A Spatial Correlation-Based Image Compression Framework for Wireless Multimedia Sensor Networks

Pu Wang, *Student Member, IEEE*, Rui Dai, *Student Member, IEEE*, and Ian F. Akyildiz, *Fellow, IEEE*

***Abstract*—Data redundancy caused by correlation has motivated the application of collaborative multimedia in-network processing for data filtering and compression in wireless multimedia sensor networks (WMSNs). This paper proposes an information theoretic image compression framework with an objective to maximize the overall compression of the visual information gathered in a WMSN. The novelty of this framework relies on its independence of specific image types and coding algorithms, thereby providing a generic mechanism for image compression under different coding solutions. The proposed framework consists of two components. First, an entropy-based divergence measure (EDM) scheme is proposed to predict the compression efficiency of performing joint coding on the images collected by spatially correlated cameras. The EDM only takes camera settings as inputs without requiring statistics of real images. Utilizing the predicted results from EDM, a distributed multi-cluster coding protocol (DMCP) is then proposed to construct a compression-oriented coding hierarchy. The DMCP aims to partition the entire network into a set of coding clusters such that the global coding gain is maximized. Moreover, in order to enhance decoding reliability at data sink, the DMCP also guarantees that each sensor camera is covered by at least two different coding clusters. Experiments on H.264 standards show that the proposed EDM can effectively predict the joint coding efficiency from multiple sources. Further simulations demonstrate that the proposed compression framework can reduce 10%–23% total coding rate compared with the individual coding scheme, i.e., each camera sensor compresses its own image independently.**

***Index Terms*—Clustered coding, image compression, spatial correlation, wireless multimedia sensor networks.**

I. Introduction

WIRELESS multimedia sensor network (WMSN) is an emerging networking paradigm that allows retrieving video streams, still images, and generic sensing data from the environment [4]. A WMSN promises a wide range of potential applications such as multimedia surveillance, advanced health care delivery, and industrial process control [4]. Different from the conventional wireless sensor networks that deal with scalar data, WMSNs are required to deliver multimedia content with a certain level of quality-of-service (QoS). This characteristic necessitates more sophisticated data compression strategies for reducing the spectrum demand and saving the energy consumption of the sensor nodes [4].

In a WMSN, a number of camera sensor nodes are deployed in a field of interest with one or more data sinks located either at the center or out of the field. The camera sensor nodes observe the phenomenon at different locations in the field and send their observations to the sink(s). In general, the observations at a camera are directly related to the camera's field of view (FoV) [7], and the spatially proximal cameras could have highly overlapped FoVs. As a result, the visual information retrieved from adjacent camera nodes usually exhibits high levels of correlation, which gives rise to considerable data redundancy in the network.

Multimedia source coding [13], [18] is a common approach to remove the redundancy of visual information. However, the resource constraints of the sensor nodes bring new challenges when applying source coding *globally* in the entire network. The conventional video coding standards, such as MPEG/H.26x [18], can achieve high compression performance. However, they require extensive computation at the encoder, which places heavy burden on the resource-constrained sensor nodes. In [19] and [10], energy-efficient image compression is achieved by distributing the workload of compressing an image over several adjacent sensor nodes. Although promising for compressing the images generated by a single node, these solutions do not explore the correlation of the observed images among adjacent sensors. In contrast, distributed source coding, such as Slepian-Wolf coding [15], only requires low-complexity encoding and leaves the intensive computations at the decoder. However, this coding strategy requires each sensor node to have the knowledge of global correlation structure, which would incur significant additional costs. For these reasons, multimedia source coding is infeasible to be applied *globally* in a large-scale network, despite its outstanding compression gains.

In such a case, the clustered coding strategy provides an effective way to resolve the above dilemma. This strategy uses the hierarchical concept where the entire network is divided into regions. Each region corresponds to a coding cluster, in which a group of camera sensors collaboratively perform data compression, according to different coding algorithms. In the case of conventional coding standards, a powerful cluster head, such as GARCIA robotic platform [4], can be placed within each cluster to serve as a single encoder, which has all correlated multimedia streams as inputs, thereby avoiding the computationally intensive operations draining the limited sensor energy store. In contrast to the conventional coding schemes that require central-

Manuscript received May 03, 2010; revised August 24, 2010 and November 18, 2010; accepted November 30, 2010. Date of publication December 17, 2010; date of current version March 18, 2011. This work was supported by the U.S. National Science Foundation (NSF) under Grant No. ECCS-0701559. The preliminary version was presented at the 29th IEEE International Conference on Computer Communications (INFOCOM 2010) . The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Francesco G. B. De Natale.

The authors are with the Broadband Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: pwang40@ece.gatech.edu; aprildai@ece.gatech.edu; ian@ece.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2010.2100374

ized realization, distributed source coding allows each sensor to encode its own data separately, assuming *a priori* knowledge of local correlation structure in its own cluster [15]. Since each cluster only covers a limited number of nodes, it is feasible to acquire this correlation information without incurring much extra cost. Therefore, the clustered coding strategy paves the way for the practical application of multimedia source coding in large-scale WMSNs.

Despite the promising perspective of clustered coding strategy, there are still many technical issues remaining to be resolved to make this technique of practical application for WMSNs. One of the major issues is how to obtain the correlation characteristics among the images captured by different sensors. Many existing solutions apply image processing methods to estimate the correlation among images from neighboring sensors, and based on the estimated results, collaborative image coding algorithms are proposed. For example, in [16], a distributed image compression method is proposed based on image registration using correspondence analysis. In [20], a shape matching method is applied to exploit the spatial correlation between images acquired from neighboring sensors. However, image processing methods are generally application dependent: different types of images will require different processing schemes [9]. Thus, how to design solutions whose applicability and flexibility would not be limited by the specific applications is of paramount importance.

To solve the problems above, we propose an information theoretic data compression framework that maximizes the overall compression of the visual information retrieved from a WMSN. This framework consists of two components: 1) compression efficiency prediction and 2) coding hierarchy construction. Both components are independent of the specific coding algorithms and images types, thus providing a generic architecture that allows users to freely customize the WMSN applications based on them. The compression efficiency prediction aims to estimate the compression gain from joint encoding of multiple cameras before the actual images are captured. To achieve this, an entropy-based divergence measure (EDM) scheme is proposed, which only takes the camera settings as inputs without requiring the statistics of real images. In the EDM, the overlapping pattern of the FoVs of multiple cameras is first identified. Then, the correlation degree among the observations from cameras with overlapped FoVs is obtained through a spatial correlation model. Based on the correlation characteristics, a dependency graph-based algorithm is designed to estimate the joint entropy of multiple cameras. This joint entropy effectively predicts the compression performance for joint encoding of multiple cameras.

Using the results from EDM, the next problem is how to establish a compression-oriented coding hierarchy, which can achieve a substantial compression gain and decoding reliability. This problem can be further formulated as an optimal coding clustering (OCC) problem, which we define as: find a set of coding clusters with the minimum total entropy, such that each camera node is covered by at least two different clusters. The minimization of total entropy guarantees that the global compression gain is maximized, while the coverage requirement ensures that the impact of cluster failures on the decoding reliability is mitigated. We prove that the OCC problem is NP-hard.

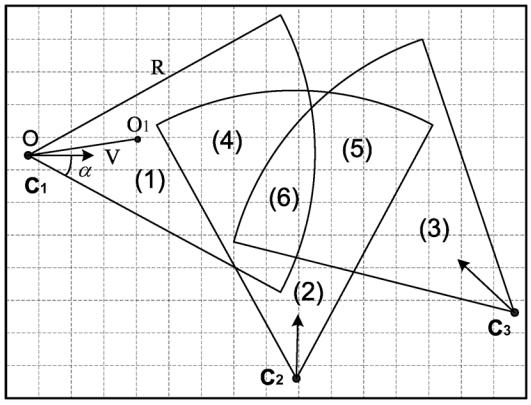

Fig. 1. Field of views of multiple cameras.

As a heuristic solution, a fully distributed protocol, called distributed multi-cluster coding protocol (DMCP), is presented to provide a $\ln \Delta$ approximation to the optimal solution, where Δ is the maximum node degree in the network. Moreover, it is shown that $\ln \Delta$ is the best achievable approximation ratio for the OCC problem.

The rest of this paper is organized as follows. Section II mathematically formulates the problems in the proposed data compression framework. In Section III, the EDM algorithm is introduced to provide a valid assessment of joint coding performance of multiple cameras. The DMCP for establishing the efficient and robust coding hierarchy is proposed in Section IV. The performance of this framework is examined in Section V. Finally, Section VI concludes this paper.

II. Problem Formulation

A. Spatial Correlation of Visual Information

In a WMSN, multiple camera sensors are deployed to provide multiple views, multiple resolutions, and enhanced observations of the environment. As shown in Fig. 1, multiple cameras are deployed in a field of interest, and the cameras' field of views (FoVs) overlap with each other. A camera can only observe the objects within its FoV. The sensing process of a camera is characterized by projection from a 3-D scene to a 2-D image. The observed images from cameras with overlapped FoVs are correlated with each other. We define the correlation of observed images caused by overlapped FoVs as *spatial correlation* in our context. The spatial correlation of the observed images further leads to data redundancy in WMSNs.

For two arbitrary camera sensors C_j and C_k with FoVs $\mathcal{A}_j$ and $\mathcal{A}_k$, suppose at a same time, their observed images are X_j and X_k, respectively. X_j and X_k are correlated when $\mathcal{A}_i$ and $\mathcal{A}_j$ overlap with each other. We introduce a spatial correlation coefficient $\rho_{j,k}$ to quantify the degree of correlation between X_j and X_k. This coefficient will be used as an important parameter in the following problems.

B. Clustered Source Coding

To remove the redundancy for correlated cameras, a group of camera sensors can form a cluster to collaboratively compress their data. Consider a cluster consisting of a cluster head (CH)

and N camera sensors, where each sensor i produces image X_i, which is encoded with rate R_i. According to basic coding theorems, we have the following observation.

Observation 1: The total coding rate of all nodes within a cluster is lower bounded by the joint entropy $H(X_1, X_2, \ldots, X_N)$ no matter if centralized or distributed source coding is applied.

For centralized source coding, each member in a cluster sends its raw or preprocessed data to the CH, while the CH acts as a single encoder that takes all collected data as inputs. According to Shannon's source coding theorem [6], each cluster can generate a total rate lower-bounded by the joint entropy $H(X_1, X_2, \ldots, X_N)$, e.g.,

$$\sum_{i=1}^{N} R_i \geq H(X_1, X_2, \cdots, X_N) \tag{1}$$

where the equality holds when an optimal encoder is used.

For distributed source coding (DSC), each node in a cluster separately encodes its own data, and the CH only acts as a relay node to forward the received data to data sink, where the compressed data are jointly decoded. In this case, Slepian-Wolf coding theorem [15] provides a conceptual basis for DSC and establishes the rate region for the rate vector $(R_1, R_2, \ldots R_N)$:

$$\sum_{i \in U} R_i \geq H\left(X(U) | X(U^c)\right) \quad \forall U \subseteq \{1, 2, \cdots, N\} \tag{2}$$

where $X(U) = \{X_j | j \in U\}$ and U^c is the complementary set of U.

Surprisingly, Slepian-Wolf coding theorem (2) indicates that the sum of rates, $\sum_{i=1}^{N} R_i$, can achieve the joint entropy $H(X_1, X_2, \cdots, X_N)$, just as for joint encoding the sources $(X_1, X_2, \cdots, X_N)$, despite separate encoders for them. Therefore, a cluster with N nodes can be optimally encoded with $H(X_1, X_2, \cdots, X_N)$ bits no matter if centralized or distributed source coding is applied.

C. Multi-Camera Entropy Estimation Problem

Joint entropy serves as a lower bound of the overall coding rate of multiple sources for both centralized and distributed source coding. If the joint entropy for a cluster of cameras can be estimated, we will be able to predict the performance of joint coding within the cluster. However, to estimate the joint entropy of visual information from multiple cameras is a challenging task. Because of the intrinsic complexity of visual information, it is difficult to model the dependency characteristics of visual sources, and moreover, it usually requires expensive computation and communication costs.

Our objective is to estimate the joint entropy of multiple cameras in WMSNs through low computation and communication costs. Given a cluster of cameras with observations $X_1, X_2, \cdots, X_N$, the joint entropy $H(X_1, X_2, \cdots, X_N)$ will be described as a function of the individual entropy $(H(X_i))$ and field of view $(\mathcal{A}_i)$ of each camera, and the correlation coefficients between any two cameras $(\rho_{j,k})$ in the cluster.

D. Optimal Coding Clustering Problem

Since joint entropy provides a benchmark on the compression gain from joint encoding of multiple sources, we can utilize a similar entropy-based concept, called *cluster entropy*, to measure the collaborative compression gain within the scope of a single *coding cluster*. The target of optimal coding clustering can then be correspondingly interpreted as to select a set of *coding clusters* according to their *cluster entropies* such that total entropy of the entire network is minimized. We describe two definitions involved in the discussion above.

Definition 1: A *coding cluster* is a finite set comprising a camera sensor and all sensors within its transmission range.

Definition 2: For each coding cluster A, its *cluster entropy* $H(A)$ is equal to the joint entropy of all cameras in A.

Now, the OCC problem can be formally stated as: given a network consisting of a finite set of camera sensors $E = e_1, e_2, \ldots e_n$ and a set of n subsets of E, $\mathcal{S} = \{S_1, S_2, \ldots S_n\}$, where each set S_i corresponds a coding cluster with its entropy $H(S_i)$, the goal is to find a collection C from $\mathcal{S}$ of minimum total entropy $\sum_{S_i \in C} H(S_i)$, such that each element e_i is covered by at least two sets in C.

The minimization of total entropy guarantees that the maximum global compression gain is achieved, while the coverage requirement ensures that the visual information encoded by each camera has more chance to be successfully delivered to, and properly decoded at data sink.

III. Joint Entropy Estimation

In this section, we propose a novel EDM scheme to estimate the joint entropy of observations from multiple cameras. This scheme only takes cameras' settings as inputs without requiring the knowledge of specific applications, thereby providing a generic framework for prior evaluation of compression under different coding solutions. Moreover, it induces little communication costs since camera nodes only need to exchange their settings via short messages among their 1-hop neighbors, and low complexity computations are required for joint entropy estimation. The EDM scheme consists of the following two components.

1) *Area division for FoVs*. Given a group of cameras, their FoVs are divided into several regions, such that each region is covered by the same set of cameras.
2) *Joint entropy estimation for regions*. For each region, a dependency graph is constructed based on the correlation among the cameras. The joint entropy of the region is then estimated by traversing the dependency graph. Finally, the total joint entropy for the group of cameras is the sum of the entropies of all the regions.

A. Area Division for Overlapped Field of Views

A camera is a directional sensor with limited sensing range. It can only observe the objects within its FoV. If a camera sensor is deployed on a ground plane, we can use a simplified 2-D FoV model [11] that models the shape of a camera's FoV as a sector. As shown in the left part of Fig. 1, a camera's FoV is determined by four parameters: O, R, $\vec{V}$, and α, where O is the location of the center of the camera, R is the sensing radius, $\vec{V}$ is the sensing

direction (the center line of sight of the camera's FoV), and α is the offset angle between the sensing direction and a radius of the sector. An arbitrary point O_1 is in the FoV of the camera if it satisfies

$$\begin{cases} |O\vec{O}_1| \leq R \\ \theta \leq \alpha, \end{cases} \tag{3}$$

where θ is the angle between $O\vec{O}_1$ and $\vec{V}$.

Another key parameter for a camera's sensing process (3-D to 2-D projection) is the camera's focal length (f). Both the FoV parameters and the focal length could be estimated through calibration methods for WMSNs, e.g., [8].

We consider the case when N cameras $(C_1, C_2, \cdots, C_N)$ are deployed on the ground plane and all the cameras are homogeneous, i.e., they have the same focal lengths (f), sensing radii (R), and offset angles (α). Denote the FoV of an individual camera C_i by $\mathcal{A}_i(O_i, R_i, \vec{V}_i, \alpha_i)$, and the overall FoV for these cameras by $\mathcal{A}$ $(\mathcal{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_N\})$. The goal of area division is to divide $\mathcal{A}$ into several regions $(P_1, P_2, \cdots, P_M)$, such that each region belongs to the FoVs of the same set of cameras. As shown in Fig. 1, the FoVs of the three cameras are divided into six different regions.

We introduce a grid-based approach to divide the overall FoV $\mathcal{A}$ into regions. As shown in Fig. 1, the overall FoV $\mathcal{A}$ is firstly divided into small grids $(G(k),\ k = 1, \cdots, K)$. Then we can check if a grid $G(k)$ is in a camera's FoV $(\mathcal{A}_n,\ n = 1, \cdots, N)$ as follows: we find the center point of the grid, and using the condition in (3), we can tell if it is in the camera's FoV; if this center point is in the camera's FoV, we regard that this grid is in the camera's FoV. (This approximation is valid as long as the size of the grid is much smaller than the size of the FoV.) After traversing all the grids in the overall FoV, regions could be formed by grouping the grids that belong to the same set of cameras.

B. Estimating the Joint Entropy of a Region

In this section, we introduce an algorithm to estimate the joint entropy of a region. Denote the cameras that can observe region P_i by $(C_1, \cdots, C_n)$. For the kth camera C_k, denote its observed visual information by X_k, and denote its observation about this region by $X_k(P_i)$. The amount of information of the region P_i is the joint entropy of the observations about this region from the cameras $(C_1, \cdots, C_n)$, given by

$$H(P_i) = H\left(X_1(P_i), \cdots, X_n(P_i)\right). \tag{4}$$

Since there is no unified probability model for images and estimating the joint probability distribution of multiple sources requires large bulk of computation, it is difficult to calculate joint entropy in resource-constrained WMSNs. (See [7] for details.) In this paper, we introduce a novel approach to estimate joint entropy based on the spatial correlation model [7] in our previous work. Our solution consists of three steps:

1) Estimate the individual entropy $H(X_k(P_i))$ in (4);
2) Study the correlation characteristics among the individual observations using the correlation model in [7];
3) With the results from 1) and 2), apply a dependency graph based algorithm to estimate $H(P_i)$ in (4).

We explain these steps in details in the following paragraphs.

1) Individual Entropy Estimation: For an arbitrary camera C_k, the entropy of its observed image X_k is $H(X_k)$. The entropy of the observation X_k about the region P_i, $H(X_k(P_i))$, can be estimated as

$$H\left(X_k(P_i)\right) \approx \frac{S(P_i)}{S(\mathcal{A}_k)} H(X_k) \tag{5}$$

where $S(P_i)$ is the area of P_i and $S(\mathcal{A}_k)$ is the area of the FoV. The entropy $H(X_k)$ is the total amount of information of X_k, which is provided by the projections of all the 3-D points in the FoV. As there is no prior knowledge about where a camera is deployed or what type of scene is observed, it is assumed that when all the 3-D points in the FoV are projected on the camera's image plane, each point provides the same amount of information. Considering that the cameras are deployed on a ground plane and a 2-D FoV model is used, the amount of information that camera C_k contributes to P_i is approximately proportional to the area of P_i, so we use the ratio $S(P_i)/S(\mathcal{A}_k)$ to estimate $H(X_k(P_i))$ in (5).

This assumption works well when there are no large obstacles in a camera's FoV. A camera's FoV might be reduced in case of obstacles; therefore, when implementing the proposed algorithm in practical applications, we might need to update the camera's FoV model to reflect the effect of obstacles.

There are many different models for images, and different values of entropy may be obtained for the same image sources. In our algorithm, we avoid calculating the exact values of entropy of images. As we consider the case that all the camera sensors in a WMSN are homogeneous, without loss of generality, we assume that the entropies of the single observed images are the same, denoted by $H(X_i) = H(\cdot)$ $(i = 1, \cdots, N)$. All the joint entropy terms in our algorithm will be expressed as relative values of $H(\cdot)$.

2) Spatial Correlation Motivated Entropy Estimation: In our earlier work, we proposed a novel spatial correlation model for visual information in WMSNs [7]. Given an area of interest and two cameras that can observe it, a spatial correlation coefficient was derived to quantify the degree of correlation between the two cameras. For example, if we take region P_4 in Fig. 1 as the area of interest, both camera C_1 and camera C_2 can observe it, with observations $X_1(P_4)$ and $X_2(P_4)$. We pick some reference vectors in region P_4, and calculate the projections of these reference vectors in C_1 and C_2 using the projection model of cameras. By studying the correlation between the projected reference vectors on the two cameras, the spatial correlation coefficient $\rho_{1,2}$ can be calculated.

In general, for cameras C_j and C_k that can observe region P_i, with P_i as the area of interest, a spatial correlation coefficient between the observations of P_i at C_j and C_k was derived as a function as follows:

$$\rho_{j,k} = f(O_j, \vec{V_j}, O_k, \vec{V_k}, P_i) \tag{6}$$

where O_j and O_k are the two cameras' locations, and $\vec{V_j}$ and $\vec{V_k}$ are their sensing directions. The spatial correlation coefficient was designed as a normalized symmetric metric, i.e., it satisfies $\rho_{j,k} = \rho_{k,j}$ and $0 \leq \rho_{j,k} \leq 1$.

More importantly, the spatial correlation coefficient was related to the estimation of joint entropy in [7]. We briefly introduce the relevant results here.

To evaluate the dependency between two visual sources, an *entropy correlation coefficient (ECC)* was introduced in [12]. The ECC for two visual sources A and B was defined as

$$ECC = \frac{2I(A;B)}{H(A)+H(B)}. \tag{7}$$

Moreover, the joint entropy $H(A,B)$ can be given as

$$H(A,B) = \left(1 - \frac{1}{2}ECC\right)(H(A)+H(B)). \tag{8}$$

By definition, the joint probability distribution of the two sources is needed to estimate the joint entropy. Due to the complexity of image contents and the difficulty in image modeling, it is difficult to get an accurate estimation of the joint probability distribution [12]. Besides, camera sensors in a WMSN must exchange their observed images to estimate the joint probability distribution, which introduces a lot of communication burden in the network. In [7], it was found that the spatial correlation coefficient (6) had the same intrinsic meaning as ECC: both ranging from 0 to 1 and denoting the degree of correlation between two sources, while the spatial correlation coefficient could be obtained through low computation and communication costs. Therefore, the ECC term in (8) was replaced by the spatial correlation coefficient.

Consequently, for cameras C_j and C_k that can observe region P_i, the joint entropy of the observations of P_i at C_j and C_k was estimated as

$$H\left(X_j(P_i), X_k(P_i)\right) \approx \left(1 - \frac{1}{2}\rho_{j,k}\right) \times \left(H\left(X_j(P_i)\right) + H\left(X_k(P_i)\right)\right) \tag{9}$$

where $X_j(P_i)$ is the observation of P_i at camera C_j, and $X_k(P_i)$ is the observation of P_i at camera C_k. This equation indicates that the amount of information gained from the observations of two cameras depends on the correlation between them. The more the two observations are correlated, the less joint entropy can be gained from them together.

From (9), we can obtain the conditional entropy as follows:

$$\begin{aligned} H\left(X_j(P_i)|X_k(P_i)\right) &= H\left(X_j(P_i), X_k(P_i)\right) - H\left(X_k(P_i)\right) \\ &\approx \left(1 - \frac{\rho_{j,k}}{2}\right) H\left(\left(X_j(P_i)\right)\right) \\ &\quad - \frac{\rho_{j,k}}{2} H\left(X_k(P_i)\right) \end{aligned} \tag{10}$$

where $H(X_j(P_i)|X_k(P_i))$ is the entropy of $X_j(P_i)$ with the knowledge of $X_k(P_i)$.

3) Dependency Graph-Based Joint Entropy Estimation: Based on the correlation coefficient (6) and the conditional entropy term (10), we propose a dependency graph-based algorithm to estimate the joint entropy of a region.

We study a two cameras' case first. Suppose there are only two cameras (C_1 and C_2) in a region P_i. We can depict their relationship using a dependency graph: $C_2 \rightarrow C_1$. The joint entropy of the observations from C_1 and C_2 can be calculated by traversing the dependency graph. The source node C_2 contributes the entropy of its observations, $H(X_2(P_i))$, and the node C_1 contributes the conditional entropy with respect to C_2, $H(X_1(P_i)|X_2(P_i))$, so the joint entropy is calculated by adding these two terms: $H(X_1(P_i), X_2(P_i)) = H(X_2(P_i)) + H(X_1(P_i)|X_2(P_i))$. The dependency graph can also be constructed as $C_1 \rightarrow C_2$, from which we can get the same result of joint entropy.

The two cameras' case can be extended to estimate the joint entropy of more than two cameras. Generally, for an arbitrary number of cameras, we construct a dependency graph to describe the dependency characteristics among them. Denote the dependency graph by $G(V,E)$, where V is a collection of cameras, and E is a collection of directed edges that stand for dependencies. Joint entropy of the region is calculated by traversing all the nodes in the graph along the directed edges. The detailed steps are described in *Algorithm* 1.

For a group of cameras $(C_1, C_2, \cdots, C_n)$ that can observe the region P_i, we can obtain a correlation matrix $(\rho_{j,k})_{n*n}$ based on (6). To simplify the problem, we assume limited number of dependencies: each camera is dependent on the camera that is most correlated with it. For example, if camera C_j is most correlated with camera C_k, we say that C_j is dependent on C_k, and we can construct a directed edge starting from C_k and ending at C_j: $C_k \rightarrow C_j$. C_j is said to be a direct successor of C_k, and C_k is a direct predecessor of C_j.

The dependency graph is designed to be a directed acyclic graph with the following constraints: a camera is either a source node (i.e., a node that has no predecessors) or a direct successor of one of the other cameras; a dependency graph may have several source nodes, but each node can have at most one direct predecessor; and there should be no loops in the graph, e.g., $C_k \rightarrow C_j$ and $C_j \rightarrow C_k$ cannot exit in the same graph. These properties could be guaranteed through the procedure of constructing the dependency graph (lines 5–12 in *Algorithm* 1). For each node C_k, if another node C_j is most correlated with it, i.e., $neighbor(C_j) = k$, the algorithm adds $C_k \rightarrow C_j$ into the graph only when two conditions are met: 1) C_j has no predecessors, and 2) C_j is not a predecessor of C_k (line 7 in *Algorithm* 1). The first condition guarantees that each node can have at most one direct predecessor, and the second one guarantees that there are no loops in the graph.

Given a dependency graph with the above features, the joint entropy is estimated by traversing all the nodes in the graph and adding the entropies of the nodes together, which corresponds to lines 13–19 in *Algorithm* 1. A source node contributes its individual entropy to the joint entropy, while a non-source node contributes its conditional entropy with respect to its direct predecessor to the joint entropy.

Algorithm 1: Dependency Graph-Based Entropy Estimation

1) P_i: $\{C_1, C_2, \ldots, C_n\}$ with correlation matrix $(\rho_{j,k})_{n*n}$.
2) **for** $j = 1$ to n **do**
3) $neighbor(C_j) = arg\max_{k \neq j}(\rho_{j,k})$;
4) ***end for***
5) **for** $k = 1$ to n **do**
6) **for** $j = 1$ to n and $j \neq k$ **do**
7) **if** $neighbor(C_j) =$ k and C_j has no predecessors and C_j is not a predecessor of C_k **then**
8) Add $C_k \rightarrow C_j$ into the dependency graph;
9) $Predecessor(C_j) = C_k$;
10) **end if**
11) **end for**
12) **end for**
13) **for** $j = 1$ to n **do**
14) **if** C_j has no predecessor **then**
15) Add $H(X_j(P_i))$ to $H(P_i)$;
16) **else if** $Predecessor(C_j) = C_k$ **then**
17) Add $H(X_j(P_i)|X_k(P_i))$ (10) to $H(P_i)$;
18) **end if**
19) **end for**
20) **return** $H(P_i)$.

Since the FoVs for a group of cameras are divided into several independent regions, the total joint entropy is the sum of the entropies of all the regions. For a group of cameras with observations $(X_1, \ldots, X_N)$, with their FoVs divided into regions $(P_1, \ldots, P_M)$, the total joint entropy is given by

$$H(X_1, \cdots, X_N) = H(P_1) + \cdots + H(P_M) \qquad (11)$$

where $H(P_i)$ $(i = 1, \cdots, M)$ is obtained by *Algorithm* 1.

To provide an overview of the whole EDM algorithm, we illustrate the steps for estimating the joint entropy of the three cameras in Fig. 1. The FoVs of the three cameras are divided into six regions. We take the sixth region (P_6) as an example. All the three cameras (C_1, C_2, C_3) can observe P_6. Suppose we find from the geometry of the cameras' FoVs that $S(P_6) = 0.1S(\mathcal{A}_i)$. The entropy of a single image is $H(\cdot)$, by applying (5), the individual entropies about this region are $H(X_i(P_6)) = 0.1H(\cdot)$ $(i = 1, 2, 3)$. Furthermore, from (6), we can obtain a correlation matrix for P_6:

$$(\rho_{jk})_{3*3} = \begin{pmatrix} 1 & 0.1 & 0 \\ 0.1 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{pmatrix}.$$

By applying *Algorithm* 1 on the correlation matrix, we can obtain a dependency graph as $C_3 \rightarrow C_2 \rightarrow C_1$. Therefore, the joint entropy of region P_6 is $H(P_6) = H(X_3(P_6)) + H(X_2(P_6)|X_3(P_6)) + H(X_1(P_6)|X_2(P_6))$, where the conditional entropies can be calculated from (10). For example, $H(X_2(P_6)|X_3(P_6)) = (1 - (\rho_{2,3}/2)) \cdot H(X_2(P_6)) - (\rho_{2,3}/2) \cdot H(X_3(P_6)) = 0.05H(\cdot)$. In the same way, we can obtain $H(X_1(P_6)|X_2(P_6)) = 0.09H(\cdot)$. Thus, the joint entropy of P_6 is $H(P_6) = 0.24H(\cdot)$. After the entropy of each region is calculated, the joint entropy of the three cameras is calculated by $H(X_1, X_2, X_3) = H(P_1) + \cdots + H(P_6)$.

The entire EDM algorithm can be run at each sensor node. To estimate joint entropy, a node just need to acquire the FoV parameters, locations, and sensing directions of its neighbors. Therefore, it does not require expensive communication costs in the network. The estimated joint entropy will serve as a criteria for the DMCP protocol in the following section.

IV. Data Compression Using Clustered Source Coding

After a WMSN is deployed in a field, we would like to select a set of coding clusters to cover the entire network with maximum compression ratio. Due to the distributed manner of WMSNs and the changing environment, a centralized algorithm is not suitable for use here. The coding cluster selection should only depend on local information. In this section, we first formulate the OCC problem as an integer program, and shows that the OCC problem is NP hard. Accordingly, we propose a DMCP, which is shown to achieve an approximation guarantee of $\ln \Delta$.

A. Integer Program Formulation of OCC Problem

To formulate the OCC problem as an integer program, we assign a variable x_S for each set $S \in \mathcal{S}$, which is allowed 0/1 values. This variable will be set to 1 iff set S is selected for the coding hierarchy. The objective function is the sum of the entropy values of all selected coding clusters. The constraint is that for each node $e \in E$, we want at least two of the clusters containing it to be selected:

$$\begin{aligned} \text{MIN} \quad & \sum_{S \in \mathcal{S}} H(S) x_S \\ s.t \quad & \sum_{S: e \in S} x_S \geq 2, \quad e \in E \\ & x_S \in \{0, 1\}, \quad S \in \mathcal{S}. \end{aligned} \qquad (12)$$

If we treat $H(S)$ as the cost $c(S)$ associated with each coding cluster $S \in \mathcal{S}$ and let the second constraint be coverage requirement for each node $e \in E$, the OCC problem can be reduced to the constrained set multicover (CSMC) problem. The CSMC problem is NP-hard and the greedy algorithm is essentially the best one can hope for [14]. In other words, the approximation ratio $\ln \Delta$ achieved by the greedy algorithm is the best one for the CSMC problem. Therefore, the greedy strategy applies naturally to our OCC problem: let us say that the node e is uncovered if it occurs in fewer than two of the selected coding clusters. In each iteration, the algorithm selects, from the currently unselected clusters, the most compression-efficient cluster, where the compression efficiency of a cluster is defined to be the average entropy of the uncovered nodes it covers. The

algorithm terminates when there are no more uncovered nodes, e.g., each node has been included by two different clusters. The pseudo-code of the above procedures is described in *Algorithm* 4 in the Appendix.

The greedy algorithm for the OCC problem can be computed in $O(n)$ rounds if a central controller (e.g., data sink) provides the full information of the network topology along with the detailed settings (e.g., sensing direction, sensing offset angle, and sensing range) for each camera. However, in a large-scale distributed network like WMSN, the centralized operations have limited flexibility and scalability. Moreover, the energy constraint of sensor nodes prohibits network-wide information exchange. Next, we will propose a distributed protocol that only needs local information exchange to achieve global compression optimization.

B. Distributed Multi-Cluster Coding Protocol

After a WMSN is initially deployed, each camera node leads its neighbors to constitute a candidate coding cluster. At this time, each sensor node could be in one of the following four states: *black*, *grey*, *half grey*, and *white*. We call sensor nodes *black* if they are selected as the CH locaters. The CH locaters will not serve as the normal CHs but indicate the coordinates at which the future mobile or fixed CHs should be placed. We call the nodes *grey* if they are covered by at least two *black* nodes, and *half grey* if they are covered by exactly 1 *black* node. A node stays in the *white* state if there exists no *black* node within its 1-hop range. The *half grey* nodes and *white* nodes are collectively referred to as *uncovered* nodes. We now describe several useful definitions.

Definition 3: The *neighbor set* of a node is a set consisting of the node itself and all nodes in its 1-hop range.

Definition 4: The *serving set* of a node is a set comprising the *uncovered* nodes that are residing in its 1-hop range.

Definition 5: The *coding effectiveness* of a node is the average entropy of all nodes in its *serving set*.

Definition 6: The *CH counter* of a node records the current number of the *black* nodes among its 1-hop neighbors.

Algorithm 2: Distributed Multi-Cluster Coding Protocol

1) $state(e) \in \{black, grey, half\ grey, white, uncovered\}$

2) $state(e) \leftarrow white$, send & receive $state(e)$ and camera settings

3) $N_e \leftarrow \{e' : state(e') = white\} \cup \{e\}$

4) {Discover *neighbor set* N_e}

5) $counter(e) = 0$ {Set CH counter}

6) **while** $state(e) = uncovered$ **do**

7) $U_e \leftarrow \{e' \in N_e : state(e') = uncovered\}$

8) {Calculate *serving set* U_e}

9) $EC_e \leftarrow H(N_e)/|U_e|$

10) {Calculate *coding effectiveness* EC_e

11) **if** $EC_e = \min_{e' \in U_e}\{EC_{e'}\}$ **then**

12) $state(e) \leftarrow black$, and $counter = 1$

13) send $COVERAGE$ msg

14) **else**

15) wait until the selection of a new $black$ node times out

16) **if** no $COVERAGE$ received **then**

17) $state(e)$ remains

18) **else if** $counter = 0$ **then**

19) $state(e) \leftarrow half\ grey$, and $counter(e) = 1$

20) **else if** $counter = 1$ **then**

21) $state(e) \leftarrow grey$, and $counter(e) = 2$

22) send & receive ADV msg containing $state(e)$

23) **end if**

24) **end if**

25) **end while**

26) $Process\_Grey\_Black()$

Now, the proposed DMCP establishes a clustered coding hierarchy as follows. Initially (lines 1–5 in *Algorithm* 2), no *black* nodes exist in the network. Thus, every node is *uncovered*. Nodes in the *uncovered* state send out their camera settings to their neighboring nodes. After receiving the setting information, an *uncovered* node discovers its *serving set* and calculates its *cluster entropy*. Based on these information, an *uncovered* node evaluates its *coding effectiveness*, which is sent out along with the node state in an advertising (ADV) message to its 2-hop neighbors.

A node in the *uncovered* (e.g., *half grey* or *white*) state collects ADV messages and extracts the *coding effectiveness* values from its 2-hop neighbors. If the node itself is the most coding-effective node amongst its 2-hop neighbors, it becomes a *black* node and sends COVERAGE messages to other *uncovered* nodes within its 1-hop range (lines 11–13 in *Algorithm* 2). Otherwise, an *uncovered* node can encounter the following scenarios: 1) if no COVERAGE message is received within the predefined maximum duration of selecting a new *black* node, the node remains *uncovered*, recalculates its *coding effectiveness*, and sends out an ADV message (lines 16–17 in *Algorithm* 2). 2) If a COVERAGE message is received, and its *CH counter* is equal to zero, the node enters *half grey* state and increments its *CH counter* by 1 (lines 18—19 in *Algorithm* 2). 3) If a COVERAGE message is received, and its *CH counter* already reaches 1, the node becomes a *grey* node and sets the *CH counter* to 2 (lines 20–21 in *Algorithm* 2). For the last two cases, an ADV message containing the node state is sent out to its immediate neighbors.

For a *grey* node, if the CH counters of all its neighbors already reach 2, the node remains *grey* for the rest of cluster selection procedure and becomes a cluster member in the end. Otherwise,

the node sends out an ADV message containing its *coding effectiveness* and collects ADV messages from all the *uncovered* nodes within its 2-hop range. If the node itself has the highest *coding effectiveness*, it enters *black* state and sends out COVERAGE messages to its *uncovered* neighbors (lines 5–6 in *Algorithm* 3). Otherwise, if the maximum duration of generating a new *black* node passes, and there still exist *uncovered* nodes within its 1-hop range, the node remains *grey* (lines 8–10 in *Algorithm* 3). A *black* finally becomes a CH locater until the value of its CH counter reaches 2 on receiving a COVERAGE message (lines 13–17 in *Algorithm* 3).

Algorithm 3: $Process\_Grey\_Black()$

1) $U_e \leftarrow \{e \in N_e : counter(e) < 2\}$
2) **while** $|U_e| < |N_e|$ **do**
3) **if** $state(e) = grey$ **then**
4) recalculate *coding effectiveness* EC_e
5) **if** $EC_e = \min_{e' \in U_e}\{EC_{e'}\}$
6) $state(e) \leftarrow black$, and send $COVERAGE$ msg
7) **else**
8) wait until the new *black* selection times out
9) **if** $|U_e| < |N_e|$ **do**
10) $state(e) \leftarrow grey$
11) **end if**
12) **end if**
13) **else if** $state(e) = black$ **then**
14) wait until a $COVERAGE$ is received
15) $counter(e) = 2$
16) send ADV msg containing $state(e)$ and $counter(e)$
17) Node e becomes a CH locator
18) **end if**
19) **end while**
20) Node e becomes a cluster member

The above procedures are performed by all nodes until each of them becomes either a CH locater or a cluster member. At the end, there is no *uncovered* node in the network, and the established clustered coding hierarchy covers the entire network. The pseudo-code of the above procedures is described in *Algorithm* 2 and *Algorithm* 3.

C. Correctness and Complexity

Theorem 1: If the minimum node degree in a WSN is 2, each node will be covered by at least two coding clusters when DMCP terminates.

Proof: Assume when DMCP terminates, a node v does not belong to any coding cluster or it is only covered by one coding cluster. This implies that v stays in the *uncovered* state. Thus, the condition in line 6 of *Algorithm* 2 is satisfied. Since the minimum node degree of v is larger than 1, v has at least one neighbor, say u. If u has higher coding efficiency than v, then u becomes a *black* node and v is covered by the coding cluster led by u. Otherwise, v enters *black* state, and thus v is covered by the coding cluster led by itself. Both cases contradict the assumption that v does not belong to any coding cluster. On the other hand, if v is only covered by one cluster, this means that either v itself or one of its neighbors is a *black* node. In this case, v's neighbor or v will become a CH. (i.e., the operations in lines 12–14 of *Algorithm* 2 or lines 5–6 of *Algorithm* 3 are executed). This implies that v is covered by two clusters, thus contradicting the assumption that v is only covered by one coding cluster. ■

Theorem 2: The DMCP protocol has a worst-case processing time complexity of $O(N^2)$ per node per round, where N is the number of nodes in the network.

Proof: In *Algorithms* 2 and 3, the computational operations include two parts: the estimation of the cluster entropy (line 9 of *Algorithm* 2 and line 4 of *Algorithm* 3) and the search of the minimum average entropy (line 12 of *Algorithm* 2 and line 5 of *Algorithm* 3). The first part is calculated by the EDM scheme presented in Section III. As indicated in *Algorithm* 1, the EDM has a time complexity of $O(N^2)$. The second part is realized by binary tree sorting, which takes $O(Nlog(N))$ iterations. Thus, in each round, DMCP protocol has a worst-case processing time complexity of $O(N^2)$ per node. ■

Theorem 3: The DMCP terminates in $O(N)$ rounds, where N is the number of nodes in the network.

Proof: Given a network with N sensor nodes, there exist total N candidate coding clusters, each of which consists of a sensor node and its neighboring nodes. As indicated in *Algorithm* 2, in each round, at least one candidate coding cluster is selected as the final coding cluster. Thus, the DMCP takes a time in $O(N)$ rounds in the worst case. ■

Theorem 4: The DMCP protocol has a worst-case message exchange complexity of $O(1)$ per node.

Proof: During the execution of *Algorithm* 2, an $uncovered$ ($white$ or $halfgrey$) node is silent until it sends notification messages $COVERAGE$ to become $black$ node or sends the joint messages ADV to become $grey$ node. The number of these $COVERAGE$ messages are strictly less than N, since at most N nodes will enter $black$ state. In addition, $uncovered$ nodes generate at most N ADV messages, since at least one node will decide to be a CH. Besides $uncovered$ nodes, during the execution of *Algorithm* 3, $black$ nodes and $grey$ nodes also send out ADV and $COVERAGE$ messages. Specifically, $black$ nodes broadcast at most N ADV messages to advertise their final status, and $grey$ nodes generate at most N $COVERAGE$ messages to announce a status change from the $grey$ to the $black$. Hence, the number of messages exchanged in the network is upper-bounded by $4N$, i.e., $O(N)$. ■

Since the clustered coding hierarchy only needs to be constructed when the network is initially deployed in the field of interest, thus, the linear time and message complexity of the proposed protocol has trivial impact on the network performance, compared with the significantly enhanced energy efficiency induced by the established coding hierarchy. In addition, to reduce

the computational delay, the protocol parameters, such as the duration of each round, can be the properly adjusted. The proposed framework can be implemented on a variety of camera sensor platforms, which are equipped with a wide range of microprocessors including ARM7, AVR, and Atmel ATmega128L [5]. Generally, users are provided with the dedicated compilers for the specific camera sensor platforms. To facilitate the cross compiling on different hardware platforms, we believe that the gcc compiler collection [1] can be a good choice because it is available for most embedded platforms equipped with a variety of microcontrollers.

D. Approximation Ratio

Theorem 5: The DMCP computes a $\ln \Delta$ approximation for the optimal coding clustering problem.

Proof: According to DMCP, the *cluster entropy* of a node is only related to camera settings of the nodes in its *neighbor set*, and the *neighbor set* is only determined by the local topology; the value of the *cluster entropy* will not change as the protocol proceeds. On the other hand, the cardinality of the *serving set*, which is equal to the number of its *uncovered* neighboring nodes, can be reduced as protocol proceeds since some *uncovered* neighboring nodes could be included by some other clusters. Thus, we conclude that the *coding effectiveness* of a non-*black* node can only be reduced if the cardinality of its serving set decreases.

Based on this conclusion, we can further show that the DMCP is equivalent to the centralized greedy algorithm. According to DMCP, a non-*black* node v with the highest *coding effectiveness* within its 2-hop neighborhood is eligible to become a *black* node. The selection of other non-*black* nodes outside v's 2-hop range as *black* nodes will not affect v's eligibility to enter the *black* state because the status change of the nodes outside v's 2-hop range cannot reduce v's *serving set* cardinality, and according to the conclusion above, v's *coding effectiveness* remains the same. Therefore, the DMCP chooses v as a *black* node before any nodes within its 2-hop range. On the other hand, the centralized greedy algorithm always selects the most *compression efficient* cluster, and v leading its neighbors represents the most *compression efficient* cluster within its 2-hop range. Therefore, the centralized approach will select the cluster led by v as a final coding cluster as the algorithm proceeds. This means that the DMCP obtains the same result as the centralized algorithm, thus achieving the same $\ln \Delta$ approximation ratio as the centralized algorithm. ■

As shown in Section IV-A, the OCC problem can be reduced to CSMC problem, for which $\ln \Delta$ is the best approximation ratio. Thus, we can conclude that no protocols perform better than the proposed DMCP in terms of application factor.

E. Intercluster Connectivity

After the DMCP is performed, the selected CH locaters send out messages to advertise their states and coordinates. Then, some more powerful multimedia nodes, such as the GARCIA robotic platform [4], can automatically move to or be manually placed at these locations, and act as normal CHs. Since the CHs are interconnected by multi-hop connections, the CHs should properly adjust their transmission range to maintain inter-cluster connectivity. We address this problem by proving the following theorem.

Theorem 6: In a WMSN with the minimum node degree $\delta \geq 1$, i.e., there is no isolated node in the network, any two CHs are two hops away at most.

Proof: Since every node has at least one neighbor, each cluster member belongs to at least two clusters after the DMCP is performed. This means that each cluster member has two different CHs within its 1-hop range. Suppose a CH v can reach the nearest CH w at least three hops away. Then, there exists a cluster member u of the CH v in the path between v and w. This implies that the CH w is at least 2-hops away from the cluster member u. Therefore, u cannot be covered by the CH w, and there has to be another CH, say x, within 1-hop range of u to meet its coverage requirement. Now, the cluster member u has two CHs, v and x, within its immediate neighborhood. This means that v and x are at most two hops away from each other, which contradicts with the assumption that a CH v can reach the nearest CH at least three hops away. ■

Therefore, in order to maintain inter-cluster connectivity, each CH only needs to adjust its transmission range to twice the 1-hop distance.

V. Performance Evaluation

We evaluate the performance of the proposed data compression framework through simulations. We first investigate the effectiveness of the EDM scheme by comparing its predicted results with the actual joint coding performance of practical coding schemes. Then, we study the compression performance of DMCP under changing network sizes and camera settings.

A. Validity of the EDM Predictions

For a cluster of N camera sensors with observations $X_1, \cdots, X_N$, the joint entropy $H(X_1, \cdots, X_N)$ is a theoretical lower bound of the total coding rate for these cameras. To predict the percentage of rate savings of joint coding, we define an *estimated joint coding efficiency* as

$$\eta_H = 1 - \frac{H(X_1, \cdots, X_N)}{H(X_1) + \cdots + H(X_N)} \tag{13}$$

where $H(X_1) + \cdots + H(X_N)$ corresponds to the total coding rate needed when the cameras compress their observations individually.

We verify the estimated joint coding efficiency using practical video coding experiments. Similar as the definition above, we introduce an *actual joint coding efficiency* as

$$\eta_R = 1 - \frac{R(X_1, \cdots, X_N)}{R(X_1) + \cdots + R(X_N)} \tag{14}$$

where $R(X_1, \cdots, X_N)$ is the total rate from joint coding, and $R(X_1) + \cdots + R(X_N)$ is the total rate from individual coding.

We consider an indoor scene and an outdoor scene as representatives of various WMSN applications. We deploy 12 camera nodes at different view points around each scene, and let each

(a) (b)

Fig. 2. (a) Indoor scene "Tables". (b) Outdoor scene "Trees".

TABLE I
EXPERIMENTAL PARAMETERS

	H.264 Baseline	H.264 MVC
RD optimization	on	on
Entropy coding	UVLC	CABAC
Search range	128 pixels	128 pixels
Num of reference frames	1	1

camera capture one image of the scene, with a resolution of 512 $\times$ 384. Fig. 2 shows two of the captured images. We record each camera's FoV parameters, so that the joint entropy and the corresponding η_H can be estimated using EDM. We also perform joint coding on the captured images, and from the resulting coding rates, we can obtain η_R.

Experiments on different cluster sizes, coding schemes, and coding parameters are performed to evaluate the joint coding efficiency. The cluster size is set to three different values (N = 2, 3, and 4). As for coding schemes, there are many standardized solutions such as the JPEG/JPEG 2000 and the MPEG/H.26x series. For joint coding on multiple images, the redundancy among different images should be removed. The JPEG/JPEG 2000 standards can only reduce the redundancy within a single image; thus, they are not suitable for use here. We use two coding schemes of the H.264 standards: the Baseline profile and the recently developed multi-view coding (MVC) extension. The H.264 reference softwares JM 8.5 [2] and JMVC 2.5 [3] are used, respectively. For both coding schemes, we obtain the coding rates under three quantization steps (QP = 28, 32, and 37). Other key parameters in the coding experiment are listed in Table I.

In (14), the rates of individual coding are obtained by performing intra coding on each image in the cluster, while the rate of joint coding are obtained by performing predictive coding among the images. For predictive coding, the images in the cluster are coded in a sequential order. We also use the dependency graphs in the EDM algorithm to guide the coding process. In a dependency graph, each camera is connected with the camera that is most correlated with it; thus, it is beneficial to perform predictive coding between cameras that are connected in the graph. For example, if three cameras have a dependency graph as $C_1 \rightarrow C_2 \rightarrow C_3$, for joint coding of the images $\{X_1, X_2, X_3\}$, we take X_1 as the reference image and encode it first, and then encode X_2 based on the prediction of X_1, and X_3 based on the prediction of X_2. The total joint coding rate is a sum of the coding rates of the three images.

When deploying the cameras around a scene, we let their locations and sensing directions be pairwisely symmetric with respect to the center of the scene. Consequently, we can have several (at least two) groups of cameras that lead to the same estimated joint coding efficiency (η_H), according to our spatial correlation model and the EDM algorithm. For each value of η_H, we perform joint coding on the corresponding groups of cameras, and take the average value of the resulting actual coding efficiencies (η_R). Comparisons of the corresponding η_H and the average η_R values for the two scenes are given in Figs. 3 and 4. For both scenes, although the actual joint coding efficiency might be smaller than the estimated joint coding efficiency, the actual joint coding efficiency increases as the estimated joint coding efficiency increases.

As shown in Figs. 3 and 4, for the same coding scheme, the value of η_R increases as the quantization step increases: as larger quantization steps result in more distortion, they may have more potential bit savings for joint coding. In particular, compared to the indoor scene "Tables", the outdoor scene "Trees" contains more details such as the textures in the tree leafs and the grass fields. Therefore, the coding performance of the outdoor scene is more sensitive to the extent of quantization. As shown in the figures, the results for the outdoor scene have more deviation when the quantization step varies. The H.264 MVC extension is more advanced than the H.264 Baseline profile, and our experiments also show that the MVC extension always produces lower bit rates under the same coding parameters. However, the joint coding efficiency of the MVC extension is not necessarily larger than that of the Baseline profile. This is because the MVC extension results in smaller denominators in (14) than the Baseline profile.

In general, the actual joint coding efficiency is proportional to the estimated joint coding efficiency, and such feature is independent of cluster sizes, coding methods, and levels of distortion. Therefore, the EDM scheme can effectively predict the joint coding performances for different sets of cameras for typical applications of wireless multimedia sensor networks.

B. Compression Performance of DMCP

We now study the compression performance of DMCP in terms of clustered coding efficiency, which has the form similar to (13), except that the joint entropy in the entire network is equal to the total entropy produced in the entire network after DMCP is performed. We consider a network with camera sensor nodes uniformly deployed in a 100 $\times$ 100 region. We vary the network size n and sensing radius R, and measure the cluster coding efficiency in Fig. 5. We observe that the DMCP incurs up to 10%–23% coding rate reduction in WMSNs. The increase in the clustered coding efficiency under larger sensing radius can be attributed to the following: larger sensing radius leads to higher probability of two adjacent nodes having overlapped FoVs, thus inducing more visual redundancy in the network. The DMCP ensures that these increased redundancy can be effectively identified and removed, thus giving a better compression performance. We also observe that the increase in the number of nodes does not impact the coding efficiency significantly, and thus, the DMCP provides good compression scalability.

We now study the impact of sensing direction $\vec{V}$ and offset angle α on the compression performance of DMCP. The devi-

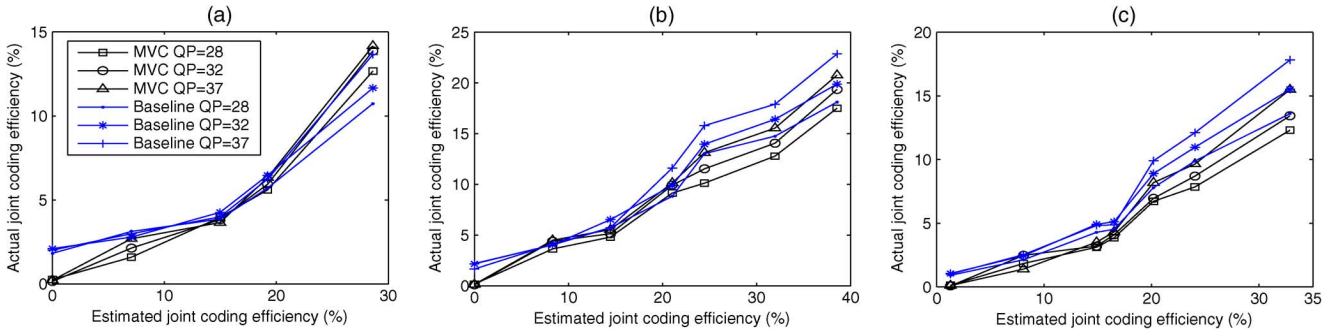

Fig. 3. Joint coding performance of the indoor scene "Tables". (a) $\mathrm{N} = 2$. (b) $\mathrm{N} = 3$. (x) $\mathrm{N} = 4$.

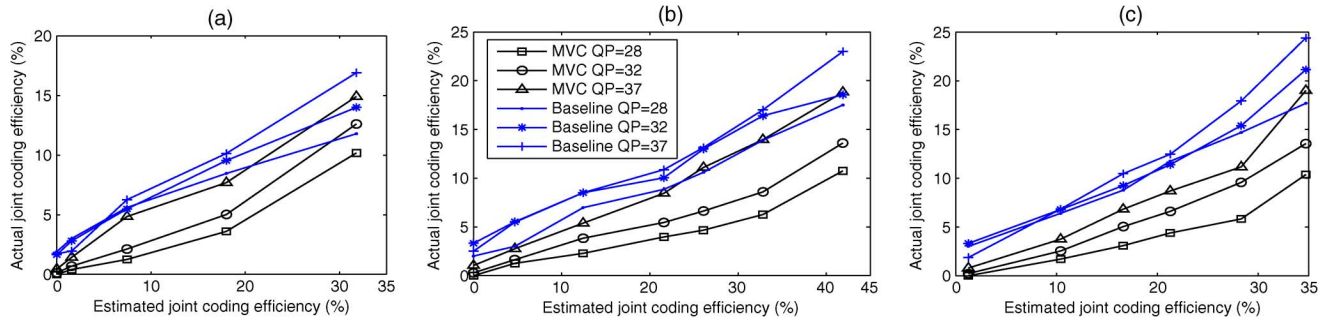

Fig. 4. Joint coding performance of the outdoor scene "Trees". (a) $\mathrm{N} = 2$. (b) $\mathrm{N} = 3$ (x) $\mathrm{N} = 4$.

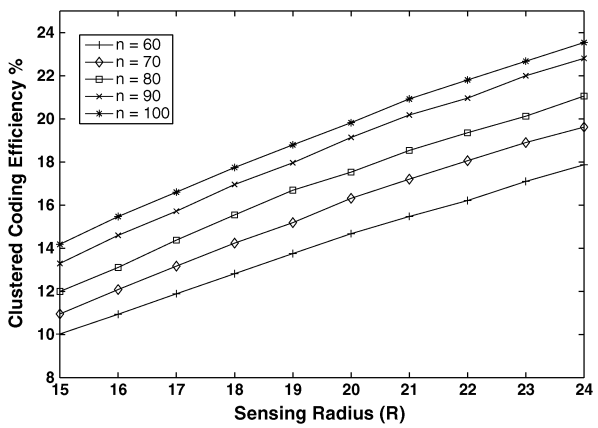

Fig. 5. Compression performance versus network size n and sensing radius R.

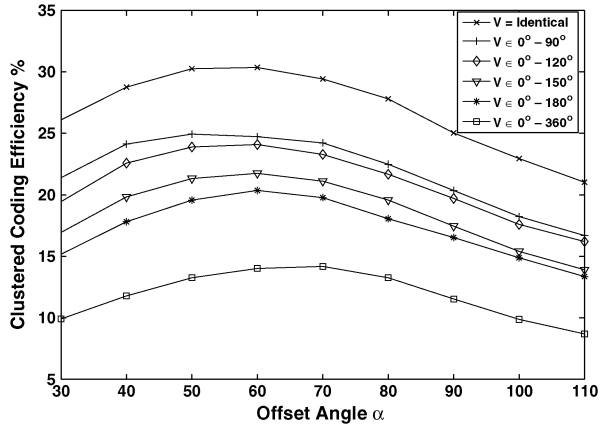

Fig. 6. Compression performance versus sensing direction $\vec{V}$ and offset angle α.

ation in the sensing directions of multiple camera sensors directly affects the similarity among their retrieved images. For a group of sensors with similar sensing directions, there is high probability that they may capture the similar visual content, thus leading to more redundancy in the network. The DMCP ensures that the sensor nodes with similar directions are grouped together, aiming to reduce the redundancy to the maximum extent. Fig. 6 depicts the coding efficiency of DMCP under changing sensing direction patterns. Here, each sensor node is randomly assigned a sensing direction within a degree region, and wider region leads to larger direction deviation. We observe that a substantial coding efficiency (10%–15%) is achieved even in the worst scenario, e.g., each sensor randomly selects a direction within a region of 0°–360°, while the optimal coding scenario (20%–29%) occurs when all the cameras have identical sensing directions.

Besides sensing direction, offset angle also has significant impact on compression efficiency. In Fig. 6, as the offset angle increases, we observe the elevation in coding efficiency, followed by a gradual decrease. This phenomenon is attributed to the following: a wide offset angle leads to a large FoV. Thus, there is greater probability that adjacent cameras cover a large common area. This indicates that more redundancy exists in the network. Therefore, higher compression performance is achievable by DMCP. When the offset angle is over a threshold, e.g., 60°–70° in Fig. 6, the increase in offset angle leads to larger size of nonoverlapped FoVs than overlapped ones, thus incurring a reduced compression efficiency.

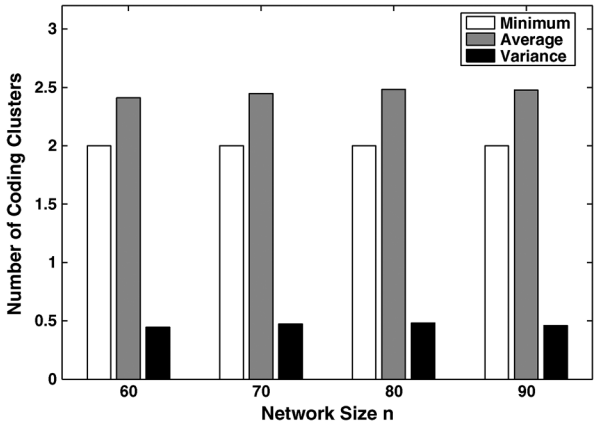

Fig. 7. Average versus minimum number of CHs covering each node.

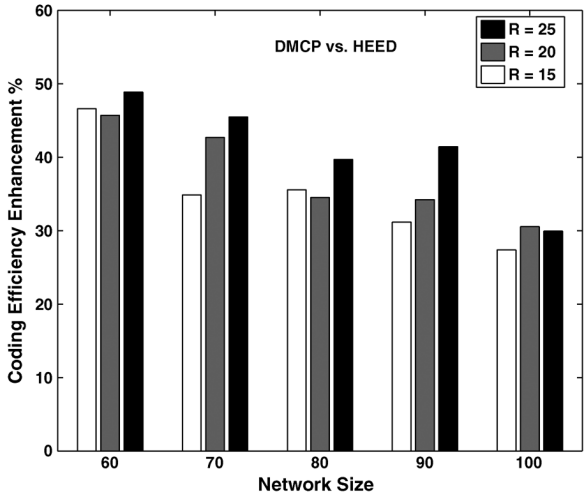

Fig. 8. Coding efficiency enhancement of DMCP compared with HEED.

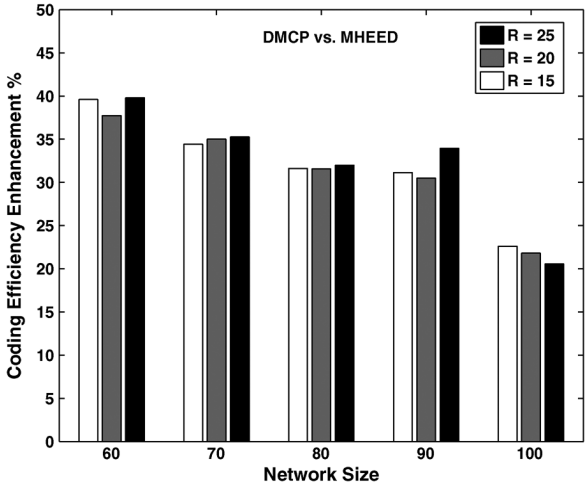

Fig. 9. Coding efficiency enhancement of DMCP compared with modified HEED.

We now investigate the decoding reliability of DMCP by examining the minimum and average number of CHs covering each camera sensor. As shown in Fig. 7, the minimum number of CHs for each sensor is 2. Meanwhile, we observe that the average number of CHs covering each node exceeds 2. This indicates that some camera sensors are included in more than 2 coding clusters, thus providing additional decoding robustness at data sink. In addition, low variance in the number of CHs is shown in Fig. 7, which proves the fairness of DMCP in terms of coverage performance.

We next compare DMCP to the hybrid energy-efficient distributed clustering (HEED) protocol [21] and its modified version MHEED. HEED is a well-known clustering protocol that is specially designed for wireless sensor networks that deal with scalar data. This protocol constructs a hierarchical network architecture by two phases: CH selection and cluster member assignment. In the first phase, sensor nodes are selected as CHs probabilistically. More specifically, each node is given a initial probability p (i.e., 0.05 in [21]) with which it becomes a CH. In the first iteration, each sensor uniformly draws a value between 0 and 1 and compares this value with the initial probability. If this value is less than p, the sensor becomes a CH and all its neighbors are covered. After this iteration, many sensors may still be uncovered since the initial probability (i.e., 0.05) is very small. Therefore, in each of the following iterations, every sensor doubles p and with this probability the uncovered sensors become new CHs. When p reaches 1, the first phase completes. In the second phase, each sensor is assigned to the closest CH as its cluster member. Different from DMCP, HEED protocol is a compression-unaware approach. To fairly compare DMCP with HEED, we design a modified HEED (MHEED) by incorporating the proposed EDM scheme. Specifically, MHEED uses the same procedure as HEED for the CH selection phase. In the second phase, we use the average cluster entropy, instead of node proximity to the CHs, as the metric to associate sensors with CHs. That is, each sensor joins the cluster with the minimum average entropy, a ratio of the estimated joint entropy of the cameras covered by a CH to the number of cameras it covers.

In Figs. 8 and 9, we measure the coding efficiency of HEED and MHEED, respectively, and evaluate the coding efficiency enhancement of DMCP, compared with HEED and MHEED, varying the network size n and sensing radius R. Since DMCP exploits the inherent correlation structure of multiple cameras, it is expected that DMCP can achieve higher coding efficiency by finely identifying and properly selecting a set of clusters that leads to higher compression performance. Accordingly, as shown in Figs. 8 and 9, DMCP achieves 28%–50% and 20%–40% enhancement, compared with the coding efficiency of HEED and MHEED, respectively. Meanwhile, in Figs. 8 and 9, we observe that higher enhancement is achieved under smaller network size (i.e., smaller camera density because of the fixed deployment area). This phenomenon is attributed to the fact that smaller camera density leads to higher variability of the joint entropy of different clusters. In this case, the visual correlation-based strategy, DMCP, has more evident advantage over compression-unoriented approaches like HEED and MHEED. Moreover, we observe that less enhancement is achieved in Fig. 9 than in Fig. 8. This implies that MHEED achieves higher coding efficiency than HEED. This is as expected because MHEED uses the average entropy as the metric

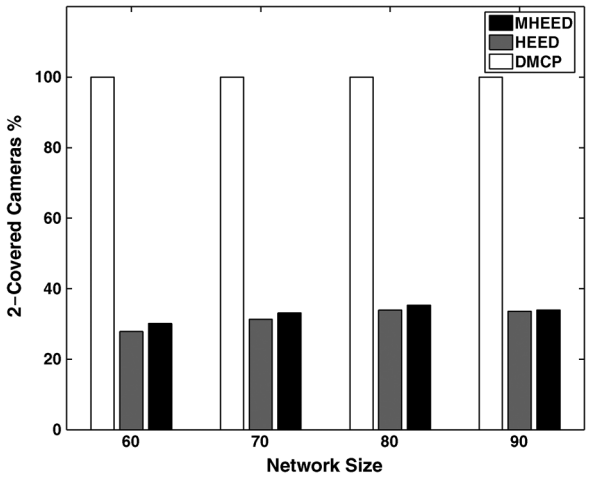

Fig. 10. Percentage of cameras covered by more than two clusters.

in selecting CHs, which is superior to just selecting the closest CH, because the average entropy of a node is a measure of the expected coding performance if this node is selected as CH. It is also worth noticing that in Fig. 9, the enhancement under different sensing radius settings is comparable, which indicates MHEED is less sensitive to the changing sensing radius than HEED because MHEED partially exploits the visual correlation information in clustering procedures.

We now investigate the reliability of HEED and MHEED by examining the percentage of 2-covered cameras, i.e., the cameras that are covered by more than two clusters. As shown in Fig. 10, the percentage of 2-covered cameras under HEED and MHEED is around 30%, comparing with 100% under DMCP. This implies that DMCP establishes a more robust coding hierarchy than HEED and MHEED. Meanwhile, we also observe a slight elevation in the percentage of 2-covered cameras under HEED and MHEED as the network size or node density increases. This is due to the fact that higher node density gives rise to more 2-covered cameras. This could further increase the percentage of cameras covered by more than two clusters in the network.

VI. Conclusions

In this paper, we provide an information theoretic data compressing framework for WMSNs with an objective to maximize the global compression gain with enhanced decoding reliability. In particular, an EDM scheme is developed to predict the compression efficiency for an arbitrary coding cluster containing multiple correlated cameras. This method is only related to the camera settings, and therefore independent of any specific image types and coding algorithms. Using the results of EDM, we then propose DMCP to select a set of coding clusters with minimum total entropy in a fully distributive manner, such that each camera sensor is covered by at least two coding clusters. The approximation factor of DMCP is also investigated. Our evaluation results show that the EDM can effectively predict the coding rates produced by practical coding standards, while the data framework yields up to 10%–23% rate reduction compared with the conventional independent coding.

Appendix

Algorithm 4: Greedy Coding Cluster Selection Algorithm

1) $C \leftarrow \emptyset$, and $E' \leftarrow E$

2) For each $S \in \mathcal{S}$, $x(S) \leftarrow 0$.

3) **while** $E' \neq \emptyset$ **do**

4) $s \leftarrow \arg\min_{X \in \mathcal{S}} H(X)/|X \cap E|$, and $x(s) \leftarrow 1$.

5) $C \leftarrow C \cup s$, and $\mathcal{S} \leftarrow \mathcal{S} \setminus s$.

6) $E' \leftarrow E' \setminus \{e \in s \cap E : \sum_{S:e \in S} x(S) \geq 2, S \in C\}$.

7) **end while**

References

[1] GCC, the GNU Compiler Collection. [Online]. Available: http://gcc.gnu.org/.
[2] JVT Reference Software JM 8.5. [Online]. Available: http://iphome.hhi.de/suehring/tml/download/old_jm/jm8.5.zip.
[3] JMVC Reference Software JMVC 2.5. [Online]. Available: http://ftp3.itu.int/av-arch/jvt-site/2008_10_Busan/JVT-AC207.zip.
[4] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Elsevier Comput. Netw. J.*, vol. 51, no. 4, pp. 921–960, 2007.
[5] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: Applications and testbeds," *Proc. IEEE (Invited Paper)*, vol. 96, no. 10, pp. 1588–1605, Oct. 2008.
[6] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[7] R. Dai and I. F. Akyildiz, "A spatial correlation model for visual information in wireless multimedia sensor networks," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1148–1159, Oct. 2009.
[8] D. Devarajan, Z. Cheng, and R. Radke, "Calibrating distributed camera networks," *Proc. IEEE*, vol. 96, no. 10, pp. 1625–1639, Oct. 2008.
[9] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Englewood Cliffs, NJ: Prentice-Hall, 2004.
[10] Q. Lu, W. Luo, J. Wang, and B. Chen, "Low-complexity and energy efficient image compression scheme for wireless sensor networks," *Comput. Netw.*, vol. 52, no. 13, pp. 2594–2603, Sep. 2008.
[11] H. Ma and Y. Liu, "Correlation based video processing in video sensor networks," in *Proc. IEEE Int. Conf. Wireless Networks, Communications and Mobile Computing*, Jun. 2005, pp. 987–992.
[12] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: A survey," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 986–1004, Aug. 2003.
[13] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: A video coding paradigm with motion estimation at the decoder," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2436–2448, Oct. 2007.
[14] S. Rajagopalan and V. V. Vazirani, "Primal-dual RNC approximation algorithms for set cover and covering integer programs," *SIAM J. Comput.*, vol. 28, no. 2, 1999.
[15] D. Slepian and J. Wolf, "Noiseless coding of correlated information sourcess," *IEEE Trans. Inf. Theory*, vol. IT-19, pp. 471–480, Jul. 1973.
[16] R. Wagner, R. Nowak, and R. Baraniuk, "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2003, pp. 597–600.
[17] P. Wang, R. Dai, and I. F. Akyildiz, "Collaborative data compression using clustered source coding for wireless multimedia sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2010.
[18] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[19] H. Wu and A. A. Abouzeid, "Energy efficient distributed image compression in resource-constrained multihop wireless networks," *Comput. Commun.*, vol. 28, no. 14, pp. 1658–1668, Sep. 2005.
[20] M. Wu and C. W. Chen, "Collaborative image coding and transmission over wireless sensor networks," *EURASIP J. Adv. Signal Process., 2007*, 2007.
[21] O. Younis and S. Fahm, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct. 2004.

Pu Wang (S'10) received the B.S. degree in electrical engineering from the Beijing Institute of Technology, Beijing, China, in 2003 and the M.Eng. degree in computer engineering from the Memorial University of Newfoundland, St. John's, NL, Canada, in 2008. He is currently pursuing the Ph.D. degree in electrical engineering at Georgia Institute of Technology, Atlanta, under the supervision of Prof. Ian F. Akyildiz.

Currently, he is a graduate research assistant in the Broadband Wireless Networking Laboratory (BWN Lab), School of Electrical and Computer Engineering, Georgia Institute of Technology. His research interests include wireless sensor networks and mobile ad-hoc networks.

Rui Dai (S'08) received the B.S. and M.S. degrees in electrical and computer engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2004 and 2007, respectively. She is currently pursuing the Ph.D. degree at the Broadband Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta.

Her research interests include wireless sensor networks and multimedia communications.

Ian F. Akyildiz (M'86–SM'89–F'96) received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981 and 1984, respectively.

Currently, he is the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering (ECE), Georgia Institute of Technology, Atlanta. Since June 2008, he has been an Honorary Professor with the School of Electrical Engineering at the Universitat Politecnico de Catalunya, Barcelona, Spain. His current research interests are in cognitive radio networks, wireless sensor networks, and nano-communication networks.

Dr. Akyildiz is the Editor-in-Chief of *Computer Networks (COMNET) Journal* as well as the founding Editor-in-Chief of the *Ad Hoc Networks Journal* and *Physical Communication Journal*, all with Elsevier. He received numerous awards, including the 1997 IEEE Leonard G. Abraham Prize Award (IEEE Communications Society) for his paper entitled "Multimedia group synchronization protocols for integrated services architectures" published in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in January 1996; the 2002 IEEE Harry M. Goode Memorial Award (IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking"; the 2003 IEEE Best Tutorial Award (IEEE Communication Society) for his paper entitled "A survey on sensor networks," published in the IEEE COMMUNICATIONS MAGAZINE, in August 2002; the 2003 ACM Sigmobile Outstanding Contribution Award with the citation "for pioneering contributions in the area of mobility and resource management for wireless communication networks"; the 2004 Georgia Tech Faculty Research Author Award for his "outstanding record of publications of papers between 1999–2003"; the 2005 Distinguished Faculty Achievement Award from the School of ECE, Georgia Tech; the 2009 Georgia Tech Outstanding Doctoral Thesis Advisor Award for his 20+ years service and dedication to Georgia Tech and producing outstanding Ph.D. students; and the 2009 ECE Distinguished Mentor Award from School of ECE, Georgia Tech. He has been a Fellow of the Association for Computing Machinery (ACM) since 1996.