Information Sciences
Manuscript Draft

Manuscript Number:

Title: *Detecting and Tracking Region Outliers in Meteorological Data Sequences*

Article Type: *Full Length Article*

Section/Category:

Keywords: *Spatial Outlier Detection, Wavelet, Image Segmentation, Data Sequence, Meteorological Data*

Manuscript Region of Origin:

Abstract:

Detecting spatial outliers can help identify significant events in spatial data sequences. In the field of meteorological data processing, spatial outliers are frequently associated with natural disasters such as tornadoes and hurricanes. Previous studies on spatial outliers mainly focus on identifying single location point over a static data frame. In this paper, we propose and implement a systematic approach to detect and track region-outliers in a sequence of meteorological data frames. First, a wavelet transformation such as the Mexican Hat or Morlet is used to sharpen and enhance the data variation. Second, an image segmentation method, λ-connected segmentation, is employed to identify the outlier regions. Finally, a regression technique is applied to track the center movement of the outlier regions for consecutive frames. In addition, we conducted experimental evaluations using real-world meteorological data and events, e.g., hurricane Isabel, to demonstrate the effectiveness of our proposed approach.

Authors: *Chang-Tien Lu, Yufeng Kou, Jiang Zhao, and Li Chen*

Correspondence Author: *Chang-Tien Lu*
Address: *Department of Computer Science*
Virginia Polytechnic Institute and State University
7054 Haycock Road, Falls Church, VA 22043
Email: *ctlu@vt.edu*
Phone: *703-538-8373*
Fax: *703-538-8348*

Detecting and Tracking Region Outliers in Meteorological Data

Chang-Tien Lu [a,*] Yufeng Kou [a] Jiang Zhao [b] Li Chen [c]

[a] *Dept. of Computer Science*
Virginia Polytechnic Institute and State University
7054 Haycock Road, Falls Church, VA 22043

[b] *QSS Group, Inc.*
4500 Forbes Blvd
Lanham, MD 20706

[c] *Dept. of Electrical Engineering and Computer Science*
The University of the District of Columbia
4200 Connecticut Avenue NW
Washington, DC 20008

Abstract

Detecting spatial outliers can help identify significant events in spatial data sequences. In the field of meteorological data processing, spatial outliers are frequently associated with natural disasters such as tornadoes and hurricanes. Previous studies on spatial outliers mainly focus on identifying single location point over a static data frame. In this paper, we propose and implement a systematic approach to detect and track region-outliers in a sequence of meteorological data frames. First, a wavelet transformation such as the Mexican Hat or Morlet is used to sharpen and enhance the data variation. Second, an image segmentation method, λ-connected segmentation, is employed to identify the outlier regions. Finally, a regression technique is applied to track the center movement of the outlier regions for consecutive frames. In addition, we conducted experimental evaluations using real-world meteorological data and events, e.g., hurricane Isabel, to demonstrate the effectiveness of our proposed approach.

Key words: Spatial Outlier Detection, Wavelet, Image Segmentation, Data Sequence, Meteorological Data

[*] Corresponding author.
Email addresses: `ctlu@vt.edu` (Chang-Tien Lu), `ykou@vt.edu` (Yufeng Kou), `jiang.zhao@noaa.gov` (Jiang Zhao), `lchen@udc.edu` (Li Chen).

1 Introduction

Due to the ever-increasing volume of spatial data, spatial data mining has become an important research area over the past decade [17, 39]. From satellite observation systems to urban planning, geography related spatial data are widely used. Other types of spatial data, such as medical images and gene maps, have received a significant amount of attention from medical professionals and researchers. As defined in [21], spatial data mining is the process of discovering hidden but valuable patterns from large spatial data sets. Similar to traditional data mining, spatial data mining techniques can be classified into four categories: classification, clustering, trend analysis, and outlier detection. The challenges regarding spatial data mining have arisen from the following issues. First, classical data mining is designed to process numerical and categorical data, whereas spatial data have more complex structures that contain extended objects such as points, lines, and polygons. Second, classical data mining treats each input independently from other inputs, while spatial patterns often exhibit continuity and high autocorrelation with nearby samples.

As the most widely-used spatial data, geographic data not only relates to three dimensional volumes, but also contains temporal information. Together, they form spatial data sequences. In recent years, spatio-temporal data has attracted the attention from various domains, such as computer scientists, geographers, environmental researchers, resource managers, and biologists. This data contains complex structures, arrives continuously, evolves over time, and needs to be processed in real time. However, unlike the video stream, the frame sampling period is in minutes, and it has no strict restrictions on processing speed. Several recent studies have been conducted to develop specific data mining techniques for detecting useful patterns from continuous data streams [11, 13, 20]. Because these techniques are not specifically designed for processing spatial data, they may not be effectively utilized by geospatial applications. Intensive research is in great demand for extracting knowledge from spatio-temporal data to help predict the trends of spatial patterns accurately [10, 26].

Outlier detection is a process to identify the objects which differ from the rest of the data sets [4, 19]. In the research on the atmospheric sciences, huge amounts of spatial data have been continuously collected from both observation and simulation modelling. Discovering useful patterns from these data, especially spatial outliers and their movements, will have great practical value and will help weather forecasting, environmental monitoring, and climate analyzing. In the meteorological data, spatial outliers are the observations that are inconsistent with their surrounding neighbors. Spatial outliers or anomalies are often associated with severe weather events, such as tornadoes and hurricanes. These events usually do not happen at a single location but over an extended region. That is to say, they are usually two dimensional region spatial outliers. Furthermore, the spatio-temporal changes in these regions are frequently associated with the variations of weather phenomena and climate patterns.

To automatically extract the outlier regions is a crucial issue. Typically, the methods used to address this problem rely on the image segmentation and pattern recognition [14, 22]. Image segmentation divides an image into constituent regions. This technique has been widely used in several practical applications, such as military satellite image analysis. Wavelet transformation is an important tool for digital signal processing, image processing, and data mining. Wavelet transformation can represent data in a hierarchical structure with multiple resolutions from gross to detail. In addition, it can provide the time and frequency information simultaneously, thus rendering a time-frequency representation of the signal. Another advantageous property of wavelet transformation is that it can distinctly capture the differences between a data item and its neighboring items [25].

In this paper, we propose and implement a systematic approach to detect and track region outliers in a sequence of meteorological data frames. First, a wavelet transformation such as Mexican Hat or Morlet is used to sharpen and enhance the data variation. Second, an image segmentation method, λ-connected segmentation, is applied to identify the outlier regions. Finally, a regression technique is used to track the center movement of the outlier regions for consecutive frames. In addition, we conducted experimental evaluations using real-world meteorological data, such as the data collected from hurricane Isabel, to validate the effectiveness of the proposed algorithms. This paper is organized as follows. Section 2 provides a literature survey; In Section 3, we discuss the problems and propose various approaches; The tools and algorithms are introduced in Section 4; Section 5 describes the real meteorological data and analyzes the experimental results; Finally, we summarize our work and discuss future research directions in section 6.

2 Background and Related Work

This section provides related research work in spatial outlier detection, image segmentation, spatio-temporal data sequence mining, and meteorological pattern identification.

Numerous studies have been conducted to identify outliers from large spatial data sets. The existing spatial outlier detection methods can generally be grouped into two categories, namely graphic approaches and quantitative tests. Graphic approaches are based on visualization of spatial data which highlights spatial outliers. Examples include variogram clouds and pocket plots [18,34]. Quantitative methods, e.g., Scatterplot [17] and Moran scatterplot [28], provide tests to distinguish spatial outliers from the remainder of the data set. Shekhar *et al.* introduced a method for detecting spatial outliers in graph data [40]. An outlier may have negative impact on its neighbors when its attribute value is much higher/lower than the average of its neighbors. Two iterative methods and one median-based approach were proposed in [27] to address this problem. Most of the existing spatial outlier detection methods are designed for point data. However, outliers may exist in other spatial forms such as lines and regions.

Image segmentation is to partition an image into different components or objects. It is a key procedure for image preprocessing, object detection, and movement tracking. The existing image segmentation approaches can be categorized into five groups. The **first** and the most popular one, called threshold segmentation, is to give a threshold or clip-level to transform a grey-scale image into a binary image. Cheriet *et al.* proposed an approach to explore an optimal threshold for minimizing the ratio of between-segments variance and the total variance [8]. Another approach, called the maximum entropy approach, is to define a threshold based on comparing the entropies of the segmented image [32]. The **second** method, proposed by Rosenfeld, treats an image as a 2D fuzzy set and uses α-cut to develop a fuzzy connectivity [37]. A variation of this fuzzy connectedness is to measure two pixels to evaluate if they are "fuzzy connected." A pixel set is called λ-connected if for any two points there is a path that is λ-connected, where λ is a fuzzy value between 0 and 1 [6]. Both threshold segmentation and λ-connected segmentation can be executed in linear time. The **third** category is called split-and-merge segmentation [14] or quad-tree segmentation. This method splits an image into four blocks or parts and checks if each part is homogenous. If not, the splitting process will be repeated; otherwise it starts to merge. This method is accurate for complex image segmentation. However, it is complicated to implement and costs more time in computation ($O(nlogn)$). The **fourth** category is related to K-mean or fuzzy c-mean. It is a standard classification method that is often applied in image segmentation [7]. This method classifies the pixels into different clusters to reach minimum total "error" where the "error" means the distance from a pixel to the center of its own cluster. This method may produce very convincing result. Nevertheless, it employs an iterative process to reach convergence. The **fifth** method is called the Mumford-Shah method that uses the variational principal [31]. This method considers three factors in segmentation: the length of edges of total segments, the unevenness of the image without edges, and the error between the original image and the proposed segmented images. When the three weighted factors reach the minimal, this iterative segmentation process stops. Chan and Vese employed level-sets to confine the search of segment edges based on contour boundaries [5]. Their approach is more efficient than Mumford-Shah method. However, level-sets may limit its reflexibility of the original method.

For meteorological data, the feature changes are usually not sharp to form clear edges. Therefore, direct application of image segmentation can not be utilized effectively to determine the coverage of the outlier regions. To distinguish the variation of feature gradient, wavelet techniques can be applied to the original spatial data before performing image segmentation [43]. Wavelet has many favorable properties, such as supporting multi-resolution and frequency localization, which make it a widely used tool for digital signal processing and image processing [12, 29]. In recent years, wavelet transformation techniques have extended their application to data mining areas, including clustering [38], classification [24], and data visualization [30].

A copious amount of attention has been devoted to identifying and tracking useful patterns from continuous data sequences. These patterns include cluster, evolution, deviation, and anomaly. **(1) Cluster**: Guha *et al.* proposed a divide-and-conquer approach for continuous data clustering [16]. Li *et al.* explored a clustering technique on moving objects to catch moving patterns of a set of similar data points [26]. **(2) Evolution**: By extending a spatio-temporal data model, Tripod [15], Djafri *et al.* developed a general approach to characterize the evolution queries in a spatio-temporal database [10]. Charu presented a framework to detect changes and identify useful trends in evolving data sequences [2]. Giannella *et al.* designed an algorithm to maintain frequent patterns under a tilted-time window framework in order to answer time-sensitive queries [13]. **(3) Deviation**: Palpanas *et al.* utilized kernel density estimators for online deviation detection in continuous data sequences [33]. **(4) Anomaly**: A neighborhood-based anomaly detection approach was proposed by Adam *et al.* for high dimensional spatio-temporal sensor data streams [1].

With the explosion in the amount of meteorological data, extensive research has been conducted to assist meteorologists in accurately identifying patterns of severe weather events. Several approaches, including fuzzy clustering [3], neural networks [9], genetic algorithms [23], and support vector machine [36, 41], have been proposed to classify storm cells. Peters *et al.* presented a rough-set-based method capable of classifying four types of storm events: hail, heavy rain, tornado and wind [35].

3 Problem and Approach

In the Earth's atmosphere, anomalies emerge at different spatial scales and may appear in different shapes, which makes a challenging task to detect outliers from continuous meteorological data sequences. Figure 1 shows an image of the water vapor distribution over the east coast of the U.S., the Atlantic Ocean, and the Gulf of Mexico. The color intensity of each region reflects its water vapor content. As can be seen, there is a "hot spot" located in the left portion of the image ($28^o N$, $90^o W$), indicating a hurricane in the Gulf of Mexico. This outlier spot is not a single point but a group of the points, a region. This region has a much higher water vapor content than its surrounding neighbors. Thus, *a region outlier is a group of adjacent points whose features are inconsistent with those of their surrounding neighbors*. The red-colored hot spot, a hurricane, in Figure 1 is a region outlier. Region outliers are determined by domain experts based on the pre-defined threshold. Our challenge is to design an efficient and practical approach to automatically detect region outliers, which could be in irregular shapes, from spatial data sequences. In real applications, such approaches can help identify spatial anomalies such as hurricanes, tornadoes, thunder storms, and other severe weather events from the observation data.

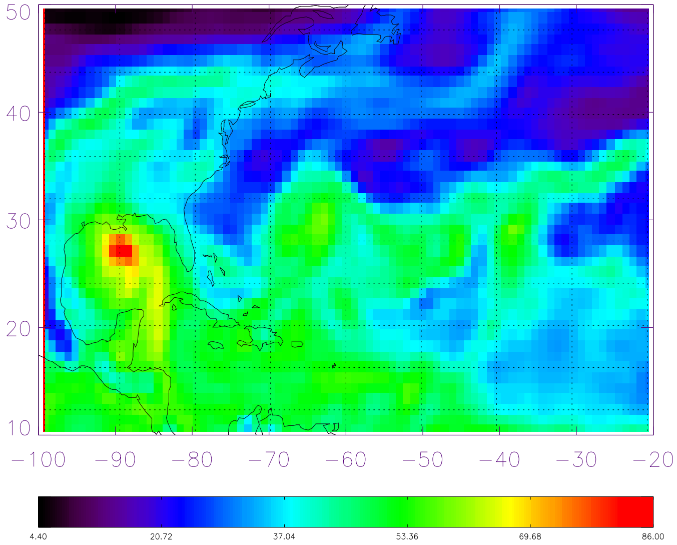

Fig. 1. A region outlier (hurricane) in meteorological data.

In order to accurately extract region outliers, it is preferable to decompose the original observations into different spatial scales to reduce the complexity and centralize the target object. Wavelet transformation provides such a capability with its multi-resolution characteristics. First, wavelet tranformation can be used to decompose the original spatial variation of the data into different scales, allowing users to focus on the scale of interest and identify the potential outliers at that scale. Second, the localization of variation in the frequency domain is useful in determining the spatial location of outliers.

In this application, we will apply wavelet transformation in the real spatial domain, then analyze the transformed data for a particular set of scales. As spatial outliers are usually small in size compared with the environment, relatively small scales will be selected for hurricanes and tornadoes. The wavelet transformation power indicates the strength of the variation and the localization of any high values reveals the places where anomalies exist. In the next section, we will discuss the wavelet transformation functions used in our application.

Image segmentation can be employed to extract spatial regions within which the meteorological characteristics is similar. The segmentation algorithm needs to perform fast in order to process sequential frames and even high-speed image streams. For example, the selected algorithms should not scan the whole frame multiple times. Ideally, we shall scan the original frame only once or even only scan a part of it. With $O(nlogn)$ time complexity, split-and-merge method will not be practical for this purpose. K-mean and fuzzy c-mean, as well as Mumford-Shah method need more time because they require numerous iterations. Thus, for achieving satisfactory speed, threshold method and λ-connected method are the only two options since they both have linear time complexity. Threshold segmentation seems to be the simplest solution. However, when an image needs multiple thresholds, the determination of threshold values will be difficult and time-consuming. The advantage of λ-connectedness approach is that it can determine segments in different intensity levels without calculating different thresholds or clip-level values. Based on the above reasons, we choose λ-connectedness approach to segment the meteorological data.

Our goal is to identify the largest outlier region in which the value of each pixel is above a reasonably predefined threshold. If we select the threshold method, the image is translated into a binary image based on a threshold, then the breadth-first search algorithm is used to label each connected component and select the largest one. The major advantage of this approach is that the process is easy to perform. Its disadvantage, however, is that it does not tolerate any noise. Using a λ-connected search algorithm [6], we can start with any pixel above a threshold, and find all neighbors that have similar values by comparing them with the starting pixel. This method is a generalized version of the former one. The details of the λ-connected search are described as follows.

An image is a mapping from a two dimensional space to the real space R. Without loss of generality, let Σ_2 be the two-dimensional grid space, the 2D digital space. A digital image can be represented by a function: $f : \Sigma_2 \rightarrow [0, 1]$. Let $p = (x, y), q = (u, v) \in \Sigma_2$, p, q are said to be adjacent if $\max\{\|x - u\|, \|y - v\|\} \leq 1$. (A pixel, i.e. picture element, is a couple of $(p, f(p))$.) So, if p, q are adjacent and $f(p), f(q)$ have only a “little” difference, then pixels $(p, f(p))$ and $(q, f(q))$ are said to be λ-adjacent. If there is a point r that is adjacent to q and $(q, f(q))$, $(r, f(r))$ are λ-adjacent, then $(p, f(p)), (r, f(r))$ are said to be λ-connected. Similarly, we can define the λ-connectness along a path of pixels.

Mathematically, let (Σ_2, f) be a digital image. If p and q are adjacent, we can define a measure called “neighbor-connectivity” as given below:

$$\alpha_f(p,q) = \begin{cases} 1 - \|f(p) - f(q)\|/H & \text{if } p\text{, } q \text{ adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $H = \max\{f(x)|x \in \Sigma_2\}$.

Let $x_1, x_2, ..., x_{n-1}, x_n$ be a simple path. The path-connectivity β of a path $\pi =$

$\pi(x_1, x_n) = \{x_1, x_2, ..., x_n\}$ is defined as

$$\beta_f(\pi(x_1, x_n)) = \min\{\alpha_f(x_i, x_{i+1}) | i = 1, ..., n-1\} \tag{2}$$

or

$$\beta_f(\pi(x_1, x_n)) = \prod\{\alpha_f(x_i, x_{i+1}) | i = 1, ..., n-1\} \tag{3}$$

Finally, the degree of connectedness (connectivity) of two vertices x, y with respect to ρ is defined as:

$$C_f(x, y) = \max\{\beta_f(\pi(x, y)) | \pi \text{ is a (simple) path.}\} \tag{4}$$

For a given $\lambda \in [0, 1]$, point $p = (x, f(x))$ and $q = (y, f(y))$ are determined to be λ-connected if $C_f(x, y) \geq \lambda$.

If equation (2) applies, λ-connectedness is reflexive, symmetric, and transitive. Thus, it is an equivalence relation. If equation (3) is used, λ-connectedness is reflexive and symmetric. Therefore, it is a similarity relation.

4 Algorithms Design

In this section, we first describe a wavelet transformation on image data. Second, we design a segmentation algorithm to obtain the largest connected region whose wavelet power is above background. Third, after the center point and boundary of the region are stored, linear regression will be employed to construct the approximate trajectory of the moving region in consecutive frames. The existence of some disturbances may introduce incorrect outlier regions. Regression can help remove these “noise” center points and obtain accurate trajectory.

4.1 Wavelet Transformation

Wavelet transformation is a practical technique in signal analysis and image processing. Wavelet transformation possesses several attractive features: (1)**Multiresolution**: wavelet transformation examines the signal at different frequencies with different resolutions. That is to say, it uses a wider window for low frequency and a narrower window for high frequency. This feature especially works well for signals whose high frequency components have short durations and low frequency components have long durations. Thus, wavelet transformation is an effective tool with which to filter the signal and focus on certain scales. (2)**Localization of the**

frequency: In Fourier transformation, the frequency domain has no localization information. Thus, if the frequency changes with time in the signal, it is hard to distinguish which frequency occurs within which time range, although all the frequencies may be detected. In the real world, signals are usually complicated and are non-stationary. If we want to know exact information for a variation, such as the frequency and the location of a certain variation or the strength of the variation at a certain location, wavelet transformation offers advantages over Fourier transforms.

In this paper, we use continuous wavelet transformation. For a wavelet function $\Psi(t)$, the continuous wavelet transformation of a discrete signal $X_i(i = 0, \ldots, N-1)$ is defined as the convolution of X with scaled and translated Ψ:

$$W(n, s) = \sum_{i=0}^{N-1} x(i)\Psi^*[\tfrac{(i-n)\delta t}{s}]$$

where (*) indicates the complex conjugate, n is the localization of the wavelet transformation and s is the scale. The wavelet transformation can also be inversely transformed to (or used to reconstruct) the original data set :

$$x_i = \tfrac{\delta j \delta t^{1/2}}{C_\delta \Psi_0(0)} \sum_{j=0}^{J} \tfrac{RealW(n,s_j)}{s_j^{1/2}}$$

Where C_δ is a constant for each wavelet function; $\Psi_0$is the normalized wavelet function; and J is the maximum scale index, which will be explained later. For more details of the wavelet transformation method, please refer to [42].

Here, we may not include all scales of the wavelet transformation into the reconstruction in order to filter out the variations of no interest, and the reconstructed data will be composed based on the scales that are of interest. For example, if the low frequency range of the variations in the data set is concerned, a low pass data set may be reconstructed to filter out the high frequency variations and make low frequency variations more visible. Many functions can be used as the base or mother function for wavelet transformation. We use two of the most widely used bases: the Mexican hat base and the Morlet base. The base function for a Morlet wavelet is:

$$\Psi_0(\eta) = \pi^{-1/4} e^{\omega_0 \eta} e^{-\eta^2/2}$$

The Mexican hat function is:

$$\Psi_0(\eta) = \tfrac{(-1)}{\sqrt{\Gamma(5/2)}} \tfrac{d^2}{d\eta^2}(e^{-\eta^2/2})$$

When performing the wavelet transformation, the scales are selected by $S_0 {*} 2^{j/2}(j = 0, 1, \ldots, J)$, where J is the maximum scale index which satisfies: $J \leq 2\log_2(\frac{N}{2})$, where N is the length of the signal, in this case $S_0 = 2\delta x$, $N = 360$. We use j as the scale index; Scale 2 means the real scale is $S_0 2^{0.5*2} = 4$. Tables 1 and 2 provide the relationship between scale index, real scale, and the corresponding period of the Fourier transform (here, since we are performing wavelet transformation on the

index	0	1	2	3	4
scale	2	2.83	4	5.65	8
period	7.95	11.23	15.9	22.47	31.79

Table 1
Scale Table for Mexican Hat Wavelet

index	1	2	3	...	6	7	8
scale	2.83	4	5.65	...	16	22.6	32
period	2.92	4.13	5.84	...	16.52	23.4	33.05

Table 2
Scale Table for Morlet Wavelet

spatial domain, it is in fact the wavelength of the spatial variation) for the Mexican hat and Morlet wavelets. From the tables, it can be seen that as the scale grows, the period (or wavelength) of the real object the wavelet focuses on also grows. However, the growth rates are different for the two wavelets. For the Morlet wavelet, the period grows slower than it does for the Mexican hat wavelet. Thus, the Morlet wavelet has a better frequency resolution than the Mexican hat wavelet. This also implies that Morlet has a poorer localization resolution.

The Morlet wavelet is a complex wavelet and the Mexican hat wavelet is a real wavelet. The Mexican hat model captures both the positive and negative variation as separate peaks in wavelet power. The Morlet wavelet power combines both positive and negative peaks into a single broad peak [42]. Figures 2 and 3 are examples of the two wavelet transformations. Figure 2(a) is the original data water vapor distribution along a particular latitude. Figures 2(b) and (c) show the wavelet transformation power at two different scales. Figure 3 uses the Morlet wavelet and higher scale indices. From Figures 2 and 3, we can see that the power of wavelet transformation can depict the distribution or localization of the variation at certain scales. The Mexican hat wavelet provides a better localization (spatial resolution), therefore we will mostly use the Mexican hat wavelet to perform the analysis.

4.2 Detection Algorithms

The proposed algorithm has two major functionalities: detecting a sequence of region outliers in consecutive frames and tracking their movements. First, a wavelet transformation is performed on the image data to identify regions with prominent spatial variations at certain scales. Then segmentation is employed to extract the largest outlier region and trace its trajectory. The algorithm is designed based on the following assumptions. First, CPU speed is capable of processing at least a

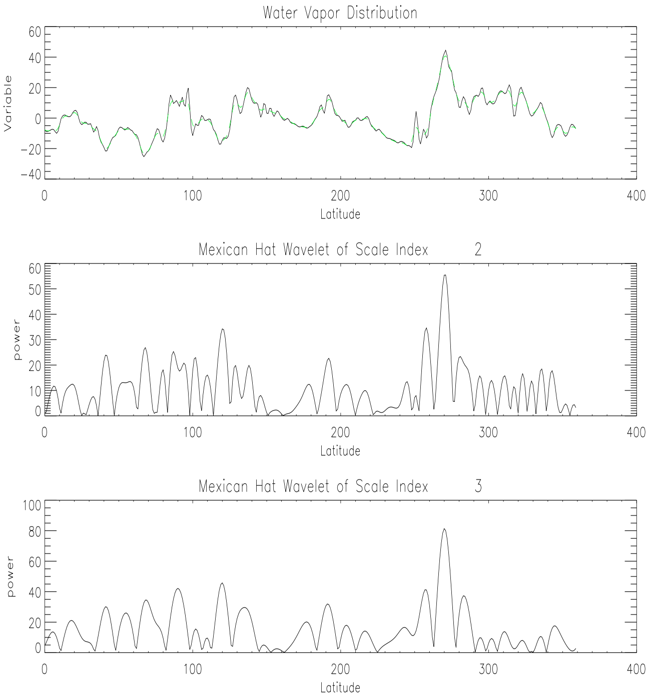

Fig. 2. A sample output of the Mexican hat wavelet (a: top, b: center, c: bottom).

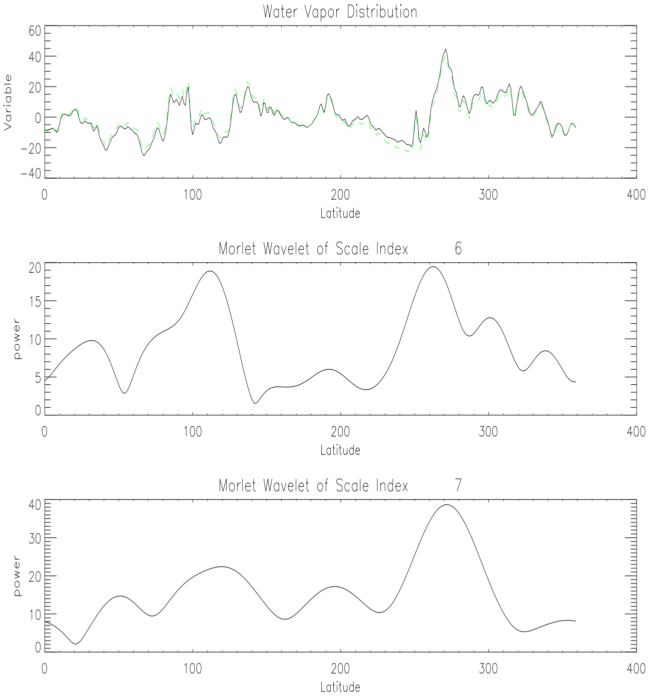

Fig. 3. A sample output of Morlet wavelet (a:top, b: center, c:bottom).

number k of data windows ($k \geq 1$). This means that the algorithm can process the continuous data window by window. The size of the window can be adjusted according to the arrival speed of the data sequence. Second, the data arrive in a specific sequence, for example, in the order of latitude or longitude. The arriving data element is thus spatially adjacent to the previous data element.

The primary algorithm is *Main*, which invokes other sub-algorithms, including *WaveletAnalysis*, *Segmentation*, and *Trajectory*. The input of algorithm *Main* includes a sequence of continuously arriving data DS, a set of selected scales S for wavelet transformation, a threshold θ, a similarity level λ for segmentation, and the trajectory T of the outlier region in previous frames. The output is the largest outlier region O_r for each image frame and its updated trajectory T.

In algorithm *Main*, firstly, a set of scales of interest is determined by domain experts. The continuous and unbounded data sequence DS will be processed in the unit of window. The window size will be determined by the size of each data item and the memory capacity. We designate each window representing an integral view of global meteorological data (180 degree by 360 degree) as one time frame. From the I/O buffer, a sequence of data elements are fetched and stored in window W. Then algorithm *WaveletAnalysis* is performed on W. $wDomain$ is the domain of wavelet power values transformed from data window W. Next, algorithm *Segmentation* is employed to extract outlier regions, which are connected components with wavelet power values above a predefined threshold θ. In particular, we focus on the largest connected region whose wavelet power values exceed the threshold, that is to say, only one region outlier will be detected. Finally, the boundary and center point of the outlier region can be calculated in order to trace the region movement. Trajectory T will be recalculated and updated once a new region is added.

In fact, identifying the moving outlier region does not need to process the whole frame. Apparently, the locations of the outlier region in adjacent frames are not likely to change dramatically. Thus, based on the region location in the previous frame, function $getPredictedArea()$ can define the predicted area Σ_p, an approximate rectangle which contains all the possible positions of the moving region but much smaller than the whole image $wDomain$. Instead of processing $wDomain$, we can obtain the outlier region by applying image segmentation to Σ_p. In this way, the cost of region detection can be significantly reduced. The center of Σ_p can be obtained by considering both the region center in the previous frame and its moving speed. As for the first several frames, Σ_p is set to be $wDomain$, that is to say, the whole image will be processed for segmentation. The utilization of predicted area can make the segmentation process four times faster if its size is a quarter of the original frame. However, the area can not be too small in order to protect the quality of search.

The detail of the three sub-algorithms are discussed below.

Algorithm *WaveletAnalysis* is designed to transform the source image data into the wavelet domain. The input of the algorithm is a sequence of data points W and a set of selected scales S. The output is wavelet power value for every point in W.

```
Algorithm : Main
 Input:
    $DS$ is a data sequence
    $S$ is a set of selected scales;
    $\theta$ is the threshold used for segmentation;
    $\lambda$ is the similarity level for segmentation;
    $T$ is the trajectory of the outlier region in previous frames;
 Output:
    $O_r$ is the set of points in the outlier region
    $T$ is the trajectory after appending the outlier region in current frame

 $T$ = $\phi$;
 /* continuously process the sequence window by window */
 while (true) {
    /* get a window of data from the sequence */
    W = getWinFromBuf($DS$);
    /*Call algorithm WaveletAnalysis to process current window*/
    wDomain = WaveletAnalysis(W,$S$);
    /* Define the predicted area to speed the image segmentation */
    $\Sigma_p$ = getPredictedArea($wDomain$, $T$);
    /* Call algorithm Segmentation to obtain largest region*/
    $O_r$ = Segmentation($\Sigma_p$, $\theta$, $\lambda$);
    /* Call algorithm Trajectory to track movement*/
    $T$ = Trajectory($T$, $O_r$);
    /* output the detected region and its moving trajectory*/
    Output($O_r$, $T$); }
```

Performing the wavelet transformation in different scales can help identify patterns with different sizes and filter noises. The algorithm first extracts the boundary of W. α_1 denotes the beginning latitude(or longitude) and α_n denotes the ending latitude(or longtitude) of the current window. Note that for meteorological data, the wavelet transformation will be performed along lines of latitude. We will discuss the justification for this in the experimental section of this paper.

Algorithm *Segmentation* aims to extract the largest connected region above threshold θ. It contains three input parameters: Σ, θ, and λ. Σ denotes a set of data points to be segmented; θ is a threshold to filter-out unwanted points (points whose values are less than θ will not be processed); λ is the similarity level. The value of θ is determined by domain experts. Ordinarily, we will designate it as the 75 percent of the difference between the maximum value and the minimum value of the data set. The output is the largest connected component in the data set, consisting of points with values greater than θ and similarity levels greater than λ. First, the algorithm picks a point p_0 from Σ whose value is greater than θ and is not labelled as ‘*’, which means “not processed.” Then p_0 is added into $QUEUE$. For each point in this $QUEUE$, its “unprocessed” neighboring points will be examined to check if

Algorithm : WaveletAnalysis

```
Input:
    $W$ is a data window from the sequence;
    $S$ is a set of selected scales;
Output:
    $wDomain$ is the wavelet power of the data window
/* get the minimum latitude(or longitude) of current window */
$\alpha_1$ = getMinBound(W);
/* get the maximum latitude(or longitude) of current window */
$\alpha_n$ = getMaxBound(W);
/*wavelet transformation along all latitudes(or longitudes)*/
for(i=$\alpha_1$;i $\leq \alpha_n$ ;i++) {
    wDomain = WaveletTransform(W,$S$,i); }
/* output the wavelet power of this data window*/
Output($wDomain$);
```

they have a similarity level greater than λ. If the condition holds, the corresponding neighboring point will be stored into $QUEUE$ and marked as "processed." Repeating the "marking" process for all the points in the $QUEUE$, we can obtain a result set S', containing the connected part of Σ. Next, the number of points in S and S' will be compared. If S' is larger than S, S will be replaced with S', ensuring that S maintains the largest component discovered heretofore. The while loop repeats until there is no "unprocessed" point with a value greater than θ. Finally, S is returned as the largest component discovered by the algorithm.

The objective of algorithm *Trajectory* is to track the moving direction and speed of a certain region, and to validate the correctness of the current detected region. The input parameters are the previously recorded trajectory T, the newly detected region R, and the number K of recent center points in T. The data structure of T includes the time, center, moving speed, and boundary of previous K regions. The detected region R is from the output of *Segmentation* algorithm. It is possible for a region to be erroneously detected by algorithm *Segmentation* due to errors in raw data or an inappropriate segmentation threshold. Therefore, a verification function is needed in order to determine the correctness of R based on the trajectory of the previous K regions. In the algorithm, first the boundary point B is extracted and the center C of the region R is computed. Then, a verification procedure is performed to compare C with the statistics of the past K center points along the trajectory. The mean μ and standard deviation σ of the past K center points are calculated. If C is located within 2σ from μ, R is considered as a valid region and C is appended to the trajectory T. Otherwise, R is flagged as a "noise" point that will be discarded. Moreover, the moving speed and direction of the region center can be obtained from two valid consecutive center points. Finally, the new trajectory T will be updated and can be stored in permanent storage for a specified period of time.

Algorithm: Segmentation

```
Input:
    $\Sigma$ : Set of data points
    $\theta$ : Threshold for the clip level
    $\lambda$: Similarity level
Output:
    $S$: the largest connected component with value above $\theta$

$\Sigma$ = $\emptyset$; {
while ($\Sigma$ contains unlabelled points)
    $p_0$ = pickOneUnLabeledPoint($\Sigma$, $\theta$);
    $L(p_0)$ = '*'; /*labeling $p_0$ as processed*/
    /*insert $p_0$ into a Queue*/
    $QUEUE$ = InsertQueue($QUEUE$, $p_0$);
    while (not Empty($QUEUE$)){
        /*get an element from the head of $QUEUE$*/
        $p_0$ = RemoveQueue($QUEUE$);
        For each $p$ that is adjacent to $p_0$ {
            if ($L(p) \neq$ '*' and $C(p, p_0) \geq \lambda$)
                $QUEUE$ = InsertQueue($QUEUE$,$p$);
                $L(p)$ = '*'; }}}
    $S'$ = $\{p : L(p) = 0\}$; /*$S'$ is a $\lambda$-connected component*/
    if ($S'$ has more points than $S$)
        $S$ = $S'$; /* save the largest component to $S$*/
}
return($S$);
```

4.3 Time Complexity and Memory Usage

The water vapor attribute value of each point is represented by a 4-byte double. If one window contains all the global water vapor data for a specific time (360*180 locations), it will take 260K byte of memory. The computation of the wavelet transformation is efficient. A fast wavelet transformation needs $O(N)$ operations, where N is the number of objects(locations). Its memory usage is also linear [25], . For each data window, the time complexity of *WaveletAnalysis* algorithm is $O(m)$, where m refers to the window size (or number of pixels in the image). The time complexity of identifying the largest λ-connected part is $O(m)$, because in the search algorithm, each pixel will be visited twice. It also validates that the breadth-first based search technique is an efficient searching algorithm. For trajectory tracking, the time complexity is $O(p+K)$, where $O(p)$ is used for extracting the boundary and center point of the outlier region (with an average of p points), and $O(K)$ is the cost of "noise" point elimination and speed calculation. Since p and K are very small compared with m, the running time will be dominated by the wavelet transformation and image segmentation operations. The total time complexity will

Algorithm : Trajectory

```
Input:
    $T$: Previous trajectory;
    $R$: Current detected region;
    $K$: Number of latest center points along $T$;
Output:
    $T$: Updated trajectory with $R$ appended
/* extract boundary of in $R$ */
$B$ = getBoundary($R$);
/* Calculate the central point of $R$ */
$C$ = getCenter($R$);
/*Eliminate “noise” points*/
$T$ = verification($T$,$C$,$K$);
/*Compute the moving speed of center point*/
$T$ = calculateSpeed($T$);
/*Output the new trajectory*/
Output($T$);
```

correspond to the total number of objects N(the aggregation of m for all windows), that is, $O(N)$.

5 Experiment Results

We used NOAA/NCEP (National Oceanic and Atmospheric Administration/National Centers of Environmental Prediction) global reanalysis data set, which provides multiple parameters with a resolution of 1 degree by 1 degree. This data set covers the entire globe and is updated 4 times a day, at 0AM, 6AM, 6PM, and 18PM. Our main objective is to trace hurricane or tropical storm from satellite data. Water vapor data are selected in our study. Even though a hurricane is not defined by high concentration of water vapor, it is always accompanied by high concentration of water vapor. Usually, the stronger the circulation wind, the lower the surface pressure, the stronger convection and the higher concentration of water vapor. Generally, surface wind and surface air pressure are better indicators to define a hurricane. However, these parameters are very difficult to be retrieved from satellite observation under cloud cover, especially for hurricanes which have deep convection and thick clouds. In contrast, total water vapor (integrated from surface to top of the atmosphere) is a well-validated satellite product which provides a good estimation of the real world even under heavy cloud. Figure 4 shows an image of global water vapor distribution on October 3, 2002. In most cases, the tropical region is covered by the high values of water vapor. Our objectives are to identify and track the movement of outlier regions. In this section, we will demonstrate the experimental

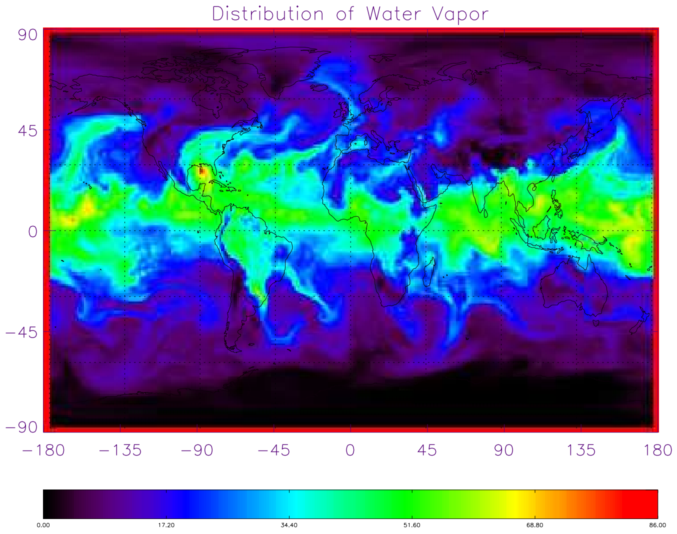

Fig. 4. Global distribution of water vapor.

results of wavelet transformation, image segmentation, and trajectory tracking.

5.1 Wavelet Transformation

We first performed a Mexican hat wavelet transformation on the data over all latitudes. Figure 5 is the water vapor data for latitude 26^o North and its wavelet power. In Figure 5(a), the solid line is the original data and the dashed line is the filtered (reconstructed with scales 2 and 3) data. Figure 5(b) is the plot of the wavelet power of the original data. Figure 5(c) is the plot of the wavelet power of the filtered data. Figure 5 shows that the variation exists on all scales and the power of variation changes at different locations. This figure also shows that the Mexican hat wavelet has a satisfactory localization resolution. We mainly focused on the anomalies with sub-weather scales, that is with variations of 1000km or 10 degrees in longitude at the mid-latitude region. Figure 6 is the global map of wavelet transformation power with scale index 3. Clearly, there are some areas where the power is especially high. In these areas the spatial variation with scale index 3 is prominent and these areas are suspected region outliers.

Comparing Figure 6 with Figure 4, the area with the high value in Figure 4 over the Gulf of Mexico also has a high wavelet power. However, the high vapor value areas near 160^oW in the tropic region do not show strong wavelet power in Figure 6, and the low value areas in South America show high wavelet power in Figure 6. Therefore, a high value does not necessarily guarantee a high wavelet power. We focus on the spatial variation, not the value of the variable. Wavelet power mainly represents the variation of the signal in the spatial domain. Another advantage of using a wavelet transformation is its multi-scale capability, as mentioned earlier:

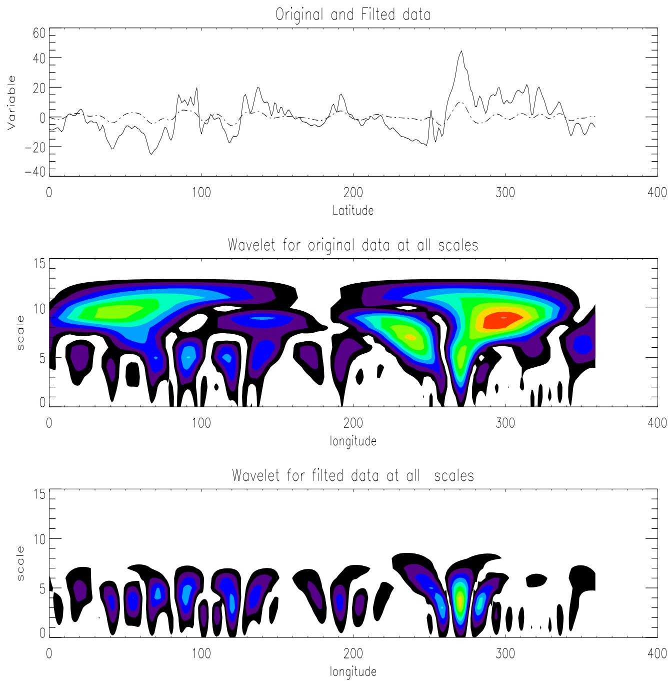

Fig. 5. Mexican hat wavelet power with locations and scales (a:top, b:center, c:bottom).

we can focus on only the scales in which we are interested. This makes easier to study the complicated variations in multi-scale meteorological data.

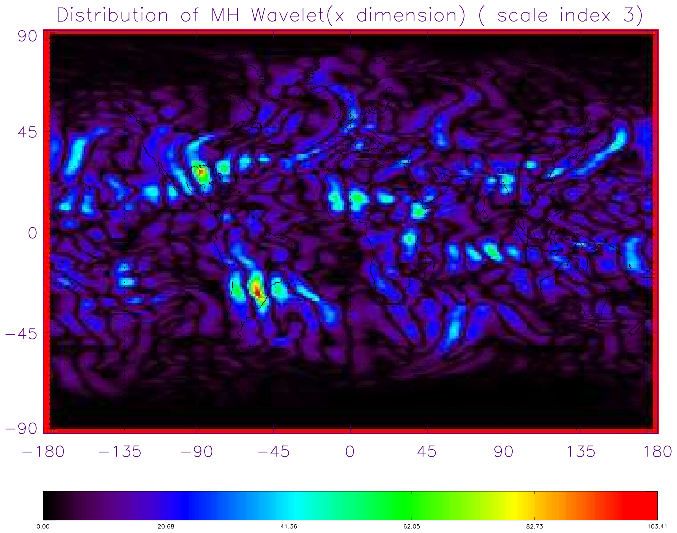

Fig. 6. Wavelet power distribution at scale index 3.

We performed wavelet transformation on the X dimension along latitude because for weather systems the scale is usually represented based on the latitude. For the basic atmospheric parameter distribution, there is a strong variation with different latitudes, such as the difference between the tropics and high latitude areas. This variation is the normal pattern of the general atmosphere and is not an anomalous feature. Thus, when detecting spatial variations, it useful to focus on the variation along the latitude (X-axis). Technically, however, we can also perform a wavelet transformation along longitude (Y-axis). Figure 7 shows the reconstructed water vapor distribution using an inverse wavelet transformation along both latitude and longitude (X and Y). Figure 7 reveals many more patterns than Figure 6. However, these patterns are caused by the normal variation along the longitude Y and are merely noises in most cases.

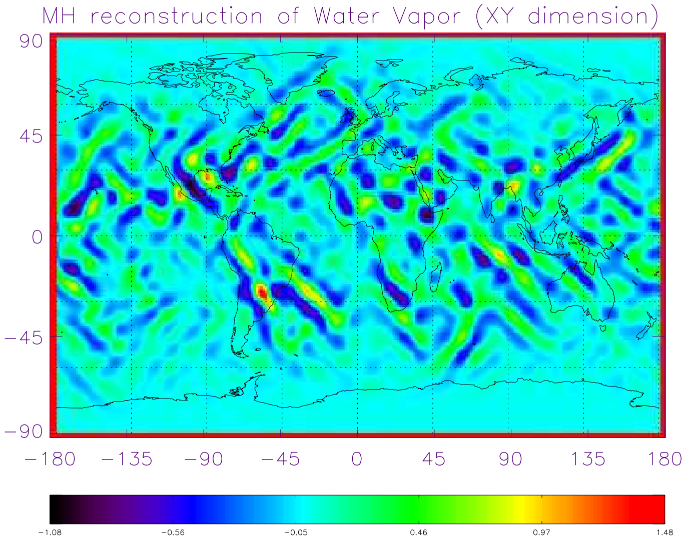

Fig. 7. Reconstruction on both X Y dimension.

5.2 *Image Segmentation and Tracking*

In this experiment, we examined the the water vapor data over the period of 9/17/2003-9/19/2003, during which Hurricane Isabel landed on the east coast of the United States. Hurricane Isabel formed in the central Atlantic Ocean on September 6th, 2003. It moved in a general west-northwestward direction and strengthened to a category five hurricane by September 11th. Weakening began on September 16th as the hurricane turned north-northwestward. On September 18th, Isabel made landfall on the outer banks of North Carolina as a category two hurricane. Portions of eastern North Carolina and Southeastern Virginia experienced hurricane-force winds. Experimental results for Hurricane Isabel demonstrate the effectiveness of our algorithms in detecting abnormal meteorological patterns. Figure 8 shows the wavelet image at 0AM on September 18th, 2003. When the boundary of Hurricane Isabel is extracted by *Algorithm Segmentation*, it shows the center is located at (32.54°N, 71.80°W). Figure 9 shows another experimental result on September 18th, 2003, at 6:00AM. The boundary of Hurricane Isabel is clearly identified, showing the center is located at (33.05°N, 72.28°W). During these six hours, the trend of Hurricane Isabel can be observed as it moves northwestward overland.

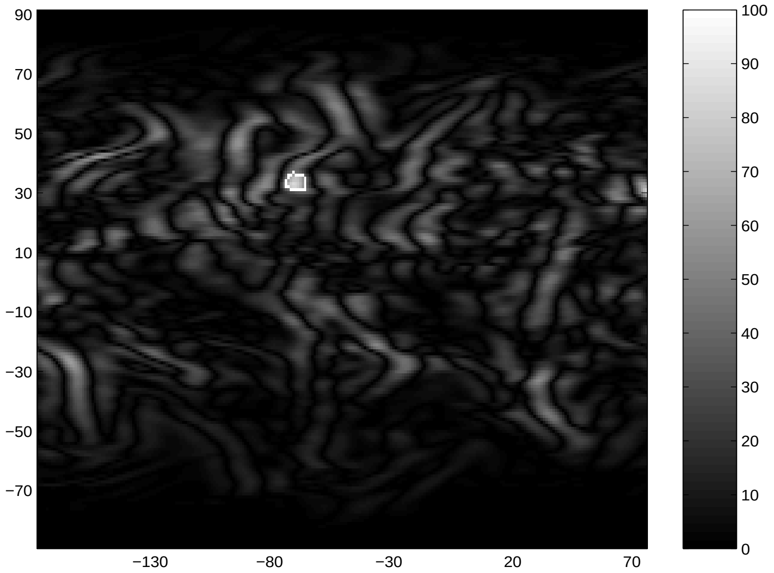

Fig. 8. Wavelet power distribution at 0AM Sept. 18th, 2003 with Hurricane Isabel identified.

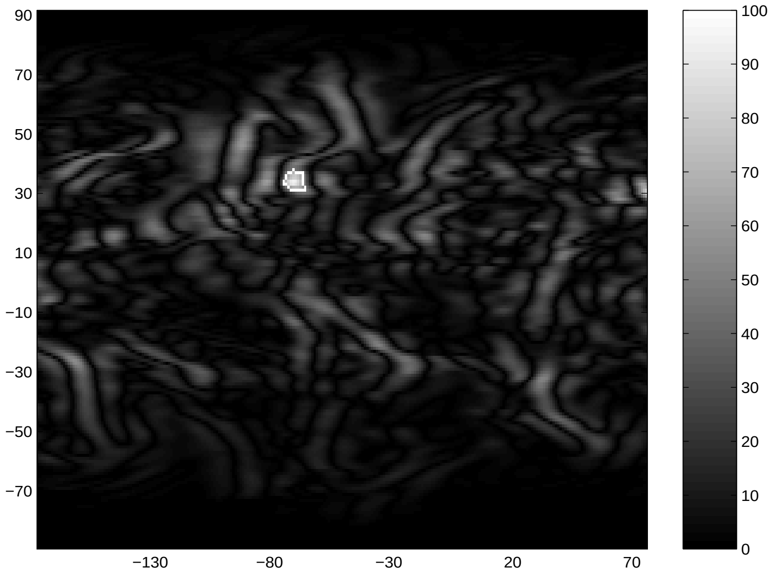

Fig. 9. Wavelet power distribution at 6AM Sept. 18th, 2003 with Hurricane Isabel identified.

Figure 11 shows the 3D trajectory of Hurricane Isabel from September 17th, 2003 to September 19th, 2003. Since the location of hurricane was measured every six hour each day, 12 regions are illustrated in this figure. The boundary of each outlier region is depicted by a dotted line and the center points are connected, so that its moving trajectory can be observed. As can be seen from the figure, region 4 is not consistent with the locations of other regions. It is a “noise” outlier caused by other weather patterns or inappropriate segmentation parameters. Region 4 is flagged by the verification procedure in *Algorithm Trajectory* and removed. Figure 10 shows the new trajectory after eliminating “noise” regions. The northwestward

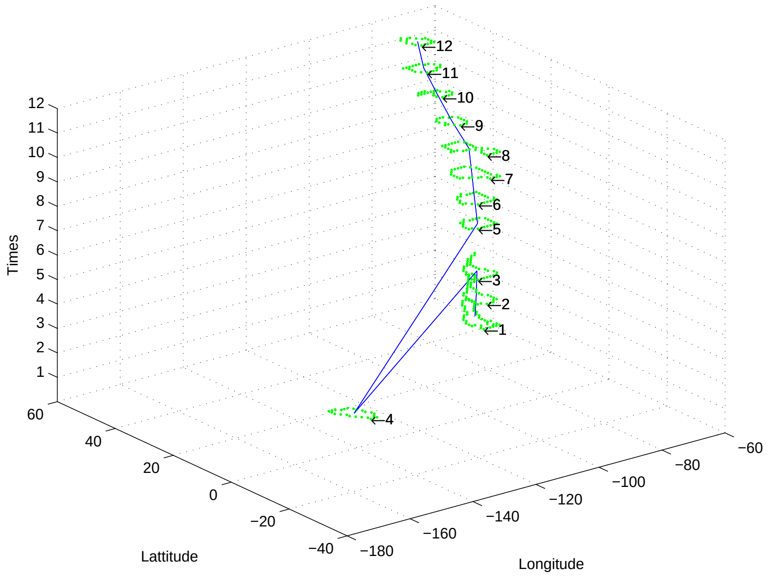

Fig. 10. Trajectory of moving region with “noise” data.

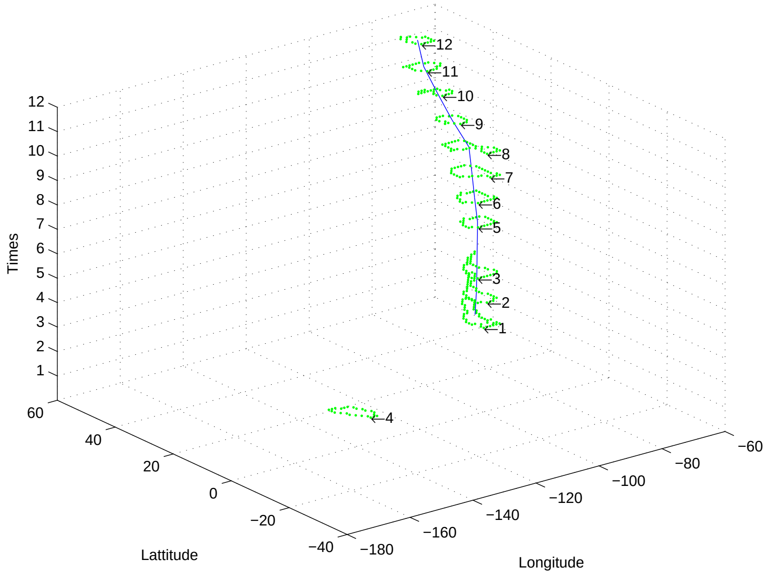

Fig. 11. Trajectory of moving region without “noise” data.

movement of Hurricane Isabel can be clearly observed. The latitude and longitude of the hurricane center are listed in Table 3. “Flag=1” denotes that the region is correctly detected and “Flag=0” denotes that the region is “noise” data and will not be recorded.

Table 4 shows the processing time of the proposed λ-connectedness based image segmentation algorithm. The size denotes the number of data frames, where each frame is made up of 180×360 data points, and the time is measured in seconds. In the experiment, we used a Pentium4 (2.8GHz) PC with 512MB memory. The experimental results show that our image segmentation algorithm is efficient to

SN	Latitude	Longitude	Time	Flag
1	35.27	-70.07	09/17/2003/0Z	1
2	34.41	-70.42	09/17/2003/6Z	1
3	33.31	-71.28	09/17/2003/12Z	1
4	-29.20	-167.82	09/17/2003/18Z	0
5	32.54	-71.80	09/18/2003/0Z	1
6	33.05	-72.28	09/18/2003/6Z	1
7	33.91	-72.34	09/18/2003/12Z	1
8	34.53	-72.70	09/18/2003/18Z	1
9	38.05	-74.86	09/19/2003/0Z	1
10	41.41	-76.52	09/19/2003/6Z	1
11	43.61	-78.68	09/19/2003/12Z	1
12	45.46	-78.97	09/19/2003/18Z	1

Table 3
The tracking data of hurricane center.

Data Size (180×360)	1	4	9	16	64
Time (Sec)	0.003	0.017	0.030	0.048	0.218

Table 4
The execution time of image segmentation.

process a high speed meteorological data sequence, taking only 0.218 seconds to process 64 image windows, with each window containing 180×360 points.

6 Conclusion

In this paper, we propose a comprehensive approach for detecting and tracking spatial region outliers in meteorological data. Our approach is based on wavelet transformation and image segmentation. First, wavelet transformation filters out noises and highlights spatial variation of specific scales. Then, an efficient image segmentation technique, λ-connectedness method, is applied to extract the largest connected region whose intensity is much higher than its neighbors. Finally, the trajectory of the outlier region is calculated for a sequence of meteorological data frames. The proposed algorithms can be executed with linear time and suitable for identifying anomalies in continuous meteorological data sequences. The experiment on the Hurricane Isabel data set validates the efficiency and effectiveness of our approach.

Our research will be extended in the following directions. First, we plan to study region outliers in three-dimensional spatial space with multiple attributes, such as pressure, rainfall, cloud cover, and temperature. Second, we will design algorithms to identify and track multiple moving outlier regions simultaneously. Furthermore, we will apply our algorithms on the real NOAA online database to discover anomalous meteorological patterns.

Acknowledgments: *The authors would like to thank the anonymous reviewers for their valuable comments.*

References

[1] N. R. Adam, V. P. Janeja, and V. Atluri. Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 576–583, Nicosia, Cyprus, 2004.

[2] C. C. Aggarwal. A framework for diagnosing changes in evolving data streams. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 575–586, San Diego, California, USA, June 9-12, 2003.

[3] M. Alexiuk, N. Pizzi, P. C. Li, and W. Pedrycz. Classification of volumetric storm cell patterns. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 4(3):206–211, 2000.

[4] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley, New York, 1994.

[5] T. Chan and L. Vese. An active contour model without edges. In *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, pages 141–151, Corfu, Greece, September 26-27, 1999.

[6] L. Chen, H.-D. Cheng, and J. Zhang. Fuzzy subfiber and its application to seismic lithology classification. *Information Sciences: Applications*, 1(2):77–95, 1994.

[7] T. W. Cheng, D. B. Goldgof, and L. O. Hall. Fast fuzzy clustering. *Fuzzy Sets and Systems*, 93(1):49–65, 1998.

[8] M. Cheriet, J. N. Said, and C. Y. Suen. A recursive thresholding technique for image segmentation. *IEEE Transactions on Image Processing*, 7(6):918–921, June 1998.

[9] T. Denoeux and P. Rizand. Analysis of rainfall forecasting using neural networks. *Neural Computing and Applications*, 3(1):50–61, 1995.

[10] N. Djafri, A. Fernandes, N. W. Paton, and T. Griffiths. Spatio-temporal evolution: querying patterns of change in databases. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, pages 35–41, McLean, Virginia, USA, November 8-9, 2002.

[11] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 71–80, Boston, Massachusetts, USA, August 20-23, 2000.

[12] G. Erlebacher, M. Hussaini, and L. Jameson. *Wavelet Theory and its Application*. Oxford University, 1996.

[13] C. Giannella, J. Han, J. Pei, X. Yan, and P. Yu. Frequent patterns in data streams at multiple time granularities. *NSF Workshop on Next Generation Data Mining*, AAAI/MIT, 2003.

[14] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (Second Edition)*. Prentice Hall, 2002.

[15] T. Griffiths, A. Fernandes, N. W. Paton, K. T. Mason, B. Huang, and M. F. Worboys. Tripod: A comprehensive model for spatial and aspatial historical objects. In *ER '01: Proceedings of the 20th International Conference on Conceptual Modeling*, pages 84–102, Yakohama, Japan, Nov 27-30, 2001.

[16] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams: theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):515–528, 2003.

[17] R. Haining. *Spatial Data Analysis in the Social and Environmental Sciences*. Cambridge University Press, 1993.

[18] J. Haslett, R. Brandley, P. Craig, A. Unwin, and G. Wills. Dynamic Graphics for Exploring Spatial Data With Application to Locating Global and Local Anomalies. *The American Statistician*, 45:234–242, 1991.

[19] D. Hawkins. *Identification of outliers*. Chapman and Hall, Reading, Massachusetts, 1980.

[20] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 97–106, San Francisco, CA, August 26-29, 2001.

[21] K. Koperski, J. Adhikary, and J. Han. Spatial data mining: Progress and challenges. In *Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96)*, pages 1–10, Montreal, Canada, 1996.

[22] S. Lee, S. Chung, and R. H. Park. A Comparative Performance Study of Several Global Thresholding Techniques for Segmentation. *Computer Vision, Graphics and Image Processing*, 52(2):171–190, 1990.

[23] P. Li, N. Pizzi, W. Pedrycz, D. Westmore, and R. Vivanco. Severe storm cell classification using derived products optimized by genetic algorithm. *Proceedings of the 2000 IEEE Canadian Conference on Electrical and Computer Engineering*, 1:445–448, March 2000.

[24] Q. Li, T. Li, S. Zhu, and C. Kambhamettu. Improving medical/biological data classification performance by wavelet preprocessing. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, page 657, Maebashi City, Japan, December 9-12, 2002.

[25] T. Li, Q. Li, S. Zhu, and M. Ogihara. A survey on wavelet applications in data mining. *ACM SIGKDD Explorations Newsletter*, 4(2):49–67, 2002.

[26] Y. Li, J. Han, and J. Yang. Clustering moving objects. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 617–622, Seattle, Washington, USA, Aug 22-25, 2004.

[27] C.-T. Lu, D. Chen, and Y. Kou. Algorithms for spatial outlier detection. In *Proceedings of the Third IEEE International Conference on Data Mining*, pages 597–600, Melbourne, Florida, USA, Nov 19-22, 2003.

[28] A. Luc. Local indicators of spatial association: Lisa. *Geographical Analysis*, 27(2):93–115, 1995.

[29] Y. Meyer. *Wavelet and Operators*. Cambridge University Press, 1992.

[30] N. E. Miller, P. C. Wong, M. Brewster, and H. Foote. A wavelet-based text visualization system. In *Proceedings of the Conference on Visualization '98*, pages 189–196, Research Triangle Park, North Carolina, USA, 1998.

[31] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communication of Pure and Applied Mathematics*, 42:577–685, 1989.

[32] N. R. Pal and S. K. Pal. Entropy: A new definition and its applications. *IEEE Transactions on Systems, Man and Cybernetics*, 21(5):1260–1270, 1991.

[33] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Distributed deviation detection in sensor networks. *ACM SIGMOD Record: Special Section on Sensor Network Technology and Sensor Data Managment*, 32(4):77–82, 2003.

[34] Y. Panatier. *Variowin. Software For Spatial Data Analysis in 2D*. Springer-Verlag, 1996.

[35] J. F. Peters, Z. Suraj, S. Shan, S. Ramanna, W. Pedrycz, and N. Pizzi. Classification of meteorological volumetric radar data using rough set methods. *Pattern Recognition Letter*, 24(6):911–920, 2003.

[36] L. Ramirez, W. Pedrycz, and N. Pizzi. Severe storm cell classification using support vector machines and radial basis function approaches. In *Proceedings of Canadian Conference on Electrical and Computer Engineering*, pages 87–92, Toronto, Canada, May 13-16, 2001.

[37] A. Rosenfeld. The fuzzy geometry of image subsets. *Pattern Recogni-tion Letters*, 2(5):311–318, 1983.

[38] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 428–439, New York, NY, August 24-27, 1998.

[39] S. Shekhar and S. Chawla. *A Tour of Spatial Databases*. Prentice Hall, 2002.

[40] S. Shekhar, C.-T. Lu, and P. Zhang. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 371–376, San Francisco, California, USA, 2001.

[41] Z. Suraj, J. F. Peters, and W. Rzasa. A comparison of different decision algorithms used in volumetric storm cells classification. *Fundamenta Informaticae*, 51(1):201–214, 2002.

[42] C. Torrence and G. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79(1):61–78, January 1998.

[43] Y. Wang. Jump and sharp cusp detection by wavelets. *Biometrika*, 82(2):385–397, 1995.