# Parallel Wavelet Transform for Spatio-temporal Outlier Detection in Large Meteorological Data

Sajib Barua[1] and Reda Alhajj[1,2]

[1] Computer Science Dept, University of Calgary, Calgary, Alberta, Canada
[2] Department of Computer Science, Global University, Beirut, Lebanon
`{baruas,alhajj}@cpsc.ucalgary.ca`

**Abstract.** This paper describes a state-of-the-art parallel data mining solution that employs wavelet analysis for scalable outlier detection in large complex spatio-temporal data. The algorithm has been implemented on multiprocessor architecture and evaluated on real-world meteorological data. Our solution on high-performance architecture can process massive and complex spatial data at reasonable time and yields improved prediction.

## 1 Introduction

This paper introduced a novel approach to locate outlier in meteorological data, which are collected over time and space. Meteorological data are spatio-temporal data because of their multidimensional properties, structures (geometrical shape), distribution over space and variation with time. Techniques for accommodating semantic (identity and attributes), spatial (geometry, location and topology), and temporal (time of occurrences and observation) properties of meteorological events try to resolve theoretical and computational issues concerning the spatial modeling of meteorological data. Processing meteorological data to forecast severe events has always been a challenging problem over the last decades because of several reasons: 1) the variability found in the properties of the meteorological data; 2) the data is complex, massive, and has spatial properties; 3) the solution strategy varies with the location and type of the collected data. Data mining has demonstrated enormous potential as a processing tool for meteorological data. While major data mining techniques try to find general patterns in the data set, outlier or deviation detection draws attention to finding exceptional, dissimilar and inconsistent patterns compared with the rest of the data. An object is defined as a spatio-temporal outlier if its non-spatial attribute is significantly different from its neighbors in spatial relation and/or from other observations in a given time period. As weather data is concerned, outliers indicate anomalies or severe events such as tornadoes and forest fires that tend to occur in an area. Therefore, identifying spatio-temporal outliers is an essential problem with both scientific and social impacts. Neglecting outliers may lead to disasters like those that lastly happened in Asia and the United states.

Mining meteorological data for identifying spatio-temporal outliers needs to consider the following aspects in general: 1) high dimensionality with arbitrary and complex data types and large data sets at high resolution; 2) the data is geographically distributed over space and time.

Although many works are done in spatial outlier detection, e.g., [5, 9, 10, 12], very few of them focus on spatio-temporal outliers. For instance, Tao *et al* [4] extended the spatial outlier mining to capture the semantic and dynamic aspects of spatio-temporal data in multi scales. They adopted multi-resolution clustering algorithm based on semantic knowledge of ST-objects and applied them to find outliers in multi-scale property of geographic phenomena. Birant *et al* [3] improved DBSCAN algorithm to capture the temporal aspects of ST-objects and introduced a scale to measure density of each cluster. Ramachandran *et al* [8] developed a flexible framework ADaM.

High performance computational architecture not only allows analysis at multiple spatial scales and locales, but also improves more realistic simulation of extreme events at high resolution to provide climate information for resource management and impact assessment. In this direction, wavelet transform (WT) can locate the frequency change (property variation) because of its multi-resolution and multi-scale properties; hence performs well in detecting the boundaries of outlier regions. Weather data (temperature, vapor distribution, rainfall) has spatial properties because of its multidimensional features and geometric pattern. Therefore, we adopt Parallel Wavelet Transform (PWT) algorithm for outlier detection in large scale spatial data to take the advantage of the computational power on the multiprocessor machine. Our parallel approach also tries to tolerate the architectural latencies by overlapping communication with computation.

The rest of the paper is organized as follows. Section 2 presents a framework for outlier mining using parallel wavelet transform. Section 3 presents experimental results. Section 4 is conclusions.

## 2 Parallel Spatio-temporal Outlier Detection

In the real atmosphere, anomalies occur at different spatial and temporal scales. Therefore, our task involves defining a region outlier as a group of adjoining points whose features are inconsistent with those of their surrounding neighbors or time frame. Spatial properties can not be analyzed with a uniform scale. Wavelet analysis allows the analysis data at different scales for two important properties. 1) **Multi-resolution**: wavelet analysis uses stretched wavelet to capture the coarser view and compressed wavelet to capture the detailed information in signal; 2) **Localization of the frequency**: WT can also detect the frequency change in the time domain.

Spatial outliers appear as small regions; hence are local outliers in most cases. On the other hand, temporal outliers are defined based on the duration of the time domain. Detection of events like El Niño and La Niña focus on outliers in long time period, whereas climatical changes in a month focus on finding local outliers. We use Daubechies $D4$ WT to analyze the data.

### 2.1 Calculation of Daubechies $D4$ Wavelet Transform

The $D4$ transform has four wavelet and scaling function coefficients: $h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}$,$h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}$, $h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}$, and $h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$. The wavelet function coefficient values are: $g_0 = h_3$, $g_1 = -h_2$, $g_2 = h_1$, and $g_3 = -h_0$. For a data set of $N$ values, the scaling function calculates $\frac{N}{2}$ smoothed values, and the wavelet function calculates $\frac{N}{2}$ differences at each step of the wavelet transform. In the ordered wavelet transform, the smoothed values are stored in the lower half of the $N$ elements input vector. Each step of the wavelet transform applies the wavelet function to the input data. If the original data set has $N$ values, the wavelet function will be applied to calculate $\frac{N}{2}$ differences (reflecting change in the data). In the ordered wavelet transform, the wavelet values are stored in the upper half of the $N$ elements input vector. The scaling and wavelet functions are calculated by taking the inner product of the coefficients and four data values. Daubechies $D4$ scaling function is $s'_i = h_0 s_i + h_1 s_{i+1} + h_2 s_{i+2} + h_3 s_{i+3}$ and Daubechies $D4$ wavelet function is $s'_{i+1} = g_0 s_i + g_1 s_{i+1} + g_2 s_{i+2} + g_3 s_{i+3}$.
**Forward Transform:** Each iteration in WT calculates a scaling function value and a wavelet function value. The index $i$ is incremented by two with each iteration, and new scaling and wavelet function values are calculated. In case of the forward transform, with a finite data set, $i$ is incremented until it is equal to $N - 2$. In the last iteration, the inner product is calculated from $s[N-2]$, $s[N-1]$, $s[N]$ and $s[N+1]$. Since $s[N]$ and $s[N+1]$ don't exist (they are beyond the end of the array), this presents a problem.

### 2.2 Parallel Daubechies Fast Wavelet Transform

Different authors worked on the design and implementation of classical wavelet construction. Our implementation is on distributed memory architecture, where each processor has fast access to its own memory. The important issues for an efficient parallel algorithm are load balancing and communication latency, which refer to an efficient distribution of task and data over multiple processors to minimize the intercommunication among them.
**Data Distribution:** The data distribution strategy for a parallel algorithm is greatly affected by the computation pattern at each stage. In sequential algorithm, we observe that at each level, input data are generated from the low pass filter of the previous level. After each level computation, redistribution on the data is done to bring the results from the low pass filter consecutive and they appear as input for the next level. Redistribution ensures data locality for next level computation and reduces inter-communication cost if data required for computation resides in remote processors. But, it requires data swapping among processors, which incurs a huge amount of communication and load balancing will be poor. The number of input data is halved at each level; hence, such kind of redistribution will only bring the input data local to some of the processors, leaving the rest of them totally idle.

We avoid the redistribution and achieve a proper load balancing by assigning equal number of inputs to each processor. If there are $N$ number of inputs and

$P$ number of processors, each processors in blocked data distribution will get $\frac{N}{P}$ number of inputs. At each level, the number of inputs is reduced by half and the process is continued until one input is available. Assuming $N$ and $P$ as powers of 2, the total number of levels required is $\lg N$.

**Computation:** The computational cost to get a new value is 4 multiplications and 1 summation. On single processor, for $N$ values each stage requires $4N$ multiplications and $N$ summations. Whereas if we employ $P$ processors, total number of multiplications and summations will be $4\frac{N}{P}$ and $\frac{N}{P}$, respectively, which is a great reduction in computational cost. $N$ should be greater than $P$ and for better performance, at initial stage we assume, each processor should have at least 4 input values. As we continue, the number of input values for the next stage will be reduced by half. The total number of stages which have at least one input value are $\lg\frac{N}{P}+1$. Among them, at the last stage each processor gets only one input value and the rest get more than one values. For $(\lg N - (\lg\frac{N}{P}+1)$ stages, not all processors have values for computation and will remain idle. Therefore, we have load balancing in major number of stages.

**Communication:** Each processor needs one send and one receive. But, when each processor has only one element to compute, the number of sending or receiving will be 3 as sent or received values are directed to or coming from 3 different processors. Therefore, $(\lg N - \lg\frac{N}{P}) = \lg P$ number of stages require 3 sending and 3 receiving if the processor has value to compute. As this number increases with the increase of $P$, therefore for a data set $(N)$, a specific value of $P$ will give better load distribution, where $\lg\frac{N}{P}$ will be higher and communication cost (for $\lg P$ stages) will never be dominating over the computational cost.

### 2.3 Parallel Outlier Mining

The wavelet analysis algorithm uses a parallel wavelet algorithm on climate data in order to discover regions with prominent spatial or temporal variation at different scales. A set of scales is provided by the domain experts beforehand. We are also given the beginning and ending latitude for our analysis. The wavelet analysis is performed on the dataset recorded along the latitude range to discover regions with prominent spatial variation at different scales. The wavelet value for each data point is compared against a threshold value ($w$). If the value is higher than $w$, we consider the data at that point as suspected outlier and record the location (latitude, longitude). In spatial domain, suspected outliers are grouped together using $Z$-value approach [11] and form outlier region.

```
WAVELET ANALYSIS FOR OUTLIER MINING
Input:-- t1: the beginning latitude (or time); tn: the ending latitude (or time);
         S: a set of selected scales; w: a pre-defined threshold wavelet power;
         X: a given data set; SuspectSet: a set of points to be outliers;
Output:-- Y: a set of outlier regions;
/*Calculation of wavelet power along all latitudes or time*/
for ( i = t1; i < tn ; i++){
   wTransform = ParallelWTransform(X,S,i);}
/* Identify point which wavelet power is higher than the thresold*/
for each p in wTransform{
   if (p > w){CandidateOutlierSet(p, CandidateSet)}}
/* Form regions using statistical approach*/
Y = FormRegionOutlier(CandidateSet); Output(Y);
```

## 3 Experimental Result

We conducted our experiment on sea surface temperature (SST) dataset collected from National Climatic Data Center (NCDC) NOMADS LAS server. We performed the spatial analysis of daily optimum interpolation (OI) SST on a $\frac{1}{4}$ degree grid at different locations (e.g., equatorial pacific). The temporal analysis is performed on real-time SST data from moored ocean buoys collected from Tropical Atmosphere Ocean (TAO) project. Our temporal outlier detection tested SST data at different temporal resolutions from 10 minutes to daily basis for locations on the equator. The tests have been conducted on WestGrid. The PWT algorithm was implemented in C with MPI library and tested on a distributed platform nexus et el. hosted by the University of Alberta. This is a collection of SGI SMP machines, ranging up to 256 processors; for our experiment, we used 32 processors. Finally, we present the data analysis and then show the scalability of the outlier detection algorithm on parallel architecture for data at high resolution spatio-temporal scale.

### 3.1 Spatial Analysis

In meteorological data, outliers appear as arbitrary small shapes or regions (eye of a cyclone), where a sharp temperature or vapor distribution change occurs. To compute this climatological change for a location, we need to identify its neighborhood and compute the feature difference. In GIS, spatial resolution is defined as the ability to define sharply and clearly the extent or shape of features within an image or area on the ground by an imaging system, such as a satellite sensor. At the highest resolution, more details become clear, and the information is sharper. Spatial scale is used to define the resolution. Such division is more or less arbitrary; for example, micro is the smallest unit which may involve $1^\circ$ degree or $1m - 1km$ scope; and mega is assigned the global scope, which may involve the whole planet. We worked on data gathered at $\frac{1}{4}^\circ$ interval across latitude, and each point on the latitude is the average SST of the $2^\circ$ north and south longitudinal positions. As our focus is on latitude climatology change, for a location the neighborhood encompasses $1^\circ$ east and west across latitude, and $2^\circ$ north and south across longitude. For a latitude-longitude plan, we apply our algorithm for each latitude at $2^\circ$ intervals and combine the outlier regions.

Fig. 1 plots the SST temperature over South Pacific (Fig. 2) recorded on Nov 28, 2006; we have shown some of the outlier locations ($A$, $B$, $C$ and $D$) in rectangles. Each of $A$, $B$ and $C$ is local outlier compared to its neighborhood region, and $C$ and $D$ can be considered as global outliers for their highest and lowest value, respectively. These locations are more obvious from the contour plot (Fig. 3) collected from NOAA/PMEL. Our algorithm can find these outlier locations from the wavelet decomposition in time-scale map. Sharp changes (high frequency or bursts) in the temperature are captured at low scales and higher time resolution window. Whereas a trend (low frequency), such as a particular temperature continuing for a longer period, is captured at high scale and poor time resolution window. In Fig. 4, we observe such phenomena in approximation

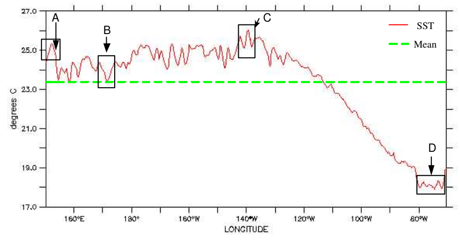

**Fig. 1.** Daily optimum interpolation (OI) SST ($^{\circ}C$): time 01-Nov-06

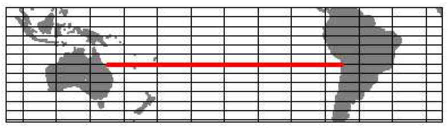

**Fig. 2.** South pacific: lat $20.6^{\circ}S$, long: $150.1^{\circ}E$ to $70.4^{\circ}W$

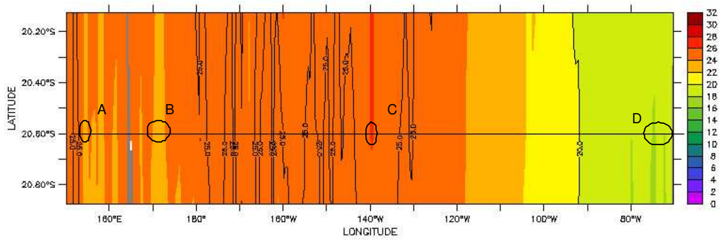

**Fig. 3.** SST contour plot

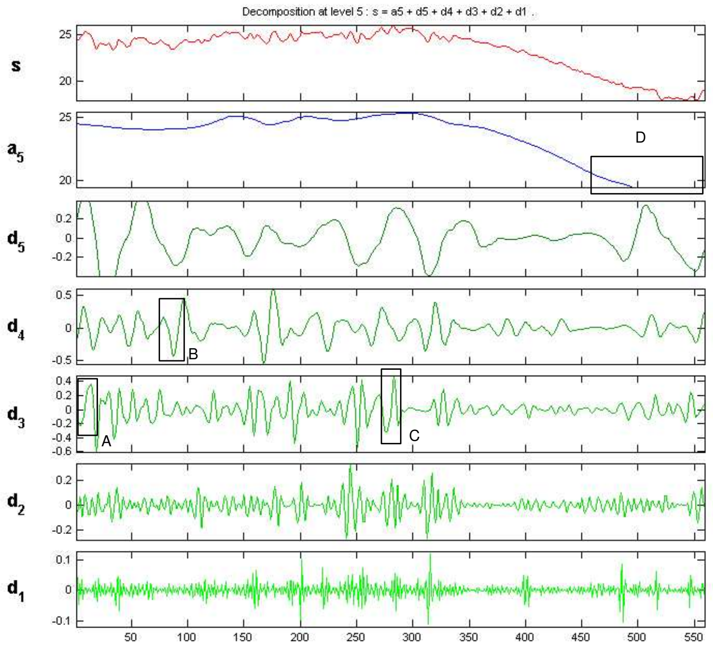

**Fig. 4.** Wavelet decomposition

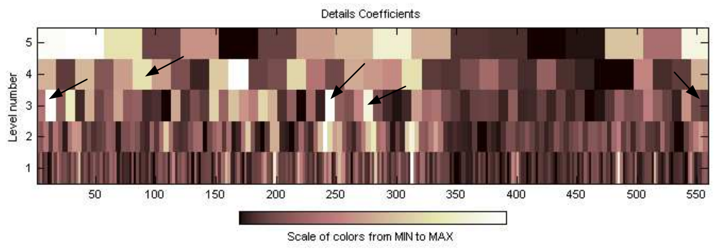

**Fig. 5.** Scalogram

level $a_5$ at interval $140-300$), which corresponds to location $179.9°E-132.6°W$ when the temperature is between $26-24°C$. Sharp SST changes appear in the detailed part of the decomposition at different scales. For example, local outliers $A$ and $C$ are clearly detected at scale 3 and $B$ in scale 4. At each scale, peaks appearing in detailed part of the decomposition are sharp SST; changes for short time duration are also good candidates of local outliers. Again, global outlier location $D$ is detected in the approximation level at scale 5.

Neighborhood difference is clearly observed in the scalogram analysis. Brighter region gives higher similarity of the feature with the wavelet and darker region gives greater dissimilarity. At a particular scale, a brighter region surrounded by darker regions in color intensity is a good candidate of outlier. Some outliers are shown in Fig. 5 using arrows which are detected as outliers (Fig. 1) at the same location for particular scale.

### 3.2 Parallel Algorithm

In this section, we show the scalability of the proposed PWT algorithm over the sequential algorithm for outlier detection. We validated the scalability of PWT for a dataset of mean temperature collected over a time period (1991-1999) by NOAA NCDC daily GLOBALSOD: Global daily WMO weather station data.

We implemented both sequential and parallel wavelet transform algorithms. The block data distribution approach in our parallel algorithm gives the first $\lg\frac{N}{P}-1$ stages computation intensive and the rest $\lg N-(\lg\frac{N}{P}-1)$ stages communication intensive. For the first $(\lg\frac{N}{P}-1)$ stages, input data required to compute the coefficients are local in the processor except two input data located in neighboring processors. As the communication is fixed and small, these stages give scalability with the increase of data volume. At $\lg\frac{N}{P}$-th stage, each processor will have one value to compute and communication will be 6. As the number of data is reduced by 2 at each level, after $(\lg\frac{N}{P}+1)$ stages, half of the processors will be idle and others will have one value to compute and number of communications is still 6. Therefore, these communication intensive stages will not contribute much to scalability. Here, scalability can be ensured in two ways. First, the outlier mining algorithm analyzes data up-to a particular number of scales (level). Spatial outliers are in general local outliers based on the neighborhood resolution, hence smaller scales are adequate to capture the

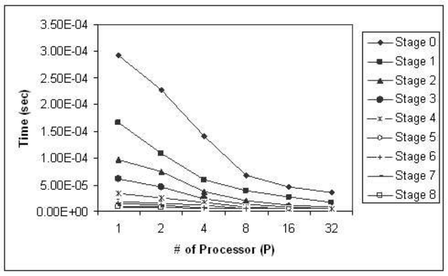

**Fig. 6.** Execution time in computation intensive stage

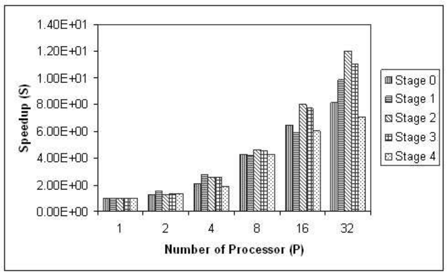

**Fig. 7.** Speedup for $N = 1024$

high frequency feature change in the data. The last few communication intensive stages generate very few coefficients. They are used to detect the trend in data, and in most cases are insignificant for the outlier mining application. Secondly, with huge dataset, the time efficiency achieved at computation intensive stages will be more to overlook the time required in the communication intensive stages.

**Table 1.** Timing analysis for communication intensive stages

| Stage | Required time (sec) in processors | | | | |
|---|---|---|---|---|---|
| | 2 | 4 | 8 | 16 | 32 |
| 5 | | | | | $1.04E-05$ |
| 6 | | | | $1.20E-05$ | $2.08E-05$ |
| 7 | | | $1.52E-05$ | $1.84E-05$ | $8.00E-07$ |
| 8 | | $2.08E-05$ | $1.60E-06$ | $8.00E-07$ | $7.99E-07$ |
| 9 | $1.28E-05$ | $2.40E-06$ | $8.00E-07$ | $8.00E-07$ | $8.00E-07$ |

Fig. 6 shows the execution time for 1024 discrete data points with different number of processors for the first $\lg \frac{N}{P}$ stages. From the plot, we find that with the increase in processors, the required time is decreasing. We also conducted the time analysis for communication (Table 1) intensive stages by varying the number of processors. In most cases, the spent time in a processor is approximately $8.00E-07$ sec and sometimes deviation is also observed, depending upon the speed of the network and distribution of nodes in the cluster. To realize the impact of communication latency on total time, we can compare the execution time required in computation and communication intensive stages. With 16 processors, at stage 1, the time required for 32 coefficients computation and 2 communications is $2.80E-05$ sec, whereas at stage 7 the required time for 1 coefficient computation and 6 communications is $1.84E-05$ sec.

As explained in Section 2.2, the number of communication and computation intensive stages depend on $\lg P$ and $\lg \frac{N}{P}$, respectively. Therefore, with higher value of $N$, the impact of $\lg P$ stages is tolerable, otherwise a specific value of $P$ can only give scalability for small value of $N$ in the total time for communication and computation intensive stages. For large complex spatial datasets, the computational cost will be higher and communication cost will be lower compared

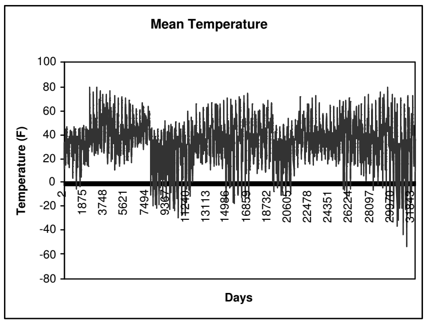

**Fig. 8.** Mean temperature

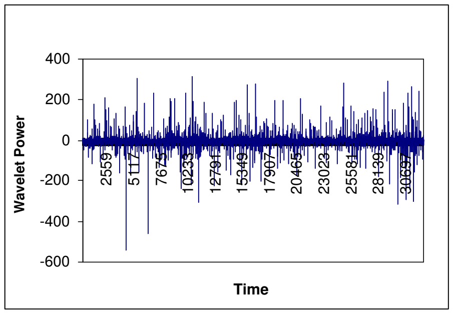

**Fig. 9.** Wavelet coefficients

to the computational cost. In such scenario, increasing the number of processors will help in ensuring more scalability. Yet a specific number of processors for a data size will give the best result. In Fig. 7, we observe speedup increase at each stage with the increase in processors. This can be explained from Fig. **??**, where we have shown the computation for 32 data points with 4 processors. For the first 4 stages, each processor sends two border data to its predecessor processor and receives two data (which are border data too) from the neighboring processor. The communication cost required is the sum of one send and one receive. Each data point needs 4 multiplications and 1 summation, and at level $i$, the number of data points computed is $\frac{N}{P \times 2^i}$. If $t_{mul}$ and $t_{add}$ denote the computational time for one multiplication and one addition, respectively, the time required to compute one data point is $4 \times t_{mul} + t_{add} = t_{comp}$. Therefore, the total computational time required in a processor at level $i$ is $\frac{N}{P \times 2^i} \times t_{comp}$. If $t_{comm}$ denotes communication time, each processor spends $2 \times t_{comm}$ for communication. Hence, the total time incurred in a processor is $\frac{N}{P \times 2^i} \times t_{comp} + 2 \times t_{comm}$. With the increase of $P$, the first part (computation) of the above expression is decreased, but communication time will remain the same and the newly added processors will add more communication. Each stage starts with the completion of the previous stage and any delay in the interprocess communication time of the previous stage delays the start of next stage, hence the total time (for all stages) is increased. But for large datasets, the computational cost is higher than communication. Hence, the time efficiency achieved from computation intensive stages is more to neglect any anomaly or a constant increase of communication and will give more scalability with increase in processors.

We used parallel wavelet transform algorithm to calculate wavelet power for both datasets. Fig. 9 plots the wavelet power of the mean temperature (Fig. 8). Our db4 parallel wavelet algorithm decomposed the signal up-to 15 levels. The wavelet coefficients at the 15$^{th}$ level accumulate the detailed and approximate parts of the decomposition. In Fig. 9, the spikes are anomalies from the regular temperatures. They are deviated from the general trend of temperature pattern and are treated as outliers. If we watch carefully, we will see that in most cases, more than one spikes are grouped in one place, thus forming a cluster.

These small groups are forming our suspected outliers, meaning the temperature deviation for few days.

## 4 Conclusions

In this paper, we applied wavelet analysis for outlier mining to take the advantage of multi-scale capability and multi-resolution feature. Zhao *et al* [14] used a wavelet based approach and they used Morlet and Mexican Hat mother wavelet. Whereas, Daubechies gives better frequency resolution and we used discrete fast wavelet transform which is more suitable to implement. Further, parallelization gives speedup when the spatial data objects are very large. So far, we deal with two features (temperature, latitude or time) in the weather data; considering its multi-dimensional attribute domain (e.g., vapor, air pressure, etc.) at the same time poses a challenging issue for meteorological forecasting. We also plan to improve the parallel algorithm for different load distributions, granularity, and data locality issues to reduce communication latency. We will test the outlier mining algorithm using other statistical (e.g., iterative-Z value) and visualization (e.g., variogram clouds, pocket plots, scatter plot) approaches.

## References

1. S. Barua and R. Alhajj, "High Performance Computing for Spatial Outliers Detection Using Parallel Wavelet Transform," *Intelligent Data Analysis,* (in press).
2. S. Barua and R. Alhajj, "A Parallel Multi-scale Region Outlier Mining Algorithm for Meteorological Data," *Proc. of ACM GIS,* 2007.
3. D. Birant and A. Kut. Spatio-temporal outlier detection in large databases. *Journal of Computing and Information Technology*, 14(4):291-298, 2006.
4. T. Cheng and Z. Li. A multiscale approach for spatio-temporal outlier detection. *Transactions in GIS*, 10(2):253-263, Mar 2006.
5. M. K. Edwin and T. N. Raymond. A unified notion of outliers: Properties and computation. In *Proc. of ACM-KDD*, pp.219-222, 1997.
6. E. Hung and D. Cheung. Parallel algorithm for mining outliers in large database. In *Proc. of IDC*, 1999.
7. E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. of VLDB*, pp.392-403, 1998.
8. R. Ramachandran, J. Rushing, H. Conover, S. Graves, and K. Keiser. Flexible framework for mining meteorological data. In *Proc. of IIPS for Meteorology, Oceanography, and Hydrology*, Feb 2003.
9. S. Ramaswamy, P. Alto, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. of ACM SIGMOD*, pp.427-438, 2000.
10. S. Shekhar, S. Chawla, S. Ravada, A. Fetterer, X. Liu, and C. Lu. Spatial databases-accomplishments and research needs. *IEEE TKDE*, 11(1):45-55, 1999.
11. S. Shekhar, C.-T. Lu, and P. Zhang. A unified approach to detecting spatial outliers. *GeoInformatica*, 7(2), 2003.
12. T. L. V. Barnett. *Outliers in Statistical Data.* John Wiley, New York, 1994.
13. D. Yu, G. Sheikholeslami, and A. Zhang. Findout: finding outliers in very large datasets. *Knowl. Inf. Syst.*, 4(4):387-412, 2002.
14. J. Zhao, C. Lu, and Y. Kou. Detecting region outliers in meteorological data. In *Proc. of ACM International Symposium on Advances in GIS*, pp.49-55, 2003.