Design and Evaluation of Correlation-Aware Scheduling for Wireless Surveillance Cameras

Chang-Yu Song and Hung-Yun Hsieh
Graduate Institute of Communication Engineering &
Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan 106

***Abstract*—In this paper, we exploit the overhearing source coding in the wireless surveillance cameras networks. Since the image collected from cameras deployed around a small area is often correlated, we can reduce the required encoded bits of the network by using a dependent source coding scheme. However, in order to serve the network in a more efficient way, we formulate a transmission scheduling problem in this paper. Our problem is for choosing which camera as a former transmitter so that the later scheduled cameras can reference from it to reduce their encoded bits. Experiment results show that the total encoded bits needed for collecting data can be reduced under this coding scheme, and thus motivate further investigation in this topic.**

I. Introduction

Many machine-to-machine (M2M) applications are characterized by a large amount of data to transport over a rather limited amount of wireless resources. While for some applications the amount of data produced by each machine is not huge and the delay sensitivity is rather low, for applications such as multimedia surveillance networks the requirement on communication is demanding. Fortunately, since such wireless surveillance cameras are typically deployed with overlapping view angles, the images or videos captured by individual cameras exhibit correlation that can potentially be leveraged for bandwidth-efficient reporting of the collected data.

In this paper, we investigate the problem of correlated data gathering from a set of cameras deployed in a city. It is required that cameras periodically send back the collected images back to the aggregator (e.g. base station) through direct wireless communications (e.g. LTE or WiMAX). Since there might be multiple cameras deployed in a neighborhood area to provide different perspectives of the area, we exploit the capability of *transmission overhearing* among cameras. If a camera can overhear transmissions from nearby cameras, it can reference the image (e.g. as an I-frame) and perform *dependent coding* to reduce the amount of bits required to encode its image (e.g. as a P-frame). Clearly, if the reference image is highly correlated with the target image, the compression ratio will be high. We propose a *correlation-aware scheduling algorithm* to determine the order of transmissions for all cameras based on their locations and the correlation of collected images. To evaluate the proposed algorithm, we resort to a 3D modeling software to generate quasi-realistic city views for all cameras and use H.264 MVC reference software to encode collected images. Evaluation results show the proposed scheduling algorithm to outperform baseline approaches, motivating further investigation along this direction.

II. Related Work

Recently, visual sensor networks (VSN) have caught much attention in different research areas. One of the most challenging research topics in VSN is the data transmission issue, which is very different from the conventional scalar sensor network [1]. For example, a surveillance system might have a huge amount of radio resource demand to provide the image/video transmission over a wireless communication link. The authors in [2] motivated the idea of intelligent vehicle surveillance system for tracking multiple vehicles. In their model, multiveiw cameras are installed to track vehicles from different angle of view. Therefore, in order to operate their system in real-time, there might have some redundancy among the video transmission and hence motivates us to study how to reduce the radio resource usage.

One way to exploit the redundancy among multiple surveillance cameras is through a clustered summary, which is able to browse and search the video in an efficient way [3]. The authors in [3] argued that regular browsing the surveillance videos is impossible, since the surveillance videos are often endless. Therefore, they improved the browsing efficiency by clustering similar activities into a shorter video summary. However, their work focused on the user space efficiency and did not mention how to collect such huge amount of videos through the wireless communication links. In work [4], the authors reduced the amount of image data that needs to be transmit over the network by selecting part of the cameras to report their information. The selection is based on each cameras' observation and the goal of selection is to figure out the views that contribute most significantly to the desire observation. Our work, on the other hand, required all cameras in the network to transmit its data while those cameras can reduce their own radio resource usage by overhearing others' information. The authors in [5] and [6] showed the possibility to use a *Virtual Video* as the test bed of a surveillance systems. They collected image data from virtual video and used those image for further investigation. Therefore, in our work, we also use a 3D modeling software to generate quasi-realistic city views and try to reduce the transmission of those views.

Spatial correlation between cameras gives the surveillance networks leverage to reduce the total amount of encoded bits for transmission. The authors in [7] proposed a spatial correlation model for cameras deployed in a neighborhood area. In their model, the correlation of two cameras are determined by their location and the difference of their sensing direction. However, the camera views are more complicated so that the correlation cannot be determined only by its geometric characteristic. Hence, in order to provide a more realistic model, the work in [8] gives an investigation of the relation among cameras through a multiview video coding software. They analyzed the performance of H.264 multiview video coding of multiple cameras and showed that the coding cost of two cameras raises as their angular difference become larger.

One way to deal with the resource reduction problem is through the distributed video coding [9], which is based on the Slepian-Wolf's and Wyner-Ziv's coding theorem. Distributed video coding is different from the conventional video coding scheme at both the encoder and decoder. More specifically, it suggests an encoder encodes individual frames independently, but the decoder decodes them jointly, which means that the previous frames are used as a side information at the decoder only. The authors in [10] extended the idea of distributed video coding into multiview systems with spatial correlation among cameras. In their work, side information can be generated by exploiting the spatial correlation and redundancies between different camera's views while the decoder has the complexity to decode those frames jointly. Our work, on the other hand, does not require the decoder to have so much complexity. We focus on an overhearing source coding scheme. That is, each cameras has the possibility to overhear other cameras' frame so that it can provide inter-frame processing to reduce its encoded bits.

The work in [11] considered overhearing in the wireless multimedia sensor networks. They made use of the model proposed in [7] and solved a relaxed integer programming problem to determine a proper schedule for reducing the most encoded bits. In this paper, we define a more realistic correlation model and exploit a *correlation-aware scheduling algorithm* to achieve the same goal as the optimization problem in [11]. Simulation result will show that our proposed algorithm can have a better performance than solving the relaxed optimization problem.

III. Formulation and Proposed Algorithm

A. Problem Formulation

Let $V = \{v_1, v_2, \cdots v_N\}$ be the set of N cameras under consideration and X_i be the snapshot of image produced by camera v_i. Denote $H(X_i)$ as the amount of bits required to encode X_i independently (entropy), and $H(X_i|X_j)$ as the amount of bits required if X_j is used as reference for encoding X_i (conditional entropy). Clearly, for camera v_i to reference the image captured by camera v_j, it is required that v_j is scheduled before v_i and the transmission range of camera v_j covers camera v_i. Based on such a notation, the total amount of encoded bits for transmission can be written as follows:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ij}H(X_i|X_j), \tag{1}$$

where $H(X_i|X_i) = H(X_i)$ for sake of notation simplicity and $\alpha_{ij} \in \{0,1\}$ is an indicator variable such that $\alpha_{ij} = 1$ when camera v_i overhears camera v_j and $\alpha_{ii} = 1$ when camera v_i performs independent encoding. We require that

$$\sum_{j=1}^{N}\alpha_{ij} = 1,\ \forall i \in V, \tag{2}$$

such that each camera in V is allowed to overhear *only one* camera in its neighborhood.

B. Dominating Set Problem

Recall that our goal in this paper is to determine a proper transmission schedule. We here claim that the transmission scheduling problem can be seen as choosing part of the cameras as I-frame transmitters (broadcasters) while the other cameras are P-frame transmitters (listeners). The transmission schedule is thus simple, we let all the broadcasters to be transmitted before listeners so that the later scheduled listeners are able to reference from those previous scheduled broadcasters for reducing its encoded bits. Therefore, what we interested in is how to select broadcasters to let those listeners have the capability to reduce the most encoded bits. Since the listeners are able to reduce their encoded bits but the broadcasters cannot, one intuitive idea for radio resource conservation is to select fewest broadcasters where the transmission range of those broadcasters can cover the whole network (since a listener can only reduce encoded bits when it can overhear a broadcaster's transmission).

We now refer to the *Minimum Dominating Set* problem for selecting the fewest broadcasters in the surveillance network. In some applications, the *Minimum Dominating Set* problem is formulated as a integer programming problem [12]. Therefore, the broadcasters selection problem can be written as:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ii}H(X_j|X_i), \\ \text{subject to} \quad & A\vec{\alpha} \succeq k\vec{1}, \\ & \alpha_{ii} \in \{0,1\}, \end{aligned} \tag{3}$$

where A is an adjacency matrix indicates the overhearing capability ($i-j$ entry $=1$ if camera v_i is able to overhear camera v_j) and $\vec{\alpha}$ is a vector where its i^{th} component is α_{ii}. The ides of Problem (3) is that we want to select a subset of broadcasters so that they can have the minimum encoded bits if other listeners reference from those broadcasters. The first constraint of Problem (3) is to protect that all cameras can have at least k candidate reference broadcasters (covered by the transmission range of at least k broadcasters). By relaxing the second constraint of Problem (3) from $\alpha_{ii} \in \{0,1\}$ to $0 \leq \alpha_{ii} \leq 1$, we can get a linear programming problem as:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ii}H(X_j|X_i), \\ \text{subject to} \quad & A\vec{\alpha} \succeq k\vec{1}, \\ & 0 \leq \alpha_{ii} \leq 1, \end{aligned} \tag{4}$$

Algorithm 1 Constructing a k-tuple dominating set $\mathcal{D}_k$

Construct a dominating set $\mathcal{D}$ using GRASP [14]
$\mathcal{D}_k \leftarrow \mathcal{D}$
while There has a node whose number of dominator neighbors is less than k **do**
  $\mathcal{F} \leftarrow$ All the nodes whose number of dominator neighbors is less than k
  Add the node that dominates the largest number of nodes in $\mathcal{F}$ to $\mathcal{D}$
  $\mathcal{D}_k \leftarrow \mathcal{D}$
end while

Therefore, Problem (4) can be solved easily as a conventional linear programming problem. After obtaining the value of $\vec{\alpha}$, camera v_i is selected as a broadcaster with probability α_{ii} and the rest listeners will choose the best broadcasters as its reference camera.

Note that $k = 1$ in the conventional *Minimum Dominating Set* problem since the goal of *Minimum Dominating Set* problem is to select the fewest nodes to cover the whole graph. However, $k = 1$ is not suitable in our paper because if a camera v_i can only select its reference frame from one candidate broadcaster, it happens that the broadcaster is not correlated with camera v_i, causing that the system performance becomes lower. Therefore, we here give the motivation to increase the value of k in Problem (4) and the influence of k will be shown in our evaluation results.

We also try to use graph theory to solve the broadcaster selection problem in surveillance cameras network. Note that the idea of k introduced in Problem (4) is similar to the k-tuple dominating set problem. The authors in [13] proposed an *ICGA* algorithm which is able to generate a k-tuple dominating set based on centralized decision. The *ICGA* algorithm constructs a dominating set first and iteratively add node whose dominator neighbors is less than k into the dominating set by a greedy criteria. We here first construct a dominating set $\mathcal{D}$ by the algorithm proposed in [14], and apply the *ICGA* algorithm for generating the k-tuple dominating set $\mathcal{D}_k$. The overall procedure of the algorithm is summarized in Algorithm 1, and the performance will also be compared in our evaluation results.

C. Scheduling Algorithm

To schedule the given set of cameras for minimizing the total encoded bits, denote $\Phi \subset V$ as the subset of cameras already scheduled and ϕ_l as the last scheduled camera in Φ. Now consider two different schedules: *Schedule* 1: $\phi_l \leftarrow v_i \leftarrow v_j$ (v_i is scheduled immediately after ϕ_l) and *Schedule* 2: $\phi_l \leftarrow v_j \leftarrow v_i$ (v_j is scheduled before v_i). If the transmission order is changed from *Schedule* 1 to *Schedule* 2, the reference frame of camera v_i will change from ϕ_l to v_j, resulting in a change in the amount of encoded bits for camera v_i as $H(X_i|X_{\phi_l}) - H(X_i|X_j)$. For camera v_j, the difference in the amount of encoded bits is $H(X_j|X_i) - H(X_j|X_{\phi_l})$. Therefore, the total amount of change in the amount of

Algorithm 2 Proposed scheduling algorithm

$\Phi \leftarrow \emptyset$, $\Phi^c \leftarrow V$
while $\Phi^c \neq \emptyset$ **do** //loop until all cameras have been scheduled
  $\omega_i \leftarrow \max_{v_j \neq v_i, v_j \in \Phi^c} \Delta R(v_i, v_j), \forall v_i \in \Phi^c$ //calculate the scheduling metric for all unscheduled cameras
  $v_k \leftarrow \arg\min_{v_i \in \Phi^c} \omega_i$ //choose camera with the smallest scheduling metric as the next
  $\Phi \leftarrow \Phi \cup \{v_k\}$ //record v_k as a scheduled camera
  $\Phi^c \leftarrow \Phi^c \setminus \{v_k\}$ //remove v_k from the unscheduled cameras set
  $\phi_l \leftarrow v_k$ //update the last scheduled camera in Φ
end while

encoded bits by changing from *Schedule* 1 to *Schedule* 2 is:

$$\Delta R(v_i, v_j) = \{H(X_i|X_{\phi_l}) - H(X_i|X_j)\} + \{H(X_j|X_i) - H(X_j|X_{\phi_l})\}. \quad (5)$$

Clearly, if the amount of encoded bits can be reduced by changing from *Schedule* 1 to *Schedule* 2, then camera v_i should be scheduled after camera v_j.

Based on this concept, let v_i and v_j be two different unscheduled cameras. $\Delta R(v_i, v_j)$ as defined in Equation (5) is the difference in the amount of encoded bits if camera v_i is not the first camera to schedule after ϕ_l but deferred to the next scheduling position after camera v_j. The proposed scheduling metric for each unscheduled camera v_i can be written as:

$$\omega_i = \max_{v_j \in \Phi^c, v_j \neq v_i} \Delta R(v_i, v_j), \quad (6)$$

where $\Phi^c = V \setminus \Phi$ is the subset of all unscheduled cameras. The proposed scheduling algorithm thus is to choose camera

$$v_k = \arg\min_{v_i \in \Phi^c} \omega_i \quad (7)$$

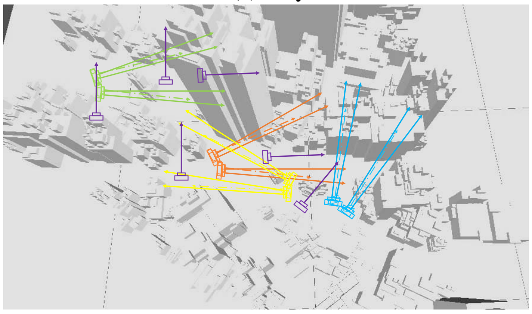
as the next camera to be scheduled for reducing the largest amount of encoded bits. As Algorithm 2 shows, the algorithm starts with $\Phi = \emptyset$ and iteratively chooses a camera to schedule based on Equation (7).

IV. Performance Evaluation

A. Experiment Settings

To create quasi-realistic 3D city views, we make use of an open-source 3D modeling software [15] and a 3D city generator [16]. A total of 30 cameras are then deployed at different locations (crossroads) inside the city of size $500m^2$ (limited by the capacity of the modeling software) for collecting the desired city snapshots (1280×720 HD images). Then, H.264 [17] is used to encode the images collected by individual cameras with or without reference frames. Figure 1 shows the 3D city view and the deployment locations for 30 cameras. With reference to real-world applications, multiple cameras are deployments at one crossroad for capturing views from different angles. The arrow in Figure 1b is the sensing

(a) City view



(b) Camera deployment

Figure 1: City view and camera deployment

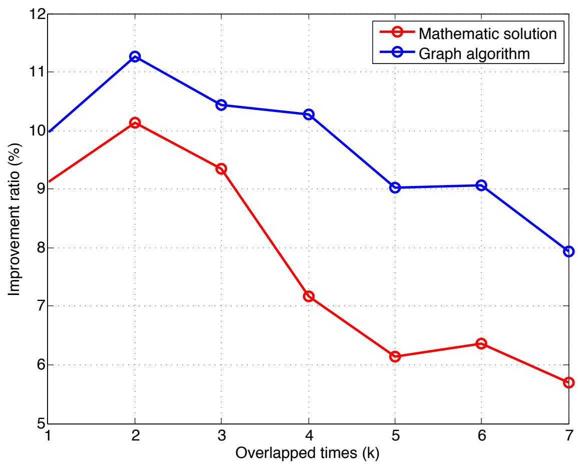


Figure 2: Number of candidate reference broadcasters

direction of each camera while different groups of cameras are shown by different colors.

Based on the above settings, we can obtain the amount of encoded bits $(H(X_1), H(X_2), \cdots H(X_{30}))$ of each camera by independent encoding its snapshot $(X_1, X_2, \cdots X_{30})$ through the H.264 reference software. The correlation between two cameras $(H(X_i|X_j), i \neq j, 1 \leq i, j \leq 30)$ is then analyzed by the multiview encoding of two different snapshots. Therefore, we can generate a correlation matrix $\mathcal{H}$ where the $i-j$ entry of $\mathcal{H}$ is $H(X_i|X_j)$. The correlation matrix $\mathcal{H}$ can be further used for our scheduling algorithm described in Section III-C.

B. Dominating Set Problem Evaluation

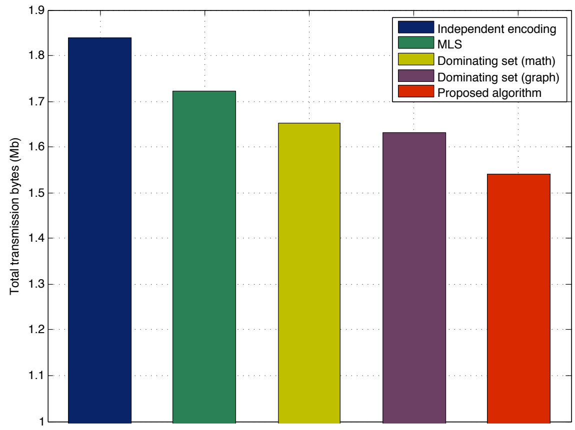


Figure 3: Performance comparison

As we mentioned before, changing the value of k in Problem (4) will cause a difference in the system performance. We here increase the value of k from 1 to 7 and compare the mathematic solution versus the heuristic graph algorithm. The results is shown in Figure 2 and we can learn that $k = 2$ is the best choice in our network scenario. This is because $k = 2$ tends to select a proper amount of broadcasters for independent transmission. Therefore, if we increase the value of k becomes larger than 2, it happens that there are too many independent transmitters and the advantage of overhearing is not that evident. However, we still want to argue that if the network becomes larger, increasing the value k is necessary to keep the overhearing performance.

C. Experiment Results

To evaluate the performance of the proposed algorithm, we compare against the MLS algorithm proposed in [11]. The authors in [11] solve a *relaxed integer programming* problem to obtain the probability that a camera should overhear transmissions of other cameras. Since the binary decision for each camera is made by approximating the probability thus solved, there is performance loss during the transformation. Figure 3 shows that MLS can improve the baseline performance (all cameras perform independent encoding) for 6.3%, whereas the dominating set problem has a 11.2% performance gain for graph algorithm while the improvement of mathematic solution is 10.1%. Most important of all, the proposed algorithm can achieve a 16.2% improvement. The result substantiates the benefits of the proposed scheduling algorithm and motives further investigation along this direction.

REFERENCES

[1] Y. Charfi, N. Wakamiya, and M. Murata, "Challenging issues in visual sensor networks," *IEEE Wireless Communications*, vol. 16, no. 2, pp. 44–49, April 2009.

[2] S. Denman, C. Fookes, J. Cook, C. Davoren, A. Mamic, G. Farquharson, D. Chen, B. Chen, and S. Sridharan, "Multi-view intelligent vehicle surveillance system," in *IEEE International Conference on Video and Signal Based Surveillance*, Nov 2006, pp. 26–26.

[3] Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg, “Clustered synopsis of surveillance video,” in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Sept 2009, pp. 195–200.
[4] L. Tessens, M. Morbee, H. Lee, W. Philips, and H. Aghajan, “Principal view determination for camera selection in distributed smart camera networks,” in *ACM/IEEE International Conference on Distributed Smart Cameras*, Sept 2008, pp. 1–10.
[5] G. R. Taylor, A. J. Chosak, and P. C. Brewer, “Ovvv: Using virtual worlds to design and evaluate surveillance systems,” in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.
[6] F. Z. Qureshi and D. Terzopoulos, “Surveillance in virtual reality: System design and multi-camera control,” in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.
[7] R. Dai and I. F. Akyildiz, “A spatial correlation model for visual information in wireless multimedia sensor networks,” *IEEE Transactions on Multimedia*, vol. 11, no. 6, pp. 1148–1159, Oct 2009.
[8] S. Colonnese, F. Cuomo, O. Damiano, V. D. Pascalis, and T. Melodia, “On the adoption of multiview video coding in wireless multimedia sensor networks,” in *Wireless Advanced (WiAd)*, June 2011, pp. 218–223.
[9] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, Jan 2005.
[10] P. Ferre, D. Agrafiotis, and D. Bull, “Fusion methods for side information generation in multi-view distributed video coding systems,” in *IEEE International Conference on Image Processing*, vol. 6, Sept 2007, pp. 409–412.
[11] P. Wang, R. Dai, and I. F.Akyildiz, “Visual correlation-based image gathering for wireless multimedia sensor networks,” in *Proceedings of IEEE INFOCOM*, April 2011, pp. 2489–2497.
[12] F. Kuhn and R. Wattenhofer, “Constant-time distributed dominating set approximation,” *Distributed Computing*, vol. 17, no. 4, pp. 303–310, 2005.
[13] Y. Wu and Y. Li, “Construction algorithms for k-connected m-dominating sets in wireless sensor networks,” in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2008, pp. 83–90.
[14] T. A.Feo and M. G.C.Resende, “Greedy randomized adaptive search procedures,” *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995.
[15] Blender Online Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Blender Institute, Amsterdam. [Online]. Available: http://www.blender.org
[16] A. Couturier, *Suicidator City Generator - Create 3D cities*. [Online]. Available: http://cgchan.com/suicidator
[17] *JMVC Reference Software*. [Online]. Available: http://ftp3.itu.int/av-arch/jvt-site/2008_10_Busan/JVT-AC207.zip.