A Study on Multi-view Video Crowdsourcing over Wireless Networks

Submitted to

Graduate School of Information Science and Technology

Osaka University

January 2020

Than Than Nu

List of Publications by the Author

I. Journals

[1] Than Than Nu, Takuya Fujihashi, Takashi Watanabe, “Soft Video Uploading for Low-Power Crowdsourced Multi-view Video Streaming,” IEICE Transactions on Communications, vol.E103-B, No.5, 2020. (Accepted)

[2] Than Than Nu, Takuya Fujihashi and Takashi Watanabe, "A Traffic Reduction Method for Crowdsourced Multi-View Video Uploading," IEEE Access, vol. 6, pp. 36544-36556, 2018.

II. International Conferences

[1] Than Than Nu, Takuya Fujihashi and Takashi Watanabe, "Power-Efficient Video Uploading for Crowdsourced Multi-View Video Streaming," 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1-7.

[2] Than Than Nu, Takuya Fujihashi and Takashi Watanabe, "Content-aware Efficient Video Uploading for Crowdsourced Multi-view Video Streaming," 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, 2018, pp. 98-104.

III. Domestic Conferences

[1] Than Than Nu, Takuya Fujihashi, Takashi Watanabe, "A Fundamental Approach for Efficient Crowdsourced Multi-view Video Uploading,” Technical Committee on Ambient Intelligence and Sensor Networks, July 2018, pp. 43-46.

[2] Than Than Nu, May Aye Khine, "A Linear Programming Model for Optimum Resource Allocation,” 2009 Fourth Local Conference on Parallel and Soft Computing (PSC 2009), pp.420-420, 2009.

IV. Awards and Scholarships

[1] Young Research Encouragement Award, "A Fundamental Approach for Efficient Crowd-sourced Multi-view Video Uploading," Technical Commitee on Ambient Intelligence and Sensor Networks, July, 2019.

[2] Scholarship for Ph.D degree in Japan, ASEAN University Network/Southeast Asia Engineering Education Development Network (AUN/SEED-Net) and Japan International Cooperation Agency (JICA), October 2016-December 2019.

To all my beloved ones

Abstract

The integration of video streams captured by many mobile video contributors at a crowded event into a multi-view video, that is, crowdsourced multi-view video streaming enables remote viewers to experience the immersive views of the event as if they were attending the event. The serviceability of crowdsourced multi-view videos streaming is not limited to entertainment purposes. As an example, it can be extended to intelligent transportation system (ITS) in which the videos captured by onboard cameras of different vehicles at one side of an intersection can be shared with the vehicles at the other side to provide the traffic condition at the intersection. Another potential application of crowdsourced multi-view video streaming is three-dimensional (3D) virtual reconstruction of an interesting location, building, or a concert using video frames to point cloud technology. Despite the diverse applicability, the crowdsourcing multi-view videos over wireless network encounters many challenges due to the resource-constrained nature of wireless networks and limited capacity of the consumer-grade mobile devices.

One of the characteristics of crowdsourced multi-view streaming is that many contributors upload the video streams captured at close spatial location simultaneously to the video collector. Thus, a large amount redundant video traffic is uploaded due to the high correlation among the video streams, which is highly inefficient usage of scared wireless network resources. In this case, traffic reduction is of paramount importance to realize the efficient of the uploading of crowdsourced video streams. On the other hand, due to the power-hungry nature of video streaming and limited battery supply of mobiles devices restrict the practicability of crowdsourced multi-view video streaming. Thus, it is demanding the traffic-reduced and low-power video uploading solutions to realize the efficient crowdsourced multi-view video streaming.

This dissertation proposes three video uploading schemes considering the above-mentioned issues. We firstly propose a traffic reduction method for multi-view video uploading from crowdsourced video contributors. 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 (IBR). 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.

Second, we propose a cluster-based redirect video uploading scheme for high-quality and low-power crowdsourced multi-view video streaming. Considering the drawbacks of conventional digital video transmission consumption our proposed scheme integrates the four approaches of network clustering, delegate selection, soft video delivery, and four-dimensional discrete cosine transform (4D-DCT) to redirectly upload the captured videos to the AP. Specifically, network clustering and delegate selection leverage the redirect path between the contributors and the AP. Soft video delivery removes power-hungry digital encoding and transmission by directly sending frequency-domain coefficients using multi-dimensional DCT and near-analog modulation. 4D-DCT exploits the content correlations between the contributors to reduce redundant transmissions.

Finally, we extend our previously proposed differential encoding-based video uploading scheme using ROI-based multi- view video encoding with the goal of further traffic reduction considering the perceptual redundancy of human visual system (HVS). To realize the ROI-based differential multi-view video uploading, we first construct a correlated network of contributors based on the correlation degrees among them. We then select a contributor with the largest average correlation in the network for extraction of ROI regions. The selected contributor uploads its video stream using ROI-based single-view video encoding while the rest of the contributors in the network overhear the transmitted stream and performs ROI-based differential multi-view video encoding with the prior knowledge ROI information.

This dissertation evaluates the performance of our proposals through computer simulations using standard multi-view video sequences. We approve the effectiveness of the proposed methods through the evaluations.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Takashi Watanabe of the Department of Information Networking in Graduate School of Information Science and Technology, Osaka University for admitting me as his Ph.D. student, for all of his valuable advices and comments on my research, and for all kindness shown to me throughout my Ph.D. life. I could never have imagined to reach my goal without his patient guidance and kind mentorship.

I would like to express my sincere gratitude to Professor Masayuki Murata, Professor Toru Hasegawa, Professor Teruo Higashino and Professor Morito Matsuoka of the Department of Information Networking in Graduate School of Information Science and Technology, Osaka University for serving as members on my dissertation committee.

I owe a debt of gratitude to Associate Professor Shunsuke Saruwatari of the Graduate School of Information Science and Technology, Osaka University for his valuable advices and discussions on my research which led my study to the correct path.

I would like to express my bottomless thank to Assistant Professor Takuya Fujihashi of the Graduate School of Information Science and Technology, Osaka University for his valuable advices and discussions on my research. I would never accomplish my Ph.D. degree without his earnest and patient guidance.

I would also like to thank the former and current secretaries of Watanabe Laboratory, Mrs. Takahashi, Mrs. Tsuda, Mrs. Nakano, Mrs. Nomura and Mrs. Takatsuka, for their countless help not only for my student life but also for my day-to-day life in Japan. Their kindness made me feel like I was with my own family.

I would like to thank the lab members of Watanabe Laboratory for their discussions and comments on my research and any single help that they have given to me throughout my stay in the laboratory.

I would like to express my deep gratitude to my scholarship organization, Japan International Cooperation Agency (JICA) for supporting me to pursue Ph.D. degree in Japan.

The special thank goes to my best friend, Dr. Ei Ei Tun, postdoctoral researcher at Chulalongkorn University, Thailand. I lovingly thank her for showing me the way and supporting

me to achieve the doctoral scholarship in Japan and for always being there for me whenever necessary.

Last but not least, the immeasurable thank goes to my family; my parents, big brother and little brother for their unconditional love and support that always give me the strength to keep moving forward.

Contents

List of Publications by the Author **i**

Abstract **v**

Acknowledgments **vii**

1 Introduction **3**

1.1 Crowdsourcing of Multi-view Videos 3

1.2 Main Contributions 5

1.3 Literature Review 6

1.3.1 Crowdsourced Video Streaming 6

1.3.2 Correlation-based Content Uploading 8

1.3.3 Power-Efficient Video Delivery 10

1.3.4 Soft Video Delivery 12

2 Fundamental Theories and Techniques **15**

2.1 Introduction 15

2.2 Video Enoding 15

2.2.1 Single-view Video Encoding 16

2.2.2 Multi-view Video Encoding 20

2.3 Human-aware Video Encoding 24

2.3.1 ROI based Video Encoding 25

2.3.2 ROI-based Multi-view Video Encoding 28

2.4 Video Streaming over Wireless Networks 29

2.4.1 Conventional Video Delivery 30

2.4.2 Soft Video Delivery 32

2.5 Summary . . . 34

3 Traffic Reduction for Crowdsourced Multi-view Video Uploading 35

3.1 Introduction . . . 35

3.2 Proposed System . . . 39

3.2.1 System Model and Assumptions . . . 39

3.2.2 Timing Diagram . . . 40

3.2.3 Information-Bound Reference Calculation . . . 42

3.2.4 Reference Selection . . . 43

3.2.5 Transmission Order Determination . . . 44

3.2.6 Encoding . . . 46

3.3 Evaluation . . . 47

3.3.1 Setup . . . 47

3.3.2 Results and Analysis . . . 51

3.4 Discussion . . . 53

3.4.1 Effect of Large Number of Contributors . . . 53

3.4.2 Number of References . . . 54

3.4.3 Effect of Correlation Degrees in the Network . . . 55

3.4.4 Overhead of Content Uploading and Transmission Order Determination 56

3.4.5 Discussion on Required Bandwidth . . . 57

3.5 Summary . . . 58

4 Soft Video Uploading for Low-Power Crowdsourced Multi-view Video Streaming 61

4.1 Introduction . . . 61

4.2 Proposed Scheme . . . 64

4.2.1 Overview . . . 64

4.2.2 Network Clustering . . . 66

4.2.3 Video Uploading . . . 67

4.3 Performance Evaluation . . . 69

4.3.1 Effect of Cluster-based Redirect Uploading . . . 69

4.3.2 Video Quality in Different Channel Quality . . . 72

4.3.3 Transmission Power Consumption 74
4.3.4 Video Quality in Reduced Transmission Rate 75
4.3.5 Discussion on Power Saving 76
4.4 Discussion on Non-correlated Contributors 78
4.5 Summary 78
5 ROI-based Crowdsourced Multi-view Video Uploading 85
5.1 Introduction 85
5.2 Proposed System 88
5.2.1 Overview 88
5.2.2 Correlation Estimation 89
5.2.3 ROI-Reference Selection 89
5.2.4 Proposed System 90
5.3 Evaluation 91
5.3.1 Settings 91
5.3.2 Results 92
5.4 Summary 93
6 Conclusion 95
6.1 Overall Summary 95
6.2 Future Works 96

List of Figures

1.1 Examples of crowdsourced video streaming from a mobile phone. 4

1.2 An end-to-end crowdsourced multi-view video streaming system. 5

2.1 Video Encoding and Decoding on Transmission. 17

2.2 Block diagram of single-view video encoding. 18

2.3 Ilustration of spatial and temporal redundancies of *claire* video sequence. [source: http://ise.stanford.edu/Video/claire.qcif.gz] 20

2.4 Transformation of 8×8 block of image using 2D-DCT. 2D-DCT gathers low-frequency coefficients at the upper left corner and high frequency coefficients at the lower right. 21

2.5 Illustration of inter prediction of P-frame and B-frame. 22

2.6 Block diagram of multi-view video encoding. 22

2.7 A typical multi-view video architecture. 23

2.8 Motion Estimation and Disparity Estimation. 23

2.9 Encoding structures of multi-view video transmission. 24

2.10 Loss of details in the sky area is more sensitive to human vision system than that in mountain ranges area. 26

2.11 Example of human attention on specific regions of an image. [source: http://ise.stanford.edu/Video/news.q

2.12 Conventional digital-based video delivery vs. soft video delivery 30

2.13 Framework of SoftCast. 32

2.14 3D-DCT of 4 frame GOP and chunk discarding. 32

3.1 System model ($C_{i=1,...3}$ = contributors). 40

3.2 Timing diagram of video uploading. (BK: Backoff time; yellow boxes: receiving packets; blue boxes: transmission packets). 41
3.3 Generating image information-bound reference. 43
3.4 Effect of random transmission order. 46
3.5 Correlation graph and corresponding dependency graph. 46
3.6 Encoding behaviors of three correlated contributors. 47
3.7 Graph structures of network of contributors. 49
3.8 Video traffic at different quantization parameters. 50
3.9 Video quality vs. traffic. 51
3.10 Traffic reduction in network of 16 contributors. 53
3.11 Maximum number of references used in 3 deployments. 54
3.12 Graph structure of Deployment 3. 56
3.13 Traffic reduction in Deployment 3. 56

4.1 System model. (C_n: contributors) 64
4.2 Timing diagram of proposed video uploading system 65
4.3 Four-dimensional decorrelation. 67
4.4 Deployment of cameras and channels between the contributors and the AP. . . . 70
4.5 Video quality of the proposed and conventional SoftCast schemes with correlated contributors as a function of channel quality difference between direct and redirect paths. 71
4.6 Video quality of the proposed and conventional SoftCast schemes with non-correlated contributors as a function of channel quality difference between direct and redirect paths. 79
4.7 Video quality of the proposed and conventional digital-based schemes as a function of wireless channel SNRs between the delegate and the AP. 80
4.8 Number of transmission symbols required by the proposed and conventional digital-based schemes as a function of received PSNRs. 81

4.9 Video quality of the proposed and conventional digital-based schemes as a function of cutoff rates in deployment 1. 82

4.10 Total uploading time spent by the proposed and conventional digital-based schemes as a function of overhead times in the proposed scheme. 83

4.11 Video quality of the proposed and conventional digital-based schemes with non-correlated contributors as a function of wireless channel SNRs between the delegate and the AP. 83

5.1 System model ($C_{i=1,...3}$ = contributors). 89

5.2 ROI extraction and tracking. 90

5.3 Prediction structure for differential encoding. 90

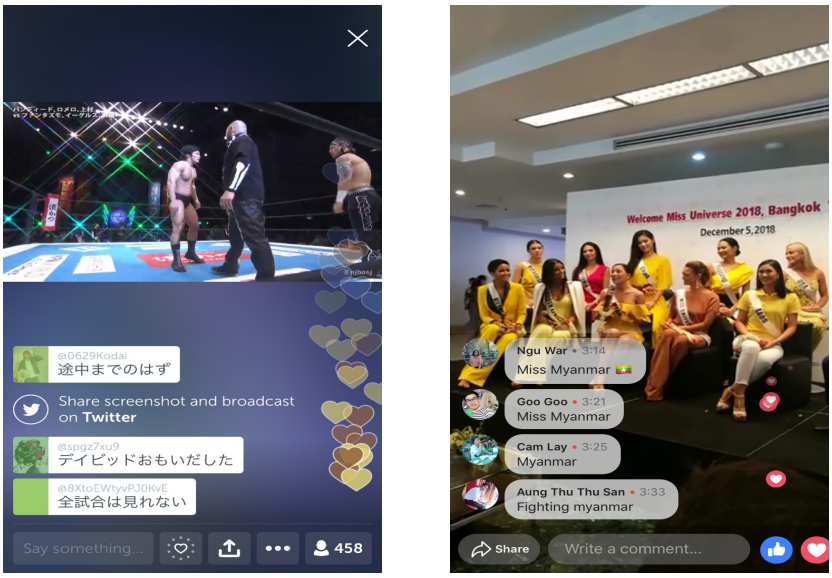
Chapter 1

Introduction

1.1 Crowdsourcing of Multi-view Videos

In June 2006, Howe, contributing editor of Wired Magazine, introduced the term *crowdsourcing* and literally defined as *outsourcing of a business function to the crowds, to acquire the creative ideas or find the best solutions with an incentivizing scheme* [2]. Typically, the crowdsourcing acquries the data and intelligent from the public by the use of Information and Communication Technology (ICT). The prime examples of crowdsourcing include Threadless.com, InnoCentive.com, Amazon's Mechanical Turk, and iStockphoto.com as mentioned in Howe's article. Other representative examples count the Wikipedia, YouTube, Flickr and open-source software platforms. In fact, the earlier practices of crowdsourcing that use ICT were featured as Web-oriented ones and the term *crowds* generally regarded as Desktop-based online communities [3].

In nearly past two decades, along with the proliferation of smartphones and ubiquitous mobile communication technologies, the model of crowdsourcing has been evolved to mobile crowdsourcing [4]. Through the sensors equipped in the smarphones such as accelerometers, compasses, GPSs, microphones, and cameras, it can collect and analyze the data of humans in mobility and environments, and then provide useful information and services to the end users. Another facet of mobile crowdsourcing is crowdsourced video streaming. Empowered by the smartphones with powerful cameras and today's trend of social networking, the crowdsourced video streaming allows the remote viewers to see what was happening in the world through others' eyes, i.e., crowdsourced video contributors. The most famous Internet platforms, which provide crowdsourced video streaming services, are Facebook Live, Youtube, Periscope, and



(a) IMPACT Westing live stream on Periscope

(b) An event of Miss Universe 2018 on Facebook Live.

Figure 1.1: Examples of crowdsourced video streaming from a mobile phone.

Twitch [5, 6, 7, 8, 9]. Fig. 1.1 illustrates two examples of crowdsourced video streaming of popular events via Twitter's Periscope and Facebook Live mobile applications.

Crowdsourced multi-view video streaming [6] is an extension of crowdsourced video streaming in which many contributors viewing the same event provide different viewpoints of the event at various angles. The integration of those videos into a multi-view video allows the remote viewers to enjoy the immersive viewing of the event as if they were at the event. The serviceability of crowdsourced multi-view videos streaming is not limited to entertainment purposes. As an example, it can be extended to intelligent transportation system (ITS) in which the videos captured by onboard cameras of different vehicles at one side of an intersection can be shared with the vehicles at the other side to provide the traffic condition at the intersection. Another potential application of crowdsourced multi-view video streaming is three-dimensional (3D) virtual reconstruction of an interesting location, building, or a concert using video frames to point cloud technology. Despite the convincing benefit, there are many challenges to realize the crowdsourced multi-view video streaming. Fig. 1.2 depicts an end-to-end system of crowdsourced multi-view video streaming which includes three main parts: video delivery from the crowdsourced contributors, i.e., uploading, encoding/transcoding of the streams at the streaming

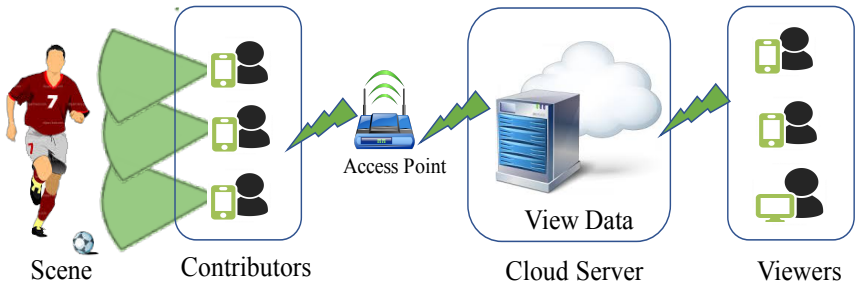


Figure 1.2: An end-to-end crowdsourced multi-view video streaming system.

server such as cloud, i.e., processing, and the delivery of the streams to the viewers, i.e., downloading. In the literature, a large number of researches have been working on crowdsourced video streaming, however, mainly focusing on downloading part. Accordingly, the uploading part of crowdsourced video streaming, especially multi-view videos, is still well unexplored [10]. To fulfill this gap, in this dissertation, we focus on the video uploading of crowdsourced multi-view video streaming.

There are two distinguishable features of crowdsourced multi-view video streaming from the conventional crowdsourced video streaming. First, in crowdsourced multi-view video streaming, a large number of contributors reside in the same geographic area, i.e., in an event, and share the limited network resources at the same time. Second, the videos captured at the same event are highly correlated leading to the redundant transmissions which consumes large amount of scarced network resources. Another challenge of crowdsourced multi-view video streaming is that mobile devices are provided by limited battery sources which is unfavorable for power-hungry video transmission. Considering the above-mentioned challenges, this dissertation proposes the means of efficient uploading of a large amount of video traffic from crowdsourced contributors within the limited network resources in power-efficient manner.

1.2 Main Contributions

Generally, video streaming over wireless link has its requirement of QoS, i.e., high video quality and low latency video delivery. To this end, the solutions proposed for high quality video streaming always monitor and control the QoS parameters of the video delivery from video encoding and/or the video transmission perspectives. As an example, the study presented

in [11] focuses on video compression optimization based on the scalable video coding and video content chunking to improve the video quality and reduce the delay. The study reported in [12] jointly controls the parameters of video encoding and transmission for energy-efficient video communication from mobile devices.

Similarly, this dissertation proposes the solutions for high quality multi-view video uploading from crowdsourcing contributors. To this end, the proposed schemes consider both video encoding and video transmission aspects. Specifically, the proposed schemes exploit the correlation and channel quality differences among contributors by utilizing the differential encoding and soft video delivery approaches to realize traffic reduced, high-quality, and low-power video uploading schemes.

The main contributions in this dissertation are as follows:

- Chapter 3 presents a traffic reduction method based on the content-aware differential encoding in crowdsourced multi-view video uploading. By exploiting the captured content correlation among the crowdsourced contributors, the proposed scheme achieves the traffic reduction with quality improvement.

- Chapter 4 presents a redirect soft video uploading scheme for high quality and low-power multi-view video uploading from crowdsourced contributors. By skipping power-hungry digital video encoding and exploiting the channel variations among the contributors, the proposed scheme achieves the high-quality video uploading with low-power requirement.

- Chapter 5 presents a region-of-interest based traffic reduction scheme by extending the proposed method in Chapter 3. By considering the features of human-vision system, the proposed scheme achieves the further traffic reduction.

1.3 Literature Review

1.3.1 Crowdsourced Video Streaming

The term crowdsourced video delivery can be roughly divided into two categories: uploading of the streams from the contributors to the server, such as a cloud, and downloading of the contents

by the remote viewers. In the study presented in [13], the authors 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. Hung 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. The authors of [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 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 [7] 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, Kashif *et al.* [16] proposed a generic framework for crowdsourced multi-view live video streaming, namely, Cloud-based Multi-View Crowdsourced Streaming (CMVCS). As in [7], the authors 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 proposed methods presented in this dissertation consider the collection of the videos from all the contributors, aiming for the applications of the multiple-viewpoints video or 3D

space of a scene. However, in some situation, it may not be neccessary to collect all the videos at the same time, considering the popularity of the certain views, the quality of the captured views, available resources of the wireless network and mobile devices themselves. For example, in [11], the authors 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. 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 [6] 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.

On the other hand, 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. Due to this, this dissertation considers 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.

1.3.2 Correlation-based Content Uploading

One of the solutions of removing the redundancy, i.e., traffic reduction, in crowdsourced multi-view video steaming is to exploit the content correlation among the video streams. Kodera *et al.* [17] considered the correlation-based traffic reduction for crowdsourced multi-view video streaming 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. The authors of [18]

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 [19] 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 [20] and [21], image processing methods were applied to estimate the correlation among images from neighboring sensors in order to conduct collaborative transmission. In the study in [20], 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 [22]. In the method described in [21], 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 [17] could be violated if cameras in close proximity to each other project in different orientations. Overlapped FoV-based correlation estimation [18, 19] 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) [23,24]. As compared to the methods in [20] and [21], 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 [19] and [18], the authors 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, the proposed scheme presented in Chapter 3 considers multiple dependencies among the cameras by exploiting all the correlations among them. In the method described in [17], two reference streams are used for bidirectional encoding based on the positions of the cameras, regardless of the actual correlation among them. In contrast, the proposed scheme effectively selects the number of references for a contributor for differential encoding based on the correlation degrees.

1.3.3 Power-Efficient Video Delivery

The power consumption in video delivery is mainly dependent on two factors: video encoding and video transmission.

Video Encoding

The conventional digital-based video compression such as H.264/AVC consumes a large amount of power due to its complexity. To realize power-efficient video streaming, previous researches tackled to reduce the complexity of the video encoding. The authors of [25] proposed an algorithm-level encoder complexity reduction approach. It reduces the computational complexity of the 4×4 block intra-prediction based on partial sampling prediction and symmetry of adjacent angle modes. As the system-level reduction of complexity, the study in [26] presented a framework which allocates the computational power of the encoder adaptive to video contents and also scales with the available battery power using a Region Of Interest (ROI) classification method. Raha *et al.* [27] proposed a hardware-oriented approach for power-efficient video encoder. They considered the reconfigurable approximate computing architecture adaptive to the characteristics of the input videos. In [28], the authors presented a game-theoretic power-efficient scalable video encoding (PSVE) for resource-limited systems. The PVSE allocates the power levels to four modules involved in the encoding such as ME so as to maximize the utility of each module, which is measured by the bit rate, visual quality and power consumption.

On the other hand, the proposed scheme presented in Chapter 4 completely replaces the

power-hungry digital video encoding process with a simple linear transform of soft video delivery to save the encoding power consumption from the mobile contributors.

Video Transmission

Apart from the video encoding, the wireless video transmission can drain the large portion of energy from streaming devices. For energy-efficient video communication, the authors of [12] introduced an approach for minimizing the power consumption on video transmission by jointly optimizing the power consumption on video encoder, channel encoder, and the transmitter. Li *et al.* [29] proposed joint optimization of video summarization, coding and modulation schemes, and packet transmission in order to minimize the power consumption on video transmission. Ye Li *et al.* [30] proposed cross-layer schemes for energy-efficient video transmission by adjusting parameters on physical (PHY) layer (e.g. modulation level, bit rate) and medium access control (MAC) layer (e.g., the buffer status). They also proposed an optimal smoothing algorithm to reduce both the peak data rate and radio front-end power consumption. In [31], Othmane *et al.* proposed a cross-layer designed framework for network lifetime extension in wireless video sensor networks based on H.264/AVC video encoding. At the application layer, the proposed framework adaptively adjusts its configuration on the basis of feedbacks from network layer to meet the desired quality while extending the network lifetime. They also proposed an energy-efficient and reliable routing protocol that considers the information from MAC layer such as the packet loss rate and the available buffer size in nodes' multi-priority queues, and the remaining energy.

All of the above studies considered the adaptation of video encoding parameters to network conditions for saving the transmission power consumption under the certain video quality. However, the adaptation costs additional power consumption especially in unstable wireless network environments [32]. In contrast to those studies, the proposed scheme in Chapter 4 uses soft video delivery to cope with even unstable network conditions for better video quality with low transmission power consumption. In addition, soft video delivery within the cluster reduces the encoding and transmission power requirement, thus saving the power consumption from the members of the cluster with no quality degradation. Since the delegate does not require

adaption of encoding parameters, soft video delivery saves the transmission power and achieves the graceful quality improvement.

1.3.4 Soft Video Delivery

In traditional video delivery, digital-based video encoding and transmission adopt a scheme of transmitting fixed source data rate over a known channel. Consequently, it suffers from a threshold effect [33]; if the channel quality drops below a threshold, video quality drops severely; even if the channel quality improves beyond the threshold, the video quality is constant due to the quantization distortion. To overcome such cliff effect and constant quality effect, soft video delivery schemes have been recently proposed. A pioneering work among these schemes is SoftCast [34, 35]. SoftCast removes non-linear quantization, entropy coding and channel coding, instead, uses a technique that directly maps linear-transformed signals based on 3D-DCT to channel symbols and assigns transmission power for the error protection. The main advantage of SoftCast is its graceful video quality improvement in wireless broadcast/multicast scenarios in which the received video quality at each receiver is proportional to its channel properties. Motivated by the virtues of SoftCast, many researches including [36, 37, 38, 39, 40, 41] have been introduced for various scenarios. ParCast (Parallel video uniCast) [36] was designed for unicast video delivery over the Multiple input Multiple output (MIMO)-Orthogonal Frequency Division Multiplexing (OFDM) links. AirScale [37] focused on antenna heterogeneity of video multicast users in Multiple User (MU)-MIMO systems. Dcast [38] employed the principle of distributed video coding (DVC) with analog transmission in video multicast. A study called FoveaCast [39] applied SoftCast's features of linear transforms and power-adjustable source components to the reduction of perceptual redundancy with graceful video quality in wireless image broadcast/multicast scenarios. The studies in [40] and [41] considered the similar goal by using SoftCast-based video transmission for free viewpoint and 360-degree video delivery. All of the above studies showed the prosperity of soft video delivery in wider domains for improving the received video quality and transmission efficiency. Beside those benefits, soft video delivery has the great potential of energy saving because of its simplicity for the power-restricted devices in both video encoding and video transmission.

Soft video delivery in above studies is designed for the direct video transmission between a server/sender and a single or multiple receivers and showed the prosperity of soft video delivery for improving the received video quality and transmission. Difference from those studies, the study presented in Chapter 4 applies soft video delivery to crowdsourced multi-view video uploading considering redirect video transmission between the contributors and the AP. Although the conventional soft video delivery with direct transmission suffers low video quality due to unstable direct path, the proposed scheme maintains video quality by utilizing redirect path between the contributors and the AP based on the network clusteirng.

Chapter 2

Fundamental Theories and Techniques

2.1 Introduction

This dissertation focuses on realizing the efficient video uploading from mobile contributors for crowdsourced multi-view video streaming. To this purpose, we have proposed three video uploading schemes as we describe in detail in the later chapters. In this chapter, we describe the fundamental theories and techniques applied in our studies.

2.2 Video Enoding

In 1948, Claud Shannon mathematically defined the amount of information existed in a source, which produces a certain type of message, and established the foundation of information theory [42]. This theory put forward some fundamental limits within which the communication channel can carry the certain amount of information, known as channel capacity. According to Shannon's capacity limit, the maximum rate of information that we can transmit over a communication channel without any error is bounded by the channel bandwidth, signal power level and noise power level. In wireless communication system, the transmission channel is inherently prone to error due to multipath effects, signal fading and noise. To successfully convey the information over a band-limited and error-prone channel with arbitrarily small error probabilities, source encoding and channel encoding is necessary. Source encoding, also known as data compression, is a task of removing the redundant information present in a source in order to save the storage and bandwidth. When the source is a sequence of images, i.e., the encoding is termed as video encoding or compression.

Traditionally, videos in its raw representation are extremely voluminous to store and transmit. As an example, an uncompressed, raw, Ultra High Definition (UHD) video has a resoultion of 3840×2160 pixels at 60 frames per second (fps). Considering the 10 bit/pixel for 3 color components, the transmission bandwidth of $3840 \times 2160 \times 60 \times 10 \times 3 = 14.92$ gigabits per second (Gbps) would be needed [43]. Unfortunately, it is almost impossible to transmit such huge amount of data over today's internet bandwidth of at most a few tens of hundred megabits per second (Mbps) without any processing to reduce the data volume. In this case, video encoding or compression [44] becomes necessary.

Since the first digital video technology coding standard, known as H.120 standardized by International Telecommunication Union (ITU) in 1984, many video encoding standards have been developed and in use today [45]. Currently, the most common video coding standard is H.264/Advanced Video Coding (AVC), ratified in 2003 by the Joint Video Team (JVT) of the ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations. H.264/AVC is widely used by Internet streaming applications like YouTube and Netflix, video player software like Flash Player, and various High Definition Television (HDTV) broadcasts over terrestrial, cable, and satellite channels [46]. These applications undertake the encoding of single video source which originates from the single camera providing the single viewpoint of a scene, i.e., single-view video encoding. Nowadays, the emerging three-dimensional (3D) video technology have laid the way for immersive 3D applications such as free viewpoint video (FTV), 3DTV and and immersive video conferencing [47, 48]. The 3D scenes rendering in these applications are portrayed by new scene representation techniques called multi-view videos [49, 50]. A standard of multi-view video coding (MVC) is amended to H.264/AVC [51] to support the multi-view videos. The fundamental concepts of single-view video coding and multi-view video coding are presented in the following sections.

2.2.1 Single-view Video Encoding

The work of video processing on transmission comprises of two fundamental components: encoding and decoding as illustrated in Fig: 2.1. In a single view video encoding, the sender

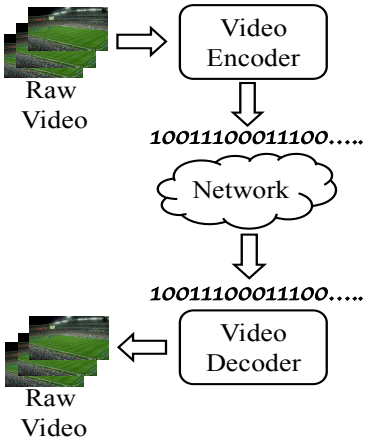


Figure 2.1: Video Encoding and Decoding on Transmission.

converts the uncompressed video originating from a source into a format that takes up less capacity before transmission. At the receiver, the compressed video is converted back into its original format by a process called decoding. Quite evidently, there are diverse ways of compressing a video source and modern-day video encoding standards including H.264, H.265, and VP9, define the universal syntax of an encoded bitstream and a process of decoding this syntax. Despite the numerous compression methods, the process of video encoding includes three complementary modules, i.e., prediction, transform and quantization, and entropy coding.

In general, video compression is the process of removing redundancies in the video which can be broadly classified as *statistical redundancy* and *psycho-visual redundancy* [43]. The statistical redundancy refers to the intrinsic redundancy in which the values of closely located pixels in one frame or across the successvie frames are significantly similar due to the inherent nature of world scene. As an illustration, Fig. 2.3 shows the frame 1 and frame 2 of *claire* video sequence [1]. As we can see in the left figure, the pixels of the news annoucer's dress, her face and those in background area are very similar. Similarly, the pixels in frame 1 and frame 2 are almost identical, except the little changes in facial expression around the annoucer's lips. Such kind of similarity in Fig. 2.3(a) is known as *spatial redundancy* and that in Fig. 2.3(b) is known as *temporal redundancy*. We say that spatially/temporally redundant pixels are strongly correlated with each other.

The objective of the **prediction** is to remove such correlation that can minimize the required number of bits to represent the video frames. When the prediction is carried out on a single frame

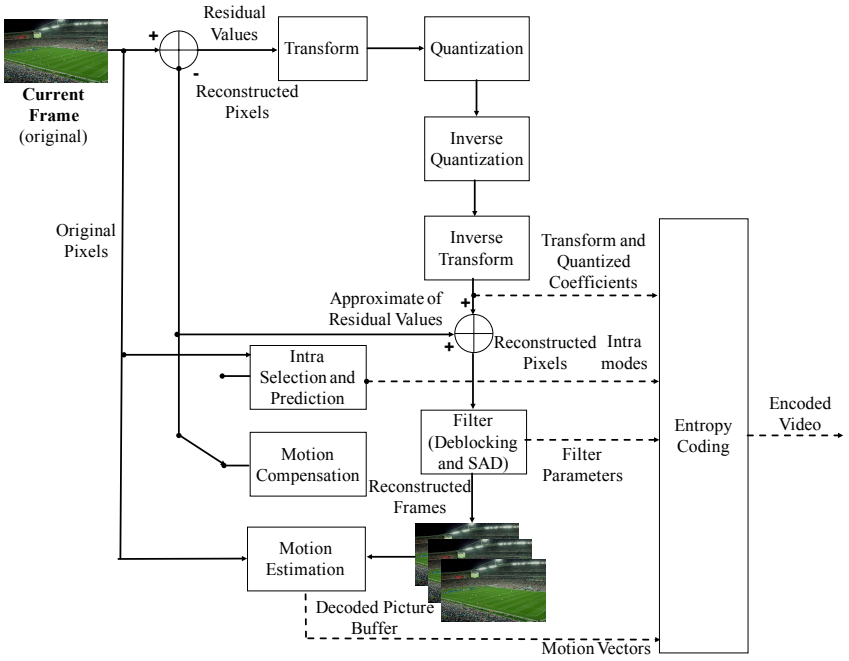


Figure 2.2: Block diagram of single-view video encoding.

for removing the spatial correlation, it is termed as intra prediction, the frame itself is labeled as intra-predictive frame or I-frame and encoding of the frame is termed as intra frame encoding. Usually, the first frame in the video sequence is independently encoded as I-frame. In contrast, when the prediction uses the previously encoded frame or frames as references to exploit the temporal correlation in the successive frames, it is termed as inter-prediction, inter-predictive frame or P-frame, and inter frame encoding, respectively. P-frame only uses forward prediction, i.e., using the frame as reference that comes temporally before it. When the inter-prediction uses the references frames which are temporally before or after the current frame, it is termed as bidirectionally predictive frame or B-frames. Fig. 2.5 depicts the example of inter prediction of P-frame and B-frame in encoding of sequence of frames. The inter-predictive encoding uses the differential encoding. That is, the current frame is compared with the previously encoded frame which is used as a reference to exploit the temporal correlation. And, it calculates the difference between their pixels values, known as *residual*, and encodes only the residual to reduce the frame size. For inter prediction, various techniques such as *motion estimation* and *motion compensation* are widely used.

After prediction is finished, the residual frames go to a series of processing states called **transform** and **quantization**. The transform state converts the residual frames into a different represention known as frequency domain. Several transforms including *discrete consine transform (DCT)*, *discrete sine tranform (DST)* and *Hadamard transform* are widely used in image/video encoding. The most popular transform is block-based two-dimensional DCT (2D-DCT) which is defined as follows:

$$F(\alpha,\beta) = \frac{1}{4}C(\alpha)C(\beta)\sum_{i=1}^{N}\sum_{j=1}^{N} f(i,j)\cos\frac{(2i+1)\alpha\pi}{2N}\cos\frac{(2j+1)\beta\pi}{2N} \tag{2.1}$$

$$c(\epsilon) = \begin{cases} \sqrt{\frac{1}{2}} & (\epsilon = 0) \\ 1 & otherwise \end{cases} \tag{2.2}$$

where $f(i,j)$ is a pixel value in each block and $F(\alpha,\beta)$ is a frequency coefficient, i.e., DCT coefficient.

Just as the block of pixels in an image shows the strong spatial correlation, the residual blocks also show the same characteristic with the smaller values. The objective of transform state is to provide decorrelation of the pixels of the residual frames to achieve maximum energy compaction. What the energy compaction means is the phenomenon of concentration of the frequency, i.e., energy of the pixels. After the transformation, the frequencies of the pixels are concentrated on the lower frequency components with respect to the higher frequency components. Fig. 2.4 shows the values of 8×8 block of pixels and its corresponding frequency coefficients after 2D-DCT.

After transformation, the frequency components are quantized using a quantization table. That is, all the DCT coefficients are divided by the values in quantization table to reduce the range of values of the coefficients. The divisors in the table are derived from Quantization Parameter (QP) which is the key parameter to control the compression efficiency and visual quality of encoding process.

Finally, **entropy coding** is performed to produce a bitstream. The main philosophy behind the entropy coding is that the information of any symbol is inversely proportional to the likelihood, i.e., probabilities of their occurrence. Thus, the most occurring symbols are assigned fewer bits

(a) Frame 1 (b) Frame 2

Figure 2.3: Ilustration of spatial and temporal redundancies of *claire* video sequence. [source: http://ise.stanford.edu/Video/claire.qcif.gz]

and the least occurring symbols are allocated more bits. Entropy coding focuses on minimizing the number of bits needed to encode the information symbol. In other words, it minimizes the *coding redundancy*, another form of statistical redundancy. The extensively used entropy coding schemes include *variable length coding* and *binary arithematic coding*.

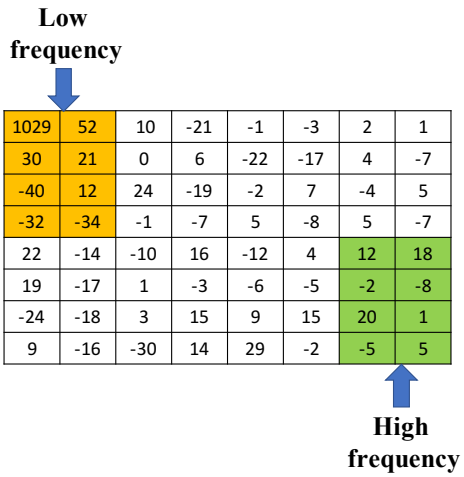
2.2.2 Multi-view Video Encoding

The multi-view videos are the basic elements of 3D video applications like free viewpoint video (FTV), 3DTV and immersive video conferencing [47, 48]. Fig. 2.7 illustrates a typical multi-view video architecture in which multiple synchronized cameras are arranged in relatively close positions to capture the scene in different perceptive. Due to this arrangement of cameras, the video captured in multi-view video environment exhibit the high similarities, i.e., strong correlation among the views.

The encoding structure of multi-view video transmission varies with the adopted encoding scheme. The most primitive scheme for multi-view encoding is H.264/AVC *simulcast* coding (SCC) [52]. In SCC, video frames in each viewpoint is independently encoded using single-view digital encoding like H.264/AVC. However, SCC is inefficient since it does not exploit the inter-view redundancy, and thus, generates the large amount of redundant video traffic. To overcome this problem, multi-view video coding (MVC), which is multi-view extension of H.264/AVC, known as H.264/MVC, is adopted [53]. Fig. 2.12(a) and 2.12(b) shows the prediction structures of SCC and MVC, respectively.

126	138	135	118	118	126	126	130
150	168	161	122	105	109	100	118
150	150	126	150	142	126	126	117
150	161	168	130	134	150	138	130
130	118	134	142	157	142	117	126
115	117	108	117	101	99	117	126
122	130	130	138	117	108	108	138
142	118	134	117	109	91	126	109

(a) Pixel values in 8×8 block



(b) Energy compaction after 2D-DCT

Figure 2.4: Transformation of 8×8 block of image using 2D-DCT. 2D-DCT gathers low-frequency coefficients at the upper left corner and high frequency coefficients at the lower right.

Multi-view video encoding basically follows the same procedures as the single-view video encoding. That is, it adopts the procedures of prediction, transform and quantization and entropy coding. The main difference is that the multi-view video encoding uses the reference frames from its adjacent viewpoints for prediction to exploit the inter-view temporal redundancy. In other words, multi-view video coding incorporates the inter-frame prediction as well as inter-view prediction in prediction phase to reduce the amount of traffic.

Traditional inter-frame prediction employs the motion estimation which exploits the similarities between the frames in a single viewpoint while the inter-view prediction utilizes *disparity estimation* [54] to exploit the similarities between frames in different viewpoints. The key idea of disparity estimation is to find the best matching of the current microblock in the reference frame within the reference view. Conceptually, the task of disparity estimation is quite similar to that of motion estimation.

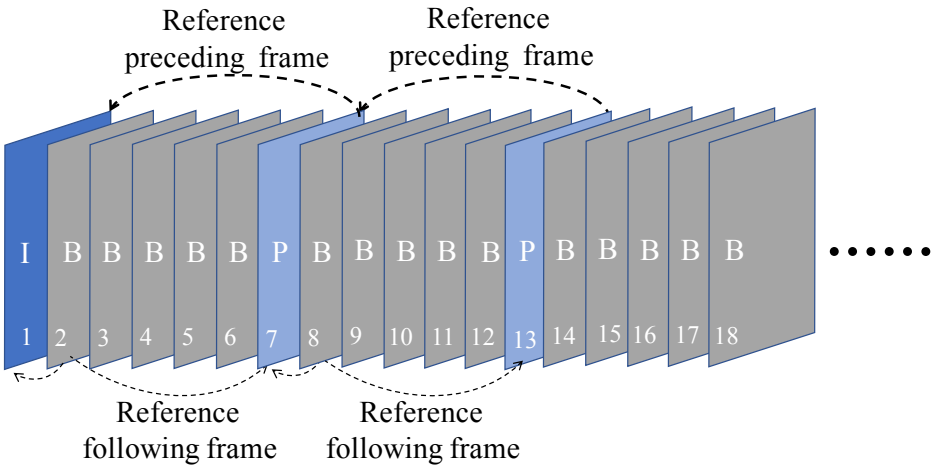


Figure 2.5: Illustration of inter prediction of P-frame and B-frame.

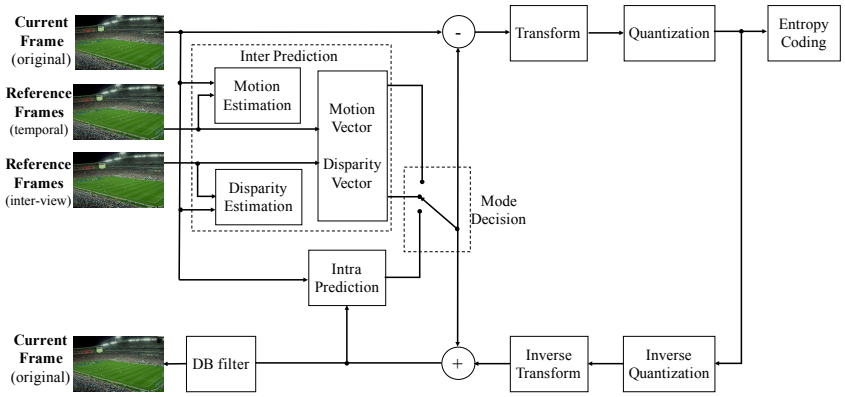


Figure 2.6: Block diagram of multi-view video encoding.

In motion estimation, the best matching of current microblock is searched within the predefined *search range* (SR) in the temporal reference frames. Once the matching block is found, the corresponding *motion vector* is generated. Similarly, disparity estimation searches for the best matching cadidate for current block within the reference frame of reference view, and once found, points the location of the block by a disparity vector. Since the inter-frame prediction is made across the successive frames of the same view, the motion vector denotes the displacement of a particular object in the scene with time. In contrast, the disparity vector represents the displacement of the given object across the views. Fig. 2.8 outlines the motion and disparity estimation realized in multi-view video coding.

After prediction is completed, transform and entropy coding phases follow as in H.264/AVC that completes the encoding process.

In conventional multi-view video coding, the selection of reference view is based on the

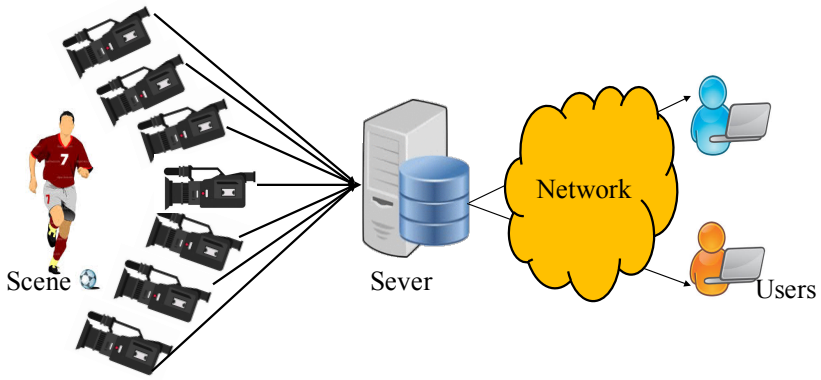


Figure 2.7: A typical multi-view video architecture.

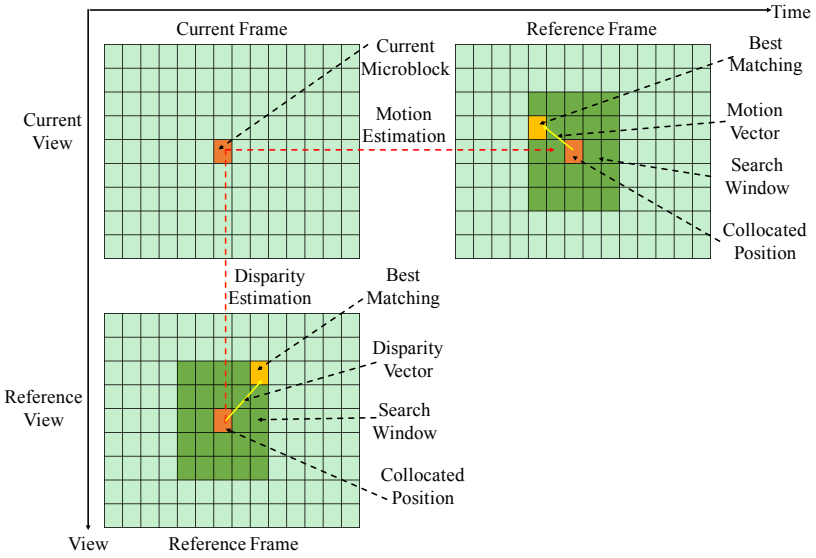
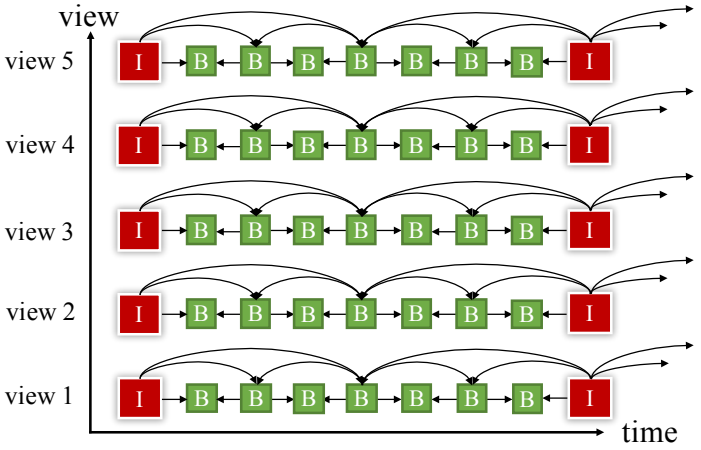
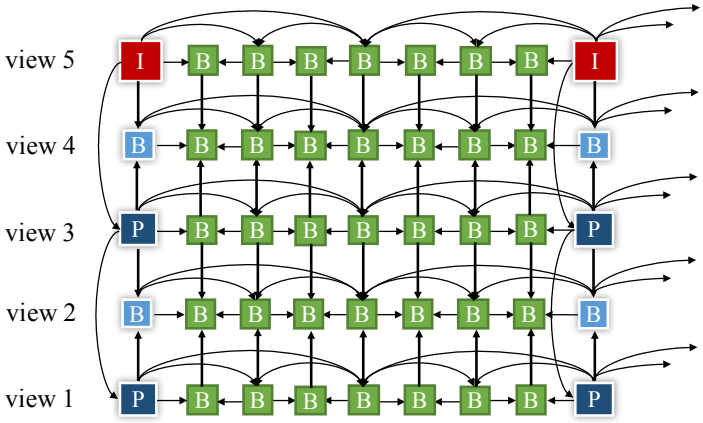


Figure 2.8: Motion Estimation and Disparity Estimation.

camera position. That is, the viewpoints take their neighboring views as references. For example, in 2.12(b), the first transmitted view, i.e., view 5, is encoded individually using single-view video coding and served as the reference for its closet adjacent viewpoints, i.e., view 3 and 4, to achieve the better prediction and thus, better compression gain. On the other hand, crowdsourced multi-view environment, the correlation among the viewpoints cannot be guaranteed with the positions of the cameras due to the possible varying capturing directions. In this case, a mechanism for deciding the reference views for each viewpoint is a critical requirement which we address in the later chapters.



(a) Simulcast



(b) H.264/MVC

Figure 2.9: Encoding structures of multi-view video transmission.

2.3 Human-aware Video Encoding

The typical video encoding essentially exploits the statistical redundancy, i.e., spatial, temporal, and coding redundancy using the techniques of prediction and entropy coding to effectively achieve the compression gain. To handle the pyscho-visual redundancy, also known as *perceptual redundancy*, it considers two important realities of human vision system (HVS). The first is that human vision is much more perceptive to luminance (brightness) information (luma) than chrominance (color) information (chroma) in the natural scene. To exploit this reality, subsampling mechanism is used by subsampling the chroma components by half while preserving the full resolution of luma components.

The second is that human eye is more sensitive to small changes in luminance over a smooth

wider area, i.e., low frequency luminance, but less sensitive to fast luminance changes (high frequency luminance). As an example, in Fig. 2.10, the lack of details in low frequency sky area can be easily detected while changes in high frequency mountain ranges area are tolerable by human eye. To this reality, transform coding and quantization takes the responsibility. The transform coding converts the luma and chroma components from pixel domain to frequency domain in which the components are grouped in the order of increasing frequencies. The quantization maintains low-frequency components in prioritized manner but selectively ignores high frequency components without affecting the visual quality.

On the other hand, these techniques do not fully exploit the perceptual redundancy of human vision system. In fact, human perceptual mechanism contains diverse aspects such as contrast sensitivity, visual masking, foveation, visual attention and multi-modality of attention [55]. According to the visual attention models, the behavior of human vision system is stimulated by the particular characteristics of the image, such as intensity, size, color and location of the objects. In other words, the visual perception of human vision system is content-variant, that is, it can concentrate on the particular contents of the image such as a human face. The regions which contain

Figure 2.11 shows some possible interesting areas of an image which human attention concentrates on. The ability of attention to specific areas which contain their interested objects so called region-of-interest (ROI) introduces the considerable perceptual redundancy in less interested regions. By exploiting this fact, ROI-based video encoding has been proposed and implemented in standard video codecs such as H.264/AVC. In the following sections, we will describe the ROI-based video encoding and how it can be extended in multi-view videos.

2.3.1 ROI based Video Encoding

The concept of ROI is employed in several image and video coding standards such as JPEG 2000 [56] and Fine Granularity Scalability (FGS) Profile of MPEG4-part2 [57]. In addition, a feature of H.264/AVC by which each microblock in a frame can be encoded using different quantization parameters gives a space for ROI-based encoding concept to join with it. Regardless of the coding standards, the core idea of ROI-based video encoding is rather straightforward.

(a) Original Frame

(b) Low Quality Frame

Figure 2.10: Loss of details in the sky area is more sensitive to human vision system than that in mountain ranges area.

It enhances the visual quality in human interested region (ROI region) by sacrificing the visual quality in peripheral region (non-ROI region) in the image or video frame. The key components of ROI-based video coding include ROI detection and tracking and ROI-based rate control.

The detection of ROIs is important task in ROI-encoding and can be done by several ways such as visual attention models, object and face detection models [58]. According to visual attention model, the ROIs of an image often include semantic objects such as human face, a car, a popular building and so on. On the other hand, ROI detection can be application-dependent. Specifically, ROI in the applications like video conferencing and internet video chatting may simply be human face which can be detected by using a face detection algorithms. However, in video surveillance application, the security personnel are more possibly interested in human movement in the video. In the latter case, ROI detection may require more advanced techniques such as human motion detection. For locating the specific objects in an image and

Figure 2.11: Example of human attention on specific regions of an image. [source: http://ise.stanford.edu/Video/news.qcif.gz]

extracting the areas containing them, i.e., ROI detection, and tracing the movements of the objects, i.e., ROI tracking, image processing tools are widely used. The common techniques for object detection and tracking includes background subtraction, optical flow analysis, and spatio-temporal filtering.

Face detection is a special class of objection detection. A lot of proposal has been reported for face detection in the literature. The method proposed in [59] utilizes the shape recognition algorithm considering the particular features of a face the skin color segmentation and contour evaluation. The work in [60] improves [59] by further combining other silent features such as color, saturation and orientation to improve the detection accuracy. On the other hand, object tracking or movement detection can be carried out with the help of video compression tools such as motion estimation. For example, the proposed method in [61] estimates the motion of traces of a given object using the motion features, which are estimated based on the microbloack motion vectors generated by motion estimation process.

After the extraction of ROI region in a frame, the ROI mask, a binary image of which pixels values inside are ROI are set to 1 and those outside of ROI are set to 0. Using the ROI mask, ROI-based rate control is performed to optimize the video quality in both ROI areas and non-ROI areas.

2.3.2 ROI-based Multi-view Video Encoding

The study in [62] extended the ROI-based encoding to multi-view video coding. The multi-view data plus depth (MVD) data format consists of the multi-view video (MVV) and multiple associated depth maps. These depth maps of MVD can be used to facilitate ROI extraction.

[62] proposed **Depth based ROI extraction** (DBDE) to extract ROI for single frame by using motion, texture and depth information of MVD. It includes spatial ROI-extraction, temporal ROI extraction and inter-view ROI extraction. Given the color image **F** and depth map **D**, firstly, motion mask, M^m, foreground mask M^f, contour and texture information, M^c, and discontinuity of the depth map, M^d are extracted. Secondly, a characteristic ROI region, $M^f \cap [M^m \cup M^d]$ is constructed as the seed of ROI depth plane. And the depth map of the resulting region is divided into different depth planes D^z. Meanwhile ROI contours are constructed by integrating foreground, motion, depth and color contours as $M^f \cap [M^m \cup M^d \cup M^c]$. Next, 8×8 block based contour recovery and noise elimination are performed to get more accurate ROI contours, M^l. Finally, background regions in depth planes D^z is excluded using M^l that finishes the ROI extraction process. A macroblock (MB)-level ROI mask is created based on the extracted ROI for the block-based multi-view video coding (MVC).

To provide a fast ROI regions extraction at the time successive frames i.e., **Temporal ROI extraction**, ROI regions in the current frame are predicted by using the temporal preceding extracted ROIs. Suppose $W_{k,t}$ and $W'_{k,t}$ be rectangle and predictive windows of the k-th ROI in the frame at time t. They predict $W'_{k,t}$ from DP-ROI windows of the previous p frames. The DBDE algorithm is performed to refine DP-ROIs within the predictive windows. The areas out of the predictive windows are directly set as background. Regarding the motions in DP-ROI prediction, a window size scaling coefficient correlated with motion magnitude, λ_ϕ is adjusted to guarantee DP-ROI is located within $W'_{k,t}$.

Since multi-view videos are captured at the same scene at different angles, inherent correlations exist among them. This correlation can be used to extract the ROI regions in view dimensions i.e., **Inter-view ROI extraction**.

Suppose $M = (X, Y, Z)$ be a point of ROI in the world coordinate system and $m_i = (x_i, y_i)$

be the coordinate of a pixel, which is projected from M, on i-th plane of the n_c-camera array. Also, let $\overline{M}$ and $\overline{m}$ be augmented vectors of M and m, i.e., $\overline{M} = (X, Y, Z, I)$ and $\overline{m_i} = (x_i, y_i)$, I_i is the pixel value on i-th image plane projected from I. Once ROI of the k-th view has been extracted, the ROI in i-th view can be generated as

$$Z_i\overline{m_i} = Z_k A_i R_i R_k^{-1} A_k^{-1}\overline{m_k} - A_i R_i R_k^{-1} t_k + A_k t_k, \tag{2.3}$$

where Z is the depth value, A_i is a matrix of intrinsic parameters of the i-th camera, R and t are rotational and translational matrices. According to the Eq. (2.3), the m_i is determined by m_k and Z_k, thus it can be expressed as $\overline{m}_i = f(\overline{m}_k, Z_k)$, where f is a short version of the Eq. (2.3). Accordingly, the neighboring point $\overline{m}'_i = (x_i + l_x, y_i + l_y, I'_i)$ of point $\overline{m}_i$ can be calculated as

$$\overline{m}'_i = f(\overline{m}'_k, Z'_k) \tag{2.4}$$

where $\overline{m}'_k = (x_k + l_x, y_k + l_y, I'_k)$ is the neighboring coordinate of $\overline{m}_k$, and Z'_k the neighboring depth value of Z_k.

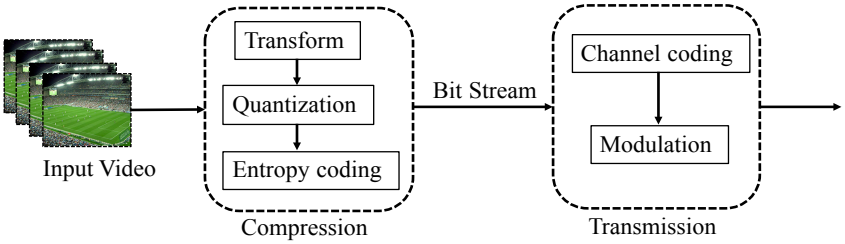
On the other hand, because depth map is relatively smooth and with high spatial correlation among neighboring pixels in the interior region of DP-ROI, $\overline{m}'_i$ can be directly calculated by the Eq. (2.5) instead of the Eq. (2.4).

$$\begin{aligned} &\overline{m}'_i \approx \overline{m_i} \\ &\text{subject to } |l_x| \le T_x, |l_y| \le T_y \end{aligned} \tag{2.5}$$

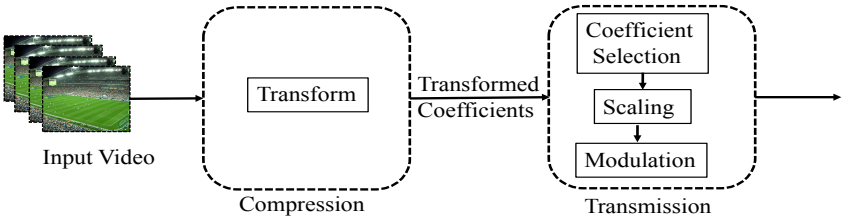
Finally, small holes are filled by applying averaging filter. DP-ROI is blocklized into MB, and MB-wise DP-ROI masks are generated for block-based MVC.

2.4 Video Streaming over Wireless Networks

The obtrusive advancement of wireless networking such as cellular networks and wireless local area networks (WLANs)) and communication technologies, e.g., 4G, 5G has popularized the progression of video services in wireless mobile networks. Accordingly, the explosive growth of video streaming over wireless networks has been witnessed in recent times, embracing the diverse applications such as live video streaming, video conferencing, video-on-demand



(a) Conventional Video Delivery



(b) Soft Video Delivery

Figure 2.12: Conventional digital-based video delivery vs. soft video delivery

(VoD) and Internet Protocol television (IPTV). On the other hand, increasing demands of wireless resources, e.g., bandwidth, from enormous number of streaming users and inherent characteristics of wireless links, e.g., fluctuation and fading, make the video delivery over wireless networks challenging.

The conventional design of wireless video transmission tackles the problem in two separate ways; digital video encoding and channel encoding. In general, the former is handled by video compression part and meant for bandwidth saving. The latter is by transmission part and meant for error protection of the encoded data during transmission. Despite the prominent use, the conventional video transmission suffers from two drawbacks; cliff effect and leveling effect [63]. To overcome the drawbacks, analog-based video delivery known as soft video delivery has been proposed in the literature [34, 35, 33, 36]. In contrast to the separate conventional design, soft video delivery adopts a unified design that both encodes the video for compression and for error protection. Fig. 2.12 depicts the separate design of video compression and transmission of conventional video delivery and its analogy with soft video delivery.

2.4.1 Conventional Video Delivery

In conventional video delivery, the digital video compression and transmission operate separately. The video compression part uses video encoder such as H.264/AVC to generate a

compressed bit stream through the various processing such as quantization and entropy coding. In essence, the digital video encoder such as H.264/AVC employs the rate-distortion optimization in which the compression bit rate, that is, the amount of data required to encode the video, is regulated with the loss video quality, that is, amount of distortion. For example, in quarter pixel-precision motion estimation, the encoder may add the extra precision to the motion of a microblock to improve the quality with the cost of additional bits required to encode the motion vector to a higher precision. Rate-distortion optimization measures both the deviation from the source (uncompressed video) and the extra bit cost so that adding extra bits is worth with the resulting increased quality. The deviation from the source is usually measured as the mean squared error (MSE) so that peak-signal-to-noise ratio (PSNR), a typical video quality metric, is maximized.

The transmission part uses a channel coding and a modulation scheme to protect and transmit the encoded bit stream. The channel coding typically comprises picking an optimal transmission rate associated with a corresponding forward error correction code (FEC), such as block codes and convolutional codes [64]. The transmission rate determines the number of bits that can be transmitted within a given period of time. As the transmission rate decreases due to the lower channel quality, the distortion due to source coding i.e., video compression, increases. In that case, the video compression part can adjust its encoding parameters to transmission rate, which is known as joint source and channel coding. However, the conventional design requires the sender to predict the channel quality of the receiver to choose the optimal rate. This task is easier in point-to-point communication in which the characteristics of the channel slowly vary. However, it is more difficult in multicast and broadcast scenarios in which different users observe different channel qualities.

Moreover, the separation of video compression and error protection causes the conventional digital-based video delivery suffer two downsides. First, the bitstream generated by the digital encoder is highly sensitive to the channel errors. When the channel signal-to-noise ratio (SNR) falls under a certain threshold, the received video quality abruptly drops since the decoder cannot decode the unsynchronized bitstream due to the bit errors occurred in the fallible wireless channel. This phenomenon is known as cliff effect. Besides, even when the channel SNR

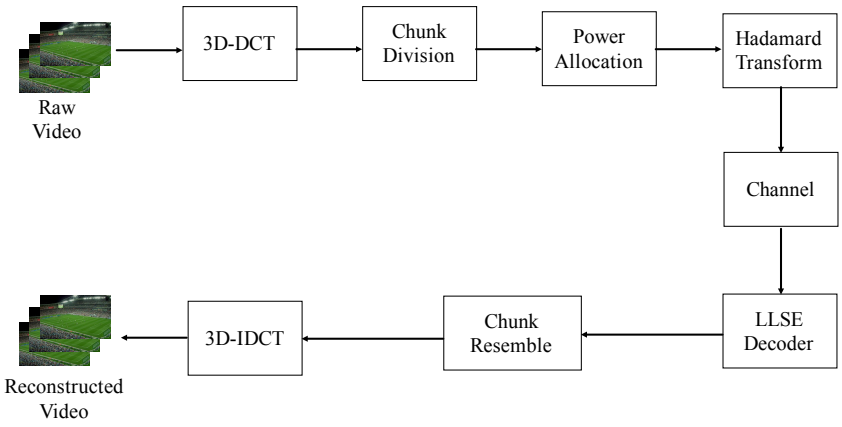
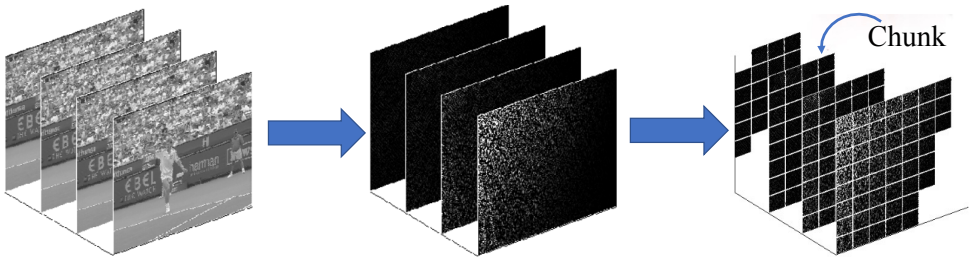


Figure 2.13: Framework of SoftCast.



(a) 4 frames GOP (b) 3D-DCT of GOP (c) Discarding zero-valued chunks

Figure 2.14: 3D-DCT of 4 frame GOP and chunk discarding.

improves, the received video quality cannot improve accordingly since the loss in quantization and entropy coding is unrecoverable at the receiver. Such unwavering video quality is termed as leveling effect.

To overcome the cliff and leveling effects of digital-based video delivery, soft video delivery schemes have been proposed in the literature.

2.4.2 Soft Video Delivery

The pioneering work among soft video delivery schemes is known as SoftCast [34]. In SoftCast scheme, the sender directly transmits the linear-transformed coefficients of a video signal over the noisy wireless channel and allocates the transmission power to the transformed coefficients conforming to their energy for the error protection. As a result, SoftCast allows the receiver to decode the video signal in accordance with the wireless channel quality, avoiding the cliff-effect and leveling effect. Fig. 2.13 illustrates the framework of SoftCast.

The design of SoftCast consists of the following components: compression, error protection, resilience to packet loss, and decoding. In **compression**, SoftCast compresses a video by applying a three-dimensional decorrelation transform, such as the 3D-DCT to remove redundant information within a frame as well as across frames, exploiting the spatial and temporal correlation in a GoP. Fig. 2.14(a) and 2.14(b) show 4 frames GOP and its 3D-DCT. The gray level in DCT frames represents the magnitude of the DCT coefficient in that pixel location. It is noted that most of the coefficients are zero, i.e., black, value which has no information. One can discard those coefficients safely without affecting the quality. Since SoftCast only transmits the non-zero DCT components, it is required to send the location of the discarded coefficients which incurs large amount of metadata. To reduce the amount of metadata, SoftCast clusters the nearby DCT coefficients into *chunks* as shown in Fig. 2.14(c). For the **error protection**, SoftCast performs power allocation to the DCT coefficients based on their magnitude. Considering the chunk-based transmission, power allocation is carried out based on scaling the magnitude of the chunks, i.e., sum of squares of DCT components in each chunk. Let $\check{s}_i \in \mathbb{R}$ be a power allocated coefficient of i-th chunk in a frame. Each DCT coefficients in a chunk is scaled by a power scaling factor g_i for noise reduction:

$$\check{s}_i = g_i \cdot c_i, \tag{2.6}$$

where $c_i \in \mathbb{R}$ is the DCT coefficient i-th chunk. The near-optimal value g_i to minimize the mean square reconstruction error is obtained as follows [34]:

$$g_i = \lambda_i^{-1/4} \sqrt{\frac{P}{\sum_k^{N_{GOP}} \lambda_k}}, \tag{2.7}$$

where P denotes a total transmission power budget, λ_i is the power of i-th chunk, and N_{GOP} is the number of coefficients in one GOP.

After power allocation, each chunk is assigned to packets. For **resilience to packet loss**, SoftCast ensures that each packet has equal importance by transforming the varying-energy chunks into equal-energy slices. SoftCast generates these slices by multiplying the chunks with the Hadamard matrix to redistribute the energy [65]. After packetization, packets are transferred to the physical layer and transmitted over the wireless links. In the **decoding** process, SoftCast

estimates the original DCT coefficients by using Linear Least Square Estimator (LLSE) with the knowledge of statistics of DCT component sent as metadata, and statistics of the channel as follows:

$$X_{LLSE} = \Lambda_x C^T (C\Lambda_x C^T + \Sigma)^{-1} \hat{Y} \tag{2.8}$$

where X_{LLSE} is the LLSE estimate of original DCT coefficients; $\hat{Y}$ is the received values; Λ_x is a diagonal matrix whose diagonal elements are the variances, λ_i, of the individual chunks; C is the encoding matrix and Σ is a diagonal matrix where its diagonal elements are the channel noise power. After obtaining the DCT coefficients in a GoP, the decoder reconstructs the original frames by taking the inverse of the 3D-DCT.

2.5 Summary

This chapter briefly describes the fundamental theories and techniques applied in this dissertation. In the three proposed schemes for crowdsourced multi-view video streaming, the techniques of multi-view video encoding, ROI-based encoding and soft video delivery are leveraged.

Chapter 3

Traffic Reduction for Crowdsourced Multi-view Video Uploading

3.1 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 [5] to remote viewers. Well-known service providers for crowdsourced video streaming services include Meerket, Periscope, and YouNow [5, 6, 7, 8]. Crowdsourced multi-view video streaming [16] 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 [19] and [17]. 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 [17] 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 [19] and [18], 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 [66, 67], 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 chapter, 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 [68], 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.

3.2 Proposed System

3.2.1 System Model and Assumptions

Fig. 5.1 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. All the contributors can reach the collector via one-hop communication and no communication errors between the collector and contributors occur.

The correlations between the contributors vary due to the variations of the cameras, their positions and capturing angles, the resolution of the videos, and so on. In addition, the clock drifts of different mobile cameras can incur the temporal differences of captured timestamps of the frames, which affects the correlation exploiting across the contributors [69]. Considering those issues, he following assumptions are made so that the high correlation among the contributors can be obtained. Contributors are closely located with each other, capturing the scene with the same type of cameras. The video resolutions of the contributors are the same. All the cameras are clock-synchronized via Global Position System (GPS)-based or Network-based synchronization protocols such as Network Time Protocol (NTP), with the synchronization accuracy to a fraction of the duration of a frame, for example within 1 milliseconds. Regarding the mobility, we assume that the degree of correlation between the contributors is the same during each unit interval of

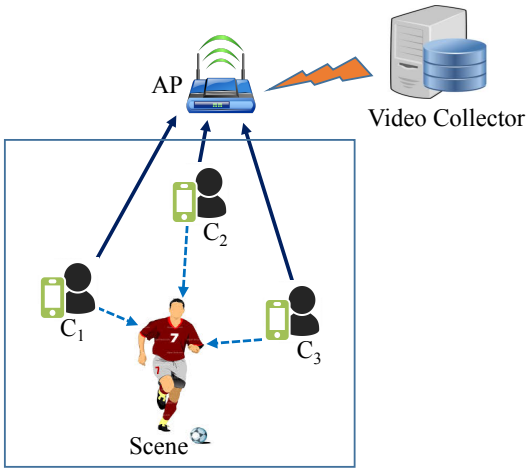


Figure 3.1: System model ($C_{i=1,...3}$ = contributors).

video transmission, that is, one group of pictures (GOP).

3.2.2 Timing Diagram

Fig. 3.2 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.

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.

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 3.2.3. Using the IBR of each contributor, the AP estimates the correlation degrees among the contributors.

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

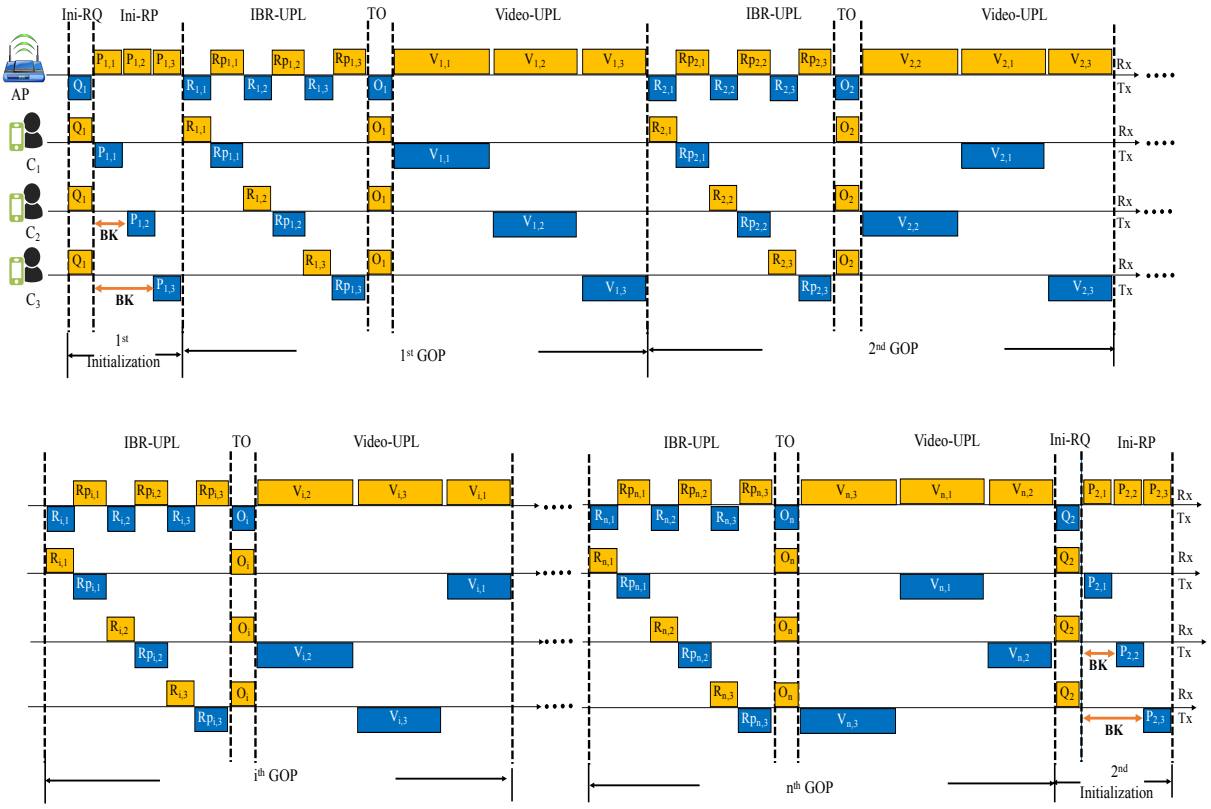


Figure 3.2: Timing diagram of video uploading. (BK: Backoff time; yellow boxes: receiving packets; blue boxes: transmission packets).

information of transmission slots and overhearing slots for each contributor.

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. The time duration of 1GOP is upper-bounded to the frame rate of the uploading video streams and size of the GOP. For example, to upload 1GOP (10 frames) of the video with the frame rate of 25 frames per second (fps) in real-time, the upper-bound duration is 0.4 second, i.e., 400 milliseconds. This time duration encompasses of three terms; 1) IBR calculation and uploading from each contributor, 2) transmission order determination and notification, and 3) video uploading from each contributor. 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.

3.2.3 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 [23,24]. According to [23], 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 [24]. 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. 3.3. 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. Given the IBRs, the AP computes the correlation coefficient, α_{v_i,v_j}, between any two contributors using

$$\alpha_{v_i,v_j} = 1 - \frac{d_{\mathrm{i,j}}}{d_{\max}}, \quad 0 \leq \alpha_{v_i,v_j} \leq 1 \tag{3.1}$$

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

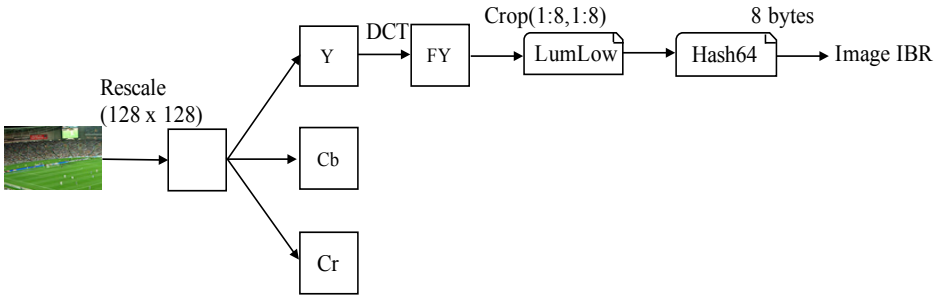


Figure 3.3: Generating image information-bound reference.

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.

3.2.4 Reference Selection

For differential encoding with multiple references, we proposed two reference selection methods in our previous work [68], namely, Maximum_Correlation and Multiple_References. In Maximum_Correlation, a contributor i selects one of its correlated neighbors j with the maximum correlation degree, $\text{argmax}_j(\alpha_{i,j})$, 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, $\beta_{\text{multi\_ref}}$ and $\beta_{\text{single\_ref}}$, to effectively select the references for a contributor. $\beta_{\text{multi\_ref}}$ is the threshold of the correlation degree that can improve the coding gain from encoding with multiple references, whereas $\beta_{\text{single\_ref}}$ is the minimum correlation degree that can benefit for differential encoding. In this study, we set $\beta_{\text{single\_ref}}$ to 0.625, because no coding gain results from differential encoding [70]. For $\beta_{\text{multi\_ref}}$, we empirically selected the correlation degree threshold of 0.875. If two or more streams having a correlation degree above $\beta_{\text{multi\_ref}}$ 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 $\beta_{\text{single\_ref}}$, for differential encoding with a single reference, as in Maximum_Correlation.

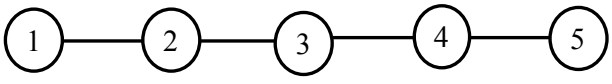
3.2.5 Transmission Order Determination

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. 3.4 shows the effect of the transmission order on the number of individual contributors. Fig. 3.4(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. 3.4(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. 3.4(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

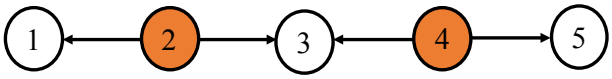
to reduce the overall video traffic. To achieve this, we designed an algorithm, described in Algorithm 1, for transmission order determination. Table 3.1 lists the notations used in the algorithm and their descriptions.

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 $V = \{v_i,\ i = 1, 2, ..., |\mathrm{V}|\}$ 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 3.2.4. 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. 3.5(a).

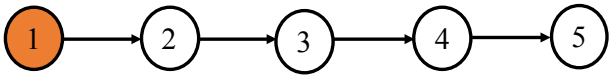
Second, the algorithm constructs a directed graph, G', based on the correlation graph to determine the dependencies among the contributors and to generate the transmission order. To construct graph G', we traverse the correlation graph in the depth-first-search manner and convert each undirected edge 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. 3.5(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).



(a) Network of correlated contributors

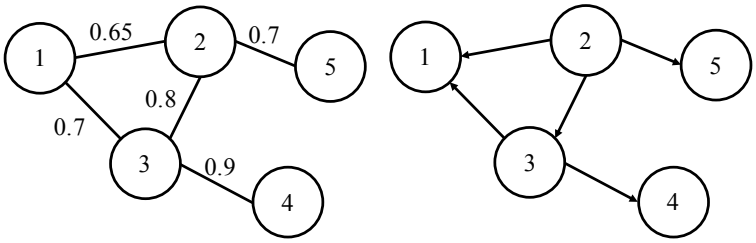


(b) Transmission order with two individually uploaded contributors



(c) Transmission order with one individually uploaded contributor

Figure 3.4: Effect of random transmission order.

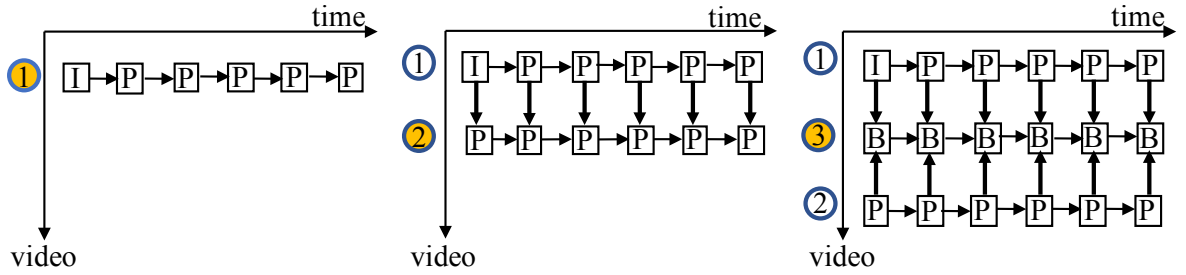


(a) Correlation graph. (b) Dependency graph

Figure 3.5: Correlation graph and corresponding dependency graph.

3.2.6 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. 3.6. In Fig. 3.6(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. 3.6(b). In Fig. 3.6(c), Contributor 3 encodes its video differentially by taking the overheard streams from its two predecessors, Contributors 1 and 2.



(a) Individual encoding (b) Differential encoding with one reference (c) Differential encoding with multiple references

Figure 3.6: Encoding behaviors of three correlated contributors.

Table 3.1: 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
$\text{cal\_Correlation}(\text{IBR}_{v_i}, \text{IBR}_{v_j})$	Calculate the correlation coefficient between contributors i and j
$\text{find\_MultiRefNodes}(\beta_{\text{multi\_ref}})$	Find the nodes whose correlation degrees is greater than $\beta_{\text{multi\_ref}}$

3.3 Evaluation

3.3.1 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.

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

$$\text{PSNR} = 10 \log_{10} \frac{(2^L - 1)^2}{\varepsilon_{\text{MSE}}}, \tag{3.2}$$

where L is the number of bits used to encode pixel luminance (typically eight bits), and ε_{MSE} is

Table 3.2: 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

the mean squared error (MSE) between all pixels of the decoded and the original videos.

Video Sequence: We used videos from the standard multi-view video sequences known as *Vassar*, *Ballroom*, and *Exit* [71] at 25 frames per second and in quarter common intermediate format (QCIF) with 176×144 resolution.

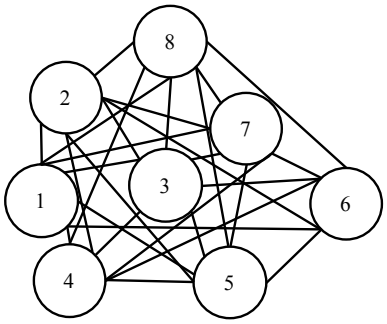
Deployment of Contributors: The deployment of the contributors is considered to reflect some uploading scenarios of crowdsourced multi-view video streaming. For example, at some point of the event, all the contributors may capture the same viewpoint of the event such as goal area in a soccer game, which causes the high correlation among the contributors. At other point, the contributors may capture their interesting viewpoints differently. In this case, not all contributors in the network would have correlation with each other. Considering such situations, we arrange the video sequences in two different fashions.

- Deployment 1 considers the first situation in which all the contributors capture the same viewpoint, resulting in a high correlation among the contributors and creating a fully connected network of contributors.

- Deployment 2 considers the second situation in which the contributors capture three different viewpoints, making three different groups. Each group constitutes the contributors with high similarities of captured videos, however, some contributors have no correlation.

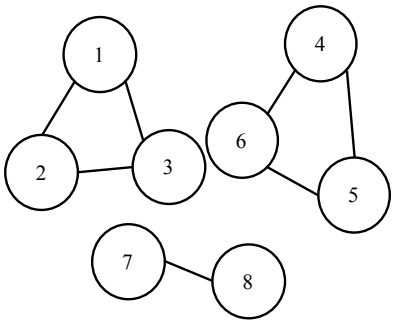
The arrangement of the video sequences in the two deployments is expressed in Table 3.2. In addition, the graph structures of the deployments and the correlation degree between the contributors are illustrated in Fig. 3.7. The length of 1 GOP is set to 10 frames. Finally, we ran the simulations using 10 different quantization parameters from 20 to 30.

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



	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) Deployment 1



	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) Deployment 2

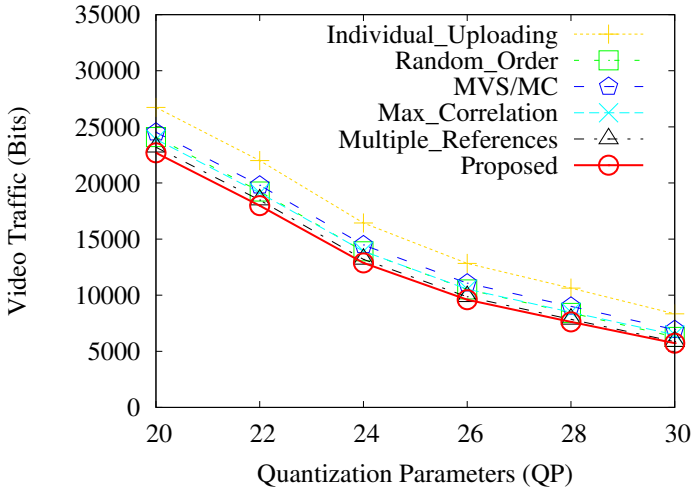
Figure 3.7: Graph structures of network of contributors.

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.

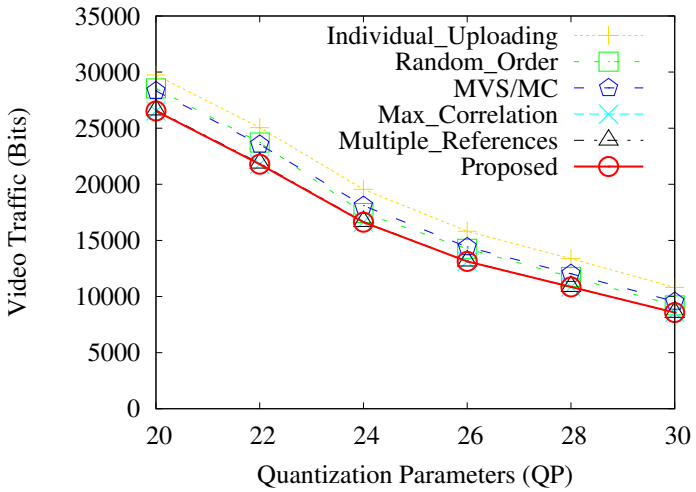
2. Max_Correlation : Max_Correlation is one of our previously proposed methods. This method conducts differential encoding considering only the maximum correlation degree between the contributors to construct the correlation network.

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.

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.



(a) Deployment 1

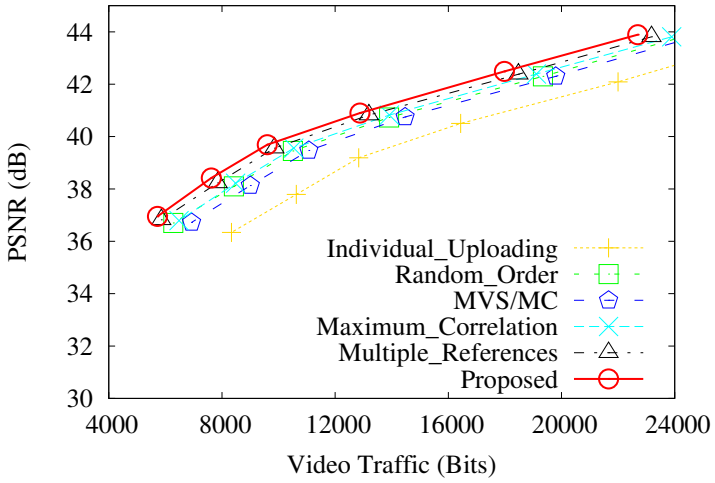


(b) Deployment 2

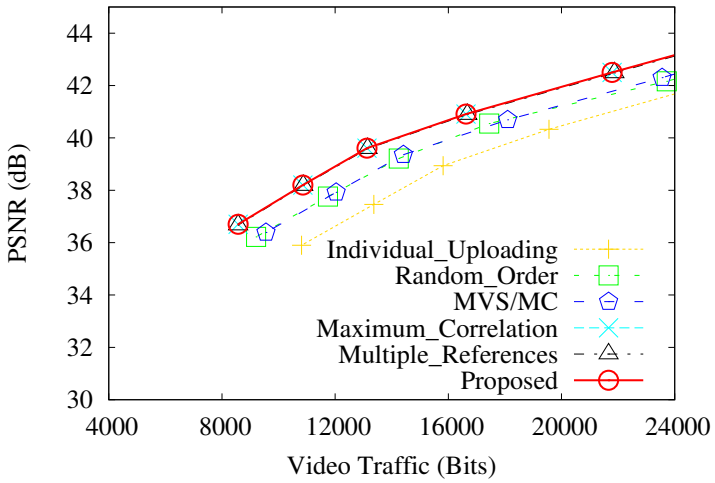
Figure 3.8: Video traffic at different quantization parameters.

5. MVS/MC : MVS/MC [17] 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 5.2.



(a) Deployment 1



(b) Deployment 2

Figure 3.9: Video quality vs. traffic.

3.3.2 Results and Analysis

Video Traffic

Fig. 3.8 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 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.

Video Quality

Fig. 3.9 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

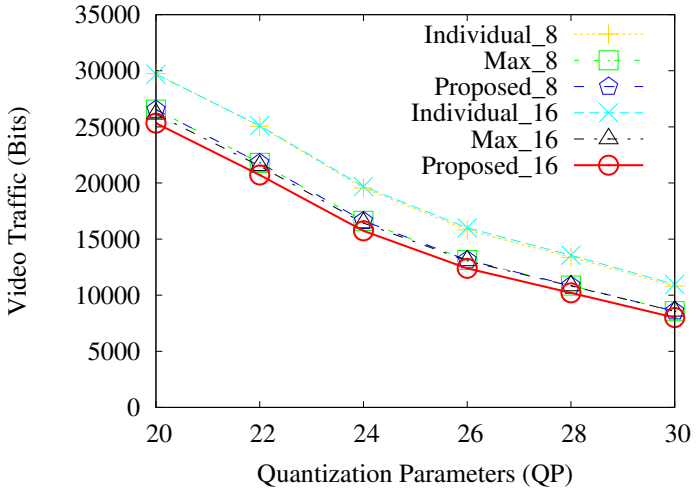


Figure 3.10: Traffic reduction in network of 16 contributors.

quality as compared to MVS/MC and Random_Order by 0.9 dB and 1 dB, respectively.

3.4 Discussion

3.4.1 Effect of Large Number of Contributors

The evaluations described above were based on deployments for eight video sequences. As shown in Fig. 3.8(b) and Fig. 3.9(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. 3.10.

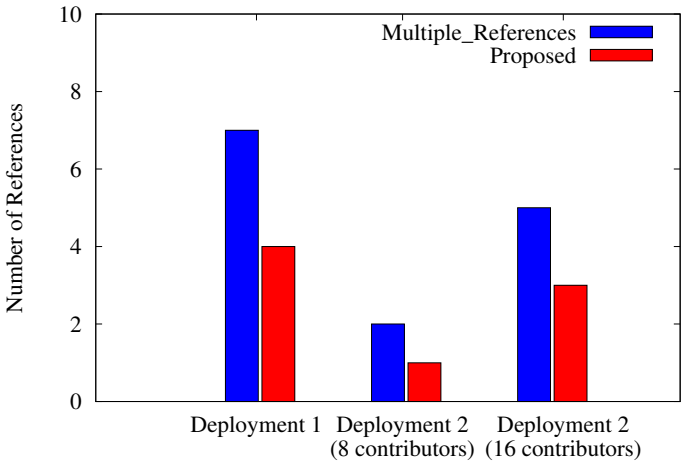


Figure 3.11: Maximum number of references used in 3 deployments.

3.4.2 Number of References

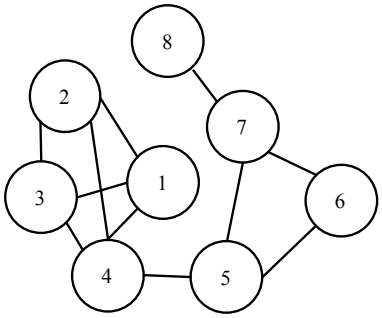
In terms of traffic reduction, the performances of the proposed and Multiple_References schemes are not significantly different in Deployments 1 and 2. Specifically, the proposed scheme outperforms Multiple_References by only 1% in terms of traffic reduction in both deployments. However, the main improvement 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. 3.11 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.

3.4.3 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. 3.12. In Deployment 3, our approach achieves an up to 11% traffic reduction as compared to Individual_Uploading, as shown in Fig. 3.13.

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 3.3 shows the attributes and corresponding traffic reduction of each deployment. 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.



	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 3.12: Graph structure of Deployment 3.

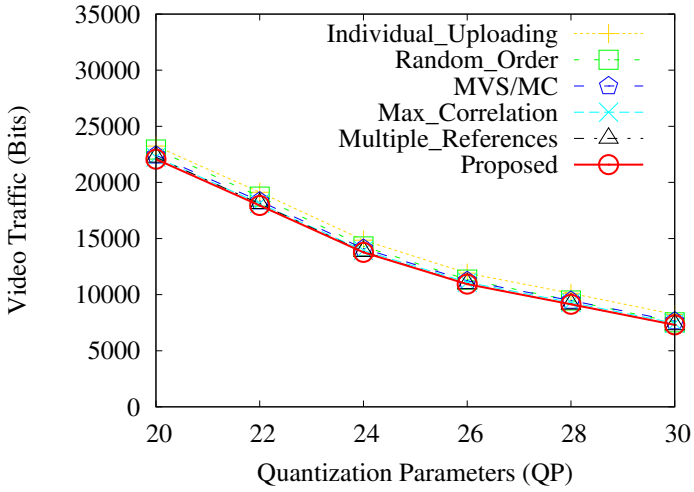


Figure 3.13: Traffic reduction in Deployment 3.

3.4.4 Overhead of Content Uploading and Transmission Order Determination

In our proposed scheme, we use a centralized approach for correlation estimation among the contributors for every GOP transmission. Specifically, the access point collects the content features from each contributor in the form of IBR. Each contributor calculates IBR of the first frame of every GOP and sends it the AP. Our reason of using the IBR for content uploading is two folds. First, the generation of IBR is simple. Each contributor takes only 10 milliseconds for IBR calculation (Measurement is made on Ubuntu 16.04 LTS 64-bit operating system with Intel Xenon (R) silver 4180 CPU and 64 GB memory). Second, each contributor needs to send a small number of bits, i.e., 64 bits to the AP to estimate the correlation degrees among the video streams. Even with the larger number of contributors, for example, 50 contributors, the number of bits is insignificant compared to the actual video traffic, i.e., approximately 3 Kbits

Table 3.3: 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%

are needed to transmit to the AP.

Regarding the overhead of transmission order determination, it is assumed that the AP has high computational capacity and thus, the time duration for transmission order determination and notification of decision can be limited to certain time interval, for example, 100 milliseconds.

3.4.5 Discussion on Required Bandwidth

The required bandwidth of the proposed scheme is conditional on and the number of contributors, the resolution of the video, and video frame rate. Suppose that there are $three$ contributors in the range of the access point, simultaneously uploading the video streams of QCIF format, i.e., 176×144 resolution with the frame rate of 25 fps. To upload the video streams from all the contributors with the GOP size of 10 frames in real time, the uploading time of one GOP must be within 400 milliseconds. From the simulation results, the proposed scheme requires each contributor to transmit an average number of approximately 100 Kbits for one GOP to achieve the video PSNR (peak signal-to-noise ratio) of 40 dB which is illustrated in Fig. 3.9 (Fig. 3.9 in Chapter 3). It implies that for three contributors, approximately 300 Kbits needs to transmit to the AP. Limiting the overheads of IBR calculation and transmission order decision as approximately 100 milliseconds, the uploading of 300 Kbits requires the minimum data rate (bandwidth) of 1 Mbps to satisfy the 1GOP uploading time of 400 milliseconds. (Here, the uploading time of 200 bits for IBR is assumed to be negligible with the considered data rate. It is also assumed the AP has the high computational capacity to conduct the scheduling of transmission of the contributors less than 100 milliseconds.)

The number of users that can access to an AP is conditional on the available bandwidth provided by the AP, the video resolution, the frame rate, and target video quality. The higher capacity wireless technologies such as 802.11 a/g/n provide the usable data rates of approximately

24 Mbps (theoretically 54 Mbps). Given the total available bandwidth of 24 Mbps and the uploading duration of 400 milliseconds for 1 GOP, an AP can serve the simultaneous video uploading from up to 72 contributors at the target PSNR of 40 dB with the QCIF resolution. (Noted that the proposed scheme requires each contributor to transmit an average number of approximately 100 Kbits for one GOP to achieve the video PSNR of 40 dB. Overhead is limited to be 100 milliseconds.)

For the video uploading scenario mentioned above, the data rate of 1 Mbps can sufficiently serve the simultaneous video uploading from three contributors to achieve the high video quality of (PSNR) of 40 dB. In a limited capacity wireless network such as 802.11b with 1 Mbps, the background traffic will incur the delay to the delivery of the videos. On the other hand, 802.11b wireless network theoretically provides the maximum data rate of 11 Mbps. In such networks, there can be a space for background traffic to coexist with the crowdsourced video traffic.

3.5 Summary

In this chapter, 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.

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

$V_{\text{G}} \leftarrow V_{\text{contributor}}$
while $|V_{\text{contributor}}| \neq 0$ **do**
  $v_i \leftarrow v \in V_{\text{contributor}}$
  for each $v_{\text{j}} \in N_{v_i}$ **do**
    $\alpha_{v_i,v_j} \leftarrow \text{cal\_Correlation}(\text{IBR}_{v_i}, \text{IBR}_{v_j})$
    $N_{v_j} \leftarrow \text{N}_{v_j} \setminus \{v_i\}$
  end for
  $V_{\text{multi\_ref}} \leftarrow \text{find\_MultiRefNodes}(\beta_{\text{multi\_ref}})$
  if $|V_{\text{multi\_ref}}| > 1$ **then**
    $E_{\text{G}} \leftarrow (v_i, v_j)\ \forall v_j \in V_{\text{multi\_ref}}$
  else if $\text{argmax}_j(\alpha_{i,j}) > \beta_{\text{single\_ref}}\ \ v_{\text{j}} \in N_{v_i}$ **then**
    $E_{\text{G}} \leftarrow (v_i, v_j)$
  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**
  $\text{Order} \leftarrow \text{Order} \cup \{v_i\}$
  for each $v_j \in V_{v_i}^{adj}$ **do**
    if $v_{\text{j}} \notin P_{v_i}$ **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

Chapter 4

Soft Video Uploading for Low-Power Crowdsourced Multi-view Video Streaming

4.1 Introduction

Video streaming from a crowded event by its attendees, i.e., crowdsourced video contributors, with the powerful smartphones' cameras has been a popular trend nowadays. The most famous Internet platforms, which provide crowdsourced video streaming services, are Facebook Live, Youtube, Ustream, and YouNow [5, 6, 7, 8, 9]. In such video streaming, the contributors provide videos with multiple viewpoints of the event captured at different perspectives. The integration of those videos into a multi-view video allows the remote viewers to enjoy the immersive viewing of the event as if they were at the event. Such kind of video streaming is known as crowdsourced multi-view video streaming [16] and useful for the wide areas of applications such as entertainment, surveillance, social sharing, and education.

Conventionally, a digital video codec such as H.264/AVC consists of complex operations such as the motion estimation (ME), mode decision for intra- and inter-frame encoding, quantization, entropy encoding, and channel coding. Besides, in crowdsourced multi-view video streaming, the videos captured at the same event are highly correlated, resulting in the redundant transmissions. To avoid this issue, the correlations among the video streams should be exploited to reduce the amount of traffic by using the disparity estimation in differential encoding [72, 17]. This can be achieved by using Multi-view Video Coding (MVC) extension of H.264/AVC. Despite the reduced traffic, MVC would consume more power due to the increase in the complexity

for video encoding. It means that conventional video codec may not be efficient for video uploading from the battery-constrained mobile devices. Although power-efficient video encoding schemes have been proposed [73] to reduce the complexity and power consumption in video encoding, the improvement comes with video quality degradation.

Besides the video encoding, video transmission is another major cause of battery usage of the mobile devices. Generally, power consumption for the video delivery is commensurate with the amount of data (source data rate) to be transmitted [74]. In addition, wireless channel conditions also affect the energy usage when the wireless channel is noisy and unstable in quality. A video encoder needs to adjust its source coding rate with the wireless channel conditions, and sometimes retransmit noise-distorted packets. These operations lead to additional energy drainage from the devices. At a low channel signal-to-noise ratio (SNR), digital video transmission may require low order modulation scheme such as BPSK to reduce the symbol error rate, resulting in a larger number of transmission symbols. According to the power consumption model of video transmission [74], the transmission power consumption is linear to the number of transmission symbols. Thus, the digital transmission requires more transmission power in lower channel SNRs.

In addition, the digitally encoded bitstream is highly vulnerable to bit errors, the received video quality drops significantly when the channel SNR falls under a certain threshold. This phenomenon is referred to as cliff effect. Moreover, even when the channel quality improves, the video quality remains constant due to the quantization error, known as constant quality effect.

The goal of our study is to realize high-quality and low-power crowdsourced multi-view video uploading system. To achieve this goal, we were inspired by the recent studies on soft video delivery [34,35] such as SoftCast. In soft video delivery, a video source does not care about the wireless channel quality at the receivers. Specifically, the sender transforms pixel values of video frames into frequency components using three dimensional discrete consine transform (3D-DCT), and then directly maps the frequency components to transmission symbols, known as near-analog modulation. By skipping digital video encoding and transmission, soft video delivery can significantly decrease power consumption in video encoding as well as video transmission. Moreover, since the received video quality is proportional to channel quality of

the receiver, it achieves graceful quality improvement avoiding the cliff and constant quality effects.

However, to apply conventional soft video delivery to crowdsourced multi-view video streaming system, we need to solve the following issues. First, conventional schemes allow contributors to directly upload the captured videos to the AP and thus it suffers from low video quality when channel quality of a given contributor to the AP is unstable. Although one of the typical solutions is to exploit stable redirect path for video uploading, the conventional schemes on soft video delivery has not been designed for redirect transmission. Consequently, conventional schemes may suffer video quality degradation when the channel quality difference between the direct and redirect paths increases. Second, since the videos captured at the same event may have high correlation in multi-view video contents, resulting in the redundant transmissions across the contributors. Since 3D-DCT operation in the conventional soft video delivery does not exploit such inter-camera correlations, the conventional scheme may suffer low video quality due to the redundant transmissions.

In this study, we propose a high-quality and low-power uploading scheme for crowdsourced multi-view videos. Considering the above mentioned issues of the conventional digital-based and soft video delivery schemes, our approach considers cluster-based redirect video uploading system which combines the four approaches of network clustering, delegate selection, soft video delivery, and four-dimensional DCT (4D-DCT). Specifically, contributors are clustered based on their locations and contributor with the lowest power consumption rate and closest to the AP is selected as the delegate to make redirect path between each contributor to the AP. The members of the cluster deliver their videos to the delegate using 3D-DCT and near-analog modulation. And the delegate performs 4D-DCT operation to exploit the correlation across the received video streams and its own and delivers the video streams to the AP. By exploiting channel quality difference between the direct and redirect paths and content correlations between the contributors, the proposed scheme can realize the high-quality and lower-power crowdsourced multi-view video uploading system.

Evaluation results show that the proposed scheme outperforms the conventional SoftCast scheme when the channel quality difference between the direct and redirect paths increases. In

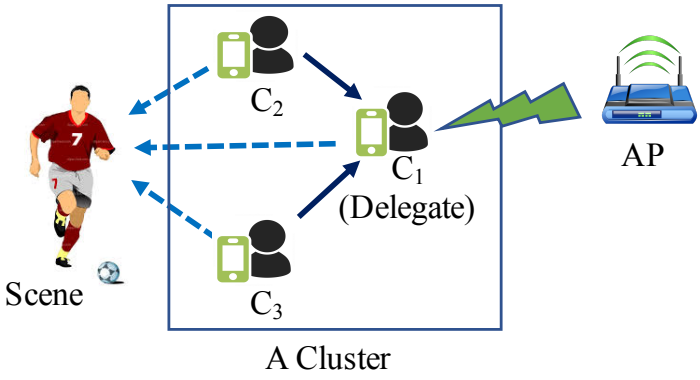


Figure 4.1: System model. (C_n: contributors)

addition, the proposed scheme achieves better video quality at the different wireless channel qualities and reduced transmission rates compared to the digital-based video uploading schemes. Moreover, the proposed scheme realizes low power encoding and transmission across the contributors compared with the digital-based schemes because the total uploading time spent by the proposed scheme is lower than that of the digital-based video uploading schemes.

4.2 Proposed Scheme

4.2.1 Overview

Fig. 4.1 depicts our considered system model for crowdsourced multi-view video uploading in which multiple contributors capture the videos of a crowded event. We assume that each contributor uses the same camera and the cameras are calibrated and synchronized before uploading. We left what adaptation techniques can be used across mobile contributors as a future work. As an example, the existing study on camera integration [75] discussed calibration and synchronization across mobile devices. The captured streams are uploaded to a cloud server via the wireless AP which is located at the event location. All the contributors are resided in the transmission range of the AP. In this study, we show the effectiveness of soft video delivery in crowdsourced multi-view video uploading by exploiting the channel quality differences and content correlations between the contributors.

The video transmissions from the contributors are conducted in the basis of group of picture (GOP). Fig. 4.2 shows a timing diagram of our proposed video uploading scheme. As shown in the figure, each transmission of 1 GOP consists of three phases: initialization, network clustering

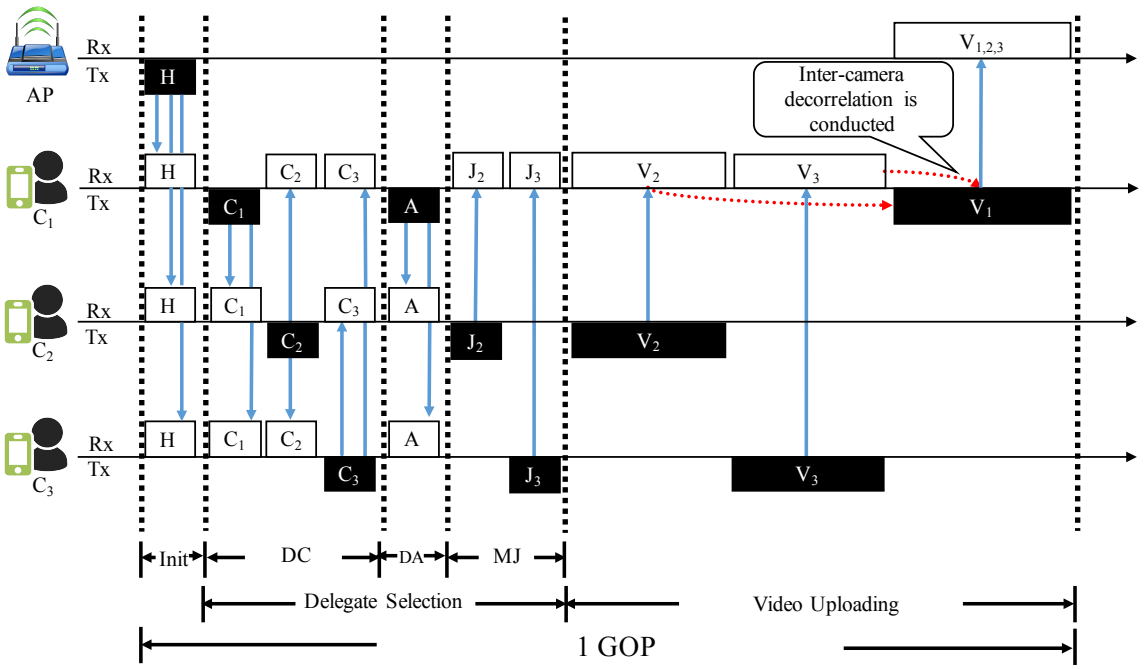


Figure 4.2: Timing diagram of proposed video uploading system

and video uploading.

1) Initialization (Init): In this phase, the AP broadcasts a hello message, H, to the contributors located in its communication range. Upon receiving the message, each contributor computes its approximate distance to the AP based on the received signal strength. This distance value will be utilized as one of the parameters in network clustering process.

2) Network Clustering: The clustering of the contributors can be accomplished by either centralized or distributed manner as in traditional wireless sensor network [76]. In centralized approach, the AP is responsible for the network clustering by collecting the necessary information from the contributors. The distributed approach can be implemented by message exchange between the contributors. In this study, we consider distributed clustering approach as in [77]. The detailed descriptions of network clustering of the contributors are provided in subsection 4.2.2.

3) Video Uploading: In this phase, each member of a cluster encodes its captured video stream and sends it to the delegate. After collecting the video streams from the members, the delegate encodes all the video streams including its own stream by using 4D-DCT to exploit the inter-camera correlations among the streams. Finally, the delegate modulates the resulting DCT

coefficients by using near-analog modulation, and uploads the video streams to the AP.

4.2.2 Network Clustering

The clustering of the contributors includes two main operations: the delegate selection and the cluster formation.

Delegate Selection

After the initialization phase, the delegate selection process is initiated.

Delegate Claim (DC): At the start of the delegate selection process, contributor i broadcasts a claim message, C (id, ω_i), sets its timer τ_i and listens to the claim messages from its neighbors, where ω_i indicates the weight of the contributor i, which is obtained by the Eq. (4.1):

$$\omega_i = \frac{E_{\text{residual}}(i)}{E_{\text{initial}}(i)} * \frac{1}{\text{d}_{\text{AP}}(i)} \tag{4.1}$$

where E_{residual} and E_{initial} are current and initial energy of contributor i, and d_{AP}(i) is the distance between contributor i and the AP. Considering the energy-efficiency, the Eq. (4.1) includes two terms; the first term is the energy consumption rate of the contributor and the second term is its distance from the AP.

Delegate Announcement (DA): When the timer expires, each contributor checks the weight values of its neighbors and the contributor with the lowest ω_i, i.e., $\text{argmin}_i(\omega_i)$, becomes a delegate. After winning the selection process, the delegate broadcasts a message, A (id), to announce its status.

Cluster formation

In this process, other contributors join the delegate to form a cluster.

Member Join (MJ): On receiving the announcement from the delegate, the contributor j sends a member-join message, J (id), to the delegate located in its nearest vicinity, completing the cluster formation process.

The network clustering is performed for every GOP considering the energy balancing and mobility of the contributors. Both centralized and distributed clustering come with their respective communication and computation costs [76]. Since this study mainly focuses on the

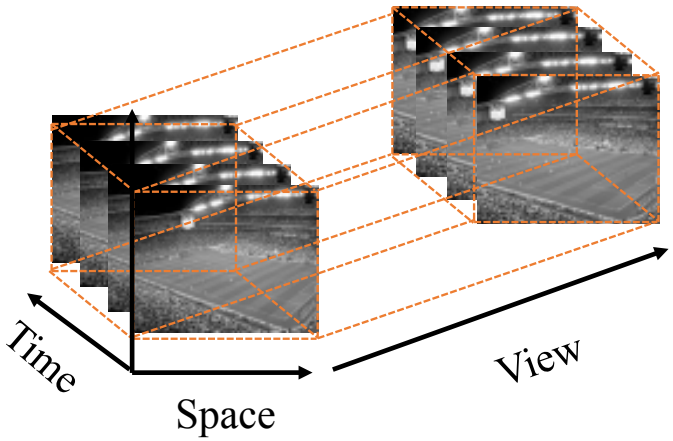


Figure 4.3: Four-dimensional decorrelation.

effectiveness of the redirect video uploading with soft video delivery in a clustered network of contributors, impacts of the clustering cost and the more sophisticated clustering algorithm for crowdsourced multi-view video uploading will be left as a future work.

4.2.3 Video Uploading

Encoding

Each member of the cluster performs 3D-DCT operation on its captured video to remove intra-frame and inter-frame correlations before sending to the delegate. On the other hand, the delegate uses 4D-DCT operation, which decorrelates pixel values across the video frames as well as across the video streams, to exploit the correlations in inter-camera domain for more energy compaction as illustrated in Fig. 4.3.

Transmission

After multi-dimensional DCT, i.e., 3D-DCT/4D-DCT, the resulting coefficients in frequency domain are power allocated for transmission. Let $\check{s}_i \in \mathbb{R}$ be a power allocated symbol of i-th analog-modulated symbol. Each DCT coefficient is scaled by a power scaling factor g_i for noise reduction:

$$\check{s}_i = g_i \cdot c_i, \tag{4.2}$$

where $c_i \in \mathbb{R}$ is the i-th DCT coefficient. The near-optimal value g_i to minimize the mean square reconstruction error is obtained as follows [34]:

$$g_i = \lambda_i^{-1/4} \sqrt{\frac{P}{\sum_k^{N_{GOP}} \lambda_k}}, \tag{4.3}$$

where P denotes a total transmission power budget, λ_i is the power of i-th coefficient, and N_{GOP} is the number of coefficients in one GOP. Finally, a transmission symbol $x_i \in \mathbb{C}$ is created by superposing the two power allocated symbols $\check{s}_i$ and $\check{s}_j$ as follows:

$$x_i = \check{s}_i + \jmath \check{s}_j, \tag{4.4}$$

where $\jmath = \sqrt{-1}$ denotes the imaginary unit. After power allocation, each transmission symbol in GOP is mapped on to I (in-phase) and Q (quadrature-phase) of the transmitted signal, and delivered to the AP over a wireless link.

Decoding

At the AP, the decoder receives the symbol which can be modeled as follows:

$$y_i = x_i + n_i, \tag{4.5}$$

where $y_i \in \mathbb{C}$ is the i-th received symbol and $n_i \in \mathbb{C}$ is an effective noise, commonly assumed as additive white Gaussian noise (AWGN) with a variance of σ^2 (which is already normalized by wireless channel strength in the present of fading attenuation). The decoder uses minimum mean-square error (MMSE) filter [34] to extract the DCT coefficients from the I and Q components of the received symbols:

$$\hat{x}_i = \frac{g_i \lambda_i}{g_i^2 \lambda_i^2 + \sigma^2} \cdot \Re(y_i), \hat{x}_j = \frac{g_j \lambda_j}{g_j^2 \lambda_j^2 + \sigma^2} \cdot \Im(y_i). \tag{4.6}$$

Finally, the decoder takes the inverse multi-dimensional DCT to obtain the original pixels in each video frame.

Analog Compression for Limited Resource

In the above mentioned designs, it is assumed that the contributor has enough resources to transmit all the coefficients in the spectral domain over the wireless medium. However, if the available wireless and/or device's resources are restricted for video uploading, the contributor needs to selectively transmit the coefficients to fit the amount of resources. In such cases, our scheme conducts a sort of data compression even for soft video delivery. Specifically, the contributor sorts the coefficients in the descending order of their energy and picks the higher-energy coefficients to fill the resource. The decoder regards the discarded lower-energy coefficients as zero. Even though some coefficients are discarded to reduce the amount of data, the receiver can still achieve a graceful video quality until reaching the distortion limit due to the compression.

4.3 Performance Evaluation

We conduct the performance comparison of our proposed video uploading scheme, conventional SoftCast, and digital-based uploading schemes through simulations implemented in MATLAB.

4.3.1 Effect of Cluster-based Redirect Uploading

Firstly, we evaluate the performance of the proposed and conventional SoftCast schemes to discuss an effect of the cluster-based redirect uploading on crowdsourced multi-view videos. Since conventional SoftCast is not designed to exploit correlation between captured videos, we investigate the performance of our proposed scheme when the captured videos have correlation/no correlation for fair comparison. For this purpose, we set up the evaluation settings as follows:

Reference Schemes:

1. *SoftCast*: In this scheme, each contributor in the cluster individually uploads the captured videos to the AP using 3D-DCT operation.

2. *Proposed*: This is the proposed analog-based scheme as explained in Section 5.2.

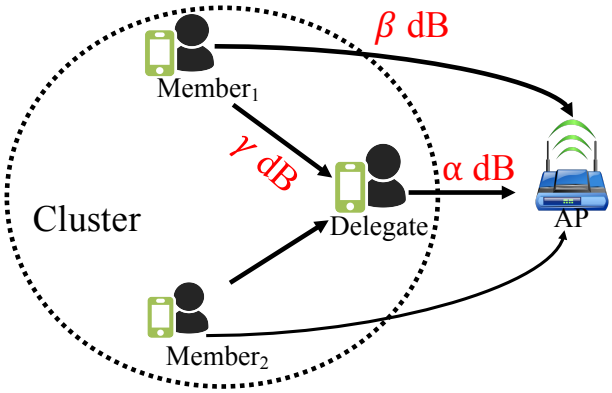


Figure 4.4: Deployment of cameras and channels between the contributors and the AP.

Metric: We measure video quality in terms of average peak signal-to-noise ratio (PSNR) which is defined as follows:

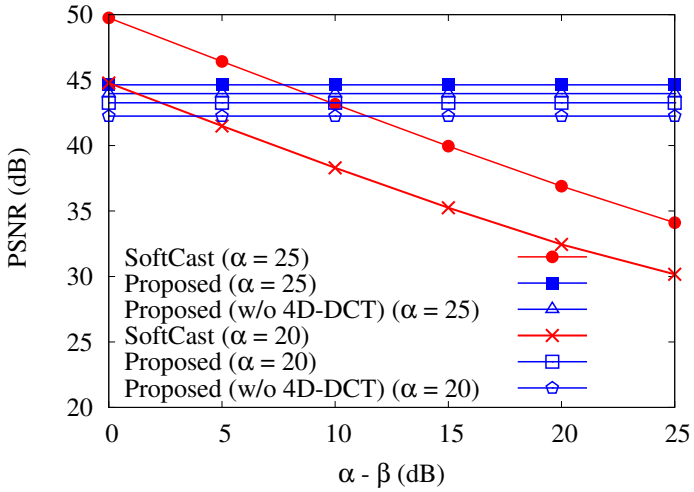
$$\text{PSNR} = 10 \log_{10} \frac{(2^L - 1)^2}{\varepsilon_{\text{MSE}}}, \tag{4.7}$$

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

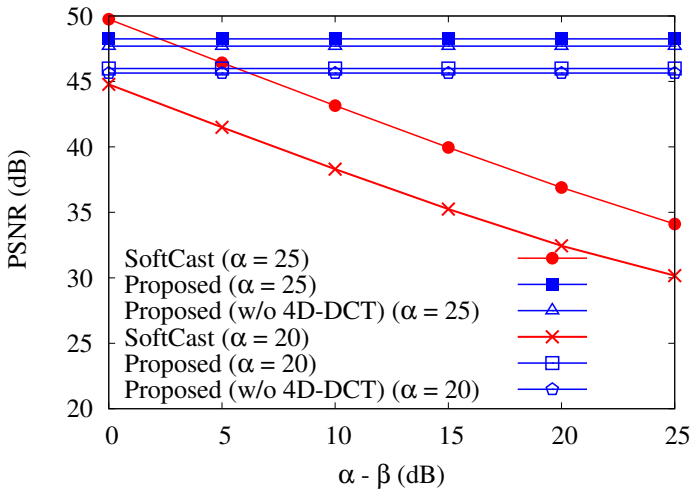
Video Sequence: We use video frames from the standard multi-view video sequences known as *ballroom*, *ballroons*, and *exit* provided by MERL [71]. Each video sequence consists of videos from 8 cameras. Videos are in quarter common intermediate format (QCIF) with 176×144 resolution. The frame rate is 30 frames per second. GOP size of 10 frames is set for all transmission schemes.

Network Deployments: We consider a simple network of three contributors to access the baseline performance of cluster-based redirect video uploading scheme. We regard three successively aligned cameras of 1, 2, and 3 from *ballroom* sequence as correlated contributors' videos. On the other hand, we use camera 1 from *ballroom*, *ballroons* and *exit*, respectively, to regard the contributors without correlation. An arrangement of the cameras and communication channels between contributors and the AP are illustrated in Fig. 4.4. In the figure, we regard channel SNR between the delegate and the AP as α, between the member and the AP as β, and between the member and delegate as γ, respectively.

Fig. 4.5 and Fig. 4.6 plots the received video quality of the reference schemes with correlated and non-correlated contributors as a function of channel quality difference between delegate to



(a) γ = 20 dB



(b) γ = 25 dB

Figure 4.5: Video quality of the proposed and conventional SoftCast schemes with correlated contributors as a function of channel quality difference between direct and redirect paths.

AP and contributor to AP channels, respectively. Moveover, we show the effect of exploiting the correlation at the delegate in Fig. 4.5 by skipping the 4D-DCT operation. For the sake of clarity, we assume that selected contributor in SoftCast experiences the same channel quality between the AP. We investigate two SNR values of α, i.e., 20 and 25 dB under the intra-cluster channel quality of 20 dB and 25 dB, respectively, for the proposed scheme.

As shown in Figs 4.5(a) and 4.5(b), the proposed scheme improves the video quality up to 1 dB by exploiting the correlation between the correlated contributors. However, SoftCast scheme suffers quality degradation when the channel quality difference between delegate to AP

and contributor to AP channels increases. Since SoftCast scheme does not consider delegate selection and network clustering, the channel quality between the selected contributor and the AP may significantly change compared with that of delegate to AP channel. For example, SoftCast scheme may select three random contributors with unstable channel quality to upload the captured videos to the AP through a certain contention-based medium access protocol.

On the other hand, by selecting the contributor with stable channel quality as a delegate and using the higher quality member to delegate path, which is leveraged by network clustering and delegate selection, our proposed scheme can take advantage of such channel quality difference between the direct and redirect paths to improve the video quality. From Figs. 4.6(a) and 4.6(b), we can see that the proposed scheme with non-correlated contributors also achieves the similar performance with that with correlated contributors, showing the effectiveness of cluster-based redirect uploading scheme compared to the conventional SoftCast.

4.3.2 Video Quality in Different Channel Quality

In this section, we evaluate the video quality of our proposed scheme and conventional digital-based schemes in unstable wireless environment under the following evaluation settings.

Reference Schemes:

1. *Digital-based*: In this scheme, each contributor in the cluster individually uploads the captured videos to the AP using conventional digital video encoding, i.e., H.264/AVC.

2. *Proposed*: This is our proposed scheme as explained in section 5.2.

Network Deployments: We consider two types of network deployments.

- As deployment 1, we consider the same deployment as described in subsection 4.3.1.

- As deployment 2, we arrange a network with eight contributors to access the performance of the reference schemes in a larger network. For this purpose, we use all the cameras in *ballroom* sequence.

Wireless Configurations: The transmitted symbols of the reference schemes are impaired by an additive white Gaussian noise (AWGN) channel. For the channel quality between the delegate

and the AP, we use channel SNRs from 0 dB to 25 dB. For intra-cluster communication, we consider three channel SNRs of 15, 20 and 25 dB. For the error protection in digital-based uploading, we use rate 1/2 and 1/4 convolutional codes with a constraint length of 8. For the modulation, we use binary phase-shift keying (BPSK), quadrature PSK (QPSK), 16-ary quadrature-amplitude modulation (16-QAM) formats. We set the channel symbol rate for the proposed scheme to the half of the number of DCT coefficients to be transmitted in one second. For example, in deployment 1, the delegate needs to transmit approximately 11.4 (= $176 \times 144 \times 30 \times 3 \times \frac{1}{2}$) Msymbols/s for three contributors. To adjust the channel symbol rate with digital-based schemes, we control the quantization parameters in digital encoder.

Fig. 4.7 plots the PSNR performance of the reference schemes as a function of wireless channel SNRs between the delegate and the AP in two deployments. In deployment 1, as shown in Fig. 4.7(a), all the digital-based schemes suffer from a cliff effect, i.e., the video quality drops drastically at a certain channel SNR. For example, the cliffs appear 2 dB in BPSK 1/4, 4 dB in BPSK 1/2, 7 dB in QPSK 1/2, 13 dB in 16QAM 1/2 schemes, respectively. In addition, the digital-based schemes receive the constant video quality in higher channel SNRs. This is because the quantization error introduced in the quantization process cannot be recovered at the receiver side. On the other hand, the proposed scheme prevents the cliff effect and the constant video quality by skipping the quantization and entropy coding, and achieves a graceful quality improvement with the improvement of wireless channel quality between the delegate and the AP regardless of the intra-cluster channel quality. However, at the intra-cluster channel quality of 15, the proposed scheme suffers video quality degradation of 1.1 dB compared to the 16QAM 1/2 when the channel quality between the delegate and the AP is between 13 dB and 15 dB. It can be explained that at a higher channel SNR, the digital-based scheme with higher-order modulation can achieve the better video quality than the proposed scheme with lower intra-cluster channel. Even this, the proposed scheme with the intra-cluster channel quality of 15 still outperforms the 16QAM 1/2 scheme when the channel quality between the delegate and the AP is above 15 dB and below 13 dB by overcoming the cliff effect and the constant quality effect because of the graceful quality improvement of analog video transmission. It shows the robustness of the proposed scheme in unstable intra-cluster and cluster-AP wireless channel environments.

Deployment 2 shows the same tendency as deployment 1. Specifically, as shown in Fig. 4.7(b), the proposed scheme with higher intra-cluster channel quality still outperforms the digital-based schemes, showing the effectiveness of our proposed scheme in a large scale network, i.e., its scalability.

4.3.3 Transmission Power Consumption

Here, we compare the number of required transmission symbols per GOP as a function of received PSNRs considering intra-cluster communication.

Fig. 4.8 plots the number of transmitted symbols at different received PSNRs of the reference schemes in two deployments. In this figure, we set the wireless channel quality between the delegate and the AP to 18 dB. In deployment 1, the proposed scheme needs to transmit the lowest number of symbols compared to all digital-based schemes regardless of the intra-cluster channel quality. For example, at the received PSNR of 38.5 dB, the proposed scheme at intra-cluster channel quality of 25 dB needs to transmit approximately 0.13 Msymbols/GOP, saving 2.7 Msymbols per GOP compared to BPSK 1/4 scheme as shown in Fig. 4.8(a). More specifically, the digital-based schemes of 16QAM 1/2, QPSK 1/2, BPSK 1/2, and BPSK 1/4 require approximately 0.3, 0.6, 1.2, and 2.5 Msymbols/GOP, respectively.

From Fig. 4.8(b), we can see the same tendency in deployment 2. Specifically, at the received PSNR of 38.5 dB, the proposed scheme at intra-cluster channel quality of 25 dB transmits approximately 0.4 Msymbols/GOP while the digital-based schemes of 16QAM 1/2, QPSK 1/2, BPSK 1/2, and BPSK 1/4 transmit approximately 0.9, 1.8, 3.6, and 7.2 Msymbols/GOP, respectively. Since the digital-based schemes require more transmission symbols to achieve a certain video quality, it means that digital-based schemes will consume more transmission power compared to the proposed scheme irrespective of the intra-cluster channel. We can see that the number of required transmission symbols in higher-order modulation scheme such as 16QAM 1/2 approaches to the proposed scheme for all considered intra-cluster channel quality. Although the digital-based schemes can use the denser modulation schemes such as 64QAM to reduce the number of transmission symbols, the transmitted symbols will become much sensitive to the channel noise, and thus it will suffer from cliff effect in lower SNR regimes as shown in

Fig. 4.7.

4.3.4 Video Quality in Reduced Transmission Rate

Due to the limited power availability, a crowdsourced contributor may intend to reduce its transmission rate for power saving. On the other hand, reducing the transmission rate may cause the received video quality degradation at the receiver due to discarding the source data. In this section, we evaluate the video quality of reference schemes at the reduced transmission rate. For comparison, we define *cutoff rate* as a metric. At the cutoff rate of 0.1, 10% of DCT coefficients will be discarded in the proposed system. For digital-based schemes, we adjust the quantization parameters to make the number of transmission symbols equal with the proposed scheme as much as possible for fair comparison.

Fig. 4.9 plots the video quality of the reference schemes as a function of cutoff rate in deployment 1 at the channel SNRs between the delegate and the AP of 4 dB and 15 dB, respectively. It is noted that we only show the video PSNRs greater than the 20 dB in these figures. From Figs. 4.9(a) and 4.9(b), we can see that the proposed scheme performs well with the acceptable video quality at both channel SNR values regardless of the intra-cluster channel quality. For example, the proposed scheme with respective intra-cluster channel quality achieves 28.8 dB at the channel SNR between the delegate and the AP of 4 dB and 31 dB at 15 dB, at the cutoff rate of 0.9, i.e., 90% of DCT coefficients are discarded in which all of the digital-based schemes suffer from low video quality.

In addition, at the wireless channel quality of 4 dB, all the digital-based schemes suffer the severe quality degradation regardless of the cutoff rate. To be specific, only BPSK 1/4 scheme achieves the PSNR of 22 dB at the cutoff rate of 0.1, i.e., when 90% of coefficients are transmitted. On the other hand, the proposed scheme keeps almost the same video PSNR from the cutoff rate of 0 to 0.5, i.e., 50% compression of DCT coefficients irrespective of the intra-cluster channel quality. The reason for this result is that maintaining higher-energy DCT coefficients can compensate the video quality degradation due to compression.

In Fig. 4.9(b), the digital-based schemes improve the video quality with lower cutoff rates

whereas the proposed scheme still outperforms the digital-based schemes except at the intra-cluster channel quality of 15 dB. At the cutoff rate of 0, 16QAM1/2 scheme slightly outperforms the proposed scheme with intra-cluster channel quality of 15 dB by 0.4 dB. However, the proposed scheme still achieves the similar video quality with the video quality degradation of 0.04 dB at the cutoff rate of 0.1 regardless of the intra-cluster channel quality whereas all the digital-based schemes suffer video quality degradation. From these results, we can conclude that the proposed scheme can safely reduce the transmission symbols for power saving purpose up to 90%, especially in higher channel quality. We noted that in the deployment 2, the proposed scheme achieves the similar performance even in a large network size.

4.3.5 Discussion on Power Saving

In this section, we discuss the power saving performance of the proposed scheme in comparison with the digital-based scheme. We assume that power consumption of the video uploading process is proportional to the total time spent of the uploading process. Specifically we define a mixed model of encoding time and transmission time in the digital-based scheme and in the proposed scheme as follows:

$$T_{\text{digital}} = N\frac{D}{R} + NT_{\text{dig\_encode}}, \tag{4.8}$$

$$T_{\text{proposed}} = (N-1)\frac{D}{R} + (N-1)T_{\text{ana\_encode}} + \frac{D}{R} + \theta \cdot T_{\text{ana\_encode}} + T_{\text{overhead}}, \tag{4.9}$$

where T_{digital} and T_{proposed} represent the total encoding and transmission time of the conventional digital-based uploading scheme and proposed scheme, respectively. In addition, N is the number of contributors, D is the number of transmission symbols, R is an available channel symbol rate, and θ represents increase rate of encoding time from 3D-DCT to 4D-DCT, respectively. Moreover, T_{overhead} is the additional time required in the proposed scheme for network clustering, synchronization across contributors, delegate selection and so on. We consider the number of transmission symbols in the conventional digital-based scheme and the proposed scheme is the

Table 4.1: Encoding time in each GOP

Encoding Scheme	**Encoding time (seconds)**
3D-DCT	0.1
H.264/AVC	32.4
4D-DCT	0.1

same and the available channel symbol rate for each contributor is the same. We note that the proposed scheme yields better video quality when the number of transmission symbols in the digital and the proposed scheme is the same as shown in Fig. 4.8.

According to the equations, we first evaluate the encoding time of H.264/AVC used in digital-based schemes and 3D-DCT and 4D-DCT performed in the proposed scheme. For the digital encoding, we use open-source H.264-based MATLAB video encoder [78]. For the analog encoding, we use multi-dimensional DCT (MDCT) provided by [79]. The specifications of the operating environment are Ubuntu 16.04 LTS 64-bit operating system with Intel Xeon (R) silver 4108 CPU and 64 GB memory.

From the viewpoint of software video codec, the encoding time varies with the compression rate, i.e., encoding parameters such as quantization parameters (QP), number of used reference frames, and ME search range. In particular, QP has a significant impact on the video quality. Due to this, we measure the encoding times at different QP values from 1 to 51, i.e., highest to lowest video qualities, and take an average of encoding times across all the quantization parameters for the comparison.

Table 4.1 shows that the encoding times of the proposed and digital-based schemes. From the table, $T_{\text{dig\_encode}}$ and $T_{\text{ana\_encode}}$ are 32.4 and 0.1, respectively. In addition, the value of θ is the 1 since the encoding time of 3D-DCT and 4D-DCT is almost the same. We consider D of 1000 KB, R of 100 KBps, and N of 3.

Fig. 4.10 shows the total uploading time of the proposed scheme and the digital-based schemes as a function of T_{overhead}. From the figure, we can see that the proposed scheme spends less time for the uploading process than the digital-based scheme when T_{overhead} is less than 90 seconds. It means that the proposed scheme can save more power consumption compared to the digital-based video uploading scheme below the overhead of 90 seconds. We note that T_{overhead}

is conditional on the number of contributors in the cluster, the frequency of additional operations during the uploading process and so on. Here, with the smaller number of contributors within the cluster, for example, three contributors, T_{overhead} is expected to be significantly lower than 90 seconds. In addition, when the contributors are in low mobility for a certain GOP, T_{overhead} may tend to zero because synchronization and network clustering are not neccessarily conducted.

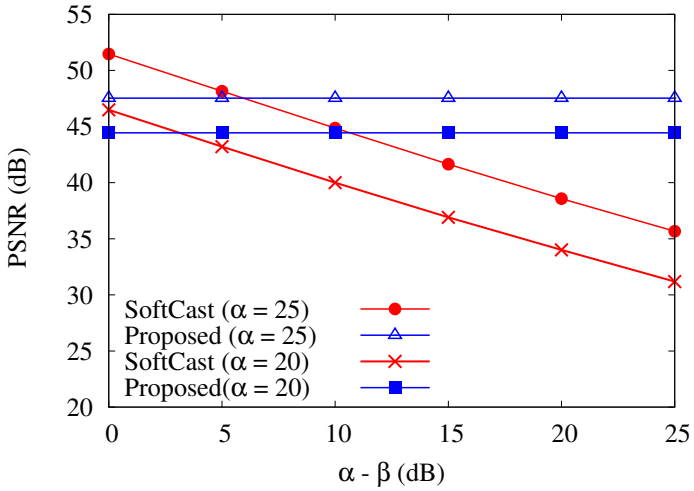
4.4 Discussion on Non-correlated Contributors

Even when the proposed scheme realizes careful adaptation across the mobile contributors, each contributor may see different angles and they may have no correlations across the contributors. In this section, we investigate the performance of our proposed scheme in such situation. For this purpose, we use three cameras taken each from three different video sequences, namely, *ballroom*, *balloons* and *exit* for evaluations.

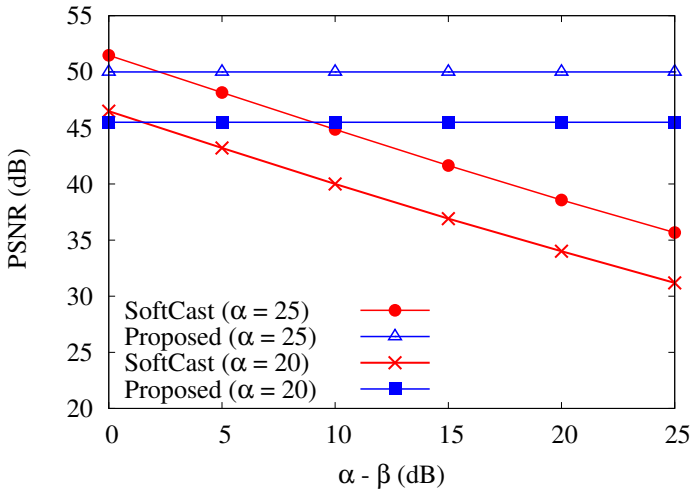
Fig. 4.11 shows the PSNR performance of our proposed scheme and the conventional digital-based uploading schemes. The proposed scheme still performs well compared to the digital-based video uploading schemes even when there is no correlation between the contributors. We note that we have discussed the performance of the proposed scheme and conventional SoftCast scheme in Fig 4.5. We can see that the proposed scheme with non-correlated contributors yields better video quality compared with conventional SoftCast as the channel quality difference between delegate to AP and contributors to AP channels increases.

4.5 Summary

In this chapter, we propose a high-quality and low-power video uploading scheme for crowdsourced multi-view video streaming from synchronized crowdsourced contributors. The proposed scheme adopts cluster-based redirect video uploading with soft video delivery to exploit channel quality differences and content correlations between the contributors, and to reduce power consumption in video encoding and video transmission. Evaluation results show that the proposed scheme outperforms the conventional SoftCast and digital-based video uploading schemes in terms of video quality and power consumption.

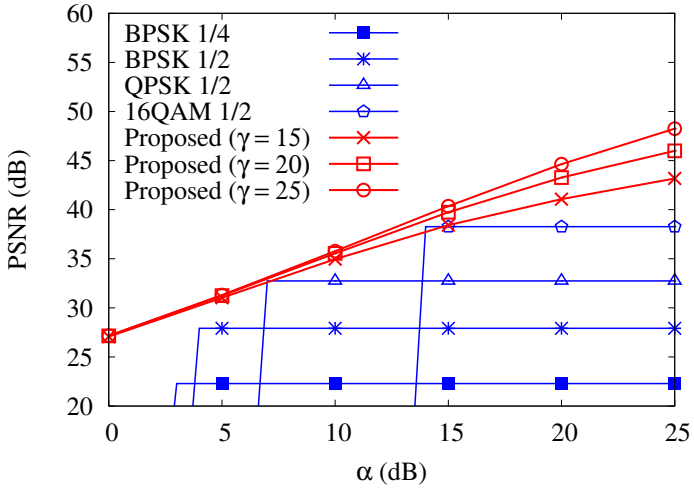


(a) γ = 20 dB

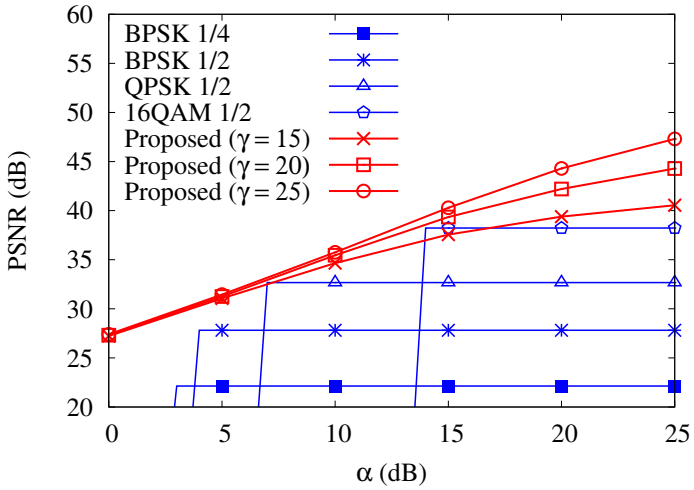


(b) γ = 25 dB

Figure 4.6: Video quality of the proposed and conventional SoftCast schemes with non-correlated contributors as a function of channel quality difference between direct and redirect paths.

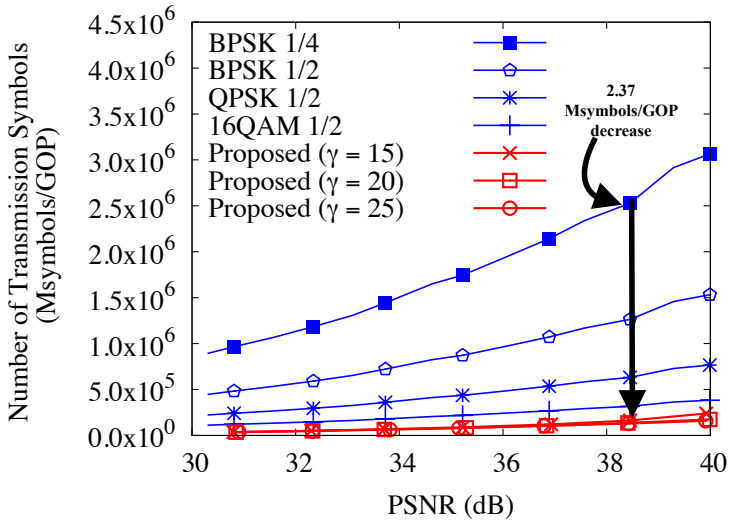


(a) Deployment 1

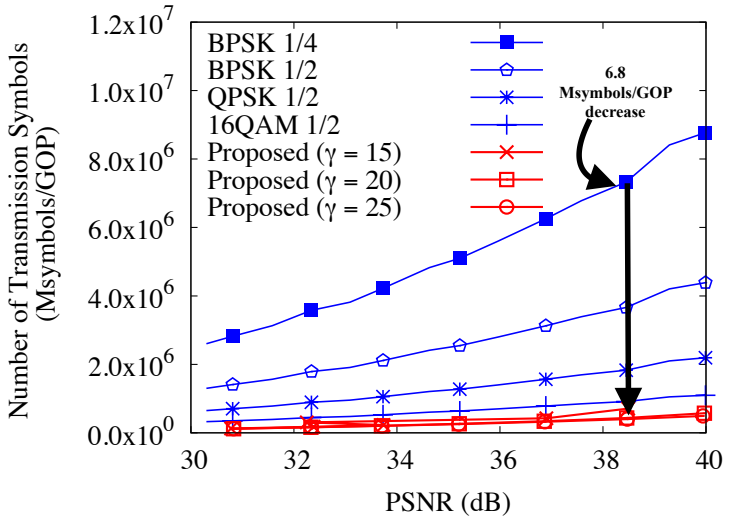


(b) Deployment 2

Figure 4.7: Video quality of the proposed and conventional digital-based schemes as a function of wireless channel SNRs between the delegate and the AP.

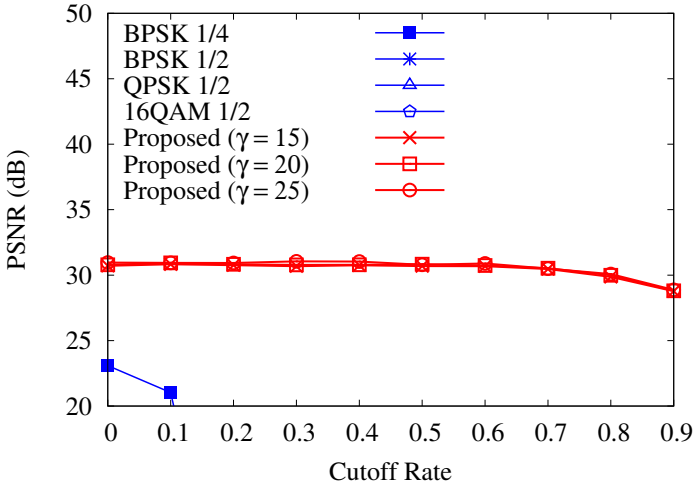


(a) Deployment 1

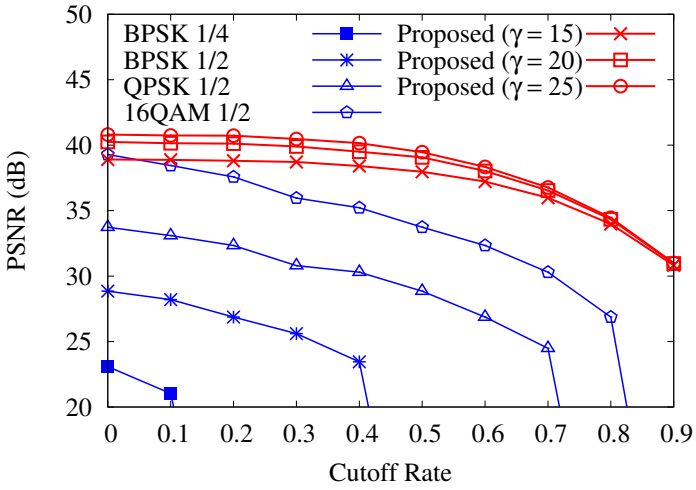


(b) Deployment 2

Figure 4.8: Number of transmission symbols required by the proposed and conventional digital-based schemes as a function of received PSNRs.



(a) α = 4 dB.



(b) α = 15 dB.

Figure 4.9: Video quality of the proposed and conventional digital-based schemes as a function of cutoff rates in deployment 1.

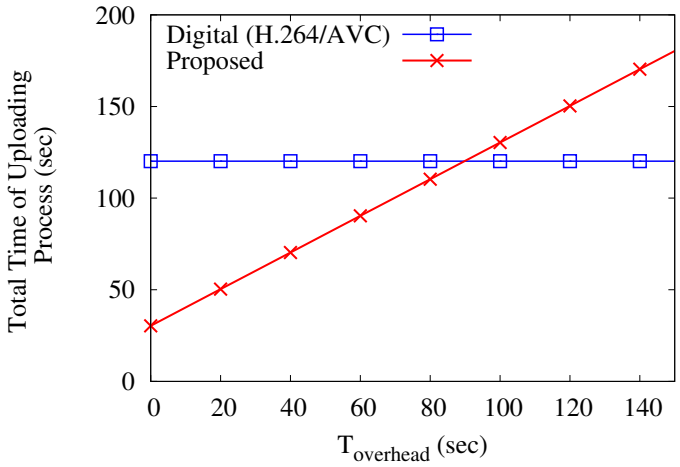


Figure 4.10: Total uploading time spent by the proposed and conventional digital-based schemes as a function of overhead times in the proposed scheme.

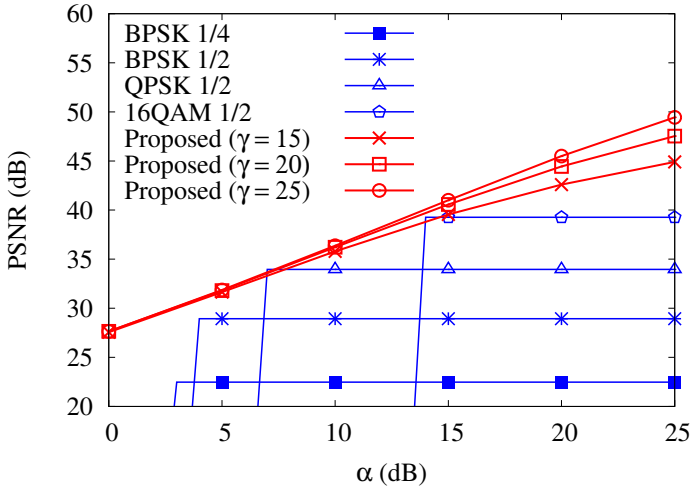


Figure 4.11: Video quality of the proposed and conventional digital-based schemes with non-correlated contributors as a function of wireless channel SNRs between the delegate and the AP.

Chapter 5

ROI-based Crowdsourced Multi-view Video Uploading

5.1 Introduction

Thanks to the advancement of wireless technology and the powerful smartphones with high resolution cameras, video streaming of an event captured in different angles by its attendees, i.e., crowdsourced video contributors, is very popular nowadays [5,6]. Such video streams uploaded by the crowdsourced contributors are integrated into multi-view videos at a server and streamed to the remote viewers as the crowdsourced multi-view video streaming services [16]. Such kinds of services allow the viewers to experience the immersive viewing of the event as if they were in the event. On the other hand, simultaneous uploading of video streams from the contributors is restricted by the inherent limitations of the wireless network such as available bandwidth. Therefore, how to efficiently upload a large amount of video traffic within the limited network resources is one of the major issues in crowdsourced multi-view video streaming.

To conduct the efficient uploading of crowdsourced multi-view videos, traffic reduction is paramount important. To this end, we have proposed a traffic reduction method for crowdsourced multi-view video uploading [80], which employs correlation-based differential encoding with multiple references, exploiting the inter-camera correlations among the contributors. In differential encoding, one contributor sends its own stream while the other contributors overhear the transmitted stream and encode their streams using overheard one before transmission. By exploiting the inter-camera redundancy at the overhearing contributors, our scheme achieved better coding gain and higher traffic reduction. To realize the differential encoding across the

whole network of contributors, we consider three operations: correlation estimation, reference selection, and transmission order determination. First, we calculate the correlation degrees among the contributors based on the content features of the captured video by each contributor. Second, we select the number of references for differential encoding based on the correlation degrees of the contributors. Third, we determine the transmission order of the contributors so that the differential encoding opportunities in the network increase. Finally, each contributor encodes their video streams according to their dependency in the network and uploads the streams to a server through a wireless access point (AP). From the evaluation results, our proposed scheme achieved the traffic reduction of up to 31% along with the video quality improvement of up to 2.7 dB.

On the one hand, our method of traffic reduction only exploits the statistical redundancy [81] between the captured videos of the contributors that can be explored by statistical methods. On the other hand, visual perception of HVS turns out the perceptual redundancy in viewing a particular scene which is redundant information that cannot be perceived by human eyes. Many studies show that HVS is more sensitive to the spatial resolution of the currently gazing regions of the scene, and the sensitivity becomes decreased to the peripheral regions. This regional sensitivity of HVS can be utilized for improving the compression efficiency in video encoding maintaining the improved perceptual quality in the region-of-interest (ROI) at the cost of decreased quality in low-interest regions.

Many studies on ROI-based video compression have been proposed in the literature [82, 83, 84]. However, those studies are meant for the single-view video coding. Zhang et al. [85] proposed ROI-based video coding scheme for stereo multi-view videos by using the depth information to extensively improve data compression efficiency by exploiting redundancies in depth perception. Their work includes two main operations: depth perception ROI (DP-ROI) extraction and DP-ROI based multi-view video coding (RMVC) with optimized bit allocation. For the sake of reducing complexity in temporal and inter-view ROI extraction, ROI regions are extracted from the first frame of the first view, and it is tracked down subsequently in frame-by-frame and/or view-by-view basis. The problem of such inter-view ROI tracking is that error is propagated due to the long tracking length in the successive views. To avoid this problem, they

select the center view as the reference view for extracting ROI regions.

In this chapter, we present a fundamental discussion for crowdsourced multi-view video uploading by taking advantage of visual perception of the HVS. Specifically, we extend our previously proposed traffic reduction scheme [80] using a depth perceptual ROI-based multi-view video encoding as in [85]. Our goal is to achieve further traffic reduction with improved video quality in ROI regions at the cost of reduced quality in non-ROI regions. To extend our approach with ROI-based multi-view video encoding, we apply ROI extraction and tracking method proposed in [62]. Since we already employ differential encoding at the subsequent transmissions for exploiting inter-view redundancy, we can utilize (track) the already-extracted ROI from one contributor in the successive ones to reduce the complexity required for the extraction. On the other hand, the selection of reference view for ROI extraction in crowdsourced environment is not straightforward as in pre-deployed multi-view video environment. Specifically, the selection of the most central view of the scene is not so easy since it needs to know the relative positions of all the contributors in the scene. In addition, the correlations between the contributors cannot always be determined by their positions [80]. To solve this problem, we use the average correlation coefficient, which is derived from the correlation degrees among the contributors.

The entire video uploading process includes three main operations: correlation-estimation, ROI-reference selection, and ROI-based differential multi-view video encoding. Firstly, the correlation coefficients among the contributors are estimated using the content features of the captured videos. Secondly, we calculate the average correlation coefficient of each contributor using the values of correlation degrees and select the contributor with the largest average correlation coefficient as ROI-reference, i.e., a reference contributor for ROI extraction. Finally, the ROI-reference contributor extracts ROI from the first frame of the captured video, conducts ROI-based single-view video encoding in which the extracted ROIs are tracked at the successive frames, and uploads the stream to the server. The subsequent contributors overhear the transmission of the ROI-reference and conduct ROI-based multi-view video encoding with the prior ROI information before uploading.

5.2 Proposed System

5.2.1 Overview

Fig. 5.1 shows a model of our study. There are multiple mobile cameras (contributors) capturing videos from different angles at a certain crowded event such as a tournament. The contributors upload their captured videos to a video collector, that is a server, over wireless links. The collector is located at the event location and continuously gathers the videos from the contributors through a wireless AP. It is assumed that all the contributors can reach the collector in one-hop communication and uploading is error-free. Videos are uploaded in the group of pictures (GOP) basis.

For the first GOP, the uploading process consists of the four phases as described as follows: initialization, content features uploading, reference selection and notification, and video encoding and uploading.

Initialization: First of all, the AP broadcasts the request, Q, to obtain the number of contributors existed in its communication range. On receiving the request, each contributor notifies its position, e.g., GPS parameters, to the AP with the response, P.

Content Features Uploading: In this phase, the AP collects content features of the captured video from each contributor with the request, R. Each contributor uploads the features information in the form of Information Bound Reference (IBR) with the response, R_p. Using IBRs of multiple contributors, the AP calculates the correlation coefficients between any two contributors.

Reference Selection and Notification: With the knowledge of the correlation coefficients among the contributors, firstly, the AP constructs the correlation network of contributors. Next, the AP calculates the average correlation coefficient for each contributor and selects the contributor with the largest coefficient as the ROI-reference. Finally, the AP notifies the decision to all the contributors by means of a request, O.

Video Encoding and Uploading: In this phase, the contributors upload their videos according to the decision made by the AP. Firstly, the ROI-reference contributor extracts ROI and conducts ROI-based single-view video encoding and uploads the video stream to the AP. Subsequently,

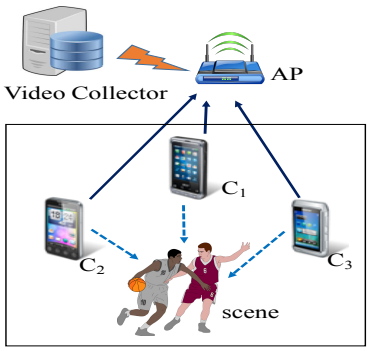


Figure 5.1: System model ($C_{i=1,...3}$ = contributors).

other contributors in the network overhear the transmission of the reference contributor and perform ROI-based multi-view video encoding before uploading.

Considering the correlation variation due to the mobility of each contributor, the IBR is refreshed for each GOP and the ROI-reference is reselected. After all the GOPs of captured videos from all contributors are uploaded, the AP updates the number of contributors in its communication range by restarting the initialization phase.

5.2.2 Correlation Estimation

In order to estimate the degree of correlation among the contributors, the content features of every first frame in each 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 image IBR.

5.2.3 ROI-Reference Selection

For the ROI-reference selection, the AP firstly constructs the correlation graph $G = (V, E)$, where $V = \{v_i,\ i = 1, 2, ..., |V|\}$ is the set of contributors and E is the set of edges which shows the correlation among the contributors. For each contributor v_i, an edge (v_i, v_j) is added to G if α_{v_i,v_j} is greater than the predefined correlation threshold of 0.625, because of no coding gain from differential encoding. For the sake of simplicity, in this work, we consider a fully connected network of contributors in which all the contributors in the network are correlated to each other, that is the correlation graph is complete. After constructing the correlated network,

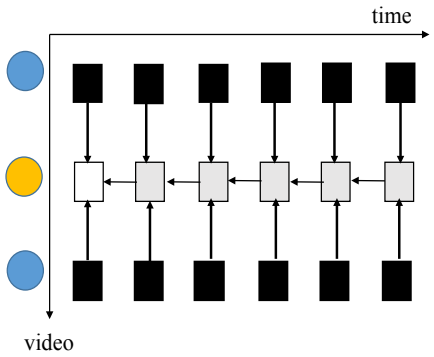


Figure 5.2: ROI extraction and tracking.

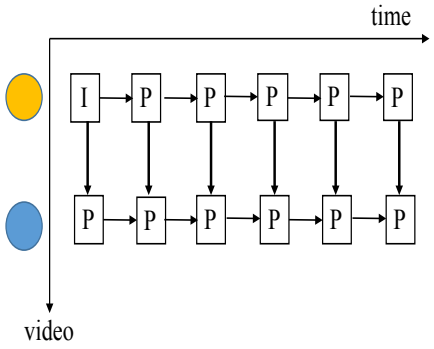


Figure 5.3: Prediction structure for differential encoding.

the AP selects the ROI-reference contributor using the Eq. (5.1):

$$\tilde{v}_i = \underset{i=1,\ldots,V}{\operatorname{argmax}} \quad \widetilde{\varphi}_i, \tag{5.1}$$

where $\widetilde{\varphi}_i$ is the average correlation coefficient of contributor v_i, which is obtained by the Eq. (5.2).

$$\widetilde{\varphi}_i = \frac{1}{V-1} \sum_{i \neq j} \alpha_{v_i, v_j} \tag{5.2}$$

5.2.4 Proposed System

After selecting the ROI-reference contributor, the AP informs the decision to the contributors. The decision initiates the uploading of videos from the contributors. Firstly, the ROI-reference

contributor conducts ROI extraction, ROI-based single-view video encoding and uploads the stream to the AP. Subsequently, the rest of the contributors overhear the ROI-reference's transmission and perform ROI tracking using the extracted ROI and differential encoding using it as reference. The ROI extraction at the reference contributor and ROI tracking in temporal and inter-view domains are depicted in Fig. 5.2, where yellow circle indicates ROI-reference and blue circle represents other contributors. The white rectangle represents the first frame of a GOP at the reference contributor that extracts ROI in its own, the gray rectangle represents a frame that extracts ROI by using the ROI of previous frame and the black rectangle represents a frame that extracts ROI by using the extracted ROI from reference contributor. The directed arrows represent the tracking directions. The detailed description of ROI extraction and tracking is described in [62]. In addition, the prediction structure for the differential encoding is illustrated in Fig. 5.3.

In HVS, ROI regions are more visually sensitive to distortion than non-ROI regions. To achieve better quality in ROI regions, many bits should be allocated to those regions while fewer bits should be allocated to non-ROI regions. For this purpose, different quantization parameters (QP) are used for ROI regions and non-ROI regions.

5.3 Evaluation

5.3.1 Settings

We evaluate the performance of our ROI-based differential crowdsourced multi-view video uploading scheme through the simulations implemented in MATLAB.

Metric: We evaluate the performance of our scheme in terms of video traffic and video quality. To consider the human visual perspective, we use peak signal-to-noise ratio within ROI region ($\mathrm{PSNR}_{\mathrm{roi}}$) and non-ROI region ($\mathrm{PSNR}_{\mathrm{nonroi}}$) which are defined as follows:

$$\mathsf{PSNR}_{\mathsf{roi}} = 10\log_{10}\frac{(2^L-1)^2}{\varepsilon_{\mathrm{MSE}_{\mathrm{roi}}}} \tag{5.3}$$

$$\mathsf{PSNR}_{\mathsf{nonroi}} = 10\log_{10}\frac{(2^L-1)^2}{\varepsilon_{\mathrm{MSE}_{\mathrm{nonroi}}}} \tag{5.4}$$

Table 5.1: Average correlattion of *Vassar* (5 cameras)

	Cam. 1	Cam. 2	Cam. 3	Cam. 4	Cam. 5
$\widetilde{\varphi_i}$	0.82	0.87	0.89	0.88	0.85

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

Video Sequence: We use video frames of 5 cameras from *Vassar* [71] in quarter common intermediate (QCIF) format. The resolution of each video is 174×144. The frame rate is 25 frames per second. The length of each GOP is 10 frames.

ROI-Reference: We select camera 3 as ROI-reference since it has the highest correlation coefficient as shown in Table 5.1. We define ROI region as a polygon $\mathrm{R(c, r)}$ where c and r are the pixel column and row indices of a given image frame. For the simulation, we set a rectangular ROI region with $\mathrm{c} = (58, 116)$ and $\mathrm{r} = (48, 96)$.

Reference Schemes:

1. Individual: This method is the baseline method for uploading the crowdsourced videos in which each contributor individually encodes its captured stream and uploads it to the AP.

2. Uniform quality: This method conducts differential encoding by overhearing the video stream from ROI-reference contributor with the uniform visual quality.

3. ROI-based: This method conducts differential encoding with different visual quality in ROI regions and non-ROI regions.

QP Values: For the uniform quality encoding, we use a QP of 20. For the ROI-based encoding, we use QPs of 20 and 28 for ROI regions and non-ROI regions, respectively.

5.3.2 Results

Table 5.2 summarizes the results of three reference schemes. It can be seen that ROI-based scheme achieves 57% of traffic reduction compared to individual uploading and 45% of traffic reduction compared to uniform quality uploading. On the other hand, the video quality of non-ROI regions in the ROI-based scheme is 4.82 dB lower than that of uniform quality uploading. However, in view of human visual perception, the quality degradation is hardly noticeable.

Table 5.2: Simulation results

	Avg. Traffic (bits)	$PSNR_{roi}$ (dB)	$PSNR_{nonroi}$ (dB)
Individual	26342	43.83	43.6
Uniform Quality	23172	44.11	43.85
ROI-based	11211	44.11	39.03

5.4 Summary

In this chapter, we present a fundamental discussion on crowdsourced multi-view video uploading. We extend our previously proposed differential encoding-based video uploading scheme to the ROI-based multi-view video encoding for achieving further traffic reduction. The results show that the ROI-based approach achieves 57% of traffic reduction compared to an existing scheme under the same visual quality in ROI regions and unnoticeable visual degradation in low-interest regions. In our future work, we will investigate the effects of network structure and correlation degrees between the contributors.

Chapter 6

Conclusion

6.1 Overall Summary

Chapter 1 summarizes the background and main contributions of this dissertation. We characterize the features of the crowdsourced multi-view video streaming system and its challenges for realization. The main issues on the practicability of crowdsourced video streaming are large amount of redundant traffic over bandwidth-limited wireless network and power-restriction of mobile devices. Considering these issues, we propose three video uploading schemes with the goal of providing the traffic-reduced, high-quality and low-power multi-view video uploading from crowdsourced video contributors.

Chapter 2 describes fundamental theories and techniques applied in this dissertation. First, we briefly discuss the procedures involved in single-view video encoding and multi-view video encoding. In addition, we shortly describe the region-of-interest based video encoding and its extension on multi-view video. Moreover, we present fundamental discussion of conventional digital-based video delivery and analog-based video delivery.

In Chapter 3, we describe our proposed content-aware video uploading scheme for crowdsourced multi-view video streaming with the goal of reducing the video traffic from crowdsourced contributors. 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.

In Chapter 4, we present cluster-based redirect video uploading scheme for high-quality and low-power crowdsourced multi-view video streaming. Evaluation results show that our

proposed scheme outperforms the conventional soft video delivery scheme when the channel quality difference between the direct and redirect paths increases. In addition, our scheme outperforms the digital- based video uploading schemes in terms of both video quality and power consumption.

In Chapter 5, we describe ROI-based crowdsourced multi-view video uploading scheme, which an extended version of our proposed scheme in Chapter 2. We consider that fact that the human visual system has more sensitive spatial resolution to the currently gazing region-of-interest (ROI) of the scene, and the sensitivity becomes decreased to the peripheral regions, creating the considerable perceptual redundancy. Evaluation results show that our approach achieves 57% of traffic reduction under the same visual quality in ROI regions and unnoticeable visual degradation in low-interest regions.

6.2 Future Works

The most important assumption of the proposed schemes in this dissertation is all the crowd-sourcing contributors are synchronized before the video uploading. While it is technically feasible, synchronizing the image/video capturing times across mobile phones' cameras is a challenging task as it is required that the contributors be synchronized to an accuracy that is the fraction of the duration of a frame [69]. The study in [75] also discusses the opportunities and challenges on the calibration and synchronization across mobile devices. Although Global Position System (GPS)-based or Network-based synchronization protocols such as Network Time Protocol (NTP) can achieve the synchronization to a high accuracy, for example, within 1 milliseconds, network latency and camera setup latency incurred by variablity of cameras may cause temporal differences between captured frames of the different mobile cameras [69]. The misalignment of captured frames can affect the correlation among the video streams, and thus, the performance of the proposed schemes. SocialSync [69] develops a subframe-based synchronization protocol for capturing images simultaneously using a smartphone camera network by estimating frame capture timestamps to within milliseconds accuracy. Such protocol could be applied to crowdsourced environment, on the other hand, its performance and impact on the proposed schemes are needed to evaluate, which will be considered in a future work.

In addition, the proposed schemes in this dissertation consider only one access point, which handles all the crowdsourcing video traffic in its transmission range. For the scalability to a large number of contributors, the proposed schemes can be extended to the multiple access points case. The management of the contributors across the multiple access points is left as a future work.

Bibliography

[1] [Online]. Available: http://ise.stanford.edu/Video/claire.qcif.gz

[2] J. Howe, *Wired Magazine*, no. 14, pp. 1–4, June 2006.

[3] D. C. Brabham, *Crowdsourcing*. Cambridge: The MIT Press, 2013.

[4] J. Phuttharak and S. W. Loke, "A review of mobile crowdsourcing architectures and challenges: Toward crowd-empowered internet-of-things," *IEEE Access*, vol. 7, pp. 304–324, December 2019.

[5] F. Chen, C. Zhang, F. Wang, and J. Liu, "Crowdsourced live streaming over the cloud," in *IEEE Conference on Computer Communications (INFOCOM)*, 2015.

[6] Y. Wu, Y. Wang, and G. Gao, "Photo crowdsourcing for area coverage in resource constrained environments," in *IEEE Conference on Computer Communications (INFOCOM)*, 2017.

[7] 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 Transactions on Multimedia*, vol. 18, no. 5, pp. 916–928, 2016.

[8] C. Zhang and J. Liu, "On crowdsourced interactive live streaming: a twitch.tv-based measurement study," in *25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 55–60, 2015.

[9] F. Zimmer, "A content analysis of social live streaming services," in *International Conference on Social Computing and Social Media*, pp. 400–414. Springer, 2018.

[10] C. Wu, Z. Wang, J. Liu, and S. Yang, "Crowdsourced live streaming over aggregated edge networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2016.

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

[12] X. Lu, Y. Wang, and E. Erik, "Power-efficient h.263 video transmission over wireless channels," in *International conference on Image processing*, Rochester, NY, USA, September 2002.

[13] M. Tang, S. Wang, L. Gao, J. Haung, and L. Sun, "Momd: A multi-object multi-dimensional auction for crowdsourced mobile video streaming," in *INFOCOM-2017*, Atlanta, GA, USA, May 2017.

[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 *17th International Conference on Advanced Communication Technology*, Seoul, South Korea, July 2015.

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

[16] K. Bilal, A. Erbad, and M. Hafeeda, "Crowdsourced multi-view live video streaming using cloud computing," *IEEE Access*, vol. 5, pp. 12 635–12 647, 2017.

[17] S. Kodera, T. Fujihashi, S. Saruwatari, and T. Watanabe, "Multi-view video streaming with mobile cameras," in *IEEE Global Communication Conference*, pp. 1412–1417, Austin, Texas, USA, December 2014.

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

[19] P. Wang, R. Dai, and I. F.Akyildiz, "A differential coding-based scheduling framework for wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 684–697, April 2013.

[20] R. N. R. Wagner and R. Baraniu, "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *IEEE International Conference on Image Processing)*, Barcelona, Spain, September 2003.

[21] M. Mu and C. Chen, "Collaborative image coding and transmission over wireless sensor networks," *EURASIP Journal on Advanced Signal Process*, 2007.

[22] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, November 2010.

[23] A. Anand, A. Balachandran, A. Akella, and S. Seshan, "Enhancing video accessibility and availability using information-bound references," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 1223–1236, April 2016.

[24] ——, "A case for information-bound referencing," in *ACM Special Interest Group on Data Communication (SIGCOMM) Hot Topics in Networks workshops (HotNets)*, Monterey, CA, USA, October 2010.

[25] C. H. Asia, J. S. C. Y. H. Wang, and T. Y. Tang, "Fast intra prediction mode decision algorithm for h.264/mpeg-4 avc," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, November 2007.

[26] A. K. Kannur and B. Li, "Power-aware content-adaptive h.264 video encoding," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, April 2009.

[27] A. Raha, H. Jayakumar, and V. Raghunathan, "A power efficient video encoder using reconfigurable approximate arithmetic units," in *27th International Conference on VLSI*

design and 13th International Conference on Embedded Systems, Mumbai, India, January 2014.

[28] W. Ji, J. Liu, M. Chen, and Y. Chen, "Enhancing video accessibility and availability using information-bound references," *Future Generation Computer Systems*, vol. 28, pp. 427–436, 2011.

[29] Z. Li, F. Thai, and A. K. Katsaggelos, "Joint video summarization and transmission adaptation for energy-efficient wireless video streaming," *Journal on Advanced Signal Processing*, no. 47, pp. 147–158, 2008.

[30] Y. Li, M. Reisslein, and C. Chakrabarti, "Energy efficient video transmission over a wireless link," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1229–1244, March 2009.

[31] O. Alaoui-Fdili, F.-X. Coudoux, Y. Fakhri, P. CorlayDriss, and Aboutajdine, "Energy-efficient joint video encoding and transmission framework for wvsn," *Multimedia Tools and Applications*, July 2017.

[32] N. Ding, D. Wagner, X. Chen, A. Pathak, Y. C. Hu, and A. Rice, "Characterizing and modeling the impact of wireless signal strength on smartphone battery drain," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 29–40, 2013.

[33] S. Jakubczak, J. Z. Sun, D. Katabi, and V. K. Goyal, "Performance regimes of uncoded linear communications over awgn channels," in *2011 45th Annual Conference on Information Sciences and Systems*, pp. 1–6, March 2011.

[34] S. Jakubczak and D. Katabi, "Softcast: Clean-slate scalable wireless video," in *The 17th Annual International Conference on Mobile Computing and Networking*, Las Vegas, NV, USA, September 2011.

[35] S. Jakubczak, H. Rahul, and D. Katabi, "One-size-fits-all wireless video," in *ACM HotNets*, pp. 1–6, 2009.

[36] X. L. Liu, W. Hu, Q. Pu, F. Wu, and Y. Zhang, "Parcast: Soft video delivery in mimo-ofdm wlans," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, pp. 233–244. ACM, 2012.

[37] H. Cui, C. Luo, C. W. Chen, and F. Wu, "Scalable video multicast for mu-mimo systems with antenna heterogeneity," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 992–1003, 2016.

[38] X. Fan, F. Wu, D. Zhao, and O. C. Au, "Distributed wireless visual communication with power distortion optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 6, pp. 1040–1053, 2013.

[39] J. Shen, L. Yu, L. Li, and H. Li, "Foveation-based wireless soft image delivery," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2788–2800, Oct 2018.

[40] T. Fujihashi, T. Koike-Akino, T. Watanabe, and P. V. Orlik, "Freecast: Graceful free-viewpoint video delivery," *IEEE Transactions on Multimedia*, September 2018.

[41] T. Fujihashi, M. Kobayashi, K. Endo, S. Saruwatari, S. Kobayashi, and T. Watanabe, "Graceful quality improvement in wireless 360-degree video delivery," in *IEEE Global Communications Conference*, Abu Dhabi, UAE, December 2018.

[42] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, July 1948.

[43] A. Ramachandran, *Decode to Encode*. Santa Clara, California, USA, 2018.

[44] K. N. Ngan, C. W. Yap, and K. T. Tan, *Video Coding for Wireless Communication Systems*. New York, NY, USA: Marcel Dekker, Inc., 2001.

[45] M. Jacobs and J. Probell, "A brief history of video coding," January 2009.

[46] S. Akramullah, *Video Coding Standards*. Berkeley, CA: Apress, 2014, pp. 55–100.

[47] M. Tanimoto, "Overview of ftv (free-viewpoint television)," in *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo*, ser. ICME'09, pp. 1552–1553, New York, NY, USA, 2009.

[48] A. Hilton, J.-Y. Guillemaut, J. Kilner, O. Grau, and G. Thomas, *Free-Viewpoint Video for TV Sport Production*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 77–106.

[49] G. Tech, Y. Chen, K. Muller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3d extensions of high efficiency video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 1–1, January 2015.

[50] Y.-S. Ho and K.-J. Oh, "Overview of multi-view video coding," pp. 5 – 12, 07 2007.

[51] Joint Video Team Of ITU-T VCEG And ISO/IEC MPEG, *JMVC (Joint Multiview Video Coding) Software*, 2008.

[52] Joint Video Team ISO/IEC JTC1/SC29/WG11 MPEG2005/N7567, *Updated Call For Proposals On Multi-view Video Coding*, 2005.

[53] R. Bruno and M. Nurchis, "Overview of multiview video coding and anti-aliasing for 3d displays," in *2007 14th IEEE International Conference on Image Processing (ICIP 2007)*, Texas, USA, September 2007.

[54] P. Merkle, H. Brust, K. Dix, K. Muller, and T. Wiegand, "Stereo video compression for mobile 3d services," in *2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 1–4, May 2009.

[55] J. Lee and T. Ebrahimi, "Perceptual video compression: A survey," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 684–697, Oct 2012.

[56] D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer Publishing Company, Incorporated, 2013.

[57] Weiping Li, "Overview of fine granularity scalability in mpeg-4 video standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317,

March 2001.

[58] M. Meddeb, "Region-of-interested-based video coding for video conference applications." Ph.D. dissertation, Paris Institute of Technology, 2016.

[59] N. B. Zahir, R. Samad, and M. Mustafa, "Initial experimental results of real-time variant pose face detection and tracking system," in *2013 IEEE International Conference on Signal and Image Processing Applications*, pp. 264–268, Oct 2013.

[60] I. Himawan, W. Song, and D. Tjondronegoro, "Automatic region-of-interest detection and prioritisation for visually optimised coding of low bit rate videos," in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 76–82, Jan 2013.

[61] N. Kiryati, T. R. Raviv, Y. Ivanchenko, and S. Rochel, in *2008 19th International Conference on Pattern Recognition*, pp. 1–4, Dec 2008.

[62] Y. Zhang, M. Yu, G. Jiang, Z. Peng, and Y. Yang, "Low-complexity region-of-interest extraction for multiview video coding," in *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on*, pp. 1–5. IEEE, 2009.

[63] S. Pudlewski, N. Cen, Z. Guan, and T. Melodia, "Video transmission over lossy wireless networks: A cross-layer perspective," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 6–21, Feb 2015.

[64] S. Faruque, *Introduction to Channel Coding*. Cham: Springer International Publishing, 2016, pp. 1–16.

[65] M. Schnell, "Hadamard codewords as orthogonal spreading sequences in synchronous ds cdma systems for mobile radio channels," in *Proceedings of IEEE 3rd International Symposium on Spread Spectrum Techniques and Applications (ISSSTA'94)*, pp. 505–509 vol.2, July 1994.

[66] N. Ozbek and A. Tekalp, "Fast multi-frame reference video encoding with key frames," in *13th European Signal Processing Conference, EUSIPCO 2005*, Antalya, Turkey, Sepember 2005.

[67] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. United Kingdom: John Wiley and Sons, 2003.

[68] T. T. Nu, T. Fujihashi, and T. Watanabe, "Content-aware efficient video uploading for crowdsourced multi-view video streaming," in *International Workshop on Computing, Networking and Communication (CNC)*, Maui, Hawaii, USA, March 2018.

[69] R. Latimer, J. Holloway, A. Veeraraghavan, and A. Sabharwal, "Socialsync: Sub-frame synchronization in a smartphone camera network," in *European Conference on Computer Vision 2014 Workshops*, pp. 561–575, 2015.

[70] S. Kodera, T. Fujihashi, S. Saruwatari, and T. Watanabe, "Video similarity based wireless multi-view video streaming," *IPSJ Journal*, vol. 37, no. 7, pp. 1–8, July 2016.

[71] ISO/IEC JTC1/SC29/WG11, *Multi-view Video Test Sequences from MERL*, 2005.

[72] P. Wang, R. Dai, and I. F.Akyildiz, "A differential coding-based scheduling framework for wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 684–697, April 2013.

[73] T. Ma, M. Hempel, D. Peng, and H. Sharif, "A survey of energy-efficient compression an communication techniques for multimedia in resource systems," *IEEE Communication Surveys and Tutorials*, vol. 15, no. 3, pp. 963–972, Third Quarter 2013.

[74] A. Ukhanova, E. Belyaev, L. Wang, and S. Forchhammer, "Power consumption analysis of constant bit rate video transmission over 3g networks," *Computer Communications*, vol. 35, no. 14, pp. 1695–1706, 2012.

[75] A. Prati and F. Z. Qureshi, "Integrating consumer smart cameras into camera networks: Opportunities and obstacles," *Computer*, vol. 47, no. 5, pp. 45–51, May 2014.

[76] V. Geetha, P. V. Kallapur, and S. Tellajeera, "Clustering in wireless sensor networks: performance comparison of leach & leach-c protocols using ns2," *Procedia Technology*, vol. 4, pp. 163–170, 2012.

[77] M. Ye, C. Li, G. Chen, and J. Wu, "An energy efficient clustering scheme in wireless sensor networks," *Ad Hoc and Sensor Wireless Networks*, vol. 3, pp. 99–119, April 2006.

[78] A. A. Muhit, "H. 264 baseline codec v1," 2013.

[79] A. Myronenko, "Multidimensional Discrete Cosine Transform (DCT)," 2010.

[80] T. T. Nu, T. Fujihashi, and T. Watanabe, "Content-aware efficient video uploading for crowdsourced multi-view video streaming," in *International Workshop on Computing, Networking and Communication (CNC)*, Maui, Hawaii, USA, March 2018.

[81] J. Shen, L. Yu, L. Li, and H. Li, "Foveation based wireless soft image delivery," *IEEE Transactions on Multimedia*, 2018.

[82] H. Wang, Y. Liang, and K. El-Maleh, "Real-time region-of-interest video coding using content-adaptive background skipping with dynamic bit reallocation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 2. IEEE, 2006.

[83] M. Wang, T. Zhang, C. Liu, and S. Goto, "Region-of-interest based dynamical parameter allocation for h. 264/avc encoder," in *Picture Coding Symposium, 2009. PCS 2009*, pp. 1–4. IEEE, 2009.

[84] F. Zhang and D. R. Bull, "A parametric framework for video compression using region-based texture models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1378–1392, 2011.

[85] Y. Zhang, G. Jiang, M. Yu, Y. Yang, Z. Peng, and K. Chen, "Depth perceptual region-of-interest based multiview video coding," *Journal of Visual Communication and Image Representation*, vol. 21, no. 5-6, pp. 498–512, 2010.