Energy Efficient Distributed JPEG2000 Image Compression in Multihop Wireless Networks

Huaming Wu and Alhussein A. Abouzeid
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute, Troy, New York 12180
Email: wuhm@rpi.edu, abouzeid@ecse.rpi.edu

Abstract— **The problem of energy efficient image transmission over a resource constrained multi-hop wireless network is considered. Two methods of data exchange in distributed wavelet transform are proposed and investigated with respect to energy consumption and image quality. An energy-balancing distributed JPEG2000 image compression scheme which uses a combination of tiling of images and load balancing by nodes rotation is proposed. Simulation results show that the proposed scheme prolongs the system lifetime by up to 4 times and has a normalized total energy consumption comparable to centralized image compression.**

I. INTRODUCTION

Advances in visual sensors [1], [2] and wireless communication have enabled the development of low-cost, low-power visual multihop wireless networks, which have recently emerged for a variety of applications, including environmental and habitat monitoring, target tracking and surveillance [3], [4]. However, representing visual data requires a large amount of information, leading to high data rates, which in turn requires high computation and communication energy. Consequently, visual sensor networks present several challenges beyond those common to low data rate sensing, such as acoustics, temperature or pressure.

We consider compressing and transmitting images in a multihop wireless network. We focus on the design and performance evaluation of distributed image compression algorithms. The benefit of using distributed image compression in sensor networks can be illustrated in the following two cases. In the first case, nodes have extremely constrained *computation* power. Hence, a node does not have sufficient computation power to completely compress a large raw image. In this case, a distributed method to share the processing task is required to overcome the computation power limitation of each single node. In the second case, even if nodes are not extremely computation power constrained, but are battery operated, distributing the computation load of processing every raw image among otherwise idle processors of other nodes extends the overall lifetime of the network.

In this research, we have made a first attempt at exploring the performance of distributed image compression in energy constrained multihop wireless (e.g. ad hoc or sensor) networks. By exploiting the characteristics of the Discrete Wavelet Transform (DWT), we propose a distributed JPEG2000 image compression scheme where nodes compress an image while forwarding it to the destination subject to a specific image quality constraint. In particular, we propose two data exchange methods of distributed wavelet transform and investigate their performance in terms of energy consumption and image quality over a multihop wireless network. By using an image tiling technique for the raw source images, our scheme is simple and easy to implement while still satisfying image quality requirement. To balance the computation load among nodes, the proposed schemes also use node rotation. Simulation results show that our scheme prolongs the system lifetime by up to 4 times and has a total energy consumption comparable to the centralized algorithm.

Up to our knowledge, distributed image compression in wireless ad hoc networks has not been studied in the literature. However, our work has been inspired by a variety of other research efforts. We describe below some of the ideas and basic concepts that inspired our work, and reflect on how our work relates to current research in the area of sensor network applications.

While early research efforts in wireless sensor networks did not investigate the issues of node collaboration, focusing more on issues in the design and packaging of small, wireless devices [5], [6], more recent efforts (e.g. [7], [8]) have considered node collaboration issues such as data “aggregation” or “fusion”. Our approach of distributed image compression falls within the domain of techniques that apply the concept of *in-network processing*, i.e. processing in the network by computing over the data as it flows through the nodes. It is worth noting that current aggregation functions (e.g., “maximum” and “average” [7]) are limited to scalar data. Our approach can be viewed as an extension to vector data aggregation. Previous *distributed signal processing/compression* problems (e.g. [9], [10]) exploit correlations between data at close-by sensors in order to jointly compress or fuse the correlated information resulting in savings in communication energy. Fusion of correlated images is not discussed in this paper.

In *parallel distributed computing theory* [11], a problem (or task) is divided into multiple sub-problems (or sub-tasks) of smaller size (in terms of resource requirements). Every node solves each subproblem by running the same local algorithm, and the solution to the original problem is obtained by combining the outputs from the different nodes. Our approach to the design of distributed image compression is similar in concept, in that we distribute the task of image encoding/compression to multiple smaller image encoding/compression sub-tasks.

However, a key difference is that distributed computation theory typically focuses on maximizing the speed of execution of the task while our primarily concern here is reducing the total energy consumption subject to a required image quality.

Thus, our proposed approach of image compression intersects with the literature on parallel wavelet transform, which primarily focuses on parallel DWT on special purpose hardware (e.g., application specific VLSI architectures [12], FPGAs [13], DSPs [14]) or general purpose multiprocessor systems [15], [16]. The difference, of course, is that previous parallel wavelet transform research assumes the energy cost of data movement between processors to be zero or very small compared to the wavelet transform computation. This assumption is no longer valid in multihop wireless networks due to costly wireless communication among nodes. Furthermore, the main objective of previous parallel wavelet transform is to speedup the transform. Wireless nodes operating on battery power will possibly have their primary objective as the conservation of their energy reserves rather than algorithm run time.

Distributed Fast Fourier Transform (FFT) in a sensor network was considered in [17]. Our work on DWT is similar in spirit, in the sense that FFT and DWT are two signal processing techniques. However, our problem has two additional dimensions. First, DWT is just one component of the image compression and transmission problem, and in that sense our problem includes more complex network-wide challenges. Second, while the spectrum of possible outcomes of an FFT is limited, in the sense that the FFT is either successfully computed or not, the image encoding and transmission problem offers a wide continuum of solutions in terms of the received image quality.

Recently, applications of classical image processing techniques in sensor networks have gained a lot of interest for the parallels between image processing and sensor networks. By mapping individual sensors as pixels in an image, [18] examined cleaning of uncorrelated sensor noise, and the decentralized detection of edges. Ganesan et al. [19] have proposed a generalized hierarchical architecture for multi-resolution querying of regularly placed sensor networks that is based on wavelet transforms. Servetto [20] also exploits wavelet transforms to decorrelate sensor data to address the sensor broadcast problem where every sensor observes only one pixel. However, these works did not consider the problem of compressing and transmitting images in sensor networks.

Finally, we believe it is appropriate to also mention some of the projects related to visual sensor networks. In [21], a data centric routing protocol is proposed for efficient image retrieval in wireless ad hoc networks by using metadata of images. The issue of the transmission/encoding of the image, once located, is not studied. A topology control approach to maximize the network lifetime of a wireless video surveillance network by arranging nodes location is presented in [22].

The rest of the paper is organized as follows. The system model is described in Section II. Section III introduces the distributed JPEG2000 image compression scheme. Simulations of the proposed scheme are presented in Section IV. We conclude the paper in Section V.

II. SYSTEM MODEL

We consider a densely deployed multihop wireless network in which some of the nodes are camera-equipped. Every camera-equipped node can respond to an image query by generating a raw image (e.g. snapshot of its sensing area in the case of a sensor network) and transmitting this image to the destination (sink). When sending an image request (query), the destination node specifies the desired image quality. A cluster based routing mechanism is assumed to be in place such that nodes can self-organize into a two-tiered cluster (e.g. [23]). In this initial investigation, the communication environment is assumed to be contention-free. A *session* refers to a source sending one image to a destination, in response to receiving a request from the destination. It is worth noting that there can be multiple sessions at a time.

For this study, we use a transceiver energy dissipation model similar to the one proposed in [24]. The energy consumed in transmission per bit is

$$E_{TX} = \epsilon_e + \epsilon_a d^{\alpha} \qquad (1)$$

and the energy consumed in reception per bit is

$$E_{RX} = \epsilon_e \qquad (2)$$

where ϵ_a is the energy dissipated per bit per m^2, ϵ_e is energy consumed by the circuit per bit, d is the distance between a wireless transmitter and a receiver, and $2 \leq \alpha \leq 4$ is the path loss parameter [25]. The energy consumed in computation per bit is

$$E_{DWT} = \gamma \qquad (3)$$

where γ is the energy dissipated for one level wavelet transform compression (to be described later) per bit in this study. Energy spent in quantization and entropy coding per bit is[1]

$$E_{ENT} = \delta. \qquad (4)$$

The performance metric in this paper is *system lifetime*, which is defined as the time duration from the time when the network starts working until the time when the first node in the network fails due to depleted energy.

Image quality is measured using the Peak Signal-To-Noise Ratio (PSNR) metric, which is defined as (in decibels)

$$PSNR = 20log_{10}\frac{2^b - 1}{E\|x(i,j) - \hat{x}(i,j)\|}$$

where $x(i,j)$ is the pixel value of original image, $\hat{x}(i,j)$ is of the reconstructed image and b is the bit per pixel (bpp) of the original image. We recognize the PSNR does not always accurately model perceptual image quality, but we use it because it is a commonly used metric in the literature. In addition, we also show a few examples on a test image to show certain aspects of the received image that are not revealed by the PSNR value.

---

[1] γ and δ are estimated by `JouleTrack` [26], detail in Section IV.

III. DISTRIBUTED IMAGE COMPRESSION

Since we are investigating the problem of transmitting images in multihop wireless networks, it is beneficial to start by describing some of the background of image compression as it relates to this paper. Then we describe our approach of distributed JPEG2000 image compression.

A. Image Compression

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. Image compression aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible. Among a variety of image compression algorithms, JPEG [27] is a commonly used standard for still image compression. More recently, the wavelet transform has gained widespread acceptance in signal processing in general, and in image compression research in particular. In many applications, wavelet-based schemes (also referred to as subband coding) outperform other coding schemes. Wavelet-based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. Because of their inherent multi-resolution nature [28], wavelet coding schemes are especially suitable for applications where *scalability* and *tolerable degradation* are important. Thus, we choose the new wavelet-based image compression standard JPEG2000 [29] as the image compression scheme is this study.

Wavelet transform coding first transforms the image from its spatial domain representation to a different type of representation using wavelet transform and then codes the transformed values (coefficients) [30]. A typical wavelet-based image compression system is shown in Fig. 1. It consists of three closely connected components namely Forward/Reverse transformer, Quantizer/Dequantizer and Entropy encoder/decoder. In terms of energy dissipation of JPEG2000 compression/decompression, wavelet transform is the dominant part [31].

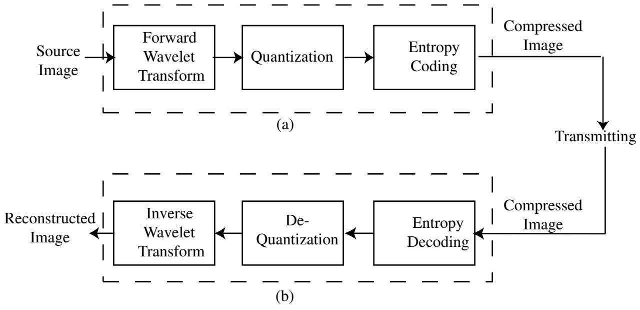


Fig. 1. A typical wavelet-based image compression (a)encoder (b)decoder.

The fundamental concept behind wavelet transform is to split up the frequency band of a signal (image in our case) and then to code each subband using a coder and bit rate accurately matched to the statistics of the band. There are several ways wavelet transforms can decompose a signal into various subbands. These include uniform decomposition, octave-band decomposition, and adaptive or wavelet-packet decomposition [32]. Out of these, octave-band decomposition is the most widely used. This is a non-uniform band splitting method that decomposes the lower frequency part into narrower bands and the high-pass output at each level is left without any further decomposition.

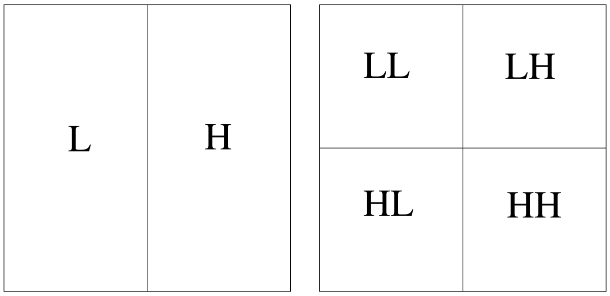
The octave-band decomposition procedure can be described as follows. A Low Pass Filter (LPF) and a High Pass Filter (HPF) are chosen, such that they exactly halve the frequency range between themselves. First, the LPF is applied for each row of data, thereby getting the low frequency components of the row. When viewed in the frequency domain, the output data contains frequencies only in the first half of the original frequency range due to the LPF is a half band filter. Therefore, they can be sub-sampled by two according to Shannon's sampling theorem, so that the output data contains only half the original number of samples. Then, the HPF is applied for the same row of data, and similarly the high pass components are separated. The low pass components are placed at the left and the high pass components are placed at the right side of output row as in Fig. 2(a). This procedure is done for all rows, which we term as 1-D wavelet transform. Next, the filtering is done for each column of the intermediate data. This whole procedure is called a 2-D wavelet transform. The resulting two-dimensional array of coefficients as in Fig. 2(b) contains four bands of data, each labelled as LL (low-low), HL (high-low), LH (low-high) and HH (high-high). The LL band can be decomposed once again in the same manner, thereby producing even more sub-bands as in Fig. 2(c). This can be done up to any level, thereby resulting in a pyramidal decomposition as depicted in Fig. 2.

After the wavelet transform, bit allocation [33], quantization and entropy coding [34] will be applied to get the final compressed image. The details of those techniques are out of the scope of this paper and we do not discuss them here.

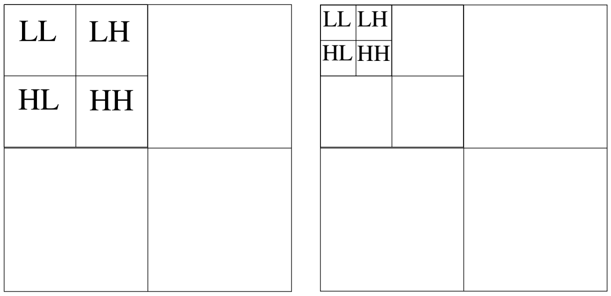
B. Distributed Image Compression

The basic idea of our distributed JPEG2000 image compression is distributing the workload of wavelet transform to several groups of nodes along the path from the source to the destination. Among the issues in the design of distributed wavelet-based image compression, data (e.g. raw image or intermediate results) exchange is of key importance due to the incurred wireless communication energy. In the research of parallel wavelet transform, data is broadcasted to all processors to speedup the execution time [35] which may increase the energy consumption. Two approaches of data exchange will be investigated and compared with respect to image quality and energy consumption; 1)Method 1: Divide by rows/columns; 2)Method 2: Tiling.

1) Method 1: In this method, we consider the data partitioning scheme proposed in traditional parallel wavelet transform [35] when applied to a multihop wireless network. But here, the energy consumed for data exchange is considered. Fig. 3 illustrates the data partitioning scheme for one wavelet decomposition level. At first, the data is partitioned into n



(a) 1-D wavelet transform (b) Single level decomposition



(c) Two level decomposition (d) Three level decomposition

Fig. 2. Illustration of wavelet spectral decomposition and ordering. L stands for low frequency component and H stands for high frequency component.

blocks $R_1, R_2, \ldots, R_n$ where each block consists of one or more rows. Then, each node runs 1-D wavelet transform on R_i. Once the 1-D wavelet transform is done on rows, one node collects the intermediate results $Q_1, Q_2, \ldots, Q_n$ and divides the results into m blocks $I_1, I_2, \ldots, I_m$. Then each node applies 1-D wavelet transform on I_i. Finally, a node gathers the 2-D wavelet transform results $J_1, J_2, \ldots, J_m$. Although there is no quality loss compared to centralized wavelet transform, it should be noted that this partitioning approach requires intensive data communication because of the large number of data exchange.

An example of distributed image compression using four nodes on each level is shown in Fig. 4. After receiving a query from a source node s, the cluster head c_1 selects a set of nodes P_1 in the cluster which will take part in the distributed wavelet transform then informs s. s divides the raw image data by rows and transmits them to P_1 (p_{11}, p_{12}, p_{13} and p_{14}). Those nodes run 1-D wavelet transform algorithm on their received data then send the intermediate results back to c_1. c_1 combines the results and divides data by columns to P_1 again. These nodes process data and send results (Level 1 data in Fig. 4) to the next cluster head c_2. After receiving the results, c_2 chooses a part of the results (corresponding to LL in Fig. 2(b)) and divides into pieces for P_2 (p_{21}, p_{22}, p_{23} and p_{24}). c_2 will code and send the other parts of received data (corresponding to LH, HL and HH in Fig. 2(b)) directly to the next cluster head c_3. The nodes in P_2 (p_{21}, p_{22}, p_{23} and p_{24}) will also send their processed results (corresponding to level 2 wavelet decomposition as in Fig. 2(c)) to c_3 after running 1-D wavelet transform twice. If required by the application (depending on the image quality specified by the query), this procedure may continue on c_3 and its following nodes until the final compressed image reaches the destination (sink) node. As shown in Fig. 4, the cluster heads (c_1, c_2 and c_3) have to collect and transmit the intermediate 1-D wavelet transform results (indicated by two-way arrows) during each wavelet decomposition level.

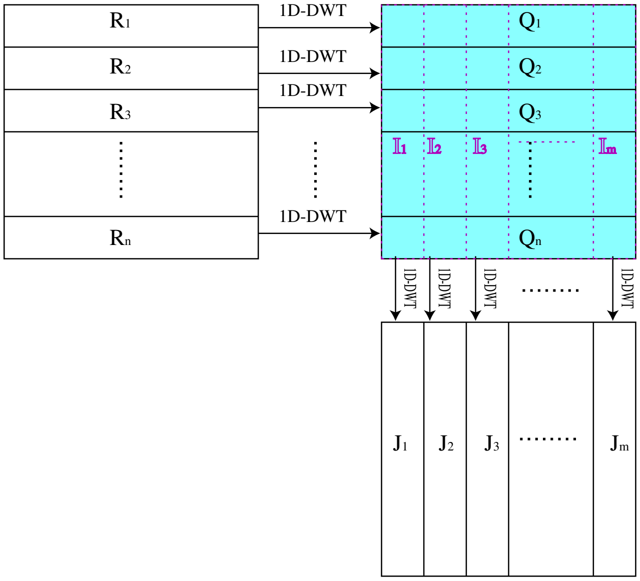


Fig. 3. Traditional data partitioning approach: divide by rows/columns for one wavelet decomposition level. The set of R_i ($i = 1, 2, \ldots, n$) is the data to be wavelet transformed. The set of J_i ($i = 1, 2, \ldots, m$) is the 2-D wavelet transform results.

2) Method 2: Tiling, which is used in JPEG [27], can also be used in wavelet based image compression. Normally, the wavelet decomposition is computed on the entire image, which inhibits annoying compression blocking artifacts that occur at low bit rate coding. However, JPEG2000 [29] also supports the concept of image tiling for operation in low memory environments. In this case, the image is partitioned into tiles (blocks), and then each block is sent to a node to do 2-D wavelet transform *independently*. Once a node completes its 2-D wavelet transform, it sends the result back to a center node. It is pointed out [16] that the processing of tiles independently leads to a rate-distortion loss and blocking artifacts as the number of tiles and/or processors is increased. We found that the quality loss and blocking artifacts are small when the number of tiles is small or when the bit rate of compressed image is not very low. Fig. 5 illustrates the image quality degradation of tiling. It is observed that the quality loss of tiling with four tiles (Fig. 5(c)) compared to the result without tiling (Fig. 5(a)) is smaller than $0.2dB$ in terms of PSNR. The blocking artifacts are also negligible with a moderate bit

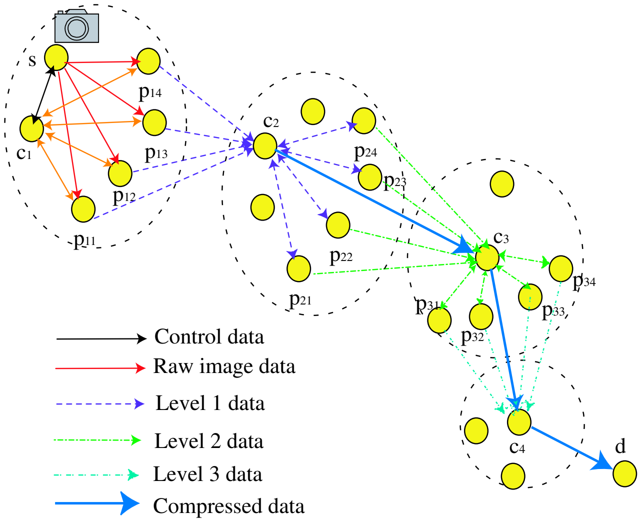


Fig. 4. Illustration of data exchange method 1 of distributed wavelet-based image compression in a multihop wireless network. Three levels of wavelet decomposition. s is the source node. d is the destination (sink) node. c_i is the cluster head of cluster i. Two-way arrows between cluster heads and nodes indicate that the cluster heads collect and transmit the 1-D wavelet transform results during each wavelet decomposition level.

rate (Fig. 5(d)). Therefore, it is still applicable in distributed wavelet-based image compression if the number of tiles is small or for not very low bit rate. Post processing techniques can also be applied at the destination to cope with the blocking artifacts [36].

Fig. 6 shows an example of distributed image compression using four tiles and three wavelet decomposition levels. Similar to method 1, after receiving a query from a source node s, the cluster head c_1 selects a set of nodes P_1 in cluster which will take part in the distributed wavelet transform then informs s. s divides raw image data into tiles and transmits them to P_1. Those nodes run 2-D wavelet transform algorithm on their received image data then send the results (Level 1 data in Fig. 6) to the next cluster head c_2. For each tile, c_2 chooses a part of the results (corresponding to LL in Fig. 2(b)) and sends it to one of its set of processing nodes (p_{21}, p_{22}, p_{23} and p_{24}). Thus, p_{21}, p_{22}, p_{23} and p_{24} get the corresponding LL components of different tiles. c_2 will also combine the other parts of received data (corresponding to LH, HL and HH in Fig. 2(b)) from all tiles then code and send directly to the next cluster head c_3. In this way, the final compressed image reaches the destination node.

C. Quantization and Coding

The bit allocation, quantization step and entropy coding method in the proposed distributed image compression use the same method as in the centralized JPEG2000 image compression. In the Embedded Block Coding method which is used in JPEG2000 standard [37], each subband (corresponding to LL, LH, HL and HH component at each wavelet decomposition level) is divided into small blocks called "code blocks". And then each code block[2] is coded independently. More details of Embedded Block Coding in JPEG2000 can be found in [29], [38].

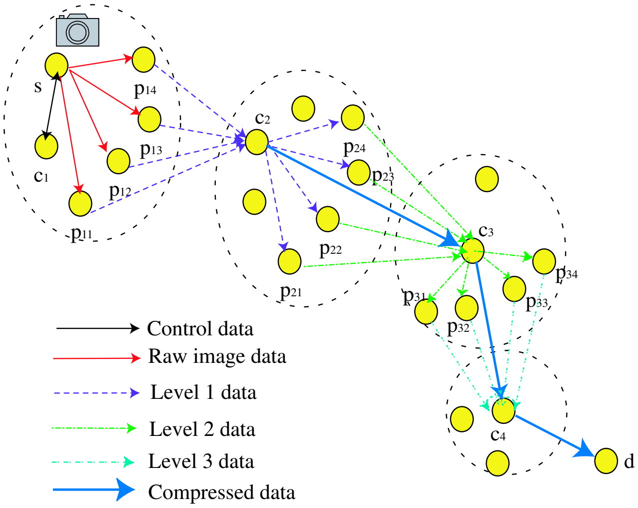


Fig. 6. Illustration of data exchange method 2 of distributed wavelet-based image compression in a wireless multihop network. Three levels of decomposition. s is the source node. d is the destination node. c_i is the cluster head of cluster i.

In both data exchange methods of our proposed distributed image compression, at each wavelet decomposition level, quantization and entropy coding are applied in the same way of centralized image compression to the data corresponding to LH, HL and HH components. It is worth noting that, in the discussion of data exchange methods, nodes run entropy coding before transmitting data (raw image pixels or transformed LL coefficients). While in the centralized or previous parallel version of wavelet transform, data between wavelet transform levels are exchanged directly without any entropy coding since no or little cost are considered for this exchange. Thus the entropy coding is applied only once after the required level of wavelet decomposition is reached (as depicted in Fig. 1). In our proposed scheme, it is applied before data exchange in order to save communication energy [39].

In the cases of distributed image compression, if the distance between source and destination is not large enough for one wavelet decomposition level per cluster, the last node on the path to the destination will run wavelet transform algorithm on remaining data till required decomposition level. For example, c_4 will run twice wavelet transform algorithm on corresponding data in Fig. 6 before sending the final compressed image to the destination, if the required wavelet decomposition level was 5.

We believe that a discussion of the choice of letting the last node finish the compression is necessary. There are two intuitive reasons:

1) The computational cost at higher level of wavelet decomposition is small compared to lower level of decom-

---

[2] The size of code blocks is a flexible parameter. A typical size of the code block is 64 by 64.

(a) JPEG2000 without tiling. Bit rate is $0.1bpp$. $PSNR = 29.30dB$.

(b) JPEG2000 with a tile size 64×64 pixels. Bit rate is $0.1bpp$. $PSNR = 25.12dB$.

(c) JPEG2000 with a tile size 256×256 pixels. Bit rate is $0.1bpp$. $PSNR = 29.12dB$.

(d) JPEG2000 with a tile size 64×64 pixels. Bit rate is $0.5bpp$. $PSNR = 35.67dB$.

Fig. 5. Compression of Lena image.

position (e.g., the computational cost at the $i + 1^{th}$ level of decomposition is only $1/4$ of i^{th} level).

2) For each session, the communication cost of the node nears the destination is smaller than the communication cost of its previous node on the path.

In fact, the required number of wavelet decomposition level in practical is small (e.g. recommend set to 5 for the best PSNR performance cost ratio in [40]). Furthermore, for a sensor network, which is normally assumed to be large scale, the number of hops between the source and destination (sink) is most likely large enough for one wavelet decomposition level per cluster. Thus, the nodes closest to the sink will less likely be overburdened with computational requirements.

D. *Load Balancing*

While our above approaches divide the computation workload of a single JPEG2000 image compression among multiple nodes in each wavelet decomposition level, they only consider the response of a single query (defined earlier as "session") between a source and a destination. If the processing node sets were chosen a priori and fixed throughout the system lifetime, it is easy to see that those unlucky nodes chosen to be processing nodes would run out of their energy quickly. Thus, we include randomized rotation of the processing node set such that it rotates among the various nodes among sessions in order to not drain the battery of a single node. In this way nodes will evenly share the workload of the network.

IV. SIMULATION

In this section, we perform two sets of simulations that compare our proposed distributed JPEG2000 image compression with the centralized algorithm based on two performance metrics: total energy consumption and system lifetime.

The values of the parameters of the computation energy model (3) and (4) are estimated as follows. We have employed `JouleTrack` [26] to estimate the energy consumption for an existing JPEG2000 coder[3] [40]. The experiment data in terms of energy expended by a StrongARM SA -1100 processor at 206Mhz is measured when running JPEG2000 image compression algorithm on test image *Lena* (512x512). From the experiment, the value of γ in (3) is estimated to be 220×10^{-9} Joule/bit and the value of δ in (4) is estimated to be 20×10^{-9} Joule/bit.

The values of the parameters of the wireless communication energy model (1) and (2) are the typical values $\epsilon_a = 100\times10^{-12}$ and $\epsilon_e = 50\times10^{-9}$ as for example in [41]. The communication range of a node d and is chosen to be 10 m for all nodes in this study, and $\alpha = 2$.

For each topology of network, the nodes organize into clusters according to a clustering algorithm[4] [41]. During each of the two sets of simulations reported below, the values of the parameters of our proposed distributed JPEG2000 image compression are chosen as follows. Every node is assumed to have an image *Lena* of size 512×512 pixels with 8 bits per pixel (bpp). We fix the number of nodes running distributed wavelet transform within each wavelet decomposition level to be 4. Let L denote the desired decomposition level of wavelet transform. The maximum value of L for every image is chosen to be 5 (determined by the size of the test image and performance cost ratio). For simplicity, the desired image quality Q is given in terms of the bit rate of final compressed image for this set of simulations. Similar trends are observed for other values not reported here (for space considerations).

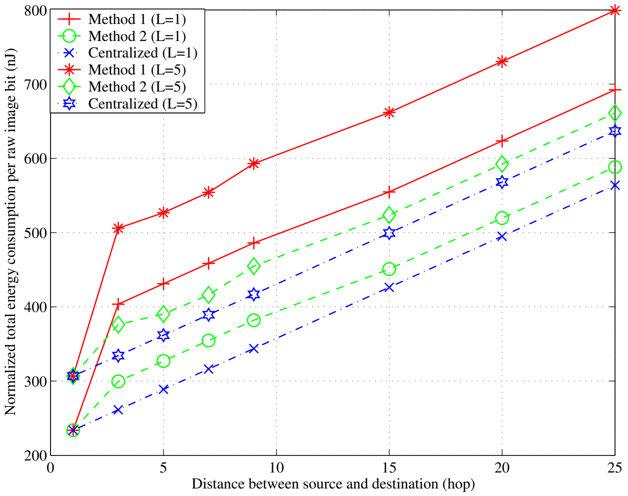


Fig. 7. Normalized total energy dissipation per raw image bit versus distance between source and destination for different desired decomposition level L. $Q = 1bpp$.

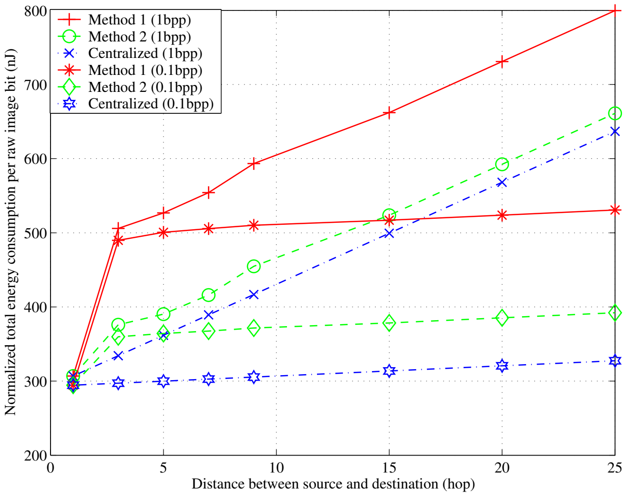


Fig. 8. Normalized total energy dissipation per raw image bit versus distance between source and destination for different desired image quality Q. $L = 5$.

A. *Total Energy Consumption*

We consider a network with 500 nodes randomly placed in an area of 125×125. The node closest to the center is chosen to be the source. The destination nodes are chosen with varied distance from 1 to 25 hops between the source and the destination. The total energy dissipation of a session, which includes communication among nodes and wavelet transform, is measured for different source destination distance.

The comparison between the total energy dissipation of the three schemes (centralized, distributed Method 1 and distributed Method 2) is shown in Fig. 7 and Fig. 8. We examine the energy consumption with respect to the distance between source and destination for different wavelet decomposition level L and image quality Q. It is observed in these figures that the total energy consumption of Method 1 is much higher than the other two under all circumstance. The total energy

---

[3] All options of this JPEG2000 coder are the default values except decomposition level that is chosen to be 1 since γ is the energy dissipated for one level wavelet image compression per bit. Recall that γ counts the energy consumed by wavelet transform, quantization and entropy coding.

[4] The choice of clustering algorithm is arbitrary and is done simply for simulation purpose. We leave the examination of using other clustering algorithms in our proposed distributed image compression for future research.

dissipation for Method 2 is slightly higher than centralized image compression (where only source node runs JPEG2000 image compression algorithm), particularly for moderate or high bit rate quality requirement. However, in terms of system lifetime, the distributed approaches have better performance, as described in the next simulation results.

B. *System lifetime*

In this section, we only compare Method 2 (and not Method 1) with the centralized image compression since it is shown in the previous section that it performs much better than Method 1 in terms of total energy consumption.

We consider five connected networks of size $N = 25$, 50, 100, 200 or 500 nodes. Nodes are randomly placed in a square area with average node degree (i.e. number of neighboring nodes) to be 10. The area size is computed to keep the network asymptotically connected[5]. We choose the node which is the closest to the center of the field as the source. For every session, the destination is randomly chosen. Each node is initially given 1 Joule of energy. The simulation is stopped if any node in the network depletes its energy. The system lifetime in terms of number of sessions is measured for both distributed and centralized JPEG2000 image compression.

The comparison between the system lifetime achieved by using our approach against a centralized scheme is shown in Fig. 10 and Fig. 9. Very similar trends are observed for different values of Q and L in these figures. In the case of conventional centralized image compression, the source node "dies" quickly since it consumes power at a very high rate compared to other nodes in the network. The system lifetime is almost constant for different network sizes. On the other hand, the results of Method 2 result in a much longer lifetime since it divides the workload uniformly (statistically speaking) among all the nodes in the network. The results show that considerably longer system lifetime (up to 4 times) can be achieved when using distributed JPEG2000 image compression (Method 2) algorithm.

V. CONCLUSION AND FUTURE WORK

In this paper, we studied the problem of energy balanced image encoding/transmission in multi-hop wireless networks. The design and evaluation of a distributed JPEG2000 image compression scheme is presented. We use a combination of tiling of images, and load balancing by nodes rotation to achieve much longer system lifetime compared to the centralized image compression. The proposed scheme is simple and easy to implement. Performance evaluation shows that this distributed scheme can have significantly longer system lifetime while satisfying the performance constraint in terms of a target image quality compared to a centralized approach.

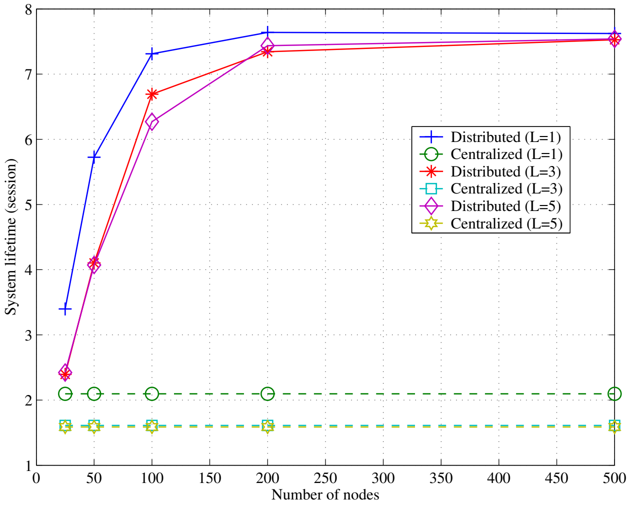


Fig. 9. System lifetime comparison: distributed versus centralized for different desired decomposition level L. $Q = 1bpp$.

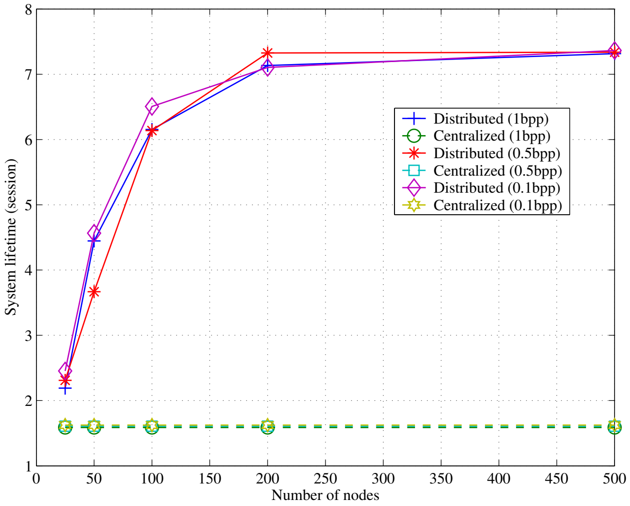


Fig. 10. System lifetime comparison: distributed versus centralized for different desired image quality Q. $L = 5$.

This work provides an important proof of concept that shows the benefits and feasibility of distributed image compression. Several aspects of future research are the impact of wireless link errors, dynamic number of processing nodes, and irregular clusters. Currently, we fix the number of nodes in each processing set. The choice of the number of processing nodes can be further investigated. Another possible direction of future research is to utilize multi-path routing to enhance the performance of distributed image compression. Furthermore, the impact of overlapping tiles on image quality and energy consumption is left for future work. In order to verify the assumptions and performance about our approach to distributed image compression, we are currently extending the network simulator `ns` [43] to simulate our algorithms operation with existing network protocols (e.g. routing and MAC). In the future, we plan to validate our approach on a sensor network

---

[5] Let L denote the side length of the square. Let g denote the average degree of a node. From [42], we know that this average node degree should be $\Theta(\log N)$ to keep the network asymptotically connected. The side length L of every network size is calculated from

$$L = \lfloor d\sqrt{N\pi/g} \rfloor$$

testbed.

ACKNOWLEDGMENT

The authors would like to thank Ms. Ying Liu, for many useful discussion on the wavelet transform, Prof. William A. Pearlman for valuable input and discussion, and the anonymous reviewers, for multiple suggestions to improve the quality of presentation.

REFERENCES

[1] "Digital pixel sensor," 2004. [Online]. Available: http://www-isl.stanford.edu/~abbas/group/

[2] A. E. Gamal, "Trends in CMOS image sensor technology and design," in *Proceedings of IEEE International Electron Devices Meeting*, San Francisco, CA, December 2002.

[3] D. Estrin, "Sensor network research: Emerging challenges for architecture, systems, and languages," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, ser. ACM SIGPLAN notices, C. Norris and J. J. B. Fenwick, Eds., vol. 37, 10. New York: ACM Press, October 5–9 2002, pp. 1–4.

[4] A. E. Gamal, "Collaborative visual sensor networks," 2004. [Online]. Available: http://mediax.stanford.edu/projects/cvsn.html

[5] G. Zaruba and S. Das, *Off-the-shelf enablers of ad hoc networks*. New York: IEEE Press Wiley, 2003.

[6] "SmartDust," 2003. [Online]. Available: http://robotics.eecs.berkeley.edu/~pister/SmartDust/

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the sixth annual international conference on Mobile computing and networking*. ACM Press, 2000, pp. 56–67.

[8] W. Zhang, Z. Deng, G. Wang, L. Wittenburg, and Z. Xing, "Distributed problem solving in sensor networks," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*. ACM Press, 2002, pp. 988–989.

[9] A. Wang and A. Chandrakasan, "Energy efficient system partitioning for distributed wireless sensor networks," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2001)*, Salt Lake City, Utah, May 7–11 2001.

[10] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51 –60, March 2002.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., 1989.

[12] M. Martina, G. Masera, G. Piccinini, and M. Zamboni, "A VLSI architecture for IWT (integer wavelet transform)," in *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, vol. 3, 2000, pp. 1174 – 1177.

[13] J. Ritter and P. Molitor, "A partitioned wavelet-based approach for image compression using FPGA's," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, 2000, pp. 547 – 550.

[14] K. Haapala, P. Kolinummi, T. Hamalainen, and J. Saarinen, "Parallel DSP implementation of wavelet transform in image compression," in *Proceedings of The 2000 IEEE International Symposium on Circuits and Systems*, vol. 5, Geneva, 2000, pp. 89 – 92.

[15] D. Chaver, M. Prieto, L. Pinuel, and F. Tirado, "Parallel wavelet transform for large scale image processing," in *Proceedings of International Parallel and Distributed Processing Symposium, IPDPS*, 2002, pp. 4–9.

[16] P. Meerwald, R. Norcen, and A. Uhl, "Parallel JPEG2000 image coding on multiprocessors," in *Proceedings of the International Parallel and Distributed Processing Symposium 2002*, 2002, pp. 2–7.

[17] C. F. Chiasserini and R. R. Rao, "On the concept of distributed digital signal processing in wireless sensor networks," in *Proceedings of MILCOM*, vol. 1, 2002, pp. 260–264.

[18] D. Devaguptapu and B. Krishnamachari, "Applications of localized image processing techniques in wireless sensor networks," in *SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, (Aerosense '03)*, Orlando, Florida, April 2003.

[19] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, "An evaluation of multi-resolution storage for sensor networks," in *Proceedings of the first international conference on Embedded networked sensor systems*. ACM Press, 2003, pp. 89–102.

[20] S. D. Servetto, "Sensing lena—massively distributed compression of sensor images," in *Proceedings of the IEEE International Conference on Image Processing (ICIP) (special session on "Distributed Source Coding")*, Barcelona, Spain, September 2003.

[21] E. Woodrow and W. Heinzelman, "SPIN-IT: A data centric routing protocol for image retrieval in wireless networks," in *International Conference on Image Processing (ICIP '02)*, September 2002.

[22] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, "Topology control for wireless sensor networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM Press, 2003, pp. 286–299.

[23] S. Bandyopadhyay and E. J. Coyle, "An energy-efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM*, 2003.

[24] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MOBICOM-01)*. ACM Press, July 16–21 2001, pp. 272–287.

[25] T. S. Rappaport, *Wireless Communications: Principles and Practice*. IEEE Press, 1996.

[26] A. Sinha and A. Chandrakasan, "JouleTrack - a web based tool for software energy profiling," in *Design Automation Conference*, 2001, pp. 220–225. [Online]. Available: http://www-mtl.mit.edu/research/anantha/jouletrack/JouleTrack/index.htm%l

[27] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[28] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[29] C. A. Christopoulos, T. Ebrahimi, and A. N. Skodras, "JPEG2000: the new still picture compression standard," in *Proceedings of the 2000 ACM workshops on Multimedia*. Los Angeles, California, United States: ACM Press, 2000, pp. 45–49.

[30] Y. Chan, *Wavelet Basics*. Norwell, MA: Kluwer Academic Publishers, 1995.

[31] D.-G. Lee and S. Dey, "Adaptive and energy efficient wavelet image compression for mobile multimedia data services," in *IEEE Intl. Conference on Communications (ICC)*, New York, 2002.

[32] J. K. Martin Vetterli, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[33] A. Ortega, "Optimal rate allocation under multiple rate constraints," in *Proceedings of Data Compression Conference (DCC)*, Snowbird, UT, March 1996.

[34] J. Storer, *Data Compression*. Rockville, MD: Computer Science Press, 1988.

[35] F. Marino, V. Piuri, and E. J. Swartzlander, "A parallel implementation of the 2-D discrete wavelet transform without interprocessor communications," *IEEE Transactions on Signal Processing*, vol. 47, no. 11, pp. 3179 – 3184, November 1999.

[36] A. Nosratinia, "Post processing of JPEG-2000 images to remove compression artifacts," *IEEE Signal Processing Letters*, vol. 10, no. 10, pp. 296–299, October 2003.

[37] D. S. Taubman and M. W. Marcellin, *JPEG2000, Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[38] D. S. Taubman, "High performance scalable image compression with ebcot," *IEEE Trans. Image Processing*, vol. 9, pp. 1158–1170, July 2000.

[39] K. C. Barr and K. Asanovic, "Energy aware lossless data compression," in *The First International Conference on Mobile Systems, Applications, and Services*, San Francisco, CA, May 2003.

[40] M. D. Adams, "The JasPer project home page," 2003. [Online]. Available: http://www.ece.uvic.ca/~mdadams/jasper/

[41] A. C. Wendi Rabiner Heinzelman and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the Hawaii International Conference on System Sciences (HICSS)*, Jan. 4–7 2000.

[42] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[43] "ns-Network Simulator," http://www.isi.edu/nsnam/ns/, 1998.