Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Information Sciences 177 (2007) 1609–1632
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Detecting and tracking regional outliers in meteorological data

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

[a] *Department of Computer Science, Virginia Polytechnic Institute and State University, 7054 Haycock Road, Falls Church, VA 22043, United States*
[b] *QSS Group, Inc., 4500 Forbes Blvd Lanham, MD 20706, United States*
[c] *Department of Computer Science and Information Technology, The University of the District of Columbia, 4200 Connecticut Avenue NW, Washington, DC 20008, United States*

Received 15 March 2005; received in revised form 20 August 2006; accepted 7 September 2006

Abstract

Detecting spatial outliers can help identify significant anomalies 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 focused on identifying single location points over a static data frame. In this paper, we propose and implement a systematic methodology to detect and track regional outliers in a sequence of meteorological data frames. First, a wavelet transformation such as the Mexican Hat or Morlet is used to filter noise 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 outlying regions for consecutive frames. In addition, we conducted experimental evaluations using real-world meteorological data and events such as Hurricane Isabel to demonstrate the effectiveness of our proposed approach.

Keywords: Spatial outlier detection; Wavelet; Image segmentation; Data sequence; Meteorological data

1. Introduction

Due to the ever-increasing amount of spatial data, spatial data mining has become an important research area over the past decade [19,44]. 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 [24], 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 for spatial data mining have arisen from the following issues. First, classical data mining is designed to process numerical and categorical data, whereas

[*] Corresponding author.
E-mail addresses: ctlu@vt.edu (C.-T. Lu), ykou@vt.edu (Y. Kou), jiang.zhao@noaa.gov (J. Zhao), lchen@udc.edu (L. Chen).

0020-0255/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2006.09.013

spatial data mining deals with more complex spatial structures that often 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 deals with three dimensional volumes, but also contains temporal information. Together, these form spatial data sequences. In recent years, spatio-temporal data has attracted a great deal of attention from 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 a video stream, the frame sampling period can be as long as minutes, and there are no strict restrictions on processing speed. Several recent studies have been conducted to develop specific data mining techniques to detect useful patterns from continuous data streams [12,14,22]. Because these techniques are not specifically designed for processing spatial data, they may not be effectively utilized by geospatial applications. Intensive research is therefore needed to extract patterns from spatio-temporal data and to accurately predict their trends [11,29].

Outlier detection is a process that is often used to identify objects which differ from the other members in the same data set [4,21]. In the research on atmospheric sciences, huge amounts of spatial data are continuously collected from both observation and simulation modeling. Discovering useful patterns from these data, especially spatial outliers and their movements, will have great practical value for weather forecasting, environmental monitoring, and climate analysis. In meteorological data, spatial outliers are those 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 do not usually happen at a single location but cover an extended region, so spatial outliers are usually two dimensional regions. Furthermore, the spatio-temporal changes in these regions are frequently associated with variations of weather phenomena and climate patterns.

The ability to automatically extract these outlier regions is therefore a crucial issue. Typically, the methods used to address this problem rely on image segmentation and pattern recognition techniques [16,25]. 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 ranging from gross to fine. 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 [28].

In this paper, we propose and implement a systematic methodology 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 applied to identify the outlier regions. Finally, a regression technique is used to track the center movement of the outlier regions through consecutive frames. In addition, we conducted experimental evaluations using real-world meteorological data, in this case the data collected during Hurricane Isabel, to validate the effectiveness of the proposed algorithms. This paper is organized as follows: Section 2 provides a literature survey; Section 3 discusses the problems and proposes various approaches; Section 4 introduces the algorithm design; Section 5 describes its application to the real meteorological data and analyzes the experimental results; and finally, we summarize our work and discuss future research directions in Section 6.

2. Background and related work

This section surveys the 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. These existing spatial outlier detection methods can generally be grouped into two categories: graphic approaches and quantitative tests. Graphic approaches are based on the visualization of spatial data, which highlights spatial outliers. Examples include variogram clouds and pocket plots [20,38]. Quantitative methods, e.g., Scatterplot [19] and Moran scatterplot [32], provide tests that distinguish spatial outliers from the remainder of the data set. For instance, Shekhar et al. introduced a method for detecting spatial outliers in graph data [45]. An outlier may have a negative impact on its neighbors when its attribute value is much higher or lower than the average of its neighbors. Two iterative methods and one median-based approach were proposed in [31] to address this problem. Most of the existing spatial outlier detection methods are designed for point data. However, outliers may also exist in other spatial forms, such as lines and regions.

Image segmentation partitions an image into different components or objects. This 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 most popular, threshold segmentation, uses a threshold or clip-level to transform a grey-scale image into a binary image. Cheriet et al. proposed an approach that explores the use of an optimal threshold for minimizing the ratio of between-segments variance and the total variance [9]. Another approach, called the maximum entropy approach, is to define a threshold based on comparing the entropies of the segmented image [36]. The *second* method, proposed by Rosenfeld, treats an image as a 2D fuzzy set and uses α-cut to develop a fuzzy connectivity [41]. A variation of this fuzzy connectedness is to measure two pixels to evaluate if they are "fuzzy connected". A pixel set is referred to as 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* method is called split-and-merge segmentation [16], 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 a merging process will be performed. This method is accurate for complex image segmentation, but is complicated to implement and costs more computation time ($O(n \log n)$). The *fourth* category is related to the K-means or fuzzy c-means. This is a standard classification method that is often applied in image segmentation [8]. Here, the pixels are classified into different clusters to reach a minimum total "error", where the "error" refers to the distance from a pixel to the center of its own cluster. This method may produce very convincing results. Nevertheless, it employs an iterative process to reach convergence. The *fifth* method, the Mumford–Shah method, uses the variational principal [35]. This method considers three factors in segmentation: the length of the edges of all segments, the unevenness of the image without edges, and the error between the original image and the segmented images. When the three weighted factors reach a minimum, 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 the Mumford–Shah method, although level-sets may limit its reflexibility compared to the original method.

The theoretical analysis of segmentation algorithms in which the iteration processes are involved has had significant development in recent years. Researchers started to view the algorithms from different angles. For instance, the performance focuses not only on the segmentation details but also on the time cost. The state-of-the-art results for these segmentation algorithms are listed below: (1) For the K-means and the fuzzy c-means, researchers already have some good algorithms. Kanungo et al. improved Lloyd's K-means clustering algorithm. Their algorithm has the time complexity of at least $O(n \log n)$ since a kd-tree is required [23]. For example, to process a 128×128 image, it needs 14 times as much time as a linear algorithm that only requires one single scan. Runkler et al. [42] concluded the algorithm for fuzzy c-means is $O(c^2 n)$, where c is the number of classes. When there are 10 segments in the image, the execution time of this algorithm equals the time it takes to scan the image 100 times. The time complexity of the maximum entropy method used to select thresholds was studied by several researchers [7,15,30]. The time complexity is $O(n^2)$ for one or two thresholds. Even though this is a very good result, it is far from a log linear algorithm. A nearest neighbor based algorithm for finding the thresholds was proposed in [48], with a time complexity of $O(n \log n)$.

For meteorological data, the feature changes are usually not sharp enough to form clear edges. Therefore, the direct application of image segmentation cannot be utilized effectively to determine the coverage of the outlier regions. To distinguish the variation of feature gradients, wavelet techniques can be applied to the original spatial data before performing image segmentation [49]. Wavelet has many favorable properties, such as supporting multi-resolution and frequency localization, which makes it a widely used tool for digital signal processing and image processing [13,33]. In recent years, wavelet transformation techniques have been extended to data mining areas including clustering [43], classification [27], and data visualization [34].

A copious amount of attention has also been devoted to identifying and tracking useful patterns from continuous data sequences. These patterns include clusters, evolution, deviations, and anomalies. (1) *Clusters*:

Guha et al. proposed a divide-and-conquer approach for continuous data clustering [18], while Li et al. explored a clustering technique for moving objects that captured the moving patterns of a set of similar data points [29]. (2) *Evolution*: By extending an existing spatio-temporal data model, Tripod [17], Djafri et al. developed a general approach to characterizing the evolution of queries in a spatio-temporal database [11]. Aggarwal also presented a framework to detect changes and identify useful trends in evolving data sequences [2], while Giannella et al. designed an algorithm to maintain frequent patterns under a tilted-time window framework in order to answer time-sensitive queries [14]. (3) *Deviations*: Palpanas et al. utilized kernel density estimators for online deviation detection in continuous data sequences [37]. (4) *Anomalies*: 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 the patterns associated with severe weather events. Several approaches, including fuzzy clustering [3], neural networks [10], genetic algorithms [26], and support vector machines [40,46], have been proposed to classify storm cells. For example, Peters et al. presented a rough-set-based method capable of classifying four types of storm events: hail, heavy rain, tornadoes, and wind [39].

3. Problem and approach

In the Earth's atmosphere, anomalies emerge at different spatial scales and may appear in different shapes, which presents a daunting challenge to those seeking to detect outliers from continuous meteorological data sequences. Fig. 1 shows an image of the water vapor distribution over the east coast of the US, the Atlantic Ocean, and the Gulf of Mexico. The color intensity of each region reflects its water vapor content, and the "hot spot" located in the left portion of the image (28°N, 90°W) indicates a hurricane in the Gulf of Mexico. This outlier spot is not a single point but a group of points, a region. This region has a much higher water vapor content than its surrounding neighbors. Thus, *a regional 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 Fig. 1 is a regional outlier. Regional outliers are determined by domain experts based on a pre-defined threshold. The challenge is to design an efficient and practical approach to automatically detect regional 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 in the observation data.

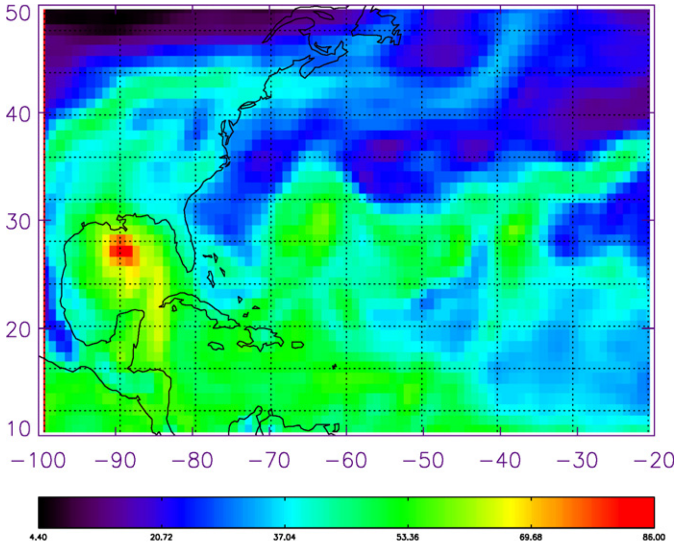

Fig. 1. A regional outlier (hurricane) in meteorological data.

In order to accurately extract regional outliers, it is preferable to decompose the original observations into different spatial scales in order to reduce the complexity and centralize the target object. Wavelet transformation provides such a capability with its multi-resolution characteristics. First, wavelet transformation 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 variations 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 and 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 power indicates the strength of the variation and the localization of high values reveals the places where anomalies exist. In the next section, we will discuss how the wavelet transformation is used in our application.

Image segmentation can be employed to extract spatial regions within which the meteorological characteristics are similar. The segmentation algorithm needs to be 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, the original frame or part of the original frame should be scanned only once. With $O(n \log n)$ time complexity, split-and-merge methods would not be practical for this purpose. *K*-means and fuzzy *c*-means, as well as the Mumford–Shah method, need extensive computation time because they require numerous iterations. Thus, to achieve a satisfactory speed, the threshold method and λ-connected method are the only two options since they both have linear time complexity. Threshold segmentation seems to offer the simplest solution, but when an image involves multiple thresholds, the determination of these thresholds values will be both difficult and time-consuming. The advantage of the λ-connectedness approach is that it can determine segments at different intensity levels without the need to calculate different thresholds or clip-level values. Based on the above reasoning, we chose the λ-connectedness approach to segment the meteorological data.

Our goal here is to identify the largest outlying region in which the value of each pixel is above a reasonable, predefined threshold. If we select the threshold method, the image is translated into a binary image based on a specified threshold, then a 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 the neighbors that have similar values by comparing them with the starting pixel. This method is therefore a generalized version of the threshold method. The details of a 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: $f{:}\Sigma_2 \to [0,1]$. Let $p = (x,y)$, $q = (u,v) \in \Sigma_2$, and p, q are said to be adjacent if $\max\{\|x-u\|, \|y-v\|\} \leqslant 1$. (A pixel, i.e. picture element, is a pair 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-connectedness 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:

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

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

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

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

or

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

Finally, the degree of connectivity between 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 deemed λ-connected if $C_f(x,y) \geqslant \lambda$.

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

4. Algorithm 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, a 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 to obtain an accurate trajectory.

4.1. Wavelet transformation

Wavelet transformation is a practical technique that is widely used in signal analysis and image processing. Wavelet transformation possesses several attractive features: (1) *Multi-resolution*: Wavelet transformation examines the signal at different frequencies with different resolutions, using a wider window for low frequencies and a narrower window for high frequencies. This feature works especially 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 a signal and focus on specific 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 has an advantage 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

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

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

$$x_i = \frac{\delta j \delta t^{1/2}}{C_\delta \Psi_0(0)} \sum_{j=0}^{J} \frac{\text{Real}\, W(n,s_j)}{s_j^{1/2}}$$

where C_δ is a constant for each wavelet function; Ψ_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 [47].

Here, not all scales of the wavelet transformation are included in the reconstruction. In order to filter out the non-related information, the data will be reconstructed based on the scales that are of interest. For example, if the low frequency range of the variations in the data set is to be studied, a low pass data set may be reconstructed in order 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 transformations. Here, we use two of the most widely used bases: the Morlet base and the Mexican hat base. The Morlet function is

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

The Mexican hat function is

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

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 $J \leqslant 2\log_2(\frac{N}{2})$, and 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 the scale index, real scale, and the corresponding period of the Fourier transform (here, since we are performing wavelet transformation on the 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 more slowly 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 variations as separate peaks in wavelet power, while the Morlet wavelet power combines both positive and negative peaks into a single broad peak [47]. Figs. 2 and 3 show examples of the two wavelet transformations. Fig. 2(a) is the original data water vapor distribution along a particular latitude. Figs. 2(b) and (c) show the wavelet transformation power at two different scales for a Mexican hat wavelet. Fig. 3 uses the Morlet wavelet and higher scale indices. From Figs. 2 and 3, we can see that the power of the wavelet transformation can be used to depict the distribution or localization of the variation at certain scales. The Mexican hat wavelet provides a better localization (spatial resolution), therefore we will use the Mexican hat wavelet to perform the analysis.

4.2. Detection algorithms

The proposed algorithm has two major purposes: 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 the 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, the CPU speed is capable of processing at least a number k of data windows ($k \geqslant 1$). 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. Each 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 the algorithm *Main* includes a sequence of continuously arriving data *DS*,

Table 1
Scale table for Mexican hat wavelet

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 2
Scale table for Morlet 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

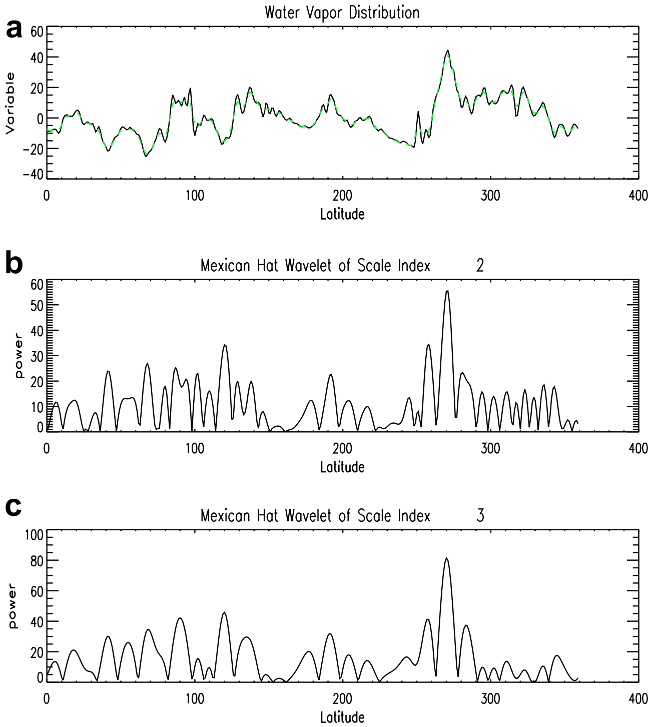

Fig. 2. A sample output of the Mexican hat wavelet (a: top, b: center, c: bottom).

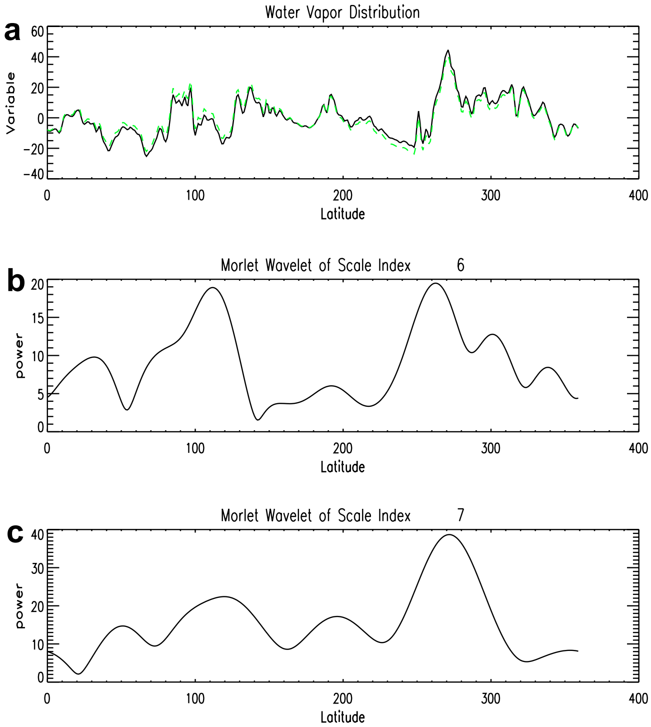

Fig. 3. A sample output of Morlet wavelet (a:top, b: center, c:bottom).

a set of selected scales S for wavelet transformation, a threshold θ, a similarity level λ for segmentation, and a 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 the algorithm *Main*, a set of scales of interest must first be determined by domain experts. The continuous and unbounded data sequence *DS* will be processed in each window, and the window size will be determined by the size of each data item and the memory capacity. We designate each window to represent an integral view of the global meteorological data (180° by 360°) as one time frame. From the *I*/*O* buffer, a sequence of data elements are fetched and stored in window *W*. Then, the algorithm *WaveletAnalysis* is performed on *W*, and the filtered data *wDomain* is generated, which focuses on particular scales. Next, based on *wDomain*, the algorithm *Segmentation* is employed to extract the outlier regions, which are connected components with attribute values above a predefined threshold θ. In particular, we focus on the largest connected region whose attribute 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's movement. Trajectory T will be recalculated and updated once a new region is added.

In fact, identifying the moving outlier region does not require processing the entire frame, as 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, the function *getPredictedArea*( ) can define the predicted area Σ_p, an approximate rectangle which contains all the possible positions of the moving region but is much smaller than the whole image *wDomain*. Instead of processing *wDomain*, we can obtain the outlier region by applying image segmentation only to Σ_p. This way, the costs 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. Note that for the first several frames, Σ_p is set to be *wDomain* and the whole image will be processed for segmentation. This utilization of the predicted area will make the segmentation process four times faster if its size is a quarter of the original frame. However, the area cannot be too small in order to maintain the quality of the search.

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 the previous frames;
Output:
    $O_r$ is the set of points in the outlier region
    T is the trajectory after appending the outlier region in the 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 the 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);}
```

The three sub-algorithms are discussed in detail as follows. The algorithm *WaveletAnalysis* is designed to filter out unimportant information from the source image data. The input of this algorithm is a sequence of data points W and a set of selected scales S. The output is the filtered image. Performing the wavelet transformation and reconstructing data based on a particular subset of scales can help filter noise and identify patterns with particular sizes. The algorithm first extracts the boundary of W. α_1 denotes the beginning latitude or longitude, and α_n denotes the ending latitude or longitude of the current window. Note that for meteorological data, the wavelet transformation will be performed along latitude lines. We will discuss the justification for this in the experimental section of this paper.

Algorithm. Wavelet analysis

```
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 */
$\alpha_1 = \text{getMinBound}(W)$;
/* get the maximum latitude(or longitude) of current window */
$\alpha_n = \text{getMaxBound}(W)$;
/*wavelet transformation and inverse transformation along all latitudes(or longitudes)*/
for($i = \alpha_1$; $i \leqslant \alpha_n$; $i$++) {
    wDomain = WaveletTransform($W, S, i$);}
/* output the filtered data window*/
Output(wDomain);
```

The algorithm *Segmentation* aims to extract the largest connected region above a threshold θ. It contains three input parameters: Σ, θ, and λ. The parameter Σ denotes the 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); and λ is the similarity level. The value of θ is determined by domain experts. Ordinarily, we will designate it as 75% 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 labeled 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 see if 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 smaller than S', S will be replaced by S', ensuring that S maintains the largest component discovered so far. The 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.

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 unlabeled 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 $C(p,p_0) \geqslant \lambda$)
             $QUEUE$ = 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($S$);
```

The objective of the algorithm *Trajectory* is to track the moving direction and speed of a given region and to determine the validity of the current detected region. The input parameters are the previously recorded trajectory T, the newly detected region R, and the number K of the latest 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 the *Segmentation* algorithm. It is possible for a region to be erroneously detected by the algorithm *Segmentation* due to errors in the raw data or an inappropriate segmentation threshold. Therefore, a verification function is needed in order to determine the validity of R based on the trajectory of the previous K regions. In the algorithm, the boundary point B is first 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. The speed and moving direction of the region center can thus be obtained from two valid consecutive center points. Finally, the new trajectory T will be updated and stored in permanent storage for a specified period of time.

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($T$);
```

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 260 K 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 [28]. For each data window, the time complexity of the *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. This 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 to m, the running time will be dominated by the wavelet transformation and image segmentation operations. The total time complexity will correspond to the total number of objects N(the aggregation of m for all windows), that is, $O(N)$.

5. Experimental results

We used the 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° × 1°. This data set covers the entire globe and is updated four times a day, at 0AM, 6AM, 12PM, and 6PM. Our main objective was to trace hurricanes or tropical storms by studying water vapor data from satellites. Even though a hurricane is not defined by a high concentration of water vapor, it is always accompanied by a high concentration of water vapor. Usually, the stronger the circulation wind, the lower the surface pressure, the stronger the convection, and the higher the concentration of water vapor. Although surface wind and surface air pressure are generally considered better indicators of a hurricane, these parameters are very difficult to retrieve from satellite observation under cloud cover, especially for hurricanes, which have deep convections and thick clouds. In contrast, the total water vapor (integrated from the surface to the top of the atmosphere) is a well-validated satellite product which provides a good estimation of the situation even under heavy cloud. Fig. 4 shows an image of global water vapor distribution on October 3, 2002. In most cases, the tropical region is covered by high values for the water vapor. Our methodology can be used to iden-

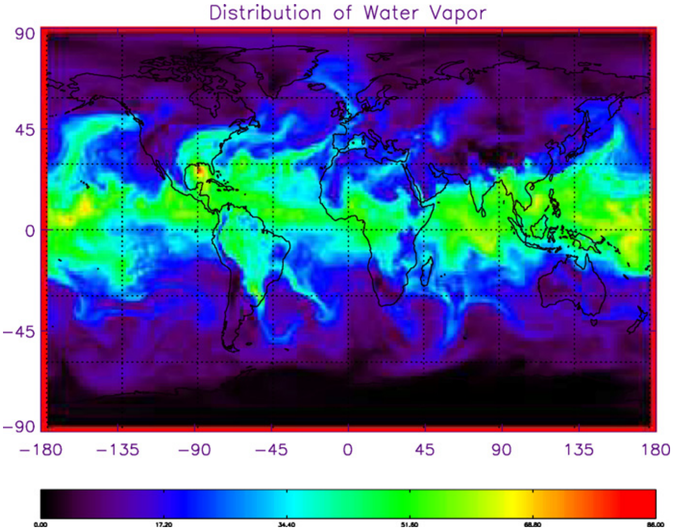

Fig. 4. Global distribution of water vapor.

tify high concentrations of water vapor that could be caused by hurricanes. The results need to be sent to domain experts for further validation.

The human eye can visually identify and track this type of extreme weather phenomena. Actually, the current weather forecast system is semi-automatic, mainly based on human recognition. However, when huge amounts of meteorological data continuously arrive, even the most experienced expert will be overwhelmed. Human eyes alone cannot handle a large number of data frames in a short period of time. In addition, humans are prone to making mistakes due to many factors such as distractions, lack of experiences, operator errors, and fatigue.

In order to automate accurate outlier identification and tracking, we proposed our systematic approach, which can handle continuous data sequences. The wavelet transform can focus on particular spatial scales and filter out the unnecessary and distracting information. Also, after the wavelet transformation, the difference between adjacent locations will be highlighted. Fast image segmentation algorithms can automatically extract regions with extreme weather patterns within a short period of time. Tracking functionality quickly reveals the movement of a weather pattern, with accurate information on its location, direction and speed, which is impossible even for human experts to find. In summary, the ultimate objective is to provide an efficient machine-based anomaly detection method to relieve weather forecasters and climate analyzers from intensive and error-prone work.

In this section, we will demonstrate the experimental 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. Fig. 5 shows the water vapor data for 26° North and its wavelet power. In Fig. 5(a), the solid line is the original data and the dashed line is the filtered data (reconstructed with scales 2 and 3). Fig. 5(b) is the plot of the wavelet power of the original data. Fig. 5(c) is the plot of the wavelet power of the filtered data. Fig. 5 shows that the variation

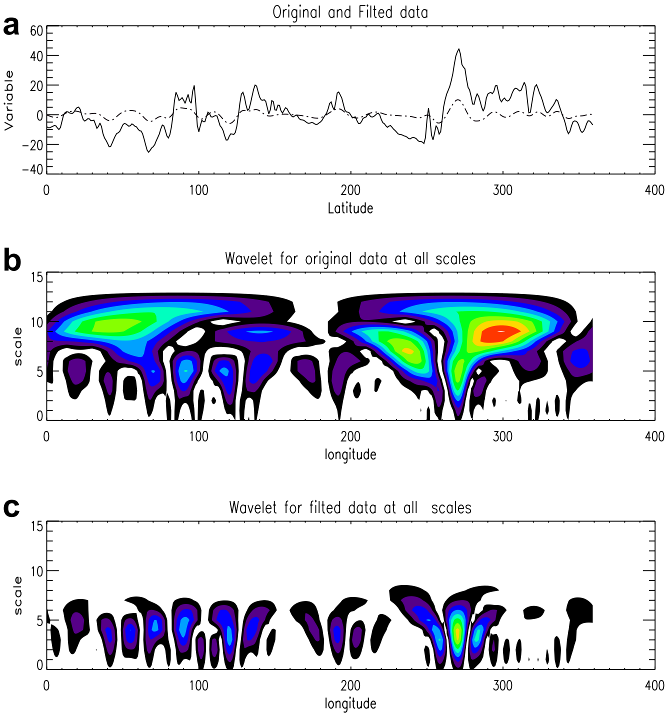

Fig. 5. Mexican hat wavelet power with locations and scales (a:top, b:center, c:bottom).

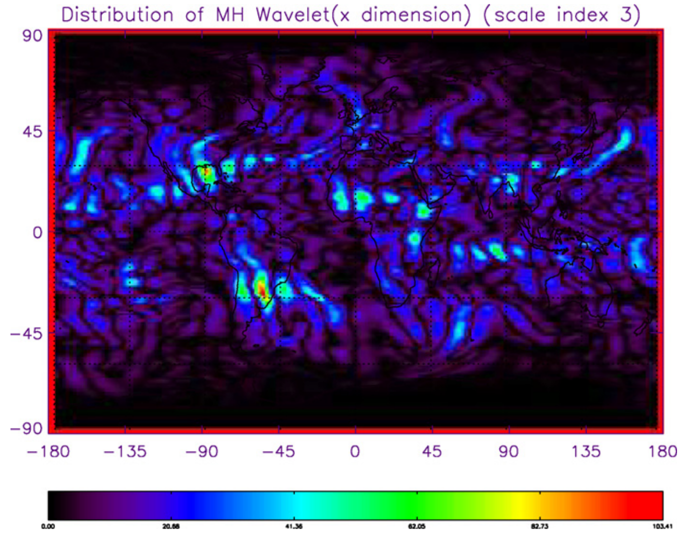

Fig. 6. Wavelet power distribution at scale index 3.

exists on all scales and the power of the 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 1000 km or 10° in longitude at the mid-latitude region. Fig. 6 shows the global map of wavelet transformation power with the scale index 3. Clearly, there are some areas where the power is especially high. In these areas, the spatial variations with the scale index 3 are prominent, and therefore these areas are viewed as suspected region outliers.

Comparing Fig. 6 with Fig. 4, in Fig. 4, the area over the Gulf of Mexico with a high value also has a high wavelet power. However, the high vapor value areas near 160°W in the tropic region do not show strong wavelet powers in Fig. 6, and the low value areas in South America show high wavelet powers in Fig. 6. Therefore, a high value does not necessarily guarantee a high wavelet power. Here, 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: we are able to focus only on the scales in which we are interested. This makes it easier to study the complicated variations in multi-scale meteorological data.

We performed the wavelet transformation in the X dimension along lines of 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 between different latitudes such as 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 is useful to focus on the variation along the latitude line (X-axis). Technically, however, we can also perform a wavelet transformation along the longitude line (Y-axis). Fig. 7 shows the reconstructed water vapor distribution using an inverse wavelet transformation along both the latitude line and longitude line (X and Y). Fig. 7 reveals many more patterns than Fig. 6. However, these patterns are caused by the normal variations along the longitude Y and are merely noise in most cases.

5.2. Image segmentation and tracking

In this experiment, we examined 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 generally 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

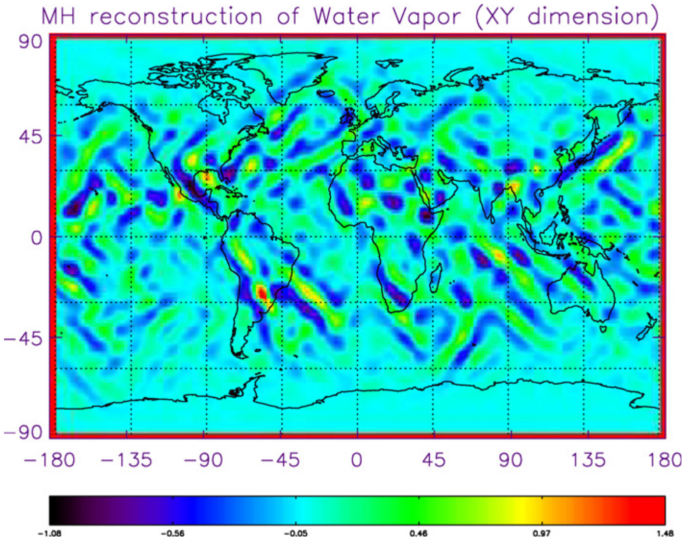

Fig. 7. Reconstruction on both *XY* dimension.

as a category two hurricane, while portions of eastern North Carolina and southeastern Virginia experienced hurricane-force winds. The experimental results for Hurricane Isabel demonstrate the effectiveness of our algorithms in detecting abnormal meteorological patterns. Fig. 8 shows the wavelet image at 0AM on September 18th, 2003. When the boundary of Hurricane Isabel was extracted by the algorithm *Segmentation*, it showed that the center is located at (32.54°N, 71.80°W). Fig. 9 shows another experimental result on September 18th, 2003, at 6:00AM. The boundary of Hurricane Isabel was clearly identified, showing the center was 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. 10 shows the 3D trajectory of Hurricane Isabel from September 17th, 2003 to September 19th, 2003. Since the location of hurricane was measured every six hours, 12 regions are illustrated in this figure covering these three days. The boundary of each outlier region is depicted by a dotted line and the center points are connected so that its trajectory can be observed. As can be seen from the figure, region 4 is not consistent with

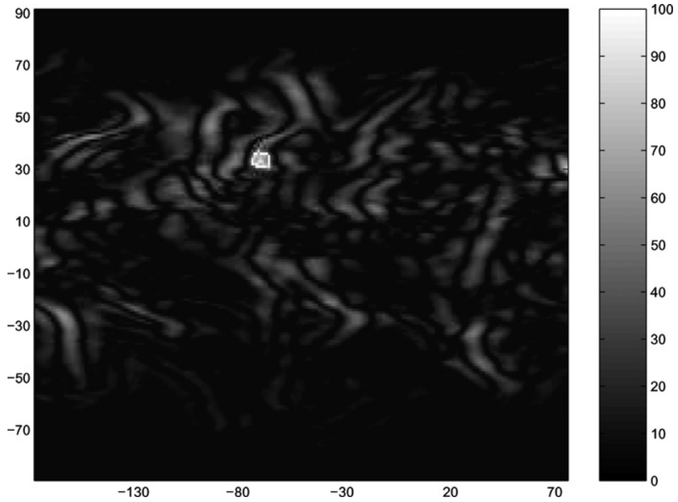

Fig. 8. Water vapor data at 0AM September 18th, 2003 with Hurricane Isabel identified.

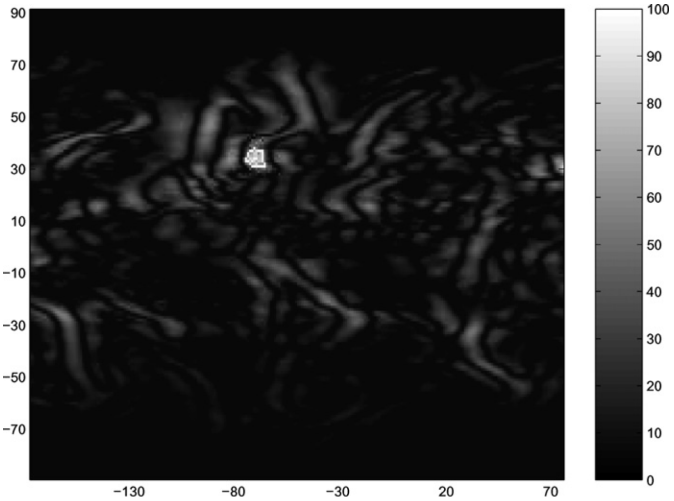

Fig. 9. Water vapor data at 6AM September 18th, 2003 with Hurricane Isabel identified.

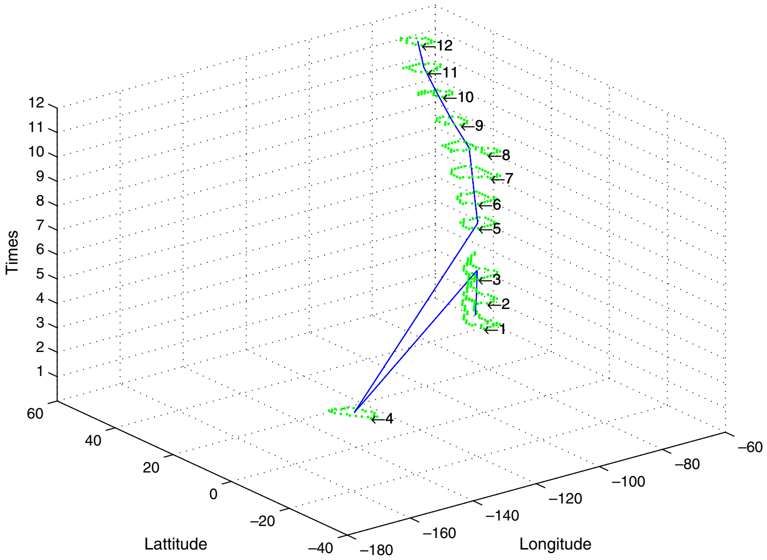

Fig. 10. Trajectory of moving region with "noise" data.

the locations of the 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 the algorithm *Trajectory*. Fig. 11 shows the new trajectory after eliminating the "noise" region. The northwestward 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 is not 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 this experiment, we used a Pentium4 (2.8 GHz) PC with 512 MB memory. The experimental results show that our image segmentation algorithm can efficiently process a high speed meteorological data sequence, taking only 0.218 s to process 64 image windows, each window containing 180×360 points.

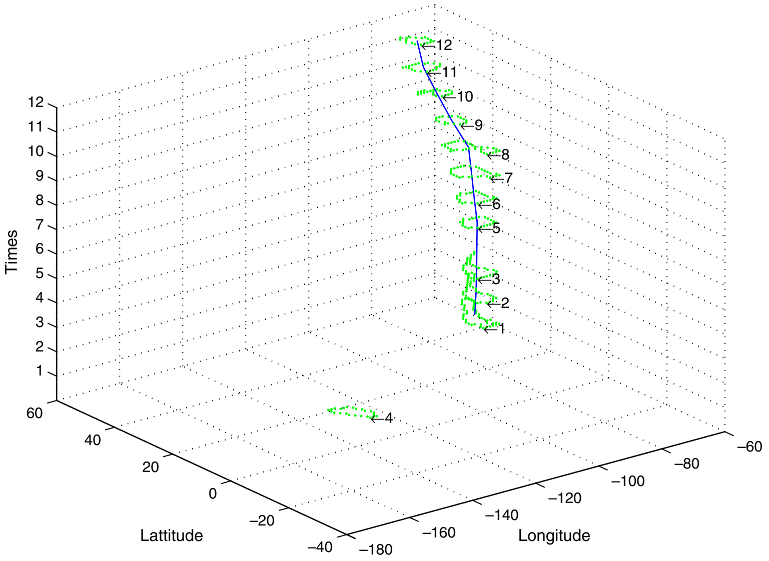

Fig. 11. Trajectory of moving region without "noise" data.

Table 3
The tracking data of hurricane center

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 4
The execution time of image segmentation

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

5.3. Comparison with other image segmentation methods

Image segmentation directly impacts the performance of region outlier detection in continuous data sequences. To validate the efficiency and effectiveness of the proposed λ-connectedness algorithm, two widely used methods, *K*-Means and Maximum Entropy, were compared. Three key elements are considered in the comparison: segmentation quality, stability, and running time. Stability measures the parameter variance between consecutive images. For continuous data sequences, a stable segmentation method is preferred, which has smaller parameter variance.

The three methods are conducted on two source images in the water vapor data which have been preprocessed by wavelet. Fig. 12(a) shows Image 1, the data collected at 0AM Sept 17, 2003. Fig. 12(b) illustrates

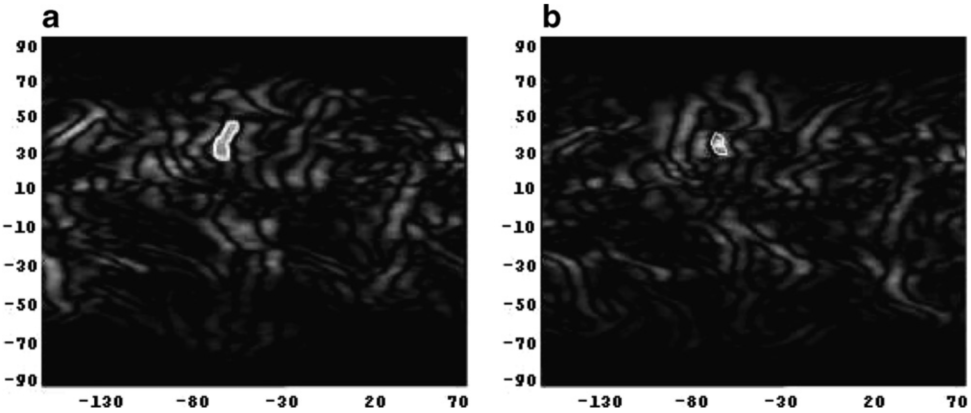

Fig. 12. Two source images for image segmentation comparison. (a) Image 1: 0AM, Sept 17, 2003 and (b) Image 2: 6AM, Sept 18, 2003.

Image 2, the data collected at 6AM Sept 18, 2003. The boundary of Hurricane Isabel is highlighted in white in both images and the center is at (35.27°N, 70.7°W) and (33.05°N, 72.28°W), respectively. For K-Means image segmentation, we experimented on various K values and demonstrated the results for $K = 2$, 4, and 8, respectively. For the Maximum Entropy method, the results are recorded for both single threshold and two thresholds. The λ-connectedness approach is tested based on $\lambda = 0.95$ and $\theta = 45\%$.

5.3.1. *Quality and stability comparison*

5.3.1.1. K-Means. The results of the K-Means segmentation are illustrated in Fig. 13. Figs. 13(a) and (b) show the results for $K = 2$, where Image 1 and Image 2 are segmented into 2 distinct gray levels, dark gray and black. The largest connected component is marked within a white rectangle. For both figures, the results are not satisfactory since the largest connected regions are not Hurricane Isabel. The results for $K = 4$ are illustrated in Figs. 13(c) and (d), where the images were segmented into four gray levels. In Fig. 13(c), Hurricane Isabel is correctly identified as the largest connected component as shown by a white arrow. In Fig. 13(d), the segmentation is not acceptable for Image 2. The region designated by a white arrow represents the location of Hurricane Isabel. However, it is far from the largest connected component. For example, the region marked by a white rectangle is apparently larger. When $K = 8$, the segments of the images are represented by eight different gray levels as shown in Figs. 13(e) and (f). In both figures, the boundary of Hurricane Isabel is identified accurately with a white arrow. In summary, the K-Means method is not effective in segmenting the water vapor data and locating the hurricane when $K = 2$ or $K = 4$. When K is increased to 8, the hurricane can be identified accurately. However, the running time will increase significantly. Please refer to Section 5.3.2 for details.

5.3.1.2. Maximum entropy. The Maximum Entropy method was tested on Image 1 and Image 2 as well. We experimented with the segmentation using a single threshold and 2 thresholds. The results of single threshold 67 are demonstrated in Figs. 14(a) and (b). The threshold 67 is automatically calculated based on Image 1. Based on this threshold, Fig. 14(a) shows that the segmentation of Image 1 is effective, with Hurricane Isabel identified using a white arrow. However, as shown in Fig. 14(b), the threshold 67 is not suitable for Image 2: the largest connected region centered at (44 °N, 90 °W) is marked with a white rectangle and it is apparently not Hurricane Isabel, which is marked by a white arrow. Based on Image 2, another threshold value 140 can be generated automatically. Fig. 15(a) shows that although this threshold can help determine the center of Hurricane Isabel in Image 1, the boundary of the detected region is not accurate and the size of the region is much smaller than the actual hurricane. As shown in Fig. 14(b), the threshold 140 is effective for Image 2, with the location of Hurricane Isabel accurately pinpointed by a white arrow. Nevertheless, one limitation of the single threshold Maximum Entropy is that different images need different thresholds and these thresholds vary significantly. This means that the threshold is not stable for continuous data sequences. The 2-threshold Maximum Entropy image segmentation achieves satisfactory results (please refer to Fig. 16(a)

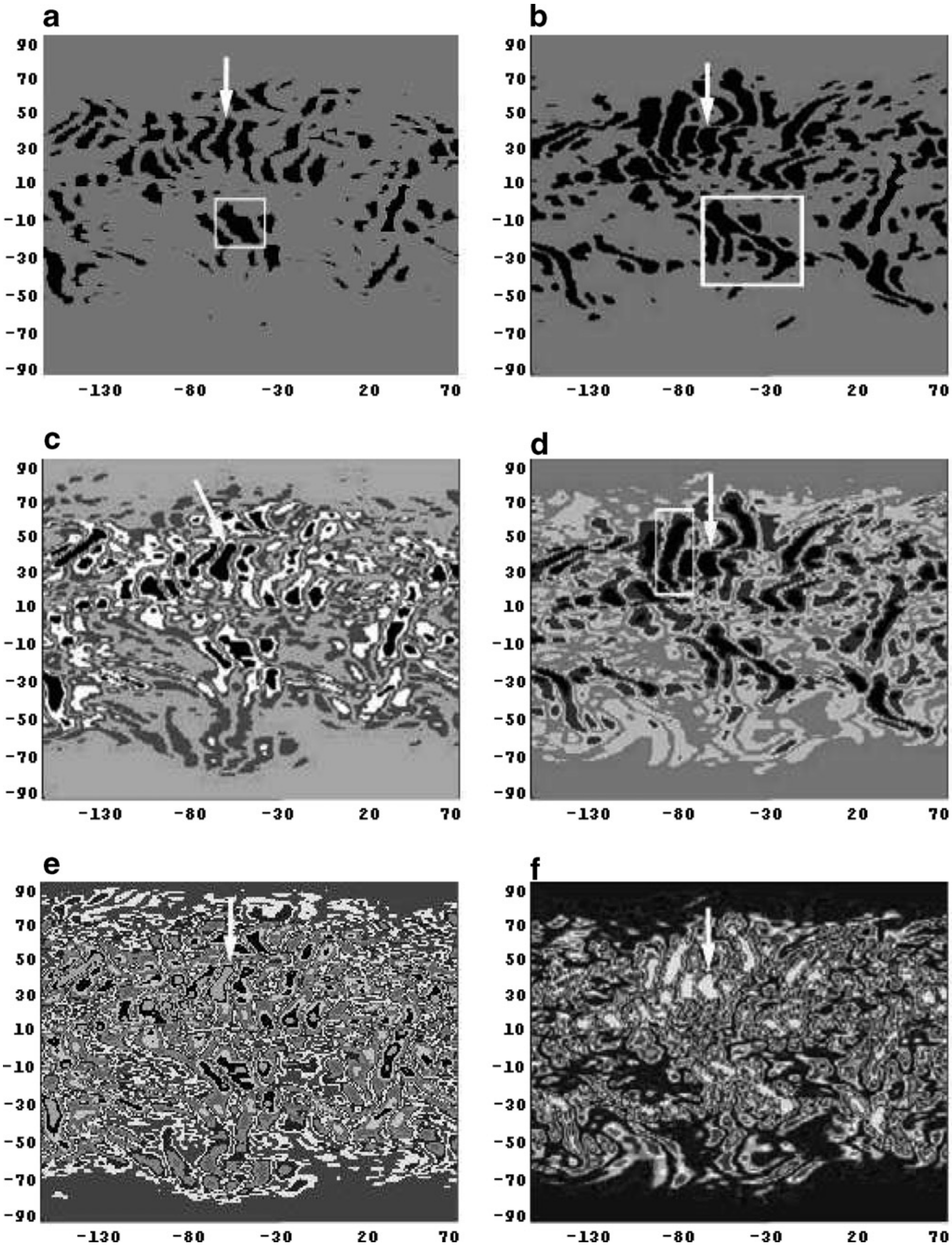

Fig. 13. The effectiveness of *K-Means* image segmentation: (a) $K = 2$ for Image 1; (b) $K = 2$ for Image 2; (c) $K = 4$ for Image 1; (d) $K = 4$ for image 2; (e) $K = 8$ for Image 1; (f) $K = 8$ for Image 2.

and (b)), with the boundary of Hurricane Isabel identified accurately. The threshold values are (56, 129) for Image 1 and (58, 140) for Image 2. However, the running time of the two-threshold Maximum Entropy exceeds 5 s, which may not be suitable for high-speed stream data processing. Please see Section 5.3.2 for details.

5.3.1.3. λ-connectedness. Fig. 17 demonstrates the results of the λ-connectedness image segmentation. λ was selected as 0.95 and the clip parameter θ was set to 45%. For both images, the location of Hurricane Isabel

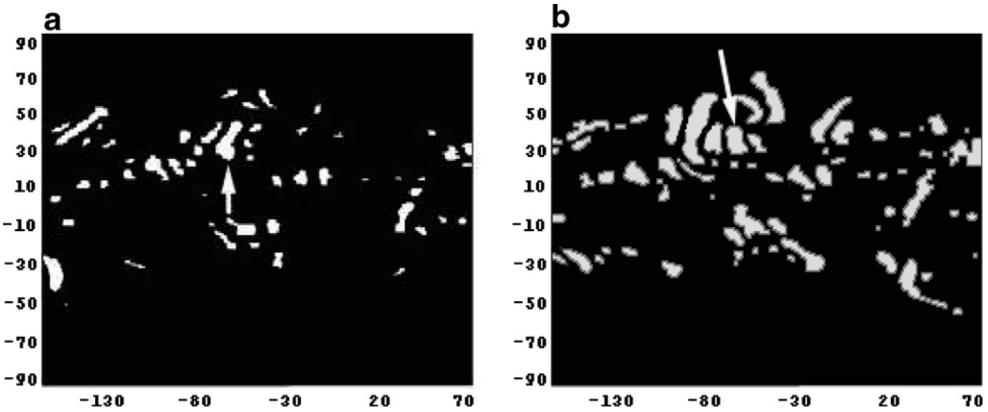

Fig. 14. Maximum entropy image segmentation: single threshold. (a) Threshold = 67 for Image 1 and (b) threshold = 67 for Image 2.

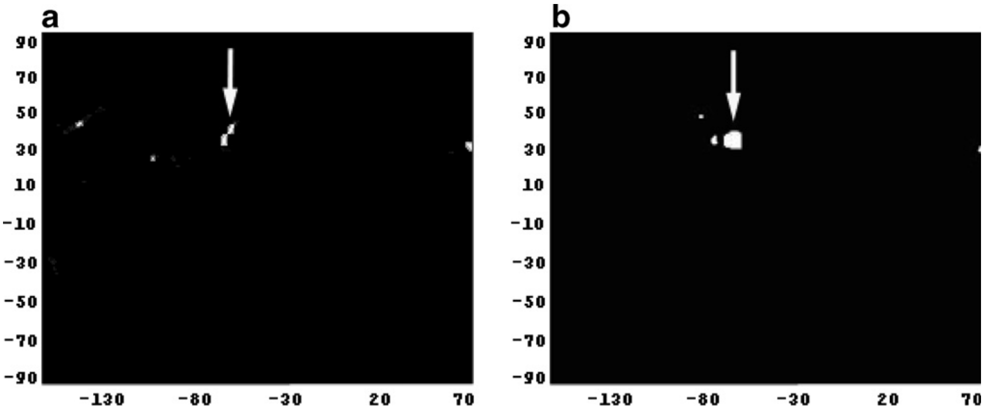

Fig. 15. Maximum entropy image segmentation: single threshold. (a) Threshold = 140 for Image 1 and (b) threshold = 140 for Image 2.

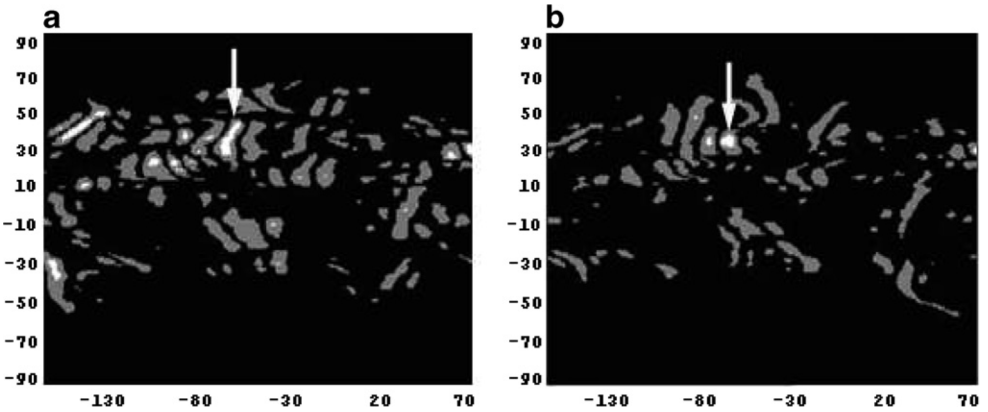

Fig. 16. Maximum entropy image segmentation: two-threshold. (a) Thresholds = (56, 129) for Image 1 and (b) thresholds = (58, 140) for Image 2.

is accurately identified using a white arrow. The segmentation quality of the λ-connectedness is comparable to that of Maximum Entropy and the λ-connectedness method has excellent stability: only one set of parameters are required for all correlated images.

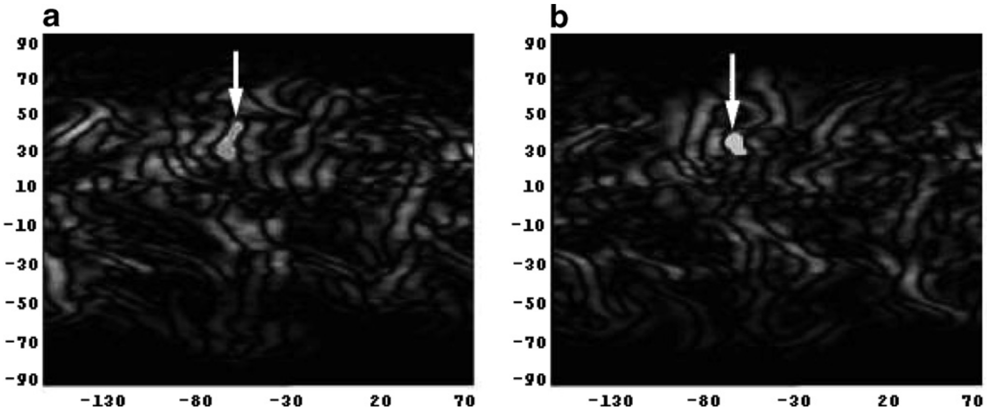

Fig. 17. λ-connectedness image segmentation: (a) based on Image 1 and (b) based on Image 2.

5.3.2. Running time comparison

The running time of the three methods were recorded for both testing images. Table 5 lists the running time of the K-Means for $K = 2$, $K = 4$, and $K = 8$. When $K = 2$, the average image segmentation time is 62 ms and 47 ms for the two images, respectively. The efficiency may be acceptable for low-speed data streams but not appropriate for high-speed data streams. When $K = 4$ and $K = 8$, the running time increases dramatically (around 500 ms), which is apparently not suitable for fast image segmentation.

Table 6 shows the running time of the Maximum Entropy. The single threshold Maximum Entropy segmented the two images in 63 ms and 16 ms, respectively, similar to the running time of K-Means when $K = 2$. However, when two thresholds are employed, the running time jumps to $5 \sim 7$ s, which is not suitable for online processing of data sequences. Table 7 demonstrates the efficiency of the λ-connectedness approach, where $\lambda = 0.95$ and $\theta = 45\%$. Compared with the other two methods, it has the shortest processing time of only 3 ms. In addition, the running time variation between different images is very small.

Table 5
Running time of K-Means

Image	$K = 2$	$K = 4$	$K = 8$
Image 1: data at 0AM September 17, 2003	62 ms	219 ms	1094 ms
Image 2: data at 6AM September 18, 2003	47 ms	188 ms	485 ms

Table 6
Running time of maximum entropy

Image	Single threshold	2 thresholds
Image 1: data at 0AM September 17, 2003	63 ms	7155 ms
Image 2: data at 6AM September 18, 2003	16 ms	5359 ms

Table 7
Running time of *λ-connectedness*

Image	*λ-connectedness*
Image 1: data at 0AM September 17, 2003	3 ms
Image 2: data at 6AM September 18, 2003	3 ms

Based on the above comparisons, we can see that the K-Means is not appropriate for data sequence processing. When $K = 2$ and $K = 4$, the image segmentation quality can not be guaranteed. When $K = 8$, the quality is acceptable, but the efficiency is not satisfactory. The Maximum Entropy achieves satisfactory segmentation quality. If the data arriving rate is not very high, the single threshold entropy method can be used. However, it has a limitation in that both the threshold value and the running time vary significantly with different images. The two-threshold Maximum Entropy provides excellent quality. Nevertheless, the long running time makes it not suitable for stream data processing. The λ-connectedness method can support high processing speed and render acceptable segmentation quality. Moreover, it provides excellent stability: for different images, one single set of parameters can be used and the processing time is steady. Based on the above considerations, the λ-connectedness method was selected to perform the image segmentation.

6. Conclusions

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 is used to filter out noise and highlight spatial variation at specific scales. Then, an efficient image segmentation technique, the 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 are suitable for use in identifying anomalies in continuous meteorological data sequences. The experiments on the Hurricane Isabel data set validate 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 seek to apply our algorithms to the real NOAA online database in order to reveal anomalous meteorological patterns.

Acknowledgements

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

References

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

[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, San Diego, California, USA, June 9–12, 2003, pp. 575–586.

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

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

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

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

[7] W.-T. Chen, C.-H. Wen, C.-W. Yang, A fast two-dimensional entropic thresholding algorithm, Pattern Recognition 27 (7) (1994) 885–893.

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

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

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

[11] N. Djafri, A. Fernandes, N.W. Paton, 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, McLean, Virginia, USA, November 8–9, 2002, pp. 35–41.

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

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

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

[15] J. Gong, L. Li, W. Chen, Fast recursive algorithms for two-dimensional thresholding, Pattern Recognition 31 (3) (1998) 295–300.

[16] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Second ed., Prentice Hall, New Jersey, 2002.

[17] T. Griffiths, A. Fernandes, N.W. Paton, K.T. Mason, B. Huang, 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, Yakohama, Japan, November 27–30, 2001, pp. 84–102.

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

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

[20] J. Haslett, R. Brandley, P. Craig, A. Unwin, G. Wills, Dynamic graphics for exploring spatial data with application to locating global and local anomalies, The American Statistician 45 (1991) 234–242.

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

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

[23] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 881–892.

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

[25] S. Lee, S. Chung, R.H. Park, A comparative performance study of several global thresholding techniques for segmentation, Computer Vision, Graphics and Image Processing 52 (2) (1990) 171–190.

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

[27] Q. Li, T. Li, S. Zhu, C. Kambhamettu, Improving medical/biological data classification performance by wavelet preprocessing, in: Proceedings of the 2002 IEEE International Conference on Data Mining, Maebashi City, Japan, December 9–12, 2002, p. 657.

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

[29] Y. Li, J. Han, J. Yang, Clustering moving objects, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22–25, 2004, pp. 617–622.

[30] P.-S. Liao, T.-S. Chen, P.-C. Chung, A fast algorithm for multilevel thresholding, J. Inf. Sci. Eng. 17 (5) (2001) 713–727.

[31] C.-T. Lu, D. Chen, Y. Kou, Algorithms for spatial outlier detection, in: Proceedings of the Third IEEE International Conference on Data Mining, Melbourne, Florida, USA, November 19–22, 2003, pp. 597–600.

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

[33] Y. Meyer, Wavelets and Operators, Cambridge University Press, 1992.

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

[35] D. Mumford, J. Shah, Optimal approximation by piecewise smooth functions and associated variational problems, Communications of Pure and Applied Mathematics 42 (1989) 577–685.

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

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

[38] Y. Panatier, Variowin: Software For Spatial Data Analysis in 2D, Springer-Verlag, New York, 1996.

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

[40] L. Ramirez, W. Pedrycz, 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, Toronto, Canada, May 13–16, 2001, pp. 87–92.

[41] A. Rosenfeld, The fuzzy geometry of image subsets, Pattern Recognition Letters 2 (5) (1983) 311–318.

[42] T.A. Runkler, J.C. Bezdek, L.O. Hall, Clustering very large data sets: the complexity of the fuzzy c-means algorithm, in: Proceedings of EUNITE 2002, Aachen, Gemany, 2002, pp. 420–425.

[43] G. Sheikholeslami, S. Chatterjee, 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, New York, NY, August 24–27, 1998, pp. 428–439.

[44] S. Shekhar, S. Chawla, A Tour of Spatial Databases, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[45] S. Shekhar, C.-T. Lu, 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, San Francisco, California, USA, 2001, pp. 371–376.

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

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

[48] O. Virmajoki, P. Franti, Fast pairwise nearest neighbor based algorithm for multilevel thresholding, Journal of Electronic Imaging 12 (4) (2003) 648–659.

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