A DISTRIBUTED WAVELET COMPRESSION ALGORITHM FOR WIRELESS MULTIHOP SENSOR NETWORKS USING LIFTING

Alexandre Ciancio and Antonio Ortega

University of Southern California
Signal and Image Processing Institute
ciancio@usc.edu, ortega@sipi.usc.edu

ABSTRACT

We address the problem of compression for wireless sensor networks, where each of the sensors has limited power, and acquires data that should be sent to a central node. The final goal is to have a reconstructed version of the sampled field at the central node, with the sensors spending as little energy as possible. We propose a distributed compression algorithm for multihop, distributed sensor networks based on the lifting factorization of the wavelet transform that exploits the natural data flow in the network to aggregate data by computing partial wavelet coefficients that are refined as the data flows towards the central node. A key result of our work is that by performing partial computations we greatly reduce unnecessary transmission, significantly reducing the overall energy consumption.

1. INTRODUCTION

Low cost programmable sensors can be deployed in the environment to perform a variety of coordinated tasks, such as object tracking, environment monitoring/ surveillance and control [1]. The fast development and decreasing cost of wireless technologies have made sensor networks a key technology for the future, and led to substantial interest on distributed algorithms targeted at these networks [2, 3, 4, 5].

Assume that a number of power-constrained sensors are spread over an area, acquiring data, e.g., temperature measurements. An estimate of the temperature in each point of that area is to be made available at a central node (sink) based on transmissions from the sensors. Each sensor is capable of transmitting data only over a small distance. Communication is done via data hoping, where data from each sensor is forwarded (using other sensors as relay stations) until it reaches the target node. Such a network is referred to as a multihop network.

A simple and naive design would be for each sensor to just transmit a quantized version of its own measurement to the central node. However, this approach would not be exploiting the fact that measurements originated from spatially close sensors are likely to be correlated, and energy would be wasted with the transmission of redundant data to the central node. As an alternative, since data is correlated, it would be reasonable to use some sort of transform as a means to decorrelate the information from sensors, and, therefore, represent the measurements using fewer bits.

Previous works [2, 3] have proposed the use of distributed transforms to decorrelate data. A main drawback of those algorithms is that in a power constrained sensor network scenario, transmission costs account for most (if not all) the energy consumption. Distributed algorithms that rely on a potentially large number of intersensor communications, or do not take into account communication costs as a function of the number of bits and the distances over which bits are transmitted, might in the end be elevating power consumption to unacceptable levels. For example, in the case of distributed wavelet algorithms, the amount of necessary data exchanges and the degree of decorrelation achieved are directly related to the number of levels of decompositions used. Depending on the network properties (data correlation, distances between sensors, etc), a specific number of decompositions may perform better.

Distributed implementation of the wavelet transform poses several challenges. First, if the filters contain anticausal terms, sensors would be required to transmit data backwards (i.e., away from the sink instead of towards the sink) or alternatively to send uncompressed data forward. Second, any data transmitted back and forth over the network has to be quantized, since tramsmissions at full precision can substantialy increase energy consumption, affecting the final performance. In this work we propose a distributed wavelet transform algorithm to decorrelate data as it flows through the network. We eliminate unnecessary transmissions by calculating partial approximations of the wavelet coefficients based on the available data at each sensor. The coefficients are refined at future nodes, as data is forwarded to the sink. We also address the impact of data quantization on the final distortion. We derive an upper bound to the resulting extra distortion introduced by quantization in terms of the bits allocated to the partial data, and use it as a tool to design the quantizers so the extra distortion is within a given threshold.

The paper is organized as follows. Section 2 introduces the proposed algorithm. The quantization effect of the partial data is addressed in Section 3. Section 4 presents some preliminary results. In Section 5 we present our conclusion and discuss future work.

2. THE PROPOSED ALGORITHM

The lifting scheme is an alternative method to compute wavelet transforms. It allows a faster implementation, along with a full in-place calculation of the coefficients [6].

We propose the use of the lifting scheme to generate the wavelet coefficients at each of the sensors. Even-numbered sensors correspond to the even samples and odd-numbered sensors to the odd samples. In-place computation reduces the memory requirements for the sensors. Implementations as the one proposed in [7] enable the partial computation of coefficients at the boundaries. In this

This work was supported in part by CAPES, Brazil.

perspective, the sensors can be divided into groups and increasing levels of the wavelet decomposition would be computed, as long as the extra energy spent in the calculations and transmissions is compensated by the increase in decorrelation.

Efficient implementation of a lifting based distributed wavelet transform becomes challenging once inter node communication costs are taken into account. Fig. 1 illustrates a typica implementation of the lifting algorithm for the 5/3 wavelet, where $d(n)$ denotes the high-pass data and $s(n)$ denotes the low-pass data after the transform is computed. In [8], we proposed the introduction of internode communication, using extra transmissions, to give sensors knowledge about their neighbors' data, allowing them to compute the wavelet coefficients. The reduction in the average number of bits per sensor compensated the cost of the extra transmissions, resulting in a smaller overall energy consumption. However, for a multihop network, where all data already flows in a particular direction, sensors that need to transmit data away from the sink may be wasting resources. A simple solution could be the introduction of a delay, making the system causal, and having the sensors calculating the transform coefficients only after all the necessary data becomes available at a future node. This approach, however, is clearly inefficient, since raw data has to be transmitted until it reaches the node that will process it, the memory requirements for the sensors can increase substantially, not to mention the potentially large increase in the energy consumption due to the raw data transmission itself.

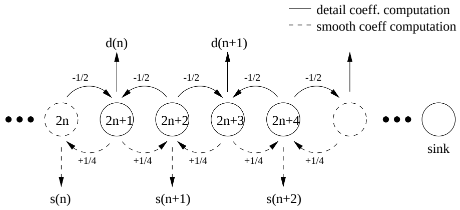

Fig. 1. Lifting implementation of wavelet transform. Sensors exchange data with neighbors in order to compute transform coefficients.

In this work, as a solution to the causality problem, we take advantage of the data flow in the network to compute a distributed wavelet transform. By performing partial computations with the available data, we eliminate unnecessary backward (against the flow) transmissions. The transmission of partially computed coefficients then raises another important issue, given that all transmissions have to be made with finite precision. A poor choice of intermediate coefficient quantization might considerably affect the final distortion. We address the issues of causality and quantization in the following sections.

2.1. The Partial Coefficients Approach

As a means to overcome the causality problem, we propose that each sensor perform partial computations using the available data that arrives with the network flow. Let $H(z) = \sum_{n=-a}^{b} h(n)z^{-n}$, $(a > 0, b > 0)$, be a general filter defined by the transform to be applied to the network. This filter can also be represented as $H(z) = A(z)z + c + B(z)z^{-1}$, where A(z) and B(z) are the anticausal and causal parts of the filter, respectively. In the first step in the partial coefficient approach, the current node receives a quantized version of $B(z)$. Then, it computes and sends to the next node a quantized version of $B(z)z^{-1}+c$. This, and the subsequent nodes will update the coefficient using their local data, until it is fully computed.

The lifting implementation of the wavelet transform facilitates the distributed implementation. The in-place computation greatly reduces the memory requirements for the sensors. Within a lifting implementation it is easy to compute the partial data updates in terms of previous partial coefficients, eliminating the need to transmit extra information. As an example, let $D(n)$ denote the raw (quantized) data and $\bar{D}(n)$ the full coefficient for the n-th sensor. The 5/3 wavelet coefficients using lifting at odd and even sensors are given by:

$$\bar{D}(2n+1) = -\frac{1}{2}D(2n) + D(2n+1) - \frac{1}{2}D(2n+2)$$

$$\bar{D}(2n) = \frac{1}{4}\bar{D}(2n-1) + D(2n) + \frac{1}{4}\bar{D}(2n+1)$$

In our approach, since sensor $2n+1$ does not have access to data from sensor $2n{+}2$, it computes just the partial coefficient $\bar{D}_p(2n+1)$ as $\bar{D}_p(2n+1) = -\frac{D(2n)}{2} + D(2n+1)$. When this partial data arrives at sensor $2n+2$, it will be updated to $\bar{D}(2n+1) = \bar{D}_p(2n+1) - \frac{D(2n+2)}{2}$. Data present at each sensor is illustrated in Fig. 2, for a 1-level 5/3 wavelet. The computations performed at each even and odd sensor for this transform are summarized as follows:

Computations performed at even (2n) sensors:	Computations performed at odd (2n+1) sensors:
$\bullet \bar{D}(2n-1) = \bar{D}_p(2n-1) - \frac{D(2n)}{2}$ $\bullet \bar{D}(2n-2) = \bar{D}_p(2n-2) - \frac{D(2n)}{8}$ $\bullet \bar{D}_p(2n) = D(2n) + \frac{\bar{D}_p(2n-1)}{4}$	$\bullet \bar{D}_p(2n+1) = D(2n+1) - \frac{1}{2}[\bar{D}_p(2n) - \frac{\bar{D}(2n-1)}{4}]$ $\bullet \bar{D}_p(2n) = \bar{D}_p(2n) + \frac{\bar{D}_p(2n+1)}{4}$

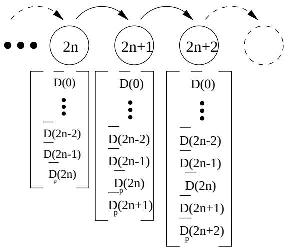

Fig. 2. Partial coefficients are transmitted forward and updated at future sensors until the full coefficients are computed.

Note that the sensors make their computations only based on previous partial or full coefficients, and not on raw data, eliminating the need for extra information. Although this representation is straightforward for the 5/3 wavelet lifting factorization, other filters might require specific optimiztion.

3. PARTIAL COEFFICIENT QUANTIZATION EFFECTS

In standard transform computing, the computations to obtain the transform coefficients are performed at full precision, and only the final coefficients are quantized. However, in a distributed network scenario, transmissions at full precision would significantly increase energy costs, making it necessary to also quantize partial coefficients. In this section we study the impact of partial coefficient quantization on the final distortion, and propose a rule to determine how many bits should be used to quantize the partial information so as to achieve a target level of degradation, in the form of added distortion as compared to calculating coefficients without partially quantized data.

Assume we have a uniform quantizer Q_L, with bin size equal to L. We want to quantize the result of $\alpha X_1 + \beta X_2$, where α and β are known constants ($0 \leq \alpha, \beta \leq 1$) and X_1 and X_2 are random variables uniformly distributed on the interval $[0, 1]$. The resulting quantization $Q_L(\alpha X_1 + \beta X_2)$ gives a mean-squared error of ε. Assume now that a second quantizer Q_ℓ, with bin size equal to ℓ, is used to quantize X_1 and X_2 before the quantizer Q_L. The resulting mean-squared error of $Q_L(\alpha Q_\ell(X_1) + \beta Q_\ell(X_2))$ is ε'. Expressing the MSE for both cases as double integrals gives:

$$\varepsilon = \iint (\alpha X_1 + \beta X_2 - Q_L(\alpha X_1 + \beta X_2))^2 dX_1 dX_2 \qquad (1)$$

$$\varepsilon' = \iint (\alpha X_1 + \beta X_2 - Q_L(\alpha Q_\ell(X_1) + \beta Q_\ell(X_2)))^2 dX_1 dX_2 \qquad (2)$$

A graphical interpretation of the distortion calculation is shown in Fig. 3 for the case where $\alpha = \beta = 1$. The rectangular grid corresponds to the quantization regions for the pair (X_1, X_2). The diagonal lines define the decision regions for the quantizer Q_L. Let $\frac{\ell}{L} = \frac{1}{2^N}$. It can be shown that the number of extra bits that should be assigned to Q_ℓ when compared to Q_L is given by N. Then, by computing the MSE ratio in terms of N, we can relate the additional extra distortion to the bits used to quantize the partial data, defining a rule to design the quantizer Q_ℓ for any given value of $\frac{\varepsilon'}{\varepsilon}$.

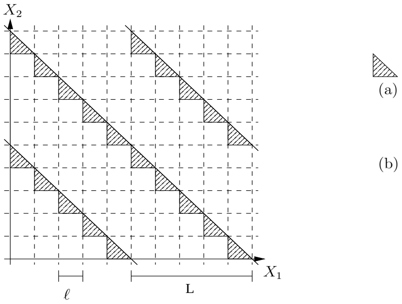

Fig. 3. Graphical interpretation of quantization effects. (a) Regions for which $Q_L(\alpha X_1 + \beta X_2) \neq Q_L(\alpha Q_\ell(X_1) + \beta Q_\ell(X_2))$ (b) Regions for which $Q_L(\alpha X_1 + \beta X_2) = Q_L(\alpha Q_\ell(X_1) + \beta Q_\ell(X_2))$

Calculating the above integrals for the case $\alpha = \beta = 1$ gives us (intermediate steps omitted due to the lack of space):

$$\frac{\varepsilon'}{\varepsilon} = 1 + \frac{4}{2^{2N}} \qquad (3)$$

This result shows that the MSE ratio is independent of the absolute value of the bin sizes of the quantizers, depending only on their ratio. Simulations have shown that the case $\alpha = \beta = 1$ represents a worst case, with the ratio $\frac{\varepsilon'}{\varepsilon}$ decreasing as α and β decrease, and that the results are little affected by the probability distributions of X_1 and X_2.

Fig. 4 shows the theoretical curve from (3) plotted together with simulated results for a number of cases where $\alpha, \beta \leq 1$ and X_1, X_2 are gaussian r.v.'s. In practical transform computations, typical values for α and β might increase or decrease depending on the specific coefficient, but in general they are always smaller than one, making the curve $1 + \frac{4}{2^{2N}}$ a still reasonable approximation for an upper bound of the ratio $\frac{\varepsilon'}{\varepsilon}$. By allocating N extra bits when quantizing X_1 and X_2 than when quantizing $\alpha X_1 + \beta X_2$ we can guarantee that the extra distortion introduced by the intermediate quantization will be bounded by the values given by the theoretical curve. The figure also shows that values of N smaller than 3 introduce a large extra distortion, and the reduction in the extra distortion has a diminishing return as N increases for values above 3.

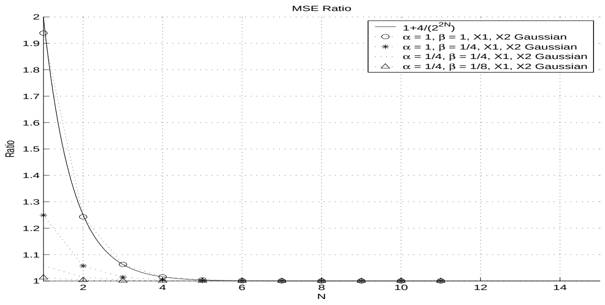

Fig. 4. Theoretical and simulated results for MSE ratio with and without partial coefficient quantization

4. SIMULATION AND RESULTS

In this section we present some preliminary results. We show that the partial coefficient quantization has indeed a major impact on the final distortion, and the results obtained in section 3 can be used to design the partial quantizers such that a good trade-off point between the allocated bits and the extra distortion introduced is achieved. We also compare the performance of the proposed scheme with the cases of raw data transmission (no encoding) and of two-way transmissions (no partial coefficients) in terms of signal-to-noise ratio against transmission cost.

Fig. 5 shows the distortion curves obtained for standard 5/3 wavelet implementations and when partial coefficients are coarsely quantized ($N = 0$) and finely quantized ($N = 3$). It can be seen that the resulting SNR is very sensitive to coarsely quantized partial coefficients. In the sensor network scenario, choosing values of N much larger than 3 will certainly lead to better signal-to-noise ratio, but at the cost of an increase in the overall energy consumption.

For the cost comparisons, the simulations considered a simple 5/3 wavelet, as described in section 2. The input process data was created using a second order AR model, with poles placed such that a reasonably smooth output would be generated from white noise (poles were at $0.99e^{\pm j\frac{\pi}{64}}$). The measurements at the sensors corresponded to a sampling of the output of the AR model. The sensors were assumed to lie at a constant distance d of each other. This restriction, however, is not required in practice. We used uniform quantization and no entropy coding at this point. The cost for the transmission of b bits over the distance d was computed as $b \cdot d^2$.

Figure 6 shows the resulting SNR and Cost for the case where 100 sensors are equally spread over 100 meters. The 2-level wavelet with finely quantized partial coefficients performs better, with raw data transmission being the one spending the most energy. The energy savings are around 30% when compared to raw data transmission for a target SNR.

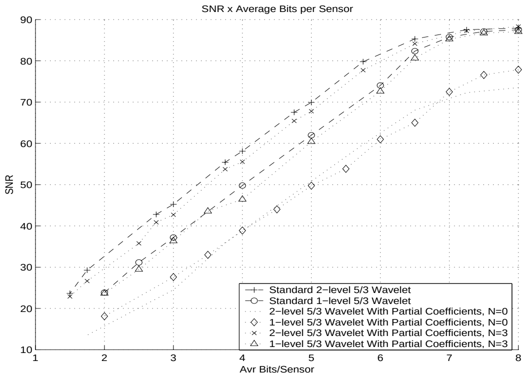

Fig. 5. Effects of Partial Coefficient Quantization. $N = 0$ corresponds to coarse quantization, and $N = 3$ to fine quantization.

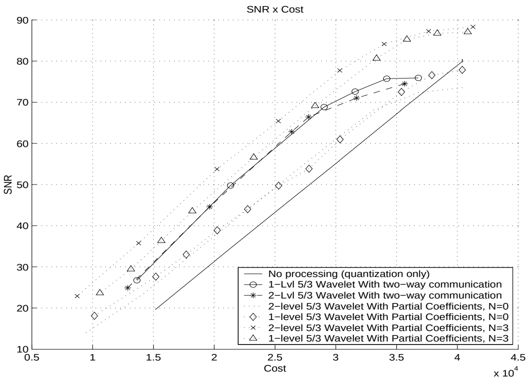

Fig. 6. Cost comparison between methods when 100 sensors are equally spread over 100 meters. Methods that result in a significant reduction in the average number of bits per sensor tend to perform better.

In Fig. 7 we plot the resulting costs for the case where only 10 sensors were spread over the same distance. It can be seen that methods that require more communication tend to perform worse than the simpler ones. This might be attributed mainly to the fact that in a large network, a large portion of the transmissions (and the energy consumption) are related to simply forwarding already computed coefficients, and not with their computation. Therefore, a method that reduces the average number of bits per sensor also reduces the computational load due to data forwarding, resulting in a smaller overall cost. Smaller networks should benefit from methods that do not introduce large extra local costs, since the energy spent with coefficient computation (including due to finer quantization for partial coefficients) and data forwarding have the same magnitude.

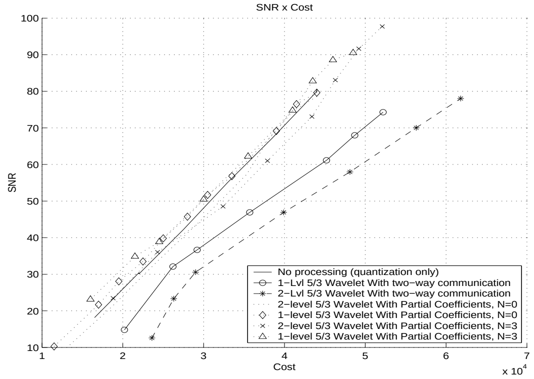

Fig. 7. Cost comparison between methods when 10 sensors are equally spread over 100 meters. Most methods perform similarly, with 1-lvl wavelet with finely quantized coefficients being just slightly better than simple raw data transmission.

5. CONCLUSION AND FUTURE WORK

We have proposed a distributed wavelet algorithm for wireless multihop sensor networks that exploits the data flow direction in the network to perform partial computations to approximate the wavelet coefficients using the available data that arrives at each sensor. We have derived an upper bound to the distortion introduced by partial data quantization, and used it as a means to design the partial quantizers such that a good trade-off between additional distortion and increase in cost (due to the extra bits to be transmitted) was achieved.

Simulations have shown that the proposed method can significantly reduce transmission costs depending on the network configuration. Also, the comparison between figures 6 and 7 suggests that the network configuration can be optimized by breaking the network into regions that would operate using different coding schemes depending on the physical location of sensors, reducing the overall energy consumption.

A number of topics are still to be addressed. Measurements at the sensors can consist of vector data, meaning that sensors are acquiring data over time. In this case, local correlation at each node can also be exploited. Different quantizers, other than uniform, also might be used. Under current development is the extension of the proposed algorithm to the 2D case, where we make use of side information to differentiate between possible paths.

6. REFERENCES

[1] C. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.

[2] Sergio D. Servetto, "Distributed signal processing algorithms for the sensor broadcast problem," *Conference on Information Sciences and Systems, The Johns Hopkins University*, March 2003.

[3] M. Gastpar, P. Dragotti, and M. Vetterli, "The distributed karhunen-loève transform," in *Proceedings of the 2002 International Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, December 2002.

[4] A. Wang and A. Chandraksan, "Energy-efficient dsps for wireless sensor networks," *IEEE Sigal Processing Magazine*, pp. 68–78, July 2002.

[5] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Sigal Processing Magazine*, pp. 51–60, March 2002.

[6] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," Technical report 1995:6, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, (ftp://ftp.math.sc.edu/pub/imi_95/imi95_6.ps), 1995.

[7] W. Jiang and A. Ortega, "Lifting factorization-based discrete wavelet transform architecture design," *IEEE Transactions on Circuits ans Systems for Video Technology*, vol. 11, no. 5, pp. 651–657, May 2001.

[8] A. Ciancio and A. Ortega, "A distributed wavelet compression algorithm for wireless sensor networks using lifting," in *Proceedings of the 2004 International Conference on Acoustics, Speech and Signal Processing - ICASSP04*, Montreal, Canada, May 2004.