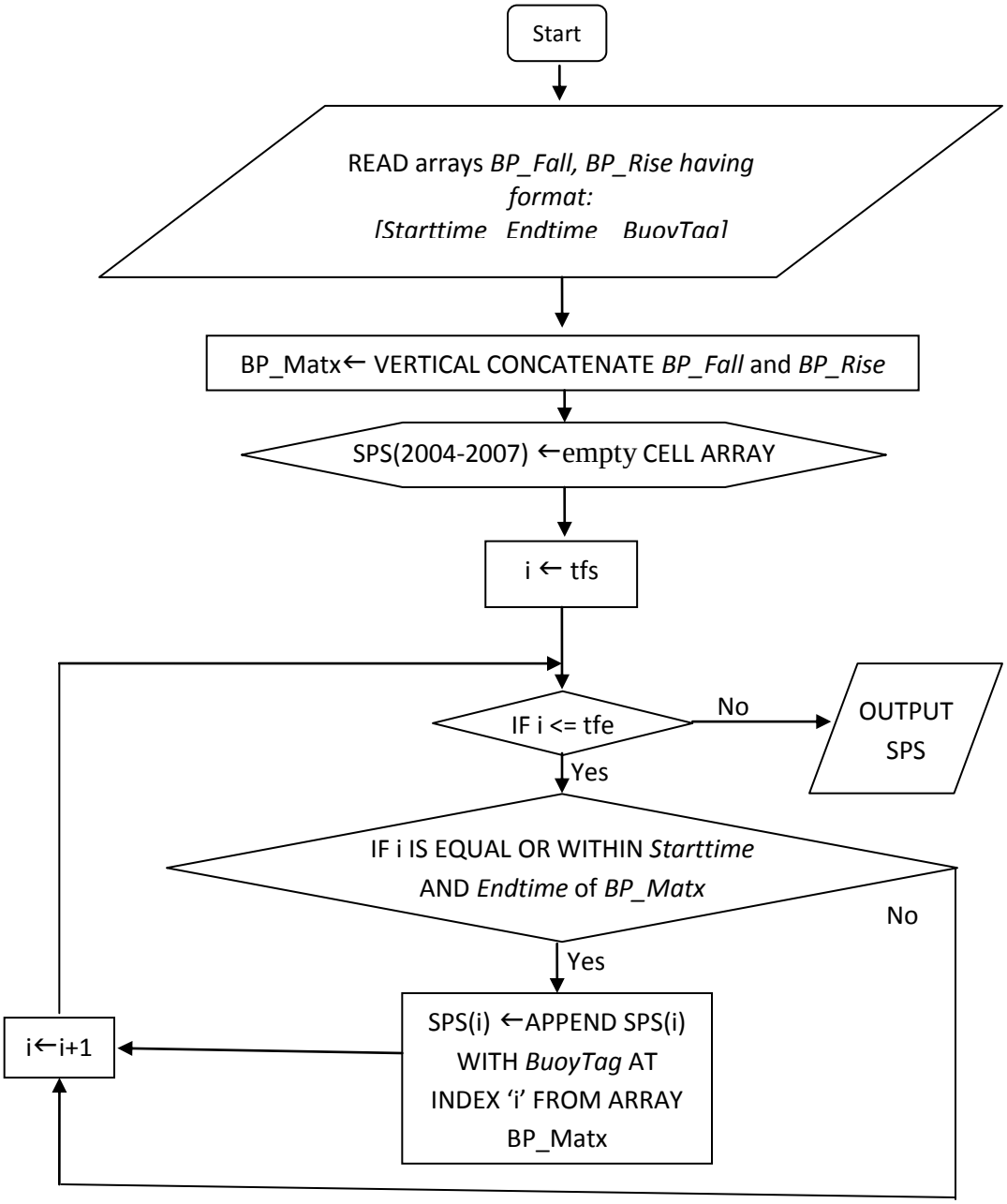


Figure 5.2 Flow chart for constructing SPS

Further, identification of the candidate storm's start and end times is carried out using the SPS created as an output of the algorithm presented in Figure 5.2. Initially, two arrays, strmC_St and strmC_End, are initialized as empty arrays of length one. A loop is created to check for empty and non-empty cells in SPS that identify the temporal clusters and coincidently the start and end times of candidate storms. The output of this algorithm as illustrated in Figure 5.3 is a strmC_Matx, which is a two-dimensional array storing start, end, and storm_SPS (the subset of SPS corresponding to the candidate storm) for each storm candidate.

Preliminary classification of candidate storms was done based on the presence of BP primitive events within the candidate storms. The first set of storms called *setFR* contains both BP_fall and BP_rise primitive events which is the ideal case as it matches the expected conditions for a candidate storm as specified by the ontology. The temporal clusters however may contain other cases in which only BP_fall primitive events are present or only BP_rise primitive events are present. Thus a second set of candidate storms denoted by *setFO* contain only BP_fall primitive events. The third set of candidate storms, denoted by *setRO*, contain only BP_rise primitive events. The classification algorithm is displayed as a flowchart in Figure 5.4. It uses a nested for loop to identify the presence of rising, falling or both types of primitive events within the storm candidate. Figure 5.4 uses the symbols '‡' and '⁋' to establish continuity between the algorithm flow charts on separate pages. The outermost for loop runs from 1 to the total number of candidate storms, which equals the number of rows in *strmC_Matx* in our arrays. The inner two for loops run for the number of rows in *BP_Rise* and *BP_Fall*,

indicating the number of primitive events. The results of these algorithms are presented in the next section.

INPUT: SPS in MATALB cell array format, tfs=[01-Oct-2004 22:00] – temporal index for start of the chosen timeframe, tfe=[04-Jul-2007 00:00]- temporal index for end of the chosen time frame
OUTPUT: Candidate storms stored as a two-dimensional array 'strmC_Matx'

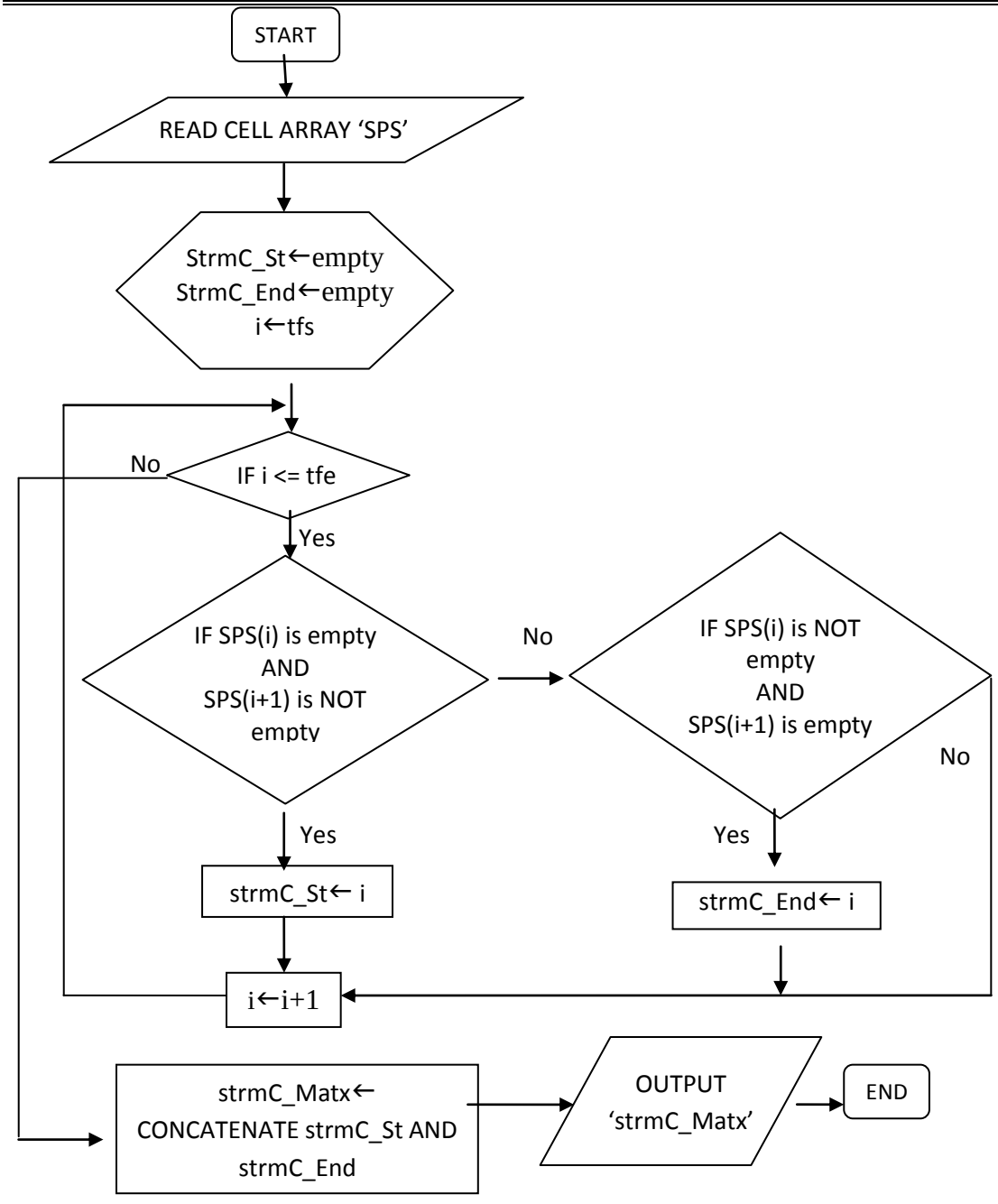


Figure 5.3 Flow chart for identifying candidate storms from SPS

INPUT: Candidate storms stored as a two-dimensional array ‘strmC_Matx’

OUTPUT: Three classes of candidate storms containing BP_fall followed by BP_rise as ‘setFR’, containing fall only as ‘setFO’, and containing rise only as ‘setRO’

Sort candidate storms into sets: *setFR* (containing *BP_Fall* followed by *BP_Rise*), setFO (containing only *BP_Fall* events) and *setRO* (containing only *BP_Rise* events)

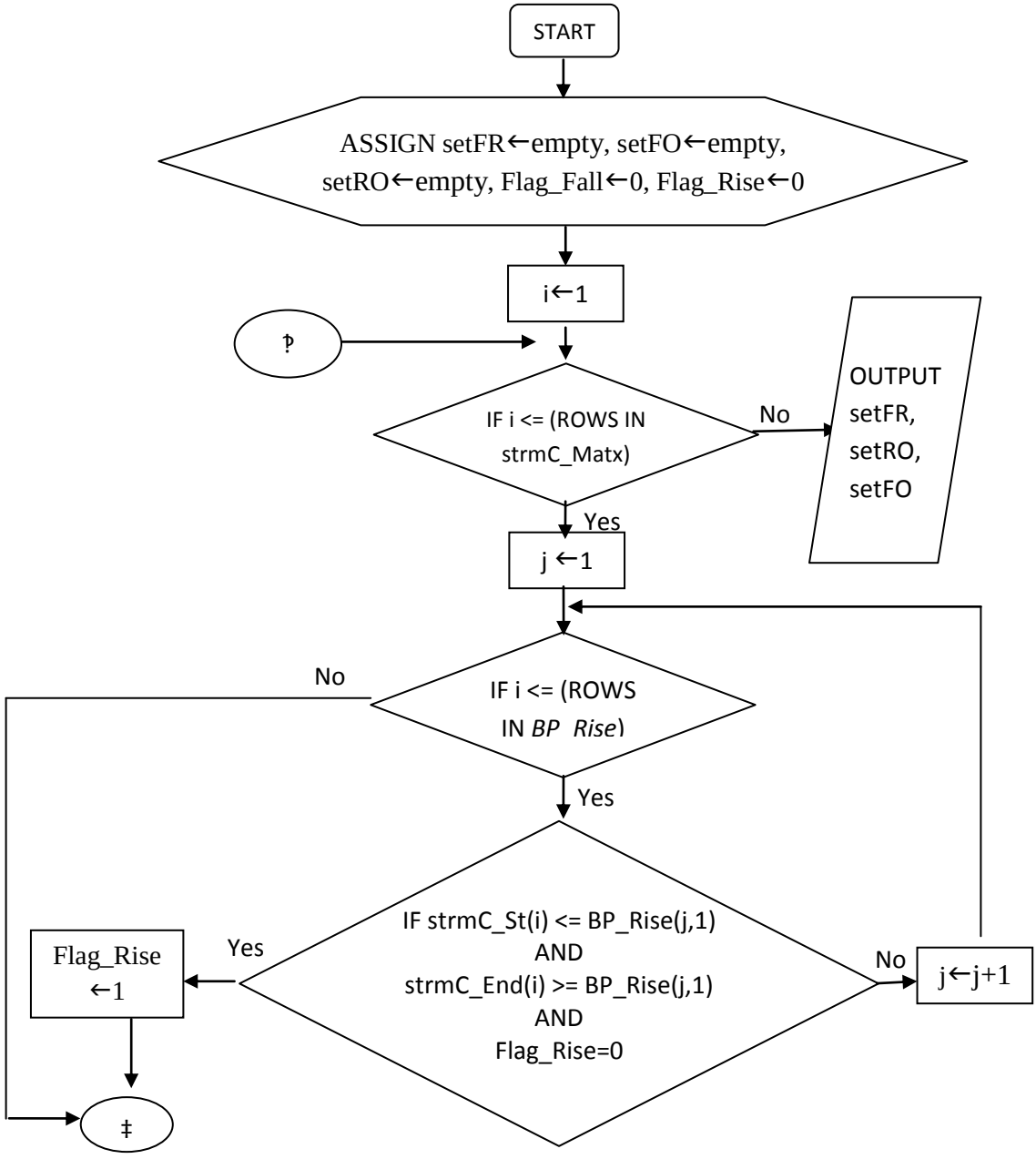
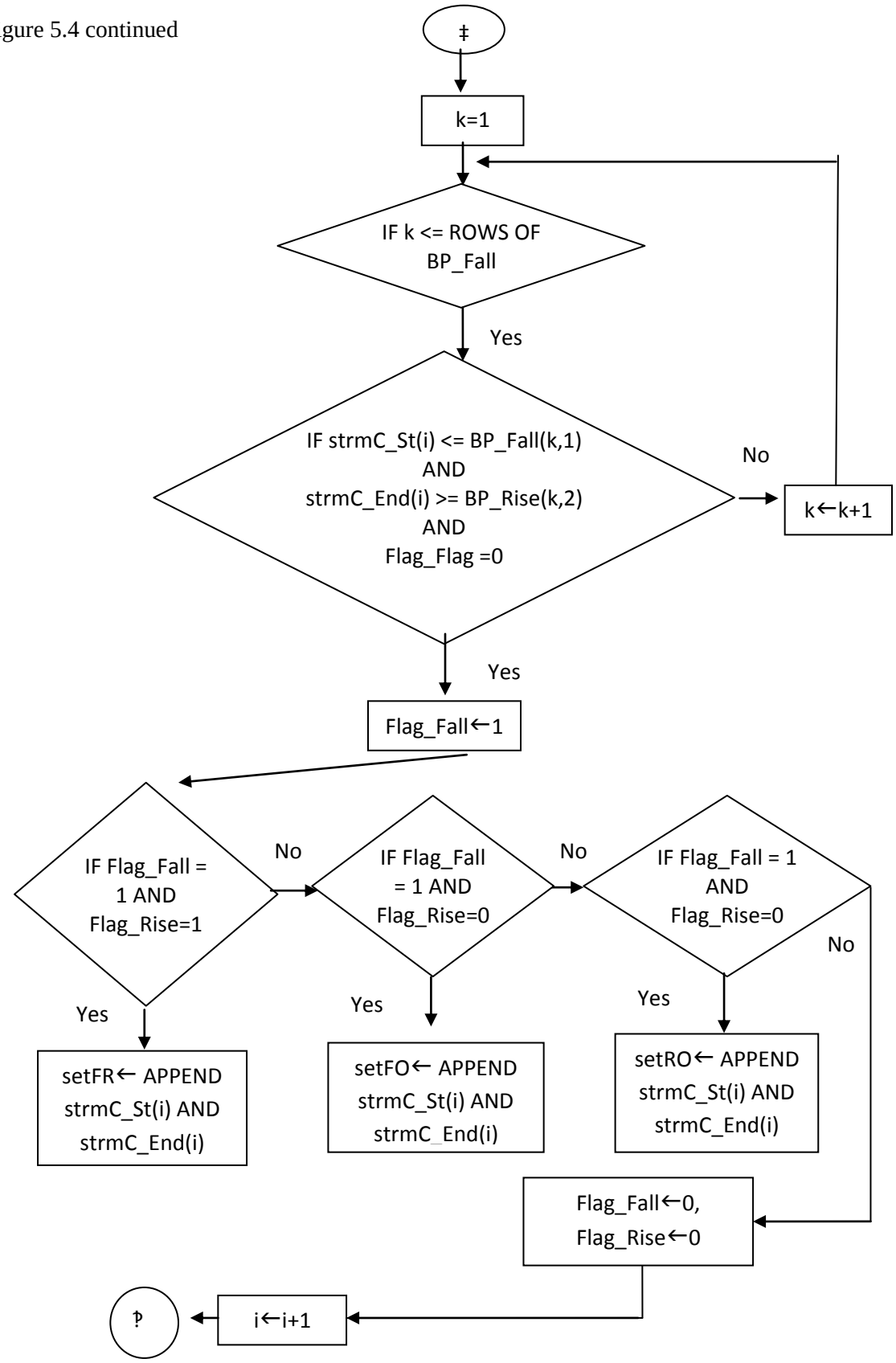


Figure 5.4 Flow chart for candidate storm classification based on primitive events

Figure 5.4 continued



5.3 Candidate Storms in GOMOOS Dataset

This section examines candidate storms generated from the composite event assembly of primitive events extracted from the GOMOOS datasets and within the *chosen timeframe*. As described in Section 4.2.1.2, a Julian date corresponding to each observation is stored within the MATLAB structure, which can be referenced using an index. The output of the algorithm illustrated in Figure 5.4 includes the following sets: *setFR* (composite event containing both *BP_Fall* and *BP_Rise* primitive events), *setFO* (composite event containing *BP_Fall* primitive events only) and *setRO* (composite event containing *BP_Rise* primitive events only). These two-dimensional arrays contain indexes for start and end times of candidate storm events. Table 5.1 presents the results of the algorithm presented in Figure 5.4 for the GOMOOS dataset. It is noteworthy that *setFO* and *setRO* have negligible number of candidate storms as compared to *setFR*. One example from each set of candidate storms is presented to illustrate the types of conditions which lead to the different outcomes. A complete list of candidate storms can be found in Appendix A.

Set	*setFR*	*setFO*	*setRO*
Number of candidate storms	113	10	10

Table 5.1 Summary of candidate storms from the GOMOOS dataset (10-01-2004 to 07-09-2007) *See Appendix A for the complete setFR*

A plot of marker parameter and primitive events BP_Rise and BP_Fall along with time for the candidate storm belonging to set *setFR*, is shown in Figure 5.5(a). It starts at 2005-12-19 23:00 and ends at 2005-01-21 12:00. For visual representation, a time series

from all buoys for the duration of the candidate storm are ploted in yellow color. The *BP_Fall* and *BP_Rise* primitive events initiating and terminating the candidate storm are shown in red and green color respectively. The rectangle highlights the boundary of the candidate storm for visualization within the time series. As one can see, there are several *BP_Fall* and *BP_Rise* events that form the candidate storm event. The Spatial Progression String (SPS) for this candidate storm (referred to as candidate storm #20) is represented as: '1B,4ABE,1AB,2ABN,2ABFN,1BFN,1Fa,3Fab,2F,1,1f,15abf,3bf,1f'. In this storm SPS, the comma-separated substrings contain letters that designate a buoy location and the upper and lower case designate the initiating and terminating primitive event types. The first few substrings contain predominantly uppercase letters, whereas in the middle, some letters within a substring change to lower case. At the end of the SPS, all the letters are lower case. This pattern of upper case letters indicates a falling gradient followed by lower case letters indicating a rising gradient in the barometric pressure parameter. This pattern is also evident from the graphic representation of the times series and candidate storm #20 as shown in Figure 5.5(a), The time series indicates all the buoys observing a falling barometric pressure followed by a period where some buoys see a rise whereas others see a fall, which in turn is followed by a period of sustained rise in barometric pressure across all buoys. Thus, the SPS represents the spatio-temporal behavior of the marker parameter during the storm. Figure 5.5(a) presents visualization of candidate storms #20 (from set *setFR)* and #1 (from *setFO*). Candidate storm #20 starts from 01-19-2005 and ends at 01-21-2005 12:00. Candidate storm #1 from set *setFO* starts at 01-22-2005 23:00 and ends at 01-23-2005 09:00. It can be seen that storm #20 contains both *BP_Fall* and *BP_Rise* events, whereas #1 contains only *BP_Fall* events. It can be

noted that candidate storm #1 ends with missing values. In case there were no missing values, there could have been a barometric pressure rise associated with the fall. The SPS of candidate storm #1 from set *setFO* is: *'2A,1AB,2B,4BN,2N*'.

Figure 5.5(b) shows a candidate storm from the set *setRO*. This candidate storm was selected because the observations from some buoys satisfied the thresholds for a short time. Since the duration of this event is only 2 hours, we filter these events out. The SPS for candidate storm #1 from *setRO* shown in Figure 5.5(b) is: '3f'.

setFR [20 01-19-2005 23:00 01-21-2005 12:00]; setFO [1 01-22-2005 23:00 01-23-2005 09:00]

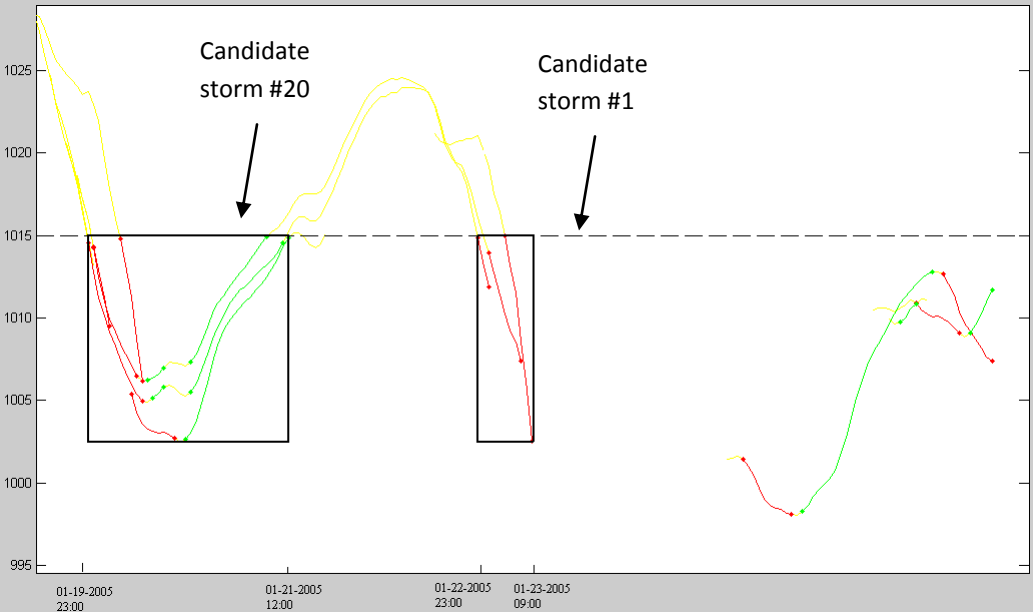
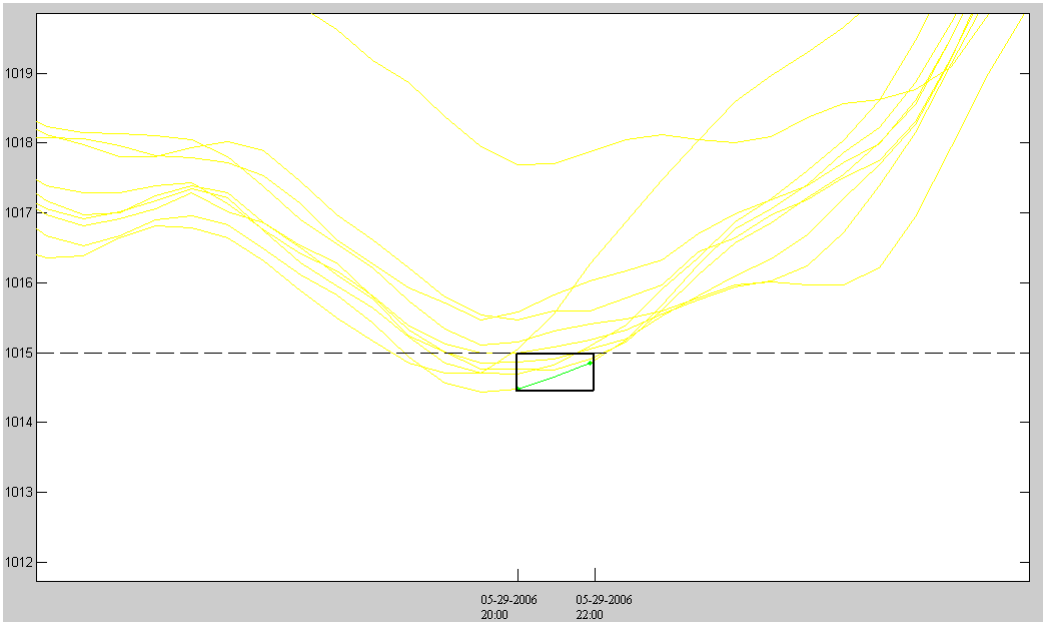


Figure 5.5 Time series plot of candidate storms from classification sets

a) Candidate storm #20 from the set *setFR and #1 from set setFO*

Figure 5.5 continued



b) Candidate storm #1 from the set *setRO*

The *setFO* and *setRO* candidate storms are typically artifacts of missing data and can be analyzed further to determine if they might be valid storm candidates. However, for the remainder of this analysis however, these potential candidate storms are ignored.

In order to evaluate the effectiveness of storm detection using the EO approach, we compare the candidate storms to an independent source of historic weather data. The next section presents the validation of the candidate storms comparing them to National Climatic Data Center (NCDC) storm events.

5.4 Validation of Candidate Storms

This section presents validation of candidate storms detected using the *EO* approach. The validation data set is an independent historic data source maintained by the National Climatic Data Center (NCDC, 2011). This database contains information about storm events that includes start and end times, location and other observations on the intensity and direction of low pressure movement. There are semantic differences between the terms used to describe a storm between NOAA's National Weather Service (NWS, 2011) and the National Climatic Data Center (NCDC). Since we are interested in validating the candidate storms assembled from primitive events into composite events, we compare the EO derived candidate storm with the storms found in NCDC Storm Events database and NWS. The EO approach identifies just a single category of potential storms based on barometric pressure. NCDC, however, recognizes several different types of storms, whereas NWS uses a broader definition of storm. In order to get a reasonable comparison, all relevant storm types needed to be selected from the NCDC database.

5.4.1 Range of NCDC and NWS storm definitions

The NWS glossary defines a *storm* as a '*disturbed state of the atmosphere, especially affecting Earth's surface and strongly implying destructive and otherwise unpleasant weather'.* A '*storm' 'warning'* is defined as *'warning of sustained surface winds, or frequent gusts, in the range of 48 knots (55 mph) to 63 knots (73 mph) inclusive, either predicted or occurring, and not directly associated with a tropical cyclone'.* '*Snow squall*' is defined as *'intense, but limited duration, period of moderate to heavy snowfall, accompanied by strong, gusty surface winds and possibly lightning'.* The term '*High Wind'* is defined as

'sustained wind speeds of 40 mph or greater lasting for 1 hour or longer, or winds of 58 mph or greater for any duration'. '*Thunderstorm'* is defined as a *'local storm produced by a cumulonimbus cloud and accompanied by lightning and thunder'.* '*Hail'* is defined as *'showery precipitation in the form of irregular pellets or balls of ice more than 5 mm in diameter, falling from a cumulonimbus cloud'.* And '*Rain'* is defined as' *precipitation that falls to earth in drops more than 0.5 mm in diameter'.*

Terms related to storms are identified by NCDC and NWS but are not identical in definition. The NCDC storms database contains events that include *Thunderstorm wind*, *Hail*, *Lightning*, *Rain*, *Flood*, *High/Strong wind*, *Heavy Snow*, *Winter Storm* and *Storm Surge*. However, the term *Winter Storm* and *Snow Storm* cannot be found in the NWS glossary. Since there are semantic differences in the use of these terms, to allow comparison, we re-categorize related storm events into weather event definitions derived from the NWS glossary as follows.

Figure 5.6 illustrates a classification tree of storm event types which explicitly specifies the hierarchical relationship between storm events. For the purpose of validation, we state the meaning and scope of each term.

Storm is the highest class of event. Subclasses of *Storm* are *Snow Storm*, *Thunderstorm* and *Rain Storm*. The definition of *Snow Storm* is similar to *Snow Squall* in the NWS glossary. *Winter Storm* is regarded as a synonym of *Snow Storm* in this work, since most records of *Winter Storm* in NCDC Storm Events database mention snowfall. The definition of *Thunderstorm* is similar to the NWS glossary. NCDC records of *Thunderstorm winds* and *Lightning* are considered subclasses, having a 'partOf' relationship to *Thunderstorm*. Similarly, *Hail* and *Flood* are subclasses with a 'partOf'

relationship with class *Rain Storm*. The definition of *Rain Storm* includes precipitation in the form of hail, sleet or water.

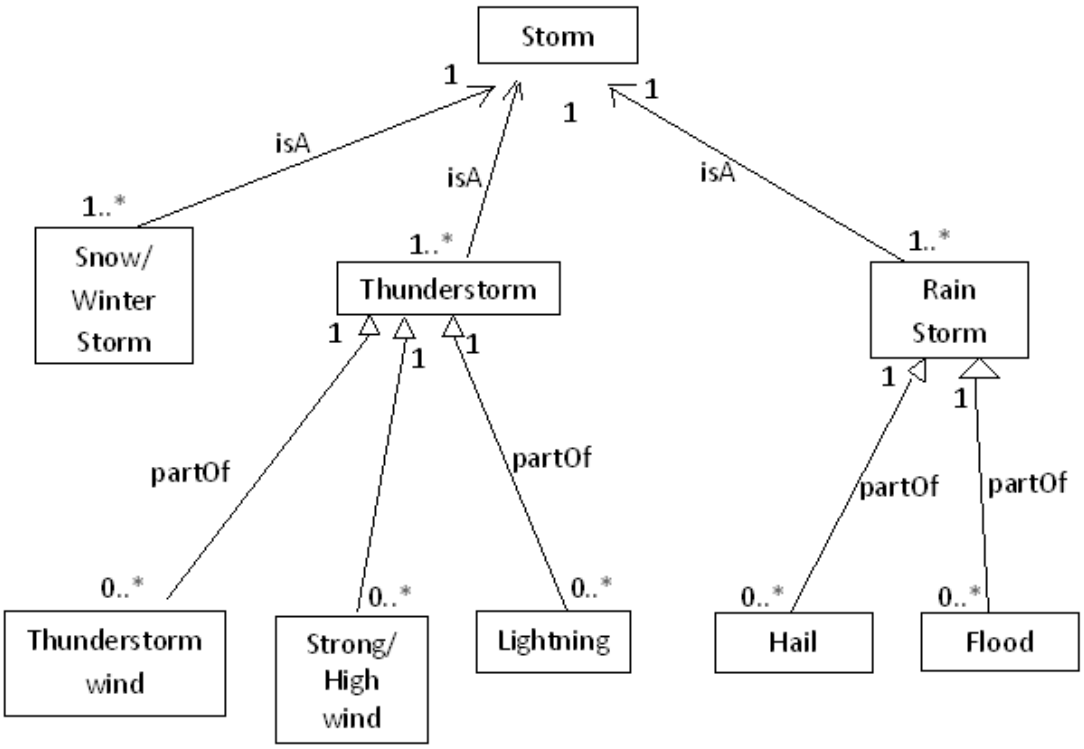


Figure 5.6 Classification tree for storm terminology used for validation

Based on the above definitions and hierarchy of NCDC storm events, we match the date of candidate storms with NCDC storms of the types summarized above. When there is a date match, the EO candidate storm is assigned the storm category assigned by NCDC. The results of validation are presented in Table 5.3. In this table, the numbers are storms seen by each database i.e., NCDC and the algorithms of the EO approach. The '+' and '-' sign represents whether corresponding records were found or not found within the data sources respectively.

	NCDC+	NCDC-	
EO+	74	39	113
EO-	9		
	80		

Table 5.2 Summary of results of candidate storm validation

Event Type	Snow Storm	Thunderstorm	Rain Storm	Strong Wind	Unidentified	Total
Event Numbers	25	14	23	12	39	113

Table 5.3 Results of validation of candidate storms using NCDC storm event types

Table 5.2 shows that out of the total 113 candidate storm events detected through EO approach, 39 had no match in the NCDC Storm Events database. This means that no corresponding storm events were found for 39 candidate storms when compared to the NCDC Storm Events database for the *chosen timeframe*. There were 9 cases in which NCDC identified a storm during the chosen timeframe but the algorithm identified no corresponding storm. Table 5.3 indicates that out of the events validated, most were of event type *Snow Storm* and *Rain Storm.* Out of the 9 storms seen by NCDC but not seen

by the algorithm in the *chosen timeframe*, 2 were *Snow Storms*, 6 were categorized as Rain storms and one was categorized as a Strong Wind event (see Table 5.4).

Rain			Strong Wind	Snow Storm
Flood	Thunderstorm Wind	Hail		
6			1	2

Table 5.4 Type of NCDC events not detected by algorithm

Sixty-four percent of the candidate storms were validated by events from the NCDC Storm Events database. About thirty four percent (i.e., 34.5% =39/113) of the candidate storms are not validated in NCDC database. Importantly, only nine (i.e., 11.2% = 9/80 of NCDC storms) of the storms recorded by NCDC are not found by the algorithm implementing the EO approach.

During validation, it was found that typically, more than one NCDC Storm Event matched the time of a candidate storm identified by the algorithm. There are two reasons for this. First, the NCDC database contains multiple entries for one storm event observed at more than one location. For example, NCDC Storm Events database may contain two records for a single Snow Storm event observed at spatially close but different locations, say Bangor, and Ellsworth. Thus, several storms which are close in time (less than 12 hour temporal difference) could be assumed to belong to a single storm.

The high number of identified false positives (34.5%=39/113) candidate storms for which no NCDC counterpart storm was identified may be a result of NCDC manually recording only significant Storm Events. In comparison, the EO candidate storms identify any disturbance in the atmosphere based on significant barometric pressure drop followed by a rise. Thus there is a potential for larger numbers of EO candidate storms than NCDC recognized storms. Another reason for the difference may be that the offshore GOMOOS buoy locations are picking up offshore storms that were not identified by the terrestrial stations used to identify the NCDC storms.

5.5 Summary

This chapter presented composite event assembly and assembly methodology and algorithms for implementing the EO approach using primitive events from the GOMOOS dataset. Results of the storm event detection, the candidate storms, were presented and discussed in Section 5.3. Validation of detected candidate storms was presented in Section 5.4. Although most storm events in the NCDC Storm Events database for the chosen timeframe were detected by the algorithm (92.5%=74/80), we also saw a significant number of false positive candidate storms. This means that the algorithm is picking up atmospheric disturbances other than those recognized in the NCDC Storm Events database. The next chapter delves into classifying the candidate storms with the goal of finding new information about storm events. New ontological knowledge found after further processing the candidate storms is presented and discussed.

Chapter 6

COMPOSITE EVENT CHARACTERIZATION AND CLASSIFICATION

This chapter presents a methodology for classifying composite events based on constituent primitive events. The methodology is illustrated using the composite storm events. The chapter presents a methodology for characterizing and classifying composite events to explore their structure and facilitate the discovery of new knowledge. The goal is to characterize the substructure of composite events based on the primitive events. Composite events are characterized based on key (i.e., initiating and terminating) primitive event behaviors. The approach is illustrated using the candidate storms identified in the previous chapter.

The first section of this chapter describes classification of composite events based on the initiating and terminating events, the spatial sequencing of their onset and termination, and the temporal relationships between key primitive events and non-key primitive events. In order to discover new knowledge, classification of candidate storms based on spatial behavior of the marker parameter is presented in the later part of the chapter. Some statistical observations on behavior of wind speed, air temperature and wind direction are presented.

6.1 Classification of Composite Events Based on Initiating and Terminating Events

There are several ways for classifying high-level events depending on the interest of the user. Composite events can be characterized by the types of their constituent primitive events, particularly the initiating and terminating primitive events. The spatial and temporal ordering of the initiating and terminating events (i.e. how similar are starting position and spatial sequences of primitive events), and their relationship to non-key primitive events provide information for characterizing the composite event. The two methods for classifying composite events used in this thesis are profile based and SPS based classifications. These are discussed in detail in the next two sections.

6.1.1 Profile Based Composite Event Classification

The term ‘profile’ refers to the qualitative and temporal behavior of key primitive events (those that define the initiating and terminating conditions for the composite event). The criterion for profile based composite event classification is the overall pattern of key primitive events within the candidate composite event interval. Given the primitive event types there are several possible shape patterns which can occur. Similar to Agrawal et al. (1995), who employed shape descriptors; a form of shape descriptor can be applied to the primitive event sequences of a composite event and assigned a symbol. If the initiating primitive event is a rising trend and the terminating primitive event is declining trend, the profile shape is a peak. In the reverse case, the profile is a valley or a V shape. Sequences of the basic profile shape can create compound shapes. For a set of AbstractionType,s

A={fall, steady rise} the following set of basic pair-wise profiles shapes are possible, as illustrated in Figure 6.1.

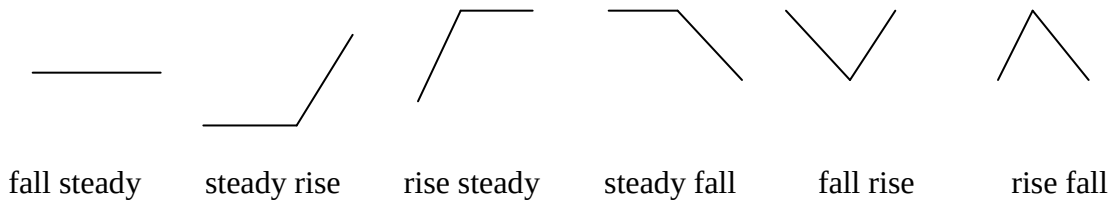


Figure 6.1 Basic pair-wise profile shapes

These primitive event profile sequences represent a particular sequence of phenomena states or trends that can be used to classify composite events.

6.1.2 Spatial Progression Based Classification

The Spatial Progression Strings provide another basis for classification of composite events. A SPS, as described in Section 5.2, represents both primitive event type and the order in which locations detect initiating and terminating primitive events. Therefore an SPS can be used to represent the spatial progression of a high-level event in detail. SPS based classification can group high-level events by similarity in spatial direction of detection, progression or termination. As an example, assume a regular grid of sensor locations as shown in Fig 6.2.

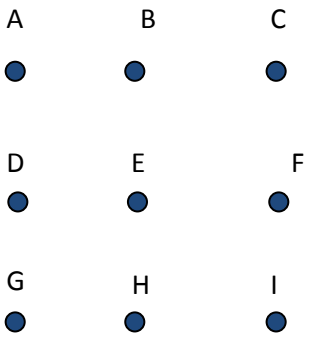


Figure 6.2 Illustration of regular grid of sensor locations

One pattern of SPS might be ‘G,E,C,g,e,c’. Such a pattern indicates a progression of the initiating condition from the lower left corner to the upper right followed by a progression of the terminating condition on the same path. It also indicates a small spatial footprint for the event, In other words, it traces a narrow path through the set of node locations. Another SPS pattern might be ‘ADG,ADGBEH,BEHCFI,CFI’. Such a pattern represents more of a frontal behavior i.e., the event moves as a front from left to right. An SPS pattern could also indicate a situation in which all locations see the initiation of the composite event simultaneously, e.g., the first SPS cell contains all locations (ABCDEFGHI). The SPS can thus be used to classify composite events on direction of movement as well as some basic patterns (e.g., path, front, or synoptic event). Classification of storms using the Spatial Progression String is presented in Section 6.2.2.

6.2 Classification of Candidate Storms

For the candidate storms, we explore how patterns in behavior of the marker parameter primitive events (i.e., barometric pressure) relate to the type of storm and non-marker

primitive events. We also explore the relationship between the spatial progression string of a storm and seasonal associations. The next section presents the methodology to classify candidate storms based on marker parameter profiles.

6.2.1 Profile Based Storm Classification

The general profile of a storm may be described as a significant and continuous *fall* in barometric pressure followed by a *rise*. From the GOMOOS dataset, BP_fall and BP_rise primitive events were obtained and assembled into candidate storms as summarized in section 5.3. Candidate storms were identified in *setFR* i.e., as temporally clustered intervals of barometric pressure *fall* primitive events followed by barometric pressure *rise* primitive events. These candidate storms can now be classified based on these constituent primitive events. The first criterion for profile based classification classifies candidate storms into Tiers I & II Storms. Tier I Storms are storms which show a barometric pressure fall followed-by a rise, such that the rise forms a *recovery*. Recovery is determined by checking if the highest value of barometric pressure in *BP_Rise* events participating in a storm candidate satisfies a line threshold, (*varLowMagTh)*. Tier II Storms contain candidate storms having barometric pressure *fall* and *rise*, but the *rise* does not *recover*, meaning that it does not meet the recovery threshold. The algorithm shown in Appendix A presents the method of classification of candidate storms into Tier I & II Storms. A buffer (2 millibars, in our case) on the hard line threshold is used to include candidate storms that do not strictly meet the line threshold requirement. Matrix *setFR*, which stores storm candidates containing both *fall* and *rise* events as output from algorithm in Figure 5.4, is the input for this classification step. Since the classification is

based on whether a barometric pressure recovers or not, we use only *BP_Rise* events, which are output from the algorithm in Figure 4.9. The algorithm in Figure 5.4 first checks for a temporal overlap within the time intervals of *setFR* and *BP_Rise*, and attaches *BP_Rise* primitive events to respective candidate storms. Second, it finds the maximum barometric pressure value in *BP_Rise* within each candidate storm, followed by using a recovery threshold condition to find candidate storms that meet the recovery condition and storing them into a new array.

Summary of results of the algorithm (see Figure 5.4) for classifying candidate storms detected in the GOMOOS dataset is shown in Table 6.1. Out of the total of 113 candidate storms containing *fall* and *rise* barometric pressure primitive events (summarized in Figure 5.2) 110 storms were classified as Tier I storms and 3 storms as Tier II. It is possible to process *fall* only and *rise* only i.e., candidate setFO and setRO events further with increasingly relaxed time thresholds. Since our main interest is limited to the candidate storms showing the typical pattern of barometric pressure fall followed by a rise we do not consider events from setFO and setRO in any further processing.

	stormC_Tier I (Contains *Fall* and *Rise*- such that *Rise* constitutes a recovery)	stormC_Tier II (Contains Fall and Rise but Rise does not constitute a recovery)
Number of candidate storms	110	3

Table 6.1 Results of candidate storm classification

The temporal clustering and temporal thresholds for storm detection can create situations in which more than one set of initiating and terminating primitive event sequences can be included within a candidate storm interval. The storm definition requires at least one BP_Fall primitive event followed by a BP_Rise primitive event but there may be additional sets leading to the following set of profile shape options: *V, W_{half}, W* and *Complex*. Shape '*V'* contains a pattern where there is one *fall* subset followed by one *rise* subset. Shape *W* contains two *falls* and two *rises*. Shape W_{half} contains either two *falls* and a *rise* or one *fall* and two *rises*. The shape *Complex* contains combinations of more than two *falls and rises*.

Implementation of the storm classification into shapes *V, W_{half}, W* and *Complex*, as described above, is shown in the algorithm in Appendix B. Outputs of the algorithm are the sets: *setV, setW, setW_{half}* and *setComplex*; each of which are two dimensional matrices

that store start and end time of candidate storms. Figure 6.3 presents time series visualization of an example candidate storm from each of these classes. Lines plotted in yellow color are barometric pressure observations at all buoy locations for the time interval of the candidate storm. The lines in *red* show *BP_Fall* primitive events and the lines in green show *BP_Rise* primitive events associated with the candidate storm. The width of the rectangle in the plot is derived from the start and end time of the candidate storm. The height of the rectangle is derived from the range of the marker parameter value i.e., barometric pressure.

Association of the shape symbols with the visual profile pattern is apparent. To identify the shapes within a pattern, the algorithm divides the BP_Fall and BP_Rise primitive events into temporal subsets and creates the profile type based on the number of subsets found within a candidate storm.

As an example, a candidate storm from setV (Figure 6.3-a) has a V shaped pattern. The values of barometric pressure at all buoy locations clearly recover from a significant fall. Thus, the condition used for classification of a storm candidate in class setV is that it has one distinct *fall* subset followed by one *rise* subset. Figure 6.3-b shows an example of a significant *fall* followed by a significant rise and another significant fall in barometric pressure. The rise after the last fall was left out of the candidate storm because it did not meet the rate threshold. Within the candidate storm, notice that there is a temporal gap after the rise event subset and before the fall event subset. A threshold value on the temporal separation of two clusters of similar event subsets (12 hours in our case) is used to determine if two subsets are sufficiently close to be included in the same candidate storm. Thus, the classifying condition for a candidate storm for the half W profile class is

to have either two *fall* subsets and a *rise* subset or one fall and two rise subsets. Therefore, the $setW_{half}$ includes those candidate storms with either an extended V shaped pattern or those which could have been W shaped, but the algorithm did not recognize either initiating or terminating events due to threshold values.

Figure 6.3-c shows an example of a candidate storm with a W shape profile. A significant fall in barometric pressure is followed by a significant rise, which is followed by another fall and rise. Thus, a 'W' shape is formed by two distinct subsets of *rises* and *falls*. Some flexibility in variations due to spatial observation can be built into the algorithm. Lastly, Figure 6.3-d shows a complex pattern containing more than two subsets of significant barometric pressure *falls* and *rises*. These patterns may indicate two or more storms following in very quick succession or a more complex storm structure.

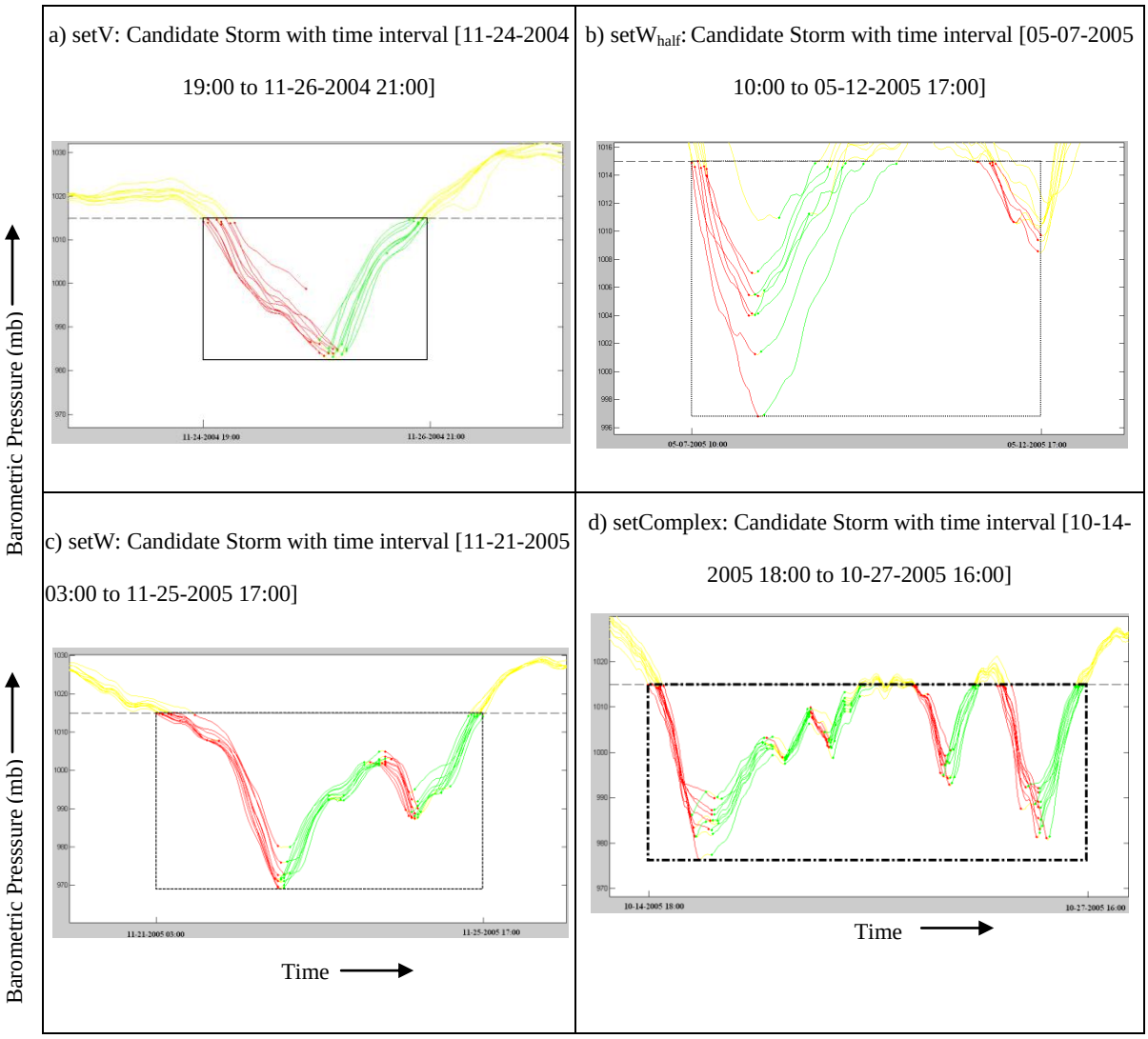


Figure 6.3 Time series visualization of setV, setW$_{half}$, setW and setComplex[1]

[1] Significant barometric pressure fall and rise events at all locations are plot in red and green resp. Non-qualifying data are plot in yellow.

	Tier I storms				Total
Profile	V	W_{half}	W	Complex	
Number of members	44	18	17	31	110

Table 6.2 Summary of profile based storm classification

Table 6.2 shows a summary of the profile based storm classification on Tier I candidate storms. It can be seen that most storms were classified as having profile shape V (44 out of 110) and Complex (31 out of 110). Profile shapes W_{half} and W have 18 and 17 candidate storm members respectively.

Summary of candidate storms by profile and storm event type is shown in Table 6.3. The storm event type is derived from the validation step described in Section 5.4. It can be seen that the number of NCDC non-validated storms contain fewer storms in the *Complex* profile class. This is likely because storms with a *Complex* shape profile tend be severe and prolonged, increasing the chances of being recorded in the NCDC storms database. Since the profile is complex, it could also include more than one NCDC storm event occurrence, thereby increasing its chances of validation.

The second noteworthy observation from Table 6.3 is that higher numbers of Snow Storms (13 out of 31) tend to be associated with Profile *Complex*. A reason for this may be that snow storms are often associated with severe weather which may contain high wind, extreme wind chill and/or rain. The NCDC strong wind events are least often associated with the complex shape type. The most frequent of the unidentified (no

matching NCDC storm) are associated with shape V. The V shape storms have the shortest durations which may have some bearing on their being detected less often.

Threshold values used in the process of detecting primitive and composite events have a bearing on the resulting candidate storms. Discrepancy in classification due to threshold values and other factors is also an issue and discussed in the next section.

		Storm Type					
		Snow Storms	Thunderstorms	Rain Storms	Strong Wind	Unidentified	Profile Type Total
Profile Type	V	8	4	8	4	20	44
	W_{half}	4	3	4	0	7	18
	W	0	3	3	3	8	17
	Complex	13	4	8	2	4	31
	Storm Type Total	25	14	23	9	39	110

Table 6.3 Profile based classes and validated storm event types

6.2.1.1 Discrepancy in Profile Based Storm Classification

There are some limitations that lead to discrepancies associated with profile based classification. Thresholds and the presence of missing data values can have an effect on the resulting candidate storm classification. Figure 6.4 presents some time series plots that show discrepancies found during profile based classification due to choice of thresholds.

Figure 6.4-a shows a Tier I candidate storm having a profile *BP_Rise* primitive event followed by *BP_Fall* event which could possibly have been a candidate storm. However, since the pattern of interest is *fall* (i.e., *BP_Fall)* followed by *rise* (i.e., *BP_Rise)*, the pattern is not included as a candidate storm. A missing data section can be seen just preceding the *BP_Rise* event, which may have included the target pattern but we do not know for sure. Due to missing observations, potential candidate storms like these are not included in the analysis.

Figure 6.4-b shows an example storm profile visually shaped like a W, but the algorithm classified it as shape *Complex*. This discrepancy could be considered as a limitation of our methodology for detecting such patterns and could be overcome using another approach such as curve fitting before classification of shape. This is however beyond the scope of this thesis.

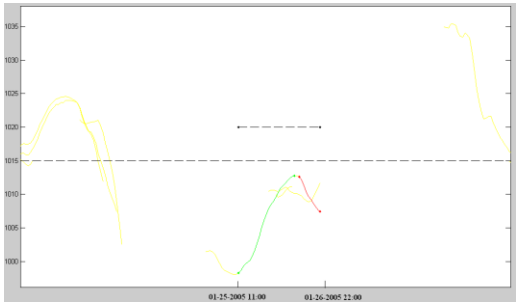
Figure 6.4-c shows a rectangle-bound candidate storm classified as shape V starting at 04-19-2006 00:55 and ending at 04-21-2006 06:00. However, it can be seen that before the detected storm, there is a barometric pressure *fall -rise* pattern in which the *rise* does not show a complete recovery. The term ‘complete recovery’ is used when the highest barometric pressure in *BP_Rise* primitive events within a candidate storm event meets the

recovery threshold i.e., *varLowMagTh.* The candidate starts at 04-14-2006 04:00 and ends at 04-17-2006 15:00. The difference between the two patterns is approximately 34 hours. Since there is more than a 12 hour separation, these were considered as separate patterns. The non-recovery pattern was classified as Tier II and not considered any further. Therefore, threshold values can contribute to discrepancies in classification and need to be carefully evaluated.

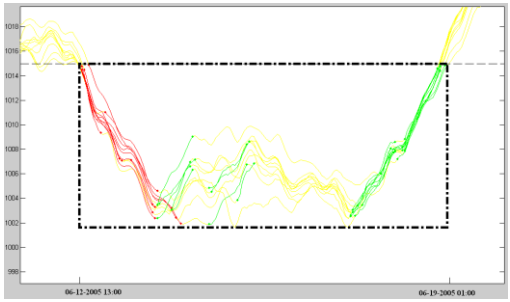
Figure 6.4-d shows a plot of *BP_Fall* and *BP_Rise* events such that the difference between *fall* and *rise* clusters of events is more than 12 hours. This separation interval leads to ignoring the fall and rise clusters in classification altogether.

Thus, it can be seen that thresholds and missing values can affect the clustering process, thereby affecting classification and candidate storm detection. The next section presents storm classification based on the spatial progression strings for candidate storms.

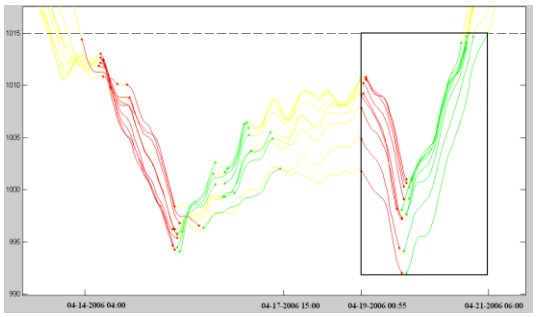
a) Discrepancy due to missing information



b) setComplex that should have been W

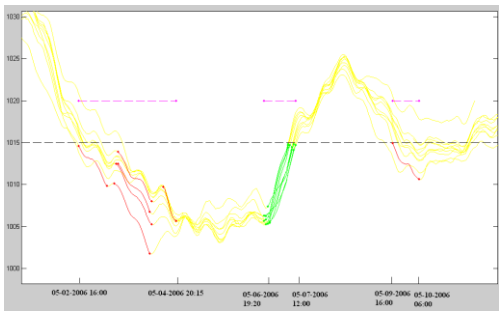


c) setV that should have been setW



Time

d) setFO and setRO , that should have been setW



Time

Figure 6.4 Discrepancy in storm classification by profile

6.2.2 Classification Based on Storm Spatial Progression Strings

SPS described in Section 5.2, represents the order in which sensor locations detect the initiating and terminating primitive events. In this section we show how the SPS can be used to classify storms. Another alternative is to derive SPS based on primitive events such as significant barometric pressure *fall* and *rise.* However, in order to include locations which may detect the high-level event in non-significant levels, according to the thresholds, we derive SPS using first differences in barometric pressure values.

Table 6.4 shows a summary of candidate storms based on first sighting location as indicated by the SPS and the season of occurrence. Within the study data set, most candidate storms are detected first by buoys A01, B01 and C01. The total number of candidate storms is more than 110 because a candidate storm could be detected simultaneously by more than one buoy. The fewest number of storms are detected first by buoys L01, M01, I01. The spatial location of these buoys, shown in Figure 4.1, may provide the reasons for this. Buoy L01 is in the eastern Gulf of Maine, and fewer storms tend to enter the Gulf of Maine from the Eastern direction. Only 4 candidate storms were first detected by buoy M01which might be due to the location of the buoy close to the centre of the Gulf of Maine. Fewer storms are detected by buoy I01 first which may be due to its Northern location, and few storms arriving from inland and a northerly direction.

Seasons	A01	B01	C02	E01	F01	I01	J02	L01	M01	N01	Total Storms
Fall	14	5	2	0	1	0	6	0	1	4	33
Winter	13	13	10	10	6	1	0	0	0	1	54
Spring	9	4	2	3	3	3	4	1	1	6	36
Summer	2	8	8	0	3	3	2	0	2	3	31
Total Events	38	30	22	13	13	7	12	1	4	14	154

Table 6.4 Summary of storms by location of first detection and season

The SPS of storms detected first by buoys L01, M01 and I01 could be examined more closely for further information on the spatial progression of these storms. For example, let us take a look at one of the candidate storms having an SPS: '*ILMN, 1EIJLMN, 14ABCEIJLMN, 1ABCEILMN, 1ABCEIMNj, 1ABCEjl, 1ABijlmn, 1ceijlmn, 22abceijlmn, 2abceilmn, 1eln, 2ln, 6n*'. From this SPS, we can see that this storm was detected first by buoys L01, I01, M01 and N01. The sequence provides information on the general progression of the storm from North-East moving towards the South-West. Further, we can deduce from the SPS that this storm, #62 exited the GOM in the South-East direction i.e., buoy N01 was the last to record recovery of the barometric pressure. Thus, classification based on spatial ordering using SPS provides a unique way to represent

important information about the storm. The reasoning behind use of SPS and its efficacy in describing a candidate storm is presented further with the Patriot's Day storm example.

The Patriot's Day storm of 2007 was recorded by NOAA (NWSFO, 2007) on 15th April 2007. The EO approach algorithm in Figure 5.3 detected a candidate storm starting at 04-12-2007 17:00 and ending at 04-20-2007 15:00. The time series plot and the rectangle highlighting the candidate storm interval is shown in Figure 6.5. It was classified as Tier I storm with a complex shape. NOAA's record of the storm on 15th April aligns with the second significant drop in barometric pressure.

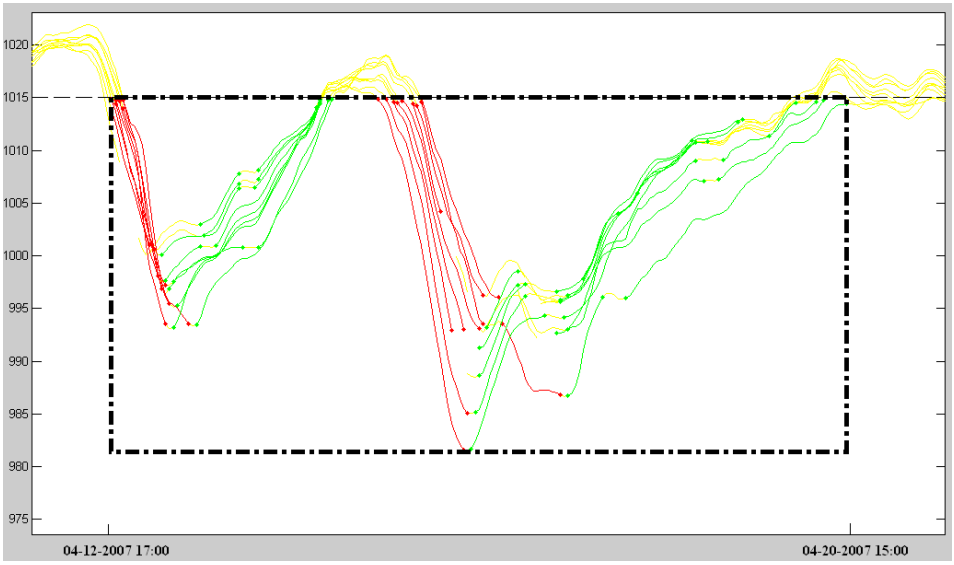


Figure 6.5 Candidate storm for Patriot's Day Storm of 2007

Consider the Patriot's Day Storm on 15th April 2007, described by NOAA:

"An area of low pressure intensified rapidly as it moved slowly from the southeastern United States on the morning of Sunday, April 15th to near New York City by the morning of Monday, April 16th. The intense low over New York City, in combination with

high pressure over eastern Canada, produced a tight pressure gradient across the area which resulted in strong east to northeast winds..."

Now consider the National Climatic Data Center (NCDC) description of the same storm:

"An area of low pressure rapidly intensified while tracking from the southeastern states to the southern New England coast from the 15th to the 16th. A tight pressure gradient developed between the low and high pressure centered over eastern Canada which also blocked the northern movement of the low. The intense low slowly drifted east from the 16th through the 19th while high pressure remained across eastern Canada…."

The common observation by both NOAA and NCDC are that the low pressure moved from South-East towards the North-East. The Patriot's Day storm was detected by the EO approach over the time interval [04-12-2007 17:00 to 04-20-2007 15:00]. The Spatial Progression String (SPS) for this candidate storm is represented by:

'1BC,1BCEF,1BCEFM,8BCEFILMN,1CEFILMN,1EFILMN,1FILMNc,1FILMce,1ILcen, 1Lcefmn,4Lcefimn,1cefimn,1cefilmn,1acefilmn,3abcfilmn,7abcefilmn,1efilmn,2efilm,1cefi lm,16abcefilmn,1abefilmn,2ln,1n,11,2A,2AB,1ABC,1ABCE,3ABCEF,1ABCEFM,1ABCE FIM,6ABCEFILMN,3ABCEFILN,3ABEFILN,1BFILN,1FILNa,1FILNab,1FILNabce,1IL Nabce,4Labcef,5Nabcef,2Nabc,5Na,2N,1Ncl,1Nbcel,1abcel,4abceflmn,6abcefilmn,4abcef ilm,1abceflm,3abceflmn,15abcefilmn,1abceflmn,2bceln,1bcen,2bcn,1bcln,5bclmn,1clmn, 6lmn,8flmn,5lmn,2ln,6n'

As described in Section 5.3, upper case letters in substrings of the SPS indicate primitive events with *fall* gradient in parameter and lower case letters indicate *rise* gradient. The number in front of each substring represents the number of times adjacent substrings

were repeated in SPS. The SPS shown above is datetime stamped such that each substring has a unique datetime stamp. The datetime stamp of the first substring is 04-12-2007 17:00 and the last substring is 04-20-2007 15:00. Since the granularity is uniform and known for all SPS strings i.e., 1 hour in our case, datetime stamp of any intermediate substring can be calculated if required.

The above SPS shows that the low pressure is detected first by buoy B01and C02 in the first hour, followed by additional detection at buoys E01 and F01 in the next hour and so forth. The NOAA and NCDC observations (i.e., low pressure moved from South-East towards North-East) are supported by the SPS, because buoys B01 and C02 are in the South-Eastern part of the Gulf of Maine, whereas buoys E01 and F01 are North-East of buoys B01 and C02. The SPS also provides further information on retreat behavior of the storm. In the previous example, the storm was last seen in retreat by buoy N01 indicating that the storm retreated in the South-East direction.

Effect of seasons on spatial detection of candidate storms is presented in Table 6.4. It appears that a higher number of storms i.e., 54 in number, are detected in the Winter. This finding is consistent with the observations in Table 6.3, that there are higher number of storm type Snow Storms (#25), which are known to occur in the *Winter* season. The number of candidate storms detected in the seasons of Summer, Fall and Spring appear to be closer in range i.e., 31, 33 and 36. The next section describes segmentation of candidate storms and discovery of new information from them.

6.3 Discovery of New Knowledge from Candidate Storms

This section examines the relationship of non-key parameter primitive events, namely wind speed, wind direction and air temperature within candidate storms. For this approach, candidate storms are segmented by uniform spatial behavior on the marker parameter. In other words, if all locations (i.e., buoys) show similar behavior (i.e, *fall* for an interval within the candidate storm); we consider the behavior of the parameter at that time to be spatially 'falling'. For example, we explore the relationship between spatial behavior of wind direction when barometric pressure is uniformly falling in a candidate storm. Temporal continuity in the spatial behavior of the parameter constitutes a 'segment' of the candidate storm. Therefore, a 'fall segment' refers to a time interval within the candidate storm when all buoys uniformly exhibit a *falling* behavior. A time interval when some buoys observed a *fall* whereas others observed a *rise* is called a '*fuzzy segment*'. Similarly, a time interval when all buoys exhibit a *rise* is called '*rise segment*'.

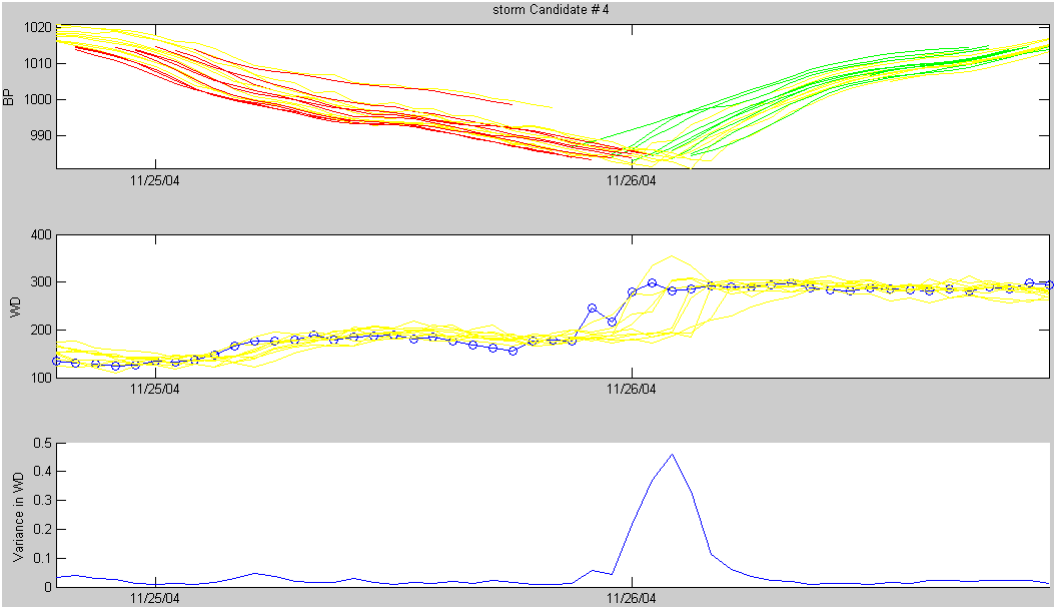
Discussion on the behavior of variance in an example candidate storm is presented next.

6.3.1 Segmentation and Variance in Candidate Storms

In this section, we present an example candidate storm and some observations on the spatial variance characteristics that will guide our approach to segmentation of the storm interval. In our discussion, *spatial variance* refers to the calculation of variance at a unit time, using observations across all available spatial locations. For example, spatial variance v at time t is calculated for values of a parameter at locations 1, through n. Only

locations with non-missing values of parameters at time t are included in calculating the variance.

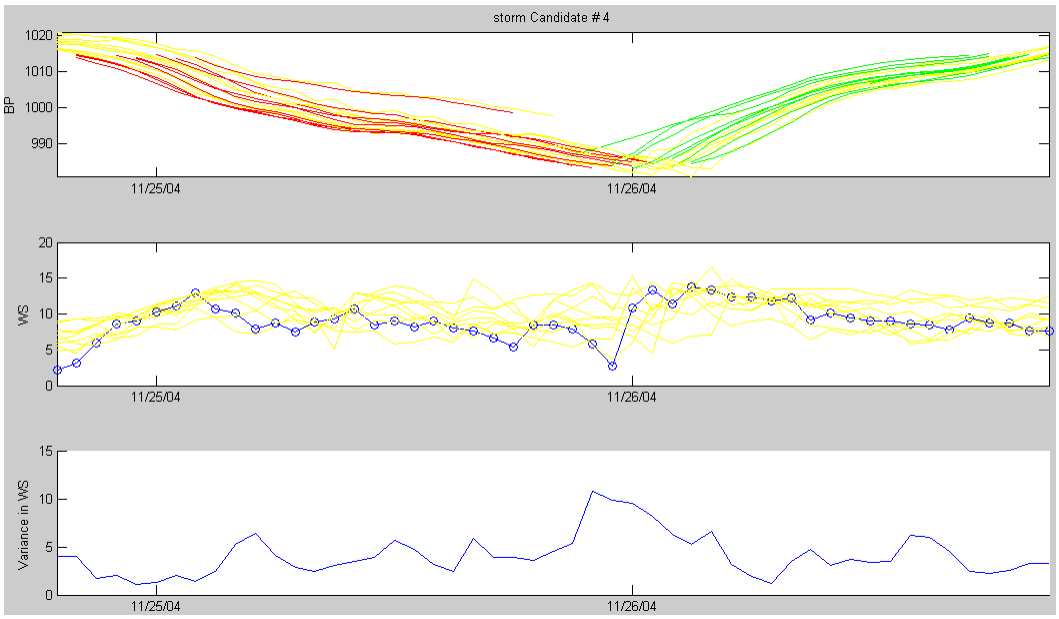
A plot for comparing barometric pressure observations with spatial variance in wind direction, wind speed and air temperature for a candidate storm with time interval [11-24-2004 19:00 to 11-26-2004 21:00] is shown in Figure 6.6-a, b and c respectively.



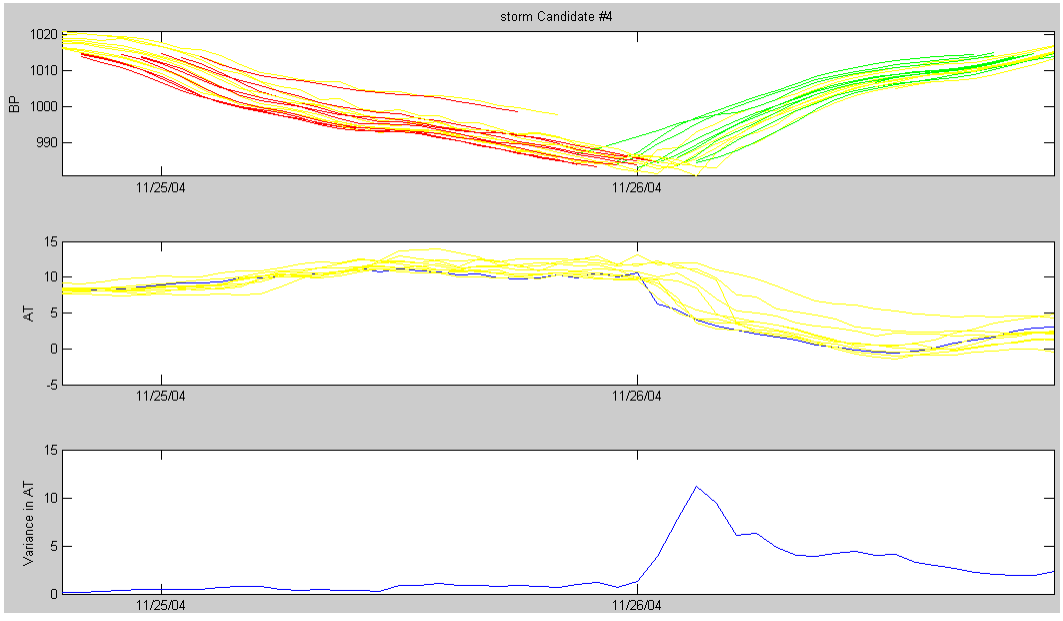
(a) Variance in Wind Direction

Figure 6.6 Comparative plot of barometric pressure and variance in Wind Direction, Wind speed and Air Temperature for Storm Candidate with time Interval [11-24-2004 19:00 11-26-2004 21:00]

Figure 6.6 Continued



(b) Variance in Wind Speed



(c) Variance in Air Temperature

There are several observations that can be made from Figure 6.6. It can be observed that spatial variance in wind direction is highest near the time when barometric pressure fall changes to barometric pressure rise. This behavior raises several questions that can be explored: Is this a common storm signature, particular to certain type of storm or severity of the storm? Does this occur due to the low barometric pressure or the high spatial variance in barometric pressure?

Similarly, it can be observed in Figure 6.6-b that the highest spatial variance in wind speed appears to coincide with the time when barometric pressure begins to recover at one location. This might be due to several possibilities, such as change of barometric *fall* to *rise* or spatial variation of barometric pressure across buoys or low barometric pressure.

In Figure 6.6-c, spatial variance in air temperature appears to be highest immediately after barometric pressure starts to recover.

We aim to explore some of these questions through segmentation of the candidate storm according to the spatial behavior of the marker parameter and calculating variance statistics. Methodology for segmentation is presented in the next section.

6.3.2 Methodology of Storm Candidate Segmentation

As mentioned in the introduction of this section, the goal of segmentation is to divide the candidate storm interval into sub-intervals corresponding to *fall*, *rise* and *fuzzy* behavior based on barometric pressure changes across buoy locations. The SPS is used to generate sub-segments within the candidate storm interval. Segmentation of candidate storms is

implemented using algorithm in Appendix C. Only candidate storms from Tier I were processed for segmentation.

The SPS of candidate storms is utilized for segmentation of candidate storms into *fall* (denoted by F), *rise* (denoted by R) and *fuzzy* (denoted by Z) segments. The algorithm evaluates each sub-string of SPS by determining if the tag indicates a capital or small letter, thereby creating a distinct string with F, R and Z tags. The length of the new string is the same as the length of the candidate storm in time, because each SPS sub-string generates one tag. Further, the new tag-string is evaluated for continuity to generate start and end times of F, R and Z segments.

6.3.3 Results of Storm Candidate Segmentation

Mean durations of segments F, R, and Z for candidate storms are presented in Table 6.5. It can be seen that generally, the fall (*F)* and rise (*R)* segment durations are longer than Z segment. This finding is consistent with the notion that locations closely placed in space will record similar behavior of a parameter due to their spatial proximity.

	Hours
Average duration of *'F'* segments	38.6
Average duration of *'Z'* segments	8.9
Average duration of *'R'* segments	49.6
Missing observations/# of storm Candidates in Tier I	16.2
Average duration of storm Candidates in Tier I	113.3

Table 6.5 Duration statistic of storm segments

Further, we explore the relationship of high spatial variance in wind direction, air temperature and wind speeds with spatial variance in barometric pressure. After segmenting candidate storm events into F, Z and R segments, we establish thresholds to determine high variance in air temperature and wind direction. High wind speed threshold is also used to determine its occurrence in segments. Threshold values of 0.5 for high wind direction variance, 0.6 for high air temperature variance, and a High wind speed threshold of 12 m/s were used. These threshold values were determined empirically and are the same for processing all candidate storms.

To determine statistics on distribution of spatial variance between segments, a simple looping algorithm counts total observations across all buoys for high wind direction variance (HWDVar), high air temperature variance (HATVar) and high wind speed

(HWS_Counts). A summary of results of the counts for all 110 candidate storms is presented in Table 6.6.

	High wind direction spatial variance, HWDVar (counts per hour)	High air temperature spatial variance, HATVar (counts per hour)	High wind speed counts, HWS_Counts (counts per hour)
F segment	0.08	0.85	0.40
Z segment	0.11	0.91	0.39
R segment	0.06	0.88	0.34

Table 6.6 Storm segment parameter statistics

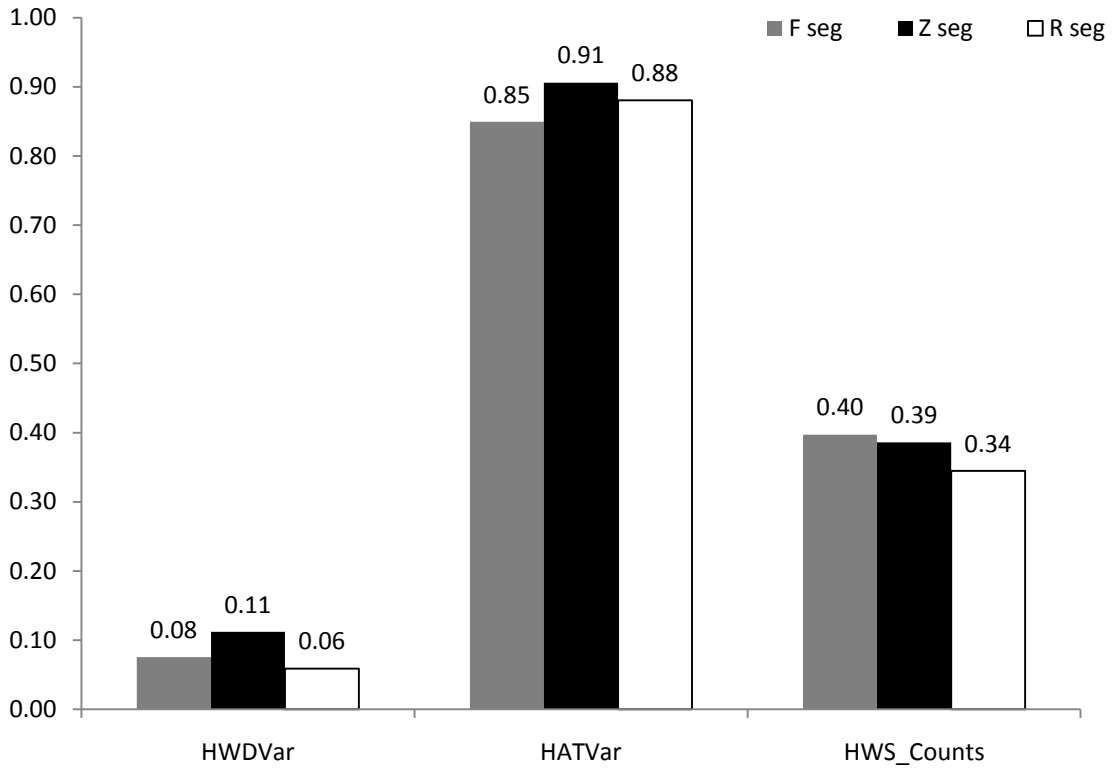


Figure 6.7 Storm segment statistic plot

The first finding from segmentation, as shown in Figure 6.7, is that high circular spatial variance in wind direction (i.e., HWDVar) is highest in the fuzzy segment. Because fuzzy segments occur mostly when there occurs an interchange from spatially uniform falling (i.e., F segment) to spatially uniform rising (i.e., R segment) or vice versa. This finding is consistent with the notion that the eye of the storm is preceded and followed by a sustained wind direction along with a reversal in wind direction.

The second finding from segmentation is that high air temperature variance occurs in the fuzzy i.e. Z segment. This finding may be of interest to meteorologists studying storm behavior.

The last finding is that wind speed tends to be highest in the initiating phase of the storm, when the barometric pressure is uniformly falling across all locations. The high wind speed in the fuzzy segment is comparable to the F segment, albeit slightly less than it. Finally, high wind speeds are least in the R segment when the barometric pressure is uniformly rising in space.

Thus, spatio-temporal behavior of non-key parameters with the marker parameter can be studied, thereby generating new knowledge about the high-level event using the proposed Event Oriented approach.

6.4 Summary

This chapter presented classification of candidate storms using two different approaches: profile based and SPS based classification. Results of both types of classification are helpful in providing more information about spatial and thematic behavior of the storm. Seasonal occurrence with respect to resulting storm classes was presented and discussed. Segmentation of candidate storms and variance statistics on air temperature, wind speed and wind direction were presented.

The next chapter presents the overall summary of this research in the light of the research questions that were posed in Chapter One. Conclusions and further work is discussed.

Chapter 7

CONCLUSIONS AND FURTHER WORK

In this chapter, major research contributions of the thesis are summarized, major contributions are highlighted, and some possible extensions of this research are presented as topics of further work.

7.1 Summary of the Thesis

This thesis presents a novel time series data abstraction approach, called the Event Oriented (EO) approach, which facilitates integration of information for detection of high-level spatio-temporal events from sensor data. As a case-study, detection of storms using the EO approach is implemented using sensor data from the Gulf of Maine Ocean Observation System. Validation of the detected storms using an independent historic data source (NCDC, in our case) is conducted to evaluate the EO approach. Classification of detected storms to yield new and additional information about them is illustrated. The EO approach could be applied in diverse domains. It could be useful to a traffic data analyst, for example, in finding a high-level event such as traffic congestion from traffic sensor data.

The EO approach takes an object view of events, which is consistent with existing event models such as the Geospatial Event Model (Worboys and Hornsby, 2004). A primitive event ontology was developed to specify primitive events and their abstraction from sensor time series. A domain level storm event ontology was developed to specify the

structure of initialization, continuance and termination of a composite event using primitive events. The storm event ontology is consistent with other domain-level ontologies such as SWEET (http://sweet.jpl.nasa.gov/ontology) and upper-level ontology such as the Event ontology (http://motools.sourceforge.net/event/event.html).

The thesis provides a formal description of the EO approach that includes threshold-based primitive event detection methods and construction of composite events. For implementing the EO approach to detect storms in GOMOOS data, Matlab® programming language has been used. Matlab® was chosen for its strong support for structures and computability.

Use of thresholds at several levels during the event detection process to detect and construct the high-level event is illustrated in the algorithms. Temporal concepts of *overlap* and *before/after* are used to find clusters of primitive events in the pattern of interest. Candidate storm events detected using the EO approach are validated against NCDC Storm Events database. A statistic on validation and an analysis of discrepancies is presented. Further, two methods of classification: Profile based and Spatial Progression String based were presented and evaluated.

Validation provided a way to understand the efficacy of the EO approach in detecting storms in GOMOOS data. Classification of candidate storms using Spatial Progression Strings provided a way to understand and represent the spatial progression of storms in the Gulf of Maine. Segmentation of candidate storms facilitated understanding the general relationship of barometric pressure with other variables such as wind speed, air temperature and wind direction.

The EO approach thus reduces dimensionality of sensor data, provides building blocks of observable system states, and facilitates information integration over different measurement protocols.

7.2 Major Results

An overall contribution of this thesis is the development, implementation and evaluation of the EO approach. Detection of a high-level storm event using univariate sensor data is illustrated. The EO approach could be applied to various other fields to identify high-level events such as forest fires, traffic jams etc. using sensor data.

The second major result is the evaluation of the results of storm event detection in GOMOOS data using the EO approach. It was found that the EO approach detected 92.5% of true positive storms and missed only 11.2% of NCDC storm records. This finding confirms the important role played by barometric pressure in storm dynamics. It facilitates the study of other less understood factors within the storm dynamic and serves to highlight discrepancies between detection and validation. An advantage of threshold based event detection in EO approach is the flexibility for the user to fine-tune and customize event detection by relaxing or tightening the thresholds.

The third result is the classification of candidate storms using profiles or spatial progression. Each type of classification provides unique ways to group and study storms further. Profile-based classification groups storms according to the complexity of behavior of barometric pressure. It was found that Snow storms are most commonly associated with complex type of profile in the marker variable barometric pressure.

Discrepancies in such classification were discussed to understand limitations of the method. The second type of classification method was based on the spatial progression of the storm. The resulting Spatial Progression String (SPS) provides an easy-to-read string, equal to the duration of the storm. It was found that a majority of the low pressure falls (therefore the storm) was first detected by buoys A01, B01 and C02, which are in the South-Western region of the Gulf of Maine.

The fourth and final result was the new knowledge generated as a result of segmentation of the candidate storm interval into system states: uniformly falling barometric pressure across all buoys, mixed, or uniformly rising across all buoys. It was found that within the candidate storms, high air temperature spatial variance occurred during the uniformly rising barometric pressure system state. The wind direction spatial variance was similar in rising and falling segments, where as in the state when there was a mix of rising and falling barometric pressure, it was least. In candidate storms, high wind speed spatial variance was found to be highest in the rising barometric pressure system state followed by the segment in which it was falling.

In the light of these findings, the hypothesis that '*high-level, spatio-temporal occurrence can be detected using low-level sensor measurements'* is not rejected. It should be noted however, that there were associated discrepancies in detection of the high-level spatio-temporal occurrence.

7.3 Further Work

This work had the limited scope of proposing the EO approach followed by a case study which included implementation of the EO approach for detection and validation of storms from GOMOOS sensor data. Immediate extensions that could be undertaken to this work are discussed in this section.

In this thesis, high-level event detection involved a single marker variable because for storm detection, barometric pressure was the appropriate and distinct marker for indicating initiation, continuance and termination of a storm. Other high-level events, a 'forest-fire' for example, could have different initiating and terminating events, such as an initiating 'spark' event and terminating 'recovery to normal air temperature' event. Since EO approach facilitates integration over more than one variable and could be applied to other domains, such a study could be undertaken to study further improvements over EO approach.

A comparative performance evaluation of EO approach to other approaches of high-level event detection using high dimensional data could be an important extension of this work. Particularly, comparison of EO approach to Unified Temporal Grammar (UTG) proposed by Ultsch (1996), an exciting prospect.

Refinement of candidate storms detected using EO approach using other parameters such as wind speed, wind direction, wind gusts, air temperature could be an extension of interest. Since these variables are an important indication of extreme weather events, like storms, these could be included in adding more information to the candidate storms, thereby helping in reducing false positive candidate storms from the results. Study of

temporal relationship between these variables with barometric pressure system state could lead to a better understanding of the storm dynamic.

Some additional topics of interest are methods to address missing values in composite event detection. The EO approach facilitates detection of missing events, patterns of which could be studied to improve performances in long standing sensor-based observation networks. Missing events could be included in high-level event detection using the Dempster-Shafer evidential theory to incorporate uncertainty into detection decision by sensors (Murphy, 1999). A confidence function could be generated which takes account of the missing data (Li, Lin, Son, Stankovic, & Wei, 2004).

Adaption of the EO approach algorithm to work in a distributed fashion within a sensor network will be an interesting extension of this work. Since the EO approach can detect high-level composite events by assembling significant primitive events over a spatial domain, this can be used to clearly identify those sensor nodes with non-significant behavior, thereby monitoring the extent of spatial spread and progression of the high-level event.

APPENDIX A

CANDIDATE STORM CLASSIFITION INTO

TIER I AND II

Input: Buoy structure, matrix, BP_Rise (*from output of Algorithm* 4.1), setFR (*from output of algorithm 5.1),* varLowMagTh (*same as input of algorithm 4.1*)

Output: storC_TierI, stormC_TierII

---

bpBuffer←user-defined value of threshold, we used *2mbs*, strmC_TierI←empty,
strmC_TierII←empty, setFR_Rises←empty

Step 1) Attach *BP_Rise* events from all buoys falling within candidate storm to respective candidate storms
FOR each j in (length of setFR)
    FOR each i in (length of matrix BP_Rise)
        IF BP_Rise(i) OVERLAPS setFR(j)
            BP_Rise(i) ←setFR_Rises
        END IF
    END LOOP
END LOOP

Step 2) Find and store maximum value of *BP_Rise* overlapping with each interval of *setFR_Rises*
FOR each i in (length of setFR_Rises)
        NESTED LOOP through all buoys within interval of *setFR_Rises* to find maximum barometric pressure value and store it as fourth column value in each setFR_Rises
END LOOP

Step 3) Segregate Recovery and non_Recovery storms
FOR each i in (length of setFR)
    MaxBP_setFR← Maximum(setFR_Rises)
    IF MaxBP_setFR >= (varLowMagTh - bpBuffer)
        stormC_TierI←s*etFR*
    ELSE
        stormC_TierII←setFR
END LOOP

RETURN stormC_TierI & stormC_TierII

APPENDIX B

CANDIDATE STORM CLASSIFICATION INTO

PROFILES V, W_{half}, W AND Complex

Input: stormC_TierI,

Output: setV, setW_{half}, setW, setComplex

---

setV←empty, setW_{half}←empty, setW←empty, setComplex← empty
Step 1) Loop through *stormC_TierI* to find subsets of temporally clustered *Fall* or *Rise* events within a candidate storm. *BP_Fall* or *BP_Rise* events from several buoys are considered separate clusters if temporal distance between events is equal to or more than x hours (we set this value at 12 hours).
Step 2) Classify storms according the number of *BP_Fall* and *BP_Rise* subsets
FOR each i in (length of stormCTierI)
    IF (Number of subsets of *BP_Fall* AND *BP_Rise* events is 1)
        setV← stormC_TierI(i)
    ELSEIF (Number of subsets of *BP_Fall* AND *BP_Rise* events is 2)
        setW← stormC_TierI(i)
    ELSEIF (Number of subsets of *BP_Fall* is 1 AND *BP_Rise* events is 2) OR (Number of subsets of *BP_Fall* is 2 AND *BP_Rise* events is 1)
        setV← stormC_TierI(i)
    ELSE
        setComplex← strormC_TierI(i)
    END IF
END LOOP
RETURN setV, setW_{half}, setW, setComplex

APPENDIX C

SEGMENTATION OF CANDIDATE STORMS IN

Fall (F), Rise (R) and Fuzzy (Z) SEGMENTS

Input: strmC_TierI, str_AllBuoyCaps (String of all possible buoy tags in capital letters) i.e, 'ABCEFGHIJLMN'

Output: event_F_segm, event_R_segm, event_Z_segm

---

stormC_segm←empty, stormC_segmHWDVar←empty, stormC_segmLWDVar←empty, stormC_segmHATVar←empty, stormC_segmLATVar←empty, stormC_segmWS(Number of candidate storms) ←empty

Step 1) Fill *stormC_segm* with letter 'F', 'R' or 'Z', using SPS for *stormC_TierI i.e. SPS_TierI.* Also calculate variance of parameters such as wind direction, air temperature, wind speed and store in appropriate matrices.

```
FOR each i in (length of SPS_TierI)
    FOR each j in (length of SPS_TierI(i))
        IF SPS_TierI(i)(j) CONSISTS OF capital AND small case buoy tag
            stormC_segm(i)(j)← 'Z'
            stormC_segmHWDVar←circular_Variance(non-NAN Wind-direction
            observations for all buoys)
            stormC_ segmATVar←Variance(non-NAN Air-temperature
            observations for all buoys)
            stormC_ segmWSVar←Variance(non-NAN Wind speed observations for
            all buoys)
         ELSE IF SPS_TierI(i)(j) CONSISTS OF all capital letter buoy tags
            stormC_segm(i)(j) ←'F'
            stormC_segmHWDVar←circular_Variance(non-NAN Wind direction
                observations for all buoys)
            stormC_segmATVar←Variance(non-NAN Air-temperature
                observations for all buoys)
            stormC_segmWSVar←Variance(non-NAN Wind speed observations for
                all buoys)
        ELSE
            stormC_segm(i)(j) ←'R'
                stormC_segmHWDVar←circular_Variance(non-NAN Wind-
                direction observations for all buoys)
            stormC_segmATVar←Variance(non-NAN Air-temperature observations
                for all buoys)
                stormC_segmWSVar←Variance(non-NAN Wind speed observations
                    for all buoys)
        END IF
END LOOP
```

END LOOP

Step 2) Find timestamps of all observations with segment tags: *'F'* (for *fall*), *'R'* (for *rise*) and 'Z' (for *fuzzy*)
stormC_segmentEvents (# of stormC_segm) ←empty,*index_F, index_R, index_Z* ←empty
FOR each i in (length of SPS_TierI)
    FOR each j in (length of SPS_TierI(i))
        IF stormC_segm(i)(j) = 'F'
            index_F← Timestamp of SPS_TierI(j)
        ELSE IF stormC_segm(i)(j) = 'R'
            index_R ←Timestamp of SPS_TierI(j)
        ELSE IF stormC_segm(i)(j) = 'Z'
            index_Z← Timestamp of SPS_TierI(j)
        END IF
    END LOOP
END LOOP

Step 3) Find *start* and *end* times of segment *'F'*, '*R*' and '*Z*'
// COMMENT: For fall segment i.e., 'F'
event_S_start←*index_F(1)*
FOR each k in index_F
    IF index_F(k+1) – index_F(k) > = 2
        event_F_end←index_F(k)
        event_F_start←index_F(k+1)
    END IF
END LOOP
//Comment: Similarly, find event_R_start, event_R_end and event_Z_start, event_Z_end
event_F_segm← concatenation of *event_F_start* and *event_F_end*
//Comment: Similarly, concatenate event_F_segm, event_Z_segm

RETURN event_F_segm, event_R_segm, event_Z_segm

BIBLIOGRAPHY

Abbott, V., Black, J. B., & Smith, E. E. (1985). The representation of scripts in memory. *Journal of Memory and Language* , 179-199.

Agrawal, R., Psaila, G., Wimmers, E. L., & Zait, M. (1995). Querying shapes of histories. *Proceesings of the 21st International Conference on Very Large Databases*, (pp. 502-514).

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks* , 393-422.

Allan, J., Papka, R., & Lavrenko, V. (1998). On-line new event detection and tracking. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 37-45). Melbourne: ACM New York.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence* , 123-154.

Allen, J., & Ferguson, G. (1994). Actions and events in interval temporal logic. *Journal of Logic and Computation* , 531-579.

Antunes, C. M., & Oliveira, A. L. (2001). Temporal data mining: an overview. *Lecture Notes in Computer Science* , 1-15.

Audi, R. (1995). *The Cambridge dictionary of philosophy.* Cambridge University Press.

Bakshi, B. R. (1999). Multiscale analysis and modeling using wavelets. *Journal of Chemometrics* , 415-434.

Barker, R. G., & Wring, H. F. (1954). *Midwest and its children: The psychological ecology of an American town.* Evanston, IL: Peterson and Company.

Basseville, M., & Nikiforov, I. (1993). *Detection of abrupt changes- Theory and application.* Prentice-Hall Inc.

Batal, I., Sacchi, L., Bellazzi, R., & Hauskrecht, M. (2009). Multivariate time series classification with temporal abstractions. *Proceedings of the Twenty-Second International FLAIRS Conference.*

Beard, K., Deese, H., & Pettigrew, N. R. (2007). A framework for visualization and exploration of events. *Information Visualization* , 133-151.

Bittner, T., & Smith, B. (2003). *Formal ontologies for space and time*. Retrieved 01 19, 2011, from http://ontology.buffalo.edu/geo/sto.pdf

Bonnet, P., Gehrke, J., & Seshadri, P. (2001). Towards sensor database systems. *Mobile Data Management, Lecture Notes in Computer Science* , 3-14.

Bower, G. H., Black, J. B., & Turner, T. J. (1979). Scripts in memory for text. *Cognitive Psychology* , 177-220.

Brazel, A. J., & Nickling, W. G. (1986). The relationship of weather types to dust storm generation in Arizona (1965-1980). *Journal of Climatology* , 255-275.

Chakravarthy, S., Krishnaprasad, V., Anwar, E., & Kim, S. K. (1994). Composite events for active databases: semantics, contexts and detection. *Proceedings of the 20th International Conference on Very Large Data Bases*, (pp. 606-617).

Chen, J., & Jiang, J. (1998). Event-based spatio-temporal database design. *ISPRS Commission IV Symposium on GIS- Between Visions and Applications* , 105-109.

Claramunt, C., & Theriault, M. (1995). Managing Time in GIS: An Event-Oriented Approach. *Proceedings of the International Workshop on Temporal Databases: Recent Advances in Temporal Databases* (pp. 23-42). Springer-Verlag London, UK.

Davis, R. E., & Rogers, R. F. (1992). A synoptic climatology of severe storms in Virginia. *The Professional Geographer* , 319-332.

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time series databases. *International Conference on Management of Data* , 419-429.

Fawcett, T., & Provost, F. (1999). Activity monitoring: noticing interesting changes in behavior. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 53-62). ACM New York.

Fayyad, U., Piatetsky, G. S., & Padhraic, S. (1996). From data mining to knowledge discovery in databases. *AI Magazine* , 37.

Fonseca, F. T., Egenhofer, M. J., Agouris, P., & Camara, G. (2002). Using ontologies for intergrated geographic information systems. *Transactions in GIS* , 231-257.

GOMOOS. (2002). *Gulf of Maine Ocean Observation System*. Retrieved 01 20, 2011, from www.gomoos.com

Grenon, P., & Smith, B. (2004). SNAP and SPAN: Towards dynamic spatial ontology. *Spatial Cognition & Computation: An Interdisciplinary Journal*, (pp. 69-104).

Gruber, T. R. (1991). Ontolingua: A mechanism to support portlable ontologies. *KSL, Stanford Knowledge Systems Laboratory* , 91-66.

Gruninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. *Workshop on Basic Ontological Issues in Knowledge Sharing.* Montreal.

Guimaraes, G., & Ultsch, A. (1999). A method of temporal knowledge conversion. *Advances in Intelligent Data Analysis, Lecture Notes in Computer Science* , 369-380.

Guralnik, V., & Srivastava, J. (1999). Event detection from time series data. *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 33-42). ACM New York.

Heitjan, D. F., & Rubin, D. B. (1991). Ignorabity and coarse data. *The Annals of Statistics*, (pp. 2244-2253).

Hoppner, F. (2002). Discovery of core episodes from sequences. *Pattern Detection and Discovery* , 199-213.

Hoppner, F. (2003). *Knowledge discovery from sequential data.* Germany: PhD Thesis, Technical University Braunschweig.

Hoppner, F. (2002). Learning dependencies in multivariate time series. *Knowledge Discovery from Temporal and Spatial Data* .

Hoppner, F., & Klawonn, F. (2002). Finding informative rules in interval sequences. *Intelligent Data Analysis* , 237-255.

Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. *Procedings of the 6th annual International conference on Mobile computing and networking*, (pp. 56-67). Boston, MA.

Jiao, B., Son, S. H., & Stankovic, J. A. (2005). GEM: Generic event service middleware for wireless sensor networks. *International Conference on Networked Sensing Systems.*

Kapitanova, K., & Son, S. H. (2009). MEDAL A coMpact Event Description and Analysis Language for Wireless Sensor Networks. *Network Sensing Systems (INSS)*, (pp. 1-4). Pittsburgh, PA.

Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2003). Segmenting time series: A survey and novel approach. In L. Mark, K. Abraham, & B. Horst, *Data mining in time series databases* (pp. 1-22). Singapore: World Scientific.

Kim, J. (1969). Events and their descriptions: some considerations. In N. Rescher, *Essays in Honor of Carl G. Hempel* (p. 198).

Kulldorff, M. (1997). *A spatial scan statistic.* Taylor & Francis.

Lewis, D. (1973). Causation. *The Journal of Philosophy* , 568-569.

Li, S., Lin, Y., Son, S. H., Stankovic, J. A., & Wei, Y. (2004). Event Detection Services Using Data Service Middleware in Distributed Sensor Networks. *Telecommunication Systems* , 351-368.

Makridakis, S., & Wheelwright, S. (1989). *Forecasting methods for management.* New York: John Wiley & Sons.

Makridakis, S., Wheelwright, S., & McGee, V. (1983). *Forecasting.* John Wiley & Sons.

Mark, D. M., Smith, B., & Tversky, B. (1999). Ontology and geographic objects: An empirical study of cognitive categorization. *Spatial Information Theory, Cognitive and Computational Foundations of Geographic Information Science, COSIT'99* .

Michotte, A. (1963). *The perception of causality.* New York: Basic Books.

Miller, G. A., & Johnson-Laird, P. N. (1976). *Language and perception.* Harvard University Press.

Morchen, F. (2003). *Time series feature extraction for data mining using DWT and DFT.* Marburg, Germany: Department of Mathematics and Computer Science, Philipps-University Marburg.

Morchen, F. (2006). *Time Series Knowledge Mining.* Marburg Germany: PhD thesis, Philipps-University.

Morchen, F., & Ultsch, A. (2005). Discovering temporal knowledge in multivariate time series. In C. Weihs, & W. Gaul, *Classification- the Ubiquitous Challenge* (pp. 272-279). Springer Berlin Heidelberg.

Morchen, F., & Ultsch, A. (2007). Efficient mining of understandable patterns from multivariate interval time series. *Data Mining Knowledge Discovery* , 181-215.

Morchen, F., & Ultsch, A. (2005). Optimizing time series discretization for knowledge discovery. *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* , 660-665.

Muller, R. A. (1977). A synoptic climatology for environmental baseline analysis: New orleans . *Journal of Applied Meteorology and Climatology* , 20-33.

Murphy, R. (1999). Dempster-Shafer theory for sensor fusion in autonomous mobile robots. *IEEE Transactions on Robotics and Automation* , 197-205.

Nagel, E. (1979). *The structure of science: problems in the logic of scientific explanation.* New York: Harcourt Brace & World.

NASA. (2010, 12 17). *SWEET Ontologies*. Retrieved 01 20, 2011, from Semantic Web for Earth and Environmental Terminology: http://sweet.jpl.nasa.gov/

NCDC. (2011). *NCDC Storm Events*. Retrieved 01 20, 2011, from NOAA Satellite and Information Service: http://www4.ncdc.noaa.gov/cgi-win/wwcgi.dll?wwEvent~Storms

Neill, D. B. (2009). Expectation-based scan statistics for monitoring spatial time series data. *International Journal of Forecasting* , 498-517.

NWS. (2011). *National Weather Service Glossary*. Retrieved 01 20, 2011, from NOAA's National Weather Service: http://www.nws.noaa.gov/glossary/

NWSFO. (2007, 05 24). *Patriot's Day Storm of 2007*. Retrieved 01 20, 2011, from NOAA's National Weather Service Forecast Office: www.erh.noaa.gov/gyx/patriots_day_storm_2007.htm

Padmanabhan, B., & Tuzhilin, A. (1996). Pattern Discovery in Temporal Databases: A Temporal Logic Approach. *International Conference onKnowledge Discovery and Data Mining*, (pp. 351-354).

Pan, F., & Hobbs, J. (2005). Temporal aggregates in OWL-Time. *Proceedings of the 18th International Florida Artificial Intelligence Conference (FLAIRS)* , 560-565.

Pettigrew, N. R., Roesler, C. S., Neville, F., & Deese, H. E. (2008). An operational real-time ocean sensor network in Gulf of Maine. *Geosensor Networks, Lecture Notes in Computer Science* , 213-238.

Peuquet, D. (2001). Making space for time issues in space-time data representation. *Geoinformatica* , 11-32.

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002, 04). Data quality assessment. *Communications of ACM- Supporting community and building social capital* , pp. 211-218.

Povinelli, R. J. (2001). Indentifying temporal patterns for characterization and prediction of financial time series events. In J. F. Roddick, & K. Hornsby, *Time Series Data Mining* (pp. 46-61). Springer-Verlag Berlin Heidelberg.

Quine, W. V. (1985). *Events and reification.*

Raimond, Y., & Abdallah, S. (2007, 10 25). *The Event Ontology*. Retrieved 01 20, 2011, from SourceForge: http://motools.sourceforge.net/event/event.html

Rosch, E. (1978). *Cognition and Categorization.* Lawrence Erlbaum Associates.

Ross, W. D. (1924). *Aristotle-Metaphysics. Text and Commentary.* Oxford University Press.

Rubin, D. B. (1976). Inference and missing data. *Biometrika* (pp. 581-592). Biometrika Trust.

Russell, B. (1923). Vagueness. *The Australian Journal of Psychology and Philosophy* , 84-92.

Schafer, J., & Graham, J. (2002). Missing data: Our view of the state of the art. *American Psychological Association* , 147-177.

Shahar, Y. (1997). A framework for knowledge-based temporal abstraction. *Artificial Intelligence* , 79-133.

Smith, B., & Mark, D. M. (1998). Ontology and geographic kinds. *Proceedings, International Symposium on Spatial Data Handling.* Vancouver, Canada.

Sowa, J. F. (1999). *Knowledge representation: Logical, philosophical, and computational foundations.* MIT Press.

Stann, F., & Heidemann, J. (2003). RMST: Reliable data transport in sensor networks. *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Application*, (pp. 102-112).

Swartout, B., Patil, R., Knight, K., & Russ, T. (1997). *Toward distributed use of large-scale ontologies.* AAAI Technical Report SS-97-06.

Tufte, E. R. (1983). *The visual display of quantitative information.* Graphics Press.

Tversky, B., & Hemenway, K. (1984). Objects, parts and categories. *Journal of Experimental Psychology* , 169-193.

Ultsch, A. (1999). A method for temporal knowledge conversion. *Advances in Intelligent Data Analysis* , 369-380.

Ultsch, A. (2004). *Unification-based Temporal Grammar.* Marburg, Germany: Department of Mathematics and Computer Science, Philipps-University.

Wallinga, J. P., Pettigrew, N. R., & Irish, J. D. (2003). The GoMOOS moored buoy design. *Procedings of OCEANS* , 2596-2599.

Weigend, A. S., & Gershenfeld, N. A. (1994). Time series prediction: Forecasting the future and understanding the past. (p. Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis). Santa Fe: Assison-Wesley.

Worboys, M., & Hornsby, K. (2004). From objects to events: GEM, the Geospatial Event Model. *Geographic Information Science* , 327-343.

Yin, J., & Gaber, M. M. (2008). Clustering distributed time series in sensor networks. *International Conference on Data Mining*, (pp. 678-687). Pisa.

Zacks, J. M., & Tversky, B. (2001). Event structure in perception and conception. *Psychological Bulletin* , 3-21.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control* , 338-353.

BIOGRAPHY OF AUTHOR

Avinash Rude was born in Yavatmal district in Maharashtra, India. He attended the Pravara Public School in Ahmednagar, India; graduating with a High School diploma in 1999. For his undergraduate degree in Civil Engineering, he went to National Institute of Technology, Hamirpur in Himachal Pradesh until 2004. Avinash worked for the National Environmental Engineering Research Institute, Nagpur as a Project Assistant between 2004 and 2007. During this period, he conducted GIS mapping for several research projects. He is a candidate for the Master of Science degree in Spatial Information Science and Engineering from The University of Maine in May, 2011.