This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Collaborative Data Compression Using Clustered Source Coding for Wireless Multimedia Sensor Networks

Pu Wang, Rui Dai, and Ian F. Akyildiz

Broadband Wireless Networking Laboratory
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332
Email: {pwang40, aprildai, ian}@ece.gatech.edu

***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 data compression framework with an objective to maximize the overall compression of the visual information gathered in a WMSN. To achieve this, 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 novelty of EDM relies on its independence of the specific image types and coding algorithms, thereby providing a generic mechanism for prior evaluation of compression under different coding solutions. Utilizing the predicted results from EDM, a distributed multi-cluster coding protocol (DMCP) is 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.**

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 [1]. A WMSN promises a wide range of potential applications such as multimedia surveillance, advanced health care delivery, and industrial process control [1]. 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.

This work was supported by the US National Science Foundation (NSF) under Grant No. ECCS-0701559.

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) [4], 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 [9], [5] 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 [9], can achieve high compression performance. However, they require extensive computation at the encoder, which places heavy burden on the resource-constrained sensor nodes. In contrast, distributed source coding, such as Slepian-Wolf Coding [7], 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 [1], 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

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

that requires centralized realization, distributed source coding allows each sensor to encode its own data separately, assuming a prior knowledge of local correlation structure in its own cluster [7]. 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 of the existing solutions for multimedia processing is that they are generally application dependent [8], [10]: different types of images will require different processing schemes [3]. Thus, how to design solutions whose applicability and flexibility would not be limited by the specific applications is of paramount importance. On the other hand, as the images captured in a WMSN may contain a substantial amount of redundancy, the construction of a compression-oriented cluster hierarchy, which can fully reduce this redundancy, is one of the primary tasks in WMSNs.

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: (i) compression efficiency prediction, and (ii) 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. 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.

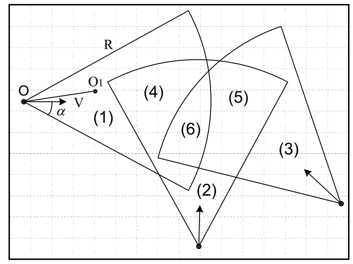

Fig. 1. Field of views of multiple cameras.

The rest of this paper is organized as follows. Section II mathematically formulate the problems in the proposed data compression framework. In Section III, we present the EDM algorithm 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. Clustered Source Coding

In a multimedia sensor network, 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' FoVs are overlapped with each other. These overlapped FoVs incur a certain degree of correlation among the observations of the cameras, which further leads to unnecessary visual redundancy. To remove this redundancy, a group of camera sensors can form a cluster to collaboratively compress their data. Consider a cluster consisting of a cluster head 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 centralized or distributed source coding is applied.

For centralized source coding, each member in a cluster sends its raw or preprocessed data to the cluster head, while the cluster head acts as a single encoder that takes all collected data as inputs. According to Shannon's source coding theorem, 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 optimal encoder is applicable.

For distributed source coding (DSC), each node in a cluster separately encodes its own data, and the cluster head only

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

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 [7] provides a conceptual basis for DSC and establishes the rate region for the rate vector $(R_1, R_2, ...R_N)$:

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

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 centralized or distributed source coding is applied.

B. 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. Since visual information is intrinsically complicated, it is difficult to model the characteristics of visual sources. Moreover, it usually requires expensive computation and communication costs to obtain the dependency characteristics for multiple visual sources, especially when the number of sources (N) is large.

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.

C. 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 Optimal Coding Clustering (OCC) problem can be formally stated as: given a network consisting of a finite set of camera sensors $E = e_1, e_2, ...e_n$ and a set of n subsets of E, $\mathcal{S} = \{S_1, S_2, ...S_n\}$, 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 Entropy-based Divergence Measure (EDM) scheme to estimate the joint entropy of the observations from multiple cameras in WMSNs. This algorithm only takes the cameras' settings as inputs without requiring the knowledge of the specific applications, thereby providing a generic framework for prior evaluation of compression under different coding solutions. Moreover, this algorithm induces little communication costs since sensor nodes only need to exchange their camera settings via short messages among their direct neighbors, and then only low complexity computations are required for joint entropy estimation. The algorithm consists of the following two components.

1) *Area partition for overlapped FoVs.* Given a group of cameras, their FoVs are divided into several partitions, such that each partition is covered by the same set of cameras.
2) *Joint entropy estimation for partitions.* For each partition, a dependency graph is constructed based on the correlation among the cameras. The joint entropy of the partition 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 partitions.

A. Area Partition for Overlapped Field of Views

A camera is a directional sensor with limited sensing range. It can only observe the objects within its field of view (FoV). In the following analysis we use a simplified 2-D FoV model [4]. 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. An arbitrary point O_1 is in the FoV of the camera if it is in the camera's sensing radius and also within the offset angle of the FoV, given as

$$\begin{cases} |\vec{OO_1}| \leq R \\ \theta \leq \alpha, \end{cases} \tag{2}$$

where θ is the angle between $\vec{OO_1}$ and $\vec{V}$.

When multiple cameras are deployed in a field, their FoVs are usually overlapped with each other. Fig. 1 shows an example of three cameras deployed on the ground plane. The FoVs of the three cameras can be divided into six different areas, with each area covered by a different set of cameras.

To estimate the joint entropy of multiple cameras, we need to investigate the overlapping pattern of their FoVs first. We

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

consider the case when N cameras $(C_1, C_2, \cdots, C_N)$ are deployed on the ground plane. Denote the FoV of an individual camera C_i as $\mathcal{A}_i(O_i, R_i, \vec{V}_i, \alpha_i)$, and the overall FoV for these cameras as $\mathcal{A}$ ($\mathcal{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_N\}$). Our goal is to divide $\mathcal{A}$ into several partitions $(P_1, P_2, \cdots, P_M)$, such that each partition belongs to the FoVs of the same set of cameras.

We introduce a grid-based approach to divide the overall FoV $\mathcal{A}$ into partitions, which is described in the first part of Algorithm 1. We firstly divide the overall FoV into discrete grids with a step size of h, as shown in Fig. 1. Using the condition in (2), we can check for each grid if its center is in the FoV of a camera. If so, we approximately regard that the whole grid is in the camera's FoV. Therefore, partitions can be found by traversing all the grids and grouping the grids that belong to the same set of cameras together.

After the overall FoV of the cameras in a group is divided into partitions, we need to estimate the entropy of each partition in order to obtain the total joint entropy of the cameras. In Section B, we derive a preliminary expression to estimate the conditional entropy between two cameras. According to this expression, we then develop a dependency graph based algorithm to estimate the joint entropy of multiple cameras in a partition in Section C.

Algorithm 1 Joint Entropy Estimation for Multiple Cameras

For a group of cameras $C_1, C_2, \cdots, C_N$, get the FoV of each Camera: $\mathcal{A}_i(O_i, R_i, \vec{V}_i, \alpha_i)$. $\mathcal{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_N\}$. The visual information observed by C_i is X_i.
Divide $\mathcal{A}$ into K small grids; each grid is of size $h * h$.
for k = 1 to K **do**
  for n = 1 to N **do**
    Check if grid $G(k)$ is in $\mathcal{A}_n$.
  end for
  Assign grid $G(k)$ to a partition.
end for
Get the area partition results: $\mathcal{A} = (P_1, \cdots, P_M)$.
for i = 1 to M **do**
  Estimate the individual entropy that a camera contributes to P_i according to equation (4).
  Calculate the correlation matrix for partition P_i.
  Estimate $H(P_i)$ using Algorithm 2.
end for
$H(X_1, \cdots, X_N) = H(P_1) + \cdots + H(P_M)$.

B. Conditional Entropy Estimation

Denote the cameras in partition P_i as $(C_1, \cdots, C_n)$. For the kth camera C_k in partition P_i, denote its observed visual information by X_k, and denote its observation about this partition by $X_k(P_i)$. The amount of information of partition P_i is the joint entropy of the observations about this partition from the cameras $(C_1, \cdots, C_n)$, given by

$$H(P_i) = H(X_1(P_i), \cdots, X_n(P_i)). \tag{3}$$

A camera can easily estimate the entropy of its own observations locally. That is to say, for camera C_k we can obtain the entropy of its observations $H(X_k)$. Suppose $S(P_i)$ is the area of partition P_i and $S(\mathcal{A}_k)$ is the area of the entire FoV of C_k. Assuming that the amount of information that camera C_k contributes to P_i is approximately proportional to the area of P_i, the entropy of the observation X_k about partition P_i, $H(X_k(P_i))$, can be estimated as

$$H(X_k(P_i)) \approx \frac{S(P_i)}{S(\mathcal{A}_k)} H(X_k). \tag{4}$$

With the individual entropies obtained by (4), the correlation characteristics among these cameras are also needed to estimate the joint entropy in (3). In our earlier work, we proposed a novel spatial correlation model for visual information in WMSNs [2]. By studying the sensing model and deployments of cameras, a spatial correlation coefficient is derived for two cameras that can observe a same area of interest. Assume that all the cameras in a network have the same focal length. Specifically, for camera C_j and camera C_k in partition 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 is derived as

$$\rho_{j,k} = f(O_j, \vec{V}_j, O_k, \vec{V}_k, P_i) \tag{5}$$

which indicates that the spatial correlation coefficient $\rho_{j,k}$ is a function of the two cameras' locations (O_j, O_k) and sensing directions ($\vec{V}_j$, $\vec{V}_k$) as well as the location of partition P_i.

More importantly, the correlation coefficient is related to the joint entropy of two cameras in [2]. For camera C_j and camera C_k in partition P_i, the joint entropy of the observations of partition P_i at C_j and C_k is estimated as

$$H(X_j(P_i), X_k(P_i)) \approx (1 - \frac{1}{2}\rho_{j,k})(H(X_j(P_i)) + H(X_k(P_i))) \tag{6}$$

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 the result in (6) we can obtain an expression of conditional entropy as follows:

$$\begin{aligned} H(X_j(P_i)|X_k(P_i)) &= H(X_j(P_i), X_k(P_i)) - H(X_k(P_i)) \\ &\approx (1 - \frac{\rho_{j,k}}{2})H((X_j(P_i)) - \frac{\rho_{j,k}}{2}H(X_k(P_i)) \end{aligned} \tag{7}$$

where $H(X_j(P_i)|X_k(P_i))$ is the entropy of $X_j(P_i)$ with the knowledge of $X_k(P_i)$. In the next section we utilize this result to estimate the joint entropy of more than two cameras.

C. Joint Entropy of a Partition

As introduced above, the joint entropy of the observations from two cameras is estimated from the correlation coefficient between them (6). To estimate the joint entropy of a partition, we should be able to deal with the case of more than two cameras. In this section, we propose a dependency graph based algorithm to estimate the joint entropy of a partition.

We study a two cameras' case as a preliminary example. Suppose there are only two cameras (C_1 and C_2) in a partition P_i. In this partition, C_1 is most correlated with

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

C_2. We can depict this relationship as a dependency graph: $C_2 \rightarrow C_1$. The joint entropy of the observations from C_1 and C_2 can then be calculated by traversing the dependency graph. The source node C_2 contributes the entropy of its observations about the partition, $H(X_2(P_i))$, and the node C_1 contributes the conditional entropy of its observations with respect to the source node, $H(X_1(P_i)|X_2(P_i))$, so the joint entropy is calculated by adding these two terms together: $H(X_1(P_i), X_2(P_i)) = H(X_2(P_i)) + H(X_1(P_i)|X_2(P_i))$. The dependency graph can also be constructed as $C_1 \rightarrow C_2$, from which we can get the same result of joint entropy.

Motivated by the above example, we design a dependency graph based algorithm to estimate the joint entropy of a partition. We firstly construct a dependency graph to describe the dependency characteristics among these cameras. 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 partition is then calculated by traversing all the nodes in the graph along the directed edges. The detailed steps are described in Algorithm 2.

Suppose a group of cameras $(C_1, C_2, \cdots, C_n)$ can observe partition P_i. We can obtain a correlation matrix $(\rho_{j,k})_{n*n}$ based on (5), where $\rho_{j,k}$ is the correlation coefficient between the observations of partition P_i from camera C_j and camera C_k. To simplify the problem, we assume limited number of dependencies: each camera is dependent on the camera that is most correlated with it. For camera C_j in the partition, we can easily find out its most correlated camera by searching the correlation matrix $(\rho_{j,k})_{n*n}$. For example, if camera C_j is most correlated with camera C_k, we say that C_j is dependent on C_k, and thus, 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 a directed acyclic graph with the following additional 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 in the graph; a dependency graph may have several source nodes, but each node in the graph can have at most one direct predecessor. Once a dependency graph with these features is constructed, the joint entropy can be estimated by traversing all the nodes in the graph and adding the entropies of the nodes together. A source node will contribute its individual entropy to the joint entropy. A non-source node has its most correlated node as its direct predecessor. It will contribute its conditional entropy with respect to its direct predecessor to the joint entropy.

For example, five cameras in partition P_i forms a dependency graph as $C_1 \rightarrow C_3 \rightarrow C_5, C_2 \rightarrow C_4$. With C_1 and C_2 as the source nodes, the joint entropy of partition P_i is calculated as $H(P_i) = H(X_1(P_i), \cdots, X_5(P_i)) = H(X_1(P_i)) + H(X_3(P_i)|X_1(P_i)) + H(X_5(P_i)|X_3(P_i)) + H(X_2(P_i)) + H(X_4(P_i)|X_2(P_i))$.

Since the FoVs for a group of cameras are divided into several partitions in Section A, and these partitions are independent of each other, the total joint entropy is the sum of the entropies of all the partitions. For a group of cameras with observations $(X_1, \ldots, X_N)$, with the cameras' FoVs divided into partitions $(P_1, \ldots, P_M)$, the total joint entropy is given by

$$H(X_1, \cdots, X_N) = H(P_1) + \cdots + H(P_M) \tag{8}$$

where $H(P_i)(i = 1, \cdots, M)$ is obtained by Algorithm 2.

Algorithm 2 Dependency Graph Based Entropy Estimation

```
P_i: {C_1, C_2, ..., C_n} with correlation matrix (ρ_{j,k})_{n*n}.
for j = 1 to n do
    neighbor(C_j) = arg max_{k≠j} (ρ_{j,k});
end for
for k = 1 to n do
    for j = 1 to n do
        if C_j has no predecessors and neighbor(C_j) = k then
            Add C_k → C_j into the dependency graph;
            Predecessor(C_j) = C_k;
            Mark C_j as a traversed node.
        end if
    end for
    Break when all the cameras are traversed;
end for
for j = 1 to n do
    if C_j has no predecessor then
        Add H(X_j(P_i)) to H(P_i);
    else if Predecessor(C_j) = C_k then
        Add H(X_j(P_i)|X_k(P_i)) (7) to H(P_i);
    end if
end for
return H(P_i).
```

The entire entropy estimation algorithm (Algorithm 1) can be run at each sensor node. To estimate joint entropy, a node just need to acquire the FoV information and individual entropies from its neighbors. Therefore, it does not require expensive communication costs in the network. The estimated joint entropy will serve as a criteria for the protocol in the following section.

IV. Data Compression using Clustered Source Coding

After the camera sensors are 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, centralized algorithm is not suitable for use here. The coding cluster selection should only depend on local information to achieve global compression optimization gain. Meanwhile, the locally established coding hierarchy should have two different clusters for each node, such that the compressed images can be properly reconstructed as long as one cluster functions normally. In this section, we first formulate the optimal coding clustering (OCC) problem as an integer program, and shows that the OCC problem is NP hard. Accordingly, we introduce a centralized greedy algorithm to provide a $\ln \Delta$ factor approximation to the optimal solution. Next, we present a distributed multi-cluster coding protocol (DMCP) for coding hierarchy establishment. Furthermore, it is shown to achieve the same approximation guarantee of $\ln \Delta$ as the centralized algorithm.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

A. Centralized Coding Hierarchy Generation

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 that at least two of the clusters containing it are selected.

$$\begin{aligned} \text{MIN} \quad & \sum_{S \in \mathcal{S}} H(S) x_S \\ s.t \quad & \sum_{S: e \in S} x_S \geq 2, \quad e \in E \\ & x_S \in \{0,1\}, \quad S \in \mathcal{S} \end{aligned} \tag{9}$$

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 [6]. In other words, the approximation ratio $\ln \Delta$ achieved by the greedy algorithm is best one for 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 2 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 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 distributed network like WMSN, the centralized operations are not desirable because they have limited flexibility and scalability to react to environment and network dynamics. In addition, the energy constraint of sensor nodes prohibits network-wide information exchange specially in large-scale networks like WMSNs. Next, we will propose a protocol to implement the centralized greedy algorithm in a fully distributive manner such that only local information exchange is needed 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 cluster head (CH) locaters. The CH locaters will not serve as the normal cluster heads but indicate the coordinates at which the future mobile or fixed cluster heads 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 definitions, which are utilized in DMCP.

Definition 3: The *neighbor set* of a node is a set consisting of the node itself and all its direct neighbors.

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 3 Distributed Multi-cluster Coding Protocol

```
state(e) ∈ {black, grey, half grey, white, uncovered}
state(e) ← white, send & receive state(e)
N_e ← {e' : state(e') = white} ∪ {e}
{Discover neighbor set N_e}
counter(e) = 0 {Set CH counter}
while state(e) = uncovered do
  U_e ← {e' ∈ N_e : state(e') = uncovered}
  {Calculate serving set U_e}
  EC_e ← H(N_e)/|U_e|
  {Calculate coding effectiveness EC_e }
  send & receive ADV msg
  if EC_e = min_{e'∈U_e}{EC_e'} then
    state(e) ← black, and counter = 1
    send COVERAGE msg
  else
    wait until the selection of a new black node times out
    if no COVERAGE received then
      state(e) remains
    else if counter = 0 then
      state(e) ← halfgrey, and counter(e) = 1
    else if counter = 1 then
      state(e) ← grey, and counter(e) = 2
    end if
  end if
end while
Process_Grey_Black()
```

Now, the proposed DMCP establishes a clustered coding hierarchy as follows. Initially, 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, a *uncovered* node discovers its *serving set* and calculates its *cluster entropy*. Based on these information, a *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. Otherwise, a *uncovered* node can encounter the following scenarios: 1) if

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

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. 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 . 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. For the last two cases, a 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 send out COVERAGE messages to its *uncovered* neighbors. 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*. A *black* finally becomes a cluster head locater until the value of its CH counter reaches 2 on receiving a COVERAGE message.

Algorithm 4 $Process\_Grey\_Black()$

```
U_e ← {e ∈ N_e : counter(e) < 2}
while |U_e| < |N_e| do
  if state(e) = grey then
    recalculate coding effectiveness EC_e
    if EC_e = min_{e'∈U_e} {EC_e'} then
      state(e) ← black, and send COVERAGE msg
    else
      wait until the new black selection times out
      if |U_e| < |N_e| then
        state(e) ← grey
      end if
    end if
  else if state(e) = black then
    wait until a COVERAGE is received
    counter(e) = 2
  end if
end while
Node e becomes a cluster member
```

The above procedures are performed by all nodes until each of them becomes either a cluster head 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. Moreover, it is easy to show that if the minimum node degree is larger than 1 (or every node has at least one neighbor), every node terminates when it has two *black* nodes (perhaps including the node itself) within its 1-hop range. Therefore, the DMCP guarantees that each node is included in at least two coding clusters. The pseudo-code of the above procedures is described in *Algorithm* 3 and *Algorithm* 4.

C. Approximation Ratio

Theorem 1: The DMCP computes a $\ln \Delta$ approximation for the optimal coding clustering problem.

Proof: According to DMCP, each non-*black* node can be a potential cluster head (CH) locater, and each CH locater associated with its immediate neighbors constitutes a coding cluster. Thus, selecting a set of clusters is equivalent to picking a set of CH locaters. Whether a non-*black* node can be selected as a CH locater only depends on its *coding effectiveness*, which is fully determined by its *cluster entropy* and *serving set* cardinality. Since 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. Based on the discussion above, 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. In other words, we prove that the distributed approach can select the same coding clusters as the centralized one. Due to the distributed manner of DMCP, multiple non-*black* nodes can go into *black* simultaneously. 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 can not 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 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. Since the centralized greedy algorithm computes a $\ln \Delta$ approximation for the optimal coding clustering problem, the DMCP can achieve the same approximation guarantee of $\ln \Delta$. ■

As shown in Section A, the OCC problem can be reduced to CSMC problem, for which $\ln \Delta$ is the best approximation ratio. Therefore, approximation ratios better than $\ln \Delta$ is also unlikely for the OCC problem. Then, we can conclude that no protoocls can perform better than the proposed DMCP in terms of appximation factor.

V. Performance Evaluation

We evaluate the performance of the proposed data compressing framework through simulations. We first investigate the effectiveness of the EDM scheme by comparing its predicted results with the joint coding performance of practical

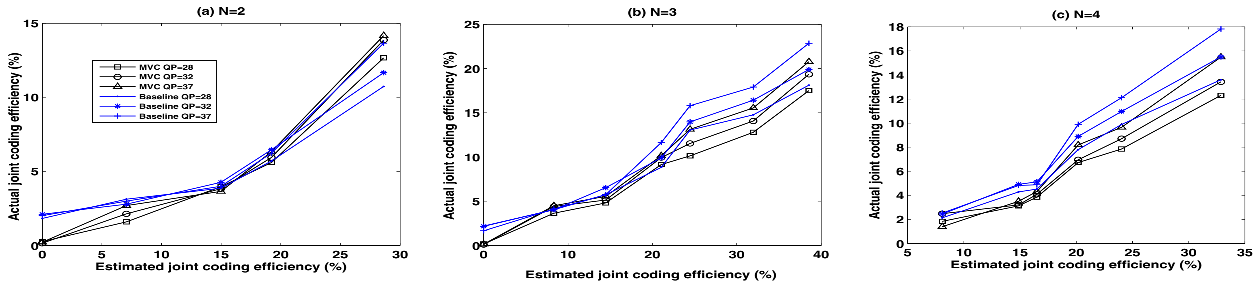

Fig. 2. Joint coding using the H.264 Coding Standards.

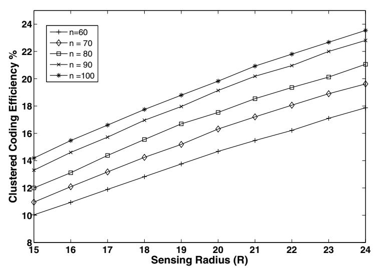

Fig. 3. Compression performance versus network size n and sensing radius R

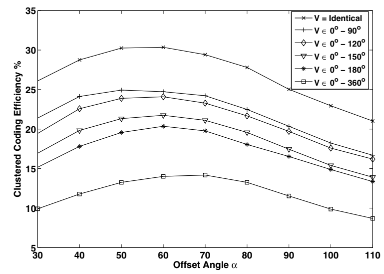

Fig. 4. Compression performance versus sensing direction $\vec{V}$ and offset angle α

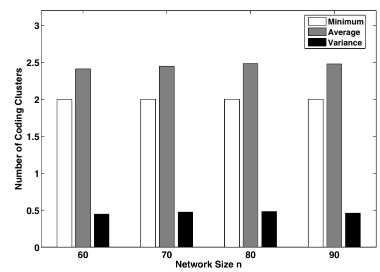

Fig. 5. Average vs minimum number of cluster heads covering each node

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 quantify the performance of clustered joint coding, we define an *estimated joint coding efficiency* as

$$\eta_H = 1 - \frac{H(X_1, \cdots, X_N)}{H(X_1) + \cdots + H(X_N)} \tag{10}$$

where $H(X_1) + \cdots + H(X_N)$ corresponds to the coding rate needed when the cameras compress their observations individually. The estimated joint coding efficiency can be used to predict the percentage of rate savings of joint coding compared to individual coding.

We propose to verify the estimated joint coding efficiency from the results of practical video coding schemes. Similar as the definition above, we introduce an *actual joint coding efficiency* as

$$\eta_R = 1 - \frac{R(X_1, \cdots, X_N)}{R(X_1) + \cdots + R(X_N)} \tag{11}$$

where $R(X_1, \cdots, X_N)$ is the total rate of observations $X_1, \cdots, X_N$ obtained from joint coding, and $R(X_i)(i = 1, \cdots, N)$ is the rate of observation X_i from individual encoding at a camera. The value of η_R will be obtained from the results of practical video coding schemes.

In our experiment, we deploy a number of camera nodes in a field and record each camera's FoV parameters. The joint entropy of multiple cameras and the estimated joint coding efficiency (η_H) can be estimated by the EDM scheme in Section III. We also let each camera capture an image at the same time, and use the widely spread H.264 standard coding algorithms to perform joint coding on these images, so that the actual joint coding efficiency (η_R) can be obtained. We compare the values of η_H with η_R under different cluster sizes (N=2, 3, and 4). Two coding schemes of the H.264 standards are used: the Baseline profile and the recently developed Multi-View Coding (MVC) extension. For both coding schemes, we obtain the coding rates under three quantization steps (QP=28, 32, and 37), where larger quantization steps result in more distortion of the coded images. Comparisons of the corresponding η_H and η_R are given in Fig. 2.

It can be seen from Fig. 2 that the actual joint coding efficiency increases as the estimated joint coding efficiency increases. The actual joint coding efficiency is smaller than the estimated joint coding efficiency. This is as expected since the estimated joint coding efficiency is calculated from joint entropy, which corresponds to the optimal coding performance. For the same coding scheme, the actual joint coding efficiency increases as the quantization step increases: as larger quantization steps result in more distortion, they may have more potential bit savings. 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 fewer bit rates under the same coding parameters. However, as shown in Fig. 2, 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 (11) 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 distortions (quantization steps) of coding. Therefore, the EDM scheme can effectively predict the joint coding performances for different sets of cameras.

B. Compression Performance of DMCP

We now investigate the compression performance of DMCP in terms of clustered coding efficiency, which has the form similar to equation (10), except that the joint entropy in the entire network is equal to the total entropy produced in the entire network after DMCP is performed. Here, we consider a network with camera sensor nodes uniformly deployed in a 100×100 region. We vary the network size n and sensing radius R, and measure the cluster coding efficiency in Fig. 3. We observe that the DMCP incurs up to 10% - 23 % coding rate reduction in WMSNs, by removing the visual redundancy generated by spatially correlated cameras. The increase in the clustered coding efficiency under larger sensing radius can be attributed to the following: larger sensing radius gives rise to higher probability of two ajacent 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 a good scalability in terms of compression efficiency.

We now study the impact of sensing direction $\vec{V}$ and offset angle α on the compression performance of DMCP. The deviation 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. 4 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. 4, 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. 4, the increase in offset angle leads to larger size of nonoverlapped FoVs than overlapped ones, thus incurring a reduced compression efficiency.

We now investigate the decoding reliability of DMCP by examining the minimum and average number of cluster heads covering each camera sensor. As shown in Fig. 5, the minimum number of cluster heads for each sensor is 2. Meanwhile, we observe that the average number of cluster heads 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 cluster heads is shown in Fig. 5, which proves the fairness of DMCP in terms of coverage performance.

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 entropy-based divergence measure (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.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Elsevier Computer Networks Journal*, 51(4):921–960, 2007.

[2] R. Dai and I. F. Akyildiz. A spatial correlation model for visual information in wireless multimedia sensor networks. *IEEE Transactions on Multimedia*, 11(6):1148-1159, 2009.

[3] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using MATLAB*. Prentice Hall, 2004.

[4] H. Ma and Y. Liu. Correlation based video processing in video sensor networks. In *IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, pages 987–992, 2005.

[5] R. Puri, A. Majumdar, and K. Ramchandran. Prism: A video coding paradigm with motion estimation at the decoder. *IEEE Transactions on Image Processing*, 16(10):2436 – 2448, 2007.

[6] S. Rajagopalan and V. V. Vazirani. Primal-dual rnc approximation algorithms for set cover and covering integer programs. *SIAM Journal on Computing*, 28(2), 1999.

[7] D. Slepian and J. Wolf. Noiseless coding of correlated information sourcess. *IEEE Transactions on Information Theory*, IT-19:471–480, July 1973.

[8] R. Wagner, R. Nowak, and R. Baraniuk. Distributed image compression for sensor networks using correspondence analysis and super-resolution. In *IEEE International Conference on Image Processing*, pages 597–600, 2003.

[9] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:560–576, 2003.

[10] M. Wu and C. W. Chen. Collaborative image coding and transmission over wireless sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007.