Statistic oriented Video Coding and Streaming Methods with Future Insight

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy
by

Li Yu

Department of Electrical Engineering and Electronics
School of Electrical Engineering and Electronics and Computer Science
University of Liverpool

November 30, 2016

Abstract

As indicated by Cisco, IP video traffic represents 70 percent of all consumer Internet traffic in 2015 globally, and it is expected to reach 82 percent by 2020. Given this, research works related to video compression, video transmission, and interactive playback are of vital importance. Most existing works solve one step of these tasks based on the currently and/or previously acquired information. One common challenge behind all these tasks is the uncertainty in the future. For example, the dynamic adaptive video streaming over HTTP (DASH) standard provides multiple quality levels for each video block to choose. The benefit of various options is that it can adapt to the bandwidth fluctuation and various client device capacity. Most methods predict the bitrate of future video blocks according to the already downloaded ones, which is usually unprecise. As a result, the mismatch between the predicted and actual bitrate of the chosen video block leads to latency or inefficient usage of the bandwidth. Thus, one of our work proposes to send the exact bitrate information of all video blocks to the client at the beginning to avoid such problems. To sum up, the focus of this thesis is to solve the video coding and streaming problems with future insight. By analyzing the uncertainties of future information in a statistical way, more efficient and suitable solutions are derived. In this thesis, how each problem is solved with future insight is described respectively.

As for video compression, inter prediction is one of the biggest contributors to the compression ratio, which removes temporal redundancies between frames. However, it is also one of the most computational complex processes. Thus, the ideal scenario is that the inter prediction is only performed within necessary areas, where there exist similar contents for reference. However, the existing encoding standards, such as H.264 and H.265, simply uses the inter prediction for all reference frames following a fixed prediction structure. Thus, it is a waste of resources to perform inter prediction in these unnecessary areas that have less probability of being referenced. Inspired by this

idea, a statistical approach for motion estimation skipping (SAMEK) is proposed to recognize these unnecessary areas and avoid using them in the motion estimation stage while encoding future frames. By doing so, the overall complexity and encoding time are reduced.

After the compression process (source coding), the channel coding is needed to protect video contents when they are transmitted over unreliable networks. Reed-Solomon (RS) erasure code is one of the most popular errors correcting codes, which detects and recovers the erasures by adding parity packets. These parity packets should be optimally allocated according to the importance of each video packet. The importance of each packet can be evaluated through its influence on the quality of the whole video. Thus, by knowing the future potential influence of each packet, a rate-distortion optimized redundancy allocation scheme is proposed to automatically allocate parity packets based on the network conditions and video characteristics.

RS based error control mechanisms are usually used for real-time streaming over the unreliable networks, such as IP, UDP; whereas for delay insensitive video streaming over reliable protocol TCP, DASH is commonly adopted. The DASH is the de-facto video delivery mechanism nowadays, which takes advantage of the existing low cost and widespread HTTP platforms. So far, most DASH works focus on the CBR (constant bitrate) video delivery. The bit rate of CBR video is kept constant over each segment. In this thesis, VBR (various bitrate) video delivery is investigated instead. Since the quality is kept constant in VBR video, the bit rate of each segment fluctuates. Thus, it is important to know the instant bit rate of future segments beforehand. In the proposed method, such accurate bit rate information of every segment is sent at the beginning of a streaming session. Then, the proposed internal QoE (Quality of Experience) goal function would take the expected future influence of each request over buffer reservation into consideration.

In addition to effective video streaming, user demands are increasing with the emergence of interactive multiview video streaming platforms, which provides immersive vision, seamless view switching, and interactive involvement. A probabilistic navigation model, which predicts the views that might be watched by the user, is incorporated in the proposed convolutional neural network (CNN) assisted seamless multiview video streaming and navigation system to guide the download of future views. In addition, a bit allocation mechanism under the guidance of the navigation model is developed

to prefetch all possibly being watched views and adapt to the network fluctuations at the same time. Besides, a convolutional neural network assisted multiview representation method is proposed to prepare the multiview videos at the server. The proposed representation method would maintain a satisfactory compression efficiency and allow random access to any subset of views with dynamic qualities at the same time. All the above methods work closely to provide a seamless viewing experience to users. They can be fused into any existing multiview video streaming frameworks to enhance the overall performance.

The main contribution of this thesis is incorporating future insight into various tasks related to video coding and streaming. By leveraging the methods proposed in this thesis, efficient results could be obtained in different application scenarios. For example, with the proposed SAMEK method, up to 9.5% encoding time (averagely 6.87%) is saved with negligible rate-distortion losses (in average 0.006 dB) when compared with classical HEVC encoder. With the proposed RS redundancy allocation scheme, an average gain of 1dB over the state-of-the-art approach is achieved. The proposed multiview video streaming and navigation system enhances the overall quality over benchmark with averagely 0.6 dB with a lower bitrate.

Contents

List of Tables

List of Figures

List of Abbreviations

AMVP	Advanced motion vector prediction
ARQ	Automatic Repeat-reQuest
AVC	Advanced Video Coding
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context- Adaptive Variable-Length Coding
CB	Coding Block
CBF	Coded Block Flag
CBR	Constant Bit Rate
CDN	Content Delivery Network
CIF	Common Intermediate Format
CNN	Convolutional Neural Network
CTB	Coding Tree Block
CTU	Coding Tree Unit
CU	Coding Unit
DASH	Dynamic Adaptive Video Streaming over HTTP
DCT	Discrete Cosine Transform
DIBR	Depth Image Based Rendering
DPB	Decoded Picture Buffer
DST	Discrete Sine Transform
FEC	Forward Error Correction
fps	frame per second
GOP	Group of Pictures
HEVC	High Efficiency Video Coding

i.i.d	independent identically distributed
IP	Internet Protocol
KLT	Karhunen-Loeve Transform
MB	Macroblock
MDC	Multiple Description Coding
MDP	Markov Decision Process
ME	Motion Estimation
MPD	Media Presentation Description
MSE	Mean Squared Error
MV	Motion Vector
MVC	Multiview Video Coding
NAL	Network Abstraction Layer
PB	Prediction Block
PLR	Packet Loss Rate
PSNR	Peak Signal-to-Noise Ratio
PU	Prediction Unit
QoE	Quality of Experience
QoS	Quality of Service
QP	Quantization Parameter
RDO	Rate-Distortion Optimization
RPS	Reference Picture Selection
RS	Reed-Solomon Code
RTCP	Real Time Control Protocol

RTP	Real Time Protocol
RTT	Round-Trip Time
SAMEK	Statistical Approach for Motion Estimation Skipping
SAO	Sample Adaptive Offset
SGD	Stochastic Gradient Decent
SSIM	Structural Similarity
TCP	Transmission Control Protocol
TB	Transform Block
TU	Transform Unit
VLC	Variable-length Coding
UDP	User Datagram Protocol
ULP	Unequal Loss Protection
URQ	Uniform reconstruction quantization
VBR	Various Bit Rate

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my primary Ph.D. supervisor Prof. Tammam TILLO for his continuous support of my Ph.D. study and research, for his patience, guidance, encouragement, kindness, and immense knowledge. His guidance helped me in all the time of my research. Besides my primary supervisor, I would like to thank my co-supervisor, Dr. Waleed Al-Nuaimy. Meanwhile, I hope to thank Lecturer Jimin XIAO for his substantial support of my Ph.D. study and research.

My sincere thanks also go to Prof. Ce ZHU in University of Electronic Science and Technology of China and Prof. Marco GRANGRTTO in the University of Turin, for offering me the opportunity to visit their research groups and giving me advice on my research.

Then, I also want to thank our research partners at Beijing Jiaotong University, including professor Yao ZHAO and Dr. Chao YAO. Meanwhile, I hope to thank my Ph.D. student colleagues in the department, Fei CHENG, Boyuan SUN, Samer JAMMAL, Yanchun XIE for their help and company during this journey.

Last but not the least, I would like to thank my parents for their continuous support, selfless love and infinite encouragement. Without them, I cannot finish my thesis.

At the end, I want to express my gratitude again to all the people supporting me and best wishes to all of you.

Chapter 1

Introduction

1.1 Background

Globally, IP video traffic made 70 percent of all consumer Internet traffic in 2015, which is expected to reach 82 percent by 2020 [3]. A more vivid evidence provided by Cisco of the prevalence of Internet video traffic is that nearly a million minutes of video content will cross the network every second by 2020. These figures demonstrate the popularity of internet video, as well as the importance of its related research.

As for videos, there are various categories, including user-generated videos, video calling, live internet TV and ambient videos. These heterogeneous video types derive different challenges to the video coding standards. Some of them require real-time encoding, while others require multiple quality versions. Besides, the number of devices connected to IP networks is predicted to be three times the global population in 2020. In addition to the challenges posed by the huge amount of data and connected devices, the huge diversity of devices adds one more dimension to the complexity space of the video coding standards. The simplest approach to tackle this problem is to prepare videos with various resolutions. The various video versions also cause challenges on the video streaming systems, since they have diversified bit rates. In addition, there are various transmission mediums, like the packet switched network and the circuit switched network. These networks have their unique characteristics. For example, the packet switched network causes packets delay which may lead to packets arriving out of order. Meanwhile, both networks are vulnerable to to errors and losses. In the circuit switched network this happened at bit level, whereas, it happened at packet level in packet switched network. All these issues need to be considered in the design of a video streaming system. Error correction mechanisms also play an important

role, especially when the network suffers from errors and losses. Furthermore, user demands for immersive and interactive watching experience call for the development of 3D videos. The challenge of large data volume for both coding and streaming comes along. Besides, user navigation models are needed to guarantee a good interactivity as well. To sum up, the era of video poses both opportunities and challenges to the research of video coding, streaming, and protection.

In the following, a general introduction to video communication systems is presented. As shown in Fig.1.1, a video communication system typically includes six stages, with the navigation as an optional choice. The video is firstly compressed by the video source encoder to reduce the bit rate. Plenty of video coding standards are developed, with H.265 (High Efficiency Video Coding, HEVC) [4] and H.264 (MPEG-4 Part 10)[5] as the most employed ones nowadays. Then, the bit stream is segmented into packets with fixed or variable length. As these compressed videos have little redundancy with respect to the original contents, they are more vulnerable to error or packet loss than original ones. One well-known consequence of error affecting video is the drifting phenomenon [6, 7]. Thus, before transmitting them through an unreliable network, channel coding is performed to protect them from errors. The Forward Error Correction (FEC) method is one commonly used protection method. After channel coding, the packets are sent to the clients through the network. Based on different network protocols, such as UDP, RTP, and HTTP, different video streaming methods are used. Upon receiving these packets, the client first executes channel decoding to recover lost packets. For those packets that cannot be recovered, error concealment is usually used to predict the lost regions. Next, the source packets are video decoded using the corresponding video codec. Finally, the decoded video contents are displayed in chronological order. As for 3D videos or multiview videos, a navigation model is used to monitor the user's view angle and guide the media player to display views accordingly. The whole system tries to provide a good watching experience to the client, which usually requires good video quality and low latency.

1.2 Motivations

As shown in the background, each stage of the video communication system is an important unit which needs to handle many challenging tasks. Meanwhile, the coherent cooperation between them is also a non-trivial task. Both the challenges in each stage

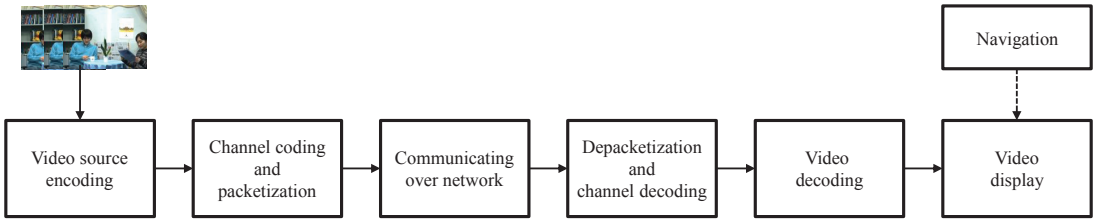


Figure 1.1: A typical video communication system over unreliable networks.

and in the cooperation are tough, owing to the contradictions within and between different stages respectively. For the former one, the conflict between goals of achieving high quality and low latency within the video streaming system is one example. While one example for the latter case is that the goal of source coding is compressing the data and reducing the overall bit rate; however, additional bits would be added to protect the source data in the channel coding stage, which is contrary to the goal of source coding. Thus, the improvement of the whole system lies in resolving these contradictions and targeting the global optimization. In this thesis, some contradictions, including internal and external ones, are investigated by introducing future insight into our proposed solutions. The detailed explanations are presented in the following paragraphs.

- **Internal Conflict: Encoding Complexity versus Compression Ratio.** In general, higher compression ratio corresponds to higher encoding complexity, which would require a longer processing time and more powerful computational devices. This trend is obvious with the updating of each generation in video coding standard. For example, H.265 doubles the compression ratio of H.264, along with a much higher computational complexity. This phenomenon is reasonable given that the compression comes from the reduction in spatial and temporal redundancies by using time-consuming searching process. Accordingly, more redundancy reduction corresponds to more efforts in searching. Thus, one of our work in this thesis tries to relieve the contradiction between encoding complexity and compression ratio. By introducing future insight into the inter prediction process, complexity is reduced with nearly no loss in compression efficiency.

- **Internal Conflict: High Quality versus Low Latency.** The goals of any video streaming system include providing video with high quality and low latency. However, given the limited bandwidth resources, these two goals collide with each other. Videos with higher quality require more bits to represent them.

Meanwhile, more bits would take more time to be transmitted over the network, thus leading to delay with higher probability. How to accommodate both goals is important in a video streaming system. In this thesis, we propose a solution within the DASH framework (the de-facto video streaming architecture nowadays). An internal QoE function which considers the current and future several steps in the streaming procedure is proposed to take care of both goals.

- **External Conflict: Compression versus Protection.** To protect compressed data, channel coding is used; but it increases the data volume, which is conflicting with the goal of source coding process. The reason behind this design (source coding followed with channel coding) is that the redundancy is reshaped in an organized way to protect the video information with a good efficiency. Thus, the main target of channel coding is to protect the source data in an efficient way and constrain the influence of error/loss within a reasonable range. The influence of error/loss over future frames is called error propagation. Thus, in this thesis, error propagation over future frames is taken into consideration in the channel coding process to increase the efficiency of redundancy allocation.

- **External Conflict: Free Navigation versus High Quality.** Similar to the second item, this conflict originates from the limited bandwidth resources. In a multiview video streaming system, free navigation requires a wide range of views for real-time switching. Usually, the data volume increases with the growth of required view numbers. Thus, given a fixed bandwidth, the bits assigned to each view decreases with the increase of view number. Accordingly, the overall quality is sacrificed. To tackle this challenge, many aspects need to be optimized. In this thesis, we propose to solve this problem from two aspects with the consideration of future navigation predictions.

To sum up, this thesis is motivated by resolving the contradictions in the video streaming system. These contradictions mainly come from the limited resources. However, the final target is the same, which is providing good watching experience with resource constraints. Thus, we tried to solve the conflicts by focusing on the final target. The final target is made accessible and feasible by introducing future insight over following several steps into each sub-target. An overview of the thesis is provided in the next section.

1.3 Overview of The Thesis

1.3.1 Contributions

In this thesis, we aim to enhance the overall performance of video communication system with future insight. Thus, how to introduce future insight into each stage and improve the performance is our objective. The main contributions of the thesis are:

- **Statistical Approach for Skipping Motion Estimation (Source Encoding Stage):** HEVC is the state-of-the-art video coding standard, which has achieved significant rate-distortion improvement over the previous standard H.264/AVC. However, the complexity, as well as long encoding time, are obstacles for its wide application. To reduce the overall complexity and encoding time, a statistical approach for motion estimation skipping (SAMEK) is proposed in this thesis. It tries to avoid unnecessary motion estimation (ME) in units with less probability of being referenced by future frames based on statistical analysis. The responsibility of ME is searching for the most similar video block among all the reference frames, which is one of the greatest contributors to the overall complexity. Thus, by tackling this critical part, up to 9.5% encoding time (average 6.87%) is saved with negligible rate-distortion losses (average 0.006 dB loss) when compared with classical HEVC encoder.

- **Future RD Influence guided Dynamic Redundancy Allocation Using SUB-GOP Based FEC Code (Channel Coding Stage):** Reed-Solomon (RS) erasure code is one of the most popular protection methods for video streaming over unreliable networks. As a block-based error correcting code, the large block size and the increased number of parity packets enhance the protection performance. However, this enhancement is sacrificed by error propagation over future frames and increased bit rate. To tackle this paradox, we propose a rate-distortion (RD) optimized redundancy allocation scheme, which not only considers the distortion caused by losing each slice but also the propagated errors over future slices. In this scheme, the redundancy allocation problem is transformed into a constraint optimization problem, which allows more flexibility in setting the block-wise redundancy. As demonstrated by massive experiments, an average gain of 1dB over the state-of-the-art approach is achieved.

- **QOE-Driven Dynamic Adaptive Video Streaming Strategy with Future Information (Video Communication Stage):** Dynamic Adaptive Video Streaming over HTTP (DASH) has become the de-facto video delivery mechanism nowadays, which takes advantage of the existing low cost and widespread HTTP platforms. Standards like MPEG-DASH defines the bitstreams conformance and decoding process, while leaving the bitrate adaptive algorithm open for research. So far, most DASH research focus on the CBR video delivery. In this thesis, VBR video delivery is investigated. The detailed instant bit rate of future segments are exploited in the proposed adaptation method to grasp the fluctuation traits of the VBR video. Meanwhile, the adaptation problem is formulated as an optimization process with the proposed internal QoE function. The internal QoE function of each segment is optimized for a series of future segments by simulation. It is worth to notice that, the internal QoE function not only keeps a good balance between various requirements, such as average quality and starvation, but also maintains a sustainable buffer reservation for the future streaming. The experimental results demonstrate that our proposed QoE-based video adaptation method outperforms the state-of-the-art method with a good margin.

- **Navigation Guided Multiview Video Streaming representation and bit allocation method (Video Source encoding and Display Stage):** Multiview video streaming has gained popularity for its great viewing experience, as well as its availability enabled by increasing network throughput and technical development [8, 9]. Increasing user demands are emerging for the interactive multiview video streaming, which provides seamless view switching upon request. However, it is a challenging task to stream stable and high-quality videos that allow real-time navigation within the bandwidth constraint. Faced with this challenge, a navigation-guided multiview video streaming system that allows seamless view switching is proposed in this thesis. The proposed method tries to predict the views that might be watched in the future in a probabilistic way. By prefetching video chunks according to this prediction, a seamless view switching experience is provided. In order to achieve better viewing quality within limited bandwidth resources, a multiview representation method and a bit allocation mechanism are designed. With the CNN assisted multiview representation method, flexible ac-

cess to a random subset of views is guaranteed without losing multiview video compression efficiency. The navigation model guided bit allocation mechanism assigns bit rate among views according to their probability of being watched so as to minimize the overall distortion. Meanwhile, the total bit rate is adapted following the network fluctuations. These two parts work closely to provide an optimized viewing experience to users. They can be fused into any existing multiview video streaming frameworks to enhance the overall performance. The effectiveness of the proposed methods is demonstrated with the experimental results.

1.3.2 Organization of This Thesis

The thesis is organized as follows. Chapter 2 gives a brief overview of the video compression and communication. Then, Chapter 3 to 6 presents the aforementioned four different works in detail respectively. Finally, both contributions and limitations of the work are summarized in Chapter 7. Besides, future works are also provided in this chapter.

Chapter 2

Overview of Video Compression and Communication

Nowadays, video related applications are prevalent. This prevalence not only boosts the growth of related industries but also stimulates the development of related research. In the following, the video compression methods will firstly be introduced, including the commonly used video compression standards and the extensions for multiview video encoding. Secondly, the error correcting code for video transmission, especially FEC, will be reviewed. Thirdly, the video streaming methods will be reviewed along with its evolving history. Meanwhile, a specific investigation of the state-of-the-art streaming technique, the dynamic adaptive video streaming, is presented. Finally, the quality assessment criteria for video streaming, QoE, is discussed at the end.

2.1 Video Coding

This section presents an overview of video coding and video compression standards for the single view and multiview videos. The principles and methodologies behind the most eminent video compression standards are identical, which will be explained in the overview section. Then, the characteristics of two state-of-the-art video coding standards are described respectively. Finally, extensions of the video compression standard for multiview video coding are introduced.

2.1.1 Overview

The video is a sequence of images displayed in order, with each one of these images called one frame. Typically, 30 frames are displayed every second, and the number of frames per second is denoted as frame rate. The raw data size of the video is huge,

which causes a burden for both transmission and storage. Thus, several compression techniques are developed to reduce the data size while maintaining the video quality. Based on the degree to which the video quality is maintained after compression, the compression methods can be classified into two categories:

- **lossless video compression:** The methods belonging to this category guarantee the non-destructive influence on the output result. In other words, the decompressed output is identical to the original image, where compression and decompression are reversible processes. It is widely used in applications where high quality is required, such as compression of medical images.

- **lossy video compression:** The methods within this category obtain high compression efficiency in trade of some losses in quality. The eminent methods try to maintain the quality by distributing losses over insignificant parts, such as the chroma components Cb/Cr and the details of texture. The modifications within these parts are usually perceptually irrelevant. These methods are widely used for the compression of the common videos.

No matter which compression method is employed, the coding system is referred to as a codec, which consists of an encoder and a decoder [10]. The codec is mainly characterized in terms of:

- **Throughput:** It is influenced by the bit rate of the transmission channel, as well as the overhead of protocols and channel coding.

- **Distortion of the decoded video:** This characteristic is influenced by the codec and transmission errors in the network. As for the lossless video coding method, it is only influenced by the transmission errors.

Thus, the video compression process is actually a tradeoff between bit rate and distortion. Based on different application scenarios, different video compression methods are proposed. The techniques used for the digital compression can be classified as follows:

- **Prediction-based:** There are many similar patterns within one frame, such as the blue sky area and the wall area, etc. These repeated contents are called spatial redundancies. While for the consecutive frames, the contents are always similar except for some minor differences. Especially for the video recorded with

a fixed camera, the background remains the same. The repeated contents from frame to frame are called temporal redundancies. Both the spatial and temporal redundancies can be reduced by the prediction process, and the corresponding processes are called intra prediction and inter prediction, respectively. After this process, only the prediction vector and the differences from the predicted values, i.e. residual image, need to be encoded. The summation of these two items are much smaller than the original data. Therefore, the bit rates are reduced without any quality degradation.

- **Transformation-based:** The purpose of the transformation is to reduce the spatial redundancies by capturing the essence of the signal using frequency analysis. It converts the image or residual image into the transform domain for further processing. The transformation method is chosen based on the following criteria:

  1. In this transform domain, data is uncorrelated and compact. The data is separated into components with minimal inter-dependence, which guarantees the uncorrelated characteristic. While most of the energy would be concentrated into a limited number of values in the transformed components to achieve compactivity.
  2. It is a reversible process.
  3. The operations are computationally tractable.

  Two famous examples of the transformation are Discrete Cosine Transform (DCT) [11] and Karhunen-Loeve Transform (KLT).

- **Quantization-based:** The process rounds off the less important digits with a certain step size, so as to reduce the range of values to be represented. Thus, fewer bits are needed to represent the reduced range. Therefore, the data is compressed. This is an irreversible process, which loses precision as a cost. There are two categories of quantization:

  1. Scaler quantiser: It maps the input signal to the quantized output value by sampling with the same quantization parameter (QP). For example, $Y = QP \times round(X/QP)$, where X and Y are input and output respectively.
  2. Vector quantiser: It maps a group of input samples to a group of quantized values. At the encoder side, a set of input data is mapped to a single value,

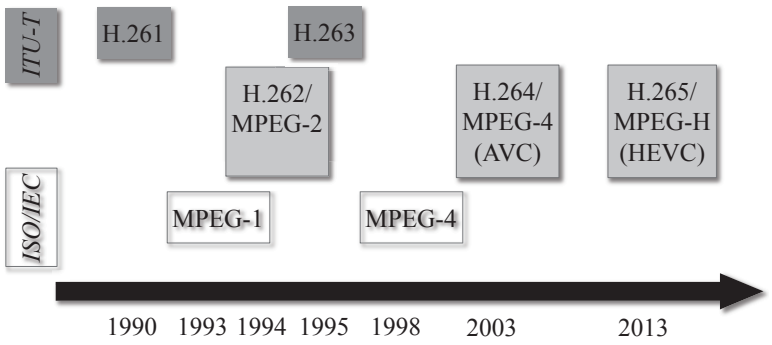


Figure 2.1: Chronology of International Video Coding Standards.

which is the index of a vector in the codebook that matches current input as closely as possible. While at the decoder, an output is reconstructed using the selected vector.

- **Entropy coding:** It converts the data into a bitstream that is suitable for transmission and storage, by taking advantage of the probability. Variable-length Coding (VLC) is a well-known entropy code. It represents symbols that frequently occur with short binary strings, while uses long binary strings to represent symbols that are less likely to appear.

2.1.2 Standards

There are two major organizations that develop the video compression standards. The first organization is International Telecommunications Union - Telecommunications Standardization Sector (ITU-T). Its Video Coding Experts Group (VCEG) mainly releases recommendation documents for video compression. The second one is International Organization for Standardization, in cooperation with International Electrotechnical Commission (ISO/IEC). Its corresponding sector is called Moving Picture Experts Group (MPEG), which issues standards of video coding. A chronology of international video coding standards is shown in Figure 2.1. In each standard, only the bitstream syntax and the decoding process are defined, which would leave space for the performance optimization and the complexity reduction during implementation.

The first compression standard that gained wide acceptance was H.261 [12], which builds the typical encoder structure that is still predominantly used today. It primarily targets the video telephony and teleconferencing applications over ISDN network. While MPEG-1 [13] is widely used for storage and retrieval of video and audio at a

bit rate of 1.5 Mbps. It defines the hybrid combination of block-based DCT and motion estimation/compensation. Besides, it allows the functionality for random access of the stored media. When it comes to H.262/MPEG-2 [14], the broadcasting of digital television is affordable with the increased bitrate, which is $2 - 20$ Mbps. Besides, the extension for scalable encoding is provided in this standard. In H.263 [15], video telephony at low data rate, i.e. 33 or 56 Kbps, is available. In this standard, flexible prediction options are provided. Besides, the error resilience mechanism is offered for transmission over unreliable networks. MPEG-4 [16] features the object-based coding scheme, encoding of synthetic (computer generated) contents, as well as content-based interactivity. As for H.264/MPEG-4 Part 10 [17], improved video compression efficiency is offered with the emerging of transmission technologies, like cable and wireless networks. The main goal is to provide 'network-aware' video representations with increased compression performance for both interactive and non-interactive applications. Finally, the state-of-the-art video coding standard, HEVC, further improves the compression efficiency with 50% bit rate reduction for an equal perceptual quality, compared to H.264/MPEG-4.

In the following paragraphs, a generic model that is compatible with major video coding standards and released since early 1990s is introduced. Then, concepts related to video compression, such as frame types, are explained.

Hybrid Codec Model

The aforementioned standards are built upon the same generic model/design of a codec, which mainly consists of motion estimation and compensation, transformation, and entropy coding. The DCT is usually used as the transformation method. Thus, this generic codec model is named as the hybrid DCT Codec. The illustrations of its encoder and decoder are shown in Figure 2.2 and 2.3 respectively.

As depicted in Figure 2.2, there are two flow paths for the hybrid encoder: encoding (left to right) and reconstruction (right to left). The operating process of the encoding path [1] is as follows:

1. The input frame F_n is divided into macroblocks. For example, each macroblock is a 16×16 region.

2. The macroblock in F_n is compared with the macroblock candidates in a reference

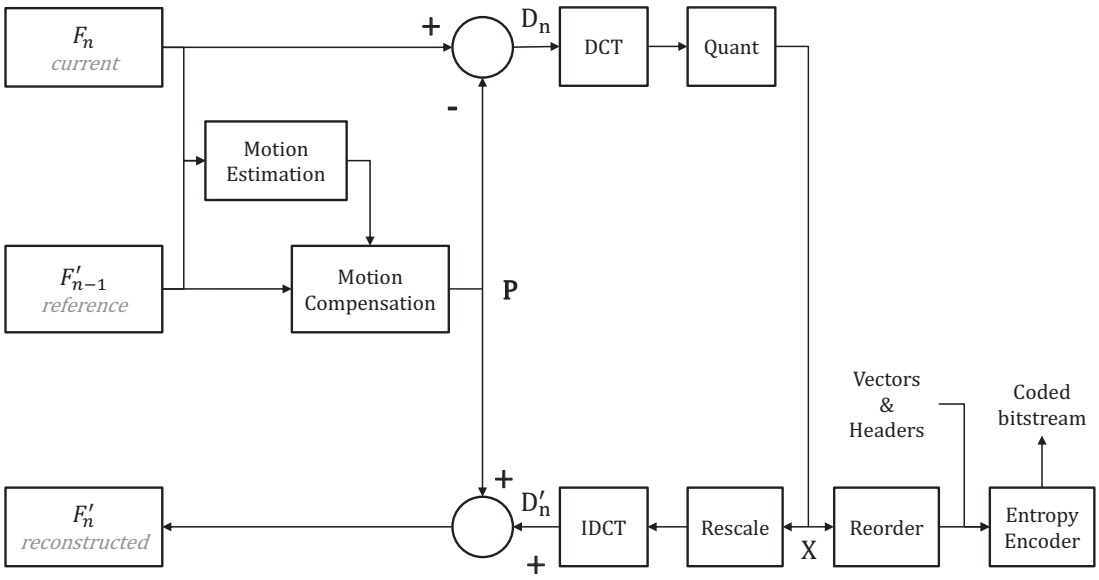


Figure 2.2: Encoder of the Hybrid DCT Codec [1].

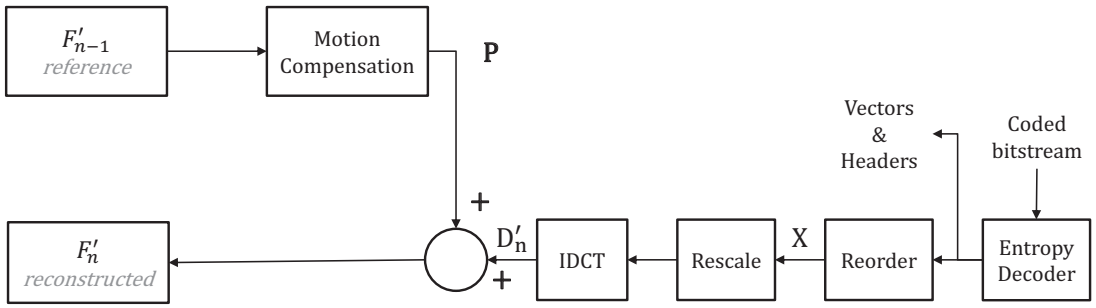


Figure 2.3: Decoder of the Hybrid DCT Codec [1].

frame, such as frame F_{n-1}. In actual scenario, more than one reference frames can be used. The macroblock candidates in the reference frame are usually within a range, whose center is at the same place as the current macroblock in F_n. The most matched candidate F'_{n-1} is selected among all candidates. The offset between F'_{n-1} and the current macroblock position is stored in a motion vector (MV).

3. Based on the MV, a motion compensated prediction P is rendered.

4. The difference D between the current macroblock and P is calculated by subtraction method. D is called the residual.

5. D is split into sub-blocks, which are then transformed using DCT separately.

6. Each transformed sub-block is quantized to generate the coefficients X.

7. X are reordered and run-level coded.

8. The compressed bitstream is generated by entropy encoding of X, MV and associated header information of each macroblock.

The reconstruction path is as follows:

1. The quantized macroblock X is rescaled and inverse transformed, which produces the decoded residual $D^{'}$. Some distortions have been introduced in $D^{'}$ because the quantization process is nonreversible and lossy.

2. The motion compensated prediction P is added to $D^{'}$, which produces the macroblock for the reconstructed frame $F_n^{'}$. $F_n^{'}$ would be used as the reference for the next frame F_{n+1}.

While for the hybrid decoder as illustrated in Figure 2.3, there are 5 steps from the right to the left:

1. The coefficients X, MV, and header information are extracted by entropy decoding.

2. X are recovered with an inverse process of run-level coding and reordering.

3. The decoded residual $D^{'}$ is obtained with rescaling and inverse transform.

4. A region located by the decoded MV within $F_{n-1}^{'}$ is used as the motion compensated prediction P.

5. P and $D^{'}$ are added together to generate the reconstructed macroblock, which is finally combined into the decoded frame $F_n^{'}$.

From the above descriptions, it is found that an identical reference frame should be used in both the encoder and the decoder.

The coded frames can be classified into 3 categories based on the reference frames used for prediction, which is shown in Figure 2.4.

- **I-frames:** Intra-coded frames take the frame itself as the reference, which is self-contained and independent of all other frames.

- **P-frames:** Predictively coded frames are coded based on a previously coded frame.

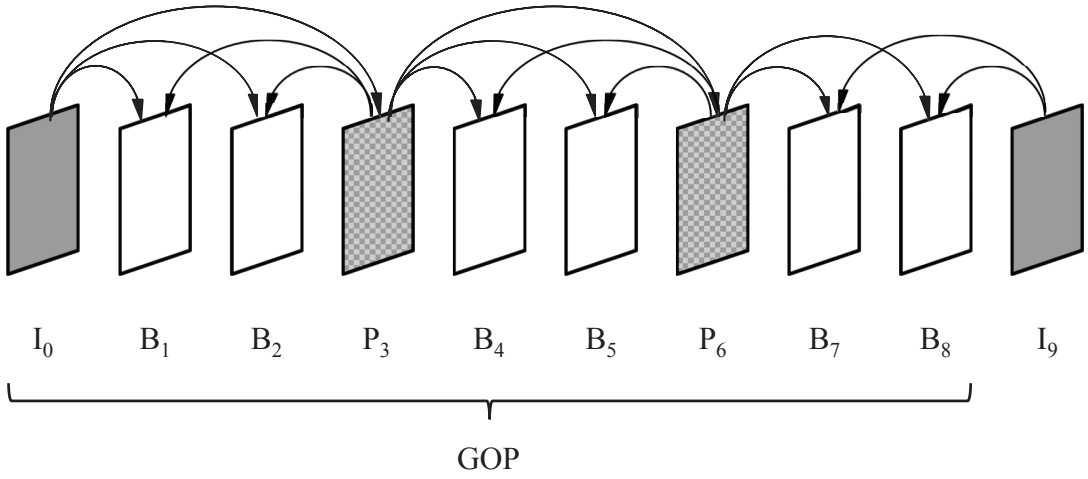


Figure 2.4: Illustration of GOP and different frame types.

- **B-frames:** Bi-directionally predicted frames are predicted from both previous and future coded frames.

The decoding process starts with an I-frame, which can be decoded independently. While P- and B-frames must be decoded when reference frames are ready. Thus, I-frame coding allows random access to video contents, such as fast forwarding. The frames between two I-frames, including the starting I-frame, forms a group of pictures (GOP). Each GOP can be decoded separately. The setting of GOP size depends on the application. With long GOP size, i.e. decreasing the frequency of I-frames, the bit rate can be reduced. By increasing the number of B-frames, latency would be increased.

Another important aspect of MPEG is the bit rate mode that is used. In most MPEG systems, it is possible to select the mode to use, CBR (Constant Bit Rate) or VBR (Variable Bit Rate). The optimal selection depends on the application and available network infrastructure. With limited available bandwidth, the preferred mode is normally CBR as this mode generates a constant and predefined bit rate. The disadvantage with CBR is that image quality will vary. While the quality will remain relatively high when there is no motion in a scene, it will significantly decrease with increased motion. With VBR, a predefined level of image quality can be maintained. This is often desirable for video surveillance applications where there is a need for constant quality, particularly if there is motion in a scene. Since the bitrate in VBR may vary, the network infrastructure (available bandwidth) for such a system requires a higher capacity even when an average target bit rate is defined.

H.264/MPEG-4 (AVC)

The main goal of H.264/AVC [5] is to enhance the compression efficiency with a "network-friendly" representation method, which can be applied in both interactive (video telephony/conference) and non-interactive (broadcasting/streaming/storage) applications. As for compression efficiency, up to 50% gain is achieved with similar bitrate comparing to previous standards. Besides, the standard is further designed with the concept of network abstraction layer (NAL) to customize the video in a network-friendly way. In addition, this standard provides enough flexibility to be applied to a wide variety of applications.

The main features that improve coding efficiency [18] are listed as follows:

- Variable block sizes are available for motion estimation and compensation, ranging from 16×16 down to 4×4.
- The accuracy of motion vector could be a fractional pixel.
- Motion vectors over picture boundaries.
- Multiple references are used for weighted prediction.
- In-the-loop deblocking filtering is enabled.
- Small block-size (4×4) is used for transformation.
- Enhanced entropy coding methods are available, including Context- Adaptive Variable-Length Coding (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC).

HEVC

The High Efficiency Video Coding (HEVC) standard [4] is developed to achieve various goals: (1) enhanced coding efficiency, with up to 50% gain over H.264; (2) simplified transport system integration; (3) resilience to data losses; (4) availability of parallel implementation.

These goals are achieved with the following multiple features:

- **Coding Tree Unit (CTU) and Coding Tree Block (CTB):** Compared with the macroblock in the previous standards, a more flexible tree-structured unit,

i.e. CTU, is introduced in HEVC that enables better compression efficiency. The CTU provides a variety of sizes to be selected by the encoder. A CTU includes a luma CTB and corresponding chroma CTBs, with syntax elements. Suppose the size of a luma CTB is $L \times L$. The value of L can be $16, 32$ or 64. The CTB can be further divided into smaller blocks following a tree structure.

- **Coding Units (CUs) and Coding Blocks (CBs):** A CU consists of one luma CB and two chromas CBs, as well as associated syntax. The CTU is split into CBs. Accordingly, the size of CBs is no larger than the luma CTB. A CTB may contain only one CU or may be divided into multiple CUs. Each CU is associated with trees of PUs and TUs.

- **Prediction Units (PUs) and Prediction Blocks (PBs):** The prediction mode (intra- or inter-prediction) is decided at CU level. The CU is the root of PUs. Based on the prediction mode, luma and chroma CBs can be further split. The size range of PB is from 64×64 to 4×4. Each PB is predicted from prediction blocks (PBs in reference frames).

- **Transform Units (TUs) and Transform Blocks (TBs):** The TUs also have their root at CU level. The size of TU is always equal to or smaller than the corresponding PU, which can be 4×4, 8×8, 16×16 or 32×32. For the smallest TU, i.e. 4×4, discrete sine transform (DST) is used as the transformation method.

- **Motion Vector Signaling:** Advanced motion vector prediction (AMVP), merge mode for MV coding, enhanced skipped and direct motion inference are provided.

- **Motion Compensation:** Quarter-sample precision MVs and 7/8-tap interpolation filters are used. Besides, the bi-predictive coding and weighted prediction are inherited from H.264.

- **Intra-picture Prediction:** 33 directional modes, plus planar (surface fitting) and DC (flat) prediction modes are supported. The selected intra-prediction modes are encoded by the most probable prediction directions derived from previously decoded neighboring PBs.

- **Quantization Control:** Uniform reconstruction quantization (URQ) is provided with supports for various block sizes.

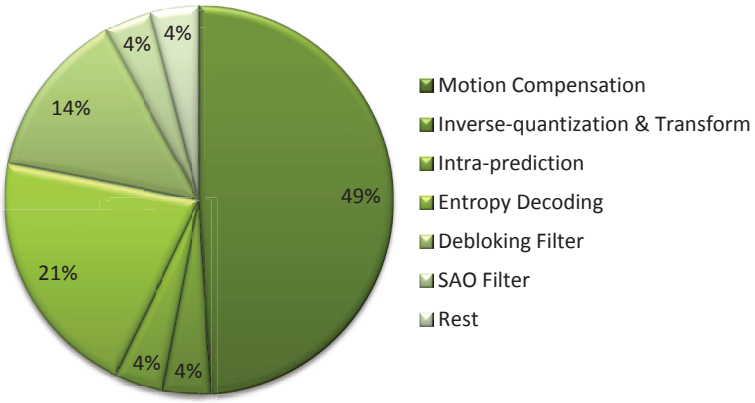


Figure 2.5: Average decoding time distribution of HM random access configuration on x86 [2].

- **Entropy Coding:** Enhanced Context-adaptive binary arithmetic coding (CABAC) is used as entropy coding method, with improved throughput speed and reduced memory requirement.
- **In-loop Deblocking Filtering:** A simplified decision-making and filter process design is introduced, which is parallel processing-friendly.
- **Sample Adaptive Offset (SAO):** In order to better reconstruct the signal amplitudes, a look-up table described with few parameters is introduced after the deblocking filter within inter-prediction.

The good performance of the HEVC codec comes with the cost of high computational complexity. The employment of HEVC encoder (HM) generally requires a large computer cluster. As shown in Figure 2.5, the complexity mainly comes from motion compensation, which accounts for half of the time on average [2]. At the encoder side, the motion estimation and compensation are also the major contributors to complexities. Thus, the complexity reduction in the inter-prediction part will be investigated in this thesis.

2.1.3 3D Video Representation and Compression

3D video provides depth perception of the scenery by delivering two different views for each eye. Then, the human brain would process these data to perceive depth. The simplest and most conventional representation method of 3D video is stereo video, which consists of the left and right view. Standard like the stereo high profile of H.264/AVC

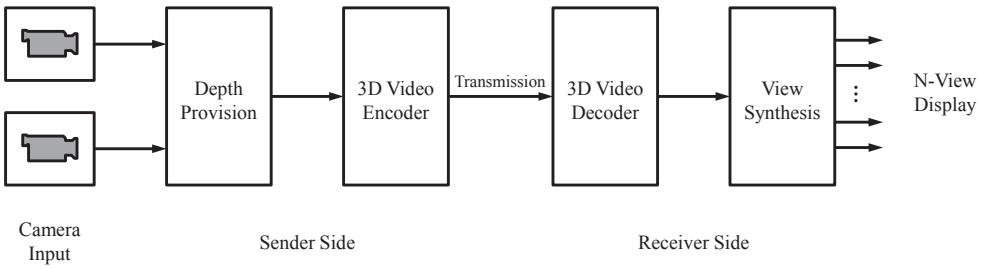


Figure 2.6: 3DV system with depth-enhanced multiview video.

is used to reduce both temporal and interview dependencies for efficient compression. However, 3D glasses are needed to watch this kind of 3D video. Meanwhile, only a fixed view angle is available.

In order to provide the flexibility of choosing views, multiview video representation is proposed with more than two views provided. The corresponding compression standards are developed, such as multiview coding (MVC) profile of H.264/AVC [19]. An MVC encoder generally consists of N parallel single view coders, with each of them reducing temporal redundancies. Besides, hierarchical B frames are inserted to exploit the statistical dependencies among spatially neighboring views [20]. The benefit of this design is that it provides back compatibility to single view decoders. However, the bit rate of MVC is linearly proportional to the number of views.

With the aim of restricting the bit rates within a reasonable range, a synthesis based representation method is developed. In this method, only a few views are encoded with MVC, with additional views synthesized from them. The plenoptic sampling problem is proposed, which investigates the allocation of cameras that allow error-free synthesis of views at arbitrary positions. However, the quality of the synthesized view is limited.

Thus, 3DV representation method [21], which contains both color and depth data, is proposed to enable high-quality view synthesis. The framework of such 3DV system is shown in Figure 2.6. The views are synthesized with both texture and depth image via Depth Image Based Rendering (DIBR) [22, 23] at the receiver side. With this method, a multiview video with a consistent quality level across different types of views is provided with limited bit rate. However, the challenge of this representation method is the mutual influence among depth provision, coding, and view synthesis. Thus, a modified representation method is proposed based on the 3DV representation method in this thesis.

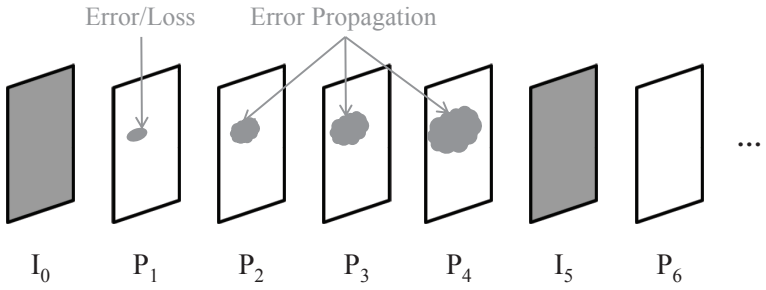


Figure 2.7: Example of error propagation.

2.2 Forward Error Correction

The compressed video data is vulnerable to data losses or errors over an unreliable network, which is owing to the dense dependencies within data. Different types of loss occur in different networks. For example, the wired packet networks, such as Internet, suffer from packet loss. While wireless channels are afflicted by bit errors and burst errors. Leaving the loss/error undisposed can result in a catastrophic effect on the quality of the reconstructed video. Suppose a loss or error occurs in frame 1 as shown in Figure 2.7, then the reconstructed frame 1 at the decoder differs from that at the encoder. As a result, the following frames that take frame 1 as prediction reference would suffer from incorrect/mismatched predictions, and this will cause what is called error propagation. The error keeps propagating until the next I-frame.

2.2.1 Error Control Methods

To tackled this problem, a lot of error control methods are proposed. These methods are roughly classified into four categories.

- **Retransmissions:** The sender retransmits the lost packets when receiving notifications from the receiver through a back-channel. It is a simple method that easily adapts to the network fluctuations. However, a back-channel is required, which is not practical for broadcast, multicast applications. Besides, the additional delay, that is roughly the round-trip-time (RTT), is incurred by retransmission.

- **Forward Error Correction (FEC):** A dedicated redundancy is added as the parity packets and sent along with the source packets, so as to recover loss/error. Compared with retransmission method, it does not require a back-channel and ensures a low delay. However, how to allocate the overhead in an effective and

efficient manner is a challenging task.

- **Error Concealment:** It tries to estimate the missing information by exploiting correlations within the data and to recover it by performing spatial/temporal interpolation/extrapolation. However, this method is ineffective when there is significant motion. Besides, error concealment of the single view video usually uses the previous frames, when a whole frame is lost.

- **Error Resilient Coding:** The goal of this method is to design the video compression algorithm, which generates bitstream resilient to specific types of error. One example for this category is Multiple Description Coding (MDC), which encodes the video data into several independent bitstreams.

The former two categories belong to channel coding based error control approaches, while the other two are source coding based error control methods.

2.2.2 Forward Error Correction

In this section, an introduction to forward error correction (FEC) is provided. The principle behind this method is to use error correcting parity packets to combat the errors/losses. The parity packets can help detect and correct the errors. As indicated by Shannon's channel coding theorem [24], a coding scheme, that guarantees data transmission over a given channel with small error probability, always exists when the data rate is less than the channel capacity. Based on this theorem, a number of FEC methods are developed [24–26].

The FEC codes can be divided into two major categories:

- **Block Codes:** This scheme divides the data into non-overlapping blocks and encodes each block independently. Thus, it is referred to memoryless code. It includes Hamming linear block error correcting codes, BCH (Bose-Chaudhuri-Hocquenghem) cyclic block codes and Reed-Solomon cyclic block codes, etc.

- **Convolutional Codes:** It is a code with memory, where the coding of a block is the function of its previous blocks. Methods include Viterbi algorithm [25], Turbo Convolutional Code (TCC) and Low Density Parity Check (LDPC) Code.

In addition, FEC is usually used based on Unequal Loss Protection (ULP) tools,

where FEC codes are assigned unequally to source packets according to their importance.

2.3 Video Streaming

In this section, research on video streaming will be discussed. A brief history of evolutions in this field is firstly reviewed. Then, dynamic adaptive video streaming technologies that are most popular nowadays are discussed in more detail. Finally, Quality of experience (QoE), which is the user-centric method for evaluating the overall performance of video streaming service, is introduced.

2.3.1 Overview

In this part, basic concepts and key challenges of video streaming are firstly introduced. Then, the approaches developed for video streaming are classified and reviewed.

Video streaming refers to the transmission of stored video. It has two modes, namely download mode and streaming mode [27]. The download mode is similar to a file download, where the video is displayed only after the entire file is downloaded. This method can take advantage of the established delivery protocols, such as TCP, FTP, etc. However, it requires long download duration and large storage spaces. Moreover, it would cause huge waste if the user decides not to watch the whole video after taking a glance at the first few seconds. The streaming mode, i.e. video streaming, tries to mitigate the problems of the download mode. It splits the video into parts and transmits them separately in succession. The received part can be decoded without downloading the whole video, which enables simultaneous playback and delivery of the video. The start-up delay, which refers to the delay between the beginning of transmission and the start of playback, is usually within $5 - 15$ seconds. Meanwhile, the required storage is much lesser than the download mode. Moreover, it is flexible to stop delivering if the user decides to quit in the midway.

The video streaming problem can be formulated as a series of constraints, mainly oriented in time aspect. Firstly, the compressed video is partitioned into packets. Then, the packets are sent in chronological order. Once the client received the packet, it is decoded and put in the buffer. After some time (start-up delay), the playback starts while the transmission of following packets continues. Suppose Frame n is displayed at time t_n. The time interval between consecutive frames is denoted as Δ. Thus,

Frame $n+1$ must be received and decoded by time $t_n + \Delta$. Accordingly, Frame $n+l$ ($l > 1$) needs to be delivered and decoded by time $t_n + l \times \Delta$. The time corresponds to the playback time, as well as transmit, decode and display deadlines. Any packet that arrives later than this deadline cannot be displayed. Thus, one important goal of video streaming system is transmitting the packets within the time constraint. However, there are many obstacles in achieving the goal, which will be introduced in the following section [28].

Key Challenges

There are mainly three basic problems afflicting time-constraint video streaming over the Internet. Since the Internet only provides best effort service, the bandwidth, delay jitter and loss rate are usually unknown and dynamic. These three problems are challenging to a video streaming system, because any one of them may cause failure in transmitting the video packets in time. In the following part, the three problems are analyzed respectively.

- **Bandwidth:** The challenge related to bandwidth mainly arises from two characteristics of the bandwidth, namely time-varying and unknown. While for a video streaming application, a minimum bandwidth is required to obtain an acceptable perceptual quality. Thus, it is challenging for the streaming system to meet this requirement given an unknown bandwidth. Besides, the time-varying characteristic of bandwidth makes the accurate estimation difficult. If the video packets are sent at a speed faster than the bandwidth, congestion and packet loss would occur. As a result, the video's quality would drop. In some cases, the display would even be paused. On the contrary, if the transmitted bit rate is lower than the bandwidth, the sub-optimal quality is provided, which is a waste of resources. Thus, the accurate estimation of bandwidth, as well as congestion control, are needed in the video streaming system. The commonly used congestion control method for video streaming is rate control, which adapts the delivery bit rate to the network bandwidth. The rate control schemes can be classified into three categories: source-based, receiver-based and hybrid rate control. For the source-based scheme, the sender regulates the sending bit rate based on the feedback information about the network [29, 30]. While for receiver-based scheme, the user side is responsible for the rate control and adapt the packet requirement

accordingly [31]. Two major methods are used for both schemes, namely probe-based and model-based approaches. When it comes to the hybrid scheme, both sender and receiver are involved in the rate control process [32, 33].

- **Delay Jitter:** Since the bandwidth is time-varying, the end-to-end delay of delivering each packet fluctuates accordingly. This variation of the end-to-end delay is defined as the delay jitter. A streaming system would require a bounded end-to-end delay to achieve the real-time playback. If the packet arrives later than the playout time (delay bound), the packet is regarded as lost and the playback is paused. As a solution, a buffer is introduced at the client side to alleviate this problem. By adding a buffer to store the pre-downloaded packets, the delay for downloading the current packet can be compensated with the already reserved buffers [34]. The proper buffer level can be maintained with a rate control method.

- **Loss Rate:** There are different types of losses, like packet loss for wired packet networks and bit/burst errors for wireless channels. These losses have a destructive influence on the final video quality. While for the video streaming system, a video stream robust to packet losses is desired. The solutions include Multiple description coding [35] and error control methods. The former one is a source coding solution, where any subset of the packets can be decoded and obtain a video clip with corresponding quality level. In this case, the loss of packets is not destructive and only influences the quality as long as one packet is received successfully. While for error control, FEC is a typical method which has been introduced in the previous section.

When it comes to video streaming over both wired and wireless networks, additional challenges appear. Firstly, the delivery time increases with the wireless network. Especially when there is retransmission requirement, the delay would be very significant. Secondly, it is difficult to estimate the network conditions which is influenced by both channels. Thus, it is hard to perform proper rate control mechanisms.

Brief History of Streaming Methods

Faced with the above challenges, the video streaming methods evolve with the development of network protocols. The network layer protocol, Internet Protocol (IP), provides the baseline delivery for all hosts in the network. It is responsible for addressing, best-

effort routing and defining the universal format. On top of IP, Transmission Control Protocol (TCP) [36] and User Datagram Protocol (UDP) [37] are two most important transport level protocols. TCP provides reliable Byte stream transmission with retransmission and acknowledgment mechanisms. Flow control and congestion control are also supported by TCP. While UDP only offers best-effort transmission for packets without any flow/congestion control. The benefit is providing flexibility in determining proper control mechanisms. Moreover, UDP does not require a back channel, which is a necessity for TCP. To sum up, the control messages are usually transmitted with reliable TCP/IP protocols, while media data is transmitted via UDP/IP protocols which offer predictable delay. On top of transport protocols, there are media delivery protocols and media control protocols, which are designed specifically for media streaming and session control. The media delivery protocols, such as Real-time Transport Protocol (RTP) [38], enable the detection of lost packets. While the media control protocols, like Real-time Control Protocol (RTCP), provide feedback on the quality of delivery, which assists the operation adaptation of the sender.

The video streaming standards define the protocols used for media transport and session control. The evolution of the streaming standards can be summarized into the following three stages:

- **Datagram Streaming:** The packets are carried by UDP in datagram streaming. On top of UDP, there are both non-open and open protocols developed. The non-open protocols includes Microsoft Media Server (MMS) protocol [39], Adobe's Real Time Messaging Protocol (RTMP) [40] and so on. While the open protocols are more widely employed, such as RTP. The major problems with datagram streaming are as follows: Firstly, the implementation is complicated, since UDP only provides packet level control. Thus, things like flow/congestion control and packet loss need to be considered in the implementation. Secondly, UDP suffers from problems caused by firewalls and network address translation (NAT) routers. Finally, higher cost of infrastructure is needed, where specialized solutions for caching and load balance are required by content delivery networks (CDNs).

- **Progressive download Streaming:** As the next generation of datagram streaming, progressive download streaming overcomes the aforementioned draw-

backs. It is widely used for its simplicity and straightforward implementation. The client downloads the video files over HTTP and displays them at the same time. Besides simplicity, other benefits of progressive download streaming are as follows: Firstly, it is firewall friendly thanks to the HTTP protocol. Secondly, it is supported by all CDNs, making utilization of transparent web caching easy. However, there are also drawbacks. Firstly, interruptions are more likely to happen in progress download streaming, comparing to datagram streaming. Thus, a larger buffer is required to alleviate the problem of interruption. Besides, multicast is not supported by progressive download streaming.

- **Adaptive Streaming over HTTP:** This is the most popular streaming scheme nowadays. It is also based on the HTTP protocol, which is firewall friendly. Different from progressive download streaming, the video stream is split in time domain into a sequence of segments, i.e. video chunk. Each segment is provided with multiple quality levels to be chosen from. Thus, the client can request a segment of the proper quality level according to the bandwidth. The quality adaptation granularity is same as one segment duration. The adaptive streaming over HTTP not only inherits all the advantages of progressive download streaming but also provides the adaptivity to the fluctuating bandwidth. There are already many adaptive HTTP streaming formats available, such as Apple's HTTP Live Streaming (HLS), Microsoft's Smooth Streaming and MPEG Dynamic Adaptive Video Streaming over HTTP (DASH).

In the following section, a detailed introduction to DASH is provided, which offers the foundation of one proposed work in this thesis.

2.3.2 Dynamic Adaptive Video Streaming over HTTP

As mentioned above, there are many different implementations of HTTP streaming. Although they share similar principle, each implementation has different segment format and manifest. Thus, a device needs different protocols to receive videos from different HTTP servers. In order to unify different formats and provide interoperability between servers and clients of different vendors, MPEG issued a Call for Proposal for an HTTP streaming standard in 2009. The resulting standard, known as MPEG-DASH over HTTP, allows any standard-based client to stream video from any standard-based

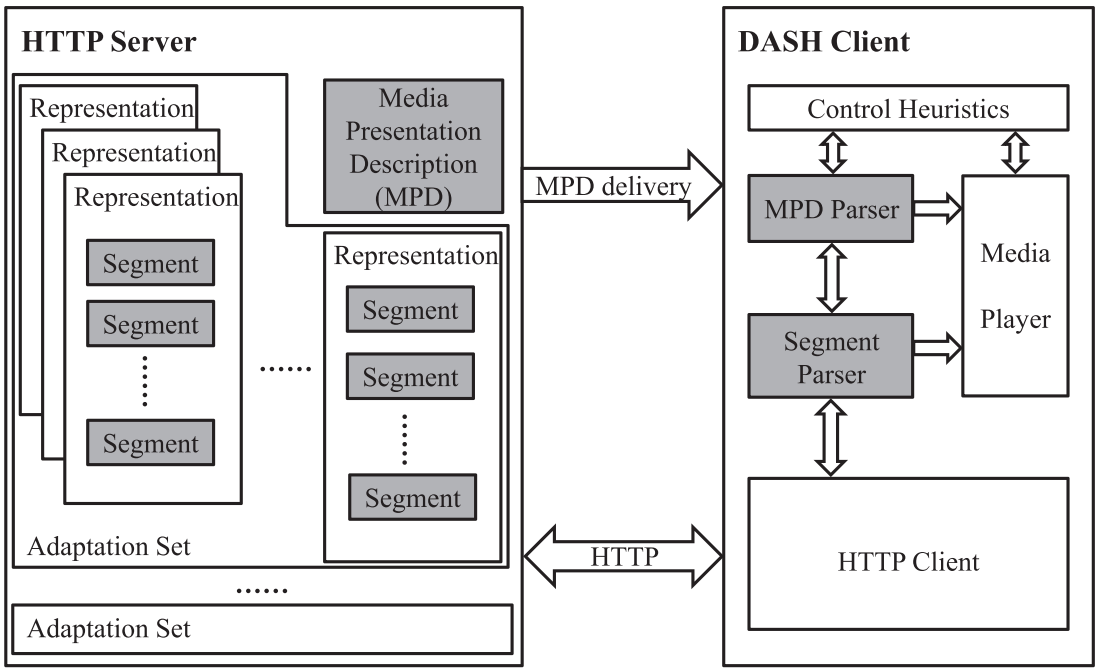


Figure 2.8: Scope of the MPEG-DASH standard. The shadowed blocks are defined in the standard, while others are open for development.

server. The MPEG-DASH standard only defines the segment formats and the manifest file. The delivery process and the adaptation heuristics are outside the scope of the standard.

As shown in Fig. 2.8, a typical DASH system consists of a HTTP server and a DASH client [41]. They communicate with each other through the HTTP network.

In the HTTP server, video contents of different versions and their description files are stored. Different versions share the same video content but are encoded using different settings, like resolution, frame rate, QP and so on. These different versions are called representations in DASH and they provide multiple choices for adaptation. All the representations constitute an adaptation set. While audio and subtitles constitute other adaptation sets.

For each video representation, it is divided in time domain into several chunks. The chunks are named as segments in the DASH standard. Each segment usually lasts for $2-10$ seconds long [42–44]. The adaptation granularity is the same as the duration of a segment since the quality can only be switched on the boundaries between segments. The shorter segment duration enables better matching to bandwidth, while longer duration provides smoother quality switching [45]. However, more starvation,

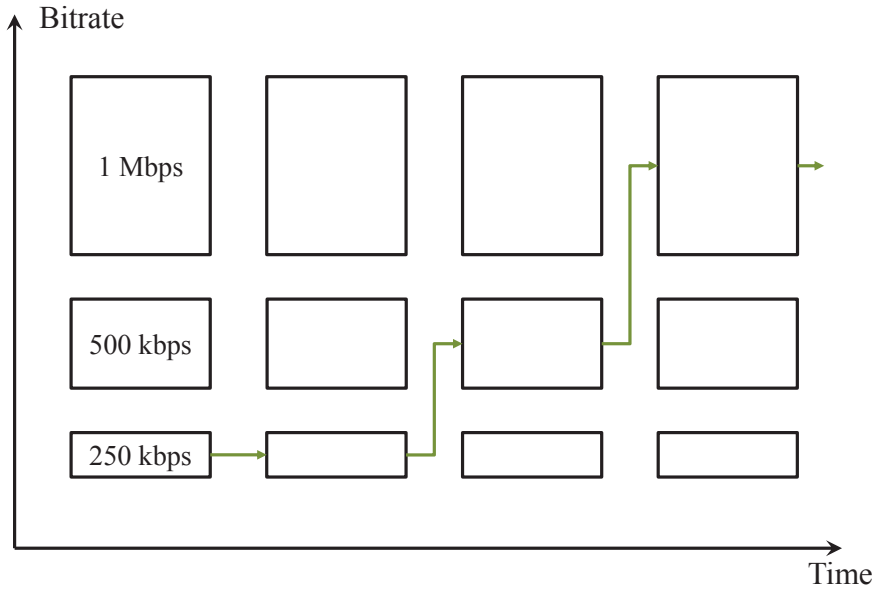


Figure 2.9: The illustration of an adaptive stream. There are 3 quality levels for adaptation, i.e. 250, 500 and 1000 kbps. Each box represents a segment. The arrow connecting boxes represents one possible video playout.

which causes stalls in the display, reveals more frequently with longer segment durations. This is owing to the long adaptation granularity can not follow rapid bandwidth changes. For live streaming applications, where there is less buffered data, shorter segments are preferred to ensure a finer switching granularity. Each segment is encoded independently and stored as an independent file with an associated URL address. By removing the dependencies among segments, seamless quality switching is guaranteed.

There is a description file, where the URL addresses of all available segments and other characteristics, like bandwidth, resolution, media types and program timing are recorded. It describes a hierarchical manifest of the available content and its various versions. This description file is called Media Presentation Description (MPD) in DASH, which is stored as an XML document.

As for the DASH client, it will first obtain the MPD file. After parsing the MPD file, the client decides which segment to request based on the parsed information and network condition. The client sends HTTP GET request to fetch the segment. After accumulating enough buffer reservation, the client starts to play. Meanwhile, following segments will be obtained based on the MPD file, as well as monitored network bandwidth trend to avoid buffer underflows. One example of requested segments is shown in Fig. 2.9. For the first two segments, they are requested at the lowest quality. While for

the third and fourth segments, they require one quality higher than their previous one. The intelligence behind the decisions lies in the control module, which usually tries to provide better video quality while maintaining adequate buffer reservation for continuous playout. This is not defined in the DASH standard, and it is open for research, which is also investigated in this thesis.

2.3.3 Quality of Experience

In order to evaluate the performance of video streaming services and ensure users' satisfaction, there are many tools and techniques developed. Traditionally, Quality of Service (QoS) method [46] is widely used to access the performance of online services. However, it is a network-centric method, which measures the ability of the network to satisfy the requirement of the services. The factors considered in QoS includes bandwidth, throughput, delay, packet loss and so on. Thus, it is more suitable to evaluate the reliability and performance of the network. When it comes to the quality perceived by users, it is subjective and influenced by many factors on top of the network. These factors include service infrastructure, client hardware, user's psychological expectations, etc. Thus, Quality of experience (QoE) [47, 48] is proposed as a user-centric method for measuring the overall acceptability of the service. QoE can be regarded as an extension of QoS, which considers more factors that are directly perceived by users, other than the network aspect. However, the user-centric factors are nontrivial to measure or predict. Many related works are summarized in the following.

For the traditional video broadcast services, QoE methods mainly compare the received contents at the client with reference. The reference refers to the original video contents at the sender side. Based on the amount of information about reference available at the client side, QoE metrics can be classified into 3 categories.

- **Full reference (FR) methods:** The complete information related to reference is available at the client side, thus the received video can be directly compared to the original contents. Common FR methods include Peak Signal to Noise Ratio (PSNR) [49], Structural Similarity (SSIM) [50], etc.
- **No reference (NR) methods:** The reference is not available at the client side, where only the received contents are used for measurements. This category is applicable to online services over shared networks.

- **Reduced reference (RR) methods:** Only several parameters of the reference are available at the client, such as bit rate, frame rate.

FR and RR methods require the feedback channel, thus they are not suitable for online streaming services. Hence, NR methods are usually used in online video services.

As for the NR methods, it can be classified into two categories, depending on the type of measurement it uses. One is subjective methods, and another is objective methods. The subjective methods rely on the user feedbacks about the experience. A certain number of human subjects are shown the video service in a controlled environment. After the display, the linear scale ratings are asked for the display. The limitation of subjective methods is not only time consuming, but also subjective to user bias. The bias mainly comes from: (1) user-based interest or purpose; (2) video-dependent characteristics, like the genre, popularity; (3) device-related aspects, like the capability of the device. In order to eliminate the biases, robust sampling methods and statistical analysis tools, as well as a sufficient number of human subjects, are necessary. The most popular subjective metric is Mean Opinion Score (MOS). Five scales for rating are provided: bad, poor, fair, good and excellent.

When it comes to the objective QoE methods, automated measurement techniques are used. These techniques include:

- **Startup Delay:** The startup delay refers to the time between the moment user clicks the video link and the time video starts to play. This delay is used to buffer the initial portion of the video, as well as downloading related data. If the delay is too long (more than 2 seconds), the user might quit the video completely [51]. If the delay is too short, not enough initial buffer is prepared. In this case, there would be a higher probability of starvation, which has a huge effect on the QoE.

- **Number of Starvation:** The starvation happens when the buffer becomes empty and the following contents can not be displayed in time. The download speed can be one reason causing the starvation. Meanwhile, the user operation of fast skipping or rapid quality changing can also lead to starvation. The starvation usually has a severe influence on user experience [52].

- **Duration of Starvation:** When starvation happens, the playback would only be resumed when the desired buffer level is accumulated. The time for accumu-

lating the desired buffer refers to the duration of starvation. It is shown in [52] that 1 second is acceptable, while 3 seconds would have a bad influence on user experience. It is also shown in [53] that, viewers prefer a single long interruption than several short stalls.

- **Video Quality:** Video quality is measured based on the bit rate, which denotes the average data required to play one-second of video. Higher quality corresponds to the higher bit rate, which would also need longer time to download given a fixed throughput.

- **Frequency of Quality Switching:** This factor is used in the DASH system. Frequent switching would be annoying to users according to [54]. The switch to a higher bit rate refers to positive switch and negative switch vice versa. Users are usually more critical to negative switches as shown in [55].

- **User Engagement:** It is measured by the number of views and the display time of the video, which reveals the user involvement. The user that is satisfied with the video content and the QoE of the service would usually lead to a good engagement.

The objective QoE methods are mainly influenced by the above-mentioned factors. There are different measurement methods for each of the factors. Besides, there are different ways to combine them in a general QoE evaluation scheme, depending on the application scenario.

Chapter 3

Statistical Approach for Motion Estimation Skipping (SAMEK)

3.1 Introduction

Motivated by the demand of high-resolution videos by the consumer market, High Efficiency Video Coding standard is established with doubled coding efficiency compared to the previous standard H.264/AVC. The improvement mainly comes from its flexible partitioning types and multiple choices of coding options. However, these factors provide larger space for exhaustive search than the former standard, which leads to increased computational complexity. As a result, the state-of-the-art standard is faced with obstacles once it comes to real-time applications, as well as devices with low processing power or limited energy supply. In order to solve this inevitable problem, many works have been done to reduce the complexity at both encoder and decoder side.

For the works at the encoder side, they are mainly targeting the reduction of motion estimation (ME) process. As ME is one of the most time-consuming units (up to 96% of the total encoding time) among all the processing units in the HM encoder [56]. These works can be classified into several categories, including fast coding unit (CU) depth decision, fast prediction unit (PU) mode decision and fast motion estimation methods. As one of the most effective methods, fast CU depth decision tries to skip specific depth level or terminate CU split based on the learned information. In [57], unnecessary CU depth levels are skipped by exploiting the historical decisions of temporal and spatial neighbors. Besides, early termination method is also proposed based on motion homogeneity checking, RD cost checking and SKIP mode checking. In [58], Bayesian decision rule is utilized to avoid exhaustive rate distortion optimization (RDO) search on all possible CU sizes and modes. When it comes to fast PU mode decision methods,

both early mode decision and motion vector merge are investigated. Based on the differential motion vector and coded block flag (CBF), the work in [59] proposes to early detect the SKIP mode. In [56, 60], early motion vector merging is exploited based on heuristics approach, which will help the inter prediction mode decision. As for fast motion estimation method, the work in [61] proposes the directional search for the integer ME. It extends the the initial motion vector into three search paths and considers highly probable horizontal and vertical movements.

In this thesis, a statistical approach for motion estimation skipping (SAMEK) is proposed to avoid some unnecessary motion estimations in units with less probability of being referenced. For example, HEVC executes intra and inter encoding for each frame based on their positions in the sequence, according to the GOP size and a fixed prediction structure. Suppose n previous frames are used for the inter prediction of frame i, and there is a scene change between frame $i-2$ and frame $i-1$. This means the video contents from frame $i-n$ to $i-2$ are different from those from frame $i-1$ and i. However, frame from $i-n$ to $i-2$ would still be used as a reference for frame i following the fixed prediction structure, which is a waste of executing time. Thus, this waste should be avoided by skipping these unrelated reference frames for frame i and following frames. As the scene change is already known when encoding frame $i-1$, this information can guide the skipping of unrelated reference frames for the encoding of future frames. Motivated by this example, two rules, namely ZeroCase and DecreaseCase, are developed to recognize these unnecessary reference units. These two rules are derived based on the relationship analysis between each PU and its references. As a result, the PU decision complexity can be reduced, which leads to the reduction of overall encoding time. What is more, the rules are simple counting and comparison operations, which can be easily incorporated into the HM encoder and causes negligible extra processing time. Since the decision to skip ME is based on the motion vector information of the previous frames, thus no delay will be caused. This method was implemented on HM 16.0 test model to provide the rate-distortion results, as well as time-saving ratios. The obtained results show that the proposed method achieves an average 6.87% time saving with little loss in terms of quality (0.006 dB on average) or minor increase in bitrate (0.228% in the worst case). To the best of our knowledge, this is the first work on shrinking the search range in the ME part by analyzing the relationships between the current frame and its references. In addition, the proposed

method is trying to reduce the complexity using orthogonal methods with aforementioned related works [56–61]. Thus, it can be flexibly integrated with aforementioned related works to further enhance the results.

The remainder of this chapter is organized as follows: Section 3.2 gives an idea of motion estimation process in HEVC; Then, section 3.3 describes the observed rules from the ME results between each frame and its references; Based on this knowledge, section 3.4 explains the proposed method, which is the main contribution of this work; Experiment settings and results are presented in section 3.5; Finally in section 3.6, conclusions, and future works are discussed.

3.2 Preliminary Knowledge on Motion Estimation in HEVC

Like H.264/AVC encoders, inter prediction is an important step in HEVC encoder to remove temporal redundancies between frames [62]. In inter prediction of HEVC encoder, motion merging (merge mode) or normal inter prediction (inter mode) is chosen to obtain the luminance motion parameters associated with each PU. The parameters consist of motion vectors (MVs) and corresponding reference frame indices (idxs). As for Chrominance MVs, they are derived from corresponding luminance ones. The merge mode infers motion parameters for the current PU from spatially and temporally inter encoded neighbors, while inter mode obtains motion parameters through motion estimation.

Motion estimation (ME) includes integer ME (IME) and fractional ME (FME) stages [63]. For every PU, IME is firstly performed to search for the best integer-pixel accurate candidates from the decoded picture buffer (DPB), which contains the reconstructed frames that are previously encoded [64]. Then, advanced MV prediction (AMVP) is used to generate best MV predictor (MVP) from spatially and temporally MVP candidates, so as to enhance the result of IME. After that, the selected integer-pixel candidates are fed into FME to refine them into 1/4-pixel accurate luminance candidates and 1/8-pixel accurate chrominance candidates. These fractional candidates are interpolated by corresponding filters. Finally, these motion parameters are delivered to motion compensation (MC) stage for further processing.

From the flow path described above, it can be found that ME is a complex step. Besides, the complexity is accumulated by multiple callings of ME, which is inevitable for the recursive searching mechanism of HEVC. Although fast algorithms like TZ search

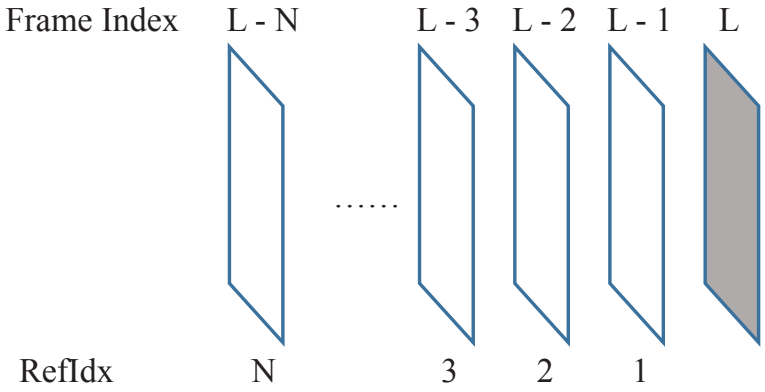


Figure 3.1: An illustration of frame index and reference index (RefIdx).

are incorporated in the HEVC, the complexity is still huge. In order to further alleviate the burden of searching among such plenty of choices, certain reference candidates can be skipped according to the rules proposed in this thesis.

3.3 Relationship Analysis between encoding PU and its references

In this work, low-delay P (LP) coding configuration is used. For each P frame, N immediate previous frames are used as references, as shown in Fig.3.1. The motion estimation process will generate better performance with the increase of N within certain ranges. However, more time will be consumed at the same time. In order to save time without sacrificing much performance, some units, which are less referenced by the current encoding unit, can be skipped in ME as they have fewer contributions. These less referenced units are defined as having a small or even zero percentage of pixels used as a reference for encoding the following frames. This mechanism is inspired by the common sense that consecutive frames are usually highly correlated, while timely farther away frames are less correlated. Thus, some farther away units can be skipped in the ME. This phenomenon can also be discovered in the Fig.3.2, where the curve of the nearest reference frame (i.e. RefIdx = 1) is almost always at the top (i.e. the highest reference probability).

The challenges behind this work are two-folds. The first challenge could be summarized by this question: when will the previously described common sense phenomenon take place? This is because different sequences have different characteristics, and even different parts of one sequence may exhibit different correlation trends. The second

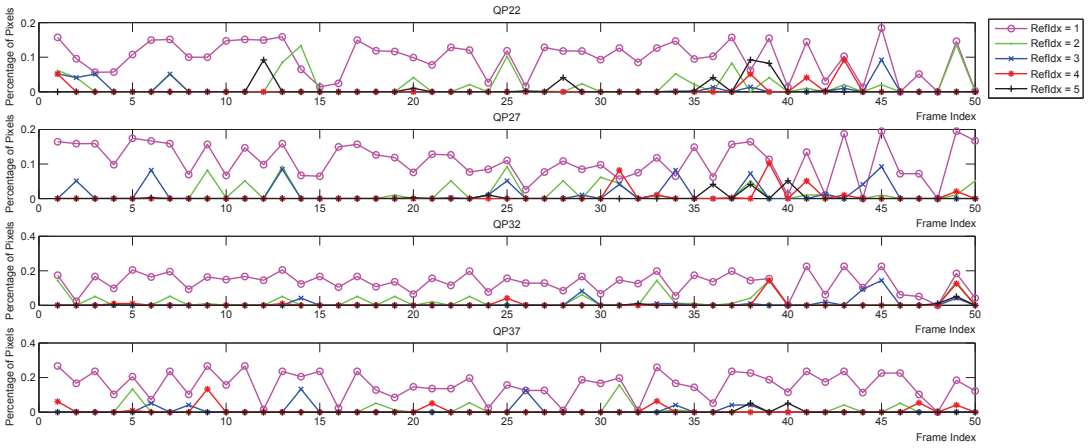


Figure 3.2: The percentage of pixels used as reference for encoding following frames versus Frame index for the first 50 frames of *BasketballPass*; The frame index denotes the POC of the frame used as reference; RefIdx denotes the position in the RPS for this frame; The data is derived over one-quarter of the sequence; QPs are 22, 27, 32, 37; Totally 5 immediate previous frames are used as reference for motion estimation.

challenge lies in how to define the size of the unit? The unit can be either one whole frame or sub-part of one frame because the proper unit size is related to the main characteristics of the sequence. The second challenge will be explained later in Section 3.4. As for the first challenge, two rules, which could be observed in Fig.3.2, are proposed. For simplicity, unit represents a whole frame here. Let $P(x, y)$ denotes the percentage of pixels in frame x used as reference when encoding frame y, then the RefIdx of frame x is $(y - x)$. The two rules are as follows:

- **ZeroCase**: If $P(L, L+1) = 0$, then $P(L, M) < \theta$ when $M > L+1$;
- **DecreaseCase**: If $P(L-1, L) > P(L, L+1)$, then $P(L, M) < \theta$ when $M > L+1$;

where $\theta = 0.1 \times \sum_{x=L-10}^{L-1} P(x, x+1)$. In the ZeroCase, frame L is not used as a reference for encoding frame $L+1$ even when it is at the first position in the DPB. This means frame L is not correlated to frame $L+1$, which means a scene change might have taken place between these two frames. Then, for the encoding of the following frames (later than frame $L+1$), frame L will not be used as a reference with high probability. Thus, the motion estimation can skip the search in frame L directly for these following frames. As for the DecreaseCase, $P(L-1, L)$ denotes the relationship between frame $L-1$ and frame L. Likewise, $P(L, L+1)$ represents the relationship between frame L and frame $L+1$. When $P(L-1, L) > P(L, L+1)$, it means frame L is more correlated with frame $L-1$ than with frame $L+1$. Then, for the encoding of the frames later

than frame $L+1$ (which are supposed to be correlated to frame $L+1$), frame L will not be used as a reference with high possibility. Thus, the skip of motion search can also be applied here. Examples like frame 46, 48 for the ZeroCase, and frame 8, 24 and 26 for the DecreaseCase with QP equals to 22 can be found in Fig.3.2. As for θ, it is set as 10% of the average $P(L, L+1)$ of ten previous frames. Experiments show that it could guarantee less than 0.1 fluctuations in the overall performance, including BD-rate and BD-PSNR, when references with $P(L, M) < \theta$ are skipped in the ME step.

3.4 Proposed Method

According to the proposed ZeroCase and DecreaseCase rules, a mechanism to skip the ME step in less referenced units is incorporated into the HM encoder. For instance, when **one unit of processing is defined as one frame**, the processing steps are as follows. Firstly, the percentage of pixels used as the reference (n) of frame N will be noted down after encoding frame $N+1$. If n equals to zero, the ZeroCase rule will be effective. That is, frame N will be skipped in motion estimation part for the encoding of frames later than frame $N+1$. Secondly, the percentage of pixels used as reference (m) of frame $N+1$ will be noted down after encoding frame $N+2$. If n is larger than m, the DecreaseCase rule will take place. That is, frame $N+1$ will be free from motion estimation step for the encoding of frames later than frame $N+2$.

Two kinds of skip unit are used in the experiment. One is full frame and another is one-quarter of a frame. For the latter one, each frame is divided into 2×2 quarters for separate analysis. This division strategy is consistent with the rule of thirds in the visual arts [65]. As proposed by this rule, an image is imagined as divided into nine equal parts by two equally spaced horizontal lines and two equally spaced vertical lines. Accordingly, the important compositional elements should be placed along these lines or their four intersections. Thus, the 2×2 division can separate the four intersections into different parts, which would be helpful to treat their different motion characteristics separately. This is also proved by the experiment that, the 2×2 division strategy generally gets more cases that follow the proposed rules than 1×1 division strategy (i.e. full frames). This phenomenon can be discovered in Table 3.1, where the correctness denotes the positive true cases among all positive cases. It is obvious that the correctness of 2×2 division strategy is higher than that of 1×1 division strategy. Besides, the 2×2 division strategy is less computational complex than other finer di-

Table 3.1: Correctness evaluation for ZeroCase and DecreaseCase; Performance evaluation of SAMEK method relative to HM encoder with TZ search enabled.

Sequence (Resolution, FPS)	Division Strategy	Quarter Index	Correctness(%)		Performance		
			ZeroCase	DecreaseCase	BD-rate (%)	BD-PSNR (dB)	DT (%)
NebutaFestival (2560 × 1600, 30)	2 × 2	1	33.33	21.3	0.127	**0.012**	5.85
		2	33.33	17.1			
		3	33.33	24.14			
		4	33.33	17.57			
	1 × 1	-	33.33	21.8	0.082	0.004	5.12
Cactus (1920 × 1080, 50)	2 × 2	1	31.25	65.01	0.164	0.005	6.84
		2	68.75	67.95			
		3	25	24.2			
		4	31.25	34.89			
	1 × 1	-	25	39.2	0.111	0.005	6.68
KristenAndSara (1280 × 720, 60)	2 × 2	1	33.33	70.72	0.123	0.0003	4.46
		2	41.67	62.79			
		3	33.33	33.59			
		4	66.67	54.06			
	1 × 1	-	33.33	58.46	0.309	0.01	4.74
BasketballDrill (832 × 480, 50)	2 × 2	1	68.75	38.68	**0.228**	0.01	**9.5**
		2	87.5	57.94			
		3	81.25	54.78			
		4	93.75	65.06			
	1 × 1	-	93.75	56.24	0.487	0.008	10.24
BasketballPass (416 × 240, 50)	2 × 2	1	96.43	74.83	0.008	0.001	7.72
		2	77.5	78.89			
		3	76.61	71.46			
		4	96.43	73.76			
	1 × 1	-	96.43	68.8	0.168	0.003	8.33
Average	2 × 2	-	57.14	50.44	0.13	**0.006**	**6.87**
	1 × 1	-	54.58	48.9	0.23	0.006	7.02

vision strategies. Thus, the 2 × 2 division strategy is used in the proposed SAMEK method.

3.5 Experimental Results

In the experiment, the first 150 frames of each selected sequence are used. The sequences are chosen from different classes, namely NebutaFestival, BasketballPass, BasketballDrill, Cactus and KristenAndSara. These videos are chosen from the recommended test sequences by the HEVC standard. There are totally 5 categories of sequences in the recommended data set. The 5 chosen example sequences in our experiment belong to each of these categories, covering different resolutions, contents and other aspects. Thus, in general, the chosen videos have a good generalization

characteristic and can represent the performance in an average video. The Low Delay configuration is used and five immediate previous frames in order are used as reference candidates for every P frame. The proposed algorithm is implemented on HM 16.0 encoder and simulated on a server with Intel Xeon processor of speed 2.6 GHz and 32 GB DDR3 RAM. The rate-distortion performance, as well as complexity reduction in terms of encoding time, are shown in Table 3.1. The BD-rate and BD-PSNR are calculated according to [66]. DT is calculated as follows:

$$DT = \frac{T_{proposed} - T_{HM}}{T_{HM}} \times 100\% \tag{3.1}$$

The results show that the proposed SAMEK method accelerate the encoding time by 6.87% on average and up to 9.5% when compared with HM encoder with TZ search enabled. And the improvement will be even higher when compared with HM encoder with full search. Besides, the PSNR degradations and bitrate increases are negligible, with 0.012 dB for BD-PSNR or 0.228% for BD-rate at most. It can be found that sequences with larger motions (namely, BasketballPass, BasketballDrill and Cactus) have larger time saving ratio. This is reasonable as these sequences consume more time for the ME step. Thus, the time reduction in the ME step will largely affect the overall encoding time. Furthermore, it can be found that the 2×2 division strategy is more flexible than the 1×1 division strategy. Although the 1×1 division strategy gains slightly higher DT (0.15% more), its average BD-rate nearly doubles that of the 2×2 division strategy.

3.6 Conclusions

The novel statistical approach for motion estimation skipping (SAMEK) is proposed to skip motion estimation processes, thus helping to reduce the complexity and encoding time of HEVC encoder. The decision to skip follows ZeroCase and DecreaseCase rules, which are summarized based on the relationships between encoding PUs and corresponding references. Experimental results reveal the effectiveness of the SAMEK method, with an average time saving of 6.87% and negligible rate distortion loss. In addition, the SAMEK method can be integrated with other CU/PU level complexity reduction methods, as they can be implemented on different levels with no conflicts.

A future work might search for more rules to further reduce the overall complexity,

as well as the encoding time.

It is worth reporting that the work reported in this section has led to the following publication:

1. Yu L, Xiao J, Tillo T, Zhu C. Statistical Approach for Motion Estimation Skipping (SAMEK)[C]//Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015: 3245-3249.

Chapter 4

Dynamic Redundancy Allocation for Video Streaming using Sub-GOP based FEC Code

4.1 Introduction

The H.264/AVC video coding standard, as well as HEVC standard, are based on a hybrid coding mechanism which utilizes transform coding and motion compensation. Thus, they all suffer from error propagations when transmitted over packet lossy networks. To tackle this problem, much research has been done with several paradigms proposed. For example, there are server-side approaches, like intra macroblock (MB) refreshment [67], redundant picture coding with equal or lower quality [68], multiple description coding (MDC) [69] and forward error correction (FEC) [70]. There are also approaches proposed on the client side, such as Automatic Repeat request (ARQ) [71], and feedback-based reference picture selection (RPS) [72]. Among these methods, FEC is, in general, more superior in terms of coding efficiency. As for FEC, two commonly used erasure codes are Reed-Solomon (RS) code and Low Density Parity Check (LDPC) code. RS erasure code is one of the most studied protection methods for video streaming over unreliable networks. Besides, it is more appropriate for small block sizes and real-time streaming [73] than LDPC code. Thus, the proposed method is based on the RS erasure code and tested on H.264 encoded streams.

As a block-based error correcting code, large block size and increased number of parity packets will enhance the protection performance of RS erasure code. However, this enhancement is sacrificed by the error propagation and the increased bitrate for video applications. Many works that use RS code have appeared in recent years. In

[74], the importance of video packets is evaluated in terms of both GOP level and data partitioning level, which helps to divide the packets into different blocks. Then, unequal loss protection is formulated according to the network conditions. In [75], macroblocks are classified into three slice groups, and then unequal error protection (UEP) of H.264/AVC streams is used. These two methods [74, 75] only have a small number of importance levels, which limits their efficiency. In [70], parity packets are allocated to slices according to their impact on the distortion over the whole GOP. In [76], a model-based FEC assignment algorithm and a heuristic FEC assignment algorithm are proposed, which incorporate unequal protection at both GOP level and resynchronization packet level. These two methods add parity packets only according to the importance of each packet without considering the side effects of the added bitrate. Thus, the benefits in terms of protection to the source packets may be overwhelmed by the side effects of the added bitrate. In [77], the parity packets allocation problem is formulated as a constraint optimization problem over the lengths of Sub-GOPs and their assigned parity packets. The goal of the constraint problem is to achieve lower distortion, which was evaluated analytically. However, this method is based on many assumptions about the video characteristics, which limits its practical usage. Besides, the redundancy rate for the whole sequence is a constant value set manually.

Thus, in this thesis, the aforementioned drawbacks are overcome by proposing a dynamic redundancy allocation approach using Sub-GOP based FEC code, where the systematic RS erasure code is used. The redundancy allocation problem is formulated as a constraint optimization problem in the proposed method, which allows more flexibility in setting the block-wise redundancy. The distortion caused by losing each slice and the propagated error in future frames are taken into consideration in the optimization. With this code, the source packets are kept intact, and parity packets are generated to protect the source packets in each Sub-GOP. The length of each Sub-GOP, as well as its redundancy rate, are determined by the measurement of expected rate-distortion (RD) cost. The RD cost accounts for both expected end-to-end distortion (including propagated distortion of future frames) and the total bitrate including added parity packets. After gathering the frame-wise information of both distortion and bitrate, a greedy search for a proper allocation of the parity packets with lowest RD cost will be performed. In addition, the amount of introduced redundancy and the way it is introduced are automatically selected without human interventions based on the

network condition and video characteristics. The proposed scheme is implemented in JM14.0 for H.264, and it achieves an average gain of 1 dB over the state-of-the-art approach.

The rest of this chapter is organized as follows. Dynamic Sub-GOP FEC coding approach is introduced in Section 4.2; this approach is used as the benchmark in the following experiment. Then, end-to-end distortion estimation model for the proposed approach is described in Section 4.3. In Section 4.4, the rate-distortion cost based redundancy allocation method is provided. Then, experimental results are presented in Section 4.5. Finally, conclusions and future works are presented in Section 4.6.

4.2 Preliminary on Dynamic Sub-GOP FEC Coding

Dynamic Sub-GOP FEC coding (DSFC) approach [77] is used as the benchmark in the experiment. This coding method tries to allocate fixed number of parity packets using the Sub-GOP concept so as to minimize the expected distortion of the whole GOP, where Systematic RS erasure code is used. In this section, the systematic RS erasure code will be briefly introduced, then the DSFC method and its corresponding concepts will be described.

Systematic RS erasure code is widely used for error protection over packet erasure networks. It detects and recovers the erasures by adding several parity packets. These parity packets are calculated based on the source packets. In RS (N, K) code, $(N-K)$ parity packets are added to the K source packets to finally generate a codeword of N packets. As long as any K out of the N packets are received at the client side, all the N packets can be recovered. If less than K packets are received, then the RS correction will fail. However, the received source packets can still be used, as they are encoded intact in the systematic configuration. Nevertheless, in our work, the error concealment tool in the decoder will be further used to conceal the lost packets in this case.

Given the fixed parity packet rate K/N, the performance of RS code will be enhanced with increased K. However, for video applications, the error propagation will be more severe as K increases, because the correction will only be performed upon the reception of the parity packets. Based on this characteristic, DSGF method tries to seek the balance by allocating the parity packets according to the expected distortion. The expected distortion is calculated as the sum of expected internal and propagating distortion in each slice. For one GOP, a series of expected distortions are

calculated by evaluating the impact of inserting one parity packet in all potential positions (frames). Thus, an optimal position for this particular parity packet which has the lowest expected distortion value, will be chosen to add one parity packet. After all parity packets are allocated in this manner, the parity packets allocation pattern is obtained. In this pattern, each RS coding block is called Sub-GOP, which is shown in Fig.4.1. The concept of Sub-GOP will be used in our proposed method as well.

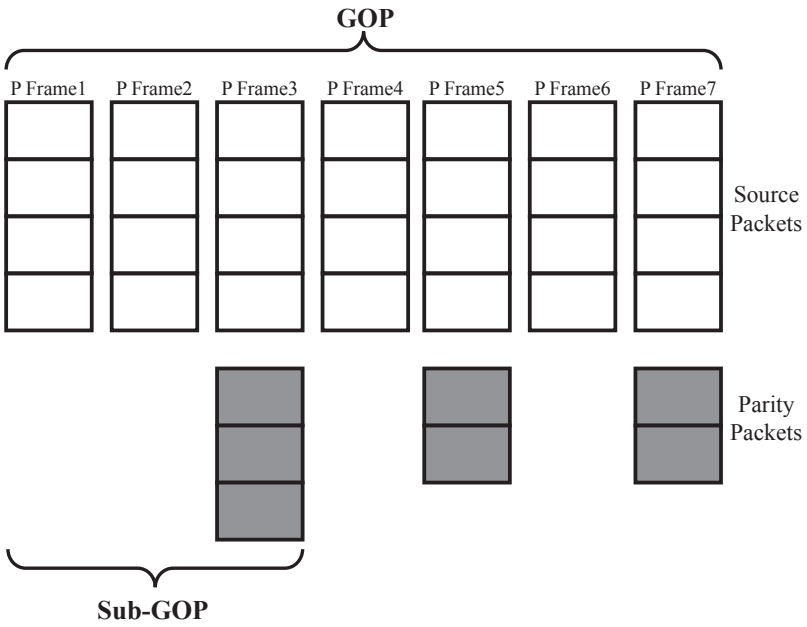


Figure 4.1: One example of RS parity packets allocation for both DSGF approach and our approach

However, the expected distortion in DSGF is calculated based on the assumptions that, every frame has a fixed number of slices and the distortion caused by losing each slice is equal. Thus, in our proposed method, real information is used instead. Another major improvement of the proposed method over DSGF is the redundancy tuning mechanism. In fact, in the DSGF approach, the redundancy rate needs to be set manually. Whereas, in the proposed approach, besides the length of Sub-GOP, the amount of redundancy and its allocation are tuned automatically according to the network conditions and video characteristics so as to enhance the overall system performance.

4.3 End-to-end Distortion Estimation

Before introducing the proposed method, the estimation of end-to-end distortion will first be described. It is an essential part of our work, as it measures the importance of the packets and influences the allocation of parity packets. The end-to-end distortion to be estimated is the expected distortion of the whole video, which is obtained by emulating the packet loss over each slice. For each slice, the end-to-end distortion accounts for two parts: one is the distortion of the lost slice itself, and another is the distortion caused by its error propagation. Finally, the expected distortion of the whole video is obtained by summing up the distortion of each slice.

In the proposed method, detailed characteristics of the video are used to evaluate the expected end-to-end distortion and bitrate, such as the number of slices in each frame, the distortion of losing different slices. As for the distortion caused by losing a slice, it is calculated by emulating its loss at the encoder side. Then, by using error concealment procedure on the emulated sequence at the encoder side, the mean square error of this slice with respect to the error-free version will be evaluated. The propagated distortion will be modeled as shown in [78]. Every slice in the sequence will be processed in this manner to get its corresponding distortion value. These values are then used in the calculation of the end-to-end distortion of the whole video. These parameters are calculated in advance to the redundancy allocation.

4.4 Proposed Method

The rate-distortion (RD) cost is used as the criterion for parity packets allocation in the proposed method. RD cost is widely used in many video compression applications, which optimizes the video quality against the required amount of data. Although different from video compression applications, our method is still aiming at enhancing the video quality against a given amount of bitrate. Thus, by applying the RD cost to the allocation of the RS erasure codes, it will find a balance between expected video quality and total bitrate. Thus, in this thesis, we will use the following RD cost equation:

$$RDscore = \frac{D + \lambda \cdot R}{N} \tag{4.1}$$

Where RDscore is the average RD cost for the current Sub-GOP, D is the expected

distortion of the Sub-GOP as described in the previous section, and R is the total bitrate of the current Sub-GOP and corresponding parity packets. The λ used here is the Lagrange multiplier for error-prone networks given by [67]. The effectiveness of this λ was verified experimentally in our work. Since our method works on frame level, we normalize the RD cost by the length of Sub-GOP (N). Given this frame level criterion, a greedy search algorithm is employed to search for the optimal allocation, which finds the allocation pattern with lowest RD cost. This algorithm iterates all the possibilities in terms of both the amount of parity packets in a predefined range of redundancies and the corresponding positions to place them. For each iteration, the RDscore is evaluated and stored. Once all the iterations are completed, the configuration with the minimal RDcost will be chosen. The detailed process of this algorithm is shown as flowchart in Fig.4.2, where the outer layer iterates every frame in one GOP, which helps to determine the proper length of each Sub-GOP. While the inner layer tries a variety of redundancy rates in the predefined range with certain steps and finally converges to the optimal value. Therefore, it can flexibly tune the redundancy rate. After this iteration, the parity packets pattern is set for the current GOP, and the processing of following GOPs are carried in a similar way.

Consequently, this method introduces two folds benefits: firstly, it enables the flexibility to determine the redundancy rate in a given range, which will guarantee the efficiency of the added data; Secondly, it helps to measure the optimal length of each Sub-GOP, which keeps a balance between the performance of RS code and error propagation.

4.5 Experimental Results

In the simulation, CIF video sequences with different characteristics, including moderate, fast movement and different motion directions, are chosen, namely, *Foreman*, *Bus* and *Stefan*. As for the codec, we use the JM14.0 H.264 codec, which enables the comparison with previous work. The GOP structure used is IPPP, with 30 frames in each GOP. The I frames are encoded as one Sub-GOP in both proposed method and DSGF method because, in general, there are many source packets for I frame, and they are essential reference frames which require more protection. While the redundancy allocation algorithm will be performed on P frames for both methods. For the motion prediction, only one reference frame is used. Each slice is transmitted in one

packet, and the length of the packet is 400 bytes. For the network part, packet loss pattern is assumed to follow the i.i.d model. In the DSGF approach, the slice per frame and redundancy rate are user-defined constant values for the whole GOP. Thus, these parameters are set for the DFSC method following the indications in [77].

The average luminance PSNR is obtained by averaging the average PSNR of all the frames over the 400 trials. For one sequence, different QPs are chosen to evaluate the performance for different bitrates. The QP range for *Foreman* is [22,30] with a step of 2, and [28,36] with the same step for *Bus* and *Stefan*. Thus, total bitrate covers a range between 400 Kbps and 2200 Kbps.

The results are shown in Fig.4.3, Fig.4.4 and Fig.4.5 for *Foreman*, *Bus* and *Stefan* respectively. All of them show a better performance than that of the DSGF method, with about 1 dB improvement for 5% packet loss rate and even larger improvements for higher packet loss rates. This is because the proposed method provides a more flexible mechanism to determine the redundancy rate according to the network conditions. Whereas, for the DSGF approach, the redundancy rate is a fixed value set manually. It can also be found that the redundancy rates of the proposed method and the DSGF method are similar for the video sequences with higher QPs, while quite different for those with lower QPs. This demonstrates that the proposed method tunes the redundancy rate according to the video characteristics. Consequently, it guarantees the efficiency of every added parity packets. It is also worth noticing that, in all tested sequences with lowest QP and for 5% packet loss rate, the proposed method achieves higher PSNR with lower redundancy rate in comparison to the DSGF method. This also confirms that the proposed approach adapts the total amount of redundancy according to the probability of packet loss rate and QP.

The above experiments are done under the i.i.d packet loss model, while the real internet packet loss is bursty. As a solution, the burst packet loss in the real internet scenario can be transformed into the i.i.d packet loss with the interlacing technique. Since our proposed method is not a real-time streaming method, the packets can be ordered using interlacing technique before sending. At the receiver side, the packets are reordered before displaying. The interlacing process can distribute the burst losses into different locations in a sequence which follows the i.i.d distribution.

As for the computational complexity, the proposed iterative algorithm includes 3 loops: The first loop iterates all positions (L) in one GOP; the second loop iterates all

possible size of sub-GOP ($[Size_{min}, Size_{max}]$); and the third loop iterates the possible numbers of parity packet ($[R_{low}, R_{high}]$). Therefore, the computational complexity is $O((Size_{max} - Size_{min}) * (R_{high} - R_{low}) * L)$. In fact, during the implementation, it is noticed that the selected sub-GOP size is limited to a range of $[1, 4]$ and the parity packet number is in the range of $[1, 9]$.

4.6 Conclusions

A novel unequal loss protection method has been proposed for H.264/AVC. The proposed scheme employs systematic Reed-Solomon erasure code to protect the video stream. For the allocation of parity packets, rate-distortion cost is incorporated into the method to search for the optimal length of each Sub-GOP, as well as the corresponding redundancy rate. By setting a redundancy rate range for searching, this method provides a more flexible and automatical redundancy allocation than other state-of-the-art approaches. Furthermore, in the calculation of RD cost, the characteristics of the video are exploited to provide more specific treatment. Experimental results showed the effectiveness of the proposed method in comparison to the DSGF approach. Further work will be devoted to reducing the complexity of the proposed approach.

It is worth reporting that the work reported in this section has led to the following publication:

1. Yu L, Xiao J, Tillo T. Dynamic Redundancy Allocation for Video Streaming using Sub-GOP based FEC Code[C]//Visual Communications and Image Processing Conference, 2014 IEEE. IEEE, 2014: 518-521.

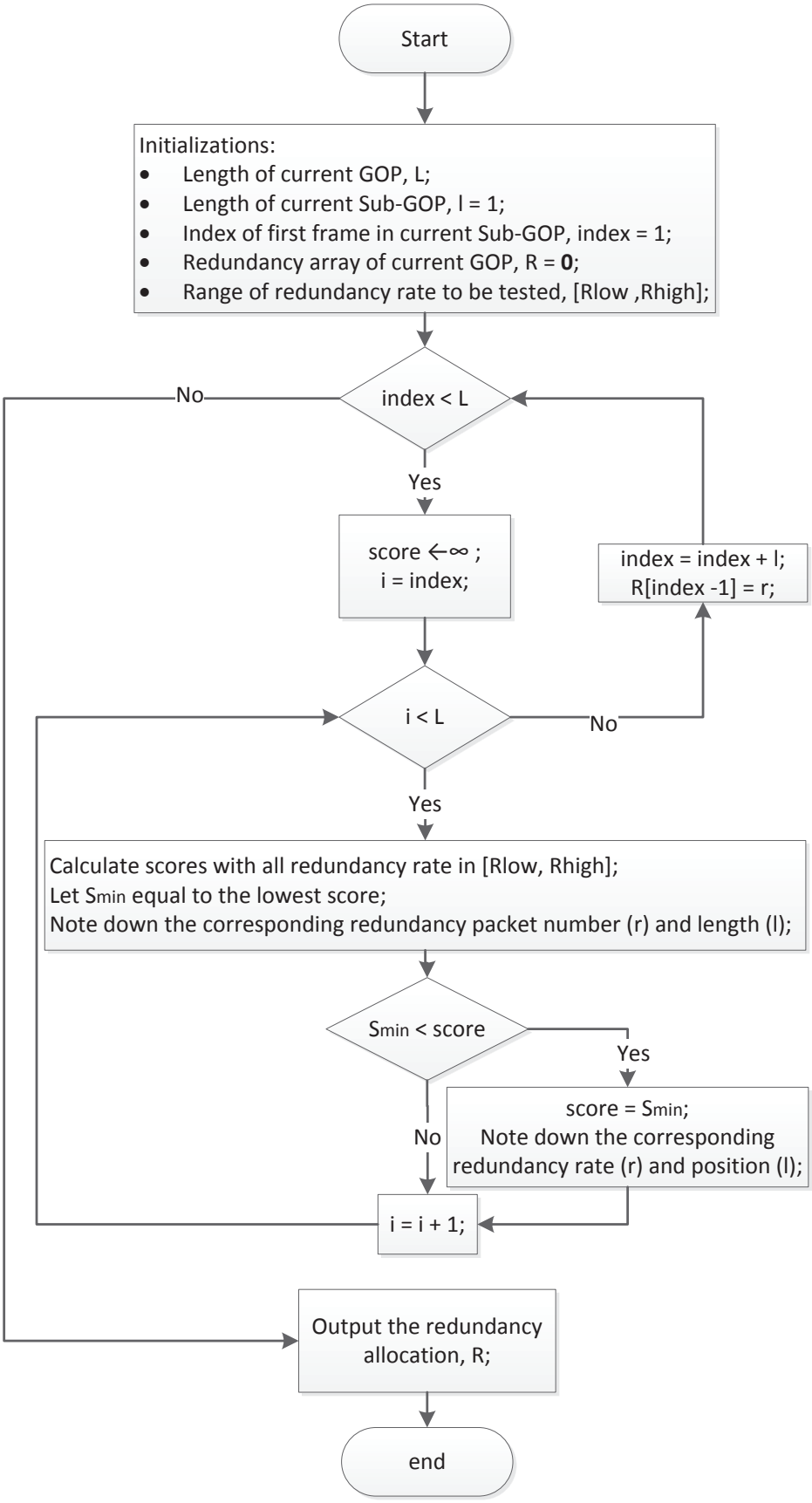


Figure 4.2: Flowchart of the proposed redundancy allocation algorithm

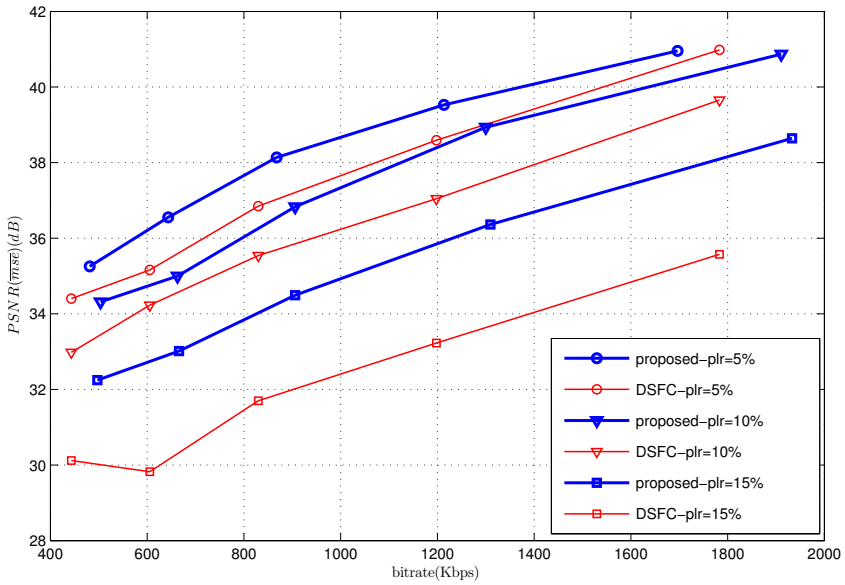


Figure 4.3: PSNR versus bitrate for CIF *Foreman* sequence; packet loss rates are 5%, 10% and 15%; RS parity packet rate for DSGF is 40%; the range of RS parity packet rate for proposed method is [1%, 50%].

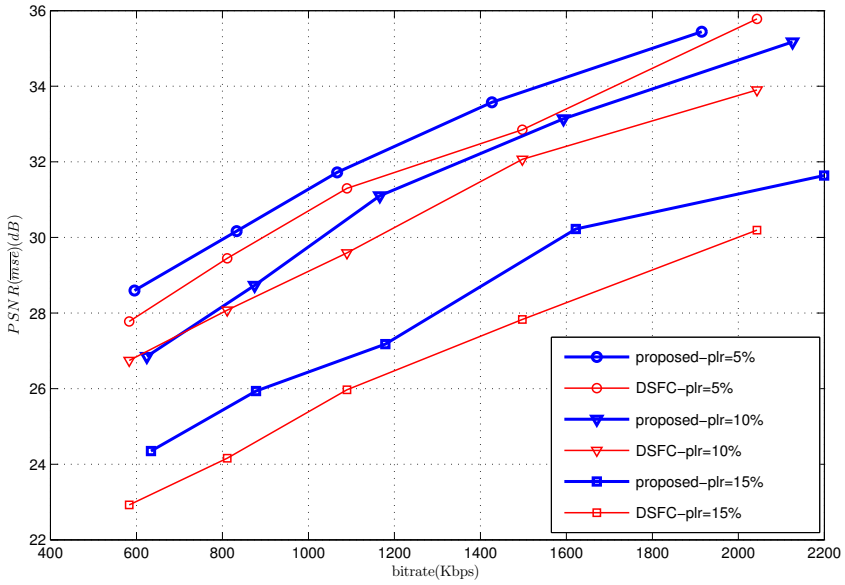


Figure 4.4: PSNR versus bitrate for CIF *Bus* sequence; packet loss rate are 5%, 10% and 15%; RS parity packet rate for DSGF is 40%; the range of RS parity packet rate for proposed method is [1%, 50%].

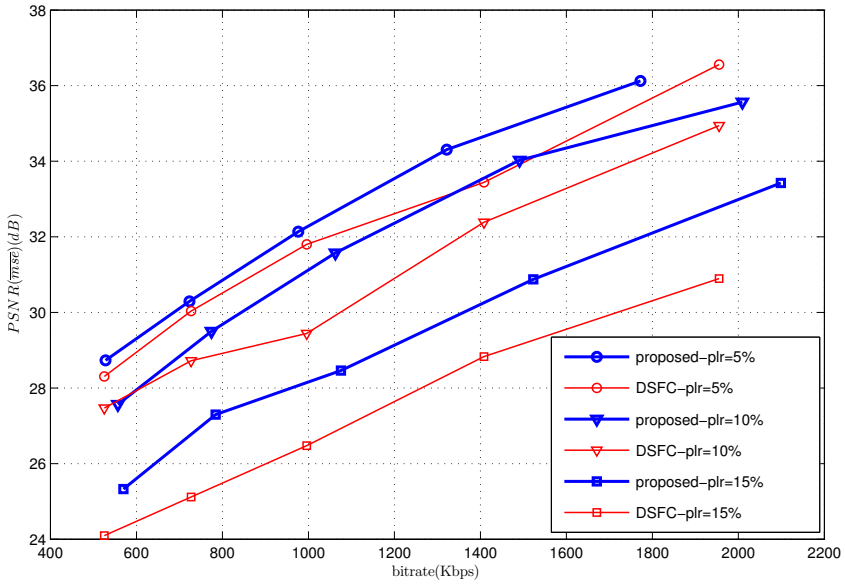


Figure 4.5: PSNR versus bitrate for CIF *Stefan* sequence; packet loss rate are 5%, 10% and 15%; RS parity packet rate for DSGF is 40%; the range of RS parity packet rate for proposed method is [1%, 50%].

Chapter 5

QoE-driven Dynamic Adaptive Video Streaming Strategy with Future Information

5.1 Introduction

It is indicated by Cisco that 64% of the Internet traffic were made up of videos in 2014, and will be 80% in 2019 [79]. It is difficult for the traditional RTP video streaming based methods [80] to meet this challenge. This is because RTP video streaming based methods do not provide good interoperability between different servers and devices. Besides, RTP packets are easily blocked by firewalls. Also, RTP video streaming is resource consuming to maintain separate streaming sessions for each server-client pair. Thus, based on the widely deployed HTTP networks, MPEG Dynamic Adaptive Streaming over HTTP (DASH) [41] was developed and standardized, which overcomes various drawbacks of RTP video streaming.

MPEG-DASH enables the adaptivity to the fluctuations of network throughput and capabilities of client devices. This adaptivity is enabled by preparing representations of various qualities for each video, along with associated metadata describing the characteristics of these different representations [81]. Meanwhile, one video is divided into a sequence of segments in time domain. These segments provide the feasibility to adapt the video quality to the network bandwidth with low latency. Based on the metadata and network condition, the client sends requests to the server to download proper representations. The mechanism of choosing a proper representation to download is an important research topic for DASH, which is also the target of this work. To sum up, DASH appeals to the market because of the following reasons: firstly, it takes good

advantage of content delivery networks (CDN), which is widely deployed in today's Internet. Secondly, it is based on HTTP protocol, which is firewall friendly. Last but not the least, it transfers the management of streaming from server side to client side, which saves much server resources and allows for dynamic flexibility.

There are two approaches to generate different representations of one video, namely CBR (constant bitrate) mode and VBR (variable bitrate) mode. The bitrate keeps constant across the whole video for CBR mode. While for VBR mode, the bitrate varies according to the contents of the video. VBR mode is commonly used in many video coding scenarios, such as using video coding standards like MPEG-2, MPEG-4 Part 10/H.264 [82]. This is because VBR mode strives to maximize the global quality of the encoded media by allowing a higher bitrate to be allocated to the more complex segments of media files while less is allocated to less complex segments [83]. For example, HEVC [4] usually sets a constant QP to guarantee a constant quality level across different frames, thus minimizing quality fluctuation and the associated visual discomfort. Consequently, the bitrate of each frame varies according to the complexity of the content.

Given the fact that the bitrate fluctuates a lot in the VBR video, it is of significant importance to explore this characteristic in a bitrate adaptation algorithm. However, this information is not specified in the metadata in MEPG-DASH standard. Instead, a general bitrate value of a bunch of frames (defined as representation in DASH) is conveyed to the client. Thus, the adaptation algorithm at the client side can only use this general information, which does not contain the detailed fluctuation characteristic. As a result, the mismatch between accurate instant bitrate and general bitrate would lead to non-optimal decisions. To tackle this mismatch problem, [84] proposed to include the instant bitrate information in the metadata. However, this proposal did not provide a final solution on how to use this information. Besides, there are also some works that attempt to estimate the instant bitrate [85–87], so as to restore this information. However, the estimation precision is limited. In our work, the accurate instant bitrate information will be included in the extension part of the metadata and sent to the client.

In this work, a QoE-based video bitrate adaptation method is proposed in the scenario of VBR coding mode and on-demand streaming [88]. The usage of the accurate instant bitrate of the segment is one of the main contributions of this work.

Other contributions are listed as follows. Firstly, the adaptation problem is modeled as an optimization problem, which tries to maximize the quality of experience (QoE) for the whole sequence. Secondly, the optimization problem of the whole sequence is solved by breaking it into sub-optimization problems of each segment to meet the real-time constraints. The goal function of each sub-optimization problem is formulated as "Internal QoE", which accommodates the need of a sustainable buffer reservation for future streaming. The overall QoE is optimized by combining all the sub-optimization solutions. Thirdly, the weights in the internal QoE metric can be flexibly tuned to meet different requirements. High preference of certain aspect can be achieved by assigning a high weight for the corresponding factor. As demonstrated in the experiments, the proposed method performs better than two typical heuristic methods in VBR modes, with over 27% gains in a smooth network and 78% gains in a fluctuated network respectively.

5.1.1 Related Works

In this part, several classic adaptation methods are described. The existing adaptation methods can be roughly classified into two categories, namely the heuristic rules-based methods and model-based methods. For the heuristic rules-based methods, they set up fixed strategies for different cases. As for the model-based methods, they treat the adaptation problem using existing models or transfer it into an optimization problem. These methods use both the network bandwidth information and the buffer reservation information.

The heuristic rules-based methods can be further classified into two subcategories based on the knowledge used to obtain the strategies. One subcategory is throughput-based methods, which only use the network bandwidth as reference [89][90]. As the throughput is used to make the decision for future segment, it needs to be estimated. The simplest way of estimation is to use the throughput of previous time slot, which can be calculated as the ratio of data size and the delivery duration of previous segment. However, this method suffers from short-term fluctuations. Thus, a smoothed throughput measurement method is proposed in [89]. This paper computes the throughput as the average download rates of the previous N seconds and tries to keep the requested bitrate around the throughput. With the smoothed bandwidth, the adaptation will be more stable and incurs less quality switches. Another algorithm which uses smoothed HTTP throughput measurement is [90]. Based on the estimated throughput, it pro-

poses a conservative step-wise up switching and aggressive down switching mechanism of representations. This method guarantees a timely adaptation to throughput, as well as reduced buffer overflow and underflow. The other subcategory is buffer-based method [43][44], which additionally uses the length of buffer reservation information. A partial-linear trend prediction model is proposed in [44] to accurately estimate the trend of client buffer level variation. Based on the estimation, the smoothness in the rate adaptation process is improved. While in [43], the future length of buffer reservation is estimated based on a trellis representation. The results shows that a smooth video quality is provided with buffer underflows eliminated. However, there is a main problem of the heuristic rules-based methods that they are deterministically tailored to specific network configurations.

When it comes to the model-based methods, more flexible solutions are provided compared to the heuristic rules-based methods. The rate adaptation behaviors are flexibly adapted to the dynamic settings of the network. [91] utilizes the reinforcement learning method to infer the optimal decisions trained in a simulated network. The action is the request of segment with certain bitrate, while the reward is the QoE estimation in the reinforcement learning method. With the QoE as the reward, human perception factor is directly involved in the algorithm. The reinforcement learning method is also introduced in [92] and [93] with the proposition of Q-Learning based clients. These two works can dynamically adjust the streaming behavior according to the current network status while maximizing the QoE. In [94], a subjective study to identify the impact of adaptation parameters on QoE is conducted. Based on this study, it proposes a method to compute the QoE-optimal adaptation strategy for DASH with mixed-integer linear programming. Similarly, [95] proposes a QoE-aware DASH system (QDASH). Besides, it proposes a probing-based network measurement method to facilitate the video quality selection. In [96], Markov Decision Process (MDP) is used to handle the stochastic decision problem, which minimizes both the number of starvation and the number of quality level changes and maximizes the quality level. The overhead of MDP based DASH approaches are evaluated and reduced in [97]. The work in [98] uses stochastic dynamic programing (SDP) techniques to achieve the tradeoff between requested quality and resulting video freezes. It considers two aspects to make the decision. One is that the requested average bitrate should be close to or below the measured bandwidth. Another is that the length of buffer reservation should be around

a predefined target value. In general, most of these works are based on the CBR-mode videos. However, VBR-mode videos are also common and easy to produce.

Thus, our paper will investigate towards the VBR-mode videos. The work [43] mentioned before also works on the VBR-mode videos, which is one of the benchmarks in our experiment. In that work, estimated bitrates of following segments are used to assist the decision, which may not be accurate. Thus, the accurate instant bitrate of each segment will be used in our proposed decision procedure. The accurate bitrate information will be sent along within the extension part of MDP file, which is standard compliant. Such modification to MDP file is also proposed in [84]. Based on this information, the mismatch between instant bitrate and specific bitrate is avoided. Thus, the decision is more accurate. Besides, the adaptation method is transformed into an optimization problem, which tries to maximize the overall QoE.

This chapter is organized as follows. In Section 5.2, the problem framework and benchmark methods will be described. While in Section 5.3, the proposed method is stated in detail. After that, experiments and discussions are presented in Section 5.4. Finally, conclusions are provided in Section 5.5.

5.2 Preliminaries and Adaptation Problem Formulation

In this section, related concepts of the adaptation algorithm are described, including Markov channel model and Quality of Experience (QoE). Besides, the two benchmark methods are introduced. Important notations and corresponding definitions are listed in Table 6.1.

5.2.1 Markov Channel Model

The wireless channel is modeled using finite-state Markov model and first-order Markovian assumption [99]. Assume there are L states of bandwidth level, denoted as $\{B_1, B_2, \ldots, B_L\}$. The probabilities of each bandwidth levels are $\{P_1, P_2, \ldots, P_L\}$. As it is based on the first-order Markovian assumption, the current bandwidth level is statistically independent of all other past and future bandwidth levels, except the previous bandwidth level. Thus, the transition probability is between two bandwidth levels, which are consecutive in time. The transition probability from B_i to B_j is defined as $P_{i,j}$. Then, the transition matrix is as follows:

Table 5.1: Descriptions of key symbols

Symbol	**Definition**
L	total number of available bandwidth state
M	total number of available quality level for the video
N	total number of segments in one video
$B_i(1 \leq i \leq L)$	all available bandwidth state
$P_i(1 \leq i \leq L)$	probability of each available bandwidth state
$P_{i,j}(1 \leq i,j \leq L)$	transition probability from bandwidth state B_i to B_j
$Q_i(1 \leq i \leq M)$	all available quality level, which $Q_i = i$
τ	duration of each segment
$t_i(1 \leq i \leq N)$	index of each decision point
$b_i(1 \leq i \leq N)$	index of bandwidth level at decision point t_i, $1 \leq b_i \leq L$
$b_i^{'}(1 \leq i \leq N)$	index of estimated bandwidth level at decision point t_i, $1 \leq b_i^{'} \leq L$
$q_i(1 \leq i \leq N)$	index of requested quality level at decision point t_i, $1 \leq q_i \leq M$
$r_{i,q_i}(1 \leq i \leq N, 1 \leq j \leq M)$	bitrate of segment with quality level j at decision point t_i, $r_{i,q_i} \in \mathbb{R}$
Θ	a chain of bandwidth levels chronologically, i.e.$\Theta = \{b_1, b_2, ..., b_N\}$
Ψ	a chain of requested quality levels, i.e.$\Psi = \{q_1, q_2, , q_N\}$
$T_i, 1 \leq i \leq N$	length of buffer reservation in time domain at decision point t_i
$T_i(\Theta, \Psi), 1 \leq i \leq N$	estimated length of buffer reservation in time domain at decision point t_i
$T_i^s, 1 \leq i \leq N$	duration of starvation at decision point t_i
$T^s(\Theta, \Psi)$	total starvation time for one sequence
$T^t(\Theta, \Psi)$	total playout time for one sequence, including the starvation durations
T^b	Size of buffer at the client side
l	the number of future segments involved in the decision for the current segment
Th	the safety threshold of buffer reservation that guarantees a smooth playout (in general benchmark)
$[T^{max}, T^{min}]$	constraint buffer range (in future benchmark)
λ	the weight of buffer reservation change factor in the internal QoE metric

$$A = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,L} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,L} \\ \cdots & \cdots & \cdots & \cdots \\ P_{L,1} & P_{L,2} & \cdots & P_{L,L} \end{bmatrix} \tag{5.1}$$

In this thesis, a five-state Markov model is employed. The probability of each state is deduced from the transition matrix, which represents a stable probability distribution for each state in the current network. This helps to reduce the influence of initial bandwidth state settings. As for the transition matrix, one bandwidth level will not jump to a non-adjacent level, that is,

$$P_{i,j} = 0, \text{if } |i - j| > 1. \tag{5.2}$$

Thus, the bandwidth level only jumps to the neighboring higher or lower bandwidth level or stays at the current level.

5.2.2 Quality of Experience

The Quality of Experience (QoE) is a concept of subjectively perceived quality, which takes into account how consumers perceive the overall quality of a service [100]. Thus, QoE is regarded as the goal of our proposed adaptation algorithm. As indicated in [100–103], QoE is mainly influenced by three key factors, namely requested media quality, quality switching frequency, and starvation events. Although startup delay (the period from time starting-to-download to time starting-to-play) is also an important aspect, a fixed startup delay (10s) is set in this work. Thus, it is not incorporated in the QoE evaluation as in [104].

Assume there are totally N segments in one video sequence. Each segment lasts for τ seconds. The DASH client requests segments of a proper quality level according to the available bandwidth. The requested quality levels are $\{q_1, q_2, \ldots, q_N\}$ correspondingly, which is denoted as the requested media sequence Ψ. While the bandwidth for downloading each segments are $\{b_1, b_2, \ldots, b_N\}$, which is labeled as bandwidth chain Θ. Then, the average requested media quality $E(\Psi)$ can be denoted as the average of all requested quality levels:

$$E(\Psi) = \frac{1}{N} \sum_{i=1}^{N} q_i. \tag{5.3}$$

The quality switching frequency $V(\Psi)$ can be evaluated as the average times of quality change between neighboring segments.

$$V(\Psi) = \frac{1}{N-1} \sum_{i=1}^{N-1} |q_{i+1} - q_i|. \tag{5.4}$$

While the starvation events under bandwidth chain Θ can be measured as the ratio of starvation event in time domain, i.e. total starvation time $T^s(\Theta, \Psi)$ over the total displaying time $T^t(\Theta, \Psi)$ as shown in the following equation:

$$P^s(\Theta, \Psi) = \frac{T^s(\Theta, \Psi)}{T^t(\Theta, \Psi)}; \tag{5.5}$$

where

$$T^t(\Theta, \Psi) = N * \tau + T^s(\Theta, \Psi). \tag{5.6}$$

The starvation event happens when the buffer becomes empty. Assume q_i is requested for i^{th} segment, and its corresponding bitrate is r_{i,q_i}. The network bandwidth is b_i at the downloading period, while the length of buffer reservation is T_{i-1} before downloading. Then, the corresponding starvation duration can be calculated as follows:

$$T_i^s = max(\frac{r_{i,q_i} * \tau}{b_i} - T_{i-1}, 0). \tag{5.7}$$

Thus, if enough buffer reservation is maintained before loading, i.e. $T_{i-1} \geq r_{i,q_i} * \tau / b_i$, the starvation will not happen. Otherwise, the starvation duration is the difference between downloading time and length of the buffer reservation. Then, the total starvation time can be calculated as the sum of all starvation durations:

$$T^s(\Theta, \Psi) = \sum_{i=1}^{N} T_i^s. \tag{5.8}$$

Up to now, all these three factors are defined. It is worth noticing that all of them are normalized, so as to guarantee fair comparisons between video sequences of different lengths. However, the goals of these three factors are conflicting with each other. When the goal is to minimize the starvation events, smallest available bitrates will always be selected. As a result, a low average media quality is incurred. Conversely, selecting highest available bitrates may lead to the high probability of starvation. Moreover, when the solution tries to have higher media quality under the constraint of low probability of starvation, the quality switching event will inevitably

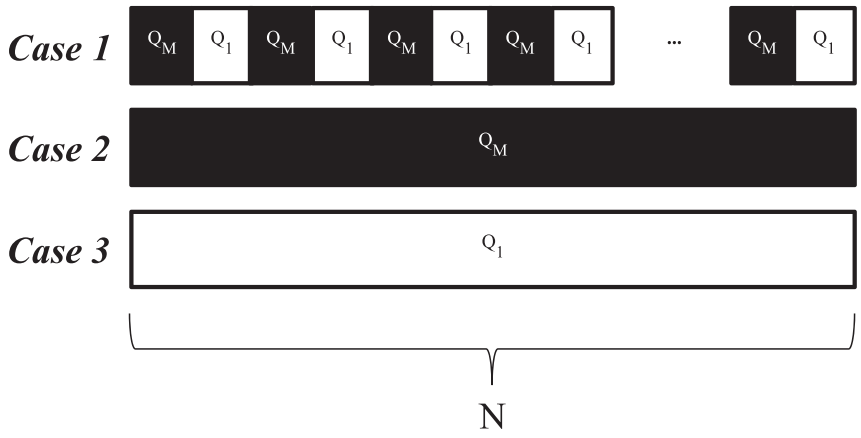


Figure 5.1: Three cases of "extreme" requested quality level sequence.

increase. Thus, these three factors are balanced with different weights in the QoE metric, which is calculated as follows:

$$QoE(\Theta, \Psi) = E(\Psi) - w_1 V(\Psi) - w_2 P^s(\Theta, \Psi); \tag{5.9}$$

where $\{w_1, w_2\}$ are the relative weights with respect to $E(\Psi)$. The weights will be tuned according to the different requirements of the client. The setting of w_2 is motivated by work in [104]. Based on the subjective tests, 10% of the starvation ratio is equivalent to 2 levels drop in the quality level. Thus, w_2 is set as 20. While for the setting of w_1, a sensible range is defined based on the following heuristic analysis. The highest available quality level is denoted as Q_M, while the lowest available quality level is denoted as Q_1. Three "extreme" cases of requested quality levels are shown in Fig. 5.1. Case 1 represents the requested quality levels with most switches, while case 2 and case 3 represent the ones with highest and lowest average quality respectively. The starvation ratio is assumed to be a constant for all cases. For simplicity, $P^s(\Theta, \Psi) = 0$. Then, the QoE values of these three cases are computed as follows:

$$QoE_{Case1} = \frac{1}{2} * (Q_M + Q_1) - w_1 * (Q_M - Q_1); \tag{5.10}$$

$$QoE_{Case2} = Q_M; \tag{5.11}$$

$$QoE_{Case3} = Q_1. \tag{5.12}$$

To most of the audience, QoE_{Case1} should be between QoE_{Case2} and QoE_{Case3}, that is $QoE_{Case3} \leq QoE_{Case1} \leq QoE_{Case2}$. Then, the range of w_1 is $[-\frac{1}{2}, \frac{1}{2}]$. Whereas,

w_1 should be a positive value since large quality fluctuation is regarded as a negative influence on QoE. Thus the range of w_1 is as follows:

$$0 \leq w_1 \leq \frac{1}{2}. \tag{5.13}$$

The weights can be flexibly tuned within a reasonable range according to different preferences. For example, if the client is not sensitive to quality level switching, a lower w_1 can be set to give a higher priority to the other two factors. If the client prefers high quality than fluent play out, then w_2 can be lowered to concentrate more on the quality factors.

5.2.3 Benchmark Methods

Given the Markov model based bandwidth and QoE evaluation metric, the key aspect of the adaptation problem is the strategy. In this section, a general framework of the bitrate adaptation problem, as well as two benchmark adaptation strategies, namely general buffer-based method [89, 90, 105] and future buffer based method [43], will be presented respectively.

Framework

The adaptation strategy is applied sequentially to consecutive decision points $\{t_1, t_2, \ldots, t_N\}$. At one decision point t_i, one quality level $q_i, (1 \leq i \leq N)$ will be selected among all available quality levels $\{Q_1, Q_2, \ldots, Q_M\}$ based on the buffer status and the predicted bandwidth $b_i^{'}$. The bandwidth prediction methods used in both benchmark are the same. A simple aggressive method [85] is employed, where the throughput of downloading the previous segment is used as the prediction of the current bandwidth. It is shown in [85] that the aggressive method obtains satisfactory result similar to the proposed method in [85], when the duration of segment is short (e.g. 2s or 4s). During the downloading time of one segment, the bandwidth $b_i (1 \leq i \leq N)$ is assumed to be stable. Besides the quality level, each segment is also associated with its bitrate value $r_{i,q_i}, (1 \leq i \leq N)$. At start, the segment with the lowest quality level will be chosen (i.e. $q_1 = Q_1$). Then, based on the adaptation strategy, the following segments will be requested, downloaded and stored in the buffer. The length of buffer reservation at each decision point t_i is $T_i (1 \leq i \leq N)$. When the filled buffer reservation is more than $10s$ (i.e. $T_i = 10, i = 5$), the client will start to play. Then, the buffer is influenced by both downloading speed and playout speed as shown below.

$$T_{i+1} = max(T_i - \frac{r_{i,q_i} * \tau}{b_i} + \tau, 0). \tag{5.14}$$

All buffer reservation T_i should be limited under the buffer size T^b at the client side. The download process stops, when buffer overflows, i.e. $T_i \geq T^b$.

General Buffer-based Method

This benchmark method is summarized based on several related works [89][90][105], which represents the core idea of general buffer-based methods. The adaptation decision is made based on the predicted bandwidth and length of the buffer reservation. At first, the estimated bandwidth $b_i^{'}$ is compared to bitrates of segments at all available quality levels r_{i,q_i}. Then, based on whether the length of buffer reservation T_i reaches the safety threshold Th, one quality level up or down is selected.

$$q_i = \begin{cases} Q_k \quad , r_{i,k} \leq b_i < r_{i,k+1} \text{ and } T_i < Th; \\ Q_{k+1}, r_{i,k} < b_i \leq r_{i,k+1} \text{ and } T_i \geq Th. \end{cases} \tag{5.15}$$

If $T_i < Th$, a lower quality is chosen, and vice versa. Instant bitrates of future segments can also be used in this method. Instead of using bitrate of current segment r_{i,q_i}, average bitrate of current and future l segments $\sum_i^{i+l} r_{i,q_i}/(l+1)$ is used to compare with the estimated bandwidth $b_i^{'}$ in the first step.

Future Buffer based Method

Similar to our proposed method, the work [43] also employs future instant bitrate information to assist the adaptation in VBR scenario. Thus, it is used as one of the benchmarks for comparison. The instant bitrates of future segments used in [43] are predicted from the downloaded segments following the prediction mechanism in [85]. For a fair comparison, accurate instant bitrates will be used in this algorithm. Based on the instant bitrates of future l segments, as well as the predicted bandwidth $b_i^{'}$, this work builds a trellis representation to estimate future buffer level $T_i^{'}$ following the rules of Up-case and Down-case. The goal of this method is to keep the buffer T_i within a given range $[T^{min}, T^{max}]$. The trellis representation contains a path of quality request decisions for future l segments. Each time one segment is downloaded, the buffer status would be checked and compared with the estimated buffer level. Once the difference is larger than τ, a new trellis representation would be built to replace the old one.

Therefore, the path is only updated at some of the decision points, i.e. the adaptation algorithm is not always performed before downloading each segment. As a result, the adaptivity is limited. Whereas in our proposed method, there is no such limitation. Because the adaptation algorithm is performed for all the segments. Thus, this is the reason that better performance is achieved using our method. Detailed discussion will be presented in the experimental section.

5.3 Proposed Method

5.3.1 Overview of the Proposed Method

The motivations of the proposed method will be explained in this section. In general, the proposed method tries to optimize QoE by exploring the future information (instant bitrates of segments to download) with a probabilistic bandwidth prediction model. To sum up, the motivations are two folds:

QoE Optimization

The viewing experience of the end user is regarded as the evaluation criteria of adaptation performance. There are two major ways to evaluate the viewing experience of users, including subjective way like MOS and objective way like QoE. The former one is usually time-consuming. Thus in this paper, the objective QoE metric is adopted as in other DASH works [99, 104]. Then, the adaptation problem is transformed into the optimization of QoE. However, the QoE metric is an overall evaluation of the adaptation results, which can only be obtained when all the segments are downloaded. Thus, internal QoE (QoE^{inter}) is proposed as the medium-term goal, which can be evaluated for each segment. By doing so, the global optimization problem is divided into a collection of simple and real-time sub-optimization problems. The details of the internal QoE metric will be discussed later.

Future Information

The future information refers to the instant bitrate of future segments. It is involved in each sub-optimization problem to provide insights into the future, which will lead to more globally optimal results than just investigating current and previous information. The reason to use future information is that bitrates of future segments differ from that of the current segment, as well as average bitrate. As mentioned in the Section 5.1,

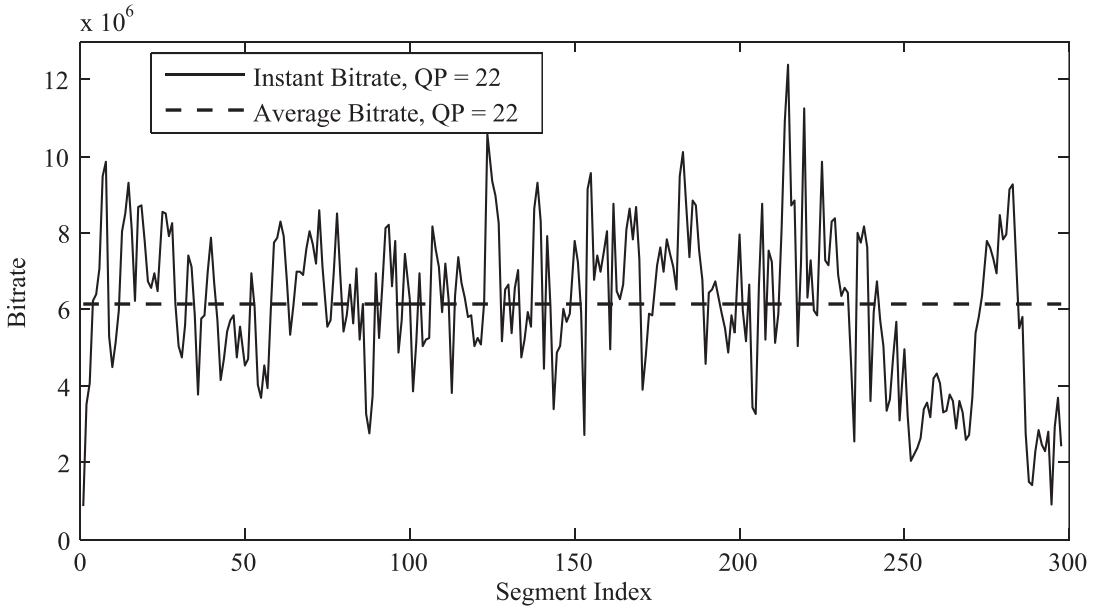


Figure 5.2: The bitrates versus segment indexes of sequence basketballPass are plotted when QP = 22. Average bitrate of the whole sequence is shown in dashed line for comparison. Similar phenomena happens for other QP settings and other video sequences.

most existing methods are based on the CBR mode video. Then, it is difficult for these methods to provide a comparable performance for VBR mode videos. It can be found in the Fig. 5.2 that, the instant bitrates of segments fluctuates a lot. Furthermore, only a small portion of the instant bitrate is similar to the average bitrate. It is worth highlighting that this is a common phenomenon in most video sequences. As a result, the methods that use the average bitrate in the adaptation mechanism will incur huge mismatches in the VBR mode videos, which will lower the overall performance. Thus, to avoid this effect, actual instant bitrates of the future segment are used in the proposed method by inserting them in the extension part of the MPD file.

Moreover, the decision at the current segment will influence the buffer status, which will further influence future decisions. For example, if the highest quality level is chosen for the current segment and the downloading time is higher than the duration of this segment. Then, a reduction in length of buffer reservation is caused. With a lower length of buffer reservation, future decisions will prefer to request lower quality levels to fill the buffer. However, this will not be a problem if future segment has a lower bitrate. Thus, it is better to consider the future trend to achieve global optimization. In the proposed method, future l segments after the current segment will be investigated

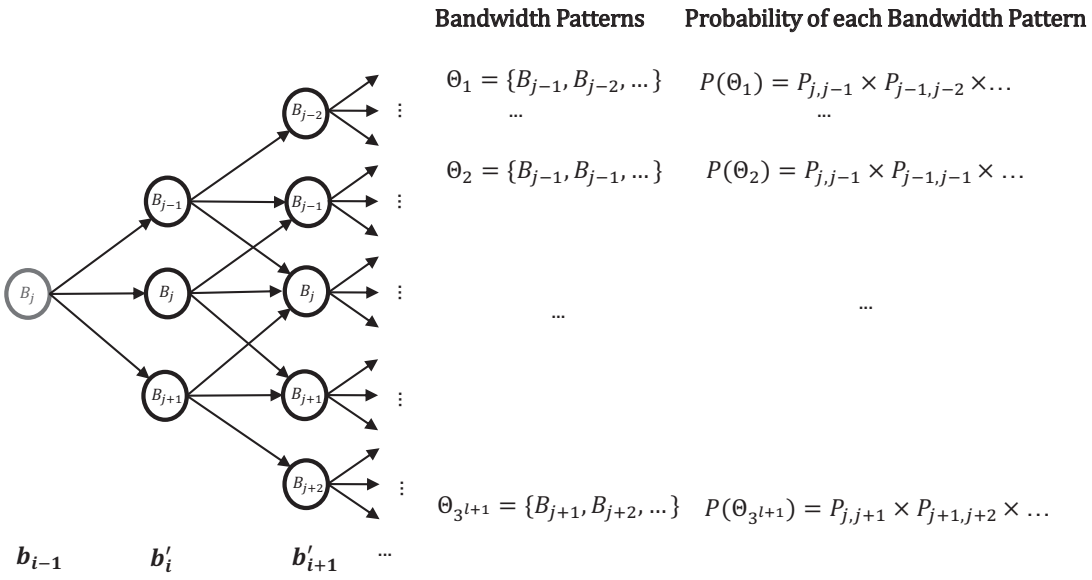


Figure 5.3: All possible bandwidth patterns over current and future l time slots $[t_i, t_{i+l}]$. $b_{i-1} = B_j$ is the bandwidth for downloading previous segment.

to help the decision for the current segment.

5.3.2 Markov Channel Model for Bandwidth Estimation

As the proposed method exploits l future segments, the bandwidth estimation is needed. In this work, the smoothed throughput is used for bandwidth estimation. Instead of using moving average as throughput estimation method, a heterogeneous Markov model is used to predict the future bandwidth [106–109]. The transition matrix used here is the same as Equ. (5.1). Supposing $(i-1)^{th}$ segment has been downloaded under the bandwidth $b_{i-1} = B_j$. The bandwidth for downloading current and future l segments are estimated as $b'_i, b'_{i+1}, ..., b'_{i+l}$, as shown in Fig. 5.3. As defined in Equ. (5.1), each state would only jump to neighboring states or stay in the current state. Thus, given $b_{i-1} = B_j$, b'_i could be B_{j-1}, B_j or B_{j+1}. Following this rule, there are totally 3^{l+1} possible throughput chains, as listed in the 'Bandwidth Patterns' column in Fig. 5.3. The probability of a bandwidth pattern $\Theta_k = \{b'_i, b'_{i+1}, ..., b'_{i+l}\}$ is calculated as

$$P(\Theta_k) = P_{b_{i-1}, b'_i} \times \prod_{j=0}^{l-1} P_{b'_{i+j}, b'_{i+j+1}}. \tag{5.16}$$

Detailed probabilities of each bandwidth pattern in Fig. 5.3 are shown in the last column. To sum up, the Markov channel model provides all possible bandwidth chains $\{\Theta_1, \Theta_2, ..., \Theta_{3^{l+1}}\}$, along with their probabilities, as the prediction output.

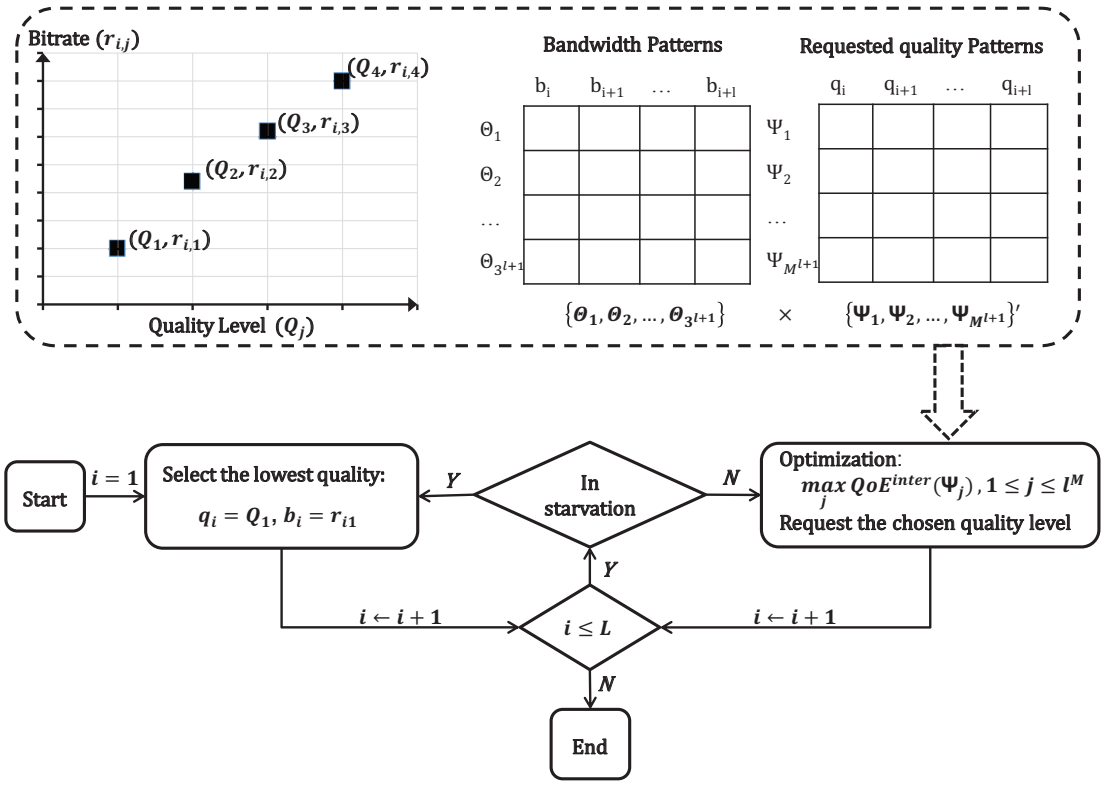


Figure 5.4: Flowchart of the proposed method is represented with solid line arrows and boxes. While dashed line arrows and boxes denote the information flow. The streaming process starts with the lowest quality level. Once the buffer is in starvation, the lowest quality level is requested until the starvation ends. When the buffer jumps out of starvation, the decision to choose which quality level follows the result of sub-optimization process. Information needed in the sub-optimization process are shown in the box of dashed line, including the accurate bitrate information, as well as all possible bandwidth patterns and requested quality patterns.

5.3.3 Proposed Method in Details

The working flow of the proposed method is shown in Fig. 5.4. At the beginning, the first segment with the lowest quality is requested to reduce the startup delay. For all the following segments, the proposed adaptation method would decide which quality level to request based on the result of the sub-optimization problem. The sub-optimization problem selects the quality level which reaches the maximum expected internal QoE score.

$$\begin{aligned} \max \quad & QoE^{inter}(\Psi_j), \\ s.t. \quad & j \in \{1, 2, \dots, M^{l+1}\}. \end{aligned} \tag{5.17}$$

The maximization problem is solved by a greedy search approach among all possible requested quality patterns. Each requested quality pattern is a chain of requested quality levels for current and future l segments. For example, $\Psi_j = \{Q_1, Q_1, \ldots, Q_1\}$ is one requested quality pattern, where the quality levels selected for the current and following segments are all Q_1. For each segment, there are M quality levels to be chosen from. Thus, there are M^{l+1} requested quality patterns $\{\Psi_1, \Psi_2, \ldots, \Psi_{M^{l+1}}\}$. The expected internal QoE score for each requested quality pattern is calculated as follows:

$$QoE^{inter}(\Psi_j) = \sum_{i=1}^{3^{l+1}} QoE^{inter}(\Theta_i, \Psi_j) * P(\Theta_i), \tag{5.18}$$

where $QoE^{inter}(\Theta_i, \Psi_j)$ represents the internal QoE score obtained from downloading Ψ_j under bandwidth Θ_i. As described in the previous sub-section, there are totally 3^{l+1} predicted bandwidth patterns with different probabilities. The expected internal QoE score for Ψ_j is a weighted average of internal QoE score under all possible bandwidth patterns Θ_i, with probability $P(\Theta_i)$ as weights.

Finally, the requested quality pattern Ψ_j with maximum expected internal QoE score will be chosen. Then, the quality level of the current segment will be selected following this Ψ_j pattern. Since now, the decision has been made, and the request will be sent to the server. Then, the requested quality level, as well as the actual network bandwidth, will be fed into the next round of decision. If the client is in starvation, the lowest quality level is requested to reduce delay.

In the following section, detailed information of the internal QoE metric, as shown in Equation (5.20) will be described .

5.3.4 Goal Function of Sub-Optimization: Internal QoE Metric

The goal function plays a vital part in the whole optimization process. The internal QoE metric is proposed as the medium-term optimization goal, which leads to a nearly optimal result of the adaptivity problem. Similar to the QoE metric that has been introduced previously, internal QoE metric also tries to improve the three factors: requested media quality, quality switching frequency, and starvation events. The difference lies in that the internal QoE metric evaluates the performance in the middle of the streaming process. In this case, the streaming will need to continue after this evaluation. Thus, it is important to incorporate the future effect into the internal QoE evaluation. Buffer reservation, which is the common fortune across the whole streaming process, plays a

key role in future effect. Thus, the change in length of buffer reservation, called "buffer change" for short, caused by the current decision is included in the internal QoE metric. Different from starvation factor in the QoE metric, which accounts for the starvation probability for now, buffer change factor takes responsibility for starvation probability afterward, i.e. in future. Like the three factors in QoE metric, buffer change factor is also normalized as a value per segment. Given a bandwidth pattern Θ and a requested quality pattern Ψ over future $l+1$ segments, the estimated length of buffer reservation at decision point t_{i+l} can be denoted as $T_{i+l}(\Theta, \Psi)$. Then, the normalized buffer change can be calculated as follows:

$$\Delta T_i(\Theta, \Psi) = \frac{T_{i+l}(\Theta, \Psi) - T_{i-1}}{l+1}. \tag{5.19}$$

It is incorporated into the internal QoE metrics as follows:

$$\begin{aligned} QoE^{inter}(\Theta, \Psi) = E(\Psi) - w_1 V(\Psi) - w_2 P^s(\Theta, \Psi) \\ +\lambda \Delta T_i(\Theta, \Psi), \end{aligned} \tag{5.20}$$

where λ is the weight of buffer change factor that balances its importance against the other three factors. When the buffer has accumulated enough reservations, the buffer change factor will not be that important. That is, the increase in the buffer will not be as important as other three factors, while the decrease in the buffer will also not cause disastrous results. Thus, λ can be assigned with a relatively small value. On the contrary, when the length of buffer reservation is under the safety threshold, it is of crucial importance to ensure an increasing trend in buffer change. In this case, λ should be set to a relatively high value. To sum up, the setting of λ can be represented as a linear function of buffer reservation with a negative slope, i.e. the bigger $T_{i+l}(\Theta, \Psi)$, the smaller λ. It can be represented as follows:

$$\lambda_i = a * T_{i+l}(\Theta, \Psi) + b, a < 0. \tag{5.21}$$

It is worth noting that, the length of buffer reservation used here is the one at decision point $i+l$, which is the final status for current decision. In the following experiment section, Equation (5.21) will be further investigated.

Table 5.2: Average bitrates of different versions of test video sequence "Big buck Bunny"

Quality Level	**QP**	**Average Bitrates (kbps)**
Q_4	22	733.66
Q_3	27	383.29
Q_2	32	183.06
Q_1	37	88.52

5.4 Experimental Results

In this section, the experimental settings are introduced first. Then, the investigation of parameters in the proposed method is discussed. Based on these settings, the comparisons between proposed method and benchmarks in both smooth and fluctuated networks are provided. Finally, the robustness of the proposed method to perturbed bandwidth prediction is shown.

5.4.1 Experimental Settings

The proposed method will be evaluated in comparison with two benchmarks as described in Section 5.2. The general buffer-based method and the future buffer based method are called "general benchmark" and "future benchmark" respectively for simplicity. Th is set as $10s$ in the general benchmark method. Buffer range $[T^{min}, T^{max}]$ in the future benchmark method is set as $[10s, 30s]$ according to [43]. The interaction between the DASH server and client is simplified and simulated using Matlab. The test video sequence "Big buck Bunny" [110] is encoded by the main profile of AVC (Advanced Video Coding) [111] with different QPs, namely $\{22, 27, 32, 37\}$, to represent different VBR video versions. That is, four quality levels will be provided, i.e. $Q_1 = 1, Q_M = 4$. Each video file has a frame rate of $24fps$ and a resolution of 352×288. Segments are generated with fixed duration $\tau = 2s$ and stored as separate files. The total number of segments is 298. Average bitrates of each quality version are shown in Table 5.2.

For the wireless network simulation, five levels of bandwidth state are used, namely $900, 600, 300, 140$ and 50 kbps. The lowest bandwidth state 50 kbps is lower than the lowest average media bitrate, which is a reasonable arrangement. The transition probabilities between different states are represented by the following transition matrix:

$$A = \begin{bmatrix} 0 & 0.05 & 0 & 0 & 0 \\ 0.03 & 0 & 0.03 & 0 & 0 \\ 0 & 0.03 & 0 & 0.02 & 0 \\ 0 & 0 & 0.02 & 0 & 0.03 \\ 0 & 0 & 0 & 0.06 & 0 \end{bmatrix}. \tag{5.22}$$

The matrix A represents a smooth network with few bandwidth fluctuations. Besides, a fluctuated network is derived with $10 \times A$ as a transition matrix. Both settings are used in the experiments to evaluate the effectiveness of the proposed method for different network scenarios. Totally 2000 unique bandwidth chains are prepared for simulation. Each obtained QoE result is averaged over these 2000 trials to obtain statistical stable results.

5.4.2 Investigation of Weights Setting

As mentioned before, one advantage of the proposed QoE-based method is the flexibility to tune the weights of different factors so as to appeal to different demands. Thus, different settings of the w_1 and w_2 in QoE metric are investigated in this part to investigate their influences on final performance. Besides, the parameter λ in the internal QoE metric, which has a direct influence on optimization result, is also investigated.

(1) w_1 **and** w_2

The values of w_1 and w_2 decide the preference on different factors. Thus, modifying weights in a proper range would help to meet different requirements and preferences of different users.

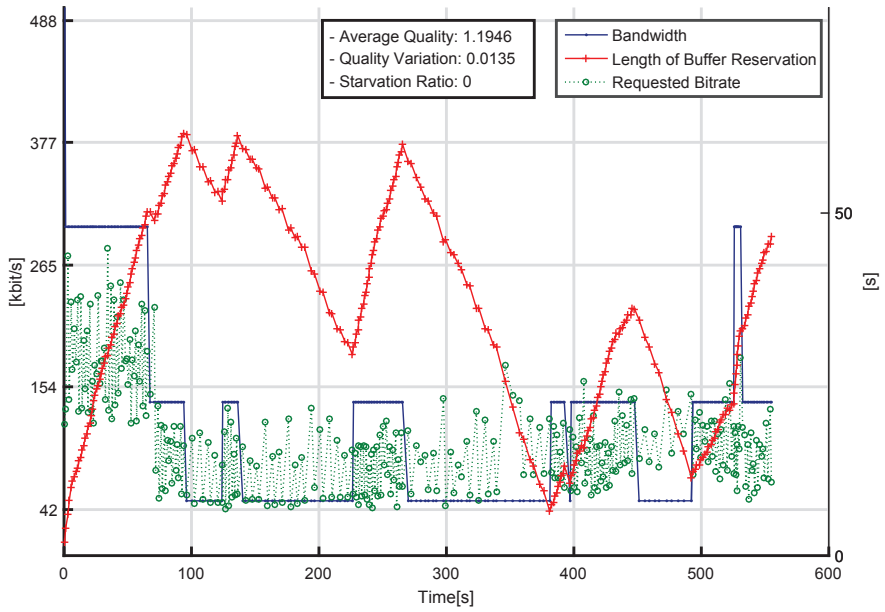
The proper range has been analyzed in Section 5.2-B. With the total number of available media quality levels set as 4, the range of w_1 is $[-1/2, 1/2]$. Meanwhile, the setting of w_2 can be determined by mapping the QoE loss caused by starvation events to that of quality degradations.

Within the range, different settings of w_1 and w_2 are assessed. As expected, when the weights changes, the range and the meaning of QoE value would change accordingly. Thus, QoE values are incomparable across different settings of weights. Instead, the scores of three factors are used for comparison here, which are shown in Table 5.3. The first row, i.e. $w_1 = 1/3$ and $w_2 = 20$, is used as benchmark for comparison. With higher w_1, quality variation is reduced when comparing the first two rows. Similar observations can be found with $w_2 = 1$, where the starvation ratio becomes higher. w_1

Table 5.3: QoE performance with different setting of w_1 and w_2 ($l = 1, \lambda = 0.9$).

w_1	w_2	Average Quality	Quality Variation	Starvation Ratio
$\frac{1}{3}$	*20*	*2.2601*	*0.0281*	*0.0347*
$\frac{1}{2}$	*20*	*1.9966*	*0.0135*	*0.0333*
$\frac{1}{3}$	*1*	*2.316*	*0.0857*	*0.1132*

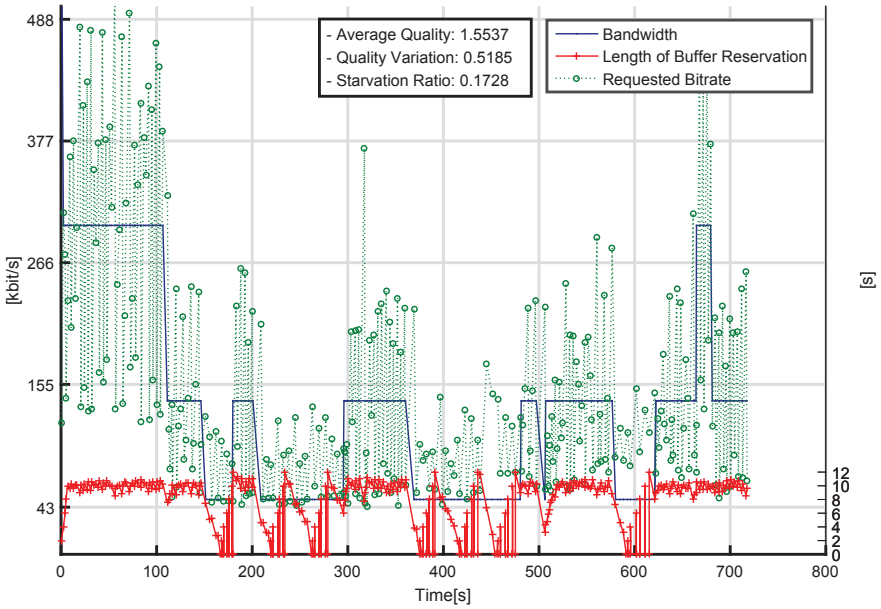
and w_2 are fixed as 1/3 and 20 respectively in the following experiments.



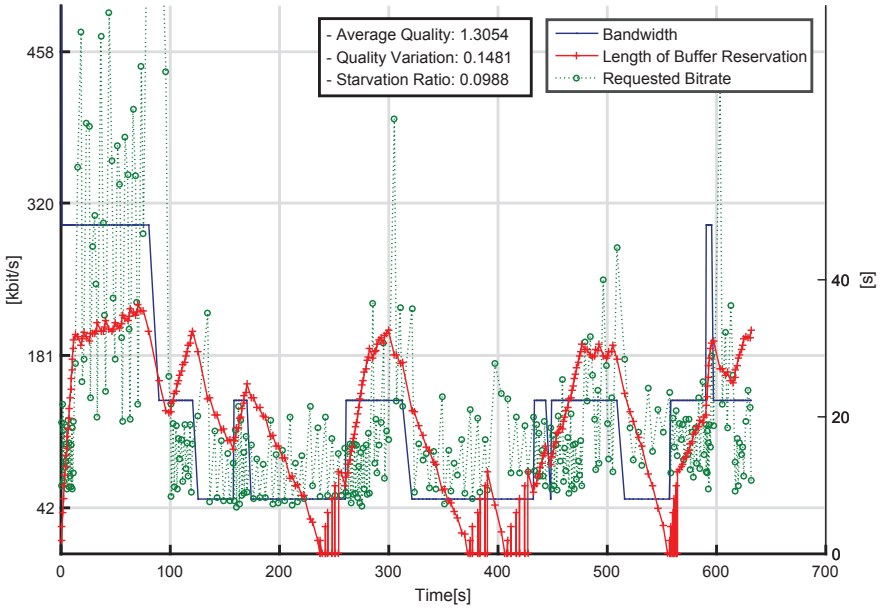
(a) Proposed QoE-based Method

(2) λ

As an important parameter in the proposed method, λ balances the tradeoff between the change of buffer reservation and other three QoE factors. The setting of λ has a direct influence on the adaptation decision, as well as the final QoE performance. Experiments under different network settings are evaluated. Generally, the peak QoE values are obtained when λ is 0.9. The linear λ is investigated based on the best fixed value for λ, i.e. 0.9. The average length of buffer reservation is around $48s$ in that setting. Thus, a and b would roughly satisfy the following equation $0.9 = a*48+b, a < 0$. Different combinations of a and b are evaluated. The combination $(a = 1.86, b = -0.02)$



(b) General Benchmark Method

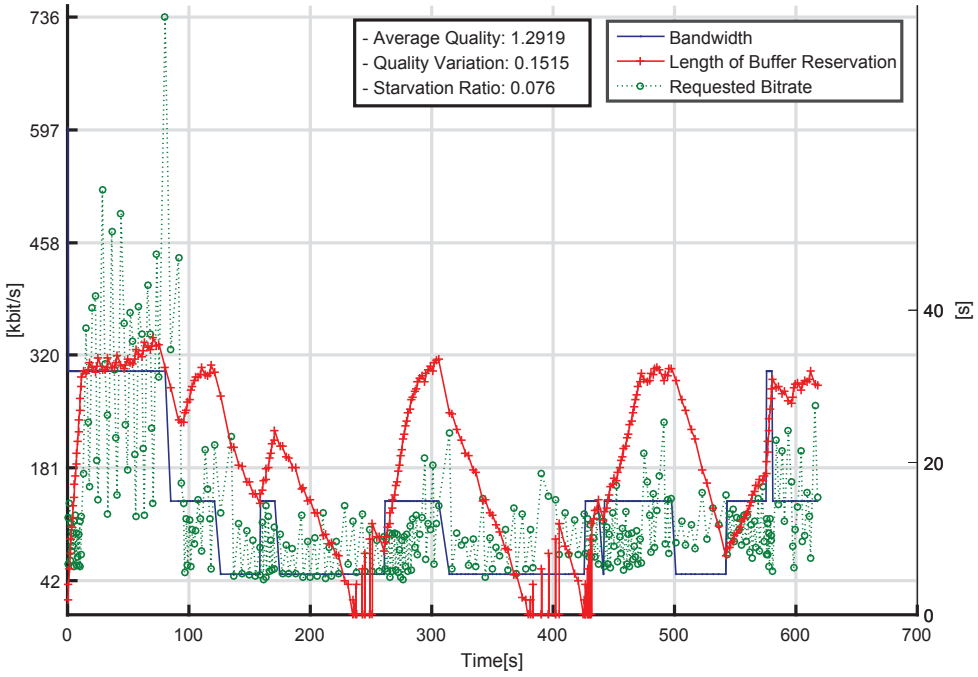


(c) Future Benchmark Method using Predicted Bitrate

is selected and will be used in the following experiments, so

$$\lambda = 1.86 - 0.02 \times T_{i+l}(\Theta, \Psi). \tag{5.23}$$

The QoE performance improves about 8% with linear λ, comparing to the fixed λ.



(d) Future Benchmark Method using Actual Bitrate

Figure 5.5: Illustration of bandwidth, requested media bitrate and length of buffer reservation for both benchmarks and proposed method for $l = 1$. Both future benchmark method using (c) predicted and (d) actual bitrate are assessed. The detailed values of average quality, quality variation and starvation ratio are also tagged. The right vertical axis is scaled with same maximum value for easy comparison of the length of the buffer reservation.

5.4.3 Comparison to Benchmarks

In this part, the proposed method is compared to the two benchmarks. To demonstrate the performance under different network scenarios, both smooth and fluctuated networks are used in the experiment.

The bandwidth, requested media bitrate and length of buffer reservation over one sample adaptation session are illustrated in Fig.5.5. The proposed method, the general benchmark and the future benchmark, which uses predicted and actual bitrate, are reported. When the length of buffer reservation falls to 0, it means the starvation happens. It can be found that there is much fewer starvation events happening for the QoE-based method than the benchmarks. When the bandwidth is even lower than the lowest media bitrate (bandwidth valley), starvation is still avoided by taking advantage of the previously accumulated buffer reservations as shown in Fig.5.5-(a).

Table 5.4: The QoE performance of both benchmark methods and proposed methods with a different look-ahead length l. For the future benchmark method, both cases are assessed, including using predicted bitrate and using actual bitrate in the adaptation module. Both smooth (A) and fluctuated ($10 \times A$) networks are evaluated.

Method		l	A				$10 \times A$			
			QoE	Average Quality	Quality Variation	Starvation Ratio	QoE	Average Quality	Quality Variation	Starvation Ratio
General Benchmark		-1	0.45	2.5602	0.5463	0.0966	-0.49	2.5205	0.6158	0.1404
		0	0.63	2.6150	0.5275	0.0905	0.37	2.5226	0.5928	0.0976
		1	0.71	2.5884	0.4374	0.0869	1.18	2.4473	0.4941	0.0552
		2	0.70	2.5903	0.4427	0.087	1.15	2.4509	0.4991	0.0569
		3	0.71	2.5908	0.4286	0.0867	1.20	2.4464	0.4912	0.0542
Future Benchmark	Predicted bitrate	1	1.28	2.3923	0.236	0.0867	1.89	2.0909	0.2723	0.0055
		2	1.30	2.356	0.2217	0.049	1.89	2.0795	0.2549	0.0053
		3	1.30	2.3616	0.1779	0.0504	1.92	2.0789	0.2433	0.0039
	Actual bitrate	1	1.31	2.3888	0.2599	0.0498	1.90	2.0931	0.2654	0.0053
		2	1.32	2.3676	0.2146	0.0487	1.90	2.0818	0.2463	0.0049
		3	1.35	2.3683	0.1587	0.0487	1.92	2.0779	0.2313	0.004
QoE-based Method		1	1.68	2.3847	0.0349	0.0345	2.11	2.1924	0.1958	0.0008
		2	1.70	2.3866	0.0373	0.0337	2.10	2.1784	0.2004	0.0007
		3	1.69	2.3826	0.0373	0.0338	2.08	2.1478	0.1632	0.0007

As for the general benchmark, the requested bitrates closely follow the fluctuation of bandwidth, even during the very short peak at $660s$ in Fig.5.5-(b). As a result, the buffer reservation is always at a low level, which makes it vulnerable to starvation during bandwidth valleys. While for the future benchmark method, the starvation is less severe than the general benchmark. It only happens when bandwidth valley lasts for over $40s$. It is worth to notice that, the future benchmark using actual bitrate avoids the starvation at $550s$ as in Fig.5.5-(d). While the one using predicted bitrate in Fig.5.5-(c) fails, which is due to the unprecise bitrate information.

Besides, the variation of requested bitrates in the proposed method is much lower than the benchmarks, which would guarantee a stable watching quality. The corresponding quality variation of the proposed method is 0.0135.

While for the general benchmark, the quality variation is 0.5185, nearly 38 times of the proposed method. As for the future benchmarks, the quality variation is almost 10 times of the proposed method. It is worth to notice that, there is fewer overshoots of requested bitrates for Fig.5.5-(d), when comparing with the requested bitrates with Fig.5.5-(c).

In our experiment, the buffer size is not limited. In reality, the buffer size depends on the policy of the service provider, as well as storage limitations. If the buffer reservation is 48s and the highest bitrate of the downloaded sequence is 4Mbps then the allocated memory should be : 4Mbps $\times$ 48s = 196Mb. In the case of buffer restriction, our method can still perform well. Because when the buffer is full filled, the download, as well as the request, of the following segments are paused. When the request restarts, the proposed algorithm still tries to maintain a balance between the requested quality level, the quality switching frequency, the probability of starvation, as well as the possible change of the buffer level. Thus, in general, the adaptation result would not be much different from the above result. The difference would occur when the bandwidth remains in the low level for a long time, there would be a higher probability of starvation given a smaller buffer size. However, our method will still outperform the benchmarks with a lower probability of starvation.

Detailed QoE performances with different look-ahead lengths l are shown in Table 5.4 under both smooth and fluctuated networks. $l = -1$ and 0 represent using the bitrate of previous and current segment respectively, while $l > 0$ denotes that bitrate of future l segments are used. It can be observed that the QoE performance enhances a lot from $l = -1$ to 0 and $l = 0$ to 1 for the general benchmark method. This is rational because more information guarantees wiser decisions. In addition, the increase mainly comes from lower quality variation and starvation ratio. This demonstrates the importance of using future information. When $l \geq 1$, the QoE performance generally remains stable for all methods. This demonstrates that the information of farther segment has fewer contributions. Based on this observation, l can be set as 1 to obtain a desirable result while maintaining a low computational complexity. When it comes to the future benchmark method, the one using actual bitrate always gets better performance than the one using predicted bitrate. This reveals the importance to use the actual bitrates, if possible, rather than the predicted ones. Generally, in the smooth network, the proposed method outperforms the benchmarks, with over 27% and 138% improvement in QoE value comparing to future benchmark method with actual bitrate and general benchmark method respectively. While in the fluctuated network, the improvements are 78% and 172% respectively. To sum up, our proposed method is effective in both smooth and fluctuated networks.

Table 5.5: QoE performance under perturbed bandwidth prediction.

Transition Matrix used for Bandwidth Prediction	QoE	Average Quality	Quality Variation	Starvation Ratio
$\begin{bmatrix} 0 & 0.05 & 0 & 0 & 0 \\ 0.03 & 0 & 0.03 & 0 & 0 \\ 0 & 0.03 & 0 & 0.02 & 0 \\ 0 & 0 & 0.02 & 0 & 0.03 \\ 0 & 0 & 0 & 0.06 & 0 \end{bmatrix}$	1.68	2.3847	0.0349	0.0345
$\begin{bmatrix} 0 & 0.06 & 0 & 0 & 0 \\ 0.02 & 0 & 0.04 & 0 & 0 \\ 0 & 0.05 & 0 & 0.01 & 0 \\ 0 & 0 & 0.04 & 0 & 0.01 \\ 0 & 0 & 0 & 0.05 & 0 \end{bmatrix}$	1.67	2.3837	0.0325	0.0354
$\begin{bmatrix} 0 & 0.6 & 0 & 0 & 0 \\ 0.2 & 0 & 0.4 & 0 & 0 \\ 0 & 0.5 & 0 & 0.2 & 0 \\ 0 & 0 & 0.4 & 0 & 0.1 \\ 0 & 0 & 0 & 0.5 & 0 \end{bmatrix}$	1.62	2.3397	0.0286	0.0357

5.4.4 Evaluation of Robustness to Perturbed Bandwidth Prediction

The accuracy of bandwidth prediction has a direct influence on adaptation algorithm. Thus, the robustness of the proposed method to perturbed bandwidth prediction is evaluated in this part. A perturbed transition matrix is used in the prediction process to investigate its influence on the final result. As shown in Table 5.5, the first row, which uses accurate transition matrix for bandwidth prediction, is used as a benchmark for comparison. The second row uses a randomly perturbed transition matrix with a similar order of magnitude as the accurate one. The QoE performance is 1.67, which is similar to the result of accurate one. While the third row uses a randomly perturbed transition matrix with a higher order of magnitude. The QoE score is 3.6% lower than the result of accurate one. These results demonstrate the robustness of the proposed method to perturbed bandwidth prediction.

5.5 Conclusions

In this work, a QoE-based video adaptation method is proposed to adapt VBR video streaming over HTTP. This method incorporates the QoE evaluation metric, which is the goal of the adaptation problem, into the decision mechanism. Besides, the adaptation problem is transformed into an optimization problem, which is divided into a collection of sub-optimization problems to make the algorithm real-time resolvable. Meanwhile, the instant bitrates of each segment are sent in the extension part of the MPD file to precisely follow the bitrate fluctuation of the VBR video. Experimental results showed the importance of using accurate instant bitrate information and looking

ahead into future segments. Also, the proposed method outperforms the two benchmarks by 27%, 138% in a smooth network and 78%, 172% in a fluctuated network respectively.

It is worth reporting that the work reported in this section has led to the following publication:

1. Yu L, Tillo T, Xiao J. QoE-Driven Dynamic Adaptive Video Streaming Strategy With Future Information[J]. IEEE Transactions on Broadcasting, 2017.

Chapter 6

Convolutional Neural Network Assisted Seamless Multiview Video Streaming and Navigation

6.1 Introduction

Nowadays, many famous movie series are produced in 3D format. The prevalence of 3D videos is owing to its immersive vision, depth perception and interactive involvement. With this trend, a lot of efforts have been made to provide on-demand and live 3DTV services to home, making 3D available outside the cinema. Sky and Virgin in Britain, ESPN in America, and CCTV in China all provides 3D channels. Besides, the Olympic game also provides 3D broadcasting channels.

In light of this, many researches have been done in fields related to 3D video. Among them, the multiview content representation and coding is a vital topic, which needs to consider the compression efficiency, as well as the flexibility and interactivity. There are mainly three types of 3D video, namely stereoscopic video [112], multiview video [19] and free viewpoint video [113, 114]. Multiview video coding (MVC) is a popular format, which allows user to interactively switch view angles without the necessity to wear glasses. The MVC format stems from the multiview extension of the H.264 standard [115]. It exploits the statistical dependencies between spatially neighboring views to increase the compression efficiency. The multiview extention of the state-of-the-art high efficiency video coding (HEVC) standard (MV-HEVC) is also available now [116, 117]. MV-HEVC exploits the spatial redundancy among several views based on the usual block based compensation mechanism. Besides, it is backward compatible with any monoscopic decoder by simply extracting the sub-bitstream of the base view.

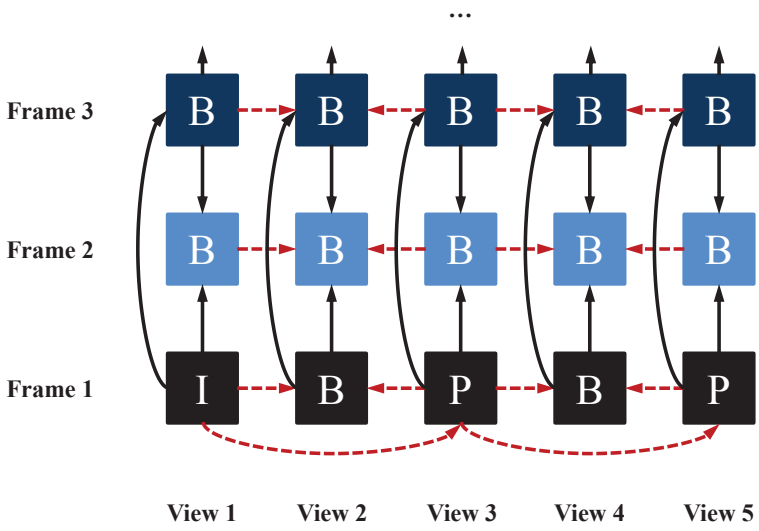


Figure 6.1: Illustration of temporal (black arrow) and inter-view (red dashed arrow) prediction in MVC.

However, the dependencies between the encoding of different views, as shown in Fig. 6.1, make the download of each views unseperatable [118]. For example, if view $2, 3, 4$ are required for frame 1, then all views $(1 - 5)$ must still be downloaded. This is because the decoding of view 3 depends on view 1, and view 4 depends on view 5. When it comes to the video with numerous views, most of the downloaded views are left unwatched. This is a huge waste, given the limited bandwidth resources. Another limitation of MVC is the predetermined bitrates of all views upon encoded, which forbids the flexibly rate adaptation to various bandwidth and device constraints. Meanwhile, bitrate of each view cannot be adaptively tuned based on its probability of being watched, unless versions with different bitrate combinations are prepared on the server. Simulcast encoding would be an option that allows selective downloading of views, since each view is independently encoded [119]. Although the stored data volume will be increased because of redundancies, the number of views that need to be transferred can be reduced, especially when there are large amounts of available views. However, redundancies between views would still increase the burden of storage and download. Thus, a multiview video representation that provides the flexibility of selecting views/bitrates and maintains a good compression efficiency is an important while non-trivial task.

To solve this problem, we propose to incorporate convolutional neural network (CNN) assisted quality enhancement model in the simulcast encoding framework, which exploits the similarities between views. Using the CNN, the low quality views can be

enhanced with the neighboring high quality views. Thus, multiview videos can be encoded with unequal qualities and then recovered/enhanced at the client side. By doing so, the redundancy between views is reduced and exploited. At the same time, each view is encoded independently with a plenty of quality levels, so as to provide multiple choices and combinations. In general, this method not only maintains the flexibility of selecting views/bitrates, but also reduce the total bits to be transmitted.

Downloading all views that might be watched is highly important for seamless view switching in an interactive multiview system. Therefore, the range of such views, along with their probabilities, needs to be predicted in advance. Similar to [120], a navigation model based on user behavior is proposed in this thesis to guide the download of views possibly to be watched. The probability of each view to be watched is deduced at the same time. Based on these information, the proposed bit allocation mechanism strives to minimize the overall distortion as an optimization problem. This bit allocation method is specially designed for videos encoded with the proposed multiview video representation method. It can adapt the overall bitrate to the varying bandwidth.

In conclusion, we propose a convolutional neural network assisted seamless multiview video streaming and navigation system. It not only provides a flexible view switching experience, but also restricts the overall bitrate to the limited bandwidth level. It can be widely incorporated in different multiview video streaming frameworks to enhance the overall quality with limited bitrate. The contributions of this work are summarized as follows:

1. A multiview video representation method is proposed in this thesis, which provides the flexibility of choosing views/bitrates while maintains a good compression efficiency. To the best of our knowledge, the idea of incorporating CNN assisted quality enhancement model into the multiview video representation method has never been investigated before. This model enhances the low quality lateral views with the high quality center view, so as to provide an enhanced overall quality without increasing the bitrate.

2. A bit allocation method is proposed, based on the probability of watching each view, which works closely with the proposed multiview video representation method. An optimized overall quality is achieved by this method.

3. Experimental results demonstrate the effectiveness of the proposed multiview

video streaming and navigation system, which enhances the overall quality by about 0.6 dB at low bitrate.

6.1.1 Related works

Multiview video streaming is an appealing while nontrivial task, since it not only needs to interact with the user but also needs to adapt to the network fluctuations. Moreover, the large volume of data represents another challenge here, compared to the single view video streaming. Thus, a multiview video representation scheme, which provides random accessibility and satisfies the bandwidth and storage constraints at the same time, is highly desired.

Some methods adapt the structure of inter-view predictions to provide the selective views to clients. Concepts like SP/SI frames [121] are employed in [122] for adaptive view switching. In [123, 124], user position is predicted to adapt the prediction structure among frames. In [120], the user navigation for the following frames is anticipated. Then, multiview sequence plus depth are prepared with their proposed coding scheme to permit a complete interactivity. While in [125], a shortest path algorithm for the optimal selection of reference views in multiview coding system is proposed to minimize both distortions of view reconstruction and coding rate cost. The main drawback of these methods is that they need an adaptive encoding mechanism that matches the network dynamism. Some other works [126, 127] try to combine distributed source coding and inter-view prediction for effective multiview switching. The limitation of these methods lies in the costly storage requirement, as well as the limited adaptivity due to the fixed prediction structure.

There are also works that propose new encoding methods to meet the challenge. A navigation domain representation for interactive multiview video is proposed in [118], which tries to provide high flexibility for interactive streaming while maintaining similar compression performance as classical inter-view coding. It divides the navigation domains into navigation segments, which contains a reference frame and some auxiliary information. These navigation segments could render any virtual view within the subdomain, thus providing some flexible navigation capacity. The work in [128] further extends the representation of navigation segment with layered depth image (LDI). However, the quality of the synthesized views is restricted because the disoccluded regions can not be perfectly filled.

The advantages of the proposed method, compared to the aforementioned works are: i) It does not restrict the interactivity and adaptivity of the multiview system, because it does not rely on the interview dependency. ii) It guarantees a flexible and satisfactory quality with the powerful CNN model. At the same time, bitrate is flexibly adapted to bandwidth constraint.

Artifacts Reduction Convolutional Neural Networks (AR-CNN) [129] is an effective tool for attenuation of various compression artifacts, including blocking artifacts, ringing effects and blurring. It leads to a better result than the traditional deblocking oriented and restoration based methods. Besides the satisfactory performance, AR-CNN provides an end-to-end mapping between low and high quality images with a unified network, which makes it easy to be incorporated into other tasks as a unit. In our previous work [130], a super-resolution method using CNN model is proposed for stereo video with mixed resolution. The low resolution image is up-sampled with the high resolution image. Inspired by these two works, we expect that a convolutional neural network would also be helpful in combining related information in the mixed-quality multiview system. In this thesis, a 4-layer CNN model that directly takes two inputs (projected view from higher quality view and low quality view) and outputs an enhanced view is proposed as one important part of the work.

In the following section, our proposed method will be introduced within the DASH framework. The proposed method under the scenario of DASH is described in detail in Section 6.2. After that, experiments and discussions are presented in Section 6.3 to show the effectiveness of the proposed method. Finally, conclusions are provided in Section 6.4.

6.2 Proposed Method

In this thesis, a multiview video streaming and navigation system which allows real-time and seamless view switching is proposed. The system mainly consists of three parts: navigation model, CNN assisted quality enhancement model and bit allocation mechanism. These three parts can be incorporated into any multiview streaming frameworks to provide an enhanced overall quality. With CNN assisted quality enhancement model, a novel multiview video representation method is proposed to ensure both flexibility and compression efficiency. The proposed bit allocation mechanism tunes the quality of each view to reduce the overall distortion for the views that are likely to be

watched, according to a predicted navigation model.

The view with the highest probability to be requested is called **center view**, while the others are called **lateral views**. Each view is encoded separately to ensure the freedom of selecting any subset of views to download. In this work, only the center view is requested with high quality. While lateral views are downloaded with low quality levels and then enhanced to the desired quality levels with CNN model at the client side.

We exploit DASH as the streaming system. Firstly, an overview of the proposed method within DASH will be presented, along with the definition of the notations. Then, detailed descriptions of navigation model, CNN assisted quality enhancement model and bit allocation mechanism will be provided, respectively.

6.2.1 Solution Overview under DASH

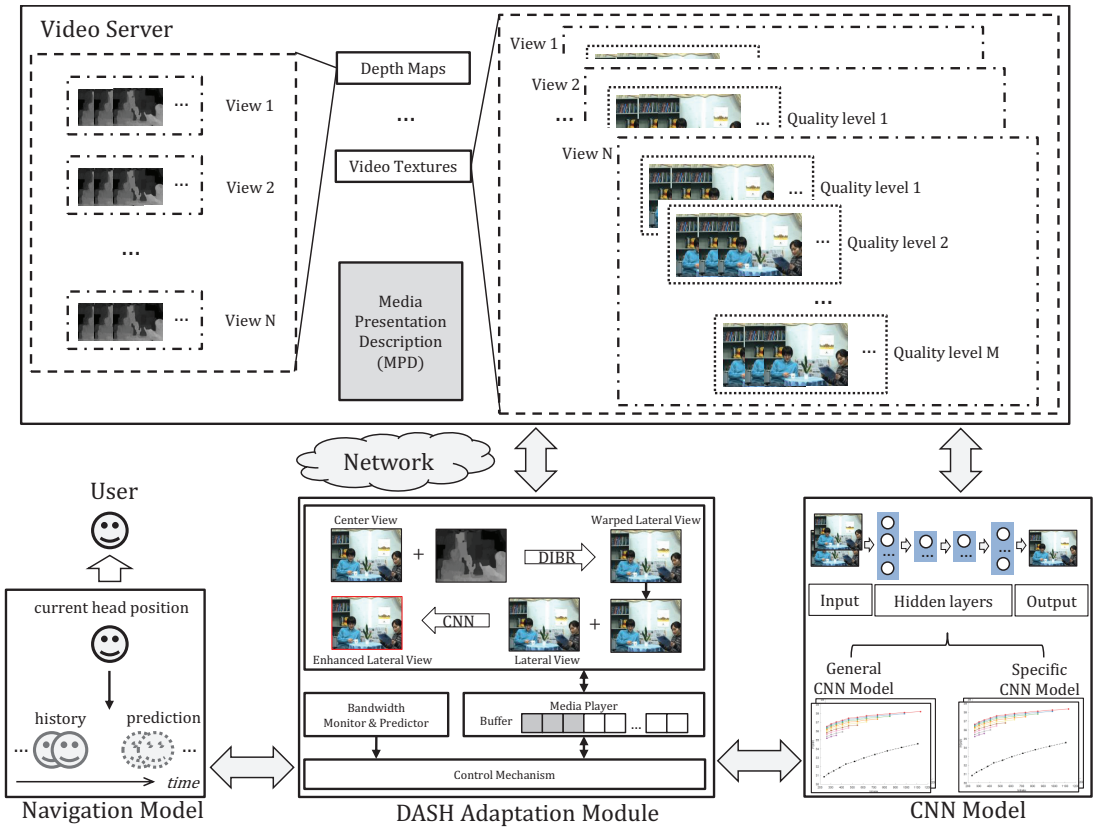


Figure 6.2: Diagram of the proposed multiview video streaming solution under the DASH framework.

The diagram of the proposed method within DASH framework is illustrated in Fig.

Table 6.1: Descriptions of key symbols

Symbol	Definition
τ	duration of one segment
K	total number of timely non-overlapping segments for one video
k	index of segments whose content is within period $[(k-1)\times\tau, k\times\tau), 1 \le k \le K$
M	total number of available quality levels
N	total number of available views
t_k	decision point of k^{th} segment
v_i	the i^{th} view among all the available views, $1 \le i \le N$
q_j	the j^{th} quality level among all available representations, $1 \le j \le M$ (larger j means better quality level)
T_k	the length of buffer reservation in time domain at decision point t_k
$B_k^{'}$	the predicted bandwidth when downloading ϕ_k
$V(t_k)$	the view angle of user at decision point t_k
$R_k(v_i, q_j)$	the bitrate of texture segment $S_k(v_i, q_j)$
$R_k^d(v_i)$	the bitrate of depth segment $D_k(v_i)$
$d_k(v_i)$	the k^{th} depth map of view v_i
$S_k(v_i, q_j)$	the k^{th} texture segment of view v_i and quality level q_j
ϕ_k	set of k^{th} segments requested by client, including n texture segments and one depth map of the center view (v_c): $\phi_k = \{S_k(v_i, q_j)\}_n \cup D_k(v_c)$
$\Re_k$	the estimated available bits for downloading textures in ϕ_k, i.e. $\{S_k(v_i, q_j)\}_n$

6.2, which consists of HTTP server, navigation model, DASH adaptation module and CNN model. The last three parts all belongs to the DASH client. The proposed bit allocation mechanism is incorporated in the DASH adaptation module.

In the following, the data preparation on the HTTP sever, the details of DASH adaptation module and a brief work flow will be described. Important notations and corresponding definitions are listed in Table 6.1.

HTTP Server

Suppose there are N views $\{v_1, v_2, \ldots, v_N\}$ provided for one video. The video of a certain view is divided into K segments with same duration τ. For each segment, totally M representations $\{q_1, q_2, \ldots, q_M\}$ are available. Thus, the k^{th} segment of view v_i with quality representation q_j is denoted as $\boldsymbol{S_k(v_i, q_j)}$. There are totally $K \times N \times M$ texture

video segments for one video movie, that are independently encoded into separate files.

Besides the texture, the depth maps are also prepared as another adaptation set. They are used to generate the virtual views in the CNN model. Different from texture, only one representation that ensures stable warping performance is prepared. Thus, the k^{th} depth segment of view v_i is denoted as $\boldsymbol{d_k(v_i)}$.

DASH Adaptation Module

This module tries to provide the best viewing experience under bandwidth constraints. To achieve this goal, it has multiple duties. First, it is responsible for coordinating the navigation model and CNN model, as well as the external communication with the server. Second, it is responsible for displaying the video, which includes parsing received data from the server, choosing the view to display according to user's current view angle and enhancing the lateral view with CNN model if view angle is switched. Third, the DASH client is responsible for monitoring and predicting the bandwidth status, as well as the buffer status. Last but not the least, it is responsible for the adaptation intelligence, which decides which data to request, based on the bandwidth and user's navigation mode.

Workflow in Brief

The DASH client operations are described as follows. After obtaining the MPD file from video server, the download of multiview videos starts. Let us suppose that at decision point t_k, the client is going to request the desired segment set ϕ_k ($1 \leq k \leq K$), which contains n views that might be watched and one depth map of the current center view v_c, which can be represented as

$$\phi_k = \{S_k(v_i, q_j)\}_n \cup d_k(v_c). \tag{6.1}$$

The detailed workflow is as follows:

1. The current head position of user $V(t_k)$ is tracked by the navigation model. Correspondingly, the current center view will be chosen: $v_c = V(t_k)$.

2. The n views that are possible candidates to be watched, $\{v_i\}_n$, are predicted along with their probabilities, by the navigation model in Section 6.2.2.

3. The future bandwidth $B_k^{'}$ for downloading ϕ_k is predicted in the DASH adaptation module according to the monitored history. Then, the estimated available bits

$\Re_k$ is deduced:

$$\Re_k = B_k^{'} \times \tau. \tag{6.2}$$

4. Then, $\Re_k$, after subtracting the bits of depth map, is allocated among the n views according to the proposed bit allocation mechanism in Section 6.2.4.

5. The bit allocation proposal, which contains the representation levels for each desired view, would be encapsuled into the request and sent to the server.

6. When user switches from the center view to a lateral view, the low quality lateral view is enhanced by the CNN model in Section 6.2.3 before being displayed.

This procedure is iterated for each segment.

6.2.2 Navigation Model

Tracking and predicting user's head position are vital parts of a multiview video system, which enables the interactivity between the user and the system. This function can be achieved by a navigation model. This model tracks the motion of user's head and predicts future possible view angles. Besides, the prediction information helps the system to prefetch all views that might be watched in the future, guaranteeing real-time view switching. Suppose at decision point t_k, the user's view angle is $V(t_k)$. The navigation model employed in this thesis is based on the following assumptions:

1. One state assumption: The current view angle $V(t_k)$ is only affected by the previous view angle $V(t_{k-1})$.

2. Smooth view switch assumption: A large view switch step is not allowed. That is, the user can at most switch to neighboring view in one step, i.e.

$$V(t_k) \in \{v_{i-1}, v_i, v_{i+1}\}, \text{ if } V(t_{k-1}) = v_i. \tag{6.3}$$

Based on these two assumptions, the navigation model can be illustrated using Fig. 6.3. Each block in the figure denotes one possible view angle at that decision point. For example, the view angle at t_k is v_i. At the next decision point t^{k+1}, there are three view angles $\{v_{i-1}, v_i, v_{i+1}\}$ possibly to be switched to. The probability of remaining in the same view angle is p, while that of switching to one of the two neighboring view

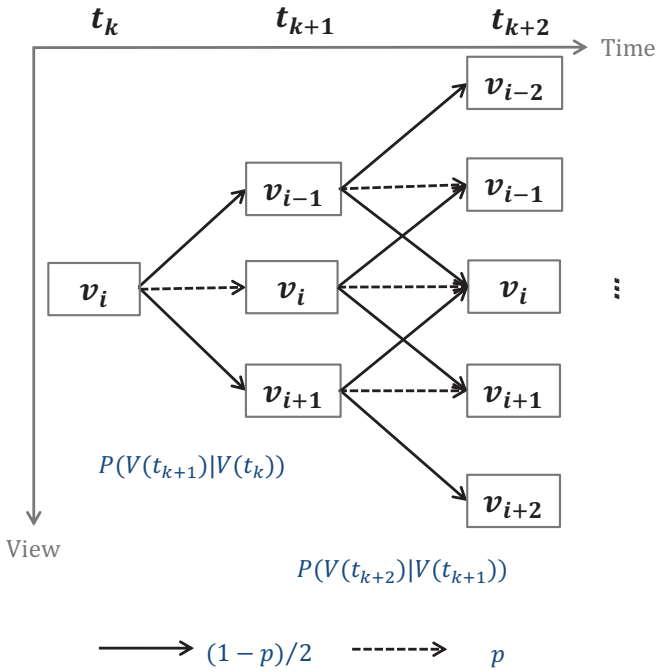


Figure 6.3: Graphical representation of user navigation model, where different transition probabilities are represented by different arrow types.

angles are equally $(1-p)/2$.

$$P(v_i|v_i) \quad = p; \tag{6.4}$$

$$P(v_i|v_{i-1}) = P(v_i|v_{i+1}) = \frac{1-p}{2}. \tag{6.5}$$

When it comes to the scenario of prefetching future segments, such as prefetching $(k+l)^{th}$ segment when k^{th} segment is being displayed, the nearest known view angle for the moment is $V(t_k)$. Thus, depending on different values of l, views possibly to be watched for $(k+l)^{th}$ segment would vary, together with the corresponding probabilities. Nevertheless, the final probability is a multiplication of all the probabilities along all possible view switching paths. Thus, once the probability of each step is known, the final probability is easy to estimate.

$$P_{k+l}(v_{i+j}) = \begin{cases} P_{k+l-1}(v_i) \times p + g(1) \times P_{k+l-1}(v_{i-1}) \times (1-p) & , j = 0; \\ g(0) \times P_{k+l-1}(v_{i+j-1}) \times \frac{(1-p)}{2} + g(1) \times P_{k+l-1}(v_{i+j}) \times p + g(2) \times P_{k+l-1}(v_{i+j+1}) \times \frac{(1-p)}{2} & , j \in (0, l]; \\ P_{k+l}(v_{i-j}) & , j \in [-l, 0). \end{cases} \tag{6.6}$$

Supposing at t_k, the probability of watching the view angle $V(t_k) = v_i$ is $P_k(v_i)$. Then, the probability of watching view v_{i+j} at $t_{k+l}, (l \geq 1)$ is calculated recursively as

$P_{k+l}(v_{i+j})$ in Eq. (6.6), where $g(x)$ is calculated as follows:

$$g(x) = \max(0, \min(1, l - j - x + 1)), x \geq 0. \tag{6.7}$$

$P_{k+l}(v_{i+j})$ is classified into three cases in Eq. (6.6): $j = 0$ represents the view angle same as $V(t_k)$; $j \in (0, l]$ denotes the view angles on the right of $V(t_k)$; $j \in [-l, 0)$ are the view angles on the left of $V(t_k)$. The probability calculation for $j \in [-l, 0)$ is identical to the one for $j \in (0, l]$. It can be observed in Fig. 6.3 that different numbers of views in the previous decision point are switching to the current view angle, based on the different l and j. Thus, $g(x)$ is employed to decide the number of views in the previous decision point that will switch to the current view angle based on l and j. Then, the probability can be calculated recursively with $g(x)$ according to Eq. (6.6).

At the end, the desired views along with their probabilities of being watched, together with the center view index v_c would be sent to DASH adaptation module for further operations.

It is worth to note that the navigation model is an independent module in the whole system, which can be substituted with more sophisticated ones. As the navigation model is not the major topic in our work, the above described model is used in the experiments.

6.2.3 CNN Model

The convolutional neural network model aims at improving the overall quality without increasing the total bitrate. This is achieved by migrating the bit budget from the lateral views to the center view. With more bit budget, the quality of center view can be increased. Meanwhile, the quality of lateral views would be maintained with equivalent level after CNN based quality enhancement. Thus, the overall quality will be improved. The details of the CNN is described as follows.

The CNN model enhances the low-quality lateral view with the high-quality center view by exploring the similarities between them. Inspired by AR-CNN [129] and our previous work [130], CNN network architecture used in this work is shown in Fig. 6.4. The inputs include the low quality lateral view and the virtual image warped from the high quality center view. The output is the image with enhanced quality. It is worth to notice that the CNN model is trained with a subset of all the sequences. Nevertheless, the model can be used for both sequences inside or outside the training

set. In the following, the pre-processing of input data, CNN network structure and training details are introduced.

1. Pre-processing, which prepares the data to be input into CNN, consists of two steps: one is 3D warping and another is batch cropping. For 3D warping, high quality center view and its depth map are used to construct a 3D image, and then generate the virtual view in the position of lateral view. The DIBR technique [22] is used to warp the pixels with inpainting process skipped. When changing the viewing point, some holes might appear due to some occluded regions becoming visible at the new viewing point. Next, input images are cropped into small sub-images (i.e. 33×33), including low quality lateral image (Y_L), DIBR warped image (Y_V) and the ground truth images. This process helps to speed up the training process. Sub-images at same position are gathered into a group. Then, a mapping between the two images and ground truth are learned in the CNN.

2. The CNN for quality enhancement consists of four layers, as shown in Fig. 6.4. Specifically, the first layer performs image fusion and feature extraction, which fuses the two inputs, i.e. Y_L and Y_V, and extracts feature vectors from them. The second layer is responsible for feature enhancement, which would remove noises from the feature vectors. Then, the non-linear mapping layer maps the low quality patches into high quality patches. Finally, the reconstruction layer merges the obtained patches to deliver the final output. The network can be represented as:

$$F_1(Y) = max(0, W_{11} * Y_L + W_{12} * Y_V + B_1); \tag{6.8}$$

$$F_i(Y) = max(0, W_i * F_{i-1}(Y) + B_i), i = 2, 3; \tag{6.9}$$

$$F(Y) = W_4 * F_3(Y) + B_4. \tag{6.10}$$

   Where the first and last layers are represented as Eq. (6.8) and Eq. (6.10), while the second and third layers share the Eq. (6.9). W_i and B_i denotes the filters and biases of the i^{th} layer, and '*' denotes the convolutional operation. Rectified Linear Unit (ReLU, $max(0, x)$) is applied on the filter responses. F represents the output feature maps. The kernel sizes are 9×9, 7×7, 1×1 and 5×5 for each layer respectively. Further details can be found in the AR-CNN paper [129].

3. Training is conducted to learn the filters and biases. Given a training dataset

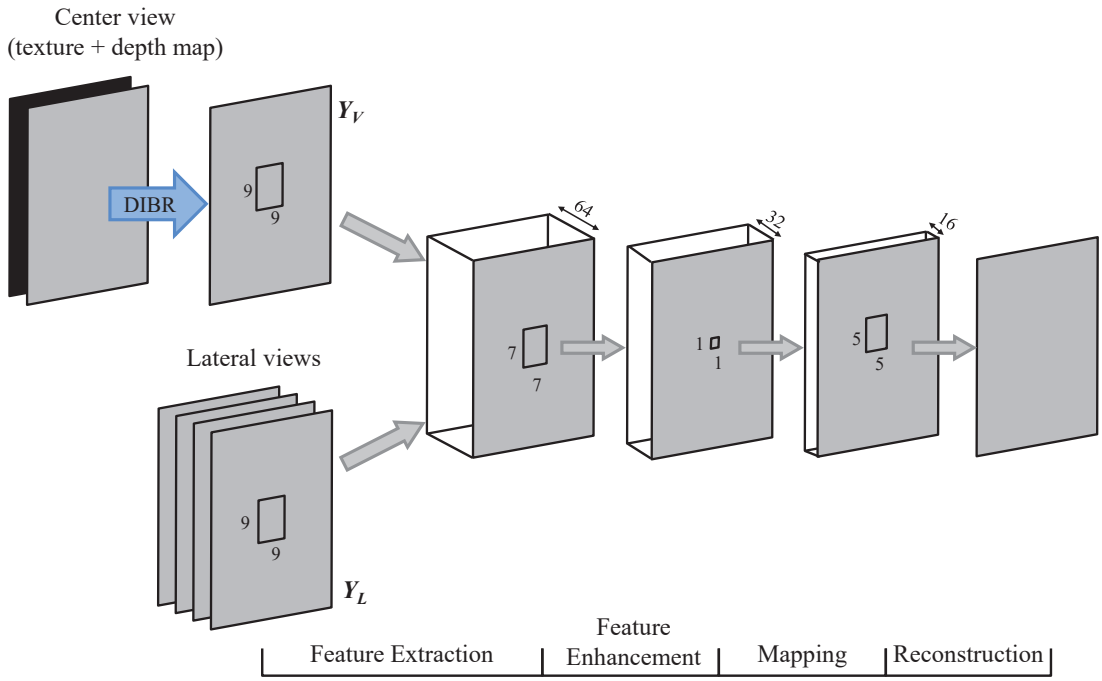


Figure 6.4: CNN network structure with 4 convolutional layers.

$\{L_i, V_i, Y_i\}$, we use Mean Squared Error (MSE) as the loss function. L and V represents the input data: the low quality image and the virtual image warped from the high quality view. While Y denotes the ground truth image. The loss function can be expressed as:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \|f(L_i, V_i, \theta) - Y_i\|^2 , \tag{6.11}$$

where θ is the parameters of the network, including filters and biases. Stochastic gradient decent (SGD) is used to minimize the loss function with the standard backpropagation. At the beginning, the weights of the network are initialized with values of random gaussian distribution with zero mean. The learning rate is set as 0.0001.

6.2.4 Bit Allocation Mechanism

As one part of the DASH client, the bit allocation mechanism is introduced in this section. The goal of bit allocation is to minimize the expected distortion of the center

view and its associated lateral views that might be watched:

$$\bar{D}_k = \sum_{1 \le i \le n} D_k(v_i, q_j) \times P_k(v_i),$$
$$\underline{\text{s.t.}} \sum_{1 \le i \le n} R_k(v_i, q_j) \le \Re_k. \tag{6.12}$$

The constrained optimization problem can be converted into an unconstrained optimization problem by Lagrange multiplier method as follows:

$$J = \bar{D}_k + \lambda \sum_{1 \le i \le n} R_k(v_i, q_j); \tag{6.13}$$

where λ is the Lagrange multiplier. To minimize J, the derivative with respect to bitrate of each view is set to zero. For a given view $v_i (1 \le i \le n)$,

$$\frac{\partial J}{\partial R_k(v_i, q_j)} = P_k(v_i) \times \frac{\partial D_k(v_i, q_j)}{\partial R_k(v_i, q_j)} + \lambda = 0. \tag{6.14}$$

Thus, for any two view v_i and $v_{i'}$,

$$P_k(v_i) \times \frac{\partial D_k(v_i, q_j)}{\partial R_k(v_i, q_j)} = P_k(v_{i'}) \times \frac{\partial D_k(v_{i'}, q_{j'})}{\partial R_k(v_{i'}, q_{j'})}.$$

That is,

$$\frac{P_k(v_{i'})}{P_k(v_i)} = \frac{\partial D_k(v_i, q_j)}{\partial R_k(v_i, q_j)} \Big/ \frac{\partial D_k(v_{i'}, q_{j'})}{\partial R_k(v_{i'}, q_{j'})}. \tag{6.15}$$

Based on the above analysis, the bit allocation problem can be solved in two steps: initial bit allocation and fine tuning based on CNN enhancement model. The former one allocates the bitrate without considering the CNN enhancement model. This initial bit allocation result helps to narrow down the search range for the next step. While the fine tuning step takes the influence of CNN quality enhancement model into consideration, and refines the result of the initial bit allocation. The two steps are described in details as follows.

Initial Bit Allocation

In this step, CNN quality enhancement model is not taken into consideration. It directly works on the results of the codec. In this work, High efficiency video coding (HEVC) [131] is used as the codec. Each quality level q_j maps to one QP value without overlaping, i.e. $q_j = F(QP_j)$. As defined in HEVC,

$$\lambda_{HEVC}(q_j) = \frac{\partial D}{\partial R} = c * 2^{\frac{QP_j - 12}{3}}, \tag{6.16}$$

where c is a parameter related to the coding structure. Since the two frames are in the same position, i.e. i^{th} frame, in a coding structure, the values of c would be the same. Thus, Eq. (6.15) can be represented as

$$\begin{aligned} P_k(v_{i'})/P_k(v_i) &= \lambda_{HEVC}(q_j)/\lambda_{HEVC}(q_{j'}) \\ &= 2^{\frac{QP_j-12}{3}}/2^{\frac{QP_{j'}-12}{3}} \\ &= 2^{\frac{QP_j-QP_{j'}}{3}}, \end{aligned} \tag{6.17}$$

where each q_j maps to an unique QP_j, i.e. $q_j = F(QP_j)$. Thus, the QP difference between any two view, v_i and $v_{i'}$, is:

$$\Delta QP(v_i, v_{i'}) = QP_j - QP_{j'} = 3log_2\frac{P_k(v_{i'})}{P_k(v_i)}. \tag{6.18}$$

This method can also be used for other codec, provided that the Eq. (6.16) is substituted accordingly. Based on Eq. (6.18), QP differences between center view and all other desired views can be obtained. Thus, the QP value of any desired view can be represented by the QP of center view (QP_c) as below.

$$QP_j = QP_c + \Delta QP(v_i, v_c).$$

Then, the bit allocation problem is converted into finding the minimum value of (QP_c) under the constraint

$$\sum_{1\leq i\leq n} R_k(v_i, F(QP_c + \Delta QP(v_i, v_c))) + R_k^d(v_c) \leq \Re_k, \tag{6.19}$$

which can be solved by an iterative search method. Till now, the optimal bit allocation without CNN enhancement is obtained.

Fine Tuning based on CNN Enhancement Model

This step tries to incorporate the influence of CNN quality enhancement into the bit allocation process. The influence mainly comes from the enhanced rate distortion (RD) relationship for lateral views. With enhanced quality of lateral views, the bits can be saved from them and dedicated to the center view. By doing so, the quality of center view is improved. Meanwhile, the quality of lateral views will also be improved, as the quality of virtual view is increased. Finally, the overall quality is enhanced. However, the tuning is not straightforward since it also needs to satisfy Eq. (6.15), which

represents the overall optimization goal. Before introducing the detailed algorithm, an analysis of the enhanced RD relationship is presented.

With CNN quality enhancement step, Eq. (6.16) is not valid for lateral views. Instead, the slope of the tangent line in the RD curve is employed. This is based on the assumption that the RD curve is regarded as linear within a local range. Each lateral view has its own RD curve, which consists of rate distortion values of different QPs. Taking QP_i as an example,

$$\frac{\partial D_i^e}{\partial R_i} = \frac{D_{i+1}^e - D_{i-1}^e}{R_{i+1} - R_{i-1}}, \tag{6.20}$$

where D_{i+1}^e and D_{i-1}^e are distortions of the CNN enhanced images with QP_{i+1} and QP_{i-1}. Correspondingly, Eq. (6.20) is used in the following fine tuning algorithm.

Algorithm 1 Fine tuning of bit allocation proposal.

Input:

Total available bits $\Re$;
The initial bit allocation proposal, $QP_c, \{QP_l\}$;
The range of available representations, $[QP_{min}, QP_{max}]$;
RD curve of HEVC encoded sequence, RD_o;
RD curve of CNN enhanced sequence, RD_e;

Output:

The fine tuned bit allocation proposal, $QP_c^e, \{QP_l^e\}$;

Initialization: $QP_c^e = QP_c$; $\{QP_l^e\} = \{QP_l\}$; $\Delta R^e = 0$;
repeat
  $QP_c^{tmp} = QP_c^e - 1$;
  Calculate $\partial D_c / \partial R_c$ of QP_c^{tmp} in RD_o;
  for all $\{QP_l\}$ **do**
    Calculate $\partial D_l / \partial R_l$ with $\partial D_c / \partial R_c$ according to (6.15);
    Searching for the QP_l^{tmp} within $[QP_{min}, QP_{max}]$ in RD_e, whose has nearest value as $\partial D_l / \partial R_l$;
  end for
  $\Delta R = \sum R(v_i, F(QP_i^{tmp}) + R^d(v_c) - \Re$;
  if $\Delta R < \Delta R^e$ **then**
    Update $QP_c^e, \{QP_l^e\}$ with $QP_c^{tmp}, \{QP_l^{tmp}\}$;
    $\Delta R^e = \Delta R$;
  end if
until $QP_c^{tmp} == QP_{min}$
if $\Delta R^e \leq 0$ **then**
  Allocate ΔR^e among lateral views according to the quality gain per bit, and update $\{QP_l^e\}$ accordingly;
end if
return $QP_c^e, \{QP_l^e\}$;

The detailed steps are shown in Algorithm 1. It starts from the initial bit allocation result and decreases the QP of center view (QP_c) step by step until reaching the avail-

able minimum QP (QP_{min}). For each QP_c, the corresponding QPs for lateral views ($\{QP_l\}$) are determined following Eq. (6.15). Then, the delta bits (ΔR) is calculated between actual used bits and total available bits ($\Re$). If $\Delta R < 0$, this means the actual used bits is within the available bit budget. Thus, this fine tuned proposal is regarded as feasible. This process will be iterated for all QP_c, then the one which maximizes the used bitrate while satisfying Eq. (6.19) will be used.

After these steps, the available bits $\Re$ usually can not be exploited fully. The remaining bits ΔR are not enough to support the quality upgrade for the center view. However, they are usually enough to upgrade the quality of lateral views. Because the bitrate due to one QP step for large QP values are much lower than those for small QP values. Thus, steps $15 - 17$ in Algorithm 1 allocate the remaining bits among lateral views according to their quality gain per bit. The gain of this proposed process will be presented in the experiments.

6.3 Experimental Results

Multiview video sequences, Kendo, Balloons, Undodancer and GT-fly [132], are used in the experiment. Each view is independently encoded using HM 16.0 [133] with different QPs (ranging from 22 to 51) to represent different quality levels. Depth maps are encoded with $QP = 50$. For Undodancer and GT-fly, view $1 - 5$ are used for this experiment, with absent view 4 rendered from view 3 and view 5. While for Kendo and Balloons, a dense view set (1 cm) is used for this experiment, to cover different view densities. With view 3 in the center, totally 5 views are used and named as view $1 - 5$ for simplicity. The duration of each segment in DASH is set as 2 seconds.

In the following, the performance of the proposed method is demonstrated under the scenario of DASH. As for DASH, there are a plenty of rate adaptation mechanisms, which is not the focus of this thesis. Thus, the experiments are evaluated at the segment set level to make it simple. While in the practical DASH system, the proposed method can be applied to enhance the performance of each segment set, no matter what rate adaptation mechanism is used. The experiments are arranged as follows: firstly, the effectiveness of CNN based quality enhancement model is examined to demonstrate the feasibility of the proposed multiview representation method. Next, the proposed bit allocation method for each segment set is assessed with comparison to benchmark. The benchmark here represents the initial bit allocation mechanism as described in

Section 5.1-D. The above experiments are applied on the luminance channel (Y channel in YUV color space), and the performance, i.e. PSNR, is evaluated in Y channel.

6.3.1 CNN Assisted Quality Enhancement Model

The goal is to enhance the quality of the lateral view Y_L through the proposed CNN model, with Y_L and the virtual view Y_V as the input. Y_V is warped from the center view Y_C. The proposed CNN model is capable of achieving this goal, because of the following reasons. Firstly, the similarities between the two input images can be explored by the CNN network. With this process, the detail information in Y_C can be learned and used to enhance Y_L. Secondly, the proposed CNN network is inspired by the AR-CNN work, which is developed for compression artifacts reduction (equivalent to the quality enhancement). Nevertheless, the enhancement task is non-trivial, since there are plenty of dynamic scenarios. It is affected by many factors, including the quality of Y_L, Y_C and their difference, the direction and distance (i.e. the baseline between the views) from Y_C to Y_L, as well as the contents of the video. Thus, the proposed CNN model is expected to work stably and effectively for all these scenarios.

The performance of the CNN model is influenced by the training set. With a more dynamic data set which incorporates more scenarios, the corresponding CNN model would guarantee a more generalized enhancement result over different scenarios. While with more specific data set, the CNN model would provide a more effective performance on that specific data. The former one is named as **general CNN model**, while the latter one is called **sequence-specific CNN model**. The general CNN model is trained with multiple sequences, which have different contents and baselines. While the sequence-specific model is trained with only one specific sequence, which is used only for this sequence.

General CNN Model

A general CNN model is trained with multiple video sequences, as well as different QP values and different distances between lateral and center views. Two sequences, Kendo and Undodancer (first 15 frames each), are used in the training set. Two distances are included, namely single and double baselines. The QPs for Y_L are 30 and 42, while the QP of Y_C is set as 20. The QP of depth map is fixed as 50. It is shown by experiments that the QP of depth map does not influence the enhancement result a lot (less than

0.1 dB). The model is trained over 1.5 million iterations.

The performance of the general CNN model, compared to the benchmark, on different sequences is shown in Fig. 6.5. Frames from 171 to 200 of each sequence are used in the testing set. Different curves are plotted with different QP for Y_C, ranging from 22 to 30. While for the QP of Y_L, its range is deduced from the proposed bit allocation mechanism. Each dot in the figure represents the averaged result of one QP setting over 30 frames. PSNR of Y_L without enhancement is plotted as benchmark for comparison. The observations are as follows:

1. The enhancement gain of general CNN model over benchmark are huge, with roughly 5 dB at most. With the increase of center view quality, the enhancement gain increases. This demonstrates that the general model works well for both camera recorded videos and computer generated videos.

2. The general model is effective for scenarios with different distances and directions between the views. Three scenarios are tested for different sequences in each row in Fig. 6.5. This demonstrates the stability of the general CNN model.

3. Although the video contents of Balloons and GTfly are completely different from contents of the training set, the results are still promising. This demonstrates the generality of the general CNN model.

Sequence-specific CNN Model

The sequence-specific CNN model is trained with specific sequence to obtain effectiveness for that specific data. As shown in Fig. 6.6, sequence-specific CNN models for Kendo and Undodancer are compared to general CNN model. The two models are trained with only one sequence, i.e. Kendo and Undodancer respectively. As expected, the sequence specific models obtain better results than the general models. It is worth to notice that, the gain for Undodancer is more uniform for all bitrate levels than that of Kendo. This is because Undodancer is computer generated sequence with high quality depth maps. While for Kendo, the depth maps are worse, which would affect the quality of the warped images. Thus, when the quality of lateral view is high, which has an almost similar quality to the warped view, the enhancement margin for Kendo becomes very limited. Thus, less gain is obtained with the increase of bitrate for Kendo. Another interesting observation in the result of Kendo is that, the gain of

Table 6.2: Comparison of bit allocation result for view $1-5$ between proposed method and benchmark. For the proposed method, both general and sequence-specific CNN model are tested, with the later one in bold.

sequence	benchmark							Proposed (Step 1-14)							$\overline{PSNR}$ gain (dB)
	QP assignment					Total bits	$\overline{PSNR}$ (dB)	QP assignment					Total bits	$\overline{PSNR}$ (dB)	
	v1	v2	v3	v4	v5			v1	v2	v3	v4	v5			
Kendo	51	37	29	37	51	1909	40.95	51	43	28	42	51	1743	41.58	0.63
								51	**45**	**28**	**44**	**51**	**1661**	**41.57**	**0.62**
	48	33	26	33	48	3094	42.60	51	40	25	38	51	2697	43.03	0.43
								51	**40**	**24**	**39**	**50**	**3051**	**43.49**	**0.89**
	46	32	25	32	46	3649	43.13	50	39	24	37	50	3170	43.52	0.39
								51	**39**	**23**	**38**	**50**	**3475**	**43.85**	**0.72**
Undo-dancer	51	38	30	38	51	4912	36.12	51	47	29	46	51	4310	36.89	0.77
								51	**46**	**29**	**46**	**51**	**4347**	**36.92**	**0.80**
	49	34	27	34	49	8520	38.07	51	43	26	42	50	7388	38.88	0.81
								51	**43**	**26**	**42**	**51**	**7370**	**38.90**	**0.83**
	46	32	25	32	46	12441	39.48	51	41	23	40	50	12317	40.89	1.41
								51	**41**	**23**	**40**	**51**	**12299**	**40.92**	**1.44**
Balloons	51	38	30	38	51	1467	40.65	51	44	28	44	51	1361	41.15	0.50
	51	35	28	35	51	1904	41.66	51	40	26	40	51	1831	41.87	0.21
	47	33	26	33	47	2532	42.50	51	38	24	38	51	2457	42.60	0.10
GTfly	51	38	30	38	51	1932	38.93	51	45	29	45	51	1706	39.56	0.63
	47	33	26	33	47	4161	40.65	51	40	25	40	51	3873	41.22	0.57
	46	32	25	32	46	5133	41.10	51	39	24	39	51	4856	41.65	0.55
Average						4305	40.49						3976	41.07	0.58
						5754	**40.06**						**5367**	**40.94**	**0.88**

sequence-specific model in the low bitrate region is very obvious. This may be owing to that the sequence-specific model trained with Kendo sequence has the function of fixing the errors in the depth map.

6.3.2 Navigation Guided Bit Allocation Mechanism

Overall PSNR of requested multiview video is accessed in this section to demonstrate the effectiveness of the proposed bit allocation mechanism. The overall PSNR gain comes from two parts: i) quality enhancement of CNN model; ii) exploitation of remaining available bits ΔR. These two parts correspond to Step $1-14$ and Step $15-17$ in Algorithm 1. They will be assessed respectively in the following.

Experiment Setup

Two navigation scenarios are tested, with 3 and 5 views as view switching range. That is, $\{v_{i-1}, v_i, v_{i+1}\}$ and $\{v_{i-2}, v_{i-1}, v_i, v_{i+1}, v_{i+2}\}$ respectively, as shown in Fig. 6.3. The probability of remaining in the current position (p) is set as 0.9, for brevity. For Kendo

Table 6.3: Comparison of bit allocation result for view $2-4$ between proposed method and benchmark. For the proposed method, both general and sequence-specific CNN model are tested, with the later one in bold.

sequence	benchmark					Proposed (Step 1-14)					$\overline{PSNR}$ gain (dB)
	QP assignment			Total bits	$\overline{PSNR}$ (dB)	QP assignment			Total bits	$\overline{PSNR}$ (dB)	
	v2	v3	v4			v2	v3	v4			
Kendo	41	30	41	1364	40.68	46	29	46	1293	41.36	0.68
						48	**29**	**48**	**1235**	**41.36**	**0.68**
	37	27	37	2139	42.31	43	26	42	2050	42.88	0.57
						45	**26**	**44**	**1968**	**42.88**	**0.57**
	33	24	33	3480	43.88	40	23	39	3256	44.28	0.40
						41	**23**	**40**	**3197**	**44.27**	**0.39**
Undo-dancer	42	31	42	3435	35.69	49	30	49	3351	36.44	0.75
						49	**30**	**48**	**3378**	**36.46**	**0.77**
	37	27	37	7091	38.24	45	26	45	6885	39.08	0.84
						45	**26**	**45**	**6885**	**39.09**	**0.85**
	34	25	34	10773	39.72	43	24	42	10549	40.58	0.86
						43	**24**	**43**	**10488**	**40.58**	**0.86**
Balloons	43	32	43	931	39.93	49	30	49	893	40.87	0.94
	40	29	40	1295	41.40	47	28	47	1127	41.72	0.32
	37	27	37	1734	42.32	44	26	44	1530	42.47	0.15
GTfly	41	31	41	1419	38.72	48	30	48	1310	39.30	0.58
	36	27	36	2968	40.39	43	26	43	2920	40.94	0.55
	35	26	35	3630	40.82	35	26	35	3630	41.03	0.21
Average				3355	40.34				3233	40.91	0.57
				4714	**40.08**				**4525**	**40.77**	**0.69**

Table 6.4: The PSNR gain of lateral views obtained with final process (step $12-14$ in Algorithm 1) for 5 views scenario of Undodancer.

sequence	Bandwidth (bits)	Step 1-14							Step 1-17							$\overline{PSNR}$ Gain of lateral views (dB)
		v1	v2	v3	v4	v5	Total bits	$\overline{PSNR}$ of lateral views (dB)	v1	v2	v3	v4	v5	Total bits	$\overline{PSNR}$ of lateral views (dB)	
Undo-dancer	4912	51	47	29	46	51	4310	35.16	50	43	29	44	51	4597	35.67	0.51
		51	**46**	**29**	**46**	**51**	**4347**	**35.32**	**49**	**44**	**29**	**43**	**49**	**4658**	**35.77**	**0.45**
	8520	51	43	26	42	50	7388	36.54	48	40	26	41	49	7775	36.82	0.28
		51	**43**	**26**	**42**	**51**	**7370**	**36.64**	**49**	**40**	**26**	**41**	**48**	**7774**	**36.92**	**0.28**
	12441	51	41	23	40	50	12317	37.37	51	40	23	40	50	12402	37.45	0.08
		51	**41**	**23**	**40**	**51**	**12299**	**37.51**	**50**	**40**	**23**	**40**	**51**	**12402**	**37.58**	**0.07**
Average	8624						8005 *(93%)*	36.36						8258 *(96%)*	36.65	0.29
							8005 *(93%)*	**36.49**						**8278** *(96%)*	**36.76**	**0.27**

and Undodancer, both general and sequence-specific CNN model are tested. While for Balloons and GTfly, the general CNN model is assessed.

Supposing view $1-n$ are requested, the overall PSNR is calculated as an weighted average as follows:

$$\overline{PSNR} = \sum_{1 \leq i \leq n} PSNR(v_i, q_j) \times P(v_i),$$

where $P(v_i)$ is obtained according to Eq. (6.6). $PSNR(v_i, q_j)$ represents the PSNR of enhanced image and original image in proposed method and benchmark respectively. The $\overline{PSNR}$ *gain* is calculated as the $\overline{PSNR}$ difference between proposed method and benchmark.

Gains from CNN Assisted Quality Enhancement Model (Step $1-14$)

In order to demonstrate the effectiveness of incorporating CNN model, the total bits of the proposed bit allocation proposal is kept lower than that of the benchmark, by assigning total available bits $\Re$ with the total bits used for benchmark counterpart. Results under switching range of 3 and 5 views are shown in Table. 6.2 and 6.3 respectively. The benchmark method allocates the bitrate without considering the CNN quality enhancement model, which searches proper QP allocation following Equation (6.18) and (6.19). Based on the total bits used in the benchmark proposal, the proposed method allocates them following steps $1-14$ in Algorithm 1, with Step $15-17$ skipped. The QP value assigned for each view are shown with a diverse range of each sequence. As expected, the QP_c in the proposed method is usually lower than that in the benchmark, while the QP_l has an opposite trend. The overall PSNR for both methods are listed, with those using sequence-specific CNN model in bold. The overall PSNR gain of proposed method over benchmark is also listed in the last column. Besides, averaged total bits, averaged overall PSNR and averaged PSNR gain are shown in the bottom of each table, with results of general CNN model and sequence-specific CNN model listed separately. The observations are as follows:

1. For each sequence, three bits levels are tested and compared. The proposed method stably outperforms the benchmark. Averagely, the gain is 0.6 dB with general CNN model for both view switching ranges. The gains with sequence-specific CNN model are even higher.

2. The sequence-specific CNN model (results in bold) shows its strength in the scenario of 5 views, with averagely 0.3 dB higher than the result of general CNN model. While for 3 views scenario, it has a slightly higher result than the general CNN model. With even more views, the gain will be more obvious. This is because the portion of the depth map in bitrate decreases with the increase of views. Thus, the bits saved from lateral views would be dedicated to enhancing the center view with a higher percentage.

3. The general CNN model works well for both sequences within and outside the training set, which demonstrates the generality of the proposed method.

Gains from Exploration of Remaining Available Bits (Step $15-17$)

For a real video streaming system, all steps in Algorithm 1 are executed. That is, the bandwidth would be fully exploited. After bit allocation in Step $1-14$, there will be remaining bits that are not enough to further increase the quality of the center view. Thus, they are allocated among lateral views according to the quality gain per bit, which corresponds to step $15-17$ in Algorithm 1. The gain comes from these final steps are shown in Table. 6.4.

In this experiment, the total available bits $\Re$ is calculated according to Equ. (6.2). Undodancer is tested as an example with the switching range of 5 views. The changes in QP values of lateral views are underlined. It can be found that 18 out of 24 QPs are increased. Accordingly, the percentage of used bandwidth increases from 93% to 96% for both general CNN model and sequence-specific model. The overall PSNR for lateral views, as well as the overall PSNR gain, are presented. Averagely, 0.3 dB is obtained with step $15-17$. Especially when the bandwidth is low, the overall PSNR gain obtained by this final process (around 0.5 dB) should not be neglected.

Complexity Analysis

The complexity of this solution consists of two parts: the complexity of the CNN quality enhancement and the complexity of the bit allocation method. For the previous one, it usually consumes around 0.5 seconds for one frame. While for the bit allocation method, it varies for different cases, depending on the number of available quality levels of the video. In general, it consumes less than 0.1 seconds for one frame. As the future work, methods that reduce the complexity of CNN are worth investigating. For

example, we can analyze the quality enhancement pixel by pixel, and find a method to distinguish the area that cannot be enhanced with CNN. With this method, some areas in an image can be skipped during the CNN quality enhancement process, which will save the total processing time.

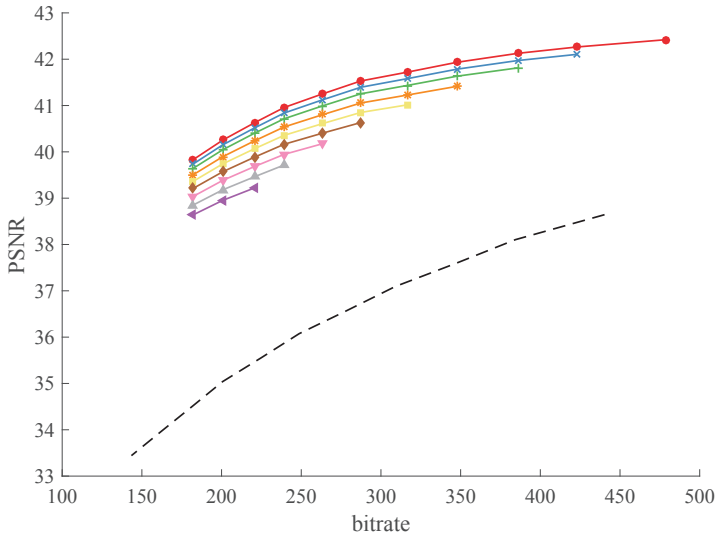
6.4 Conclusions

In this work, we proposed a convolutional neural network assisted seamless multiview video navigation system. The system consists of two parts: a CNN assisted multiview representation method and a navigation guided bit allocation mechanism. The former representation method removes the dependencies between different views so as to provide flexibility. At the same time, redundancies are reduced and exploited, leading to an increase in compression efficiency. As for the proposed bit allocation mechanism, it optimizes the overall quality within the throughput bound, with seamless view switching provided. These two modules can be incorporated in any multiview video streaming system to provide satisfactory viewing experience within the bandwidth constraint.

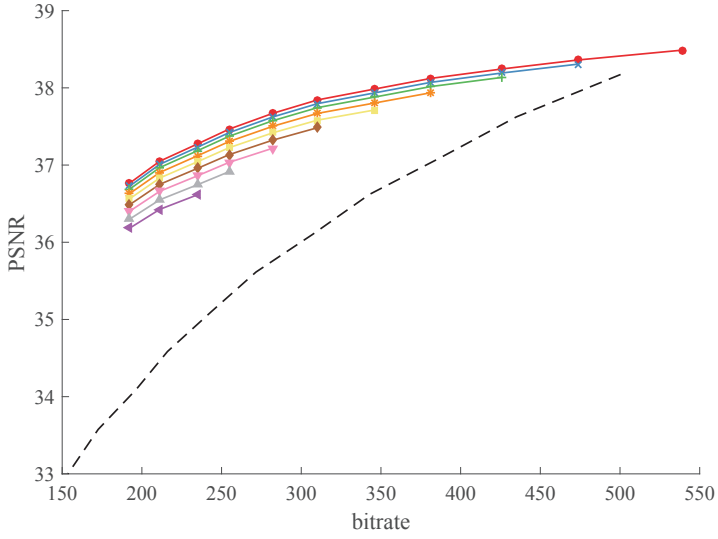
As for the future works, different switching scenarios in navigation model will be investigated, which can be classified as high, medium and low dynamism. High movement comes with higher switching probability than the other two categories, which would lead to a wider range of views that might be watched. The challenges brought along will be investigated. Besides, MVD coding will be also considered in the future work, especially when given a high switching probability.

It is worth reporting that the work reported in this section has led to the following publication:

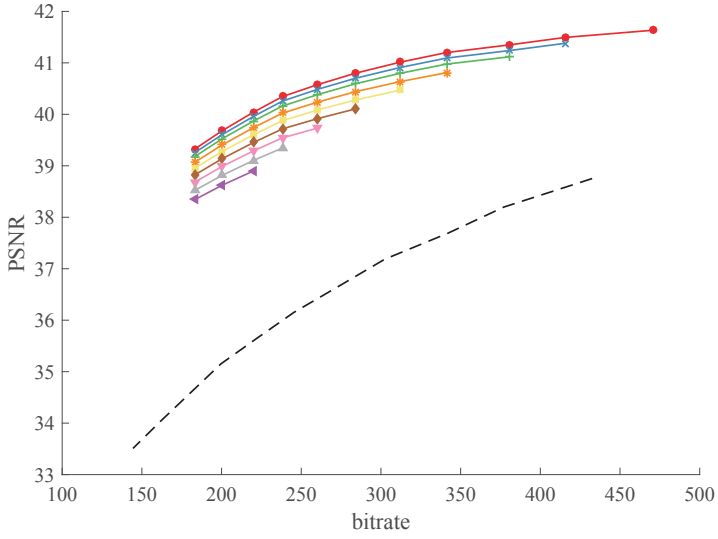
1. Li Yu, Jimin Xiao, T. Tillo, and Macro Grangetto. Convolutional neural network assisted seamless multiview video streaming and navigation. Under revision (IEEE Transactions on Multimedia).



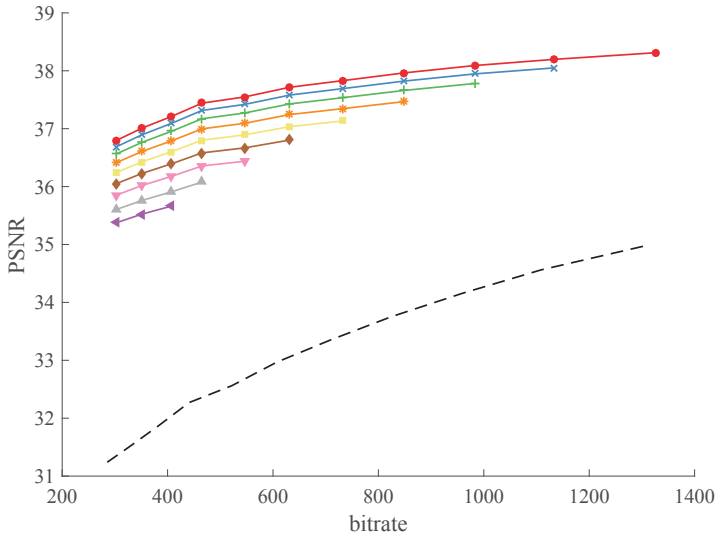
(a) Kendo, $Y_L = 2, Y_C = 3$



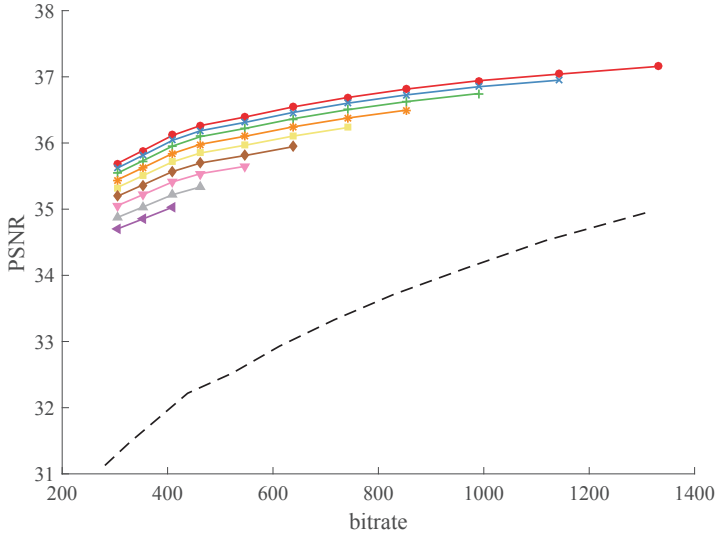
(b) Kendo, $Y_L = 1, Y_C = 3$



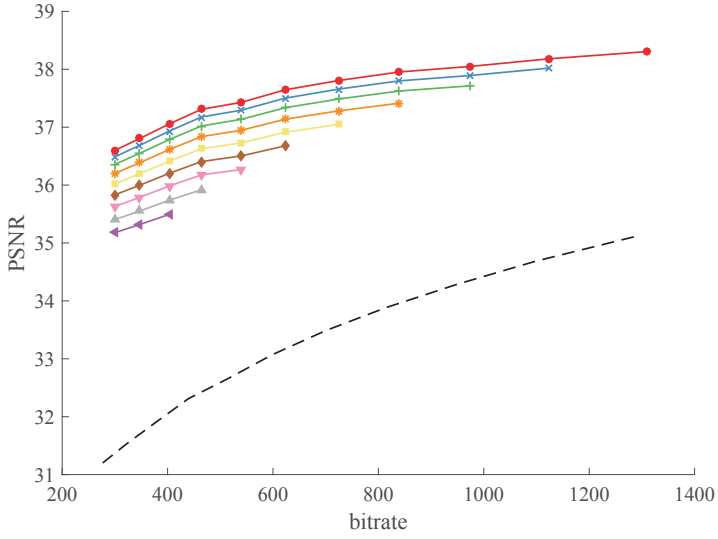
(c) Kendo, $Y_L = 4, Y_C = 3$



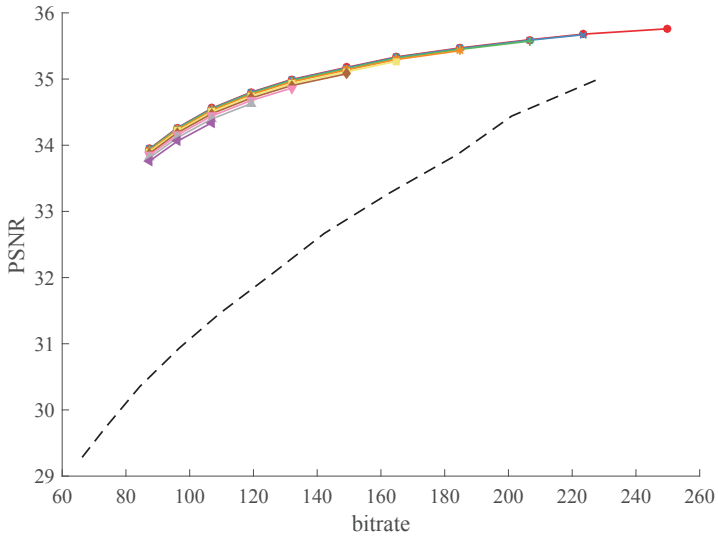
(g) Undodancer, $Y_L = 2, Y_C = 3$



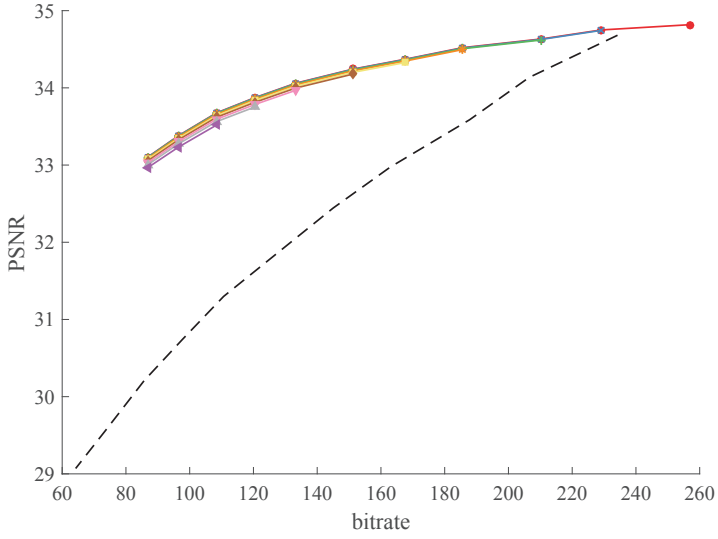
(h) Undodancer, $Y_L = 1, Y_C = 3$



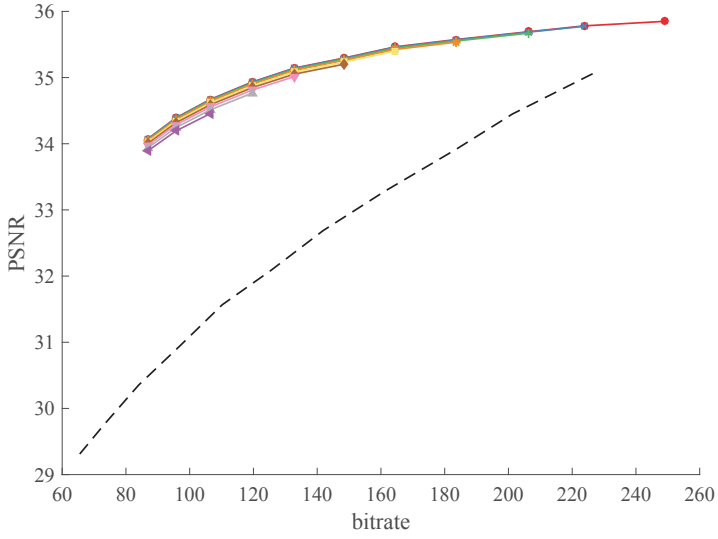
(i) Undodancer, $Y_L = 4, Y_C = 3$



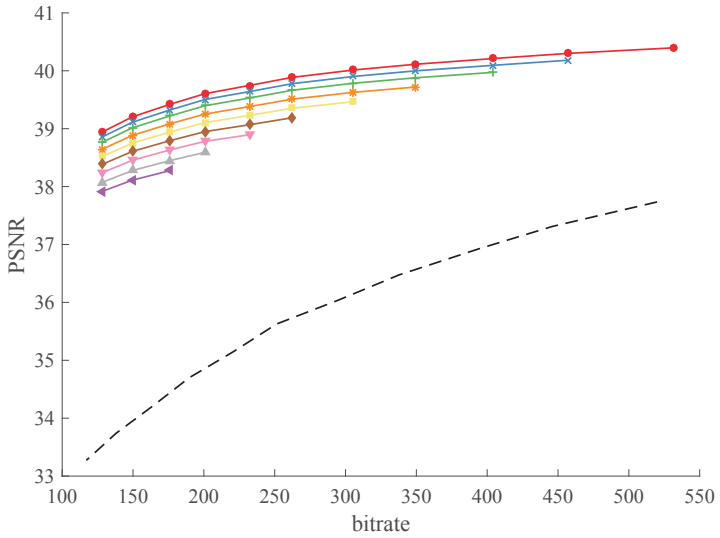
(m) Balloons, $Y_L = 2, Y_C = 3$



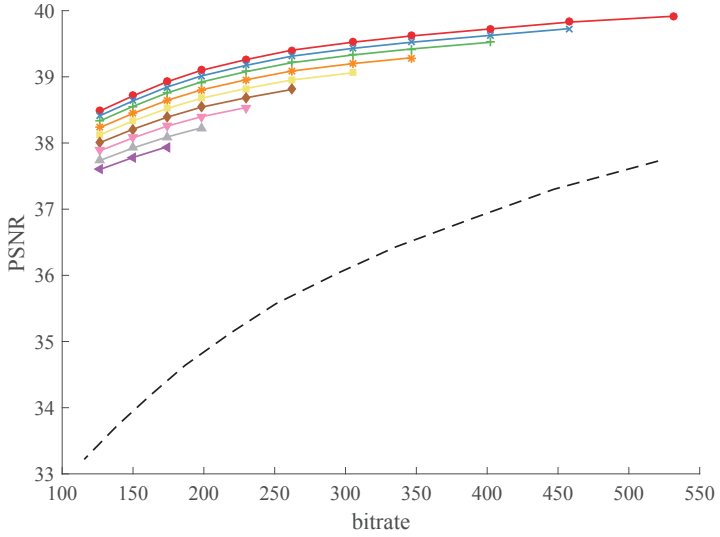
(n) Balloons, $Y_L = 1, Y_C = 3$



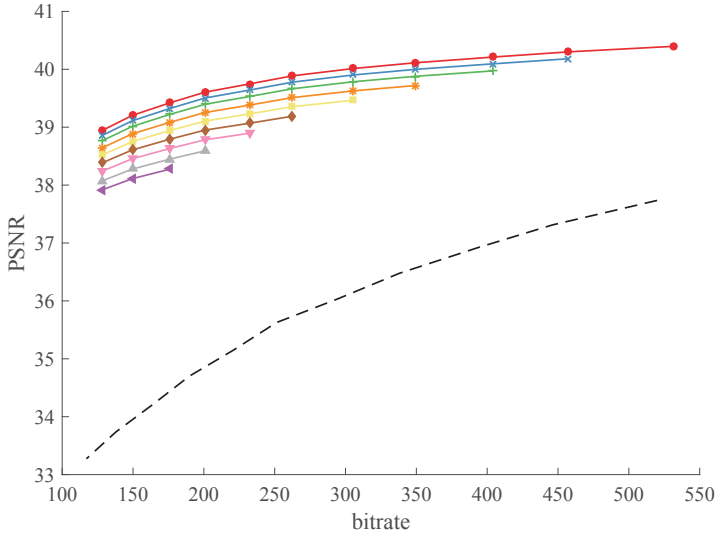
(o) Balloons, $Y_L = 4, Y_C = 3$



(p) GTfly, $Y_L = 2, Y_C = 3$



(q) GTfly, $Y_L = 1, Y_C = 3$



(r) GTfly, $Y_L = 4, Y_C = 3$

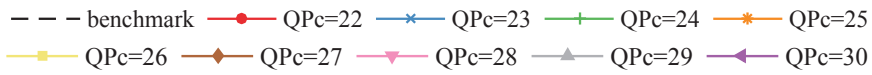
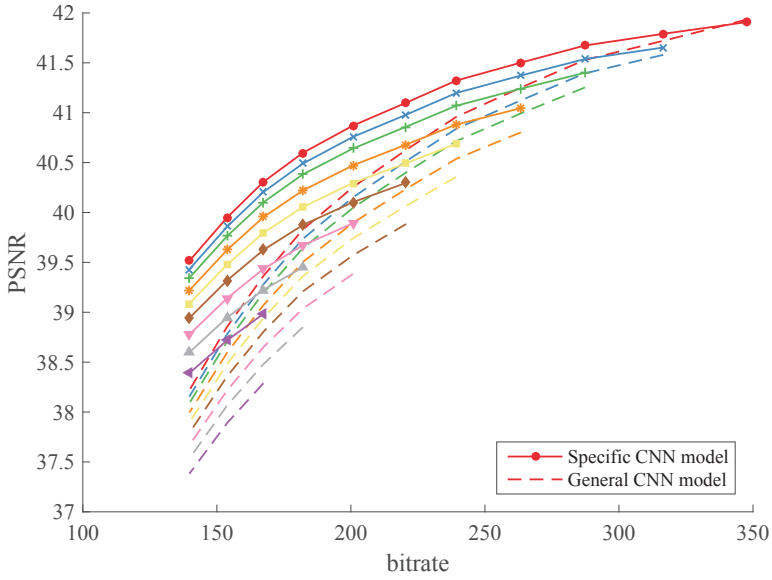
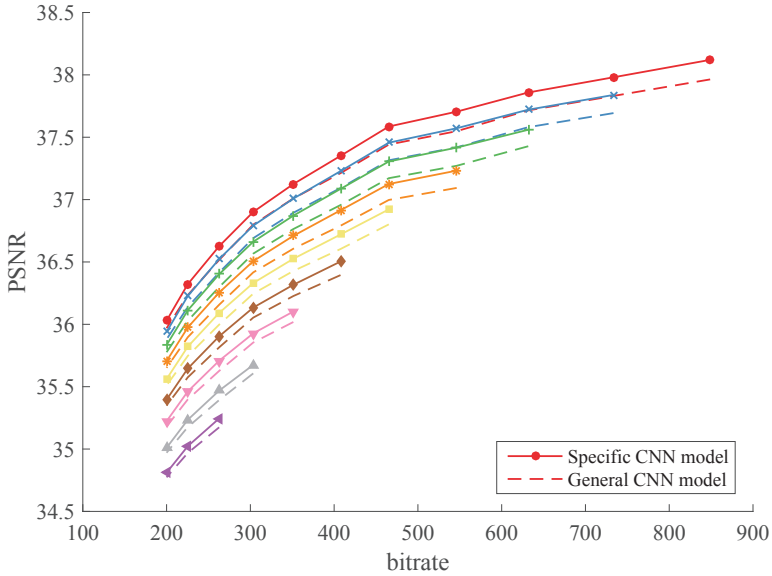


Figure 6.5: Rate Distortion curves of **General CNN Model**, with comparison to benchmark. The benchmark represents the HEVC encoded sequence without any enhancement. The views on the left column $Y_L = 2, Y_C = 3$; center column $Y_L = 1, Y_C = 3$; right column $Y_L = 4, Y_C = 3$. The above two rows are results of sequences within the training set, while the bottom two rows are those of sequences outside the training set.



(a) Kendo



(b) Undodancer

Figure 6.6: Rate Distortion curves of **Sequence-specific CNN Model**, in comparison to General CNN Model ($Y_L = 2, Y_C = 3$).

Chapter 7

Conclusion

7.1 Contributions

In this thesis, we introduce future insight into different stages of video communication system to enhance the overall efficiency. Different optimization algorithms are developed, including a statistical approach for motion estimation skipping, dynamic redundancy allocation for video streaming using Sub-GOP based FEC code, QoE-driven dynamic adaptive video streaming strategy, as well as CNN-assisted seamless multiview video representation and navigation method. Through extensive experiments, we have found that the future information helps a lot in enhancing the performances. With the fast development and deployment of powerful electronic devices, the computational expenses would not be an obstacle. Specifically, the contributions of this thesis are listed in details as follows:

1. We have developed an encoding complexity reduction method for HEVC with negligible losses in RD performance. This method skips unnecessary motion estimations based on the prediction of each unit being referenced in the future. It works best for those sequences with large motions, where the motion estimations are intensive. Besides, it can work coherently with other CU/PU level complexity with no conflicts.

2. A dynamic allocation method of RS parity packets is proposed for FEC code in this thesis. With the goal of optimizing overall RD performance, the parities are assigned based on the importance of each frame. Correspondingly, the importance of each frame is evaluated through its influence over itself and future frames, which is evaluated precisely by simulations with detailed information about video

content. The benefit of the proposed method is providing the flexibility in determining both position and amount of the redundancies. As a result, the efficiency of the added data is optimized. Besides, a balance between performance of protection and influence of error propagation is achieved with this flexibility.

3. We proposed to incorporate QoE as the optimization goal for rate adaptation method in DASH. In order to make the QoE evaluative in the middle of the streaming process, an internal QoE method which measures the local quality of experience is proposed. Meanwhile, the future information, i.e. instant bit rates of segments to download, is exploited to guarantee an accurate result. Besides, a probabilistic bandwidth prediction model is used to ensure the robustness of the proposed method to the fluctuating network. As shown in the experiments, the utilization of future information has a significant influence on the final performance.

4. A navigation guided multiview video streaming solution is proposed, along with a CNN assisted multiview video representation method. The proposed representation method not only offers the flexibility of downloading views by removing their inter-dependencies, but also constraints the overall data size by exploring the similarities among view with the CNN network. Besides, the navigation model provides a guidance on determining the quality of each requested views and thus optimize the overall performance.

7.2 Future Work

1. In the future work, we will investigate the automatic division strategy for SAMEK method depending on the characteristics of the video contents. Currently, the SAMEK follows a fixed division pattern, which may not be optimal for all sequences and frames. Thus, it is worthwhile to develop a method for determining the optimal division strategy for each frame. Based on the automatic division strategy, a further reduction in computational complexity would be obtained.

   In order to develop an automatic division strategy based on the video characteristic, we are planning to analyze the motion vectors of each frame. The motion vectors within one object are similar with high probability, which means their references would be within the same region. Thus, it is better to have them in

one processing unit. While for blocks with quite different motion vectors, it is better to separate them into different processing units. Based on the distribution of motion vectors, each frame can be segmented into units with a flexible shape. Then, SAMEK method can be applied respectively to these processing units.

2. The parameters in our proposed internal QoE formula are tuned manually based on the preference of users. In our future work, we are planning to develop an automatic method to tune the parameters based on the analysis of user behavior, as well as the network situation. The user's profile would be generated based on his/her historical behavior. This profile would demonstrate the user's tolerance over the duration of stalls, as well as the acceptable worst quality level, etc. Then, the parameters would be tuned based on these characteristics. Besides, the monitor of the bandwidth would also influence the setting of the parameters. Especially when the bandwidth is low, a higher weight would be given to the starvation factor.

3. In the future work of the multiview video representation method, the depth of the center view would be directly generated at the client side without transmitting them. By doing so, the saved throughput can be used to further increase the overall quality of downloaded views. While the depth map would instead be generated with two of the downloaded views, i.e. center view and rightmost view. By fed them into a CNN for depth estimation, the depth with satisfactory quality can be derived and used in the following process.

Appendix A

List of publications

1. Yu L, Xiao J, Tillo T, Zhu C. Statistical Approach for Motion Estimation Skipping (SAMEK)[C]//Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015: 3245-3249.

2. Yu L, Xiao J, Tillo T. Dynamic Redundancy Allocation for Video Streaming using Sub-GOP based FEC Code[C]//Visual Communications and Image Processing Conference, 2014 IEEE. IEEE, 2014: 518-521.

3. Yu L, Tillo T, Xiao J. QoE-Driven Dynamic Adaptive Video Streaming Strategy With Future Information[J]. IEEE Transactions on Broadcasting, 2017.

Bibliography

[1] Iain E Richardson. H. 264 and MPEG-4 video compression: video coding for next-generation multimedia. John Wiley & Sons, 2004.

[2] F. Bossen, B. Bross, K. Suhring, and D. Flynn. Hevc complexity and implementation analysis. Circuits and Systems for Video Technology, IEEE Transactions on, 22(12):1685–1696, Dec 2012.

[3] I Cisco. Cisco visual networking index: Forecast and methodology, 2015–2020. CISCO White paper, pages 2015–2020, 2015.

[4] G.J. Sullivan, J. Ohm, Woo-Jin Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. IEEE Trans. Circuits Syst. Video Technol, 22(12):1649–1668, Dec 2012.

[5] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. Circuits and Systems for Video Technology, IEEE Transactions on, 13(7):560–576, 2003.

[6] S Wenger. H. 264/avc over ip. Circuits and Systems for Video Technology, IEEE Transactions on, 13(7):645–656, 2003.

[7] T Stockhammer, M M Hannuksela, and T Wiegand. H. 264/AVC in wireless environments. IEEE Transactions on Circuits and Systems for Video Technology, 13(7):657–673, 2003.

[8] Yo-Sung Ho Yo-Sung Ho and Kwan-Jung Oh Kwan-Jung Oh. Overview of Multiview Video Coding. 2007 14th International Workshop on Systems, Signals and Image Processing and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services, 3164(d), 2007.

[9] Anthony Vetro, Thomas Wiegand, and Gary J. Sullivan. Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard. Proceedings of the IEEE, 99(4):626–642, 2011.

[10] G. J. Sullivan and T. Wiegand. Video compression - from concepts to the h.264/avc standard. Proceedings of the IEEE, 93(1):18–31, Jan 2005.

[11] N Ahmed, T Natarajan, and K R Rao. Discrete cosine transform. IEEE Transactions on Computers, 23(1):90–93, 1974.

[12] Int. Telecommun. Union-Telecommun. (ITU-T). Video codec for audiovisual services at p * 64 kbit/s, recommendation h.261. version1, 1990; version2, 1993.

[13] ISO/IEC 11 172-2(MPEG-1) Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) JTC 1. coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s - part 2: Video. Mar. 1993.

[14] ISOIEC. Information technology generic coding of moving pictures and associated audio information: Video. In International Standard, page 705715, 1993.

[15] Int. Telecommun. Union-Telecommun. (ITU-T). Video coding for low bit rate communication. version1, 1995; version2, 1998; version3, 2000.

[16] ISO/IEC 14 496-2(MPEG-4 visual version 1) Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) JTC 1. Coding of audio-visual objectspart 2: Visual. 1999-2003.

[17] Int. Telecommun. Union-Telecommun. (ITU-T), Recommendation H. 264 Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) JTC 1, and ISO/IEC 14 496-10(MPEG-4) AVC. Advanced video coding for generic audiovisual services. 2003.

[18] H. Kalva. The h.264 video coding standard. IEEE MultiMedia, 13(4):86–90, Oct 2006.

[19] A. Vetro, T. Wiegand, and G. J. Sullivan. Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard. Proceedings of the IEEE, 99(4):626–642, April 2011.

[20] H. Schwarz, D. Marpe, and T. Wiegand. Analysis of hierarchical b pictures and mctf. In 2006 IEEE International Conference on Multimedia and Expo, pages 1929–1932, July 2006.

[21] K. Muller, P. Merkle, and T. Wiegand. 3-d video representation using depth maps. Proceedings of the IEEE, 99(4):643–656, April 2011.

[22] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand. Depth image-based rendering with advanced texture synthesis for 3-d video. IEEE Transactions on Multimedia, 13(3):453–465, June 2011.

[23] P. Kauff, N. Atzpadin, C. Fehn, M. Mller, O. Schreer, A. Smolic, and R. Tanger. Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability. Signal Processing Image Communication, 22(2):217–234, 2007.

[24] George C. Clark and J. Bibb Cain. Error-correction coding for digital communications /. Plenum Press, 1981.

[25] Shu Lin and Daniel J. Costello Jr. Error control coding : fundamentals and applications / s. lin, d.j. costello jr. Principles of Mobile Communication, 1983.

[26] A. M. Michelson and A. H. Levesque. Error-control techniques for digital communication. Wiley, 1985.

[27] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya Qin Zhang, and Jon M. Peha. Streaming video over the internet: Approaches and directions. IEEE Transactions on Circuits and Systems for Video Technology, 11(3):282–300, 2001.

[28] John G. Apostolopoulos, Wai-tian Tan, and Susie J. Wee. Video Streaming: Concepts, Algorithms, and Systems. HP Laboratories Palo Alto, HPL-2002-2, 2002.

[29] Dapeng Wu, Y. T. Hou, Wenwu Zhu, Hung-Ju Lee, Tihao Chiang, Ya-Qin Zhang, and H. J. Chao. On end-to-end architecture for transporting mpeg-4 video over the internet. IEEE Transactions on Circuits and Systems for Video Technology, 10(6):923–941, Sep 2000.

[30] Dapeng Wu, Yiwei Thoms Hou, and Ya-Qin Zhang. Transporting real-time video over the internet: challenges and approaches. Proceedings of the IEEE, 88(12):1855–1877, Dec 2000.

[31] Steven McCanne, Van Jacobson, and Martin Vetterli. Receiver-driven layered multicast. SIGCOMM Comput. Commun. Rev., 26(4):117–130, August 1996.

[32] Quji Guo, Qian Zhang, Wenwu Zhu, and Ya-Qin Zhang. A sender-adaptive and receiver-driven layered multicast scheme for video over internet. In Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on, volume 5, pages 141–144 vol. 5, 2001.

[33] Shun Yan Cheung, M. H. Ammar, and Xue Li. On the use of destination set grouping to improve fairness in multicast video distribution. In INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE, volume 2, pages 553–560 vol.2, Mar 1996.

[34] G J Conklin, Gary S Greenbaum, K O Lillevold, A F Lippman, and Y A Reznik. Video coding for streaming media delivery on the internet. IEEE Transactions on Circuits and Systems for Video Technology, 11(3):269–281, 2001.

[35] Yao Wang, M. T. Orchard, and A. R. Reibman. Multiple description image coding for noisy channels by pairing transform coefficients. In Multimedia Signal Processing, 1997., IEEE First Workshop on, pages 419–424, Jun 1997.

[36] J. Postel. Transmission control protocol (tcp). http://tools.ietf.org/html/rfc793.

[37] J. Postel. User datagram protocol (udp). http://tools.ietf.org/html/rfc768.

[38] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Real-time transport protocol (rtp). http://tools.ietf.org/html/rfc3550.

[39] Microsoft Corporation. Microsoft media server (mms) protocol specification. http://msdn.microsoft.com/en-us/library/cc234711%28v=prot.10%29.aspx.

[40] Adobe Systems and Inc. Real-time messaging protocol (rtmp) specification. http://www.adobe.com/devnet/rtmp.html.

[41] Thomas Stockhammer. Dynamic adaptive streaming over http: standards and design principles. In Proc. of the 2nd Annu. ACM Conf. on Multimedia Syst., pages 133–144. ACM, 2011.

[42] Jimin Xiao, Miska M. Hannuksela, Tammam Tillo, and Moncef Gabbouj. A paradigm for dynamic adaptive streaming over HTTP for multi-view video. In Advances in Multimedia Inform. Process. - PCM 2015 - 16th Pacific-Rim Conf. on Multimedia, Gwangju, South Korea, September 16-18, 2015, Proc., Part II, pages 410–418, 2015.

[43] Tuan Vu, Hung T. Le, Duc V. Nguyen, Nam Pham Ngoc, and Truong Cong Thang. Future buffer based adaptation for vbr video streaming over http. In IEEE 17th Int. Workshop on Multimedia Signal Process. (MMSP), 2015, 2015.

[44] Y. Zhou, Y. Duan, J. Sun, and Z. Guo. Towards simple and smooth rate adaption for vbr video in dash. In Visual Communications and Image Processing Conference, 2014 IEEE, pages 9–12, Dec 2014.

[45] Christian Timmerer, Matteo Maiero, and Benjamin Rainer. Which adaptation logic? an objective and subjective performance evaluation of http-based adaptive media streaming systems. arXiv preprint arXiv:1606.00341, 2016.

[46] ITU-T Recommendation P.10/G.100-Amendment 3. Vocabulary for performance and quality of service. Telecommunications Sector, Recommendations of the ITU, 12 2012.

[47] ITU-T Recommendation P.10/G.100-Amendment 2. Vocabulary for performance and quality of service. Telecommunications Sector, Recommendations of the ITU, 12 2012.

[48] Raimund Schatz, Tobias Hoßfeld, Lucjan Janowski, and Sebastian Egger. From packets to people: quality of experience as a new measurement challenge. In Data traffic monitoring and analysis, pages 219–263. Springer, 2013.

[49] René Serral-Gracià, Eduardo Cerqueira, Marília Curado, Marcelo Yannuzzi, Edmundo Monteiro, and Xavier Masip-Bruin. An overview of quality of experience measurement challenges for video applications in ip networks. In

International Conference on Wired/Wireless Internet Communications, pages 252–263. Springer, 2010.

[50] Wang Zhou, Lu Liang, and A Bovik. Video quality assessment using structural distortion measurement. In Proc. 2002 Int. Conf. on Image Processing, Rochester, NY, USA, volume 3, 2002.

[51] S Shunmuga Krishnan and Ramesh K Sitaraman. Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. IEEE/ACM Transactions on Networking, 21(6):2001–2014, 2013.

[52] Tobias Hoßfeld, Sebastian Egger, Raimund Schatz, Markus Fiedler, Kathrin Masuch, and Charlott Lorentzen. Initial delay vs. interruptions: between the devil and the deep blue sea. In Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on, pages 1–6. IEEE, 2012.

[53] S Rugel, TM Knoll, M Eckert, and T Bauschert. A network-based method for measurement of internet video streaming quality. In European Teletraffic Seminar Poznan University of Technology, Poland, 2011.

[54] Liu Yitong, Shen Yun, Mao Yinian, Liu Jing, Lin Qi, and Yang Dacheng. A study on quality of experience for adaptive streaming service. In 2013 IEEE International Conference on Communications Workshops (ICC), pages 682–686. IEEE, 2013.

[55] Nicola Cranley, Philip Perry, and Liam Murphy. User perception of adapting video quality. International Journal of Human-Computer Studies, 64(8):637–647, 2006.

[56] Felipe Sampaio, Sergio Bampi, Mateus Grellert, Luciano Agostini, and Julio Mattos. Motion vectors merging: low complexity prediction unit decision heuristic for the inter-prediction of hevc encoders. In Multimedia and Expo (ICME), 2012 IEEE International Conference on, pages 657–662. IEEE, 2012.

[57] Liquan Shen, Zhi Liu, Xinpeng Zhang, Wenqiang Zhao, and Zhaoyang Zhang. An effective cu size decision method for hevc encoders. Multimedia, IEEE Transactions on, 15(2):465–470, 2013.

[58] Xiaolin Shen, Lu Yu, and Jie Chen. Fast coding unit size selection for hevc based on bayesian decision rule. In Picture Coding Symposium (PCS), 2012, pages 453–456. IEEE, 2012.

[59] Jaehwan Kim, Jungyoup Yang, Kwanghyun Won, and Byeungwoo Jeon. Early determination of mode decision for hevc. In Picture Coding Symposium (PCS), 2012, pages 449–452. IEEE, 2012.

[60] Seungha Yang, Hiuk Jae Shim, Kwanghyun Won, and Byeungwoo Jeon. Fast inter sub-partition prediction unit mode decision for hevc. In Consumer Electronics (ICCE), 2014 IEEE International Conference on, pages 15–16. IEEE, 2014.

[61] Shih-Hsuan Yang, Jia-Ze Jiang, and Hsien-Jie Yang. Fast motion estimation for hevc with directional search. Electronics Letters, 50(9):673–675, 2014.

[62] Ce Zhu, Xiao Lin, L. Chau, and Lai-Man Po. Enhanced hexagonal search for fast block motion estimation. Circuits and Systems for Video Technology, IEEE Transactions on, 14(10):1210–1214, Oct 2004.

[63] J Ohm, Gary J Sullivan, Heiko Schwarz, Thiow Keng Tan, and Thomas Wiegand. Comparison of the coding efficiency of video coding standardsincluding high efficiency video coding (hevc). Circuits and Systems for Video Technology, IEEE Transactions on, 22(12):1669–1684, 2012.

[64] Bing Xiong and Ce Zhu. A new multiplication-free block matching criterion. Circuits and Systems for Video Technology, IEEE Transactions on, 18(10):1441–1446, Oct 2008.

[65] Bryan Peterson. Learning to see creatively. Random House LLC, 2011.

[66] Gisle Bjontegaard. Calcuation of average psnr differences between rd-curves. Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April 2001, 2001.

[67] Yuan Zhang, Wen Gao, Yan Lu, Qingming Huang, and Debin Zhao. Joint source-channel rate-distortion optimization for h. 264 video coding over error-prone networks. Multimedia, IEEE Transactions on, 9(3):445–454, 2007.

[68] Tammam Tillo, Marco Grangetto, and Gabriella Olmo. Redundant slice optimal

allocation for h. 264 multiple description coding. Circuits and Systems for Video Technology, IEEE Transactions on, 18(1):59–70, 2008.

[69] Ivana Radulovic, Pascal Frossard, Ye-Kui Wang, Miska M Hannuksela, and Antti Hallapuro. Multiple description video coding with h. 264/avc redundant pictures. Circuits and Systems for Video Technology, IEEE Transactions on, 20(1):144–148, 2010.

[70] Enrico Baccaglini, Tammam Tillo, and Gabriella Olmo. Slice sorting for unequal loss protection of video streams. Signal Processing Letters, IEEE, 15:581–584, 2008.

[71] Sohraab Soltani, Kiran Misra, and Hayder Radha. Delay constraint error control protocol for real-time video communication. Multimedia, IEEE Transactions on, 11(4):742–751, 2009.

[72] Shunan Lin, Shiwen Mao, Yao Wang, and Shivendra S Panwar. A reference picture selection scheme for video transmission over ad-hoc networks using multiple paths. In ICME, 2001.

[73] Abdelhamid Nafaa, Tarik Taleb, and Liam Murphy. Forward error correction strategies for media streaming over wireless networks. IEEE Communications Magazine, 46(1):72, 2008.

[74] Xingjun Zhang, Xiaohong Peng, Scott Fowler, and Dajun Wu. Robust h.264/avc video transmission using data partitioning and unequal loss protection. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, pages 2471–2477. IEEE, 2010.

[75] Nikolaos Thomos, Savvas Argyropoulos, Nikolaos V Boulgouris, and Michael G Strintzis. Robust transmission of h. 264/avc video using adaptive slice grouping and unequal error protection. In Multimedia and Expo, 2006 IEEE International Conference on, pages 593–596. IEEE, 2006.

[76] Xiaokang Yang, Ce Zhu, Zheng Guo Li, Xiao Lin, and Nam Ling. An unequal packet loss resilience scheme for video over the internet. Multimedia, IEEE Transactions on, 7(4):753–765, 2005.

[77] Jimin Xiao, Tammam Tillo, Chunyu Lin, and Yao Zhao. Dynamic sub-gop forward error correction code for real-time video applications. Multimedia, IEEE Transactions on, 14(4):1298–1308, 2012.

[78] Niko Farber, Klaus Stuhlmuller, and Bernd Girod. Analysis of error propagation in hybrid video coding with application to error resilience. 2:550–554, 1999.

[79] I Cisco. Cisco visual networking index: Forecast and methodology, 2014-2019. CISCO White paper, 2014.

[80] Henning Schulzrinne, Stephen Casner, Ron Frederick, and Van Jacobson. Rtp: A transport protocol for real-time applications. Technical report, 2003.

[81] Truong Cong Thang, Quang-Dung Ho, Jung Won Kang, and Anh T Pham. Adaptive streaming of audiovisual content using mpeg dash. IEEE Trans. Consum. Electron., 58(1):78–85, 2012.

[82] Wikipedia. Variable bitrate. `http://en.wikipedia.org/wiki/Variable_bitrate`.

[83] MSDN. Variable bit rate (vbr) encoding. `http://msdn.microsoft.com/en-us/library/windows/desktop/dd743964(v=vs.85).aspx`.

[84] TC Thang, JY Lee, JW Kang, SJ Bae, S Jung, and ST Park. Proposal on signaling for dash. ISO/IEC JTC1/SC29/WG11 m18445, Guangzhou, 2010.

[85] Truong Cong Thang, Hung T Le, Huan X Nguyen, Anh T Pham, Jung Won Kang, and Yong Man Ro. Adaptive video streaming over http with dynamic resource estimation. J. of Commun. and Networks, 15(6):635–644, 2013.

[86] Aninda Bhattacharya, Alexander G Parlos, and Amir F Atiya. Prediction of mpeg-coded video source traffic using recurrent neural networks. IEEE Trans. Signal Process., 51(8):2177–2190, 2003.

[87] Salahuddin Azad, Wei Song, and Dian Tjondronegoro. Bitrate modeling of scalable videos using quantization parameter, frame rate and spatial resolution. In IEEE Int. Conf. on Acoust. Speech and Signal Process. (ICASSP), 2010, pages 2334–2337. IEEE, 2010.

[88] P. Venkat Rangan, H.M. Vin, and S. Ramanathan. Designing an on-demand multimedia service. IEEE Commun. Mag., 30(7):56–64, July 1992.

[89] H.T. Le, Nam Pham Ngoc, T.A. Vu, A.T. Pham, and Truong Cong Thang. Smooth-bitrate adaptation method for http streaming in vehicular environments. In IEEE Veh. Networking Conf. (VNC), 2014, pages 187–188, Dec 2014.

[90] Chenghao Liu, Imed Bouazizi, and Moncef Gabbouj. Rate adaptation for adaptive http streaming. In Proc. of the 2nd Annu. ACM Conf. on Multimedia Syst., pages 169–174. ACM, 2011.

[91] V. Menkovski and A. Liotta. Intelligent control for adaptive video streaming. In IEEE Int. Conf. on Consumer Electron. (ICCE), 2013, pages 127–128, Jan 2013.

[92] Maxim Claeys, Steven Latré, Jeroen Famaey, Tingyao Wu, Werner Van Leekwijck, and Filip De Turck. Design of a q-learning-based client quality selection algorithm for http adaptive video streaming. pages 30–37, 2013.

[93] M. Claeys, S. Latre, J. Famaey, and F. De Turck. Design and evaluation of a self-learning http adaptive video streaming client. IEEE Commun. Lett., 18(4):716–719, April 2014.

[94] Tobias Hofeld, Michael Seufert, Christian Sieber, Thomas Zinner, and Phuoc Tran-Gia. Identifying qoe optimal adaptation of http adaptive streaming based on subjective studies. Computer Networks, 81:320 – 332, 2015.

[95] Ricky K. P. Mok, Xiapu Luo, Edmond W. W. Chan, and Rocky K. C. Chang. Qdash: A qoe-aware dash system. In Proceedings of the 3rd Multimedia Systems Conference, MMSys '12, pages 11–22, New York, NY, USA, 2012. ACM.

[96] D. Jarnikov and T. Ozcelebi. Client intelligence for adaptive streaming solutions. In IEEE Int. Conf. on Multimedia and Expo (ICME), 2010, pages 1499–1504, July 2010.

[97] A. Bokani, M. Hassan, and S. Kanhere. Http-based adaptive streaming for mobile clients using markov decision process. In 2013 20th International Packet Video Workshop, pages 1–8, Dec 2013.

[98] S. Garcia, J. Cabrera, and N. Garcia. Quality-optimization algorithm based on stochastic dynamic programming for mpeg dash video streaming. In IEEE Int. Conf. on Consumer Electron. (ICCE), 2014, pages 574–575, Jan 2014.

[99] Yuedong Xu, Yipeng Zhou, and Dah-Ming Chiu. Analytical qoe models for bit-rate switching in dynamic adaptive streaming systems. IEEE Trans. Mobile Computing, 13(12):2734–2748, Dec 2014.

[100] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hoßfeld, and Phuoc Tran-Gia. A survey on quality of experience of http adaptive streaming. IEEE Communications Surveys & Tutorials, 17(1):469–492, 2015.

[101] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao. Deriving and validating user experience model for dash video streaming. IEEE Transactions on Broadcasting, 61(4):651–665, Dec 2015.

[102] N. Staelens, J. De Meulenaere, M. Claeys, G. Van Wallendael, W. Van den Broeck, J. De Cock, R. Van de Walle, P. Demeester, and F. De Turck. Subjective quality assessment of longer duration video sequences delivered over http adaptive streaming to tablet devices. IEEE Transactions on Broadcasting, 60(4):707–714, Dec 2014.

[103] Claudio Alberti, Daniele Renzi, Christian Timmerer, Christopher Mueller, Stefan Lederer, Stefano Battista, and Marco Mattavelli. Automated qoe evaluation of dynamic adaptive streaming over http. In Quality of Multimedia Experience (QoMEX), 2013 Fifth International Workshop on, pages 58–63. Ieee, 2013.

[104] Xiaoqi Yin, Vyas Sekar, and Bruno Sinopoli. Toward a principled framework to design dynamic adaptive streaming algorithms over http. In Proceedings of the 13th ACM Workshop on Hot Topics in Networks, page 9. ACM, 2014.

[105] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz. Adaptation algorithm for adaptive streaming over http. In 19th Int. Packet Video Workshop (PV), 2012, pages 173–178, May 2012.

[106] Travis Andelin, Vasu Chetty, Devon Harbaugh, Sean Warnick, and Daniel Zappala. Quality selection for dynamic adaptive streaming over http with scalable

video coding. In Proceedings of the 3rd Multimedia Systems Conference, pages 149–154. ACM, 2012.

[107] Min Xing, Siyuan Xiang, and Lin Cai. Rate adaptation strategy for video streaming over multiple wireless access networks. In Global Communications Conference (GLOBECOM), 2012 IEEE, pages 5745–5750. IEEE, 2012.

[108] C. D. Iskander and P. T. Mathiopoulos. Fast simulation of diversity nakagami fading channels using finite-state markov models. IEEE Transactions on Broadcasting, 49(3):269–277, Sept 2003.

[109] Chao Zhou, Chia-Wen Lin, and Zongming Guo. m dash: A markov decision-based rate adaptation approach for dynamic http streaming. IEEE Transactions on Multimedia, 18(4):738–751, 2016.

[110] Christian Timmerer Stefan Lederer, Christopher Muller. Dynamic adaptive streaming over http dataset. Proc. of the 3rd Annu. Acm Sigmm Conf. on Multimedia Syst. Mmsys 2012 Chapel Hill Nc Usa February 22 24 2012, pages 89–94, 2012.

[111] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi. Video coding with h.264/avc: tools, performance, and complexity. IEEE Circuits Syst. Mag., 4(1):7–28, First 2004.

[112] I. Sexton and P. Surman. Stereoscopic and autostereoscopic display systems. IEEE Signal Processing Magazine, 16(3):85–99, May 1999.

[113] P. Merkle, A. Smolic, K. Muller, and T. Wiegand. Multi-view video plus depth representation and coding. In 2007 IEEE International Conference on Image Processing, volume 1, pages I – 201–I – 204, Sept 2007.

[114] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand. 3d video and free viewpoint video - technologies, applications and mpeg standards. In 2006 IEEE International Conference on Multimedia and Expo, pages 2161–2164, July 2006.

[115] A. Vetro, T. Wiegand, and G. J. Sullivan. Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard. Proceedings of the IEEE, 99(4):626–642, April 2011.

[116] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall, and A. Vetro. Standardized extensions of high efficiency video coding (hevc). IEEE Journal of Selected Topics in Signal Processing, 7(6):1001–1016, Dec 2013.

[117] K. Mller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. H. Rhee, G. Tech, M. Winken, and T. Wiegand. 3d high-efficiency video coding for multi-view video and depth data. IEEE Transactions on Image Processing, 22(9):3366–3378, Sept 2013.

[118] T. Maugey, I. Daribo, G. Cheung, and P. Frossard. Navigation domain representation for interactive multiview imaging. IEEE Transactions on Image Processing, 22(9):3459–3472, Sept 2013.

[119] J. Xiao, M. M. Hannuksela, T. Tillo, M. Gabbouj, C. Zhu, and Y. Zhao. Scalable bit allocation between texture and depth views for 3-d video streaming over heterogeneous networks. IEEE Transactions on Circuits and Systems for Video Technology, 25(1):139–152, Jan 2015.

[120] T. Maugey and P. Frossard. Interactive multiview video system with low decoding complexity. In 2011 18th IEEE International Conference on Image Processing, pages 589–592, Sept 2011.

[121] M. Karczewicz and R. Kurceren. The sp- and si-frames design for h.264/avc. IEEE Transactions on Circuits and Systems for Video Technology, 13(7):637–644, July 2003.

[122] Ying Chen, Ye-Kui Wang, Kemal Ugur, Miska M Hannuksela, Jani Lainema, and Moncef Gabbouj. The emerging mvc standard for 3d video services. EURASIP Journal on Applied Signal Processing, 2009:8, 2009.

[123] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp. Client-driven selective streaming of multiview video for interactive 3dtv. IEEE Transactions on Circuits and Systems for Video Technology, 17(11):1558–1565, Nov 2007.

[124] A. M. Tekalp, E. Kurutepe, and M. R. Civanlar. 3dtv over ip. IEEE Signal Processing Magazine, 24(6):77–87, Nov 2007.

[125] T. Maugey, G. Petrazzuoli, P. Frossard, M. Cagnazzo, and B. Pesquet-Popescu. Reference view selection in dibr-based multiview coding. IEEE Transactions on Image Processing, 25(4):1808–1819, April 2016.

[126] G. Cheung, A. Ortega, and N. M. Cheung. Interactive streaming of stored multiview video using redundant frame structures. IEEE Transactions on Image Processing, 20(3):744–761, March 2011.

[127] G. Petrazzuoli, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu. Using distributed source coding and depth image based rendering to improve interactive multiview video access. In 2011 18th IEEE International Conference on Image Processing, pages 597–600, Sept 2011.

[128] U. Takyar, T. Maugey, and P. Frossard. Extended layered depth image representation in multiview navigation. IEEE Signal Processing Letters, 21(1):22–25, Jan 2014.

[129] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In Proceedings of the IEEE International Conference on Computer Vision, pages 576–584, 2015.

[130] Y. Xie, J. Xiao, T. Tillo, Y. Wei, and Y. Zhao. 3d video super-resolution using fully convolutional neural networks. In 2016 IEEE International Conference on Multimedia and Expo (ICME), pages 1–6, July 2016.

[131] B. Li, J. Xu, D. Zhang, and H. Li. Qp refinement according to lagrange multiplier for high efficiency video coding. In 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013), pages 477–480, May 2013.

[132] Fujii Laboratory at Nagoya University. Nagoya university sequences. `http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/`.

[133] HM (HEVC Test Model)16.0. `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.0/`.