Evaluation of
Event-Aware
Environmental Data
Compression Schemes for
Wireless Sensor Networks

Mark Moss
School of Computer Science and Software Engineering
The University of Western Australia
mmoss@student.uwa.edu.au

This report is submitted as partial fulfilment
of the requirements for the Honours Programme of the
School of Computer Science and Software Engineering,
The University of Western Australia,
2005
Abstract

Wireless Sensor Networks offer scientists new ways to measure the environment by utilising remote sensors to record important events or variables in an area of interest. Sensor nodes transmit their readings across the network back to a base station where they can then be processed for further analysis. Transmission is a costly operation for sensor nodes due to its relatively high consumption of the sensors’ limited energy source. This raises the question, “Can we reduce the number of transmissions by a sensor without losing any data quality?” Reducing transmission costs will prolong network existence resulting in data collection over a longer period. One solution to this problem is the use of data compression techniques to reduce the number of transmitted readings and thus conserve a sensor’s power. Piecewise linear representation and Haar wavelets are two data compression methods that can be implemented on a sensor node; their effectiveness is evaluated in this paper. Both algorithms were simulated using data collected from field experiments and results find piecewise linear approximation to be superior.

The piecewise linear approximation algorithm evaluated in this dissertation is proven as a solution for controlling transmission costs. However, it operates by compressing data in a non-discriminatory fashion. When monitoring particular environmental phenomena, periods exist where measurement parameters, like the compression threshold, need to be adjusted. This is especially relevant during important events where a sensor should modify its measurement process to record data with maximum accuracy. To achieve this functionality, an adaptive algorithm is developed for soil moisture data sets that extends piecewise representation by incorporating a level of event awareness during measurement. Results show the adaptive measurement technique records critical events at an improved resolution, without significantly reducing the energy savings associated with the standard compression process.

**Keywords:** time-series compression, piecewise linear representation, Haar wavelets, event detection

**CR Categories:** C.2.1, E.4.
Acknowledgements

I would like to thank Dr Rachel Cardell-Oliver for her support and guidance throughout this project. This has been my first experience in the field of wireless sensor networks and her knowledge and advice has been priceless. I must also thank my girlfriend Jackie, for her unlimited supply of patience and motivation which has helped keep me on track during my final year of study. Finally, I would like to thank my family for providing me a place to rest my mind and forget about the realities of university, if only temporarily.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.2 Difficulties in Implementation</td>
<td>25</td>
</tr>
<tr>
<td>4.1.3 Signal Noise</td>
<td>26</td>
</tr>
<tr>
<td>4.1.4 Identifying Events</td>
<td>28</td>
</tr>
<tr>
<td>4.1.5 Event Completion</td>
<td>29</td>
</tr>
<tr>
<td>4.1.6 Events Containing Static Segments</td>
<td>31</td>
</tr>
<tr>
<td>4.2 Implementation and Results</td>
<td>33</td>
</tr>
<tr>
<td>4.2.1 Reacting to Significant Events</td>
<td>34</td>
</tr>
<tr>
<td>4.2.2 Reacting to Event Conclusions</td>
<td>34</td>
</tr>
<tr>
<td>4.2.3 Results</td>
<td>36</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>42</td>
</tr>
<tr>
<td>5.1 Future Work</td>
<td>43</td>
</tr>
<tr>
<td>A Original Honours Proposal</td>
<td>44</td>
</tr>
<tr>
<td>B Fixed Compression Estimation Error</td>
<td>48</td>
</tr>
<tr>
<td>C Event Completion Parameter Analysis</td>
<td>50</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Sensor Statistics.</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Data Collection Summary.</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>Fixed Compression Summary.</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>Soil Moisture Data.</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>Error During Important Events.</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>The Number of Readings Required to Represent a Data Series.</td>
<td>40</td>
</tr>
<tr>
<td>C.1</td>
<td>Required Reduction in Soil Moisture to Avoid Incorrect Event Completion.</td>
<td>50</td>
</tr>
<tr>
<td>C.2</td>
<td>Required History Length $z$ to Avoid Incorrect Event Completion.</td>
<td>50</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Representing a continuous phenomenon with a minimal set of data points. .................................................. 1
3.1 Diagram of the piecewise linear algorithm [19]. ................. 13
3.2 Graph of the Piecewise Linear Representation of the data series. . 14
3.3 Wavelet Coefficient Matrix as a Tree [4]. ........................ 16
3.4 Gradient Error Tree [4]. ................................................ 16
3.5 Reconstructed Wavelet Coefficient Matrix at Threshold = 2 [4]. 17
3.6 Graph of the Haar Wavelet and Piecewise Representation of the data series. ............................................... 17
3.7 Percentage of data series compressed for Em5 data. ............. 20
3.8 Percentage of data series compressed for Mica 2 Mote data. . . 21
3.9 Percentage of data series compressed for Temperature data. . . 21
3.10 Percentage of data series compressed for Humidity data. ........ 22
3.11 Percentage of data series compressed for Wind Speed data. ..... 22
4.1 Data series of Soil Moisture readings with important events highlighted. .......................................................... 25
4.2 Comparison of raw data versus EWMA data. ...................... 27
4.3 Event detection using the EWMA. ................................. 29
4.4 Identifying event conclusion using the EWMA. .................. 31
4.5 Incorrect identification of event completion due to static data. . . 32
4.6 Faulty readings that could cause false-positive event detection. . 33
4.7 Diagram of Event focused data. ...................................... 35
4.8 Diagram of EWMA values used to represent data. ............... 36
4.9 Darling Range trial data processed with event-aware compression. 37
4.10 Pinjar trial data processed with event-aware compression. ...... 38
4.11 Jolimont trial data processed with event-aware compression. ... 39
B.1 Euclidean distances for EM5 Data Logger data. . . . . . . . . . . 48
B.2 Euclidean distances for Mica 2 Mote data. . . . . . . . . . . . . 48
B.3 Euclidean distances for Weather Station Temperature data. . . . 49
B.4 Euclidean distances for Weather Station Humidity data. . . . . . 49
B.5 Euclidean distances for Weather Station Wind Speed data. . . . . 49
CHAPTER 1

Introduction

The role of wireless sensor networks in environmental applications is to monitor a set of environmental variables across a landscape over time and return gathered information for analysis. To capture the trends and properties of environmental variables, sensors record measurements at a specific frequency to generate a discrete representation of the continuous phenomenon. The discrete data points form the sample data, which establishes an initial representation of the continuous data stream. Sample data can be refined by compression techniques to produce a reduced representation set. Figure 1.1 illustrates the two stage process from a continuous signal (1), to a discrete representation (2) and finally the creation of a minimal representation set (3).

Figure 1.1: Representing a continuous phenomenon with a minimal set of data points.

The main constraints on sensors are their computational power and battery life. Data compression techniques can conserve battery but they need to be lightweight [19] with no heavy processing requirements. Another important element is the trade-off between compression and data quality. Higher compression ratios will result in fewer transmissions, but they may conceal minor movements in the measured variable. Therefore, the function of compression algorithms is
to minimise the number of required readings, but still maintain a faithful representation of the underlying data series.

In-network data compression can provide a solution to combat the high energy consumption of transmission operations, but it must comply with the computational restrictions inherent in remote sensor nodes. This dissertation demonstrates that piecewise linear representation and Haar wavelets are two approaches that can successfully operate within a wireless sensor network without exceeding sensor node capabilities. Our motivation stems from the research of Schoellhammer et al. [19], Chen, Li and Mohapatra [4] and Cardell-Oliver [1]. These articles discuss the application of data compression techniques in sensor networks and outline different methods to reduce transmission costs. We will apply and extend their procedures in a simulated environment using data collected in our field experiments, which includes soil moisture, temperature, humidity and wind speed readings.

The compression techniques described in this dissertation generate a reduced representation of the underlying data series - but is this the most appropriate representation of the data? Our results illustrate the potential data compression has for conserving energy and extending network life, but occasions exist where compression should be reduced and sensors should increase measurement rates to deliver more accurate data. Conversely, are there periods where the system only needs to report general trends with low accuracy? These questions motivate us to examine the viability of an adaptive sampling mechanism for soil moisture applications, that modifies its measurement parameters in response to movements in the underlying data series. The hypothesis of this approach assumes incorporating event awareness in data acquisition will result in a more relevant representation of the observed phenomenon.

The desire to develop a reactive measurement process requires the capacity to identify major events. The ability to detect important events allows the system to differentiate between periods requiring increased monitoring and periods where measurements can be reduced to conserve energy. Given the contribution both measurement and compression provide in constructing the final data representation, the metrics underlying these processes can be adjusted to produce the required level of adaptability for the application. Measurement frequency and compression threshold are two parameters that provide the necessary influence to create an adaptive algorithm. Therefore, a system must incorporate measurement and compression strategies to efficiently describe a phenomenon.

A specific soil moisture event detection algorithm was created to evaluate the performance of adaptive measurement in soil moisture monitoring by extending the piecewise linear approximation method developed by Schoellhammer et al. [19].
The remainder of this dissertation will be structured in the following manner; section 2 will summarise existing compression techniques and the potential for adaptive measurement. Section 3 will explain and outline the fixed compression algorithms. Section 4 will describe the adaptive approach and finally, section 5 will conclude the study and propose future experiments.
CHAPTER 2

Compression Techniques

2.1 Fixed Threshold Compression

Two popular methods of data compression include Piecewise Linear Representation of a time series and the use of Haar Wavelets. Piecewise linear representation can be loosely defined as a method which divides up a given time series into a series of straight lines and can adopt a sliding window, top-down or bottom-up approach [9]. This method can also be simplified to develop a Piecewise Constant Approximation whereby the time series is represented by a sequence of constant line segments [10]. Alternatively, Haar Wavelets operate by deconstructing a time series into a sequence of averages and differences [13]. The final sequence of averages and differences is called the wavelet coefficients and it can be used to approximate the original data series [4]. Haar wavelets are superior to other wavelet techniques for wireless sensor networks due to their low computational complexity, which justifies their inclusion in this dissertation.

Existing studies have found both Piecewise linear approximation and Haar wavelet techniques to be effective at compressing sensor data [4,19]. However, the two algorithms’ effectiveness at compressing soil moisture data has yet to be tested. The dynamics of soil moisture is quite unlike that of temperature and humidity - two environmental variables regularly used to test compression schemes. Soil moisture levels rise dramatically only in the presence of precipitation or flooding, otherwise they remain at a low and stable level. This is much different to the cyclical nature of temperature and humidity. Another significant difference between soil moisture and temperature/humidity is the measurement process and equipment used. Many meteorological bodies employ advanced measurement procedures with accurate equipment to obtain high quality meteorological data. In contrast, soil moisture measurement typically involves some fraction of noise which can distort the recorded data and its effect on the compression algorithms needs to be evaluated. Compression techniques will operate most effectively when the network structure can tolerate high latency data measurement. Put simply,
the more data we record, the more we can compress and the greater the ben-
efit acquired from compression. The remainder of this sub-section provides a
description on each approach.

2.1.1 Piecewise Linear Approximation

Piecewise linear approximation algorithms or segmentation algorithms provide
substantial benefits when incorporated in compression techniques. Keogh et
al. [9] provide a good overview of the process of time series segmentation. They
describe three approaches for segmentation algorithms. However, for our study
we will focus on the sliding window algorithms and their variants. Sliding window
is a greedy algorithm, so it can begin computation as soon as it receives sensor
readings - this is an advantage for systems that require timely information. An-
other important advantage of sliding window algorithms is their simplicity, which
makes them well suited to the limited processor capabilities of sensor nodes.

One successful implementation of a sliding window segmentation algorithm is de-
scribed by Schoellhammer et al. [19]. The algorithm is called Lightweight Tempo-
ral Compression and follows the same process for a sliding window algorithm as
described by Keogh et al. [9]. The critical operation in the algorithm is the range
calculation that is performed when a new reading is received. The range is depen-
dent on a threshold value, which defines the algorithm’s sensitivity to movements
in the data series. A larger threshold will command the algorithm to ignore minor
changes in the measured variable and compress more readings into an individual
line segment. Hence, a larger threshold results in greater compression, but it may
also conceal minor trends in the time series. The algorithm delivers a compression
ratio of 20-to-1 and was tested against environmental variables of temperature,
humidity and wind speed [19]. Performance of the algorithm was shown to be
comparable to Lempel-Ziv and complex Wavelet techniques [19]; however, it was
not tested against a simple Haar Wavelet approach. Our experiments conduct
further tests of the Lightweight Temporal Compression algorithm against other
environmental parameters and compare it against the Haar Wavelet approach.

2.1.2 Haar Wavelet Approximation

Wavelet base compression algorithms follow an entirely different framework com-
pared to piecewise linear representations and are commonly used in image com-
pression [7, 24]. Chen, Li and Mohapatra [4] describe a wavelet based, error
aware compression algorithm called RACE (Rate Adaptive with Compression
Error Bound). They employ a Haar wavelet transformation whereby, given a
sequence of readings, neighbouring elements are averaged and stored along with their differences in an array. The resulting array is called the wavelet coefficient matrix, which can be reconstructed to form the original time series. From this point Chen, Li and Mohapatra [4] propose the creation of a Gradient Error Tree, which is constructed from the wavelet coefficient matrix. The Gradient error tree represents the difference between the actual readings and the computed averages under the Haar wavelet transformation. Using the error tree, we can eliminate elements of the coefficient matrix if its corresponding error gradient is less than some threshold value [4]. This is analogous to saying, if the average of two readings has a value close to the readings themselves, use the average to represent the two readings.

The threshold value dictates which readings are better represented by only their average. Each time two readings are represented by one average there is data compression and a sensor node will only have to transmit one point rather than two. The larger the threshold value, the greater the compression, but as we represent more values by their averages, we can lose accuracy. The threshold value used by Chen, Li and Mohapatra [4] correlates to the threshold value used in the piecewise linear transformation of Schoellhammer et al. [19]. Although both techniques are different in their approach, there is a similar compression process underpinning both algorithms.

Our experiments extend the algorithm of Chen, Li and Mohapatra [4] by reconstructing the data series at the sensor, rather than transmitting just the compressed coefficient matrix. This does add extra computation but can further reduce the number readings for transmission without significant loss in approximation accuracy.

2.2 Adaptive Compression

Engineering ways to compress data and reduce transmission costs provides many benefits for wireless sensor networks. The compression techniques described previously in this study illustrate two approaches to the problem [4, 19], but the initial algorithms were developed within a limited framework. When compressing data, the compression algorithms applied a fixed threshold to the data series. Having the ability to apply a variable threshold during data acquisition gives a sensor the flexibility to adapt to changes in its environment. This approach allows the system to relax measurement restrictions during important periods in an attempt to capture more accurate data. Important periods occur when the monitored environment experiences an interesting event, like a rainfall event during a soil moisture experiment. However, increased data accuracy comes at the cost of
increased energy use and reduced network life. This short-coming is acceptable when the data is critical to the application and the increased energy consumption can be sacrificed. Wind speed measurement, soil moisture and object detection networks are examples where this trade-off is appropriate, because they contain important periods whose data is critical to the success of the application. These significant events are often short lived, but need to be recorded at a maximum resolution, which can only be achieved by maximising measurement parameters. Implementing this technique of adaptive measurement results in a more accurate data representation for the observed phenomenon.

2.2.1 Quality of Service

The characteristic of accurate data representation is related to a sensor network’s quality of service (QoS). The data extracted from a network should be sufficiently complete, so a reliable description of the environment can be derived by the end user or application [15]. Hence, one aspect that defines QoS for sensor networks is that the collected data has a high level of relevance, which leads to the best possible data representation. To deliver relevant data, the network is required to monitor events of importance with greater accuracy. Chen and Varshney [3] provide a detailed description of the QoS issues involved in establishing wireless sensor network applications. They describe event driven applications as being delay intolerant and mission critical with the observed events being very important to the success of the application. These factors fuel the need to monitor important events with increased accuracy.

The relationship between QoS and the requirement of relevant data extraction can be associated with the threshold value controlling the compression algorithms. As we reduce threshold values, we compress less information and return results closer to the raw data. Hence, by setting compression thresholds we are determining the QoS to be achieved by the application. This is similar to the approach of Choi and Das [21] in establishing their desired sensing coverage parameter, such that changes in this parameter will redefine the quality of service for the sensor network. Therefore, with the ability to tune the threshold parameter, we can modify QoS for periods of the network lifecycle.

2.2.2 Event Detection

Given the necessity to maximise QoS, methods need to be developed to accurately identify periods of interest. Cardell-Oliver et al. [2] proposed an innovative approach to identifying critical moments by implementing a wireless sensor
network that reacted to its environment, changing its sensing rate, to capture important information. Significant changes in soil moisture occur during rain events. Therefore, by monitoring rainfall, the network can recognise when important soil moisture movements are about to occur. Environmental phenomena are often correlated and exploiting common relationships holds many advantages for wireless sensor networks. The only complication with this approach is the increased network complexity associated with adding extra sensor nodes to monitor correlated variables. Event detection using data analysis does not require any communication with neighbouring nodes or dedicated event-monitoring sensors. It can be performed by a single sensor, using its own stream of data, which reduces any overhead communication costs and conserves energy.

Wireless sensor network applications are also employed to monitor anthropogenic variables like machine condition and military operations. In these situations, correlations may not exist with other variables and the measured events may be more complex in nature. Natkunanathan et al. [14] confronted the problem of detecting and classifying the acoustic signatures of military vehicle with sensor networks. Their solution involved creating a signal search engine whereby monitored signals could be referenced against a predefined library. Once classified, important signal events can be transmitted back to the base station and insignificant signals can be ignored. Applying this technique to environmental monitoring does have complications when constructing a signal classification library: vehicle acoustics maintain a level of consistency, as opposed to variables like soil moisture, which vary dramatically across a spatial landscape given soil type, overhead vegetation coverage, precipitation intensity and seasonal weather variations. This makes creating a predefined signal library difficult.

This dissertation focuses on identifying important events from a sensor’s collected data as it is received. This method increases local processing at each sensor node, but reduces the amount of communication and data exchange required to report events. This is an approach shared by Liang and Wang [11] as they classify events by shifts in the time series. They adapt their algorithms to develop a fuzzy logic system to simplify the classification process. Results revealed a 99.97% probability of detection [11] for a simulated acoustic data set, yet the detection techniques were not tested against any environmental time series.

2.2.3 Tuning Parameters

Identifying events represents the first task of a two part decision-making process. Once an event has been detected, a sensor must be able to react and adapt its measurement procedures accordingly. In regards to the techniques described in
this dissertation, compression rates can be increased during times of minimal activity, but relaxed during times of interest. This would involve decreasing threshold values for compression schemes during periods of high activity. Along with thresholds, a sensor’s reporting rate can be adjusted, or a sensor can be turned off completely to maximise energy conservation [15]. Both modifications will result in more accurate information over the time intervals of most significance and maximum energy conservation when possible. This form of local optimisation is similar to the signal search engine system [14] described in the previous sub-section on event detection where the design of the architecture allows for reconfiguration of the classification library.

Tuning of parameters is not a function that, once performed, only affects the local sensor. It is feasible that a collection centre or base node could broadcast parameter changes when required [25]. This method is similar to the implementation of Cardell-Oliver et al. [2], where a rain gauge node broadcasts “Rate” messages to inform gathering nodes when to adjust their sampling frequency. Yang, Yuan and He [22] developed a centralized optimal management mechanism in their experiments on energy efficiency in wireless sensor networks. The central mechanism was able to adjust modulation levels adaptively in a modulation scaling system to achieve energy savings at the cost of minor latency loss. Neighbouring sensors could also cooperate to adjust thresholds [25]. However the energy expenditure associated with increased communication could well exceed any possible savings or improved data accuracy. This characteristic drives us to develop effective tuning mechanisms that can be implemented locally and with low computational cost.

The storage and energy limitations of wireless sensor networks are well documented, so it is sensible to extend a system to react to depleting energy levels as well as important events. If battery life drops, compression could be increased to reduce the number of required transmissions leading to extended sensor node life. Likewise, increased compression creates fewer records for storage, which is useful when available memory is limited. Reducing the measurement frequency is another option to combat dwindling storage and energy levels, but it increases the likelihood of missing important events; whereas increased compression will still identify large events. The concept of sensors reacting to their own health is discussed by Roadknight et al. [17].

The discussion so far has focused on individual sensors responding to an event by modifying their own properties to capture data with improved accuracy. Batalin et al. [16] approached the problem from a different perspective by implementing a hybrid network comprising static sensors and a high fidelity mobile sensor. When static sensors detect an event, the mobile sensor can navigate to the location
and begin sampling with increased reliability. In their experiments, Batalin et al. [16] suspended the mobile node on a cable to measure light intensity and solar radiation. Engineering this level of mobility in an experiment requiring subsurface monitoring like soil moisture would be highly complex and any associated advantages would be minimal when compared to the development costs.

2.2.4 Redundancy and Correlation

A common characteristic of wireless sensor network applications that over sample their environment, is the high correlation between data from neighbouring sensors [25]. This can be a desirable property since it strengthens the fault tolerance capacity of the network - should one sensor fail, its data stream is covered by a highly correlated neighbour. However, the communication costs involved in transmitting redundant information across the network often outweighs any increase in data delivery reliability. Of greater importance is the possibility that a sensor transmitting redundant data may prevent a sensor with critical data from joining the network if network congestion is an issue [15]. The techniques of data aggregation or data fusion provide solutions to this inefficiency, while maintaining robustness at the cost of latency. Tiny Aggregation (TAG) [12] is a popular aggregation protocol where parent nodes combine data from descendant nodes with their own to remove redundancy. ResTAG [8] extends the TAG protocol by implementing a more resilient system that can detect faulty nodes and returns confidence values for the data. Zhu and Papavassiliou [25] developed a resource adaptive information gathering algorithm (RAIG) that aggregates data like TAG, but considers energy levels and latency when processing and forwarding data to the base node.

Data aggregation is one solution to data redundancy in wireless sensor networks, but the problem can also be solved by restricting redundant sensors from monitoring all together [15]. This way, the redundant data created by these sensors never enters the network - aggregation algorithms are not required and the dormant sensors conserve energy. The correlation between sensors that generates redundant data is dependent on the spatial diversity of the sensors. When a sensor’s coverage area overlaps with another sensor then it is going to provide marginal information to the application [15]. Choi and Das [21] approached this problem by engineering a novel strategy to select the minimum number of sensors, given their sensing range, sufficient to sample a desired coverage. Their results revealed that a higher network density produced higher energy conservation with no increase in computational cost. However, adapting this approach to measure certain environmental phenomena is not a straightforward process. For instance, the dynamics of soil moisture means sections of a desired coverage
may have vastly different properties than other areas, due to varying soil type and hydro-geological variation. Hence, some areas may require more sensors than others and the approach of Choi and Das [21] will not apply.

Data redundancy does not only occur between sensors, but can also occur within the data collected by a single sensor. Compression algorithms can exploit the correlation and redundancy of information among multiple measurements on the same sensor [6, 10]. From this perspective, compression algorithms are analogous to aggregation techniques in the sense that they both eliminate redundant information from a source. However, implementing one does not exclude the other since each technique operates over a different domain.

2.3 Extensions

As compression techniques for wireless sensor networks evolve, new concepts emerge to increase the functionality of existing methods. One developing topic is incorporating prediction into the compression process. Lazaridis and Mehorta [10] approached this topic by considering the contribution a prediction model could provide to a piecewise constant representation algorithm. The idea proposed assumes that a prediction model exists for the measured environmental event. If both the sensor nodes and the base node are aware of the model, then the sensor nodes could transmit the difference between the actual reading and the prediction model estimate back to the base node. On arrival at the base node, the actual reading could be reconstructed by adding the difference to the prediction model. This has advantages over transmitting the actual readings when the differences compress more efficiently than the actual time series. This is similar to the approach of Rossi et al. [18] in their investigation of partial differential equations (PDE) and their ability to model environmental phenomena. Theoretically, transmitting PDE coefficients will cost much less than transmitting raw data, but without a reliable PDE to approximate the time series, compression is not possible.

The feasibility of these approaches would depend on the predictability of the time series. An unpredictable time series would require regular updates of the prediction model, which would lead to increased over-head costs [10]. The dynamics of environmental variables like soil moisture and wind speed make implementing a successful prediction model a difficult process. These environmental events have no consistent trend like temperature or humidity; hence the computational complexity of predicting them exceeds the minimal processing capabilities of sensor nodes. As compression techniques become more complex, there will be an eventual trade-off between the computational cost of compression and the cost
of communicating compressed information [23]. The compression schemes and extensions discussed in this paper attempt to avoid this trade-off by maintaining minimal processing requirements.

There do exist numerous ways to enhance and extend the capabilities of data compression techniques. However, these methods need to be tailored to meet the needs of the particular wireless sensor network. Measurement of different environmental variables will require application specific compression methods; hence, there is no absolute solution to enhancing data compression.
CHAPTER 3

Fixed Threshold Compression

3.1 Methodology

As part of this dissertation, two algorithms were constructed to compress data that was collected from previous wireless sensor network projects. A brief description of each algorithm was outlined in chapter 2; using this as a foundation, we will now delve into the inner workings of both approaches. At the heart of each process lies the threshold figure, which controls the level of compression. Apart from this similarity, the algorithms are structured differently.

3.1.1 Piecewise Linear Approximation

The approach of Piecewise Linear Approximation is to model the series using line segments of any gradient:

\[ y = \alpha \times x + \beta, \text{ where } \alpha, \beta = \text{some constant}. \]

Methods exist that approximate time series based on constant line segments \((\alpha = 0)\) \([10]\). However, for the purposes of this dissertation we will use the more general method. A diagram of the process is shown in Figure 3.1.

![Figure 3.1: Diagram of the piecewise linear algorithm \([19]\).](image)
Beginning with an initial value, the algorithm receives incoming sensor readings and calculates if they are within a certain range. As mentioned in Section 2.1.1, the threshold value determines the range. The dark grey area in Figure 3.1 is the range and the vertical lines at the points represent $+/-$ the threshold value. From Figure 3.1(a), we can see how the range is determined by the extremities of the threshold at the second point. A larger threshold value would generate a wider range. Because the third point is within this range, the algorithm adds it to the recent history of readings (points 1,2,3), and recalculates a new range. This new range is derived using the $+/-$ threshold values at the third point and the original range. It is calculated as the intersection between the range derived at the third point (using the same process as Figure 3.1(a)) and the original range. Figure 3.1(b) demonstrates this process and displays the new smaller range area.

The algorithm continues processing points and recalculating the range, until a new point is evaluated that falls outside the range, see Figure 3.1(c). From here, the algorithm begins a new iteration with the final accepted reading (point 3) used as the new initial value. The final sequence of collected readings, (points 1,2,3), are approximated by a single line segment. Figure 3.1(c) illustrates this with a line connecting the initial point (point 1) and the last accepted reading (point 3). Hence, three or more readings that are accumulated by the algorithm can be represented as a single straight line, which itself is represented by just two points. The three or more readings are therefore compressed to two. Figure 3.2 displays the outcome of applying piecewise linear approximation.

![Image of Figure 3.2](image)

Figure 3.2: Graph of the Piecewise Linear Representation of the data series.

### 3.1.2 Haar Wavelet Approximation

The Haar Wavelet technique is not a form of greedy algorithm like piecewise linear approximation, it requires all, or some set, of the data series before it can
begin computation. More precisely, it requires $2^n$ readings for processing, so a strategy might be employed to perform computation every 64 or 128 readings. For our analysis we take the largest possible subset of the data series, provided its length is $2^n$ for some integer $n$.

The Haar wavelet process begins by taking every pair of neighbouring readings and calculating their average. This value is stored along with the difference between the pair’s second reading and the average. The algorithm continues finding new pairs of averages and differences by applying the same process to the set of newly created averages [4]. The process terminates when only one average remains:

Example:

A series (2,6,5,11) transforms to (4,8,-2,-3):

- Neighbouring Readings: (2,6) and (5,11)
- Average: $\frac{2+6}{2} = 4$, $\frac{5+11}{2} = 8$ and
- Differences: $4 - 6 = -2$, $8 - 11 = -3$.

The second iteration:

(4,8,-2,-3) transforms to (6,-2,-2,-3):

- Neighbouring Readings: (4,8). (-2,-3) are not averages.
- Average: $\frac{4+8}{2} = 6$ and
- Differences: $6 - 8 = -2$. (-2,-3) remains from the first iteration

The final array (6,-2,-2,-3) is called the wavelet coefficient matrix. To demonstrate this more completely we will use the example from Chen, Li and Mohapatra (2004). Given a series of data points,

(3, 4, 3, 2, 6, 8, 9, 7, 2, 3, 1, 2, 10, 8, 7, 9)

we can construct the wavelet coefficient matrix,

(5.25, 0, 2.25, 3.25, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 1, 1, 0.5, 0.5, 1, 1)

using the process applied in example 1. We can illustrate this in a tree structure as shown in Figure 3.3:

From here, we create the Gradient Error Tree [4], which is constructed from the wavelet coefficient matrix. At this point in the process no formal compression has been performed, the time series has only been manipulated to create new data structures. The Gradient error tree represents the difference between the actual readings and the computed averages under the Haar wavelet transformation. Figure 3.4 displays the Gradient Error Tree.

The number in each leaf of the error tree represents the greatest error between the average and the readings it approximates. For instance, according to Figure 3.4, $G(5) = 1.5$. Referring to Figure 3.3, that particular leaf of the branch relates to readings (6, 8, 9, 7). These 4 readings could be approximated by $7.5$ ($\frac{4+8+9+7}{4}$). If
this approximation was used, the greatest error would be 1.5 (7.5 – 6 and 9 – 1.5). This is how the gradient error tree values are calculated. It could be described as representing the level of error suffered if an average value is used to represent a number of readings. Using the error tree, we can eliminate elements of the coefficient matrix if its corresponding error gradient is less than some threshold value [4]. This is analogous to saying, if the average of two readings has a value close to the readings themselves, use the average to represent the two readings. Figure 3.5 shows the coefficient matrix after a threshold of 2 has been applied.

Referring to Figure 3.4, we can isolate all the leaves in the gradient error matrix with a value less than 2, the corresponding leaves of the wavelet coefficient matrix are set to zero, this results in Figure 3.5. Chen, Li and Mohapatra [4] propose representing the empty branches of the tree by a single identifier and then transmitting the compressed time series back to the base station. This would result in the array: (5.25, 0, 2.25, 2.25, t, t, t, t) where t = an empty branch. Our approach reconstructs the compressed time series,

(3, 3, 3, 7.5, 7.5, 7.5, 2, 2, 2, 8.5, 8.5, 8.5, 8.5, 8.5)

and then compresses this array further to (3, 7.5, 2, 8.5). A further modification
is required to assign new time stamps to the compressed readings. In Section 3.1.1, Figure 3.2 was provided to illustrate how the piecewise linear algorithm would appear, Figure 3.6 displays the same series of readings and the comparison between the Haar wavelet and piecewise compression approaches.

Figure 3.6: Graph of the Haar Wavelet and Piecewise Representation of the data series.

3.2 Implementation and Results

The results achieved in this dissertation were gathered from simulation. Algorithms were not physically installed on remote sensors and tested over an extended period of time. Instead, data was used from previous experiments and the algorithms were coded and run in MATLAB.
3.2.1 Implementation

The data used was gathered from three sources: Berkley Mica 2 Motes, Decagon Em5 data logger and a Meteorological Station supplied by Motorola Precision Agriculture Systems. The latter two had their readings averaged, so some smoothing of the results does exist. The data acquired from the motes had no storage averaging. Soil Moisture was obtained via the Motes and the Em5 data logger, while temperature, humidity and wind speed came from the Weather Station. As mentioned earlier, the Haar wavelet approach can only process data sequences of length \(2^n\), so the largest possible subset of the data was used, that fitted this constraint. See Table 3.1 and Table 3.2 for a summary of the data and sensor statistics.

<table>
<thead>
<tr>
<th>Monitored Variable</th>
<th>Network Node</th>
<th>Sensor</th>
<th>Sensor Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soil Moisture (1)</td>
<td>Mica2 Mote/MDA Sensor Board</td>
<td>Echo20 soil moisture probe</td>
<td>±1%</td>
</tr>
<tr>
<td>Soil Moisture (2)</td>
<td>Em5 Data Logger</td>
<td>Echo20 soil moisture probe</td>
<td>±1%</td>
</tr>
<tr>
<td>Temperature</td>
<td>Motorola Weather Station</td>
<td>Vaisala HMP45A</td>
<td>±0.1° C</td>
</tr>
<tr>
<td>Humidity</td>
<td>Motorola Weather Station</td>
<td>Vaisala HMP45A</td>
<td>±1.0% RH</td>
</tr>
<tr>
<td>Wind Speed</td>
<td>Motorola Weather Station</td>
<td>RM Young Wind Sentry</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

Table 3.1: Sensor Statistics.

<table>
<thead>
<tr>
<th>Monitored Variable</th>
<th>Range</th>
<th>Units</th>
<th>Averaged</th>
<th>Readings</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soil Moisture (1)</td>
<td>375 - 1000 mV</td>
<td>milliVolts</td>
<td>No</td>
<td>4096</td>
<td>5 months</td>
</tr>
<tr>
<td>Soil Moisture (2)</td>
<td>375 - 1000 mV</td>
<td>milliVolts</td>
<td>Yes</td>
<td>2048</td>
<td>4.5 months</td>
</tr>
<tr>
<td>Temperature</td>
<td>-40 - 60°C</td>
<td>degrees Celsius</td>
<td>Yes</td>
<td>16384</td>
<td>2.5 years</td>
</tr>
<tr>
<td>Humidity</td>
<td>0.8 - 100% RH</td>
<td>Relative Humidity</td>
<td>Yes</td>
<td>16384</td>
<td>2.5 years</td>
</tr>
<tr>
<td>Wind Speed</td>
<td>0 - 60 m/s</td>
<td>metres/second</td>
<td>Yes</td>
<td>16384</td>
<td>2.5 years</td>
</tr>
</tbody>
</table>

Table 3.2: Data Collection Summary.

To test the application of compression in wireless sensor networks, the collected data was run through both the piecewise linear and Haar wavelet algorithms. The algorithms were tested on the data at multiple threshold levels to gauge the change in compression ratio as threshold values were increased. This was done for soil moisture, temperature, humidity and wind speed data. Both the Euclidean distance and \(L_\infty\) measures were also recorded at each threshold level. With compression ratios and error ratings, we can compare which algorithm achieved better compression and which had less approximation error.
3.2.2 Results

The results from our experiments were conclusive. Both methods illustrated the benefits of applying compression techniques to sensor data, but in all situations the piecewise linear approach was superior. Before interpreting the results a quick refreshment on some key terms is required. The threshold determines the level of compression performed by the algorithm; hence as our threshold increases, we expect to see greater compression. The sensor error represents the level of uncertainty associated with each reading. If our threshold is set below the sensor error, we are effectively compressing data without violating the operating margin of error specified by the component manufacturers [19]. When thresholds exceed the sensor error we expect some loss in signal accuracy.

The first experiment was analysis of the Em5 Data Logger data. As previously mentioned, the readings were averaged, generating a less noisy time series. The results are displayed in Figure 3.7. The series comprised 2048 soil moisture measurements, taken every 30 minutes. An initial threshold of zero yields savings of 49% and 58% for wavelet and piecewise linear algorithms respectively. This result is extremely promising given that a zero threshold corresponds to zero error. This means we have achieved lossless compression of up to 58%. The compression ratio stabilises around 10 mV, but at every point in the series the piecewise linear algorithm is superior to the Haar wavelet approach. The piecewise linear algorithm also operates with similar error at the optimal threshold of 10 mV. It has a Euclidean distance of 20.32 mV and $L_\infty = 9.97$ mV. At the same threshold the Haar wavelet algorithm has a Euclidean distance of 20.20 mV and $L_\infty = 94.98$ mV. A complete description of the Euclidean Distance under both algorithms can be found at Appendix B, Figure B.1.

The sensor error associated with the soil moisture readings is 30mV, at that threshold the piecewise linear algorithm achieves 98% compression or 55-to-1. The Haar wavelet algorithm achieves 97% compression at a 30mV threshold. Both are significant results and given the compressed data stabilises well before the manufacturer’s error level we can conclude that both compression techniques effectively process the Em5 data.

Given the first experiment was on a data series comprising averaged readings, it was an obvious progression to test the algorithms on more noisy data. The Mica 2 Mote data provides a well suited set of readings to apply the second test - the results are shown in Figure 3.8. Due to equipment settings, the Mica 2 Mote data exhibited greater data variation due to the presence of signal noise during measurement. The mote data set contained 4096 soil moisture readings. When comparing it with the Em5 data, the first important difference is the compression ratio at a threshold of zero. The Em5 data yields a 58% compression saving
whereas the mote data generates only a 5% saving. This can be attributed to the greater variation, caused by signal noise, in the mote data. The 58% saving from the Em5 data at a zero threshold is in fact the lowest for all our experiments, which highlights the difference between averaged data and raw readings. A zero threshold compresses consecutive readings with the same value. The noise associated with the mote data reduces the number of consecutive readings of equal value and hence diminishes zero threshold compression. Similarly to the Em5 data, the piecewise approximation method outperforms the Haar wavelet algorithm throughout the test. The estimation error results also support the piecewise linear approximation approach; Euclidean Distances for the Mica 2 Mote data is presented in Appendix B, Figure B.2.

The first of the tested weather station data series is Air Temperature, Figure 3.9 displays the results. All weather station data sets contained 16384 readings. As expected, both algorithms compress the series effectively, but there is no compression at the zero threshold. This indicates the data contains no consecutive readings of the same value. At greater threshold values the two techniques do converge and the wavelet approach does produce marginally better compression past thresholds of 16°C. However, the temperate sensor has an error of ±0.1°C, consequently 16°C is an unrealistic option for the threshold. The Haar wavelet algorithm has a Euclidean distance of 0.04°C and $L_\infty = 4.87°C$ at a 0.1°C threshold. The piecewise approach has a Euclidean distance of only 0.001°C and $L_\infty = 0.1°C$ at the same threshold. The piecewise approximation is both more effective and accurate at representing the temperature data set. Appendix B, Figure B.3 illustrates the differences in Euclidean distance between the two compression
Figure 3.8: Percentage of data series compressed for Mica 2 Mote data.

Figure 3.9: Percentage of data series compressed for Temperature data.

The results for the test on the Humidity weather station data, in Figure 3.10, are similar to those for the temperature data series. The piecewise linear method dominates the Haar wavelet method for all threshold values, more so for smaller values. When comparing humidity to temperature, it is apparent that the humidity data contains greater variation. This can be seen by the gentler gradient and the reduced compression ratios at similar thresholds. However, the humidity data set also generates no compression at a zero threshold, which is similar to the temperate data.

The sensor error for humidity is rendered in Figure 3.10 at 1.0% Relative Humidity. At this threshold the piecewise linear representation achieves 56% com-
Compression of Weather Station Humidity Data

Figure 3.10: Percentage of data series compressed for Humidity data.

Compress, which surpasses that Haar wavelet equivalent of 40%. Comparing estimation error reinforces the superiority of piecewise linear approximation for humidity data. Given a 1.0% RH threshold, the piecewise algorithm has a Euclidean distance of 0.15 and $L_\infty$ equal to 1. The respective Haar wavelet values are 1.192 for Euclidean distance and 23.4 for $L_\infty$. A complete analysis of the Euclidean Distance for humidity data can be found at Appendix B, Figure B.4.

Figure 3.11: Percentage of data series compressed for Wind Speed data.

Figure 3.11 displays the compression ratios achieved on the Wind Speed Data. Wind speed naturally contains a large amount of variation compared to the other weather station data sets. This causes the compression ratio to be comparatively low for similar thresholds. Given a threshold of 3 m/s, the piecewise linear algorithm achieves compression values of 94% for temperature, 76% for humidity, but only 64% for wind data. Despite the increased variation in the data, the piecewise linear algorithm still performs better than the Haar wavelet algorithm. No sensor error was supplied with the wind speed sensor documentation, there-
before it is not represented in Figure 3.11. However, Euclidean distance for both algorithms is presented in Appendix B, Figure B.5.

<table>
<thead>
<tr>
<th>Monitored Variable</th>
<th>Error Threshold</th>
<th>Piecewise Compression</th>
<th>Piecewise Euclidean</th>
<th>Haar Wavelet Compression</th>
<th>Haar Wavelet Euclidean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soil Moisture (1):Mote</td>
<td>10mV*</td>
<td>93%</td>
<td>11.07</td>
<td>90%</td>
<td>137.10</td>
</tr>
<tr>
<td>Soil Moisture (2):Em5</td>
<td>10mV*</td>
<td>98%</td>
<td>20.32</td>
<td>95%</td>
<td>20.21</td>
</tr>
<tr>
<td>Temperature</td>
<td>0.1°C</td>
<td>44%</td>
<td>0.0012</td>
<td>26%</td>
<td>0.041</td>
</tr>
<tr>
<td>Humidity</td>
<td>1.0% RH</td>
<td>56%</td>
<td>0.15</td>
<td>39%</td>
<td>1.20</td>
</tr>
<tr>
<td>Wind Speed</td>
<td>3 m/s**</td>
<td>64%</td>
<td>1.46</td>
<td>57%</td>
<td>2.94</td>
</tr>
</tbody>
</table>

Table 3.3: Fixed Compression Summary.

*10mV was used instead of 30mV because it represents an optimum threshold.

**3m/s was chosen as a demonstration threshold, given no sensor error was available for the wind speed sensor.

Table 3.3 summarises the results illustrated in the graphs above. The piecewise representation algorithm achieves superior compression with less estimation error for all data sets. The Haar wavelet technique does process the Em5 data with slightly less error, but this does not compensate for the inferior compression ratio.

The experiments have demonstrated the benefits compression algorithms can bring to wireless sensor networks. In particular, substantial lossless compression can be achieved using a zero threshold - in some cases up to 58%. Results also found variation in the data strongly influenced the rate of compression. Data series created from averaged or smoothed readings received improved compression ratios as opposed to the unadjusted data series. Most importantly, our tests show that piecewise linear approximation should be the preferred method of compression.
CHAPTER 4

Adaptive Compression

4.1 Methodology

The ability to detect important events allows the system to differentiate between periods requiring increased monitoring and periods where measurements can be reduced to conserve energy. Reducing the measurement rate and increasing compression can create a data representation that approximates the real data with fewer data points. This approach conserves energy by reducing the required number of readings for transmission; however, it can also create a less accurate representation for two reasons. By reducing the measurement rate the sensor is at greater risk of missing important changes in its environment. For example, if a sensor records measurements every 10 minutes, an event could occur within that interval and not be recorded. Secondly, by increasing compression we reduce the sensitivity to small changes in the data, which means minor movements do not get recorded. These pitfalls can reduce data accuracy when the data series is varying due to the presence of an event. However, for periods of inactivity, the data series varies less and reducing measurement variables will not generate less accurate data. In contrast, during important events the system should increase its reporting accuracy to deliver higher quality data when it is most needed.

When monitoring soil moisture an important event would be the presence of rain, which almost always results in an increase in soil moisture levels, as shown in Figure 4.1. With the ability to efficiently detect events, the system can tune measurement parameters to capture accurate data, resulting in an improved data representation.

4.1.1 Background

Creating event detection algorithms requires an understanding of the behavior of the variable under observation. In this dissertation, soil moisture has been selected to test the effectiveness of detection techniques. Figure 4.1 demonstrates
4.1.2 Difficulties in Implementation

Event detection can be achieved by simple data analysis to identify events of interest, performed locally at each sensor. The benefit of this approach is minimal communication requirements, which translates to reduced transmission costs. However, relying on data analysis makes the network susceptible to any flaws in the data - in particular, signal noise. Signal noise can mask the first signs of a significant event and delay the system’s ability to react to important changes in its environment. The reason for this is the difficulty in differentiating between noise and the beginning of an event. How does the system know a small positive movement is the initial stages of an event or signal noise? Eventually the event will become apparent, but identifying it halfway through its duration delivers little information to the user and spoils the ability to generate the desired data representation for the time series. It is essential to detect important events as early as possible and record them with maximum accuracy.

Another issue associated with event detection is the ability to capture all forms of events. Figure 4.1 illustrates two possible event formations, the second being much more sudden and dramatic. Some events are less pronounced than others,
which make capturing these subtle movements a difficult process. An effective algorithm should be able to detect all possible formations of event and minimise any false-positive occurrences. Acknowledging when an event has completed is equally important to the system, but has its own associated complications. If an event contains a portion of static data, should the system declare the event as completed, or wait for any further significant movements? Some solutions to signal noise and event classification are presented in the next sections.

4.1.3 Signal Noise

As mentioned earlier, a steep increase in soil moisture will indicate the occurrence of an important event. However, to detect these events, the existence of any signal noise needs to be minimised. Signal noise produces undesired interference with intended operations, which creates unnatural fluctuations in the underlying data series. It can be overcome by implementing an exponentially weighted moving average (EWMA) which will smooth noisy data, but still provide the capability to identify unusual events. The EWMA is defined as [20]:

\[ X_k = \alpha \cdot X_{k-1} + (1 - \alpha) \cdot X_k, \]

Where \( \overline{X}_k \) is the new EWMA value, \( \overline{X}_{k-1} \) is the previous EWMA value, \( X_k \) is the current raw value and \( \alpha \) is the tuning parameter. An exponentially weighted moving average is used instead of other averaging techniques because of its simple structure and minimal storage and memory requirements. It is also reactive to small shifts in the data series [20] that makes it well adapted to detecting important events. The level of sensitivity is governed by the tuning parameter \( \alpha \). A lower \( \alpha \) value will result in a more dynamic EWMA, whereas a value closer to one will produce a more smoothed result. For our purposes of event detection, we need a \( \alpha \) value that will dampen signal noise, but not smooth the beginning of a significant event. Woo [20] evaluated the reactivity of the EWMA to varying \( \alpha \) values and classified a value of 0.9125 as agile and 0.99 as stable. Based on this knowledge, a \( \alpha \) of 0.9 is selected as it provides the agility to highlight critical soil moisture events without excessively smoothing the data. With a \( \alpha \) of 0.9, the previous EWMA value \( \overline{X}_{k-1} \), accounts for 90 percent of the new EWMA value \( \overline{X}_k \). As a result, 10 percent \( 1 - 0.9 \) of the new EWMA is derived from the current reading \( X_k \), generating the desired responsiveness to event driven changes in the data. Figure 4.2 illustrates the affect of applying EWMA to noisy mote data.

The example displayed in Figure 4.2 reveals some valuable properties derived
from employing an EWMA. Firstly, its noise suppressing qualities generate a stabilised data representation that still reacts to substantial data movements. The local minima and maxima in the raw data from Figure 4.2, correspond to a noticeable decrease and increase respectively in the averaged results. However, given the average value includes 90% of the previous EWMA value (0.9 \* X_{k-1}) and only 10% of the current reading ((1-0.9)\*X_k), the moving average readings will not always follow the trend of the raw data. Referring to Figure 4.2, the local minimum of 644.5mV is followed by the data point of 647mV, an increase of 2.5mV (647mV - 644.5mV). Despite this increase, the EWMA decreases over the same interval. A similar response occurs between readings 982 and 983 where the raw data decreases, yet the moving average increases slightly. This behaviour is caused by the 90%, 10% split in calculating the EWMA and results in the moving average becoming and effective representation for the momentum in the data series. This characteristic makes it ideal for detecting soil moisture events within a noisy data set, because minor movements do not affect the momentum. A small decrease during the initial stages of an event may compromise event identification using raw data; however, the EWMA is more resilient to opposing movements, which makes it a more reliable and robust technique.
4.1.4 Identifying Events

With the EWMA established, we can use its results to decide when an important event is occurring. A simple implementation would be to measure the change in the EWMA. If the EWMA increases by some value $y$, over some interval of length $x$, then we can conclude that an important event is transpiring. We can make this assumption with confidence knowing that the influence of signal noise has been significantly reduced by utilising a moving average. Determining appropriate values for $x$ and $y$ is largely dependent on the application, however some guidelines do apply. The larger the measurement interval $x$, the more data required before an EWMA can be calculated. A greater $x$ value also requires more storage space - a scarce resource for wireless networks. Both these factors influence the decision to select the smallest $x$ value possible. However, an interval too short may fail to detect a subtle increase in soil moisture caused by light rain. As shown in Figure 4.1, the first rain event is characterised by a series of minor increments in moisture as opposed to the second, more drastic, event. Given a short measurement interval, no significant moisture increase would be registered at the beginning of the first event due to the slow increase in soil moisture.

Modifying the sensitivity, $y$, to increases in soil moisture will also influence the ability to accurately detect major events in the data. The larger the value of $y$, the greater the required increase in soil moisture before an event is positively identified. If a value of $y$ is too large, then subtle events may pass unidentified. However, a value of $y$ that is too small may incorrectly categorise insignificant points as being important. Natural variation will exist throughout the data series and a slight positive trend will trigger a small $y$ value, even if the positive trend does not represent an event. These properties mean an appropriate $y$ value must be small enough to detect all the important events, but large enough to pass over insignificant positive movements. Therefore, when determining both $x$ and $y$ together, the problem boils down to finding the lowest possible values for both variables, such that all the important events are detected without error.

Our experiments found $x = 3$ and $y = \text{sensor error } \times 50\%$ yielded accurate results for noisy soil moisture data sets such as that shown in Figure 4.1. Smoother data sets constructed from averaged readings perform equally well and the additional smoothing from the EWMA does not cause any adverse effects. By setting $x = 3$, the system will measure the previous three EWMA moisture readings to see if any noticeable increase has occurred, which would signify an event. The EWMA might smooth measurements, but over three time steps even a subtle event will trigger the $y = \text{sensor error } \times 50\%$ limit. It is also important to note that the $y = \text{sensor error } \times 50\%$ limit is substantial enough to filter out any positive noise and is reinforced by the interval length of $x = 3$. The probability of signal
noise generating three consecutive positive readings exceeding $y = sensor\ error \ * 50\%$ is virtually impossible from a functioning soil moisture sensor. Without the introduction of the EWMA and its noise dampening properties, values for $x$ and $y$ would need to be considerably larger to achieve similar results from raw data. An example of the algorithm’s detection ability is presented in Figure 4.3.

Figure 4.3: Event detection using the EWMA.

The dashed vertical line signifies that the beginning of an event has been detected, which in this series occurred at measurement 495. At this point, the EWMA had increased at a rate greater than the $y = sensor\ error \ * 50\%$ threshold over the previous three readings. Figure 4.3 also demonstrates the EWMA’s inability to imitate event driven data. The EWMA does not increase at the same rate as the raw data, due to it deriving 90% of its value from the previous EWMA value - see Equation 4.1. Consequently, EWMA values can not be used to represent the steep incline which characterises soil moisture events; instead, raw data is employed to attain the most accurate data representation. One could argue that the system should detect the event earlier, which could be achieved by setting a lower threshold. However, this increases the chance of incorrect detection or false-positives, especially in noisy data sets like the one in Figure 4.3.

4.1.5 Event Completion

Having the ability to detect critical events allows the sensor to tune itself to capture data at a higher resolution, by recording more regularly and lowering
compression thresholds. However, it is equally important to be able to detect when an event has passed. Without this skill, sensors will capture the less interesting data that follows an event with higher resolution, leading to excess energy use and reduced network life. Therefore, in the interests of prolonging network existence, any effective event detection algorithm must be matched with an equally effective event completion detection mechanism. This can also be achieved by utilising the exponentially weighted moving average.

Gradients

An important feature of EWMA and its noise reduction property is that it provides the ability to calculate efficiently a data series’ gradient. Signal noise can cloud the true gradient of a segment of the data series, especially in the presence of large outliers. By removing its influence, we can confidently measure gradients and use the results as part of our event detection decision-making processes. The gradient of a data series can be used to detect when an important event has subsided. Recalling the soil moisture example mentioned earlier, we know that the presence of rain initiates a sudden increase in soil moisture levels. When the rain passes, the water drains from the top soil and moisture levels return to equilibrium. As moisture levels stabilise, so does the gradient of the data series - this characteristic can be used to identify the conclusion of a significant event. Data stabilisation correlates to a gradient of approximately zero. Hence when the gradient returns to a value close to zero after an event, we can declare that the important event has concluded. A gradient will rarely ever exactly equal zero. Hence in our calculations we use a value of 0.01 to represent the stabilisation threshold and measure this against the absolute value of the gradient.

At each point, we can calculate a gradient value, but we must decide on a measurement interval. Do we calculate a gradient for the entire preceding data series? Or should we only calculate for the previous \( z \) readings? The smaller the data history \( z \), the less storage space required to calculate the gradient. However, we are using the gradient to determine when the data series has stabilised; hence, the shorter the measurement interval, the greater the chance of incorrectly interpreting a temporary lull during an important event as its conclusion. Therefore, as we calculate the gradient over a smaller data history and conserve storage capacity, we become less confident that the gradient can determine the correct event completion point. With this knowledge, our experiments calculated the gradient over a history length \( z \), of ten (refer to Appendix C). This may seem like a large amount of unnecessary storage, but as we are determining the end of an event, it is likely that we have already stored the previous ten readings because they represent part of the important event. Secondly, during an event
the system is likely to increase the monitoring frequency, which means the data history may span a shorter period than usual. This influences our decision to select a larger interval than we would use for detecting events. The results of detecting event conclusion are illustrated in Figure 4.4.

![Identifying Event Completion using EWMA](image)

Figure 4.4: Identifying event conclusion using the EWMA.

The solid vertical line represents the moment the system recognises an event has passed. It is clear the momentum has stabilised at this point by referring to the EWMA and this is confirmed by the gradient being less than 0.01.

### 4.1.6 Events Containing Static Segments

In the previous sub-section an event completion algorithm was devised by correlating a stabilisation in the gradient with the conclusion of an event. This proves to be an effective implementation, but is susceptible to unexpected soil moisture formations. Figure 4.5 illustrates one such soil moisture event that triggers premature event termination.

The event illustrated in Figure 4.5 contains a large section of static data generated by an unexpected flood within the sampling environment. This causes the system to declare the event as completed, represented by the solid vertical line, because the gradient is approximately equal to zero. In reality, the event continues until the moisture levels return to a lower equilibrium of roughly 425mV. To combat this issue, the system needs to incorporate a mechanism to verify that event
completion is only identified after soil moisture has receded by some value \( w \), from its maximum point during the event. The procedure devised in this study is to declare event completion only after soil moisture levels have reduce by 45% (refer to Appendix C). Using Figure 4.5 as an example, if soil moisture levels rise from 375mV to 700 mV (325mV increase) during an event, then event completion will not trigger until the moisture recedes by 146.25mV (45% of 325mV) from the event maximum (700mV). Therefore, moisture levels would have to be below 553.75mV (700mV - 146.25mV) before the system allows event completion.

An alternative approach was initially considered when devising a solution to the problem presented in this subsection. Rather than attempting to alter the event completion point in Figure 4.5, why not declare that another event has begun when the soil moisture levels decrease rapidly? This method would capture the drainage segment of the flood event with the desired accuracy. An event detection process has been described previously that identifies events after measuring a significant increase in soil moisture. This process could be adjusted to identify events after detecting a significant increase or decrease in soil moisture. The short-coming of this process is related to the quality of the data used to perform event detection. Any faults causing the sensor to record drop-out measurements would trigger the redefined event identification algorithm. Figure 4.6 illustrates faulty data readings that could cause the sensor to incorrectly detect an event and create a false-positive result. The original approach of redistributing the event completion point does not suffer from this problem.

Figure 4.5: Incorrect identification of event completion due to static data.
4.2 Implementation and Results

The results of implementing an event detection scheme that integrates with an established compression process are described in this section. The algorithm extends the piecewise linear approximation technique due to its flexible sliding-window approach and superior performance when compared to the Haar wavelet technique. The change to the original piecewise algorithm is that it now operates with a variable, rather than fixed, sensor error. The sensor error will be directly proportional to the magnitude of the current readings; hence higher value readings should be associated with increased error potential. Sensor error of the Echo20 soil moisture probe is 3% or 1% if it has been calibrated [5]. This makes the Variable Sensor Error = (EWMA * 0.03) or (EWMA * 0.01) if calibrated. The EWMA is chosen to represent the current readings as it filters out noise and is already calculated for other purposes.

Engineering a reactive network requires an event aware algorithm. Event detection simply partitions the data series into sections of importance, it is then up to the system to decide the appropriate course of action in response to these events. For the purposes of this dissertation, event identification classifies elements of the data series as being either significant or insignificant. How these classifications are processed is described in the next subsections.

Figure 4.6: Faulty readings that could cause false-positive event detection.
4.2.1 Reacting to Significant Events

When an important event has been identified, the system should endeavour to capture its associated data with maximum accuracy. At a local level, a sensor can achieve this by reducing compression and increasing measurement frequency. Executing these modifications will produce more frequent data points with less approximation until the system returns to its maximum energy conserving state. Determining appropriate values for measurement frequency and compression threshold during an event is dependent on the phenomenon under observation. If the network is monitoring mission critical events similar to those in military operations, then measurement frequency may need to be increased to a maximum and a compression threshold of zero would be applied to ensure lossless compression. Less drastic modifications can be made for environmental applications where simply doubling the measurement frequency would be an acceptable increase. Our simulations reset the compression threshold to 20% of its original value during an event, which does introduce a small amount of information loss during compression but does not degrade the relevance of the final data series. Given the threshold is reset to a value below the sensor error, we are effectively compressing data without violating the operating margin of error specified by the component manufacturers [19].

It is important to clarify that under this system the final data series used to represent an event consists exclusively of raw data values. The exponentially weighted moving average values are only used to detect important events, they are not used to represent data values. Referring to Figure 4.7, the clustering of circles around the peaks in the data illustrates the increased data acquisition during important events.

4.2.2 Reacting to Event Conclusions

During uneventful periods, the sensor is required to gather data at the minimum measurement frequency and with a standard compression threshold equal to the sensor error. Therefore, when an event concludes, the system must reset these variables back to their standard values. The maximum energy saving state will deliver extensive power and storage savings due to the low rate of measurement and high probability of compression. The data series generated for these periods will be a general representation of the raw measurements, which is perfectly acceptable given it represents uneventful data.

Despite the reduced significance of this data, it can be processed further to produce a more useful data representation, at no extra computation cost. During the
compression of uneventful data segments, the sensor can process EWMA values rather than raw data points. This measurement method will generate a more general representation of the trend in the data, plus remove the effect of any unusually noisy data points. The piecewise linear approximation technique has a tendency to store the outlying points caused by signal noise because they exceed the compression threshold. Recording EWMA in the place of the outliers creates a smoother data representation.

The EWMA value is already calculated to check for any important events, so utilising it in compression comes at no additional cost. The result of this approach is illustrated in Figure 4.8. Implementing this functionality into the system relies on piecewise linear approximation being the active compression technique because it is a sliding window class of algorithm. Sliding window algorithms approximate a series of points by a line to achieve compression. EWMA values can be used to represent the end points of the line without seriously modifying the piecewise linear approximation algorithm. Developing this functionality with Haar wavelets would be a complex process and our earlier results demonstrating the superiority of piecewise approximation makes pursuing a Haar wavelet solution redundant.
4.2.3 Results

Three soil moisture data sets were selected from previous field tests to evaluate the effectiveness of the event aware compression algorithm. Each set exhibits different characteristics due to environmental dynamics and hardware specifications. Table 4.1 summarises the three soil moisture time series.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Network Node</th>
<th>Device Averaged</th>
<th>Storage Averaged</th>
<th>Sensor Error</th>
<th>Data Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Darling Range Data</td>
<td>Mica 2 Mote</td>
<td>Yes : 16</td>
<td>No</td>
<td>±1%</td>
<td>1810</td>
</tr>
<tr>
<td>Pinjar Data</td>
<td>Em5 Data Logger</td>
<td>Yes : 16</td>
<td>Yes : 30</td>
<td>±1%</td>
<td>2208</td>
</tr>
<tr>
<td>Jolimont Data</td>
<td>CSIRO Fleck</td>
<td>Yes : 16</td>
<td>No</td>
<td>±1%</td>
<td>614</td>
</tr>
</tbody>
</table>

Table 4.1: Soil Moisture Data.

The first of our event aware compression experiments evaluates the algorithm against the Darling Range moisture data, recorded on the Mica 2 Mote node. The appearance of this particular data series has been presented in Figure 4.9. The data contains a considerable amount of noise and is characterised by a single prominent event. To accurately process this data series the algorithm must be able to recognise the event through the noise, without incorrectly identifying insignificant segments of the series. The probability of false-positives is much greater when the data set is noisy.
Figure 4.9: Darling Range trial data processed with event-aware compression.

Referring to Figure 4.9, the beginning of an event is signified by a dashed vertical line and the event conclusion is classified by the solid vertical line. The major event is successfully captured by the algorithm. An important characteristic of this data set is the event contains more than one local maximum. The adaptive algorithm captures all peaks within a single event boundary. This indicates that the algorithm determines moisture levels are not reaching equilibrium in between peaks. This is an encouraging result as it demonstrates the algorithm is not prematurely declaring an event as finished. The reduced compression threshold during significant events also insures the local maxima for each peak is recorded. Figure 4.9 also verifies that the algorithm is not incorrectly detecting events. The event detection algorithm is not triggered during any insignificant data periods, which achieves our goal of eliminating false-positives.

The second soil moisture data set was recorded at Pinjar Banksia Reserve, using Em5 Data loggers, which employed an averaging scheme to smooth the data. Readings were taken every minute and averaged over 30 minutes to generate a data point. Unlike the two other field experiments, the Pinjar network was exposed to a flooding incident, which can be seen clearly between readings 940 and 1060 in Figure 4.10. The data also contains a small event before the flood, which is successfully identified by the algorithm, but the importance of testing against this data set is examining the algorithm’s reaction to the unexpected flood episode.

At the height of the flood, the soil is saturated and the data series flattens out.
This plateau effect would usually trigger the adaptive algorithm to declare the event as completed, due to the EWMA gradient being approximately equal to zero. However, because the system has not registered a satisfactory drop in moisture level, the sensor continues measuring with increased accuracy. This ensures the sensor captures the drainage dynamics and identifies event termination at the appropriate location.

The final data series under investigation is the CSIRO Fleck data collected from Jolimont. The data contains the smoothest time series, with no presence of signal noise and sensor error. The results from applying the event-aware algorithm to the smooth data is illustrated in Figure 4.11.

The Jolimont data only spans a short time period as opposed to the previous two data sets, but still contains a significant event to test the event-aware algorithm. An interesting characteristic of this time series is the mild reaction to the identified event. The compression threshold is significantly reduced when an event is detected, yet the algorithm only stores a relatively small number of points when compared to the previous data sets. This can be attributed to the level of smoothing present in the Jolimont data. The application of comprehensive data smoothing significantly increases the efficiency of the piecewise-linear approximation compression technique. Information is not being lost during the event period; it is just being approximated efficiently by the compression algorithm. The segments outside the event boundaries are compressed at the full compression ratio and consequently approximated by fewer line segments. The
Figure 4.11: Jolimont trial data processed with event-aware compression.

section of the time series after the event completion boundary does exhibit an event-like formation, with a small peak near reading 500. It fails to trigger the event detection mechanism due to its mild increase in soil moisture. This indicates the threshold could be reduced for smooth data sets to produce a more accurate data representation.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Adaptive Euclidean Distance</th>
<th>Adaptive $L_\infty$</th>
<th>Standard Euclidean Distance</th>
<th>Standard $L_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Darling Range Data</td>
<td>0.013</td>
<td>1.13</td>
<td>0.612</td>
<td>6.58</td>
</tr>
<tr>
<td>Pinjar Data</td>
<td>0.091</td>
<td>1.41</td>
<td>2.915</td>
<td>7.0</td>
</tr>
<tr>
<td>Jolimont Data</td>
<td>0.090</td>
<td>1.01</td>
<td>3.756</td>
<td>5.07</td>
</tr>
</tbody>
</table>

Table 4.2: Error During Important Events.

Table 4.2 presents the error associated with approximating significant events using an adaptive and fixed measurement scheme. A quick recollection of Euclidean distance and $L_\infty$ is required before analysing the data.

\[
EuclideanDistance = \frac{\sum_{t=0}^{n}(X_t - X'_t)}{n}
\]  

\[
L_\infty = \max(|X_t - X'_t|)
\]
Euclidean distance represents the average error between the raw and estimated values, whereas $L_\infty$ measures the largest error between any raw and estimated value for the time series. According to Table 4.2, for all three soil moisture data sets, the adaptive algorithm represents important events with less estimation error than the standard compression algorithm. The Pinjar and Darling Range data contain events where moisture levels exceed 700mV. During these events, the sensor error increases because it is proportional to current soil moisture values. An increase in sensor error will generate a larger compression threshold, which corresponds to a greater Euclidean distance and $L_\infty$ value. Therefore, under a standard approach with no measurement adaptability, extreme events enlarge estimation error. This property accounts for the Pinjar and Darling Range $L_\infty$ ratings being higher than the Jolimont equivalent, because the Jolimont data experiences a soil moisture event of less magnitude.

The benefits of an adaptive measurement scheme are amplified in data sets containing major events with substantial moisture readings. As described previously, the compression threshold of the standard approach increases during these events, leading to increased estimation error. The adaptive implementation reduces its compression ratio during significant events to minimise error. This explains the large difference between adaptive and standard $L_\infty$ measurements for the soil moisture data sets. All data sets experience a much lower Euclidean distance under the adaptive algorithm, except the noisy Darling Range data. The presence of signal noise in the raw data causes the standard algorithm to record the majority of the event’s data points, because they exceed the compression threshold. The averaged data sets have much less data variation and therefore experience more compression throughout the event.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Raw Points</th>
<th>Adaptive Points</th>
<th>Standard Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Darling Range Data</td>
<td>1810</td>
<td>124</td>
<td>473</td>
</tr>
<tr>
<td>Pinjar Data</td>
<td>2208</td>
<td>151</td>
<td>64</td>
</tr>
<tr>
<td>Jolimont Data</td>
<td>614</td>
<td>25</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 4.3: The Number of Readings Required to Represent a Data Series.

The adaptive algorithm delivers a reduction in estimation error for significant events at the cost of increased energy consumption. Table 4.3 contains the number of data points used in the adaptive and standard measurement processes to illustrate the entire data series, not just the important events. The adaptive approach stores more readings to generate a representation than the standard process for the Pinjar and Jolimont data. However, the adaptive process out-
performs the standard compression for the Darling Range data. This can be attributed to the extended periods of uneventful data which are compressed at the maximum energy conserving rate under the adaptive approach.

When comparing the adaptive algorithm to the raw data representation, the adaptive approach performs well. The Darling Range data set experiences a 93% compression ratio, which is considerable given the level of signal noise existing in the data. The Pinjar and Jolimont data sets experience a 93% and 96% compression ratio respectively. The Jolimont compression exceeds the other two due to the smooth data set, which enables more efficient compression.

Having the ability to detect important events in a timely fashion has many benefits for wireless sensor networks, especially when monitoring environmental phenomena. If the system can identify significant periods in a data series, it can adjust its measurement processes to record the important event with greater accuracy. To achieve this, the system must be able to recognise when an event begins and ends. An energy efficient method for detection involves utilising simple data analysis at sensor level, however, signal noise can disrupt this approach. Introducing an exponentially weighted moving average enables the system to detect events by suppressing the negative effects of signal noise. EWMA can be used to help detect when an event begins and ends and can be used to represent periods of inactivity. The final implementation creates an adaptive process that involves a compression scheme integrated with an event detection mechanism. Compression ratios of 96% can still be achieved despite the adaptive algorithm increasing measurement parameters during significant events. These results prove the accuracy of data representation can be improved without violating minimal energy consumption requirements.
CHAPTER 5

Conclusion

The limited energy supply of sensors is one of the main limitations in wireless sensor networks. Developing procedures to solve this problem is an important source of research and provides the motivation for this study. In-network data compression has been proposed as one solution to the problem, with two popular approaches being piecewise linear approximation and Haar wavelet approximation. We tested both techniques on data collected from field experiments to evaluate their value in reducing energy consumption. Our results found both techniques were effective in reducing the amount of data required for transmission and in many cases with no increase in error. This means we were able to achieve lossless data compression and maintain the integrity of the underlying data series. Our tests also found the piecewise linear approach to be more effective than the Haar wavelet technique for all data series. These findings prove the ability of data compression to reduce transmission cost and should be of significance to environmental wireless sensor applications.

Following on from our promising results with the fixed data compression techniques of piecewise linear approximation and Haar wavelets, an adaptive measurement scheme for soil moisture networks was developed. The adaptive approach extends the piecewise linear representation algorithm by incorporating the ability to modify measurement parameters during important events. A sensor can capture event data with increased accuracy and less approximation by increasing measurement frequency and reducing compression threshold. But to achieve this, the network must first be able to identify significant events. A mechanism to detect soil moisture events was engineered using the Exponentially Weighted Moving Average (EWMA). The EWMA allows a sensor to recognise events with confidence, by removing the effect of signal noise. The adaptive algorithm represented important soil moisture events with superior accuracy, in some cases six times better than the standard compression technique. The adaptive algorithm also achieved a 94% compression ratio on average, despite the increase in data resolution. These results justify the implementation of an event-aware measurement process.
5.1 Future Work

The techniques presented in this dissertation focus on applying in-network data compression to reduce energy consumption. Incorporating reactivity into the sensor network also provides many desirable outcomes including increased data resolution during critical periods. One method of event detection using data analysis has been described in this dissertation, but others do exist. Testing the adaptive algorithm, developed in this dissertation, in field experiments would be the next stage of the evaluation process. All results so far have been obtained through simulation; a true test of its potential can only be completed in the field.

Secondly, evaluating our adaptive algorithm’s effectiveness against similar techniques would also be appropriate. One possible candidate would be data series’ gradients and correlation. Correlation between data series can also be exploited to effectively detect important events. Periods of environmental inactivity are associated with highly correlated sensor readings. However, during significant events, sensor readings are poorly correlated due to the fluctuating data source. Therefore, when correlated data series suddenly diverge, an interesting event is most likely occurring. Correlation between data series can also be used in data aggregation. If three soil moisture sensors are all returning highly correlated data, then the data stream from one can be used to represent all three. This raises the possibility of a hybrid event-aware and correlation-aware measurement scheme.
Title: In-network Data Processing in Sensor Networks

Author: Mark Moss

Supervisor: Dr Rachel Cardell-Oliver

Background

Wireless sensor networks are a promising technology, well adapted to meet a growing need for environmental data collection. Implementation allows spatial and temporal monitoring which yields information for solving complex environmental problems. Further technological development in this field will yield more accurate and valuable information of our surrounding environment. One of the encumbrances existing in sensor networks involves the limited power supply of the remote nodes called motes. Motes have the ability to monitor environmental parameters (such as soil moisture and salinity) and broadcast their results across the network back to a base node. Of all operations, radio communication expends the most power; therefore engineering ways to reduce the amount of radio transmissions can greatly increase the life of the network and in turn deliver more data over a longer time period. In-network data processing provides a means to reduce the number of wireless communications.

Data compression is one in-network data processing technique that can be employed to achieve the goal of lower transmission costs. Processing data locally at ‘mote level’ can lead to data minimisation before it is transported across the network. For networks involving many data transmitting motes, the recipient’s resources are also burdened. For this reason data compression techniques can benefit the recipient as well as the individual motes.

Existing compression algorithms focus on converting the incoming sensor read-
ings into piecewise linear segments. They rely on the observation that over small enough windows of time, samples of environmental data are linear [19]. Consequently, linear graph segments can be used to represent multiple data points and this results in fewer radio transmissions. The only limitation of the compression algorithm is the fact that it must be implemented at the sensor and therefore must be lightweight in regards to processing and memory allocation [10].

Other studies have applied more complex procedures to reduce data volume where data is recorded depending on various factors or events. One study employs a ‘node fitness’ concept by reducing high battery usage operations when energy levels are low and prioritising low consumption operations [17]. Similar techniques have been developed where sensor recordings are reduced during times of environmental inactivity and increased during periods of high activity. These networks can react to important environment events like increasing soil moisture readings during rainfall periods [2]. These approaches conserve battery power and extend the life of the network.

Aim

The aim of the project will be to implement a data compression algorithm that generates linear segments to represent sensor readings as described above. The project will also involve comparing different compression techniques to develop an algorithm most efficient at capturing the underlying data series. Further work expanding on this initial design to incorporate data aggregation among on board sensors and filtering of ‘noisy’ readings will also be undertaken. Upon successfully creating and deploying efficient data compression techniques, the life time of a sensor network should be increased with no measurable loss in data quality.

In-network processing has been described as the enabling technology for long-lived deployments of sensor networks [19]. Research on in-network data compression reveal compression rates of 20-to-1 [1,19] highlighting its significance for wireless sensor networks. Reports of this nature emphasise the need for further investigation on the topic.

The benefits of developing advanced in-network data compression techniques are not restricted to specific environmental variables. New procedures can be applied to any measurable quantity like temperature, light, soil moisture, humidity or pressure which justifies the argument for further research.
Method

The project will follow a straightforward approach involving initial research, followed by implementation, testing, and a dissertation phase. Descriptions of these phases are listed below.

1. Initial Research
   - Literature Review of existing material and TinyOS documentation
   - Completed During: March

2. Implementation
   - Develop rudimentary algorithm to be run on existing data sets.
   - Experiment and test different compression techniques in a simulated environment.
   - Transfer algorithm to TinyOS system by modifying existing data logging software.
   - Expand algorithm to cater for outliers and data aggregation between sensors with a goal to achieve spatial compression.
   - Investigate gateway for data collection from field.
   - Completed During: April - June/July

3. Testing
   - Experiment and test robustness of algorithm with Mica2 Motes and CSIRO Flecks in the field.
   - Reconfigure to improve performance and remove bugs.
   - Implement gateway to transport field results to local database.
   - Completed During: June/July - September

4. Dissertation
   - Complete Dissertation, Speech and Poster
   - Completed During: September - October
Software and Hardware Requirements

Specific requirements revolve around the unique hardware and software required to build wireless sensor networks. These will be available in the CSSE Sensor Network group.

- Berkeley Mica2 Motes fitted with dielectric soil moisture sensors
- CSIRO Fleck nodes and prototype sensor boards
- Superlite GSM Modem
- TinyOS software
APPENDIX B

Fixed Compression Estimation Error

Figure B.1: Euclidean distances for EM5 Data Logger data.

Figure B.2: Euclidean distances for Mica 2 Mote data.
Figure B.3: Euclidean distances for Weather Station Temperature data.

Figure B.4: Euclidean distances for Weather Station Humidity data.

Figure B.5: Euclidean distances for Weather Station Wind Speed data.
APPENDIX C

Event Completion Parameter Analysis

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Maximum Required Moisture Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Darling Range Data (Mica 2 Mote)</td>
<td>45%</td>
</tr>
<tr>
<td>Pinjar Data (EM5 Datalogger)</td>
<td>2%</td>
</tr>
<tr>
<td>Jolimont Data (CSIRO Fleck)</td>
<td>12%</td>
</tr>
</tbody>
</table>

Table C.1: Required Reduction in Soil Moisture to Avoid Incorrect Event Completion.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Average History Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Darling Range Data (Mica 2 Mote)</td>
<td>9.9</td>
</tr>
<tr>
<td>Pinjar Data (EM5 Datalogger)</td>
<td>9.4</td>
</tr>
<tr>
<td>Jolimont Data (CSIRO Fleck)</td>
<td>9</td>
</tr>
</tbody>
</table>

Table C.2: Required History Length $z$ to Avoid Incorrect Event Completion.
Bibliography


