

Model-Based Event Detection in Wireless Sensor Networks

Jayant Gupchup, Randal Burns, Andreas Terzis
Department of Computer Science
Johns Hopkins University
3400 N. Charles St
Baltimore, MD 21218
{gupchup, randal, terzis}@jhu.edu

Alex Szalay
Department of Physics and Astronomy
Johns Hopkins University
3400 N. Charles St
Baltimore, MD 21218
szalay@jhu.edu

ABSTRACT

We present an application of statistical signal processing techniques to the problem of event detection in wireless sensor networks used for environmental monitoring. The proposed approach uses the well-established Principal Component Analysis (PCA) technique to build a compact model of the observed phenomena that captures daily and seasonal trends in the collected measurements. We subsequently use the divergence between actual measurements and model predictions to detect the existence of discrete events within the collected data streams. Our preliminary results show that this event detection mechanism is sensitive enough to detect the onset of rain events using the temperature modality of a wireless sensor network.

1. INTRODUCTION

A number of testbeds (*e.g.*, [1, 2, 3]) have shown the potential of wireless sensor networks (WSNs) to collect environmental data at previously unimaginable spatial and temporal densities. At the same time, these developments present many novel data management challenges. First, our experience deploying an environmental monitoring network has demonstrated the shortcomings of the static behavior of current sensor networks. For example, scientists would like to sample the environment at a high frequency to capture detailed information about “interesting” events, but doing so continuously would create an inordinate amount of data. On the other hand, sampling at a lower frequency generates less data but can potentially miss important temporal transients. Second, the large amount of data that these networks generate complicates the querying and post-processing stages. Rather than manually traversing through the collected data, scientists would prefer to query for measurements related with certain events (*e.g.*, significant rainfall).

To address these issues, we need WSNs that can reason about the phenomena they observe and adapt their behavior based on events they detect. Possible adaptation strategies include changing the sampling rate as well as waking up nodes in the network to increase spatial coverage of the detected event [4, 5].

Sensors measure the superpositions of several processes driving

the phenomena under observation. These processes are often dominated by predictable foregrounds, which can be significantly larger than the subtle trends and variations that scientists are trying to measure or the small events that they try to detect. In order to interpret the measurements, it is then important to separate these different signals into independent components. In environmental monitoring, most sensors witness daily variations of all quantities and seasonal trends. In addition, there are discrete natural events (*e.g.*, storms, rainfall, and strong winds) that have a separable effect on our data.

We present an approach using techniques of statistical signal processing to decompose the sensor readings into various physically meaningful components. In our approach, we perform a step-by-step identification of various foregrounds. We identify the diurnal cycle present in both the air and soil temperature sensor data and we account for the effect of seasonal drift. We use all these priors (daily cycle, seasonal drift) to detect events by identifying when measurements diverge from those expected by the foregrounds.

Specifically, we explore variants of the Principal Components Analysis method (PCA) [6] to extract features from the data collected by the network and discover the multiple underlying physical processes that generate the observed data. This process produces a *model* of “normal behavior.” Observations that diverge from the model correspond well with punctuated events. We note that one can build the PCA model offline using historical data and that a small number of parameters summarize the phenomena that the motes sense. Such a compact representation of the model enables the design of a lightweight event detection mechanism that runs in real time on the network’s motes.

We evaluate the performance of the proposed mechanism using data from the Life Under Your Feet environmental sensing network [1]. We execute the event detection algorithm to detect rain events within the deployment area over ten months of the network’s lifetime. We compare the list of detected events with precipitation data recorded by a weather station at BWI airport.

This specific application reveals another aspect of the proposed approach: while the motes in our network have soil moisture sensors, these sensors cannot detect the onset of a rain event, because soil moisture rises only after the water seeps through the soil. Instead, we use a combination of air and soil temperature measurements to detect when rain starts to fall. Figure 1 indicates that temperature varies immediately with the onset of an event, while soil moisture lags by at least an hour. The model allows us to detect the rain event rapidly based on indirect evidence prior to the rain’s direct effect on soil moisture.

All the data and code used in this paper are publicly available at: <http://lifeunderyourfeet.org/en/src/>

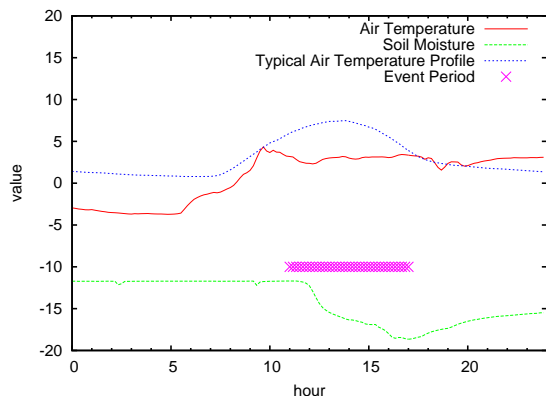


Figure 1: Air temperature is a better indicator of the onset of a rain event compared to soil moisture.

1.1 Environmental Sensing

While our solution generally applies to WSNs that collect large amounts of data through multiple modalities, we present our design through an environmental monitoring application we developed and was deployed for over 18 months at an urban forest in Baltimore, MD. The purpose of the *Life Under Your Feet* network is soil monitoring in which each of the network’s ten motes periodically collects measurements, including soil temperature and soil humidity, as well as ambient temperature and light.

We also extract weather information (air temperature and rain events) from a weather station at the BWI airport located 25 miles away from our deployment site. The data scraping program we use inserts this data into the same database, allowing meteorological information, such as rain duration and amount of rainfall, to be correlated to the data collected by the sensor network.

2. RELATED WORK

PCA event detection constructs a model of system behavior. We consider two applications of model-based event detection in describing related work. The first is an offline variant in which event detection happens at the database that stores the measurements collected by the network and is used to automatically identify “interesting” regions within the swaths of data acquired by the sensor network. The other is online in that motes in the network use models to detect events and subsequently alter their behavior.

Offline event detection provides a model suitable for querying events from noisy and imprecise data. Both database systems [7, 8] and sensor networks [9, 10, 11] have explored model-based queries as a method for dealing with irregular or unreliable data. Models in these systems include Gaussian processes [9], interpolation [12, 13], regression [9, 14], and dynamic-probabilistic models [8, 10]. We provide a PCA-based model specifically suited to event detection.

In the online case, sensor networks reduce the bandwidth requirements of data collection by suppressing results that conform to the model or compressing the data stream through a model representation. This has coincident benefits on resource and energy usage within the network. If sensors measure spatially correlated values, values collected from a subset of nodes can be used to materialize the uncollected values from other nodes [15, 16]. Similarly, temporally-correlated values may be collected infrequently and missing values interpolated [10, 17]. By placing models in the mote itself, the mote may transmit model parameters in lieu of the

data, compressing or suppressing entirely the data stream [18, 19, 20]. Our PCA model may be used for suppression and compression and may also be used to alter the behavior and configuration of the network, *e.g.* only collecting data when events occur and turning off large portions of the network at other times.

Most research on event detection describes data fusion and in-network event processing, rather than the detection of an event based on the data. REED provides in-network joins to report event conditions that are programmed declaratively [21]. Other systems ensure that multiple sensors detect an event prior to reporting it [22, 23]. Our work focuses on using PCA models to rapidly and accurately report an event at a single mote. This single mote report serves as an input to fusion and event query evaluation. Other ecological monitoring systems use simple rising edge or trigger/threshold based event detectors at each mote [24].

We use PCA to determine that a single or a sequence of measurements are *dissimilar* to the normal behavior of the system, characterized by its principal components. Similar uses of PCA include anomaly and intrusion detection in computer networks [25, 26] and leakage detection in gas sensor arrays [27]. Recently, PCA has been applied to event detection in the Internet, specifically identifying correlated throughput and loss events on multiple Internet paths [28]. However, the authors provide no details of their approach. There is a wealth of literature on the application of PCA for process control and process monitoring. [29] is one of many papers that address the application of PCA to process control, often referred to as multivariate process control. Moreover, PCA finds application as a tool for visualizing outliers in multidimensional data described by a large collection of similar curves [30].

Finally, modeling diurnal cycles exhibited by atmospheric temperature is a well studied problem in the environmental and atmospheric sciences community. Smith et al. [31] describe how they model surface temperature using empirical orthogonal functions (EOFs) and use the projections on the orthogonal functions for extracting features and analyzing the temperature variability from different regions. EOFs are nothing but the basis vectors obtained from PCA.

3. METHODOLOGY

Principal component analysis [6], also known as the Karhunen-Loève transform (KLT), is a powerful statistical tool for simplifying data by reducing high-dimensional datasets into datasets with lower dimensions that approximate the original data. It does so through singular value decomposition (SVD): an orthogonal linear transform of a matrix containing the original data into an equivalent diagonalized matrix.

Mathematically speaking, let \mathbf{X}^T be an $n \times m$ matrix, in which each row represents a new set of measurements and each column represents the measurements at a given time of day. In other words, the number of columns represents the dimensionality of the original space. The PCA transformation is then given by $\mathbf{Y}^T = \mathbf{X}^T \cdot \mathbf{V}$, in which $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \text{SVD}(\mathbf{X}^T)$, $\mathbf{\Sigma}$ is the $m \times m$ diagonal matrix containing the singular values of \mathbf{X}^T , \mathbf{V} is the $m \times m$ matrix of right singular vectors that form an orthonormal basis. In this paper, we also refer to these as our *basis vectors*. \mathbf{U} is the $n \times m$ matrix containing the left singular vectors. Then, \mathbf{Y}^T is the projection of the original data on the basis vectors which is another $n \times m$ matrix.

In practice, we select the first $p \ll m$ eigenvectors with maximum eigenvalues. These eigenvectors represent the “most important” dimensions in that these dimensions have the maximum variance and strongest correlation in the dataset. We then calculate another projection $\mathbf{Y}_p^T = \mathbf{X}^T \cdot \mathbf{V}_p$, where \mathbf{V}_p is generated by taking the first p columns of \mathbf{V} (*i.e.* the columns that correspond to the

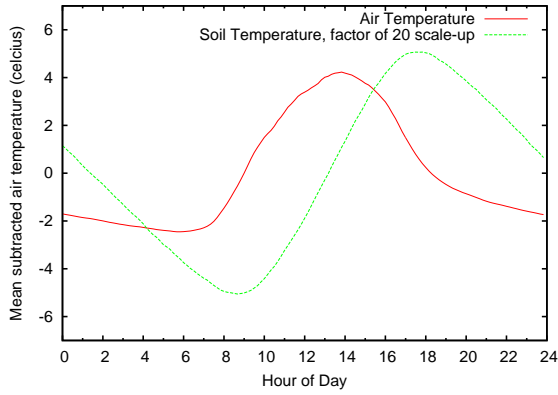


Figure 2: Mean subtracted profile of air and soil temperature (latter scaled up by a factor of 20) for a typical 24 hour cycle.

highest p eigenvalues). The dimensionality of \mathbf{Y}_p^T is $n \times p$, where p is the dimensionality of the resulting subspace.

Thus, the original dataset can be reduced to just those dimensions (eigenvectors) with large eigenvalues. Data analysis may be performed in the lower dimensional representation with good fidelity to results on the original data. The lower dimensional space offers benefits not only in data size, computational complexity, and ease of visualization, but also these vectors represent the “typical” patterns of the data, whereas the residuals correspond to “atypical” behavior.

3.1 Applying PCA to sensor measurements

We apply PCA to air temperature and soil temperature sensor readings. Sensor readings exhibit typical diurnal cycles, which dominate every other signal present. Figure 2 shows the mean-subtracted profile of a typical day for air temperature and soil temperature. We note the rise in temperature as the sun rises in the morning and the fall in temperature as the sun sets in the evening for air temperature. We also observe that soil temperature changes lags air temperature changes by several hours, owing to the inertia of the soil. There is a noticeable phase shift between air temperature and soil temperature. This pattern (AC component) is exhibited by all normal (non-event) days of all seasons around the average value (DC component) for that day.

LUYF sensors record measurements once every minute. We aggregate multiple readings, which produces a data series with readings every ten minutes. We find empirically that a ten minute average reveals useful information from the data. It smooths transients, yet samples at a relatively high frequency. This data series is then converted into a data matrix such that each row vector represents the data a sensor collects during a day, from midnight to midnight. In a given day, we have 144, ten minute intervals.

We normalize the data prior to building the model in order to best characterize the “normal” behavior of the system. To do so, we subtract the mean temperature of that given day (calculated separately for each sensor) from each of these row vectors and normalize the readings in the RMS sense. Using normalized vectors ensures that the diagonal elements of the correlation matrix are unity. Thus, each vector contributes equally to the PCA basis. This balances the contribution of summer and winter to the model even though summer days have higher variance. In order to obtain a well-behaved basis, we censor the days which have considerable inherent noise and jitter from our training set. We apply a simple median filter to remove these “bad” days.

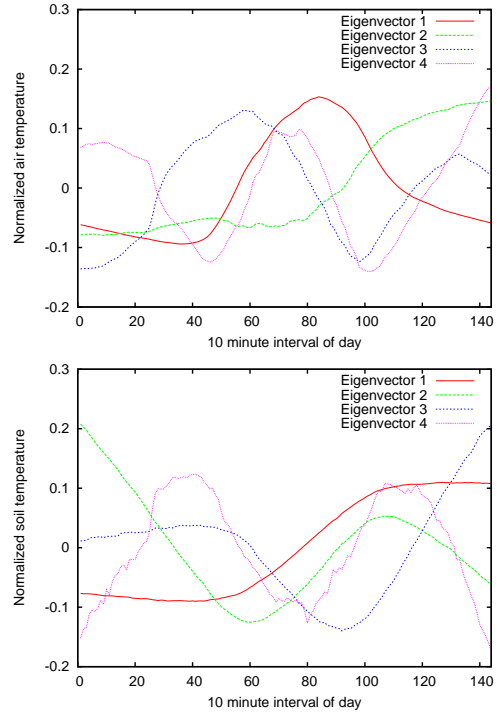


Figure 3: Daily temperature eigenvectors (basis vectors) in decreasing order of eigenvalues. The top panel shows the eigenvectors for the air temperature while the lower panel displays the eigenvectors for soil temperature.

After cleaning the data, we compute the orthogonal eigenvectors (basis vectors) and order these vectors by decreasing eigenvalues as described at the start of Section 3. Figure 3 shows the first four eigenvectors obtained for air temperature and soil temperature for the period of September 2005 to July 2006. We find that the first four eigenvectors cover 90.95% of the total variation in the air temperature data and 98.89% in the soil temperature data (as defined by the sum of the first four eigenvalues of the diagonal matrix divided by the trace). We note that the first eigenvector accounts for 55% of the total variation in the air temperature data.

The physical meaning of the different eigenvectors can be interpreted in the following way. The first component of the air temperature is a bell shape curve, corresponding to the slow rise of the temperature around 7 am, then cooling after 3pm. The second eigenvector is rising throughout the day monotonically, describing a warming/cooling trend from one day to another. The third vector causes the bell shaped curve of the temperature to slide forward or backward, representing the effect of the seasonal warming and cooling. Finally, the fourth eigenvector is the broadening and shortening of the daily temperature cycle, again a seasonal effect.

The soil has a large inertia in responding to changes in the external temperature, the characteristic timescale is longer than a day. This manifests itself in the fact that the most significant eigenvector is the cooling/warming, and all others (daily cycle, shift and broadening) are substantially suppressed in amplitude and have a significant phase shift.

3.2 Expansion on the Basis and Long-Term Trends

To complete the model, we factor the contributions of all sensors over all the days. We project all the daily row vectors on the basis vectors. This gives us four coefficients (e_{i1}, \dots, e_{i4}) to describe the daily behavior of the temperature for each sensor i (five, if we add the mean temperature as e_{i0}). Next, we create the data series (E_1, \dots, E_4) by averaging the coefficients (e_{i1}, \dots, e_{i4}) for all sensors for a given day. In other words, the n^{th} entry in E_1 , for example, represents the average of e_1 taken across all the sensors i for the n^{th} day. In order to identify long-term trends, we iteratively run a low-pass filter with a fixed width of one week over $E_1..E_4$ to get the smooth series of S_1, \dots, S_4 . Hereafter, we will use capitals to denote a time series averaged over all the sensors.

The smoothed series exhibit strong correlations. S_3 and S_4 describe the beginning and the length of daytime, whereas S_2 describes the slow warming and cooling trends, associated with the changes of seasons. These smooth trends serve as the background to all the other variations.

3.3 Event detection

We begin by looking at the projections of each day’s mean-subtracted air temperature on the first four eigenvectors. Although the first four eigenvectors for air temperature represent 90.95% of the total variation in the data, most of the information is shown by the coefficients of the first eigenvector. Thus, we are able to analyze an entire day’s data by looking at its E_1 value, thereby achieving a massive compression. We then apply a suitable threshold on the E_1 series to detect events: values below the threshold correspond to behavior that deviates from the expected behavior of the model. We refer to this method as the BASIC method. This method however does not take into account the seasonal drift.

We improve on the BASIC detector by removing the seasonal drift and running a high pass filter on the E_1 data series. We implement the high-pass filter by computing the difference $D_1 = E_1 - S_1$. We then apply the same threshold mechanism on the D_1 series. We refer to this method as the HIGHPASS method. As we show in Section 4 this method significantly increases the number of events detected and reduces the number of false negatives.

The last approach we present uses the inertia exhibited by the soil temperature. Since soil temperature changes much slower compared to the air temperature, we look at the differences between the high-pass filtered series, D_1 for air temperature and the high-pass filtered data series, D_1 for soil temperature and then set a suitable threshold for detecting events. We refer to this approach as the DELTA method.

4. EVALUATION

We use our model to detect events on the deployment for the period between September 2005 and August 2006 and compare the results with the actual known events recorded by a weather station at Baltimore-Washington International (BWI) airport [32]. We assume that rain at BWI implies rain at Johns Hopkins University, Baltimore which is located 25 miles away. In our evaluation, we only consider rain events which are prominent. That is, we consider event days as days having precipitation greater than 3 mm. We considered 225 days starting from September 17, 2005 and July 20, 2006, and found that 48 events fit this criterion.

There are many other types of events which have also occurred during the days of our sampling: faulty sensors, motes running out of power, etc. Particularly interesting was a period of about 45 days from the middle of March 06 to the end of April 06 in which there

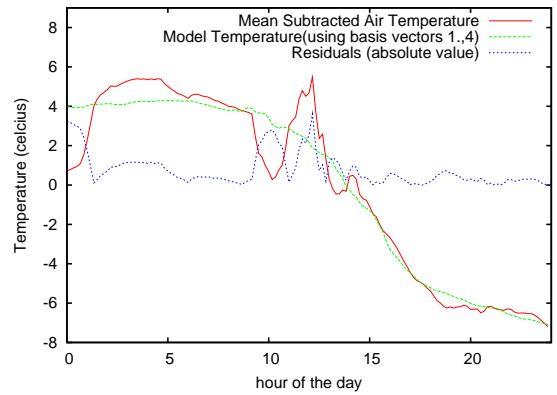


Figure 4: Difference between Air temperature measurement and model projection for the rain event on 2006-01-18.

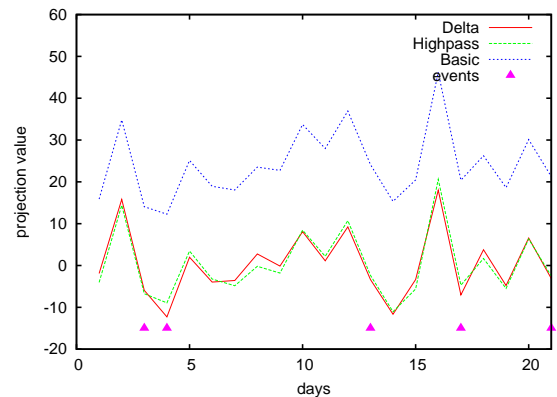


Figure 5: Projection values for different techniques on event and non-event days. The marker at the bottom indicates an event.

were numerous anomalies in the E_1 series. This was the result of sporadic direct sunlight heating up the motes. After April, there was enough foliage cover that the motes (located at ground level) were not exposed to the direct heating of the sun.

We focus on the efficiency of detecting the rain events only using temperature data. There is a good physical basis for this: during rainfall the temperature suddenly drops, but once the rain is over it recovers. This produces a large transient on the shape of the 24 hour cycle for that particular day, resulting in a smaller e_1 coefficient and a larger residual. Figure 4 illustrates this observation. We observed a major event on 2006-01-18. There was heavy rain between 9:00 AM and 11:00 AM. We can clearly see large residuals for this period.

We evaluate the performance of the three methods i.e. BASIC, HIGHPASS and DELTA method. In our evaluation, we use the stan-

Table 1: Performance of different methods for detecting events.

| Method | Precision | Recall | False Negatives |
|----------|-----------|---------|-----------------|
| BASIC | 52.459% | 64% | 18 |
| HIGHPASS | 51.28% | 80% | 10 |
| DELTA | 54.795% | 85.106% | 7 |

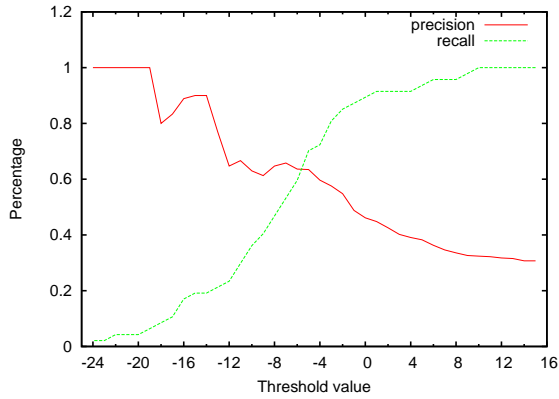


Figure 6: Precision-recall curve for the Delta method.

standard information retrieval metrics of precision and recall. In this case, precision is the fraction of reported events that were actually rain events and recall is the fraction of rain events that our model reported correctly. We also report false negatives, which affect recall but not precision. We attempt to strike a balance between precision and recall. Our goal is to detect as many events as possible with a true positive rate (precision) of at least 50%. Higher precision is difficult to achieve given that our system also detects other (non-rain) events. Recall may be affected by the assumption that rain at BWI implies rain at JHU and vice versa. This is not always the case.

Table 1 summarizes the results for the different methods. Using high-pass filtering and including soil temperature increases recall without affecting precision substantially. The DELTA method significantly outperforms the BASIC and the HIGHPASS methods. We find that because of the inertia shown by soil temperature, the eigen-coefficients E_1 for soil temperature show sharp changes on the day(s) after the event. This simplifies the identification of event days.

Figure 5 illustrates the data series generated by each method for the period between 12/13/2005 and 01/02/2006. The rain events are indicated by a triangular marker at the bottom. We can see that the DELTA method shows sharper negative peaks compared to the other methods on event days and shows lower peaks for non-event days. Notice that the large downward spike shown on day 4 (12/16/2005) corresponds to a large event.

We are able to detect most events days, missing only 7 with the DELTA method. Again, we focus on recall, given that non-rain events occur and pollute our precision statistics. The precision-recall curves for different threshold values (Figure 6) shows that good recall can be achieved at higher than 50% precision. The converse is not true. High recall matches well with our application needs; reporting events when they occur supports network adaptation and identifies interesting regions of data to scientists. In all likelihood, precision and recall would be much improved with more accurate and local weather monitoring – a more accurate “ground truth” – and considering multiple types of events.

5. DISCUSSION AND FUTURE WORK

In this paper we present an application of techniques from statistical signal processing to detect the presence of events (*e.g.*, rain events) that deviate from the regular physical patterns witnessed by a sensor network. We do this by using a variant of the Principal Component Analysis (PCA) technique to generate a compact pro-

file for ‘normal’ measurements. We can then compare actual mote measurements with model predictions and classify the instances in which the two diverge significantly as events of interest. We evaluate the performance of the proposed mechanisms using temperature measurements, collected over a year by a small environmental monitoring network, to detect the onset of rain events. Our preliminary results show that this technique is able to detect most rain events, with small number of false positives, even in the presence of large foreground variations and a substantial seasonal drifts.

This is only the beginning—one can carry this approach much further. While we present event detection in its offline setting, the observation that only a small number of components can accurately describe the collected data suggests that the same mechanism can be implemented on the network’s motes. In an online setting, one can load the motes with the basis vectors and thresholds computed from historical data and project the daily time-series on the basis. In order to build the daily time-series, one can create a series comprising of what is already observed for the current day and combine it with sample bin means for the remaining portion of the day (*i.e.* wrap around the average vector to the yet to be observed series). This is a light-weight, computationally inexpensive, adaptive sampling algorithm that will enable real-life WSN deployments confronted with slowly varying environments as well as sudden, discrete events. Efficient event detection is at the core of any adaptive observing strategy, and we demonstrate how this can be done on today’s WSN platforms.

At this point the method is able to detect global events, *i.e.* events that all the sensors experience. However, one would like to detect localized events. While it is seemingly possible to apply the same PCA technique to detect events experienced by a single mote, it becomes harder to differentiate between an actual event and a malfunctioning sensor. The question is then how much additional information is necessary to separate faults from actual events. The sensors are expected to have variations due to their local environment (located near/far from a stream, sitting on a hillside with a steep gradient, etc.) which will cause small, but consistent, correlated changes. The task is then to find groups of sensors with correlated measurements. We can do so by removing the obvious daily foregrounds, and the long seasonal trends, at which point we expect to see these small correlated differences in the behavior of sensors in the same group. Once such groups are created, we can compare the projected measurements of a mote with the measurements of other group members. If those agree, then a localized event is most likely occurring, otherwise one (or more) of the sensors are faulty.

So far, we completely exclude from the training set, days with partial data in which due to some hardware errors we did not get a reading for every one of the 144 sampling periods. However, it is easy to apply a “gappy” Karhunen-Loève transformation [33], in which the expansion coefficients can still be computed over a partial support. Doing so, will enable the creation of a more representative compressed model of the measurement data and hopefully lead to higher detection accuracy.

Acknowledgments

We would like to thank Ching-Wa Yip (JHU, Department of Physics and Astronomy) for making available to us her PCA C# library and providing us her valuable time in the discussions. The data collected here was done in collaboration with Katalin Szlavecz (JHU, Department of Earth and Planetary Science) and Razvan Musaloiu-E (JHU, Department of Computer Science). The on-line database was built in collaboration with Jim Gray and Stuart Ozer (Microsoft Research). Their help and contributions are gratefully acknowledged.

Last but not least, we would like to thank the anonymous reviewers whose comments significantly improved the quality of the paper.

6. REFERENCES

- [1] R. Musáloiu-E., A. Terzis, K. Szlavecz, A. Szalay, J. Cogan, and J. Gray, "Life Under Your Feet: A Wireless Soil Ecology Sensor Network," in *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets 2006)*, May 2006.
- [2] G. Tolle, J. Polastre, R. Szewczyk, N. Turner, K. Tu, P. Buonadonna, S. Burgess, D. Gay, W. Hong, T. Dawson, and D. Culler, "A Macroscopic in the Redwoods," in *Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2005.
- [3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of 2002 ACM International Workshop on Wireless Sensor Networks and Applications*, Sept. 2002.
- [4] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proceedings of IPSN*, 2005.
- [5] L. Gu and J. Stankovic, "Radio triggered wake-up capability for sensor networks," in *Real-Time Applications Symposium*, 2004.
- [6] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley, 2001.
- [7] IBM, "Db2 intelligent miner," 2007, available at <http://www-306.ibm.com/software/data/iminer/>.
- [8] A. Deshpande and S. Madden, "Mauvedb: supporting model-based user views in database systems," in *Proceedings of ACM SIGMOD*, 2006.
- [9] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of VLDB*, 2004.
- [10] A. Jain, E. Change, and Y. Wang, "Adaptive stream resource management using kalman filters," in *Proceedings of ACM SIGMOD*, 2004.
- [11] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High-Performance Computing Applications*, vol. 16, no. 3, 2002.
- [12] S. Grumbach, P. Rigaux, and L. Segoufin, "Manipulating interpolated data is easier than you thought," in *Proceedings of VLDB*, 2000.
- [13] L. Neugebauer, "Optimization and evaluation of database queries including embedded interpolation procedures," in *Proceedings of SIGMOD*, 1991.
- [14] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed Regression: an Efficient Framework for Modeling Sensor Network Data," in *Proceedings of IPSN 2004*, Apr. 2004.
- [15] H. Gupta, V. Nacda, S. Das, and V. Chowdhary, "Energy-efficient gathering of correlated data in sensor networks," in *Proceedings of MobiHoc*, 2005.
- [16] Y. Kotidis, "Snapshot queries: towards data-centric sensor networks," in *Proceedings of ICDE*, 2005.
- [17] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos, "Compressing historical information in sensor networks," in *Proceedings of SIGMOD*, 2004.
- [18] D. Chu, A. Deshpande, J. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *Proceedings of ICDE*, 2006.
- [19] A. Silberstein, R. Braynard, G. Filpus, G. Puggioni, A. Gelfand, K. Munagala, and J. Yang, "Data-driven processing in sensor networks," in *Proceedings of Conference on Innovative Data Systems Research*, 2007.
- [20] D. Tulone and S. Madden, "PAQ: Time series forecasting for approximate query answering in sensor networks," in *Proceedings of the European Conference on Wireless Sensor Networks*, 2006.
- [21] D. J. Abadi, S. Madden, and W. Lindner, "Reed: Robust, efficient filtering and event detection in sensor networks," in *Proceedings of VLDB*, 2005.
- [22] S. Li, Y. L. adn S. H. Son, J. A. Stankovic, and Y. Wei, "Event detection services using data service middleware in distributed sensor networks," in *Proceedings of IPSN*, 2003.
- [23] A. Herbold1, T. Lamarre, N. Bulusu, and S. Jha, "Resilient event detection in wireless sensor networks," in *Proceedings of Intelligent Sensors, Sensor Networks and Information Processing*, 2004.
- [24] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *CACM*, vol. 47, no. 6, 2004.
- [25] W. Wang, X. Guan, and X. Zhang, "A novel intrusion detection method based on principle component analysis in computer security," in *Proceedings of Advanced in Neural Networks*, 2004.
- [26] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proceedings of ACM SIGCOMM 2005*, Aug. 2005, pp. 217–228. [Online]. Available: <http://www.cs.bu.edu/faculty/crovella/paper-archive/sigc05-mining-anomalies.pdf>
- [27] A. Perera, N. Papamichail, N. Bârsan, U. Weimar, and S. Marco, "On-line event detection by recursive dynamic principal component analysis and gas sensor arrays under drift conditions," in *Proceedings of IEEE Sensors*, 2003.
- [28] A. Iqbal, "Multi-route event detection using PCA," 2006, available at – <http://confluence.slac.stanford.edu/display/IEPM/Multi-route+Event+Detection+Using+PCA>.
- [29] J. F. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control Engineering Practice*, vol. 3, pp. 403–414, 1995.
- [30] M. C. Jones and J. A. Rice, "Displaying the important features of large collection of similar curves," *The American Statistician*, vol. 46, no. 2, 1992.
- [31] G. L. Smith and D. A. Rutan, "The diurnal cycle of outgoing longwave radiation from earth radiation budget experient measurements," *Journal of the Atmospheric Sciences*, vol. 60, no. 13, 2003.
- [32] "Baltimore-Washington International airport, weather station," Available at: <http://weather.marylandweather.com/cgi-bin/findweather/getForecast?query=BWI>.
- [33] "A Robust Classification of Galaxy Spectra: Dealing with author = Connolly, A. J. and Szalay, A. S., Noisy and Incomplete Data," *The Astronomical Journal*, vol. 117, pp. 2052–2062, May 1999.