A Spatial Correlation Model for Visual Information in Wireless Multimedia Sensor Networks

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

***Abstract*—Wireless multimedia sensor networks (WMSNs) are interconnected devices that allow retrieving video and audio streams, still images, and scalar data from the environment. In a densely deployed WMSN, there exists correlation among the visual information observed by cameras with overlapped field of views. This paper proposes a novel spatial correlation model for visual information in WMSNs. By studying the sensing model and deployments of cameras, a spatial correlation function is derived to describe the correlation characteristics of visual information observed by cameras with overlapped field of views. The joint effect of multiple correlated cameras is also studied. An entropy-based analytical framework is developed to measure the amount of visual information provided by multiple cameras in the network. Furthermore, according to the proposed correlation function and entropy-based framework, a correlation-based camera selection algorithm is designed. Experimental results show that the proposed spatial correlation function can model the correlation characteristics of visual information in WMSNs through low computation and communication costs. Further simulations show that, given a distortion bound at the sink, the correlation-based camera selection algorithm requires fewer cameras to report to the sink than the random selection algorithm.**

***Index Terms*—Camera selection, spatial correlation, visual information, wireless multimedia sensor networks.**

I. Introduction

WIRELESS multimedia sensor networks (WMSNs) are interconnected devices that allow retrieving video and audio streams, still images and scalar data from the environment [1]. WMSNs are widely used in applications such as video surveillance, environmental monitoring, and industrial process control. Usually, WMSNs should be designed to deliver multimedia content with a certain level of quality-of-service (QoS). Compared with traditional wireless sensor networks that deal with scalar data, WMSNs have more design challenges. The resource constraints of sensors such as energy constraints and limited processing capabilities still exist. Moreover, visual information, the dominating part of multimedia data, requires more sophisticated processing techniques and much higher bandwidth to deliver. Our study will focus on the processing and communication of visual information in WMSNs.

Since uncompressed raw video streams require excessive bandwidth that is impossible to be supported by wireless multihop networks, multimedia source coding must be employed to achieve high compression efficiency. Today's standardized video coding technologies, such as MPEG and H.26x [19], achieve high compression performance at the expense of extensive computation at the encoder. In contrast, distributed video coding [21] allows simple and low power encoder, while the decoder is high power and loaded with extensive computation burden. Thus, distributed video coding is regarded as a promising solution for video coding in WMSNs. Current distributed video coding technologies rely on channel coding to exploit the correlation among adjacent frames [7], [15]. However, it is not easy to attain accurate estimations of the correlation structure among adjacent video frames, resulting in limited encoding efficiency of distributed video coding.

Multimedia source coding is not the only way to reduce the amount of data to be transmitted in the network. In [1], collaborative multimedia in-network processing is suggested as an effective way to avoid the transmission of redundant information. According to the requirements of specific applications, each sensor node can filter out uninteresting events locally, or coordinate with each other to aggregate correlated data. To design filtering and aggregation algorithms for WMSNs, the correlation characteristics of visual information from different sensors need to be studied. In [16], a theoretical spatio-temporal correlation model is developed for scalar data in wireless sensor networks. However, as visual information is much more complex than scalar data, the model in [16] for scalar data cannot be directly applied to visual information.

In many recent research efforts for WMSNs, image processing techniques are utilized to design collaborative processing. In [17], images from correlated views are roughly registered using correspondence analysis. Each sensor transmits a low-resolution version of a common area, and the sink combines multiple low-resolution versions into a high-resolution image. In [20], spatial correlation is obtained by an image shape matching algorithm, while temporal correlation is calculated via background subtraction. Based on spatial and temporal correlation information, images from correlated sensors are transmitted collaboratively. However, the performances of image processing algorithms are application dependent: different types of images will require different processing schemes [8]. Also, image processing techniques are complicated and computation extensive, which will bring about extra computation costs for sensor nodes.

Manuscript received October 01, 2008; revised April 21, 2009. Current version published September 16, 2009. This work was supported by the U.S. National Science Foundation (NSF) under Grant No. ECCS-0701559. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aggelos K. Katsaggelos.

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: 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.2009.2026100

Cameras are directional sensors with limited field of views [12], and the image observed by a camera is directly related to its field of view. In [12], the correlation degree of two cameras is defined as the portion of overlapped sensing area to the entire area of the field of view. A video processing scheme based on correlation is also proposed in [12]: two sensors cooperate with each other, and each sensor transmits a part of its observed image to the sink, and then the sink will combine the partial images together. But this scheme is only valid when the sensing directions of the two sensors do not differ very much. Besides, this processing method is limited between two sensors. How to deal with cooperative processing of more than two sensors is a problem that has not been well investigated.

In this paper, we study the correlation characteristics of visual information in WMSNs. Trying to avoid specific image algorithms, we propose a general correlation model for resource-constrained sensor networks. Our main contributions include the following.

1) We design a novel *spatial correlation function* to describe the correlation characteristics for the images observed by cameras with overlapped field of views.
2) We propose an *entropy-based analytical framework* to evaluate the joint effect of multiple correlated camera nodes.
3) Based on the entropy-based framework, we introduce a *correlation-based camera selection* algorithm.

The remainder of the paper is organized as follows. Section II briefly states the problems to be studied. The spatial correlation function is introduced in Section III. Section IV introduces the entropy-based framework for multiple correlated cameras and the correlation-based camera selection algorithm. Experimental results are presented in Section V, and conclusions are given in Section VI.

II. Problem Statement

In a multimedia sensor network, multiple camera sensors are deployed to provide multiple views, multiple resolutions, and enhanced observations of the environment [4]. Fig. 1 gives an example of a WMSN deployed with cameras. A typical scenario of WMSN is: the application specifies which area it is interested in, and the cameras that can observe this area will transmit their observations to the sink. For a certain area of interest, suppose there are N camera sensors that can observe the area of interest, we denote them as a group $S = \{S_1, S_2, \ldots, S_N\}$, and their observed images as $\{X_1, X_2, \ldots, X_N\}$. There exists correlation among the observations of this group of cameras, which can be exploited to design multimedia in-network processing schemes.

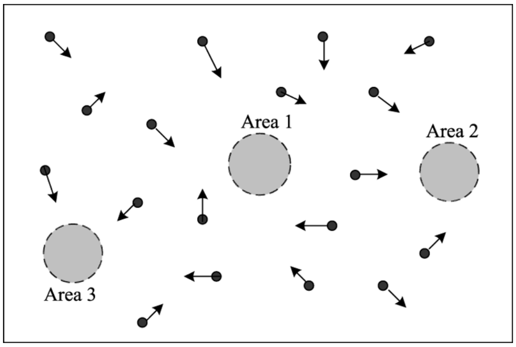


Fig. 1. WMSN with cameras and areas of interest.

A. Spatial Correlation

Firstly, we study the correlation characteristics of the images observed by different cameras. For Camera i and Camera j in the group $S = \{S_1, S_2, \ldots, S_N\}$, we will derive a correlation coefficient ρ_{ij} to describe the degree of correlation between image X_i and image X_j. For the group of camera sensors, the correlation among the images observed by these cameras ($\{X_1, X_2, \ldots, X_N\}$) will be represented as a correlation matrix C, denoted as $C = (\rho_{ij})_{N*N}$, where ρ_{ij} is the correlation coefficient of image X_i and image X_j.

B. Joint Effect of Multiple Correlated Cameras

After we obtain the spatial correlation coefficient, we study the joint effect of multiple correlated cameras in WMSNs. In particular, we study how to measure the amount of visual information from multiple cameras in a WMSN. Intuitively, the visual information provided by multiple cameras should be related to the correlation characteristics of the observed images. If the images observed by these cameras are less correlated, they will provide more information to the sink. We develop an entropy-based framework to estimate the amount of information from multiple correlated cameras.

C. Correlation-Based Camera Selection

Since the delivery of visual information needs very high bandwidth, which may reduce the lifetime of the network, the communication load in WMSNs should be reduced as much as possible. Suppose a total number of N cameras can observe the area of interest, if network resources permit, we can let all these N cameras transmit their observed images to the sink, so that the users at the sink can obtain comprehensive information about the area. However, if the sink/application allows a certain level of distortion of the observations, it may not be necessary for all the cameras to report their observed information to the sink.

Consequently, we define a camera selection problem as follows: if only M cameras $(M \leq N)$ are allowed to transmit their observed images to the sink, how to select M cameras out of the N cameras so that the sink can gain the maximum amount of information. Based on our study on the joint effect of multiple cameras, we design a correlation-based algorithm to select cameras under distortion constraints.

III. Spatial Correlation for Visual Information

A. Sensing Model

Different from scalar data sensors, the sensing of a camera is characterized by directional sensing and 3-D to 2-D projection. In computer vision, this sensing process is usually described by the pinhole camera model [6]. Fig. 2 illustrates an example of a pinhole camera. The camera's center of projection is at the

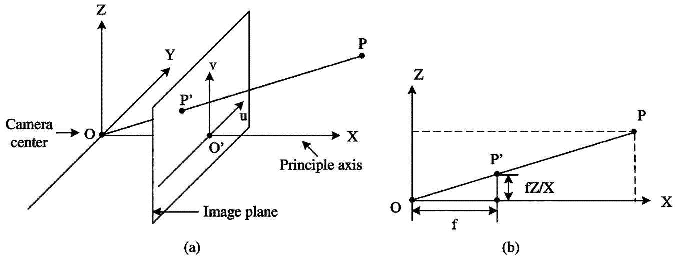


Fig. 2. Camera projection model.

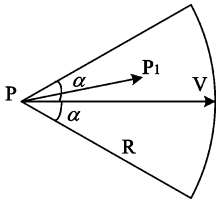


Fig. 3. Field of view.

origin of a Euclidean coordinate system, and its sensing direction is along the x axis. The focal length of the camera is f, so the image plane is the plane $x = f$. A scene point P with coordinates $(x, y, z)^T$ is mapped to $P'(u, v)^T$ on the image plane, where u and v are given by

$$\begin{cases} u = \frac{fy}{x} \\ v = \frac{fz}{x}. \end{cases} \tag{1}$$

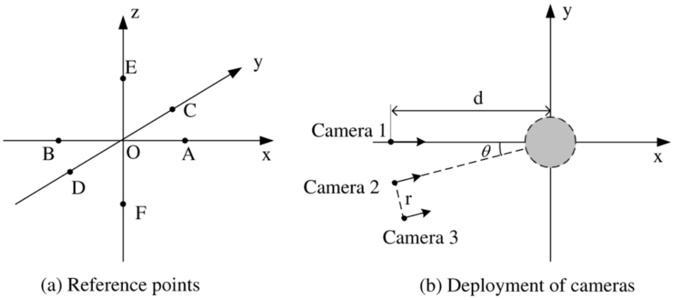
A camera also has limited sensing range. It can only observe the objects within its field of view (FoV). A simplified 2-D FoV model is proposed in [12]: as shown in Fig. 3, a camera's field of view is determined by four parameters $(P, R, \vec{V}, \alpha)$, where P is the location of the camera, R is the sensing radius, $\vec{V}$ is the sensing direction (the center line of sight of the camera's field of view), and α is the offset angle.

A camera's focal length can be estimated by various calibration methods [6]. More recently, several methods have been proposed for the calibration and localization of cameras in sensor networks [2], [5]. Each camera's focal length (f), location (P), and sensing direction $(\vec{V})$ can be estimated as shown in [5]. In the following analysis, we will derive a spatial correlation function based on these parameters.

B. System Model

We set up a world coordinate system $(W) = (O_W, i_W, j_W, k_W)$ for the area of interest as shown in Fig. 4(a), in which the origin is the center of the area of interest, and the XOY plane is the ground plane. Seven reference points, which can also be regarded as *feature points* or key points in a scene, are chosen as: $O(0,0,0)^T$, $A(1,0,0)^T$, $B(-1,0,0)^T$, $C(0,1,0)^T$, $D(0,-1,0)^T$, $E(0,0,1)^T$, $F(0,0,-1)^T$. These reference points form six unit reference vectors along the orthogonal directions in the 3-D world: $\vec{OA}$, $\vec{OB}$, $\vec{OC}$, $\vec{OD}$, $\vec{OE}$, $\vec{OF}$.

We consider the case when all the camera sensors are placed on the ground plane (XOY) and their sensing directions are also within the ground plane. For a camera sensor S_i, the coordinates of its optical center can be denoted as $(x_i, y_i)^T$. The sensing direction of S_i can be described by a unit vector $\phi(\phi_x, \phi_y)$, where $\phi_x = cos\theta$, $\phi_y = sin\theta$, and θ is the angle between the sensing direction and the x axis.



(a) Reference points

(b) Deployment of cameras

Fig. 4. Reference points in the area of interest and deployment of cameras.

The projections of the reference points on a camera will change as the camera's location and sensing direction change. By comparing the projections of the same reference points at different cameras, we can understand the correlation characteristics among different cameras.

C. Projection Geometry

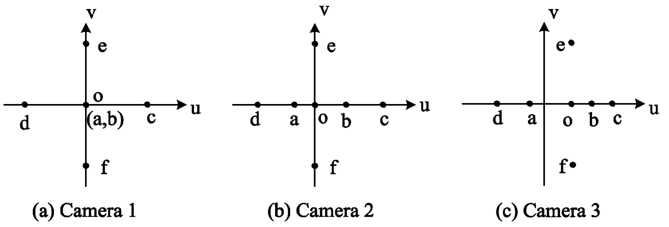
Fig. 4(b) shows the deployment of three cameras in the world coordinate system (W), where the origin is the center of the area of interest, and the XOY plane denotes the ground plane. Camera 1 is located at $(-d, 0)^T$ with its sensing direction along the x axis. Camera 2 is located at $(-d\cos\theta, -d\sin\theta)^T$, and its sensing direction rotates an angle of θ about the x axis. For both Camera 1 and Camera 2, their principle axes pass the center of the area of interest (the origin). Camera 3 has the same sensing direction as Camera 2, but its principle axis does not pass the origin. The distance from the center to its principle axis is r, as shown in Fig. 4(b). The optical center of Camera 3 is $(-d\cos\theta + r\sin\theta, -d\sin\theta - r\cos\theta)^T$. Although the locations and sensing directions of these three cameras are different, the depths for the center of the area of interest in all the three cameras have the same value d. In addition, we assume that all these cameras have the same focal length f.

To calculate the projections of the reference points in a camera, a *coordinate transform* is first needed to obtain the coordinates of the points in the camera's coordinate system. For example, the coordinate system of Camera 1 (Fig. 2) and the world coordinate system (W) are separated by a pure translation. For an arbitrary point P in the space, we have

$$^{1}P =^{W} P +^{1} O_W \tag{2}$$

where ^{1}P is the coordinate vector of point P in the coordinate system of Camera 1, while ^{W}P is the coordinate vector in the world coordinate system (W). These two vectors are related by $^{1}O_W$, the coordinate vector of the origin in (W) seen in the coordinate system of Camera 1. Here $^{1}O_W = (d, 0, 0)^T$.

Therefore, the coordinates of the reference points in the coordinate system of Camera 1 are as follows: $O_1(d,0,0)^T$, $A_1(d+1,0,0)^T$, $B_1(d-1,0,0)^T$, $C_1(d,1,0)^T$, $D_1(d,-1,0)^T$, $E_1(d,0,1)^T$, $F_1(d,0,-1)^T$.



(a) Camera 1 (b) Camera 2 (c) Camera 3

Fig. 5. Projections of reference points and vectors.

TABLE I
PROJECTIONS OF REFERENCE POINTS

Points	Projections		
	Camera 1	Camera 2	Camera 3
O	$(0,0)^T$	$(0,0)^T$	$(\frac{r}{d}f,0)^T$
A	$(0,0)^T$	$(\frac{-sin\theta}{d+cos\theta}f,0)^T$	$(\frac{r-sin\theta}{d+cos\theta}f,0)^T$
B	$(0,0)^T$	$(\frac{sin\theta}{d-cos\theta}f,0)^T$	$(\frac{r+sin\theta}{d-cos\theta}f,0)^T$
C	$(\frac{f}{d},0)^T$	$(\frac{cos\theta}{d+sin\theta}f,0)^T$	$(\frac{r+cos\theta}{d+sin\theta}f,0)^T$
D	$(-\frac{f}{d},0)^T$	$(\frac{-cos\theta}{d-sin\theta}f,0)^T$	$(\frac{r-cos\theta}{d-sin\theta}f,0)^T$
E	$(0,\frac{f}{d})^T$	$(0,\frac{f}{d})^T$	$(\frac{r}{d}f,\frac{f}{d})^T$
F	$(0,-\frac{f}{d})^T$	$(0,-\frac{f}{d})^T$	$(\frac{r}{d}f,-\frac{f}{d})^T$

Based on the projection model in (1), we can find the projections of these reference points in Camera 1: $o_1(0,0)^T$, $a_1(0,0)^T$, $b_1(0,0)^T$, $c_1((f/d),0)^T$, $d_1(-(f/d),0)^T$, $e_1(0,(f/d))^T$, $f_1(0,-(f/d))^T$. The projections of reference points on Camera 1 are plotted in Fig. 5(a).

As for Camera 2, its coordinate system can be derived from the world coordinate system (W) as follows: rotate the world coordinate system counterclockwise for an angle of θ, and then translate the rotated system along the negative direction of x axis for a length of d, given as

$$^2P =^2_W R^W P +^2 O_W \tag{3}$$

where ^2O_W is the translation offset vector, $^2O_W = (d,0,0)^T$, and ^2_WR is the rotation matrix

$$^2_WR = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Similarly, the relationship between the coordinate system of Camera 3 and the world coordinate system is related as

$$^3P =^3_W R^W P +^3 O_W \tag{4}$$

where the rotation matrix is the same as that of Camera 2, $^3_WR = {}^2_WR$, while the translation offset vector satisfies $^3O_W = (d,r,0)^T$.

As the case of Camera 1, the projections of reference points on Camera 2 and Camera 3 can be calculated in the same way. Table I lists the projections of the seven reference points on the three cameras, and Fig. 5 illustrates the positions of reference points on the three cameras. Based on the coordinates of the reference points, the values of the corresponding unit vectors are also calculated, which are listed in Table II.

TABLE II
PROJECTIONS OF REFERENCE VECTORS

Vectors	Projections		
	Camera 1	Camera 2	Camera 3
$\vec{OA}$	$(0,0)^T$	$(\frac{-sin\theta}{d+cos\theta}f,0)^T$	$(\frac{r-sin\theta}{d+cos\theta}f - r\frac{f}{d},0)^T$
$\vec{OB}$	$(0,0)^T$	$(\frac{sin\theta}{d-cos\theta}f,0)^T$	$(\frac{r+sin\theta}{d-cos\theta}f - r\frac{f}{d},0)^T$
$\vec{OC}$	$(\frac{f}{d},0)^T$	$(\frac{cos\theta}{d+sin\theta}f,0)^T$	$(\frac{r+cos\theta}{d+sin\theta}f - r\frac{f}{d},0)^T$
$\vec{OD}$	$(-\frac{f}{d},0)^T$	$(\frac{-cos\theta}{d-sin\theta}f,0)^T$	$(\frac{r-cos\theta}{d-sin\theta}f - r\frac{f}{d},0)^T$
$\vec{OE}$	$(0,\frac{f}{d})^T$	$(0,\frac{f}{d})^T$	$(0,\frac{f}{d})^T$
$\vec{OF}$	$(0,-\frac{f}{d})^T$	$(0,-\frac{f}{d})^T$	$(0,-\frac{f}{d})^T$

D. Study of Correlation

1) Scaling Effect: Comparing the projections of the reference vectors in Table II, we find that the lengths of OA, OB, OC, and OD are different in the three cameras, but the length of OE/OF remains to be a constant value. The reason is that the points O, E, and F have the same depth (d) in all the three cameras, and that the cameras also have the same focal length (f). Both the depth and the focal length can influence the size of a projection. Thus, we define a scaling factor, s, as the lengths of the projections of OE and OF, given by

$$s = \frac{f}{d}. \tag{5}$$

2) Translation Effect: As can be seen in Fig. 5, the projections on Camera 1 and Camera 2 are both in the center of the image planes, but the projections on Camera 3 have an offset from the center of the image plane. The deviation from the center of the area of interest to the camera's principle axis has caused the translation of the projections. Based on the projections of reference points on Camera 3, we define a translation factor as

$$t = r\frac{f}{d} = rs. \tag{6}$$

3) Correlation Coefficient: As shown in Table II, the lengths of vectors OA, OB, OC, and OD will change as the camera's location and sensing direction change. Based on this observation, we design a *disparity function* to reveal the disparity between the projections of reference vectors on different cameras. Suppose that Camera i and Camera j are two arbitrary cameras on the ground plane that can observe the area of interest, the disparity function is derived as follows.

1) Determine the positions and sensing directions of Camera i and Camera j.
2) Based on the projection model in (1), compute the projections of reference vectors in each camera.
3) Divide the projections of reference vectors by the scaling factor $s = (f/d)$ (5), so that we can get a set of *normalized projection vectors* for each camera.
4) Compute the distance for each pair of normalized vectors OA, OB, OC, and OD. For example, if the projection of OA is $o_ia_i = (u_i,v_i)^T$ on Camera i, and $o_ja_j = (u_j,v_j)^T$ on Camera j, the distance is calculated as

$$d_{OA} = \sqrt{(u_i-u_j)^2+(v_i-v_j)^2}. \tag{7}$$

5) The disparity between the images at Camera i and Camera j, denoted by δ, is defined as the average distance of the four vectors:

$$\delta = \frac{1}{4}(d_{OA} + d_{OB} + d_{OC} + d_{OD}). \tag{8}$$

For Camera 1 and Camera 2 in Fig. 4(b), according to the results in Table II, the disparity between the images at Camera 1 and Camera 2 is calculated as

$$\delta = \frac{1}{4}\left(\left|\frac{d\sin\theta}{d+\cos\theta}\right| + \left|\frac{d\sin\theta}{d-\cos\theta}\right| + \left|\frac{d\cos\theta}{d+\sin\theta} - 1\right| + \left|\frac{-d\cos\theta}{d-\sin\theta} + 1\right|\right). \tag{9}$$

Generally, for Camera i and Camera j with position parameters (d_i, r_i, θ_i) and (d_j, r_j, θ_j) [Fig. 4(b)], the disparity between the images at the two cameras is given by

$$\begin{aligned}\delta_{ij} = \frac{1}{4}\Bigg(&\left|\frac{-d_i\sin\theta_i - r_i\cos\theta_i}{d_i+\cos\theta_i} - \frac{-d_j\sin\theta_j - r_j\cos\theta_j}{d_j+\cos\theta_j}\right| \\ &+ \left|\frac{d_i\sin\theta_i + r_i\cos\theta_i}{d_i-\cos\theta_i} - \frac{d_j\sin\theta_j + r_j\cos\theta_j}{d_j-\cos\theta_j}\right| \\ &+ \left|\frac{d_i\cos\theta_i - r_i\sin\theta_i}{d_i+\sin\theta_i} - \frac{d_j\cos\theta_j - r_j\sin\theta_j}{d_j+\sin\theta_j}\right| \\ &+ \left|\frac{-d_i\cos\theta_i + r_i\sin\theta_i}{d_i-\sin\theta_i} - \frac{-d_j\cos\theta_j + r_j\sin\theta_j}{d_j-\sin\theta_j}\right|\Bigg).\end{aligned} \tag{10}$$

We present a simulation to show how the disparity value δ varies as a function of the deployments of cameras. Refer to Fig. 4(b): we let Camera 1 stay fixed, and let the sensing direction of Camera 2 (θ) change from -90° to 90°. The *sensing direction difference* between Camera 2 and Camera 1 is also θ. Set the depth $d = 2.5$ (meters) for Camera 1 and Camera 2. The disparity between Camera 1 and Camera 2 in (9) is illustrated as a function of θ (in degrees) in Fig. 6.

The disparity value increases as the sensing direction difference increases. The larger the disparity value, the more differences exist between the two images, i.e., the images are less correlated. In the above scenario, the largest disparity value goes to 1 when the sensing directions of the two cameras are perpendicular, for which we can say that the two cameras are weakly correlated. For the convenience of further analysis, we bound the disparity value from 0 to 1 as follows:

$$\delta = \min(\delta, 1). \tag{11}$$

Consequently, we can define a *correlation coefficient* that is complementary to the disparity function:

$$\rho = 1 - \delta. \tag{12}$$

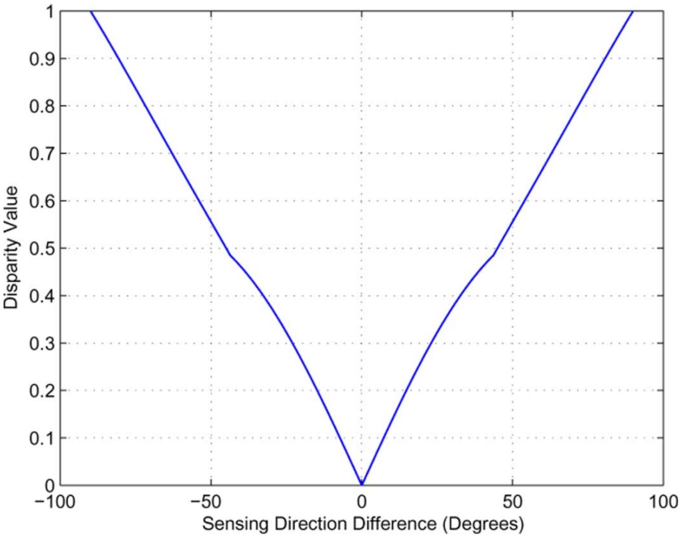


Fig. 6. Illustration of the disparity function.

When the correlation coefficient is 0, it means that the two images are independent of each other. If it equals to 1, the two images are highly correlated. The larger the correlation coefficient, the more correlated are the two images.

E. Discussion

In WMSN applications, as long as the area of interest is specified, and the locations and sensing directions of cameras are estimated, the correlation characteristics of cameras with overlapped field of views can be obtained as introduced above. The proposed correlation model can help to design the differential source coding between cameras as well as the aggregation of visual information in the network.

The proposed model depends on the selection of reference points/vectors in the area of interest. Six unit vectors along three orthogonal directions in the 3-D world are chosen in the above analysis. For a WMSN application, the reference points should be chosen properly based on specific application requirements. In addition, a camera's field of view will be reduced when it is blocked by some obstacles. To guarantee that our model works well, a camera's practical field of view needs to be estimated.

IV. Joint Effect of Multiple Correlated Cameras

In this section, we study the joint effect of multiple correlated cameras. We investigate how to measure the amount of visual information from multiple correlated cameras and then propose a correlation-based camera selection algorithm.

A. Entropy-Based Approach

In information theory [3], the concept of entropy is used to measure the amount of information of a random source. If an image is interpreted as a sample of a "gray-level source", the source's symbol probabilities can be modeled by the gray-level histogram of the observed image. An estimate of the source's entropy can be generated as [8]

$$\widetilde{H} = -\sum_{k=1}^{L} p(r_k) \log p(r_k) \tag{13}$$

where L is the number of all possible gray-levels, and $p(r_k)$ is the probability of the kth gray-level. It denotes the average amount of information per pixel in the image.

If a camera S_i transmits its observed image X_i to the sink, the amount of information gained at the sink is $H(X_i)$. (We do not consider the information loss caused by lossy compression or packet loss during transmission.) If the group of camera sensors, $S = \{S_1, S_2, \ldots, S_N\}$, transmit their observed images $\{X_1, X_2, \ldots, X_N\}$ to the sink, the amount of information gained at the sink will be the joint entropy $H(X_1, X_2, \ldots, X_N)$. Our objective is to estimate the joint entropy of multiple cameras.

B. Joint Entropy of Two Cameras

We consider two cameras that can observe the area of interest. Suppose each camera has captured one image about the area of interest, denoted as image A and image B. The joint entropy of A and B is

$$H(A, B) = H(A) + H(B) - I(A; B) \tag{14}$$

where $I(A; B)$ is the mutual information of the two sources. $I(A; B)$ can be interpreted as the reduction in the uncertainty of one source due to the knowledge of the other source:

$$I(A; B) = H(A) - H(A|B) = H(B) - H(B|A). \tag{15}$$

The definition of $I(A; B)$ in probability form is given as

$$I(A; B) = \sum_a \sum_b p(a, b) \log \frac{p(a, b)}{p(a)p(b)} \tag{16}$$

where $p(a)$ and $p(b)$ are the probability distributions of the pixels in image A and image B, and $p(a, b)$ is the joint probability distribution of the two sources.

Mutual information is a measure of dependence between two sources: the more A and B are correlated, the larger the mutual information $I(A; B)$.

In [14], a normalized form of mutual information, *entropy correlation coefficient (ECC)*, is defined as

$$ECC = \frac{2I(A; B)}{H(A) + H(B)}. \tag{17}$$

The entropy correlation coefficient (ECC) ranges from zero to one, where zero indicates that source A and B are independent, while one indicates that source A equals to source B. The larger the ECC value, the more these two sources are correlated.

Based on (14) and (17), the joint entropy of A and B can be expressed as a function of $H(A)$, $H(B)$, and ECC:

$$H(A, B) = \left(1 - \frac{1}{2} ECC\right) (H(A) + H(B)). \tag{18}$$

Since $H(A)$ and $H(B)$ can be calculated at each camera using (13), if ECC can be estimated, the joint entropy $H(A, B)$ will be obtained. However, to calculate $I(A; B)$ and ECC, a joint probability distribution of the two sources needs to be estimated (16). 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 [14]. Besides, estimating the joint probability also requires large bulk of computation [14]. If joint probability distribution is to be estimated in a sensor network, cameras at different locations must exchange their observed images, which will introduce a lot of communication burden in the network.

It can be seen that the proposed correlation coefficient in (12) has the same intrinsic meaning as ECC: both ranging from 0 to 1 and denoting the degree of correlation between two sources. However, if cameras' parameters and deployment information are given, it is much easier to obtain the proposed correlation coefficient. Considering the limited processing capability of sensors, we propose to estimate ECC by the proposed correlation coefficient. If we replace ECC in (18) by the proposed correlation coefficient ρ, we can obtain an estimation of the joint entropy of A and B as

$$H(A, B) \approx \left(1 - \frac{1}{2} \rho\right) (H(A) + H(B)). \tag{19}$$

Therefore, the amount of information that can be gained from image A and image B together depends on the correlation degree between A and B. The more A and B are correlated, the less joint entropy can be gained from A and B together. That is to say, if two camera sensors transmit their images to the sink, the amount of information gained at the sink will be larger if the two sensors are less correlated.

C. Joint Entropy of Multiple Cameras

In this section, we extend our study of joint entropy to the case of more than two cameras. Suppose there is a group of camera sensors $S = \{S_1, S_2, \ldots, S_N\}$ with their observed images $\{X_1, X_2, \ldots, X_N\}$. We are interested in estimating the joint entropy $H(X_1, X_2, \ldots, X_N)$ for this group of sensors. If $H(X_1, X_2, \ldots, X_N)$ is to be computed by its definition in probability, the joint probability distribution of these N images needs to be estimated. However, it is difficult to estimate the joint probability distribution of multiple sources, especially when N is large.

A feasible approach is to make use of the joint entropy of two cameras in the last section. As there are N individual elements in the group $\{X_1, X_2, \ldots, X_N\}$, we can merge two of them together, so that the joint entropy of these two elements can be calculated by (19). We treat these two elements as a whole element, then the number of elements in the group reduces to

$N-1$. If we repeat this process, the N individual sensors will be combined into a single element in the end. As the joint entropy of merged sensors are calculated along the merging process, the joint entropy $H(X_1, X_2, \ldots, X_N)$ can be obtained when the merging process is completed.

We design an algorithm to estimate the joint entropy of multiple cameras based on the idea of hierarchical clustering [10]. As long as the entropy of each single image $(H(X_i), i = 1, 2, \ldots, N)$ and the correlation matrix $(C = (\rho_{ij})_{N*N})$ are given, the joint entropy $H(X_1, X_2, \ldots, X_N)$ can be estimated through the hierarchical clustering process. The details of the estimation algorithm are presented in *Algorithm 1*, where χ denotes the set of clusters, and $\rho(\{X_i\}, \{X_j\})$ is the correlation coefficient between cluster $\{X_i\}$ and cluster $\{X_j\}$.

Algorithm 1: Estimate the Joint Entropy of Multiple Cameras

```
$H(X_1, X_2, \ldots, X_N) = \mathrm{JointEntropy}(H(X_i), (\rho_{ij})_{N*N})$
begin
$\chi = \{\{X_1\}, \{X_2\}, \ldots, \{X_N\}\}$, $\rho(\{X_i\}, \{X_j\}) = \rho_{ij}$.
for $k = 1$ to $N-1$ do
    Find $(\{X_i\}, \{X_j\}) = \arg\max_{\{X_i\},\{X_j\} \in \chi} \{\rho(\{X_i\}, \{X_j\})\}$
    { Find the most correlated pair of clusters in $\chi$. }
    Merge $\{X_i\}$ and $\{X_j\}$ into a new cluster $\{X_{N+k}\}$.
    $H(X_{N+k}) = H(X_i, X_j)$ (19).
    for $X_l \in \chi$, $l \neq i$, $l \neq j$ do
    Compute $\rho(\{X_{N+k}\}, \{X_l\})$. $(*)$
    end for
    Remove $\{X_i\}$ and $\{X_j\}$ from $\chi$; Add the new cluster
    $\{X_{N+k}\}$ into $\chi$.
end for
$H(X_1, X_2, \ldots, X_N) = H(X_{2N-1})$
return $H(X_1, X_2, \ldots, X_N)$
end
```

In step $(*)$ of *Algorithm 1*, the correlation coefficient between one cluster and another cluster can be obtained by the *greatest/shortest/average* correlation coefficient from any member of one cluster to any member of the other cluster [10], which are referred to as *single-linkage/complete-linkage/average-linkage* clustering.

The following is an example of the estimation of joint entropy. Suppose there is a group of five camera sensors. Without loss of generality, we assume that the entropy of a single image is a constant value, denoted as $H(X_i) = H(\cdot)(i = 1, \ldots, 5)$. A correlation matrix for these five sensors is given by

$$(\rho_{ij})_{5*5} = \begin{pmatrix} 1 & 0 & 0.2942 & 0.9443 & 0 \\ & 1 & 0 & 0 & 0.7359 \\ & & 1 & 0.3416 & 0 \\ & & & 1 & 0 \\ & & & & 1 \end{pmatrix}. \quad (20)$$

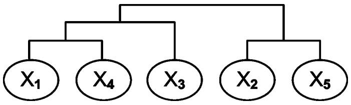


Fig. 7. Example of hierarchical clustering.

TABLE III
HIERARCHICAL CLUSTERING STEPS

Steps	Nodes for Clustering	Estimation of joint entropy (Relative value to $H(\cdot)$)
1	$\{X_1\}$,$\{X_4\}$	$H(X_1, X_4) = 1.0557$
2	$\{X_2\}$,$\{X_5\}$	$H(X_2, X_5) = 1.2641$
3	$\{X_1 X_4\}$,$\{X_3\}$	$H(X_1, X_3, X_4) = 1.7290$
4	$\{X_1 X_3 X_4\}$,$\{X_2 X_5\}$	$H(X_1, X_2, X_3, X_4, X_5) = 2.9931$

Apply *Algorithm 1* to this group of sensors, and use the *average-linkage clustering* [10] metric in step $(*)$. The clustering process is illustrated in Fig. 7, and the results in each step of clustering are shown in Table III. Comparing the values in the correlation matrix (20) and the clustering steps in Fig. 7, one can find that in every clustering step, nodes that contain the most correlated images are merged into one cluster. As can be seen from (19), the value of the joint entropy decreases as the correlation degree of the two images increases. Therefore, the joint entropies obtained from the clustering process are always relatively small. The final result of the estimation algorithm is a conservative estimation of joint entropy.

D. Correlation-Based Camera Selection

Suppose for an area of interest in a WMSN, a total number of N cameras can observe the area of interest. If network resources permit, all these cameras can transmit their observed images to the sink, so that the applications at the sink can gain comprehensive information about the area. However, as the processing capabilities of sensors are limited, and the communication among sensors causes huge energy consumption, sometimes the network cannot support all these cameras to report their observations to the sink. Consequently, we define a camera selection problem: if only M cameras $(M \leq N)$ are allowed to transmit their observed images to the sink, how to select M cameras out of the N cameras so that the sink can gain the maximum amount of information.

As in the last section, we also assume that the entropy of a single image is a constant value here. The estimation of joint entropy in (19) indicates that the less correlated are the two sensors, the more information can be provided by the two sensors together. Thus, to maximize the joint entropy of M cameras, we should try to minimize the correlation among the cameras to

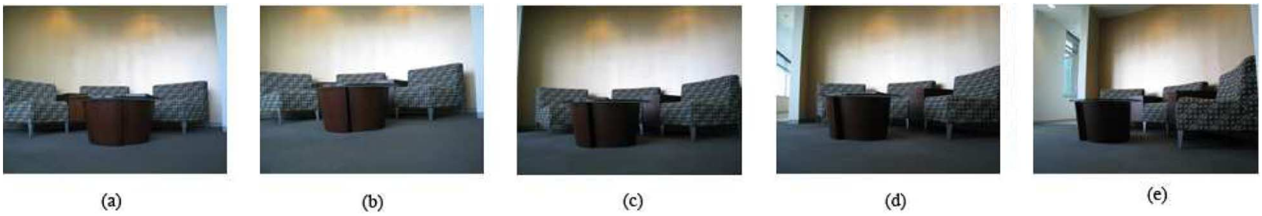


Fig. 8. Multiple view images. (a) Reference Image. (b) $\theta = 15°$. (c) $\theta = 30°$. (d) $\theta = 45°$. (e) $\theta = 60°$.

be selected. We propose a *correlation-based* algorithm to maximize the joint entropy of M cameras. At each step of the algorithm, we select one camera that is least correlated with the cameras that have already been selected. The details are presented in *Algorithm 2*, where $\chi = \{X_1, X_2, \ldots, X_N\}$ is the set of images observed by these N cameras, and S denotes the set of cameras that are already selected.

Algorithm 2: Correlation-Based Selection of Cameras

```
$S = \mathrm{CorrSelection}(\{X_1, X_2, \ldots, X_N\}, (\rho_{ij})_{N*N}, M)$
**begin**
$S = \varnothing, \chi = \{X_1, X_2, \ldots, X_N\}$, $\rho(X_i, X_j) = \rho_{ij}$.
Find $(X_i, X_j) = \arg \min_{X_i, X_j \in \chi} \{\rho(X_i, X_j)\}$. { Find the least correlated pair of cameras. }
Add the corresponding $X_i$ and $X_j$ into $S$. $\{M = 2\}$
**if** $M > 2$ **then**
  **for** $k = 1$ to $M - 2$
    **for** $X_l \in \chi, X_l \notin S$ **do**
      $\rho(X_l, S) = \max_{X_j \in S}\{\rho(X_l, X_j)\}$.
    **end for**
    $X_m = \arg \min_{X_m \in \chi, X_m \notin S}\{\rho(X_m, S)\}$; Add $X_m$ into $S$.
  **end for**
**end if**
**return** $S = \{X_{i1}, X_{i2}, \ldots, X_{iM}\}$
**end**
```

E. Distortion Function

For an area of interest in WMSN, we suppose a total number of N cameras can observe it, and denote their observed images as $\{X_1, X_2, \ldots, X_N\}$. The joint entropy of all these N sensors, $H(X_1, X_2, \ldots, X_N)$, is the maximum amount of information that can be gained for the area of interest. If a subset of these sensors, denoted as $\{X_{i1}, X_{i2}, \ldots, X_{iM}\}$, is selected to report to the sink, the information gained at the sink is $H(X_{i1}, X_{i2}, \ldots, X_{iM})$.

We define a distortion function as the ratio of the decrease in the amount of information to the maximum amount of information, given by

$$D = \frac{H(X_1, X_2, \ldots, X_N) - H(X_{i1}, X_{i2}, \ldots, X_{iM})}{H(X_1, X_2, \ldots, X_N)}. \quad (21)$$

The value of D satisfies $0 \leq D \leq 1$. It can be interpreted as the percentage of information loss due to network resource constraints. Applications of WMSNs can use this distortion function as a metric to describe their requirements. For example, an application may ask the network to transmit information within 10% or 20% of information loss.

It should be emphasized that the proposed distortion function is different from existing image/video quality metrics. Commonly used image quality metrics, such as peak signal-to-noise ratio (PSNR) and the recently developed structural similarity (SSIM) [18], are designed to evaluate the degradation of a distorted image compared to an original image, where distortion is caused by lossy compression or loss during transmission.

However, our proposed distortion function is designed to evaluate the joint effect of multiple images. Distortion is the percentage of information loss caused by reporting a subset of images to the sink. According to the derivation of joint entropy in *Algorithm 1*, we can find that the value of distortion is related to the number of selected cameras as well as the correlation among the selected cameras.

V. Performance Evaluation

A. Spatial Correlation Coefficient

In this section, we present a set of experiments to evaluate the performance of our spatial correlation model.

1) Validity of the Proposed Spatial Correlation Coefficient: We set up a scene as shown in Fig. 4(b): Camera 1 and Camera 2 are placed to take pictures of an area of interest. Camera 1 is placed along the x axis, and Camera 2 rotates an angle of θ, so the *sensing direction difference* between Camera 2 and Camera 1 is θ. Set $d = 2.5\ (\mathrm{meters})$. A reference image is obtained at Camera 1, and then a group of ten images are taken for Camera 2 with the following θ values: $\{-75°, -60°, -45°, -30°, -15°, 15°, 30°, 45°, 60°, 75°\}$. Fig. 8 presents some of the images.

In Section III, we showed that the degree of correlation is relevant to cameras' sensing directions and their relative positions. Since the sensing directions and positions are already known, the disparity between the test images on Camera 2 and

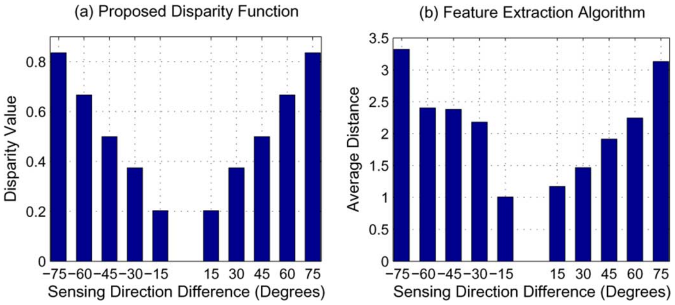


Fig. 9. Proposed disparity function versus feature extraction algorithm.

Camera 1 can be easily calculated by function (10). The results of the disparity values are presented in Fig. 9(a) as a function of the *sensing direction difference* θ. The disparity increases as the sensing direction difference increases.

The correlation between images can also be obtained by applying image processing algorithms. Here, we refer to a commonly used feature extraction algorithm in [13]. In this algorithm, texture features are extracted from the images by Gabor wavelet transform, and based on the wavelet coefficients, feature vectors in multiple scales and multiple directions are constructed. Finally, an average distance is calculated by averaging all the feature distances in each feature space [13]. The average feature distances between the test images on Camera 2 and Camera 1 are calculated accordingly, and the results are presented in Fig. 9(b).

Comparing the results of the proposed model [Fig. 9(a)] and the results of the feature extraction algorithm in [13] [Fig. 9(b)], we find that in both cases, the disparity/distance value increases as the sensing direction difference increases. This is also in accordance with our common sense: if we just observe the test images in Fig. 8 with our eyes, we can also find that the two images from Camera 2 and Camera 1 look more different when their sensing direction difference (θ) is larger. Therefore, the proposed spatial correlation coefficient is effective as it can reveal the correlation characteristics between images.

The slight differences between the results in Fig. 9(a) and (b) may be explained by the intrinsic differences of the two schemes. The proposed model is derived by studying cameras' sensing model and deployments; thus, the results are just dependent on a few parameters. In contrast, the feature extraction algorithm goes into details in an image. It is sensitive to the noise in the images, and even a little change of the light condition might influence the final results.

2) Costs for Exploiting Correlation: In this section, we discuss about the costs for exploiting correlation in WMSNs. As shown in the example of WMSN in Fig. 1, a typical scenario of WMSN application is: the application specifies which area it is interested in, and the cameras that can observe this area will work together to provide enhanced observations for the application. Given an area of interest, suppose a group of N cameras can observe it. If the cameras in this group want to know their correlation characteristics with each other, communication and computation operations are needed for these camera sensors. Note that it is a repetitive process to exploit correlation in the network: the correlation characteristics are obtained with respect to a certain area of interest; therefore, once the application specifies for a different area of interest, the correlation characteristics need to be investigated again.

We study a single hop case between two cameras, so that the results will be independent of specific communication protocols and network topologies. Assume two arbitrary cameras in the group, Camera i and Camera j, are within the transmission range of each other, and they will cooperate with each other to obtain the correlation of their observed images.

The proposed correlation model is derived based on the sensing model and deployment information of camera sensors. In most sensor networks, localization algorithms are already implemented, so that each camera knows its position in the network. The focal length and sensing direction for each camera can be estimated [5] and recorded in the deployment stage of the network. Thus, when the application specifies a certain area of interest, each camera can easily figure out its position with respect to the area of interest [d, r, and θ as shown in Fig. 4(b)].

To calculate the correlation between Camera i and Camera j, Camera i just needs to transmit its four parameters to Camera j: d, r, θ as shown in Fig. 4(b), and its focal length f. Once Camera j receives the four parameters, it can calculate the correlation coefficient based on (12). The total energy consumption will be composed of the energy consumption for transmitting and receiving the four parameters and the energy consumption to calculate the correlation coefficient. It can be seen that the energy consumption for the proposed model is independent of image sizes.

We take the commonly used feature extraction scheme in [13] as a representative of the various image processing schemes. As we have introduced above, the feature extraction scheme [13] implements Gabor wavelets to extract features vectors from multiple scales and multiple resolutions. If this scheme is implemented in sensor networks, Camera i will need to exchange its extracted features with Camera j to obtain the correlation degree between Camera i and Camera j. A typical process is as follows: Camera i extracts the features of its observed image using Gabor wavelet, and transmits its feature vectors; Camera j receives the feature vectors from Camera i, and also implement the Gabor wavelet to extract the features of its own image. Finally, the correlation of images at Camera i and Camera j can be calculated by comparing their feature vectors.

The proposed scheme needs to transmit four parameters. As each parameter needs 32 bits to present, the total bits for transmission is 4×32 bits. In the feature extraction scheme [13], features are extracted from four resolutions and six orientations, and each feature space contains two elements. We also assume that each element in the feature space needs 32 bits to present, so the total bits for transmission is $4 \times 6 \times 2 \times 32$ bits.

According to the energy model for communications in [9], we can calculate the energy consumption for communication between Camera i and Camera j. The average energy consumption for communication to exploit correlation is illustrated in Fig. 10. Fig. 10(a) is a comparison of energy consumption for both schemes, and Fig. 10(b) is the energy consumption per node for the proposed scheme. For both schemes, the energy consumption per node increases as the distance between the two

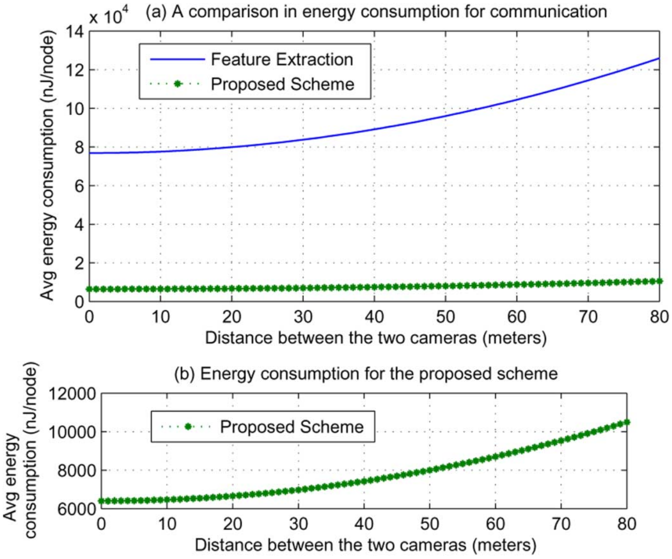


Fig. 10. Average energy consumption for communication per node.

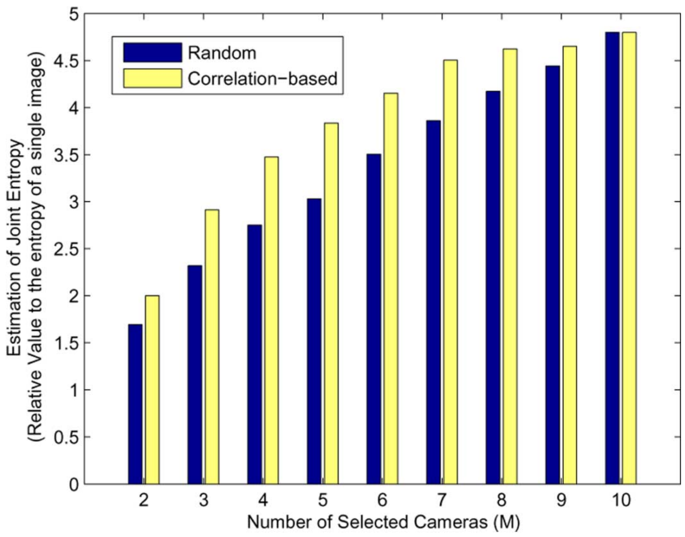


Fig. 11. Estimation of joint entropy.

nodes increases. But the proposed correlation model requires much less energy for communication than the feature extraction scheme.

It is commonly believed that communication is the most energy consuming operation for sensors, which requires much more energy than processing; however, due to the complexity of processing algorithms for visual information, the energy consumption for processing visual information is not negligible. The feature extraction algorithm [13] depends on wavelet transform that makes the energy consumption for computation comparable to communication energy dissipation [11]. Moreover, as image processing schemes are usually implemented in the unit of pixels, the energy consumption for processing is proportional to the size/resolution of the observed image. When the resolution increases, the energy consumption of the image processing-based scheme will increase accordingly. In contrast, the computation process for the proposed model is very simple and straightforward, and the energy consumption for computation will not be influenced by image resolutions.

We have evaluated the validity as well as the costs of the proposed spatial correlation coefficient. From the above experimental results, we can conclude that the proposed correlation coefficient can effectively model the correlation characteristics of visual information through low computation and communication costs.

B. Joint Effect of Multiple Cameras

In this section, we present a set of simulations to evaluate the joint effect of multiple cameras and the correlation-based camera selection algorithm. In a field of 500 m $*$ 500 m, we set an area of interest that is located in the center of the field and has a radius of 10 m. We randomly deploy N cameras that can observe this area of interest. Let M be the number of cameras to be selected by the sink to transmit their observed images. Suppose each camera obtains one image about the area of interest. Let $\{X_1, X_2, \ldots, X_N\}$ denote the images observed by these N cameras, and let $\{X_{i1}, X_{i2}, \ldots, X_{iM}\}$ denote the images observed by the M selected cameras.

Without loss of generality, we assume that the entropy of a single image is a constant value, denoted as $H(X_i) = H(\cdot)(i = 1, 2, \ldots, N)$. For these N cameras, we can obtain a correlation matrix $(\rho_{ij})_{N*N}$ as introduced in Section III. So the joint entropy of $H(X_1, X_2, \ldots, X_N)$ and $H(X_{i1}, X_{i2}, \ldots, X_{iM})$ can be estimated using *Algorithm 1*.

We compare the following two camera selection schemes.

1) Random Selection: Randomly select M cameras out of the N cameras. For each M, repeat the experiment for 50 times. Compute the joint entropy at each time, and take the average value of the 50 trials as the final joint entropy.

2) Correlation-Based Selection: This is the proposed method described in *Algorithm 2*. It makes use of correlation by selecting a group of M cameras that are least correlated with each other, so that the amount of information from the selected cameras can be maximized.

In our first experiment, we randomly deploy ten cameras in the field $(N = 10)$, and let M change from 2 to 10. The results of both schemes are shown in Fig. 11. The value of joint entropy increases as the number of nodes increases, which indicates that if more cameras transmit their observed images to the sink, more information can be gained about the area of interest at the sink. When $M = 10$, all the cameras are selected to transmit their observed images, so both schemes produce the same results. But for $M = 2$ to 9, the correlation-based algorithm always results in larger joint entropy than the random selection of cameras.

According to the numerical results, when the number of selected cameras are the same for these two schemes, the correlation-based algorithm can increase the joint entropy by $0.5466 * H(\cdot)$ in average (increase by 18.37% in average compared to the random selection algorithm). It should be noted that the values of joint entropy in our simulation are expressed as relative values to the entropy of a single image, $H(\cdot)$. We find in our experiments that a typical value of $H(\cdot)$ is 5–6 bits/pixel for images of 8-bits depth, so the correlation-based scheme can result in

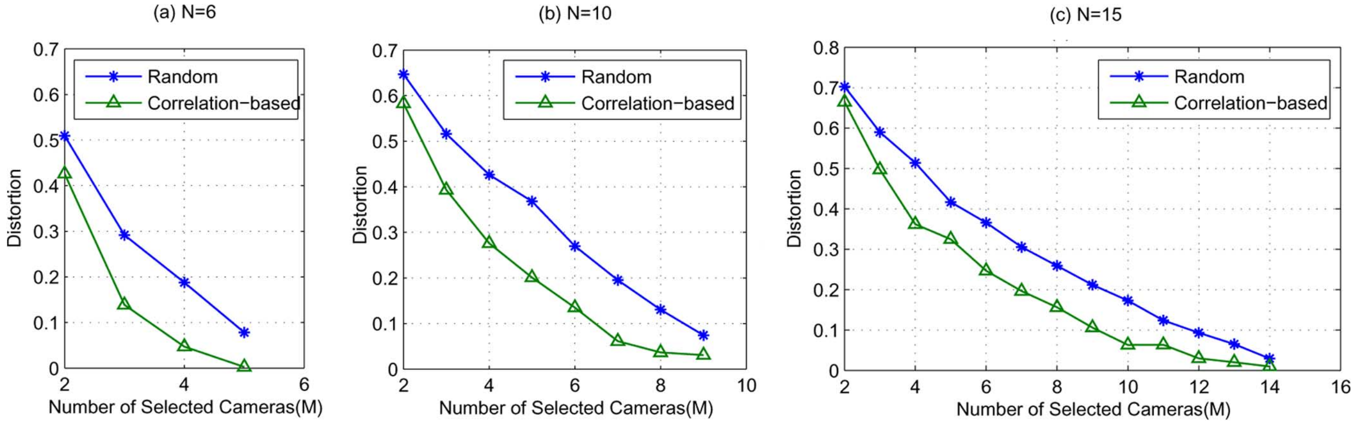


Fig. 12. Distortion function.

about 3 bits/pixel increase in joint entropy than the random selection scheme.

Next we introduce more simulations to evaluate the distortion performance of both schemes. We implement both schemes for three different network topologies, where the total number of cameras, N, equals to 6, 10, and 15, respectively. Fig. 12 plots the distortion performance of both schemes. The distortion decreases as the number of selected nodes increases. For the same number of selected cameras (M), the proposed correlation-based scheme results in lower distortion compared to the random selection scheme.

From another perspective, if a certain distortion bound is required at the sink, we may need fewer cameras to transmit their information using the correlation-based selection scheme. For example, in Fig. 12(b), a total number of ten cameras are deployed to observe an area of interest. If the sink wants to obtain 80% of the total information, the maximum distortion is 0.2. As shown in Fig. 12(b), seven cameras are needed on average when cameras are randomly selected, but only five cameras are needed when the correlation-based selection scheme is used. Therefore, given a distortion bound at the sink, the correlation-based selection scheme requires fewer cameras to report to the sink than the random selection scheme.

VI. Conclusions and Future Work

The correlation characteristics of visual information in WMSNs can be exploited to design multimedia in-network processing schemes. By studying the sensing model and deployment of cameras in the network, we propose a novel spatial correlation function to describe the degree of correlation for the images observed by cameras with overlapped field of views. Extentions of this work will include the study of correlation in the temporal domain for multimedia streaming applications.

In addition, we study the joint effect of multiple correlated cameras in WMSNs. We propose an entropy-based analytical framework to measure the amount of information provided by multiple cameras. We find that the joint entropy of multiple correlated cameras is related to the correlation degrees among the cameras. The entropy-based framework will serve as a useful tool for designing multimedia in-network processing schemes in WMSNs.

A correlation-based camera selection algorithm is also introduced to show how to select cameras from sensor networks under distortion constraints. The proposed camera selection algorithm aims to minimize the number of cameras to report to the sink under distortion constraints. In our future work, we can consider more factors for the camera selection problem, such as the residual energy of sensors, the locations of correlated cameras, the costs of communication between correlated cameras, and the costs for encoding correlated images.

Acknowledgment

The authors would like to thank E. Ekici, C. Ma, J. Zhang, and P. Wang for their valuable suggestions.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Comput. Netw. (Elsevier)*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[2] A. Barton-Sweeney, D. Lymberopoulos, and A. Savvides, “Sensor localization and camera calibration in distributed camera sensor networks,” in *Proc. Int. Conf. Broadband Communications, Networks and Systems*, Oct. 2006, pp. 1–10.

[3] T. Cover and J. Thomas*, Elements of Information Theory*. New York: Wiley, 1991.

[4] R. Cucchiara, “Multimedia surveillance systems,” in *Proc. ACM Int. Workshop Video Surveillance and Sensor Networks*, Nov. 2005, pp. 3–10.

[5] D. Devarajan, Z. Cheng, and R. J. Radke, “Calibrating distributed camera networks,” *Proc. IEEE*, vol. 96, no. 10, pp. 1625–1639, Oct. 2008.

[6] D. A. Forsyth and J. Ponce*, Computer Vision: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[7] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[8] R. C. Gonzalez, R. E. Woods, and S. L. Eddins*, Digital Image Processing Using MATLAB*. Englewood Cliffs, NJ: Prentice-Hall, 2004.

[9] W. Heinzelman, “Application-specific protocol architectures for wireless networks,” Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, 2000.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[11] Q. Lu, W. Luo, J. Wang, and B. Chen, “Low-complexity and energy efficient image compression scheme for wireless sensor networks,” *Comput. Netw. (Elsevier)*, vol. 52, no. 13, pp. 2594–2603, Sep. 2008.

[12] H. Ma and Y. Liu, “Correlation based video processing in video sensor networks,” in *IEEE Int. Conf. Wireless Networks, Communications and Mobile Computing*, Jun. 2005, vol. 2, pp. 987–992.

[13] B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

[14] 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.

[15] 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.

[16] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: Theory and applications for wireless sensor networks," *Comput. Netw. (Elsevier)*, vol. 45, no. 3, pp. 245–259, Jun. 2004.

[17] R. Wagner, R. Nowak, and R. Baraniuk, "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *IEEE Int. Conf. Image Processing (ICIP 2003)*, Sep. 2003, vol. 1, pp. 597–600.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[19] 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.

[20] M. Wu and C. W. Chen, "Collaborative image coding and transmission over wireless sensor networks," *EURASIP J. Adv. Signal Process.*, vol. 2007, 2007, Article ID 70481.

[21] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004.

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, GA.

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, Georgia Institute of Technology, Atlanta, GA. 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 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.