IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 13, 2018, accepted May 30, 2018, date of publication July 2, 2018, date of current version July 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2852293

A Traffic Reduction Method for Crowdsourced Multi-View Video Uploading

THAN THAN NU[1], TAKUYA FUJIHASHI[2], (Member, IEEE), AND TAKASHI WATANABE[1], (Member, IEEE)

[1]Graduate School of Information Science and Technology, Osaka University, Osaka 5650871, Japan
[2]Graduate School of Science and Engineering, Ehime University, Matsuyama 7908577, Japan

Corresponding author: Than Than Nu (than.than.nu@ist.osaka-u.ac.jp)

ABSTRACT The integration of video streams captured by many mobile cameras (contributors) at a crowded event into a multi-view video enables remote viewers to experience the different perspectives of the event. However, because of the resource-constrained nature of wireless networks and the redundant transmission due to the highly correlated streams from multiple mobile cameras at the same event, traffic reduction is necessary to ensure the efficiency of the uploading of crowdsourced video streams. In this paper, we propose a content-based video uploading scheme for crowdsourced multi-view video streaming with the goal of reducing the video traffic from crowdsourced contributors. To achieve this, the proposed scheme uses differential encoding with multiple reference streams by means of packet overhearing. To realize differential encoding across the network of contributors for higher traffic reduction, our scheme combines three techniques: correlation estimation, reference selection, and transmission order determination. First, we utilize the correlation among the contributors based on the content features of the captured video streams using the information-bound reference. Second, in the design of the reference selection that determines the dependencies among the contributors we use two threshold values, determining the number of references for differential encoding at each contributor. Finally, we schedule the transmission order of the contributors to increase the number of differential encoding opportunities within their network. Our evaluation results show that the proposed scheme achieves a traffic reduction of up to 31% with a quality improvement of up to 2.7 dB in the connected network of contributors.

INDEX TERMS Crowdsourcing, mobile cameras, multi-view, uploading, video streaming.

I. INTRODUCTION

The proliferation of the use of smartphones with high resolution cameras together with easily accessible wireless networks have created the current trend of sharing and reporting video information over the Internet. The sharing of their captured video streams of the event via the Internet by people at a crowded event, such as a concert or a tournament, is no longer uncommon. Crowdsourced video streaming is the delivery of the video streams originating from such crowdsources [1] to remote viewers. Well-known service providers for crowdsourced video streaming services include Meerket, Periscope, and YouNow [1]–[4]. Crowdsourced multi-view video streaming [5] is an extension of crowdsourced video streaming in which many contributors viewing the same event provide different viewpoints of the event at various angles, allowing remote viewers to experience more immersive views of the scene. The applications of such services are not limited to entertainment but can be extended to other areas, such as surveillance and education.

However, simultaneous uploading of video streams from crowdsources is restricted by the inherent limitations of wireless networks, such as the available bandwidth. Therefore, the means of efficiently uploading a large amount of video traffic within the limited network resources is one of the major issues in crowdsourced multi-view video streaming.

One of the simplest methods to upload crowdsourced video streams is that each contributor independently uploads its captured streams. However, independent uploading leads to a large video traffic volume because of the redundant transmission of highly correlated video streams captured at the same event. So that the uploading will be efficient, it is necessary to reduce the amount of video traffic. To achieve traffic reduction, in this study, we considered the differential encoding-based video uploading approach presented in [6] and [7]. Differential encoding exploits inter-camera correlations to increase the coding efficiency, thus reducing the amount of video traffic. To realize differential encoding-based traffic reduction, one contributor sends its own stream, while the other contributors overhear the transmitted stream and encode their streams using the overheard one before transmission. However, three issues are involved in rendering the differential encoding efficient across the entire network of contributors, as explained in the following paragraphs.

The first issue is the acquisition of the correlation characteristics among the different video streams. In differential encoding, coding efficiency can be achieved only if the encoding and overheard streams are highly correlated; otherwise, mobile devices' resources will be wasted without any benefit being gained. To avoid this waste, it is important to determine the degrees of correlation among the contributors in order to perform differential encoding. To determine the degrees of correlation, in the study reported in [7] the topological-based approach was used in which the correlation between two cameras is decided by their positions, assuming that adjacent cameras have the highest correlations. This assumption could be violated when the two cameras in close vicinity capture the scene in different orientations. In other studies described in [6] and [8], overlapped field of view (FoV) based correlation estimation was adopted. However, all the camera and geographical parameters are needed in advance for the estimation of overlapped FoVs.

The second issue is the number of reference streams that is used in differential encoding by each contributor. In conventional video encoding, such as H.264/AVC, the use of multiple reference frames can increase the coding efficiency and/or video quality as compared to the use a single reference [9], [10], by allowing the encoder to choose the best reference from the previously decoded frames. Crowdsourced contributors can take a similar advantage, because it is likely that the video stream of a mobile camera will be correlated with that of more than one camera. However, the use of multiple reference streams can be expensive in terms of energy usage and processing power, which are limited resources in mobile devices. The more important factor is that the coding gain is not linear to the number of references. Specifically, although an improvement can be achieved from additional references, the coding gain achieved by multiple reference encoding is contributed mainly by the highest correlated reference. This is because a video encoder usually searches the best matching points in terms of a particular metric, such as mean squared error (MSE), in the two views (frames) for removing the inter-view redundancy. It is most likely that the best matching points will be found in the highest correlated reference. In this case, the effective number of reference streams that can improve the coding gain at the lowest resource cost is desirable for each contributor.

The third issue is the transmission order of the contributors. In differential encoding-based traffic reduction, the amount by which the traffic is reduced is calculated by the function of the average traffic of all the contributors in differential encoding-based and individual uploading. In this case, the traffic reduction of the entire network is affected by the number of differential encoding opportunities in the network of contributors. Here, the transmission order becomes significant for the traffic reduction. If the previously transmitted streams do not help a successive contributor to perform differential encoding, then the contributor will encode its video individually and upload independently. In this case, the number of individually encoding contributors in the network of contributors will increase. Although the later contributors can use the transmitted streams of these contributors as multiple references, the increased number of individual contributors will affect the overall traffic reduction. This type of situation occurs in the random transmission of contributors. For overall traffic reduction, the scheduling of a transmission order that can produce the smallest number of individually encoding contributors is necessary.

In this paper, we present an effective scheme for traffic reduction in crowdsourced multi-view video streaming. Considering the three issues mentioned above, our scheme consists of three parts: correlation estimation, reference selection, and transmission scheduling. To address the first issue, we use content-aware correlation estimation. Specifically, the content features of each video are extracted and compared with those of its neighbors to reveal the similarities between contributors. To address the second issue, we use a reference selection method in which a contributor optionally decides whether to conduct single-reference or multiple-reference encoding based on its correlated neighbors. For this purpose, we define two threshold values for determining the types of encoding: $\beta_{\text{multi\_ref}}$ for multiple-reference encoding and $\beta_{\text{single\_ref}}$ for single-reference encoding. If there are two or more correlated streams having a correlation degree greater than $\beta_{\text{multi\_ref}}$, the contributor performs multiple reference encoding; otherwise, it selects the highest correlated stream having a correlation degree greater than $\beta_{\text{single\_ref}}$ for the differential encoding with a single reference.

To envisage the differential encoding with multiple references, a contributor would have to listen to all the transmissions of its neighbors and determine the correlation degrees between its own and the overheard streams, which would consume the contributors' resources, such as batteries. To avoid this problem, we present a centralized decision-making scheme in which an access point (AP) determines the encoding dependencies among the contributors based on their degrees of correlation and schedules the contributors' transmission order. For the transmission order determination, the AP first constructs the correlated network of contributors considering the reference selection criterion. Then, it traverses the resulting network in a depth-first-search manner to generate the transmission order. The decision includes the transmission slots and overhearing slots for each contributor. Finally, the AP broadcasts the transmission order to the contributors to initiate their uploading of video streams.

We evaluated our proposed scheme in comparison with other reference schemes, including our previously proposed methods [11], namely, Maximum_Correlation and Multiple_References. The evaluation results show that the proposed scheme can reduce the amount of traffic by up to 31% as compared to individual uploading. The proposed scheme outperforms Maximum_Correlation and Multiple_References by 9% and 1%, respectively. In terms of the number of references, the proposed scheme uses up to four references, whereas Multiple_References uses up to seven references in a correlated network of eight contributors.

Our contribution is three-fold.

- First, our content-based correlation estimation overcomes the weakness of the overlapped FoV- and topological-based approaches; for example, mobile cameras that are adjacent but facing in different directions would have no correlation.
- Second, we consider differential encoding with multiple reference streams by exploiting all the correlations among the contributors to obtain a higher compression gain. Moreover, our proposed method shows a balance between the traffic reduction and resource requirements of the devices achieved by selecting the effective reference streams for differential encoding without overloading the video encoding mechanism.
- Third, our scheduling algorithm determines the transmission slots and overhearing slots for each contributor. Consequently, a contributor needs to listen only to its correlated neighbors, avoiding wasting its device's resources.

II. RELATED WORK

A. CROWDSOURCED VIDEO STREAMING

The design of an end-to-end system of crowdsourced video streaming involves many research areas, which can be roughly divided into two categories: uploading of the streams from the crowdsources to the server, such as a cloud, and downloading of the contents by the remote viewers. In the study presented in [12], Tang *et al.* focused on the downloading aspect of crowdsourced mobile video streaming. They proposed a multi-object multi-dimensional auction-based incentive framework for cooperative downloading of crowdsourced video streams, which enables mobile users located close to each other to form cooperative groups and share their network resources for more efficient video streaming. Unlike the authors of the study presented in [12], we focused on the uploading aspect of the correlated crowdsourced video streams to a server.

The studies reported in [2] and [13]–[15]were focused on the aspect of the uploading of the streams considering the resource constraints of the crowdsources and the wireless network. Tai *et al.* [14] tackled the delay aspect of crowdsourced video uploading. They focused on reducing the uploading time of mobile users sharing multimedia contents at an event. For this purpose, they proposed a proxy offloading server at the wireless AP, which assigns Wi-Fi bandwidths to the mobile users. Through knowledge of the file-uploading time of each task, the uploading time of the mobile users can be reduced. Bommes *et al.* [15] considered scalable video coding (SVC) and chunked video content for optimizing video quality and delay in live video sharing from mobile devices. They proposed a set of uploading scheduling algorithms that select video chunks with various layers of quality for uploading and determine the order of uploading in order to optimally balance the quality-delay tradeoff.

In [13], Venkatagiri *et al.* considered the pull-based on-demand uploading of crowdsourced mobile videos at an event, where it was not intended that all captured videos be uploaded to the server. To achieve this, in the scheme a subset of video clips from smartphones is selectively collected, which consists of only a sufficient number of clips to satisfy a viewer's spatio-temporal query that includes the desired viewing angle θ, point of interest γ, and duration (t_1, t_2) of the event. These values are the metadata calculated from the captured videos and reported by the smartphones periodically. The videos are selected to balance the viewers' satisfaction and the uploading cost incurred by the smartphones, resulting in a tradeoff between the accuracy of the video clips, as required by the orientation and temporal coverage, and the power budget of the devices. A similar research study was conducted on a method for photo crowdsourcing from mobile devices [2] aimed at selecting the photos with the largest utility. The method measures the extent to which the photos cover the target area, based on metadata, such as the location, orientation, FoV, and range of a camera.

In crowdsourced video streaming systems, the video contributors and viewers are heterogeneous in terms of the generated video quality and network configurations. The delivery of heterogeneous video streams to heterogeneous viewers requires massive transcoding and demands high computational resources. To tackle this, in [3] a generic framework that uses cloud computing services for crowdsourced live streaming with heterogeneous contributors and viewers was presented. The authors focused on the cloud resource allocation to the contributors for transcoding a set of video representations, i.e., on quality in order to maximize the users' quality of experience (QoE) and minimize the computational cost. Similarly, Bilal *et al.* [5] proposed a generic framework for crowdsourced multi-view live video streaming, namely, Cloud-based Multi-View Crowdsourced Streaming (CMVCS). As in [3], He *et al.* formulated the resource allocation problem to transcode the views in an optimal set of representations, subject to the computational and communication resource constraints. In both of the studies, popularity-based selection of views (contributors) and a set of representations that optimizes the viewers' satisfaction were considered.

The selective uploading of views based on the user's request and coverage may incur a delay and necessitate a tradeoff between the accuracy of the request and the resource constraint, such as the battery capacity of the devices. In contrast, in our study, we considered the uploading of video streams from all contributors in which redundant information is removed by exploiting the inter-view correlation in encoding, to achieve efficient video uploading.

B. CORRELATION-BASED CONTENT UPLOADING

In our study, we considered the correlation-based traffic reduction for crowdsourced multi-view video streaming. A similar study was conducted by Kodera *et al.* [7] for traffic reduction in multi-view video streaming with multiple mobile cameras. They focused on reducing the amount of traffic between the mobile cameras and the AP by using packet overhearing and bidirectional encoding. Each camera overhears two other cameras' frames and uses bidirectional inter-view prediction to exploit the correlation between its own and the overheard frames. In addition, the transmission order of the cameras is controlled by the AP, which enables bidirectional encoding based on the positions of the mobile cameras, assuming that the cameras nearest to each other have the highest correlation.

Many studies on correlation-based content uploading in wireless sensor networks have been conducted. Wang and Akyildiz [8] proposed a spatial correlation-based image compression framework for wireless multimedia sensor networks to maximize the overall compression of the collected visual information. They also proposed a differential encoding-based scheduling framework [6] for uploading visually correlated images to wireless multimedia sensor networks. The paper describes the design of a schedule for the sensor nodes to maximize the network lifetime by performing differential encoding using overheard transmissions of correlated neighbors. In both studies, the authors considered the overlapped FoV of the cameras to predict the correlation among them, calculated using the camera setting parameters of position, sensing direction, and the location of the area of interest. In the studies presented in [16] and [17], image processing methods were applied to estimate the correlation among images from neighboring sensors in order to conduct collaborative transmission. In the study in [16], images from correlated views were approximately registered utilizing the image feature points and feature point correspondence. In this scheme, each sensor transmits the low-resolution version of a common area, and the sink reconstructs the high-resolution version using the super-resolution technique [18]. In the method described in [17], images from correlated sensors are collaboratively transmitted to the sink based on the spatial and temporal correlation. A shape matching method is used to obtain the spatial correlation between images acquired from neighboring sensors, whereas background subtraction is used for temporal correlation.

The assumption of positional correlation [7] could be violated if cameras in close proximity to each other project in different orientations. Overlapped FoV-based correlation estimation [6], [8] may overcome the deficiency of the positional approach. However, all the camera and geographical parameters are required in advance for estimation. In our study, we considered the image processing-based correlation estimation approach that uses the information-bound reference (IBR) [19], [20]. As compared to the methods in [16] and [17], our IBR-based correlation estimation is less complex, because it does not require feature extraction to reveal the similarity between two images. Instead, it uses a multimedia fingerprint algorithm to generate a 64-bit hash-code from the discrete cosine transform (DCT) components of the image to uniquely represent the content features.

In [6] and [8], Wang *et al.* and Wang and Akyildiz assumed a limited number of dependencies between the cameras. Specifically, in their method each camera is dependent on the camera that is most closely correlated with it and it must be a direct successor of its predecessor. In a crowdsourced environment, the dependencies among the cameras may be more complex and it is very likely that a predecessor itself can be dependent on another camera. Thus, we consider multiple dependencies among the cameras by exploiting all the correlations among them. In the method described in [7], two reference streams are used for bidirectional encoding based on the positions of the cameras, regardless of the actual correlation among them. In contrast, we effectively select the number of references for a contributor for differential encoding based on the correlation degrees.

III. PROPOSED SYSTEM

A. SYSTEM MODEL AND ASSUMPTIONS

Fig. 1 shows the system model of our study. In this model, mobile cameras (contributors) capture videos of a crowded event, such as a concert or a sports competition, from different viewpoints and upload them to a video collector, e.g., server, via a wireless channel. The collector is located at the same location as the event and continuously requests the videos from the contributors through a wireless AP.

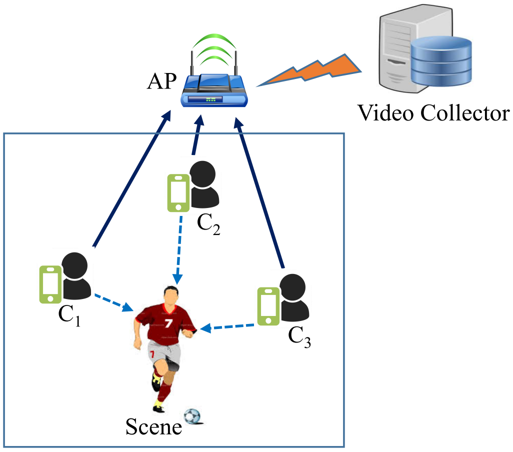


FIGURE 1. System model ($C_i = 1, \ldots, 3$ = contributors).

In our model, it is assumed that all the contributors can reach the collector via one-hop communication and no communication errors between the collector and contributors occur. Regarding mobility, we assume that the degree of correlation between the contributors is the same during each unit interval of video transmission, that is, one group of pictures (GOP) transmission.

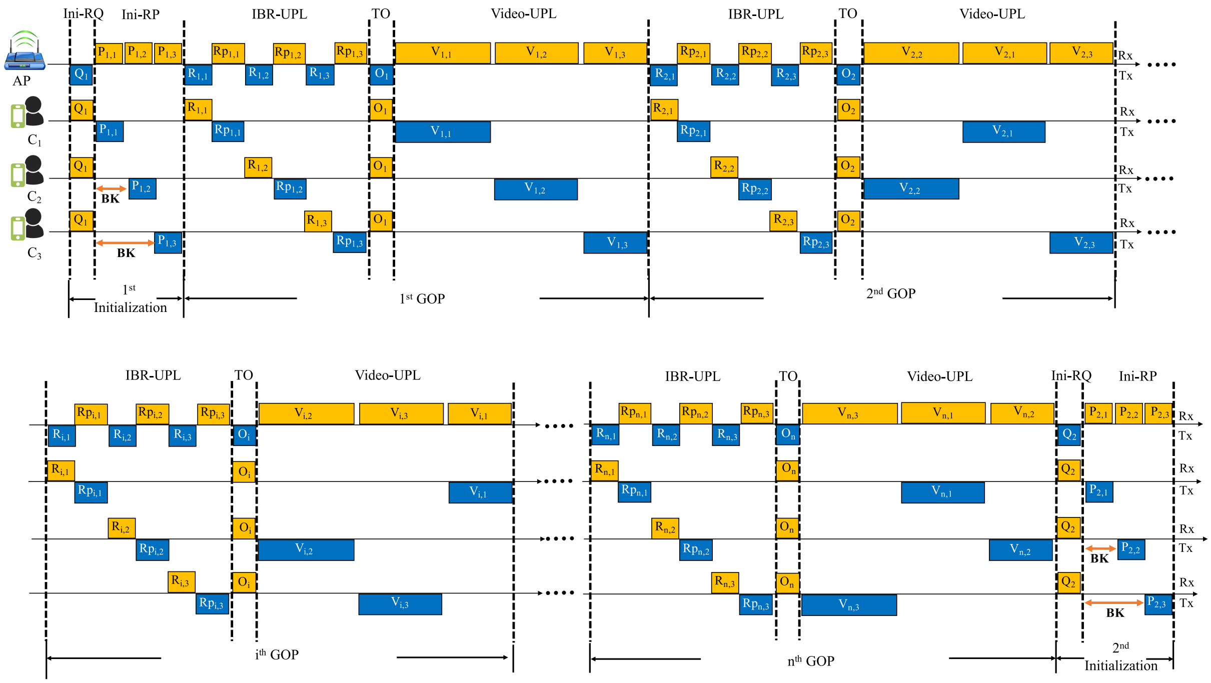


FIGURE 2. Timing diagram of video uploading. (BK: Backoff time; yellow boxes: receiving packets; blue boxes: transmission packets).

In addition, we assume that the contributors have synchronized in advance using Global Positioning System (GPS)-based synchronization.

B. TIMING DIAGRAM

Fig. 2 shows the timing diagram of our video uploading scheme. In this figure, it is assumed that the videos from three contributors are uploaded to the video collector through the AP on a GOP basis. For the first GOP, the uploading process consists of four phases, as described below.

1) INITIALIZATION (INIT)

First, the AP broadcasts the request message, Q, in order to obtain the number of contributors located in its communication range. On receiving the message, the contributors notify their existence to the AP with the response message, P, within the predefined interval set by the AP in the request message. The response message contains the location of each contributor, e.g., GPS data.

2) IBR UPLOADING (IBR-UPL)

In this phase, the AP collects the content information of the captured video from each contributor with the request, R. The contributors upload the information in the form of the IBR with the response, R_{p}. The generation of the IBR from the captured video is described in Section III-C. Using the IBR of each contributor, the AP estimates the correlation degrees among the contributors.

3) TRANSMISSION ORDER NOTIFICATION (TO)

With the knowledge of the correlation degrees, the AP determines the encoding dependencies among the contributors and schedules the transmission order of the contributor. Then, it broadcasts the decision by means of a message, O, so that the contributors can initiate the uploading of their captured videos. The message contains the information of transmission slots and overhearing slots for each contributor.

4) UPLOADING VIDEOS (VIDEO-UPL)

In this phase, the contributors start uploading video streams according to the transmission order. According to the encoding dependencies, some contributors encode their videos individually and upload them independently; otherwise, they conduct the inter-camera differential encoding using the overheard video streams before uploading.

Considering the correlation degree variation due to the mobility of each contributor, the IBR is refreshed for each GOP, and the AP reschedules the transmission order. After every n GOP videos from all contributors have been uploaded, the AP updates the number of contributors in its communication range by restarting the initialization.

C. INFORMATION-BOUND REFERENCE CALCULATION

In order to estimate the degrees of correlation among the contributors, the content features of every first frame in 1 GOP of each contributor are extracted and compared with those of its neighbors to reveal the similarities. The content features are reported from the contributors to the AP in the form of the image IBR.

The IBR is an alternative to the links and content references that are the interaction means utilized by users of today's Internet. The links and references currently used are bound to a protocol, a host, a filename, a specific data presentation format, encoding, and resolution [19], [20]. According to [19], the links are fragile and users are usually concerned with the intent of the reference link rather than with low-level representations. Therefore, a content reference should be bound to the underlying information of the content.

Multimedia fingerprint algorithms can be used to generate the IBR because of their similar characteristics, such as unique representation of the content. In our study, we used the scheme described in [20]. An IBR is generated from an uncompressed frame, which is the first frame taken from each GOP. An overview of the image IBR generation is shown in Fig. 3. First, the first frame in each GOP is resized to the baseline resolution of 128 × 128 pixels. The resized frame contains sufficiently detailed structures of the content. Next, the YCbCr representation of the resized image is generated. We take the Y component from this representation and apply the DCT operation to it to obtain the DCT coefficients. From the DCT coefficient matrix of Y, we take the lower end 8 × 8 submatrix. Then, we find the median value of the coefficients and quantize each coefficient to be 0 or 1 if they are higher or lower than the median to generate a 64-bit hash value. Our reasons for using the IBR are that its generation is simple and a small number of bits are required by the AP to estimate the correlation degrees among the video streams. Given the IBRs, the AP computes the correlation coefficient, α_{v_i,v_j}, between any two contributors using

$$\alpha_{v_i,v_j} = 1 - \frac{d_{\mathrm{i,j}}}{d_{\mathrm{max}}}, \quad 0 \leq \alpha_{v_i,v_j} \leq 1 \tag{1}$$

where d_{max} is the maximum Hamming distance and $d_{\mathrm{i,j}}$ is the Hamming distance of the image IBR of contributors i and j. The correlation coefficient between two contributors i and j is symmetric, that is, $\alpha_{v_i,v_j} = \alpha_{v_j,v_i}$. We assume that α_{v_i,v_j} is zero for any two contributors that are not neighbors of each other, i.e., there is no correlation between them. The neighborhood of each of the contributors is determined by the AP according to their locations.

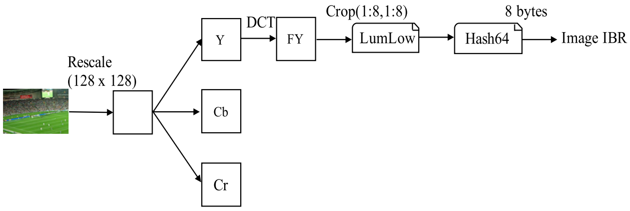


FIGURE 3. Generating image information-bound reference.

D. REFERENCE SELECTION

For differential encoding with multiple references, we proposed two reference selection methods in our previous paper [11], namely, Maximum_Correlation and Multiple_References. In Maximum_Correlation, a contributor i selects one of its correlated neighbors j with the maximum correlation degree, $\mathrm{argmax}_j(\alpha_{i,j})$, as a reference. In Multiple_References, a contributor listens to the transmitted streams from all of its correlated neighbors. Our evaluation results showed that Multiple_References outperforms Maximum_Correlation in a densely correlated network of contributors. However, encoding with Multiple_References demands more resources, such as energy and processing power. These resources are not abundantly available in consumer-grade mobile devices.

Therefore, we propose an efficient reference selection method to achieve high traffic reduction so that fewer resources are required. In Multiple_References, a certain contributor differentially encodes its video stream using reference streams with different correlation degrees. In this case, it is more likely that the achieved coding gain is contributed mainly by the highest correlated reference, since the best matching points for exploiting the correlation between two frames are more likely to be found in the most highly correlated frames. An additional coding gain can be obtained from the references with similar correlation degrees. This means that a contributor can improve its coding gain by using multiple references only if the references have higher or similar correlation degrees; otherwise, it should use only the single reference from the contributor most highly correlated with it. Considering this, we define two threshold values, namely, $\beta_{\mathrm{multi\_ref}}$ and $\beta_{\mathrm{single\_ref}}$, to effectively select the references for a contributor. $\beta_{\mathrm{multi\_ref}}$ is the threshold of the correlation degree that can improve the coding gain from encoding with multiple references, whereas $\beta_{\mathrm{single\_ref}}$ is the minimum correlation degree that can benefit for differential encoding. In this study, we set $\beta_{\mathrm{single\_ref}}$ to 0.625, because no coding gain results from differential encoding [21]. For $\beta_{\mathrm{multi\_ref}}$, we empirically selected the correlation degree threshold of 0.875. If two or more streams having a correlation degree above $\beta_{\mathrm{multi\_ref}}$ are available for a contributor, it takes multiple-reference encoding; otherwise, it selects the highest correlated reference, the correlation degree of which is greater than $\beta_{\mathrm{single\_ref}}$, for differential encoding with a single reference, as in Maximum_Correlation.

E. TRANSMISSION ORDER DETERMINATION

The traffic reduction of the entire network of contributors is affected by the number of differential encoding opportunities in the network. If the overheard streams are not useful for differential encoding, then the number of individually encoding contributors increases. Although these contributors can use the overheard streams for differential encoding, the increasing number of contributors using individual encoding will affect the overall traffic reduction. Fig. 4 shows the effect of the transmission order on the number of individual contributors. Fig. 4(a) shows an example network of contributors in which the undirected arrows indicate the correlation among them. It is assumed that contributors transmit their video streams in random order. In Fig. 4(b), Contributor 2 starts transmission followed by Contributor 4. The directed arrows indicate the usefulness of overheard streams for differential encoding at subsequent contributors. Contributor 4 cannot use the overheard stream from Contributor 2, because they have no correlation. In this case, the number of individually encoding contributors is 2, i.e., Contributors 2 and 4, regardless of the transmissions order of the rest of the contributors. However, if the contributors transmit their video streams in the order shown in Fig. 4(c) (starting from Contributor 1, followed by 2, and 3, etc.), the number of contributors that take individual encoding is only 1, i.e., Contributor 1. Considering this, our goal is to schedule the transmission order of the contributors in order to reduce the overall video traffic. To achieve this, we designed an algorithm, described in Algorithm 1, for transmission order determination. Table 1 lists the notations used in the algorithm and their descriptions.

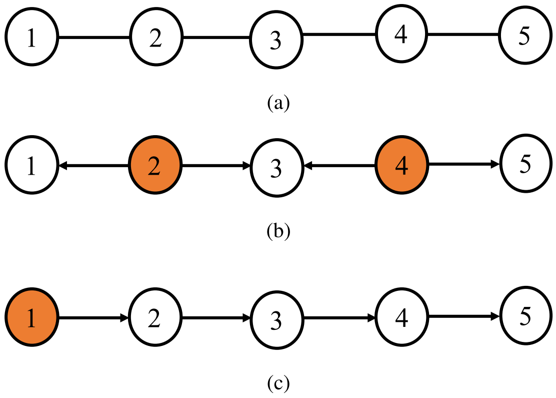


FIGURE 4. Effect of random transmission order. (a) Network of correlated contributors. (b) Transmission order with two individually uploaded contributors. (c) Transmission order with one individually uploaded contributor.

TABLE 1. Summary of notations.

Notation	Description
$V_{\text{contributor}}$	Set of contributors in the range of the AP
IBR_{v_i}	IBR value of contributor i
N_{v_i}	Set of neighbors of contributor i
Order	Transmission sequence of the contributors
$V_G, V_{G'}$	Set of nodes in the graphs G and G'
$E_G, E_{G'}$	Set of edges in the graphs G and G'
(v_i, v_j)	Edge between nodes i and j
α_{v_i, v_j}	Correlation coefficient between nodes i and j
$V_{v_i}^{adj}$	Set of adjacent nodes of node i
cal_Correlation($\text{IBR}_{v_i}, \text{IBR}_{v_j}$)	Calculate the correlation coefficient between contributors i and j
find_MultiRefNodes($\beta_{\text{multi\_ref}}$)	Find the nodes whose correlation degrees is greater than $\beta_{\text{multi\_ref}}$

Algorithm 1 Transmission Order Determination

Input: $V_{\text{contributor}}, \text{IBR}_{v_i}, N_{v_i}, \forall v_i \in V_{\text{contributor}}$
Output: Order

Step 1: Constructing the Correlation Graph, G

```
$V_G \leftarrow V_{\text{contributor}}$
while $|V_{\text{contributor}}| \neq 0$ do
    $v_i \leftarrow v \in V_{\text{contributor}}$
    for each $v_j \in N_{v_i}$ do
        $\alpha_{v_i,v_j} \leftarrow$ cal_Correlation($\text{IBR}_{v_i}, \text{IBR}_{v_j}$)
        $N_{v_j} \leftarrow \text{N}_{v_j} \setminus \{v_i\}$
    end for
    $V_{\text{multi\_ref}} \leftarrow$ find_MultiRefNodes($\beta_{\text{multi\_ref}}$)
    if $|V_{\text{multi\_ref}}| > 1$ then
        $E_G \leftarrow (v_i, v_j) \ \forall v_j \in V_{\text{multi\_ref}}$
    else if $\text{argmax}_j(\alpha_{i,j}) > \beta_{\text{single\_ref}} \ v_j \in N_{v_i}$ then
        $E_G \leftarrow (v_i, v_j)$
    end if
    $V_{\text{contributor}} \leftarrow V_{\text{contributor}} \setminus \{v_i\}$
end while
```

Step 2: Constructing the Dependency Graph, G'

```
$V_{G'} \leftarrow V_G$
$v_i \leftarrow v \in V_G$
while $|V_G| \neq 0$ do
    Order $\leftarrow$ Order $\cup \{v_i\}$
    for each $v_j \in V_{v_i}^{adj}$ do
        if $v_j \notin P_{v_i}$ then
            $E_{G'} \leftarrow (v_i, v_j)$
        end if
    end for
    $V_G \leftarrow V_G \setminus \{v_i\}$
    $V_{v_j}^{adj} \leftarrow V_{v_j}^{adj} \setminus \{v_i\}$
    if $|V_{v_i}^{adj}| > 0$ then
        $v_i \leftarrow v \in V_{v_i}^{adj}$
    else
        $v_i \leftarrow v \in V_G$
    end if
end while
return Order
```

The algorithm consists of two operations: the construction of the correlation graph and the construction of the dependency graph. First, the algorithm constructs an undirected graph, $G = (V, E)$, called a correlation graph, where $V = \{v_i, \ i = 1, 2, \ldots, |\text{V}|\}$ is the set of contributors and E is the set of edges that shows the correlation among the contributors according to the reference selection criterion, as described in Section III-D. For each contributor v_i, edges (v_i, v_j), where the value of j is above 1, are added to G if the correlation degrees between contributors v_i and v_j are greater than the threshold value of $\beta_{\text{multi\_ref}}$; otherwise, an edge (v_i, v_j) is added to G, which has the highest correlation degree and the correlation degree is greater than $\beta_{\text{single\_ref}}$. The resulting correlation graph is shown in Fig. 5(a).

Second, the algorithm constructs a directed graph, G', based on the correlation graph to determine the dependen-

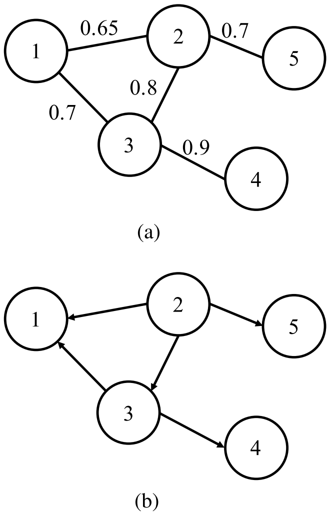


FIGURE 5. Correlation graph and corresponding dependency graph. (a) Correlation graph. (b) Dependency graph.

cies among the contributors and to generate the transmission order. To construct graph G', we traverse the correlation graph in the depth-first-search manner and convert each undirected edge to a directed edge pointing to its neighbor nodes. This type of graph construction can guarantee that all the nodes (except the first one) become the successors of one or more predecessors in each connected component of the correlation graph. This increases the number of differential encoding opportunities for the network of contributors. Graph G' needs to be acyclic so that the transmission order contains no loop. In other words, edges (v_i, v_j) and (v_j, v_i) cannot exist in graph G' at the same time. To avoid a loop, we ensure that a directed edge from a node does not point to its predecessor by eliminating the predecessor from the neighbor list of each successor node (line 26). The resulting dependency graph G' is illustrated in Fig. 5(b), which shows the dependencies among the contributors. In other words, the number of reference streams used by each contributor depends on the number of its predecessors in graph G'. The transmission order of the contributors is generated as the order of the nodes in the graph traversal (line 19).

F. ENCODING

After determining the transmission order, the AP broadcasts the decision to the contributors to initiate the uploading of video streams from the contributors. Before transmission, each contributor encodes its video stream in accordance with its dependency on other contributors in graph G'. Specifically, the source nodes in graph G', which have no predecessor, encode their streams individually, while the successor nodes overhear the transmissions from their predecessors and take differential encoding. As an example, the encoding behaviors of three correlated contributors are illustrated in Fig. 6. In Fig. 6(a), Contributor 1 is a source node and encodes its video individually and becomes a predecessor of Contributors 2 and 3. Contributor 2 overhears the transmission of Contributor 1 and takes differential encoding with one reference, as shown in Fig. 6(b). In Fig. 6(c), Contributor 3 encodes its video differentially by taking the overheard streams from its two predecessors, Contributors 1 and 2.

IV. EVALUATION

A. SETUP

In order to quantify its performance, we investigated the behaviors of our proposed method in different scenarios of crowdsourced video uploading by means of simulations using MATLAB.

1) METRIC

We evaluated the performances of the proposed and reference schemes in terms of video traffic and peak signal-to-noise ratio (PSNR). Video traffic represents the number of bits needed to transmit from all the contributors. The PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \frac{(2^L - 1)^2}{\varepsilon_{\text{MSE}}}, \tag{2}$$

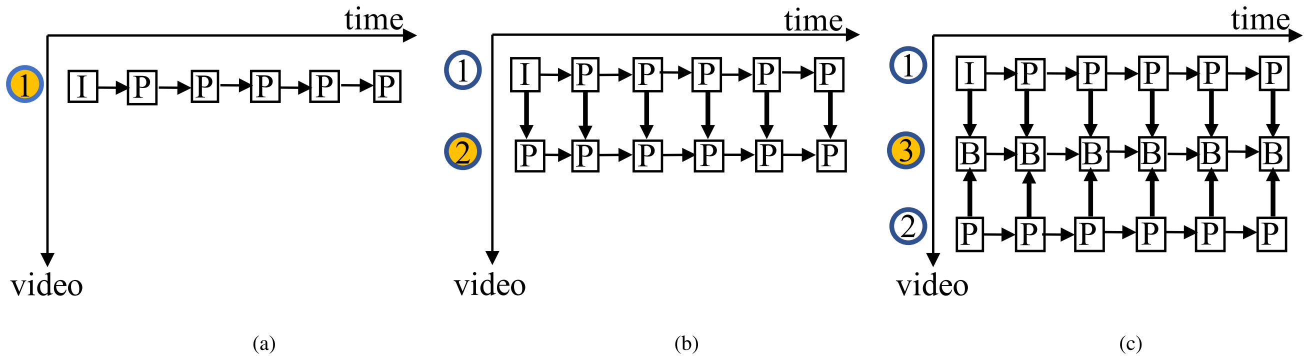


FIGURE 6. Encoding behaviors of three correlated contributors. (a) Individual encoding. (b) Differential encoding with one reference. (c) Differential encoding with multiple references.

where L is the number of bits used to encode pixel luminance (typically eight bits), and $\varepsilon_{\mathrm{MSE}}$ is the mean squared error (MSE) between all pixels of the decoded and the original videos.

2) VIDEO SEQUENCE

We used videos from the standard multi-view video sequences known as *Vassar*, *Ballroom*, and *Exit* [22] at 25 frames per second and in quarter common intermediate format (QCIF) with 176 × 144 resolution.

3) DEPLOYMENT OF CONTRIBUTORS

We arranged the video sequences in two different fashions to reflect some uploading scenarios of crowdsourced multi-view video streaming.

- As Deployment 1, we selected eight videos from *Vassar*. This deployment considers a fundamental situation in which all the contributors capture the same view, resulting in a high correlation among the contributors and creating a fully connected network of contributors.
- Deployment 2 was arranged as three groups of contributors capturing different views. Such a scenario is more likely to occur in crowdsourced video uploading. Each group constituted the contributors with high similarities of captured videos.

The arrangement of the video sequences in the two deployments is expressed in Table 2. In addition, the graph structures of the deployments and the correlation degree between the nodes are illustrated in Fig. 7. The length of 1 GOP is set to 10 frames. Finally, we ran the simulations using 10 different quantization parameters from 20 to 30.

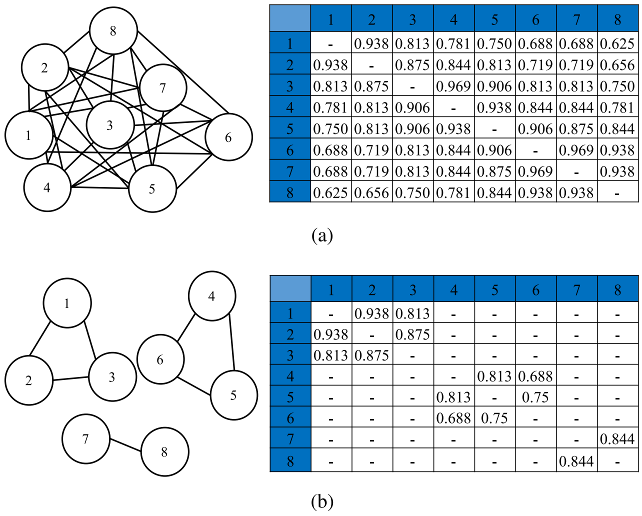
TABLE 2. Deployments of eight videos.

Sequence	Deployment 1	Deployment 2
Vassar	Cam. 0 to 7	Cam. 0, 1, and 2
Ballroom	-	Cam. 0, 1, and 2
Exit	-	Cam. 0 and 1

4) REFERENCE SCHEMES

We compared the video traffic and PSNR of our proposed scheme with those of five other schemes described as follows.

1) Individual_Uploading : Individual_Uploading is the baseline method for uploading crowdsourced videos, in which each contributor encodes its captured stream individually and uploads it to the AP.
2) Max_Correlation : Max_Correlation is one of our previously proposed methods. This method conducts differential encoding considering only the maximum correlation degree between the contributors to construct the correlation network.
3) Multiple_References : Multiple_References is also one of our previously proposed schemes; it exploits all the correlations among the contributors and conducts differential encoding using the multiple reference streams.



	1	2	3	4	5	6	7	8
1	-	0.938	0.813	0.781	0.750	0.688	0.688	0.625
2	0.938	-	0.875	0.844	0.813	0.719	0.719	0.656
3	0.813	0.875	-	0.969	0.906	0.813	0.813	0.750
4	0.781	0.813	0.906	-	0.938	0.844	0.844	0.781
5	0.750	0.813	0.906	0.938	-	0.906	0.875	0.844
6	0.688	0.719	0.813	0.844	0.906	-	0.969	0.938
7	0.688	0.719	0.813	0.844	0.875	0.969	-	0.938
8	0.625	0.656	0.750	0.781	0.844	0.938	0.938	-

(a)

	1	2	3	4	5	6	7	8
1	-	0.938	0.813	-	-	-	-	-
2	0.938	-	0.875	-	-	-	-	-
3	0.813	0.875	-	-	-	-	-	-
4	-	-	-	-	0.813	0.688	-	-
5	-	-	-	0.813	-	0.75	-	-
6	-	-	-	0.688	0.75	-	-	-
7	-	-	-	-	-	-	-	0.844
8	-	-	-	-	-	-	0.844	-

(b)

FIGURE 7. Graph structures of network of contributors. (a) Deployment 1. (b) Deployment 2.

4) Random_Order : In this scheme, the AP does not control the transmission order of the contributors; instead, the contributors transmit their streams in random order. The contributors conduct differential encoding by overhearing without considering the degrees of correlations between their own and overheard streams.
5) MVS/MC : MVS/MC [7] supports the transmission order control based on topological information to realize bidirectional differential coding using the two overheard streams from its adjacent contributors.
6) Proposed : This is our proposed scheme, as described in Section III.

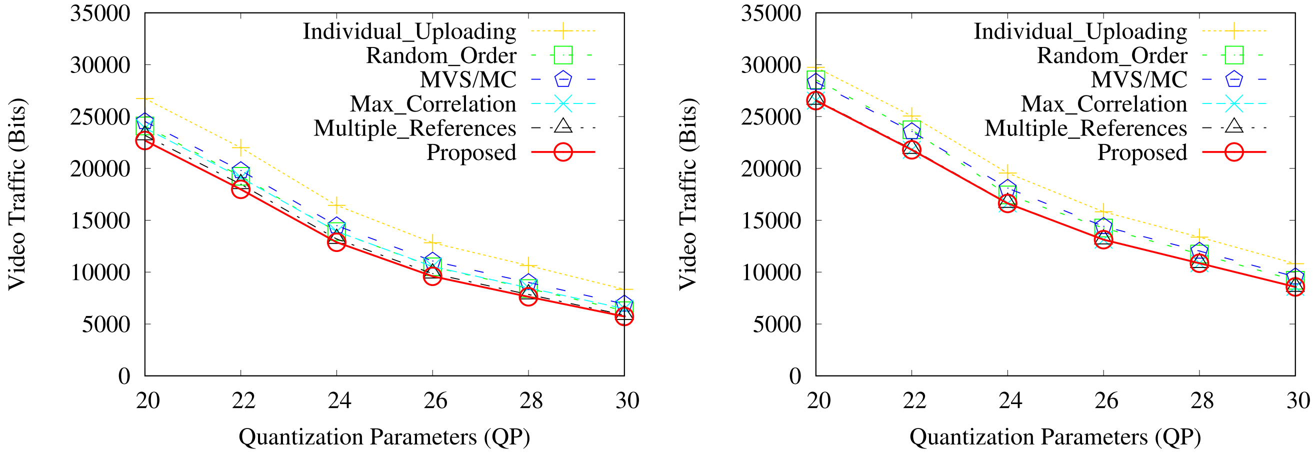
B. RESULTS AND ANALYSIS

1) VIDEO TRAFFIC

Fig. 8 shows video traffic at different quantization parameters for two deployments. In both deployments, all other reference schemes achieve a traffic level lower than Individual_Uploading, showing the benefit of differential encoding for traffic reduction.

In Deployment 1, the proposed scheme reduces the video traffic by 31% as compared to Individual_Uploading. In addition, the performance of the proposed scheme is superior to that of Max_Correlation and Multiple_References by 9% and 1%, respectively. In terms of selecting the correlated reference, the proposed scheme achieves a 14% greater traffic reduction than MVS/MC. In addition, Random_Order has more video traffic than the proposed scheme, irrespective of the quantization parameters. This proves the effectiveness of the scheduling algorithm for transmission order determination among the contributors.

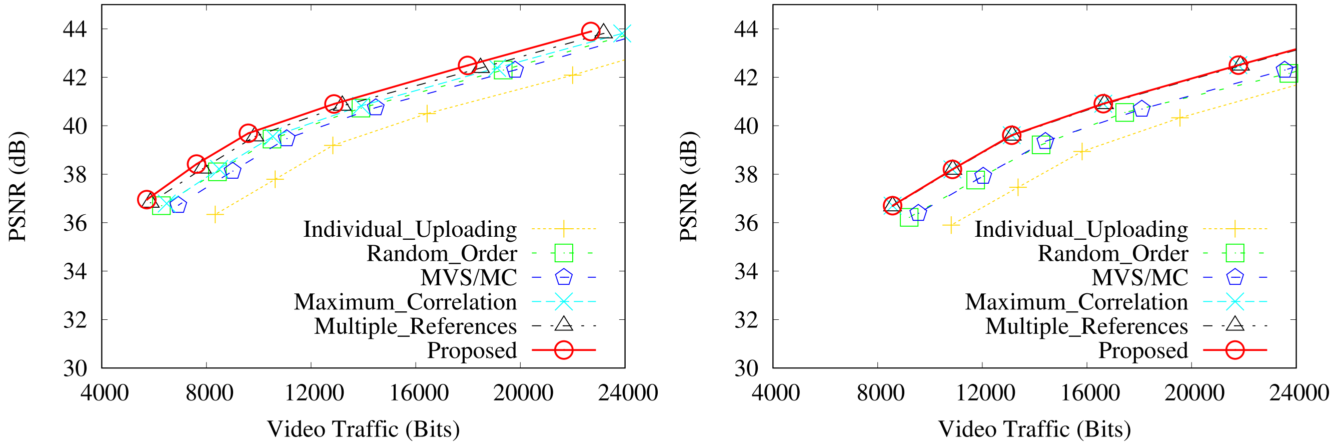
However, in Deployment 2, the traffic reduction between the proposed scheme and Individual_Uploading decreases to 21%. In each disconnected component, there is one source node. The number of source nodes monotonically increases with the number of disconnected components in the network.



(a)

(b)

FIGURE 8. Video traffic at different quantization parameters. (a) Deployment 1. (b) Deployment 2.



(a)

(b)

FIGURE 9. Video quality vs. traffic. (a) Deployment 1. (b) Deployment 2.

A large number of source nodes causes a large number of individually encoding contributors, and thus, low traffic reduction. Moreover, the performances of Max_Correlation and Multiple_References are similar to that of the proposed one scheme. This can be explained by the fact that the numbers of references used in Max_Correlation and the proposed scheme become the same. Although Multiple_References uses many references for differential encoding, the coding gain is not significantly high. However, the proposed scheme still achieves a traffic reduction that is larger than that of MVS/MC and Random_Order by 9% and 7%, respectively.

To summarize, our proposed scheme outperforms all the reference schemes in terms of traffic reduction, regardless of contributor's network.

2) VIDEO QUALITY

Fig. 9 shows the video quality of the reference schemes as a function of the video traffic. In both deployments, it can be seen that utilizing the inter-camera correlations for compression achieves a higher video quality. In deployment 1, the proposed scheme improves the PSNR performance by 2.7 dB as compared to Individual_Uploading at an average video traffic of 9500 bits. Moreover, at the same video traffic, the proposed system achieves a quality improvement that is greater than that of Max_Correlation and Multiple_References by 0.9 dB and 0.1 dB, respectively. In addition, the proposed scheme outperforms MVS/MC and Random_Order by 1.5 dB and 1 dB, respectively.

In Deployment 2, the proposed scheme improves the video quality as compared to Individual_Uploading by 2.2 dB at a traffic volume of 12000 bits. The same quality improvement is obtained by Max_Correlation and Multiple_References. However, the proposed scheme improves the video quality as compared to MVS/MC and Random_Order by 0.9 dB and 1 dB, respectively.

V. DISCUSSION

A. EFFECT OF LARGE NUMBER OF CONTRIBUTORS

The evaluations described above were based on deployments for eight video sequences. As shown in Fig. 8(b) and Fig. 9(b), the performances of Max_Correlation and the proposed method are the same in Deployment 2, because they use the same number of references for differential encoding because of the limited number of contributors. To examine the performance difference of two methods in a disconnected network of contributors in more detail, we considered their behavior in Deployment 2 with a large number of contributors. For this purpose, we increased the number of contributors by adding 8 new videos to Deployment 2 to create a network of 16 contributors with three disconnected components. For the arrangement of the videos, we used six videos from *Vassar*, six videos from *Ballroom*, and four videos from *Exit*. The results show that the traffic reduction of the proposed scheme increases from 21% to 27% and a 5% performance improvement over Max_Correlation is achieved, as shown in Fig. 10.

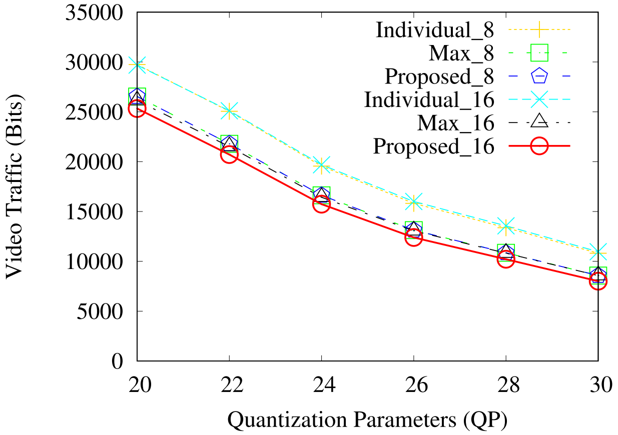


FIGURE 10. Traffic reduction in network of 16 contributors.

B. NUMBER OF REFERENCES

In terms of traffic reduction, the performances of the proposed and Multiple_References schemes are not significantly different in Deployments 1 and 2. Specifically, the proposed scheme outperforms Multiple_References by only 1% in terms of traffic reduction in both deployments. However, the main improvement of the proposed scheme over Multiple_References is that a smaller number of references is required for traffic reduction. This advantage saves the resources of the contributors' devices, such as batteries. Fig. 11 shows the maximum number of references used for differential encoding by the two schemes in the deployments of 8 and 16 contributors. As shown in the figure, Multiple_References uses a large number of references when the number of contributors increases. However, the proposed scheme selects an effective number of references based on the correlation degrees among the contributors. Specifically, Multiple_References uses up to seven references, whereas the proposed scheme uses up to four references in Deployment 1. On the other hand, in Deployment 2 with eight contributors, Multiple_References and the proposed schemes use two references and one reference, respectively. In Deployment 2 with 16 contributors, Multiple_References uses five references, whereas the proposed scheme uses only three references.

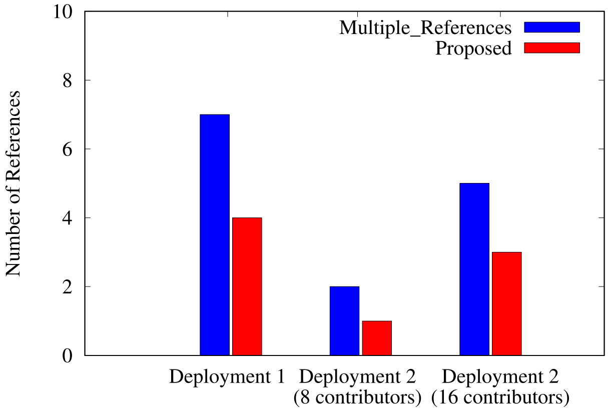
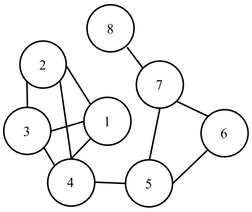


FIGURE 11. Maximum number of references used in 3 deployments.

C. EFFECT OF CORRELATION DEGREES IN THE NETWORK

According to our results, Deployment 1 achieves a better performance in terms of traffic reduction than Deployment 2. In other words, the traffic reduction is dependent on the structure of the network, which is determined by the correlation degrees among the contributors. In this case, it is desirable to evaluate the manner in which the correlation degrees among the contributors affect the traffic reduction. However, because we used the standard multi-view video sequences for our evaluation, it is difficult to construct a deployment with arbitrary correlation degrees between the contributors. In this section, we describe the evaluation of the performance of our approach using different video contents in order to observe the effect of different degrees of correlation among the contributors on traffic reduction. For this purpose, we used eight videos from the *Exit* sequence to create Deployment 3. The resulting graph is a neither completely connected nor disconnected graph, as shown in Fig. 12. In Deployment 3, our approach achieves an up to 11% traffic reduction as compared to Individual_Uploading, as shown in Fig. 13.



	1	2	3	4	5	6	7	8
1	-	0.844	0.688	0.625	-	-	-	-
2	0.844	-	0.813	0.719	-	-	-	-
3	0.688	0.813	-	0.875	0.688	-	-	-
4	0.625	0.719	0.875	-	0.781	-	-	-
5	-	-	0.688	0.781	-	0.688	0.75	-
6	-	-	-	-	0.688	-	0.719	-
7	-	-	-	-	0.75	0.719	-	0.656
8	-	-	-	-	-	-	0.656	-

FIGURE 12. Graph structure of deployment 3.

The amount of traffic reduction in Deployment 3 is less than that in Deployments 1 and 2. To explain this result, we investigated three attributes of each deployment that are derived from the correlation degrees among the contributors. These attributes are 1) the number of edges in the deployment, 2) the average correlation degree of the edges, and 3) the number of edges with a high correlation degree, that is, greater than 0.9. Table 3 shows the attributes and corresponding traffic reduction of each deployment.

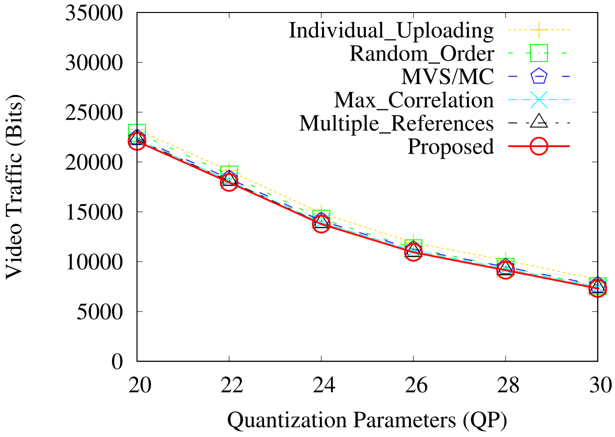


FIGURE 13. Traffic reduction in deployment 3.

TABLE 3. Attributes of the deployments.

Deployment	# of Edges	Avg. Correlation Degree	# of Edges with $\alpha_{v_i,v_j} > 0.9$	Traffic Reduction
1	28	3.29	8	31%
2	7	0.83	1	21%
3	12	0.74	0	11%

Deployment 1, which achieves the highest traffic reduction, has 28 edges, an average correlation degree of 3.29, and 8 highly correlated edges. In Deployment 2, the number of edges and the average correlation degree are significantly lower than those in Deployment 1, and thus, Deployment 2 achieves a lower traffic reduction than Deployment 1. However, Deployment 3 achieves the lowest traffic reduction among the three deployments. Although Deployment 3 has a larger number of edges than Deployment 2, it has a lower average correlation degree than Deployment 2 and there are no highly correlated edges.

In conclusion, a large number of high correlated edges in a contributor's network can result in a greater traffic reduction, regardless of the number of edges in the network.

VI. CONCLUSION

In this paper, we proposed a novel solution for uploading crowdsourced multi-view videos from mobile video contributors to a video collector. To achieve a large reduction in the volume of video traffic from the contributors together with an improvement in the video quality, our proposed scheme considers correlation-based differential encoding with multiple reference streams. By exploiting the inter-camera correlations among the captured streams, our scheme achieves a significant amount of traffic reduction, as well as quality improvement. The evaluation results show that our approach can contribute to a traffic reduction of up to 31% with a quality improvement of 2.7 dB as compared to the existing individual uploading schemes in a network of eight contributors.

REFERENCES

[1] F. Chen, C. Zhang, F. Wang, and J. Liu, "Crowdsourced live streaming over the cloud," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2015, pp. 2524–2532.

[2] Y. Wu, Y. Wang, and G. Gao, "Photo crowdsourcing for area coverage in resource constrained environments," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[3] Q. He, J. Liu, C. Wang, and B. Li, "Coping with heterogeneous video contributors and viewers in crowdsourced live streaming: A cloud-based approach," *IEEE Trans. Multimedia*, vol. 18, no. 5, pp. 916–928, May 2016.

[4] C. Zhang and J. Liu, "On crowdsourced interactive live streaming: A twitch.tv-based measurement study," in *Proc. 25th ACM Workshop Netw. Oper. Syst. Support Digit. Audio Video*, 2015, pp. 55–60.

[5] K. Bilal, A. Erbad, and M. Hafeeda, "Crowdsourced multi-view live video streaming using cloud computing," *IEEE Access*, vol. 5, pp. 12635–12647, 2017.

[6] P. Wang, R. Dai, and I. F. Akyildiz, "A differential coding-based scheduling framework for wireless multimedia sensor networks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 684–697, Apr. 2013.

[7] S. Kodera, T. Fujihashi, S. Saruwatari, and T. Watanabe, "Multi-view video streaming with mobile cameras," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 1412–1417.

[8] P. Wang and I. F. Akyildiz, "A spatial correlation-based image compression framework for wireless multimedia sensor networks," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 388–401, Apr. 2011.

[9] N. Ozbek and A. Tekalp, "Fast multi-frame reference video encoding with key frames," in *Proc. 13th Eur. Signal Process. Conf. (EUSIPCO)*, Antalya, Turkey, Sep. 2005, pp. 1–4.

[10] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. London, U.K.: Wiley, 2003.

[11] T. T. Nu, T. Fujihashi, and T. Watanabe, "Content-aware efficient video uploading for crowdsourced multi-view video streaming," in *Proc. Int. Workshop Comput., Netw. Commun. (CNC)*, Maui, HI, USA, Mar. 2018.

[12] M. Tang, S. Wang, L. Gao, J. Haung, and L. Sun, "MOMD: A multi-object multi-dimensional auction for crowdsourced mobile video streaming," in *Proc. INFOCOM*, Atlanta, GA, USA, May 2017, pp. 1–9.

[13] S. P. Venkatagiri, M. C. Chan, and W. T. Ooi, "On demand retrieval of crowdsourced mobile video," *IEEE Sensors J.*, vol. 15, no. 5, pp. 2632–2642, May 2015.

[14] H. T. Tai, W. C. Chung, C. J. Wu, R. I. Chang, and J. M. Ho, "SOP: Smart offloading proxy service for wireless content uploading over crowd events," in *Proc. 17th Int. Conf. Adv. Commun. Technol.*, Seoul, South Korea, Jul. 2015, pp. 659–662.

[15] M. Bommes, A. Fazekas, T. Volkenhoff, and M. Oeser, "Optimized upload strategies for live scalable video transmission from mobile devices," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 1059–1072, Apr. 2017.

[16] R. N. R. Wagner and R. Baraniu, "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, Sep. 2003, pp. I-597–I-600.

[17] M. Mu and C. Chen, "Collaborative image coding and transmission over wireless sensor networks," *EURASIP J. Adv. Signal Process*, vol. 2007, pp. 223–223, Dec. 2007.

[18] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[19] A. Anand, A. Balachandran, A. Akella, and S. Seshan, "Enhancing video accessibility and availability using information-bound references," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1223–1236, Apr. 2016.

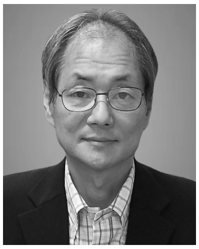
[20] A. Anand, A. Balachandran, A. Akella, and S. Seshan, "A case for information-bound referencing," in *Proc. ACM Special Interest Group Data Commun. (SIGCOMM) Hot Topics Netw. Workshops (HotNets)*, Monterey, CA, USA, Oct. 2010, p. 4.

[21] S. Kodera, T. Fujihashi, S. Saruwatari, and T. Watanabe, "Video similarity based wireless multi-view video streaming," *IPSJ Trans. DCON*, vol. 57, no. 7, pp. 1–18, Jul. 2016.

[22] *Multi-View Video Test Sequences from MERL*, document ISO/IEC JTC1/SC29/WG11, 2005.

THAN THAN NU received the B.C.Sc. and M.C.Sc. degrees in computer science from the University of Computer Studies, Yangon, Myanmar, in 2007 and 2010, respectively. She is currently pursuing the Ph.D. degree with the Intelligent Networking Laboratory, Osaka University, Japan, under the supervision of Prof. T. Watanabe. Her current research interests are in the areas of crowdsourced multi-view video uploading, especially focus on the traffic reduction and energy consumption of video transmission from mobile devices.

TAKUYA FUJIHASHI (M'16) received the B.E. and M.S. degrees from Shizuoka University, Japan, in 2012 and 2013, respectively, and the Ph.D. degree from the Graduate School of Information Science and Technology, Osaka University, Japan, in 2016. He was a Research Fellow (PD) of the Japan Society for the Promotion of Science in 2016. From 2014 to 2016, he was a Research Fellow (DC1) of the Japan Society for the Promotion of Science. From 2014 to 2015, he was an Intern at the Electronics and Communications Group, Mitsubishi Electric Research Labs. He has been an Assistant Professor with the Graduate School of Science and Engineering, Ehime University, since 2017. He selected one of the Best Paper candidates in the IEEE International Conference on Multimedia and Expo 2012. His research interests are in the area of video compression and communications, with a focus on multi-view video coding and streaming over high and low-quality networks.



TAKASHI WATANABE (S'83–M'87) received the B.E., M.E., and Ph.D. degrees from Osaka University, Japan, in 1982, 1984, and 1987, respectively. He joined the Faculty of Engineering, Tokushima University, in 1987, and moved to the Faculty of Engineering, Shizuoka University, in 1990. He was a Visiting Researcher with the University of California at Irvine, Irvine, from 1995 to 1996. He has been a Professor of the Graduate School of Information Science and Technology, Osaka University, Japan, since 2013. His research interests include mobile networking, ad hoc sensor networks, IoT/M2M networks, intelligent transport systems, specially MAC and routing. He is a member of IPSJ and IEICE. He has served on many program committees for networking conferences, IEEE, ACM, IPSJ, and IEICE.