ADVANCED TRANSCODING TECHNIQUES FOR VIDEO COMMUNICATION

SHU HAIYAN

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2007

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

....

Date SHU HAIYAN

To my parents
for their love.

Acknowledgements

First and foremost, I would like to express the deepest appreciation and sincere gratitude to Dr. Chau Lap-Pui, my supervisor, for his precious time and invaluable instruction. It is he that gave me strong support and sincere encouragement throughout the tough time of research. Dr. Chau has always inspired me with his insights and ideas, furthering my research to the right direction. I have learned not only specialized knowledge from him, but also the spirit of exploring science. This work would not have been possible without his excellent guidance and persistent encouragement.

I would also like to acknowledge School of EEE, NTU, Singapore, for awarding me the research scholarship and providing me with the excellent research facilities. I would like to thank graduate students, faculty, and technicians in Media Technology Lab, for their help and support in my research. In particular, I thank Dr. Zheng Jinghong, Mr. Jing Xuan, Mr. Gao Yunlong, Mr. Fang Tao, and Ms. Wang Yu, for so many enjoyable meetings and discussions. I wish to thank all my friends for letting me have so many wonderful memories.

Last but not the least, thank my parents for their continuous understanding, encouragement and selfless support. Without their love and support, I would not have reached my current goals.

Summary

Today, more and more user devices are accessible to multimedia content via different networks. Applications based on visual content delivery over networks have grown rapidly. These applications are designed especially for different end users or for different networks. There is an interoperability problem between different systems and different networks. Video adaptation is adopted to solve this problem. The explosive growth of multimedia contents, the accessible worldwide networks, the heterogeneous video-enabled devices, and diversity of video-related coding standards have expedited the need for efficient video adaptation.

Video adaptation is to transform the input video to an output in a video or augmented multimedia form by utilizing manipulations at multiple levels (signal, structure, or semantic) to meet diverse resource constraints and user preferences while optimizing the overall utility of video. In order to provide seamless service for end users, video transcoding has been adopted as an efficient tool to realize manipulations of coded representations and bit allocation, which is a signal level adaptation. Video transcoder serves as a gateway to convert a pre-coded bitstream to another coded bitstream according to different requirements. This facilitates the content providers to generate source infor-

mation without considering the channel conditions and the user requirements. End users also benefit from this and can receive the information they require.

According to different applications, there are four types of transcoding: rate-related transcoding, format-related transcoding, error-related transcoding, and information-related transcoding. The main goal of rate-related transcoding is to change the bit rate of the pre-coded bitstream, and present a "best" output video quality. This is the most commonly used scheme in homogeneous transcoding. Format-related transcoding employs transcoder to convert the pre-coded bitstream into different coding standards or different coding format (e.g. scalable to non-scalable, interlace to progress, etc.). To cope with the channel error, error-related transcoding is introduced to insert error resilient component to the pre-coded bitstream. The transcoded bitstream becomes more robust to the channel error. Apart from these three types, information-related transcoding involves the combination of visual or visual related information to generate a new bitstream. Here, the visual or visual related information can be the video signal, logo, or user preference information, etc.

In this work, downsizing transcoding was the first main concern. Two aspects of downsizing transcoding were discussed. First is the resizing algorithm. Two resizing schemes were proposed to realize arbitrary downsizing in the DCT domain directly. This helps to improve the video quality compared with spatial domain schemes. In these two schemes, one is designed for the inter frame and the other is a more general solution. The second aspect of downsizing transcoding is the frame size selection. With introducing of arbitrary downsizing, it is desired to select a suitable frame size for the transcoded bitstream to fully make use of the bandwidth and present higher video quality. In the proposed scheme,

the frame size selection problem has been converted to a bit rate estimation problem. This reduces the computation for the selection.

Next, frame skipping transcoding was discussed. In frame skipping transcoding, bits saved from the skipped frames can be used to code those preserved frames. The spatial quality of those preserved frames is improved. In order to reduce the jerky effect introduced by frame skipping, the concept of "motion change" was introduced to evaluate the motion continuity. Combined with the motion activity information, these two parts of information can be used to determine the skipping pattern of the transcoded bitstream. The nonlinear motion in the transcoded bitstream is reduced. This will improve the quality of the reconstructed skipped frames.

In order to enhance the robustness of the transcoded bitstream against the channel error, error resilient transcoding was investigated. Based on the conventional rate-distortion optimized intra refresh scheme, an enhanced intra/inter macroblock mode decision scheme was proposed. The proposed scheme considers not only the error propagated from previous frames, but also the error propagated to the next frame. The transcoded bitstream presents higher robustness when compared with the conventional schemes since it presents a global suboptimal solution. Considering the availability of information of the next frame, two cases were discussed respectively. They both outperform the conventional schemes in terms of PSNR.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Background

With the blossoming of multimedia system and network technology, there has been an ever-increasing thrust on developing the corresponding applications and services. Among them, video delivery is one of the most prominent applications in multimedia communication. Up to now, various multimedia devices have been developed to display visual contents for different purposes. At the same time, telecommunication techniques promote more benefits for end users to access to visual contents from networks. All of these make video delivery more popular in our daily life.

As multimedia applications are designed for different users or different networks, this gives rise to a serious problem — interoperability. As shown in Figure 1.1, video contents are obtained from live sources (e.g. video camera and video surveillant system), and coded with different coding standards (e.g. JPEG, H.261, H.263, H.264, MPEG-1, MPEG-2, MPEG-4). These coding stan-

dards are targeted for different applications. Besides, the transmission environments are diversiform as well. For Local Area Network (LAN), the transmission rate is high enough to guarantee a satisfactory quality. When it comes to wireless network, the transmission rate is quite low, and therefore, the quality of transmission is dubious. Such diversity of networks makes it very hard to transmit the pre-coded video without any modification.

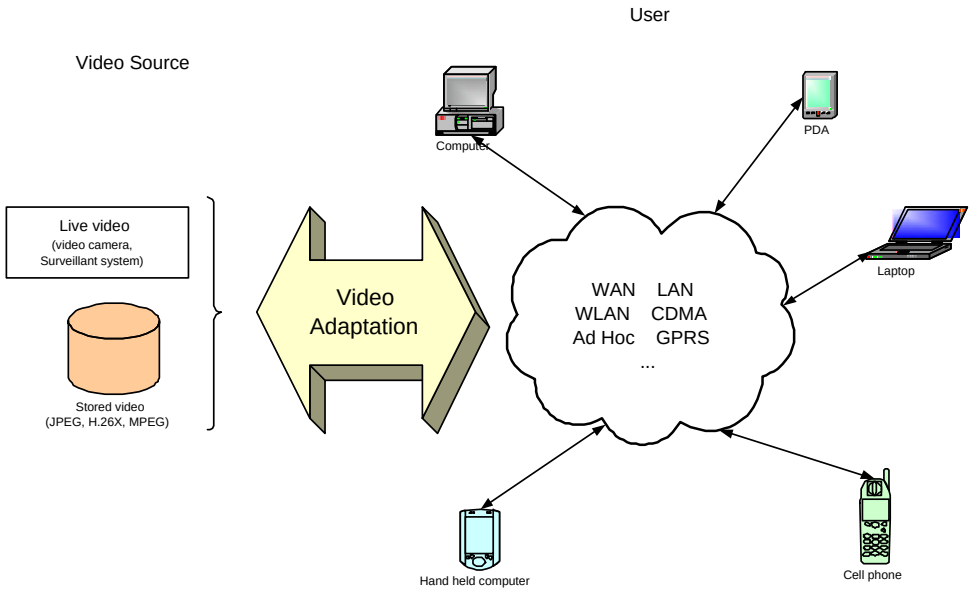


Figure 1.1: Environment of video adaptation

In addition, more and more emerging devices are video-enabled. At first, only TV was available to display video information. It obtains video via cable line. Then the computer system was used to display digital video. This kind of devices obtains the video via LAN or wireless LAN (WLAN). Nowadays, many devices are available to display the video. They obtain video contents via different ways. Personal Digital Assistant (PDA) and hand held computer are able to acquire video information via wireless network; Mobile phone can

display video-related information such as Multimedia Messaging Service (MMS) or Enhanced Messaging Service (EMS), and it receives video via General Packet Radio Service (GPRS) or Code Division Multiple Access (CDMA) network.

The explosive growth of multimedia contents, the accessible worldwide networks, the heterogeneous video-enabled devices, and the diversity of video-related coding standards have expedited the need for efficient video adaptation. On one hand, various kinds of incompatible source coding algorithms that are characterized by different target bit rates and compression techniques need inter-network communications and media gateways. On the other hand, the transmission of video contents is required to accommodate to the different user preferences. Sometimes, adaptation process may also involve issues related to copyright or description problem.

Video adaptation tries to ensure a satisfactory service quality for video delivery. As defined in [1], video adaptation is a process to "transform the input video to an output in video or augmented multimedia form by utilizing manipulations at multiple levels in order to meet diverse resource constraints and user preferences while optimizing the overall utility of video". Video adaptation is not a simple video coding process, and there are some differences between them. First, video adaptation adopts a pre-coded bitstream as input, and the corresponding output is another coded bitstream or correlated multimedia presentation. Next, video adaptation may be deployed in the intermediate location between server and users, which facilitates the independent process of video contents by the server.

Video adaptation can be classified into semantic, structure, and signal levels [1]. In semantic event based adaptation, semantic highlights or events in

bitstream that are defined by the content provider or derived from user preferences are detected. These events are interpreted as segments with the highest semantic level utilities. Analysis results can be utilized for different adaptations. Event information can be used to determine the optimal coding scheme. Important information can be coded with a high quality, whereas unimportant information can be replaced by still visual, text summaries, and/or audio only.

In structure level adaptation, the relation between structural elements is studied. It can be used for different applications. First, key frames extracted from the original bitstream can be used to summarize the information in the shot. Next, image sequences captured by continuous camera can be used to generate a panoramic view.

Below the semantic and structural level is the signal level adaptation. In order to provide seamless service for users, several techniques are designed to solve the mismatch problem in signal level adaptation. Transcoding has been adopted as an efficient tool to realize manipulations of representations and issues of bit allocation. Video transcoder serves as a gateway to convert a pre-coded bitstream to another coded bitstream according to different requirements. This facilitates content providers to generate source information independently. Also for users, they can receive the information they require without further modification. In this thesis, the research is focusing on the signal level adaptation, especially on video transcoding.

In practical applications, the mismatch between content providers and users relies on different aspects. First of all, the bit rate mismatch is one aspect. Traditional video coding methods encode the video at a given bit rate. However, the received video quality becomes very poor if the channel bandwidth is lower

than the target bit rate. Also, some users may require a low bit rate bitstream. To save the bandwidth and fulfill the requirements of users, it is desirable to reduce the bit rate of bitstream for a successful transmission.

Format mismatch is also a serious problem. End users may not have the capabilities to decode the bitstreams coded with different coding standards. And some bitstreams require the decoder with strong computation ability, which is hard to realize in some mobile devices. There are also some needs to transcode a scalable video to a non-scalable video to save the bandwidth.

In addition to format and bit rate mismatch, channel condition may also lead to mismatch. Some bitstreams are coded for error-free network delivery. When such bitstreams are transmitted over an error-prone network, severe video quality degradation will be likely to occur. Some efficient schemes should be applied to solve this problem by adding redundancy information.

In order to solve the above-mentioned problems, video transcoding has attracted a lot of research attention, and many techniques have been proposed to provide efficient and robust video streaming services. However, there are still many problems left to be solved. For example, how to improve the performance of transcoder is still a challenging task. Motivated by this, novel techniques are expected to improve the current transcoding schemes.

1.2 Objectives and Major Contributions

As described in section 1.1, there are overwhelming demands in video content delivery over networks. At the same time, the mismatch between content providers and users is a serious problem that obstructs the seamless commu-

nication. Video transcoding plays an important role in delivering such huge diverse video sources to accommodate to different requirements.

From the perspective of convenience, the easiest way to transcode a pre-coded bitstream is to directly decode and then re-encode it. However, this will incur high computational complexity and quality degradation. Generally speaking, there are two objectives of a transcoder. One is to reduce the computational complexity, which is a necessity of real time transmission. The other is to improve the output quality of the transcoded bitstream. The desired transcoder, which can obtain useful information from the pre-coded bitstream, should be able to outperform the re-encoding process with respect to the above-mentioned two aspects.

Video transcoding techniques and related works have been extensively studied in this work. Furthermore, some algorithms have been developed to further improve the current video transcoding schemes. The main contributions of the present work are summarized below:

1. **Propose arbitrary resizing schemes in transform domain**.

   Arbitrary resizing transcoding is an efficient tool to reduce the overall bit rate by transmitting a smaller size video. In the resizing process, information loss may occur inevitably as downscaling filters are applied. Therefore, low computation and low information loss scheme is desirable. To overcome these problems, two resizing schemes were proposed in this work. One is applicable to inter frame downsizing transcoding, while the other is a more general resizing scheme.

2. **Design frame size selection scheme for arbitrary downsizing transcoding**.

With introducing of arbitrary downsizing, fine gradual reduction of bit rate and video quality becomes feasible. In downsizing transcoding, a suitable reduced frame size is desirable for the transcoded bitstream. Frame size selection is the key issue to realize arbitrary downsizing transcoding. Based on some simulation results, the frame size selection problem is converted to a bit rate estimation problem. Furthermore, the bit rate estimation scheme is solved by estimating the lower bounds of bit rate in different frame sizes. With these lower bounds, a suitable frame size for arbitrary downsizing transcoding can be determined.

3. **Design motion information based frame skipping transcoding scheme**.

   In frame skipping transcoding, one important issue is to reduce the motion information loss and avoid "jerky" effect. Based on the observation from human visual perception, motion change was introduced to detect the scene change in the pre-coded bitstream. It is used to evaluate the motion continuity information. Combining the motion continuity information and the motion activity information, a frame skipping transcoding scheme was designed to minimize the motion information loss, which will benefit the reconstruction of skipped frames.

4. **Propose enhanced intra/inter macroblock mode decision scheme in error resilient transcoding**.

   Intra refreshment is an efficient tool to reduce the error propagation in error-prone network transmission. With the consideration of error propagation to the next frame, an enhanced intra/inter macroblock (MB) mode decision scheme was proposed, which considers consecutive two frames in

a sequence, and takes the error propagation to the next frame into account. Moreover, two cases were discussed according to the availability of the next frame information. The proposed scheme enhances the overall robustness of the transcoded bitstream against the packet loss.

1.3 Organization of the Thesis

This thesis is divided into seven chapters. It is organized as follows.

Chapter 1 contains a brief introduction on the background of video adaptation and the existing problems in video transcoding. The objective and contributions of this thesis are also introduced.

Chapter 2 gives a brief review on the video coding and video transcoding. Related techniques and development are introduced. Four types of transcoding are summarized, i.e. rate-related transcoding, format-related transcoding, error-related transcoding, and information-related transcoding.

Chapter 3 presents the proposed arbitrary image/video resizing schemes. Two schemes are presented to resize picture directly in the DCT domain to save computation and improve quality.

Chapter 4 discusses the realization problem in downsizing transcoding. An efficient scheme is designed to choose a suitable frame size for arbitrary downsizing transcoding.

Chapter 5 studies the motion information within the video. Combining motion continuity with motion activity information, a motion based frame skipping transcoding is presented to determine the optimal frame skipping pattern in the transcoded bitstream.

Chapter 6 presents the research on error resilient transcoding. First, a brief introduction on conventional MB mode decision scheme is given. Then the impact of channel error on the next frame is derived, and the enhanced intra/inter MB mode decision schemes are presented.

Chapter 7 concludes this thesis. Some recommendations on future work are discussed in this chapter.

Chapter 2

Video Transcoding

2.1 Fundamentals of Video Coding

A natural video scene is composed of multiple objects each with its own characteristic shape, color, and texture. The process of digitizing analogue video involves three basic operations of filtering, sampling and quantization. Digital video is a representation of a natural visual scene. Presenting and processing visual content in digital format facilitate the storage and transmission of high quality visual content with low cost.

Video data is either to be saved on storage devices such as CD and DVD or to be transmitted over a communication network. However, the bandwidth requirement of raw video data is quite high, making it impossible to transmit raw video data in practical applications. For the same reason, to store raw video data is also a serious problem. Therefore, video compressing is required for the video storage and delivery.

Video coding is the process of compressing and decompressing a digital video

signal. The encoder converts the source data into a compressed form (occupying a reduced number of bits) prior to transmission or storage, and the decoder converts the compressed form back into a representation of the original video data. Video coding is an important tool for both video storage and delivery.

Video compression is achieved by removing redundant information in video data. Many types of data contain statistical redundancy and can be effectively compressed using lossless compression, so that the reconstructed data at the output of the decoder is a perfect copy of the original data. Unfortunately, lossless compression of image and video information gives only a moderate amount of compression. As a result, lossy compression is preferable to achieve higher compression. In a lossy compression system, the decompressed data is not identical to the source data, so a much higher compression ratio can be achieved at the expense of a loss of visual quality. Lossy video compression systems are based on the principle of removing subjective redundancy, elements of the image or video that can be removed without significantly affecting the viewer's perception of visual quality.

Most video coding methods exploit both temporal and spatial redundancy to achieve compression. In the temporal domain, there is usually a high correlation (similarity) between video frames that were captured at around the same time. Temporally adjacent frames (successive frames in time order) are often highly correlated, especially when the temporal sampling rate (the frame rate) is high. Taking the advantages of this characteristic, motion compensation (MC) technique is applied to reduce temporal redundancy.

In the spatial domain, there is usually a high correlation between pixels (samples) that are close to each other, i.e. the values of neighboring samples

are often very close. In order to reduce the spatial redundancy, transform domain quantization technique and coefficient clipping are adopted. The coding algorithm used in most of the video coding standards is a combination of transform domain quantization technique and motion compensation algorithm.

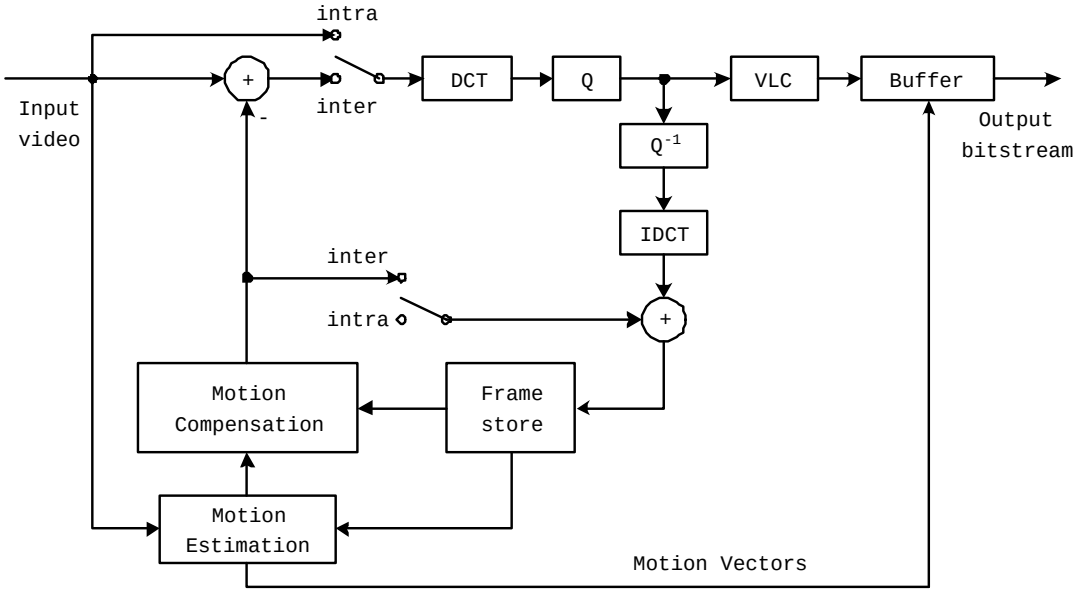


Figure 2.1: Video encoder

Figure 2.1 shows a typical diagram of a video encoder structure. The previous picture is reconstructed in the encoder side by applying inverse discrete cosine transform (IDCT) and dequantization (Q^{-1}), and it is used as the reference frame of the current frame. Motion estimation is used to estimate the motion by matching each MB in the current picture with the reference picture and find the motion vector (MV) that specifies the best matching area. The residue, which is the difference between the current MB and the best matching MB, has much less energy than the original signal. Discrete cosine transform (DCT) and quantization are applied to the residue to further reduce the spatial

redundancy. For intra frame, best predictions are those from the neighboring pixels within the same frame. It is coded independently, and only spatial redundancy is removed. Finally, the generated symbols are converted into bitstream by entropy coding (e.g. variable length coding (VLC)).

A typical decoder structure is shown in Figure 2.2. In the decoder side, inverse operation is applied to the video bitstream to reconstruct the original video sequence. The received bitstream is first decoded by variable length decoding (VLD), then de-quantized, and finally, the frequency domain coefficients are inversely DCT transformed to get the decoded image. For inter frame, the previously decoded images are stored in a buffer as references. The motion compensated image is summed with the decoded residue to generate the final decoded image. Detailed description about these coding techniques can be found in some video coding books [2–4].

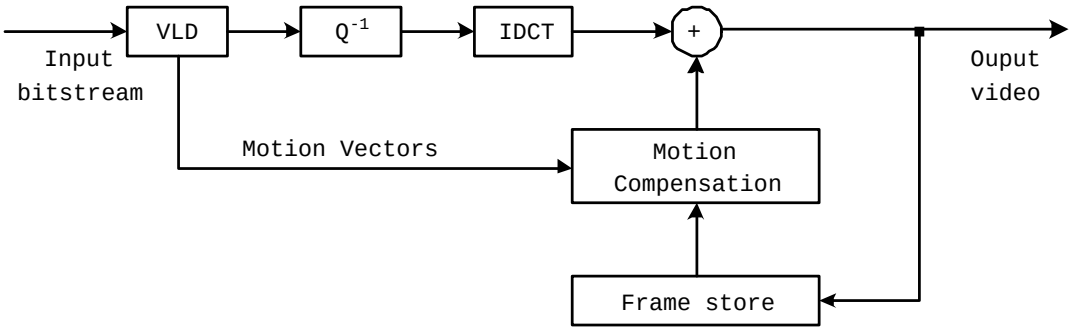


Figure 2.2: Video decoder

Two standard bodies, the International Standards Organization (ISO) and the International Telecommunications Union (ITU), have developed a series of video coding standards that have shaped the development of the visual communication industry (see Figure 2.3).

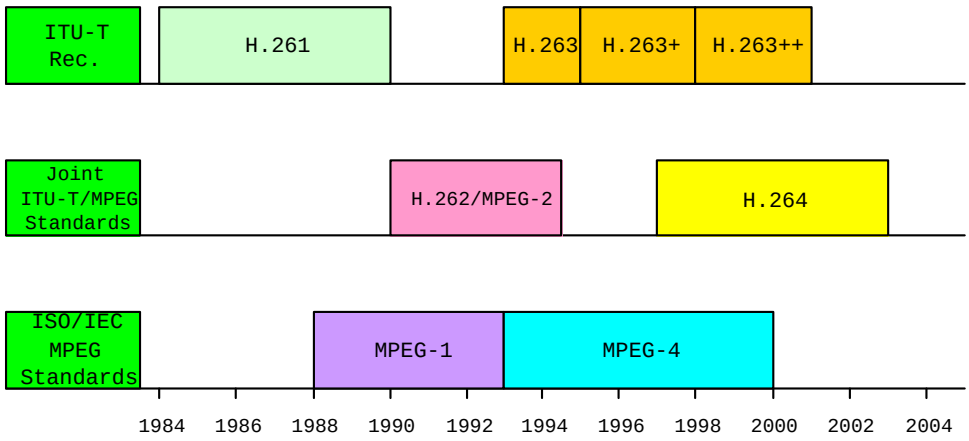


Figure 2.3: Development of coding standards

H.261 is the first standard that is widely used for video conferencing [5]. It was developed by ITU-T to support video telephony and video conference over ISDN channels of $p \times 64kbps$. A combination of inter frame Differential Pulse Code Modulation (DPCM) for minimum coding delay and DCT is used in this standard. H.263 is an ITU-T standard designed for low bit rate communication [6]. Half pixel precision is used for H.263 motion compensation. There are now four options available to improve performance: unrestricted MV's, syntax based arithmetic coding, advance prediction, and forward and backward frame prediction called P-B frames, which is similar to that in MPEG.

MPEG-1 is the first MPEG standard [7]. It is designed for the application of video storage, such as CD-ROMs. It is able to transmit a bit rate of $1.5Mbit/s$. For storage application, coding delay is not a major constraint, so it can be traded for a higher compression efficiency. MPEG-2 aims to support a large potential market, and makes a significant impact in a range of applications such as digital cable TV, digital versatile disc (DVD), and many others [8]. The target bit rate for MPEG-2 is $4-15Mbits/s$. The MPEG-2 standard does cater

for scalability, and it is the first standard to introduce the concepts of Profiles and Levels (defined conformance points and performance limits) as a way of encouraging interoperability without restricting the flexibility of the standard.

The MPEG-4 standard focuses on providing tools and algorithms for efficient storage, transmission and manipulation of video data [9]. The intention of MPEG-4 is to provide a suitable compression scheme for video conferencing, i.e. data rates less $64Kbits/s$. The most fundamental feature in the MPEG-4 standard is the shift towards object based or content based coding, with which a video scene can be handled as a set of foreground and background objects rather than just as a series of rectangular frames.

H.264, known variously as Advanced Video Coding (AVC), MPEG-4 part 10, is a new video compression scheme that is becoming the worldwide digital video standard for consumer electronics and personal computers [10]. In contrast with the highly flexible approach of MPEG-4 Visual, H.264 concentrates specifically on efficient compression of video frames. Key features of the standard include compression efficiency (providing significantly better compression than any previous standard), transmission efficiency (with a number of built-in features to support reliable, robust transmission over a range of channels and networks), and a focus on popular applications of video compression.

2.2 Introduction of Video Transcoding

With the development of telecommunication technology, many user devices are accessible to visual content via different networks: LAN, Digital Subscriber Line (DSL), cable, wireless network, Integrated Services Digital Network (ISDN), and

dial-up. These different access networks have different channel characteristics such as bandwidths, bit error rates, and packet loss rates. At the same time, diverse types of user devices are developed with capability to access to networks. These user devices are characterized with different application purposes. They vary significantly in resources such as computing power and display capability.

To flexibly deliver multimedia data to users with different available resources, through different access networks, and for different applications, multimedia contents need to be adapted dynamically according to the usage environment [1]. The seamless interconnection of various communication networks has become a challenging issue in both research and development areas. Video transcoding has also received its share of attention for the provision of video communication services [3,11–13]. As illustrated in Figure 2.4, a video gateway embedded proxy is located at the interconnection point between different networks. The operating video coding standards within these networks can be different from each other. In such a case, the video proxy performs the necessary syntax translations between different standards in order to achieve the required interoperability.

Video transcoding converts a pre-coded bitstream into another coded bitstream with different format, size, transmission rate, or simply translates it to a new syntax [11]. The tool that makes use of this algorithm to perform the necessary conversion is called a video transcoder. Using transcoding, an improvement in the quality of decoded video can be achieved, and at the same time, the complexity, processing power, and delay incurred by the necessary conversion operations are kept in minimal.

The conventional solution of transcoding has been applied by employing a full decoder/re-encoder pair (see Figure 2.5). This cascaded pixel domain ap-

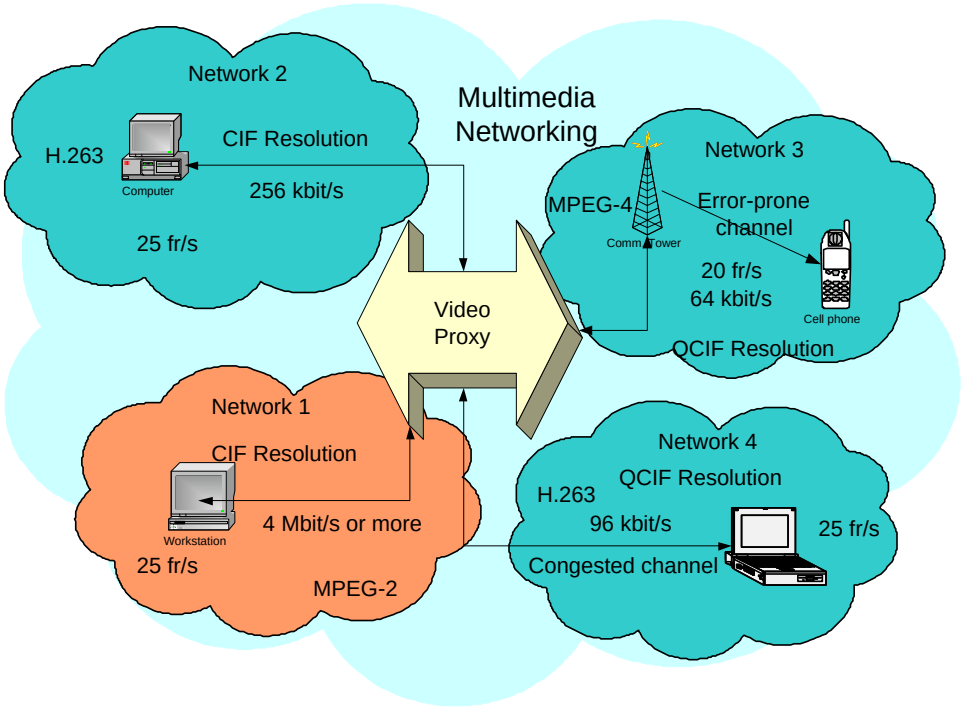


Figure 2.4: Video transcoding environment

proach decodes the input bitstream, performs the appropriate post-processing, and fully re-encodes it subject to new constraints. This scheme involves complex full scale motion estimation operations and DCT/IDCT. Therefore, it has the highest complexity, processing time and power consumption. In addition, motion estimation is performed by using the pictures with reduced quality, thus a low video quality will be presented. So far, many efficient video transcoding schemes have been proposed to reduce the computational complexity and improve the video quality.

In the following, some topics related to transcoding are introduced.

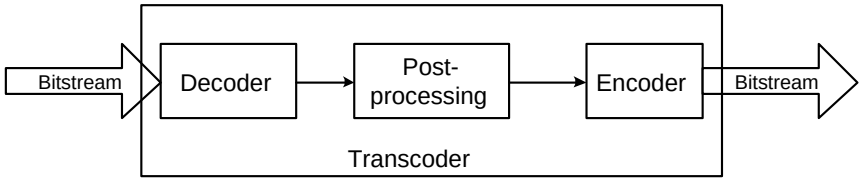


Figure 2.5: Conventional transcoding architecture

2.2.1 MV Reusing in Video Transcoding

Video compression employs motion compensated prediction to reduce the temporal redundancy. Motion estimation has to be carried out for motion compensation. However, motion estimation is a computationally intensive operation. Typically, it consumes 60% of the computation in the encoding process [14]. In video transcoding, MV reusing and re-estimation have been developed to reduce the computation cost on motion estimation [14–19]. Transcoder reuses the MV's from the input bitstream, and therefore, full search for MV's is saved. This increases the speed of software based transcoding by more than 100% [15]. MV reusing is the main tool to reduce the computational complexity in transcoding.

Several schemes have been proposed for reusing MV according to the different applications. When the spatial resolution and the temporal rate of the transcoded bitstream are the same as those of the input bitstream, the MV's in the pre-coded bitstream are very close to those in the transcoded bitstream. For simplicity, the transcoder can reuse them from the pre-coded bitstream directly. However, due to the variation of texture information, the MV's from the input bitstream may not be accurate enough, so they are not optimal for the transcoded bitstream. Directly reusing these MV's lowers the coding efficiency and leads to bad video quality. In order to get accurate MV's at an acceptable cost, MV refinement is employed.

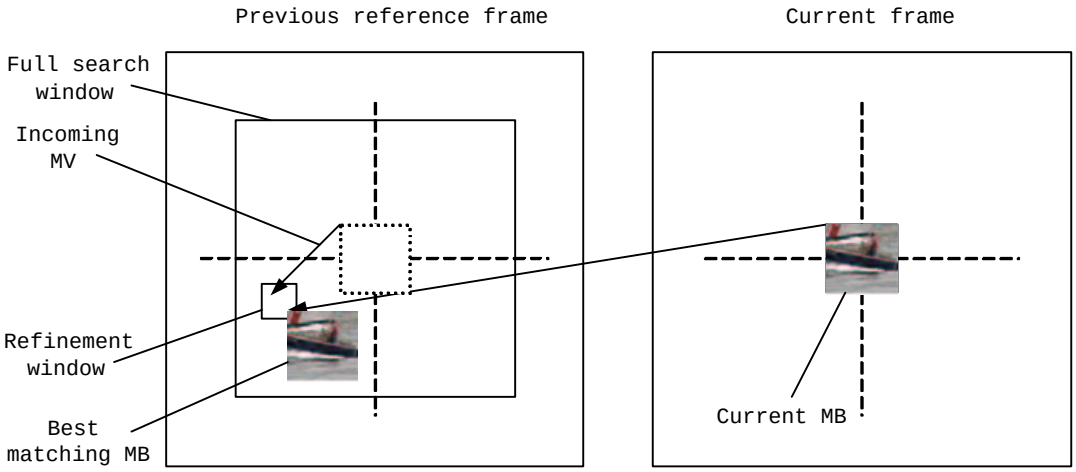


Figure 2.6: Motion vector refinement

In full search motion estimation, to find the MV for an MB in the current frame, the best matching MB is searched within a predefined search window in the previously reconstructed reference frame. This searching range is rather large and it requires quite a lot of computation. For most MB's, the deviation of the optimal MV between the input bitstream and the transcoded bitstream is within a small range. Therefore, compared with applying full scale motion estimation within ± 15 pixels (or larger), the optimal MV can be easily obtained by refining the incoming MV within a small range (see Figure 2.6), and it can still produce almost the same video quality as the full scale motion estimation. In [19, 20], MV is refined within a small range (e.g. ± 2 pixels).

When spatial resolution of the transcoded bitstream is different from that of the input bitstream, MV cannot be reused directly. Multiple MV's in the input bitstream are merged into a single MV in the transcoded bitstream. Adaptive MV re-sampling is proposed for integer downsizing [14]. The final result is obtained by weighting the original MV's toward the worst prediction. Weighting

factor is calculated by counting the number of nonzero AC coefficients.

Scheme for integer downsizing transcoding has been extended to arbitrary downsizing transcoding [18]. All the MV's that concern to a downsized MB are weighted by their related areas. This is because in arbitrary downsizing, some MB's may partially contribute to a final downsized MB. Extensive research has been conducted in this area [18, 21, 22]. Five fast motion re-estimation methods are examined in [22]: "Simple average", "Area-weighted average", "QB-area-weighted average", "Maximum QB-area", and "Median". "Simple average" is realized by averaging all the concerning MV's. "Area-weighted average" is a weighted average of the related areas. In "QB-area-weighted average", QB is a measure of MB coding complexity, which is calculated by multiplying the quantization factors and the generated bits of each MB. It is introduced to calculate the weighting factor based on "Area-weighted average". For "Maximum QB-area", the MV of an MB with the highest QB is chosen as the candidate MV for the estimated result. And in "Median", the MV that has minimal Euclidian distance to other MV's is selected. Among them, "Maximum QB-area" and "Median" outperform other methods when no motion refinement is applied.

In downsizing transcoding, the re-estimated MV is not the optimal one for the downsized bitstream. MV refinement is also specified for downsizing transcoding [17, 23]. Here, the MV is not simply refined by searching within a small search window [-2, +2]. In [17], a localization MV (LMV), which is defined as the average of other MV's except the re-estimated one, is introduced to narrow the search range. LMV specifies a search direction, and refinement is done according to this direction within the refinement window (see Figure 2.7).

In frame skipping transcoding, the frame rate of the input bitstream is re-

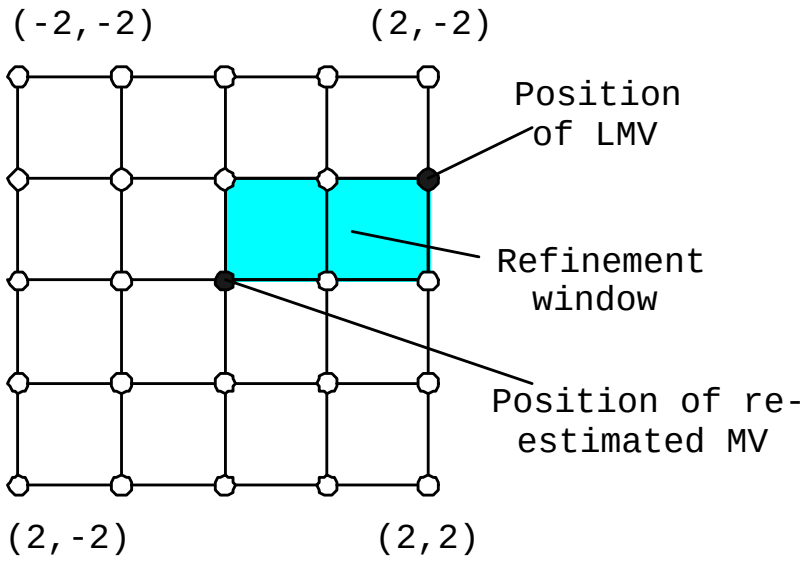


Figure 2.7: Motion vector refinement using LMV

duced. Since some frames are dropped, MV's of the following frames are not valid to be reused directly. Several papers have addressed this problem on how to generate the MV's when the previous frame is skipped [16, 20, 24–26]. Generally, MV's of non-skipped frames are estimated by tracing back to the previously coded frame. It is obvious that this can only generate a baseline MV that is close to the optimal one. With the help of MV refinement, an optimal MV can be reached with less computation.

MB mode decision is another problem related to motion compensation. An MB can be coded with intra, inter, or skipped mode to generate lowest coding cost. Some papers address this problem for re-quantization transcoding, downsizing transcoding, and/or format conversion transcoding [22, 27]. Generally, MB coding mode in the pre-coded bitstream gives a useful indication for the transcoded bitstream. Especially in H.264, rate distortion optimization is introduced to decide the coding mode of each MB. However, the computation cost

of mode decision is rather high. A suitable reuse of the information from the pre-coded bitstream helps to reduce the computation.

2.2.2 Video Transcoding Architectures

The conventional solution of transcoding has been realized by employing a full decoder/re-encoder pair. A typical cascaded pixel domain transcoding architecture (CPDT) is shown in Figure 2.8. A CPDT decodes the input bitstream, performs the appropriate post-processing, and fully re-encodes it subject to new constraints. The computation spending on motion estimation is saved by reusing MV's from the incoming bitstream. Nevertheless, it still requires two motion compensation loops and DCT/IDCT. Therefore, this scheme still involves high computation, high processing time and power consumption.

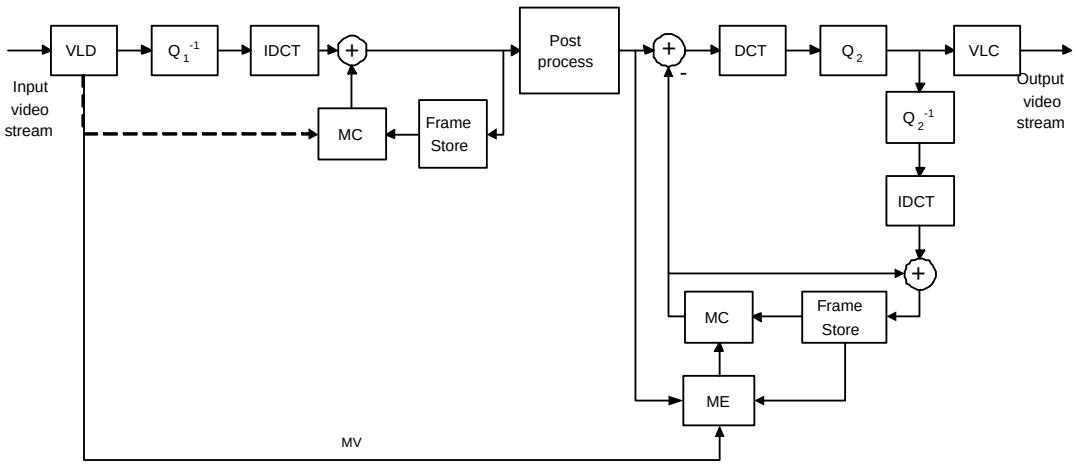


Figure 2.8: Cascaded pixel domain transcoding architecture

Open loop architecture is the simplest architecture with the lowest computational complexity. This approach reuses the MV's and processes the residue separately. Here, the residue is derived by applying necessary operation on

the partially decoded residue information directly. A re-quantization open loop structure is shown in Figure 2.9. In this process, the original video is not reconstructed. Without doing DCT/IDCT, motion estimation, and motion compensation, most of the computation is saved. However, compared with other architectures, the drift problem is serious.

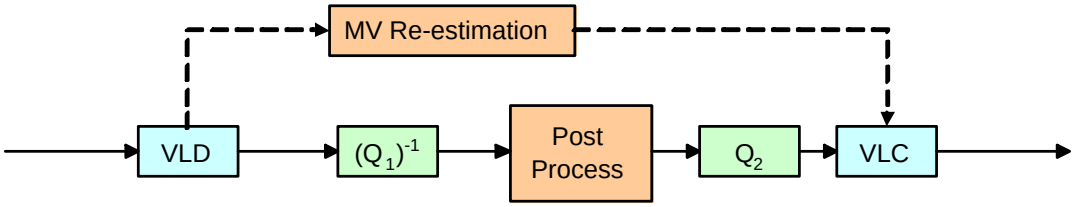


Figure 2.9: Open loop architecture with re-quantization

Drift comes from the non-commutative property of the motion compensation between the encoder and the decoder. When the reference frames in the decoder and the encoder are not exactly the same, drift occurs [28]. This error will further propagate to the following frames until an intra MB/frame is refreshed. It degrades the overall video quality since it is non-compensable. The drift problem is serious in some existing transcoding architectures, and has been analyzed for re-quantization transcoding in [15, 28, 29].

Open loop architecture is not suitable for those applications when no drift-stopping scheme is adopted. In video compression, intra coded frames (I frames) are coded independently. The reconstruction of I frame does not require motion compensation. Therefore, I frame will not be affected by drift. However, the drift from I frame will propagate to other frames that refer to it. Inter coded frames (P frames) require other frames as the reference frame. They may also be referred to by other frames. Therefore, open loop architecture is not suitable

for I or P frames. Bi-directionally prediction coded frames (B frames) are not used as the reference frames for other frames. Any error happening in B frame will not affect other frames. Open loop architecture can be employed to B frame to save the computation without leading to drift. As a result, no serious video quality degradation is occurred.

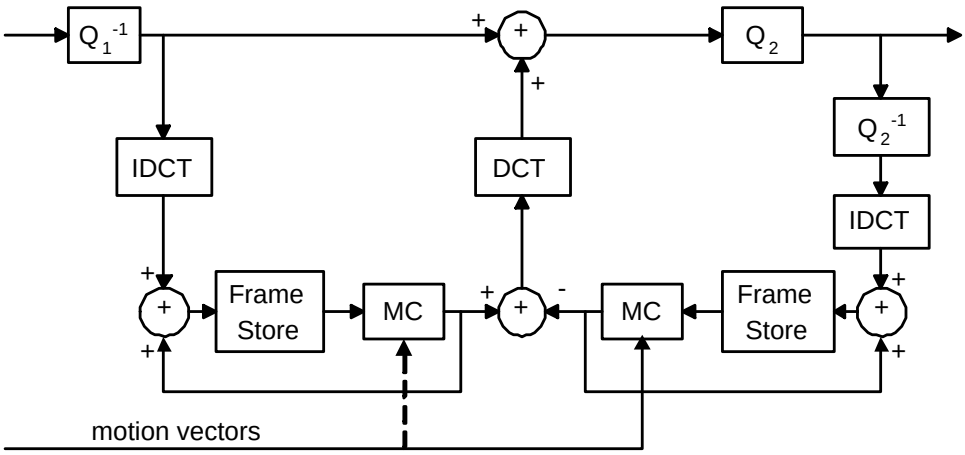


Figure 2.10: Simplified pixel domain transcoding architecture

Several other architectures were proposed to reduce the computation [15, 28, 30, 31]. A simplified pixel domain transcoding structure is illustrated in Figure 2.10. In this transcoder, the incoming bitstream does not need to be fully decoded. Some loops in the cascaded architecture are merged into a feedback loop. This feedback loop accumulates the errors introduced by the different quantization factors and rounding errors, and then adds them together, and finally feeds them back to the next frame. In such a way, error propagation is stopped. As the frame store memories and the motion compensation units are independent, no drift is introduced. In this process, the encoding processing depends on the decoder part, and drift free coding is achieved by applying the

feedback loop. One DCT operation is saved in this architecture.

By utilizing the unitary orthogonal characteristic of DCT, a DCT domain motion compensation method is proposed in [32]. In such a way, it is possible to perform transcoding in the DCT domain directly. The benefits of DCT domain transcoding lie in two folds: better preserving of video quality and partial prioritized processing. The former benefit is drawn from the fact that practical DCT and IDCT implementations use floating point arithmetic, which introduces some round-off errors. These errors can be avoided by performing motion compensation in the DCT domain. Additionally, DCT domain transcoding can distinguish the significant signal components and hence facilitate partial priority channels or limit user processing power, capacity, or time limits.

Several related architectures have been proposed in [29, 33, 34]. A typical DCT domain transcoding architecture is illustrated in Figure 2.11. In this architecture, the transcoding error, which is the difference between the inverse quantized input and output DCT coefficients, is accumulated in the DCT domain and kept in the frame store. After motion compensation in the DCT domain, it is added to the inverse quantized current frame. And the output is generated by applying re-quantization and VLC to this compensated bitstream. In this architecture, all the operation are carried out in the DCT domain, and no DCT or IDCT operation is required.

DCT domain video transcoding is generally more efficient than spatial domain transcoding. However, since the data is organized block by block in the DCT domain, inverse motion compensation becomes the bottleneck for DCT domain approaches. Several novel algorithms has been proposed to overcome this problem [29,33,35–37]. These methods help to reduce the computation and

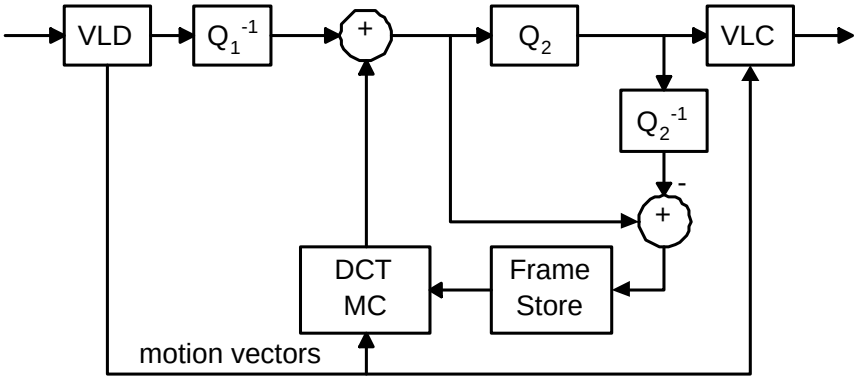


Figure 2.11: DCT domain transcoding architecture

make real time processing applicable.

Special architectures are designed for frame skipping transcoding [38, 39]. These architectures separate the different inputs and specify different transcoding loops for them. Motion compensation is saved in some cases by storing intermediate results without reconstructing the original video.

In [40], four architectures are examined specially for downsizing transcoding: drift compensation in reduced resolution, drift compensation in original resolution, partial encode architecture, and intra refresh architecture. These four architectures reduce some operations that are necessary for drift free transcoding. This results in drift to different extent. Drift compensation and drift stop operations are adopted in these architectures to reduce drift. Drift analysis is also presented for downsizing transcoding in [41]. In [42], DCT domain downsizing transcoding is studied. A new architecture that employs a submacroblock motion compensation method is proposed. This method enables a drift compensation loop to be performed entirely in the reduced resolution. The memory space required for the architecture is significantly reduced since the full resolu-

tion frame buffer is no longer required.

2.2.3 Video Transcoding and Scalable Video Coding

Scalable video coding (layered coding) is another solution to solve the mismatch between the content provider and end users. It was originally proposed to increase the robustness of video codec against packet loss in ATM networks [2,43]. The concept of scalability has now been extended to other areas to accommodate to the variation of network condition. The basic scalable tools in scalable video coding are data partitioning, SNR scalability, spatial scalability, and temporal scalability. Moreover, combinations of these basic scalability tools are also supported and are referred to as hybrid scalability [2].

Scalable video coding is to compress a raw video sequence into multiple bitstreams. One of the compressed bitstreams is a base layer bitstream, which can be independently decoded and provides coarse visual quality; other compressed bitstreams are the enhancement layer bitstreams, which can only be decoded together with the base layer bitstream and provides better visual quality. The combination of all the bitstreams provides the highest video quality. End users with a higher bandwidth could receive more enhancement layer bitstreams, and consequently they could enjoy a higher quality video. Compared with the traditional non-scalable coding method, scalable video coding is more robust over the bandwidth variation networks. The most obvious benefit of scalable video coding is to encode the video only once, and then by simply truncating certain layers or bits from the original stream, lower qualities, spatial resolutions, and/or temporal resolutions could be obtained.

To make a comparison between scalable coding and transcoding is rather

complex since they address the same problem from different points of view. Scalable coding specifies the data format at the encoding stage, which is independent of the transmission requirements, while transcoding converts the existing data format to meet the current transmission requirements. Although scalable coding can provide low cost flexibility to meet the target bit rate, spatial resolution, and temporal resolution, it sacrifices the coding efficiency compared with single layered coding. Considering a cascaded transcoding architecture that fully decodes and re-encodes the video according to the new requirement, its coding performance will be always better than that of the scalable coding.

There are some drawbacks of scalable coding [44, 45]. First, scalable coding generates more layers. This requires more bits to code the enhancement layers. Next, scalable coding requires higher decoding computation. This introduces high complexity in the decoder part. Some users may have low power and low computation requirement, which cannot successfully decode the scalable coded video. Last, scalable coding cannot provide format conversion. This makes it impossible to realize heterogeneous delivery.

However there are still some benefits of scalable video coding for video delivery over networks. First, scalable coding codes the video only once. When the network condition is not good enough to transmit all the layers, the only operation is to discard some enhancement layers. No re-encoding is required. Another important benefit is its application in multicast system. When one bitstream is delivered to multiusers, scalable coding can code the video with multiple layers and deliver them to multiusers with different network connections. However, for video transcoding, a large amount of transcoders are required in the intermediate of the encoder and the decoders to accommodate to the requirements

of end users.

Video transcoding and scalable coding have their different benefits for video delivery. They can co-operate to reach a low cost video delivery as introduced in subsection 2.3.2.

2.3 Classification of Video Transcoding

There are many video transcoding applications. For consideration of simplicity, they are classified into four types according to the purposes and the corresponding operations: rate-related transcoding, format-related transcoding, error-related transcoding, and information-related transcoding.

2.3.1 Rate-Related Transcoding

The main goal of rate-related transcoding is to change the bit rate of the pre-coded bitstream, and present a "best" output video quality. It is the most commonly used scheme in homogeneous transcoding, because there are large amount requirements of bit rate conversion to fulfill the requirements of the bandwidth and the user preference.

Content providers for high bit rate and high resolution video applications, such as DTV and DVD, have already adopted single layer high quality video coding as the default format. Hence, a large number of high quality coded video contents already exist. The pre-coded bitstream is coded with highest quality in the full frame size and high temporal rate. This ensures that the video content is provided with low information loss. Such multimedia content is not appropriate for low bandwidth transmission. When transmitting this

bitstream over bandwidth-limited networks, delay and information loss will be introduced. This leads to severe quality degradation. To solve this problem, the bit rate of the pre-coded bitstream may have to be reduced to accommodate to the network constraint (see Figure 2.12).

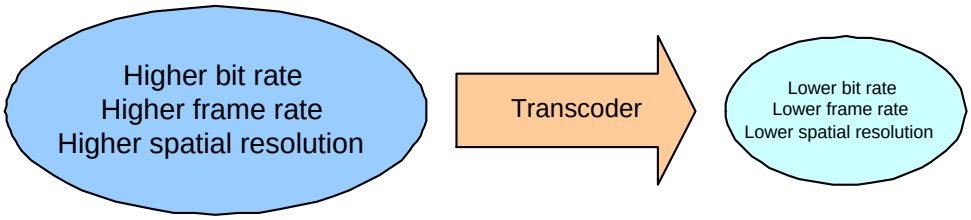


Figure 2.12: Rate-related transcoding

Several techniques can be applied to reduce the bit rate of the pre-coded bitstream. One solution is re-quantization or truncation of the transform coefficients [15, 29, 46–48]. By applying a coarse re-quantization parameter or truncating the high frequency coefficients, more DCT coefficients are set to zero. This reduces the bits allocated to residue. This scheme maintains the spatial and temporal resolution of the original bitstream, and no motion information loss is introduced. It is also easy to realize since the MV from the input bitstream can be reused directly. Several fast transcoding architectures are designed for re-quantization transcoding to save the computation. The drawback of this scheme is that when the quantization parameter is high, artifact may be serious. Although the bit rate can fulfill the bandwidth constraint, the subject visual quality is not acceptable. The target bit rate is not guaranteed by this scheme. Sometimes the generated bit rate cannot reach the target bit rate even when the largest quantization parameter is applied.

Another way to reduce the bit rate is to reduce the temporal resolution of

the video (see Figure 2.13). This is realized by coding fewer frames than the original video [16, 38, 49–53]. Considering a condition of the human eyes called persistence of vision, frame skipping is often used as an efficient scheme to reduce the overall bit rate. Some frames are dropped to improve the spatial quality. Frame skipping allows allocating more bits to the preserved frames, particularly those frames containing more important information. An acceptable quality for coded frames can be maintained and limitations imposed by channel constraints can be satisfied. For those coded frames, a smaller quantization parameter is applied, which reduces the artifact. In frame skipping, motion information may be lost because some frames are skipped, and "jerky" effect may be introduced. In practical application, efficient schemes are needed to reduce the jerky effect in order to provide satisfactory subjective visual quality. Since some frames are dropped, MV's from the input bitstream cannot be reused directly. A new set of MV's has to be derived.

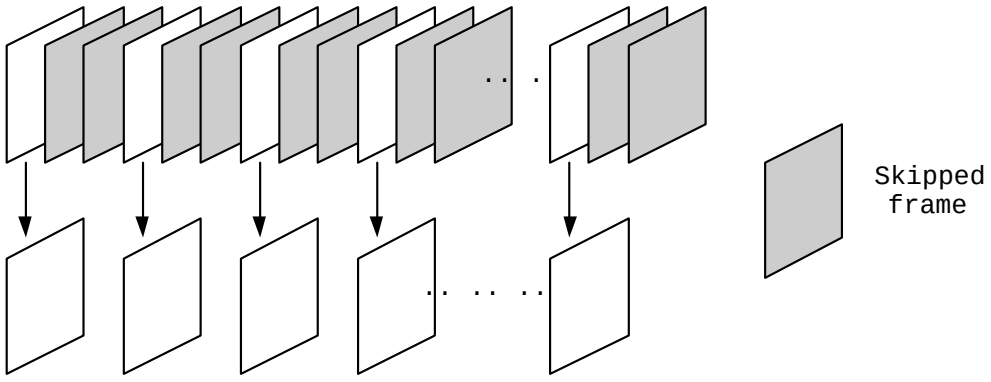


Figure 2.13: Frame skipping transcoding

Downsizing transcoding is also an efficient scheme to reduce the overall bit rate (see Figure 2.14). By transmitting the video with a reduced frame size to the destination, the bit rate required is lowered. In this process, the spatial

resolution of each frame is reduced to generate a smaller resolution due to the constraint of bit rate. In the decoder side, users can upsize the video to the original size. The complexity of this scheme is a bit high for it involves downsizing in the transcoding process. Information will be lost when downsizing is applied. Moreover, a new set of MV's is also required to be generated. However, this scheme can reach very low bit rate without losing any motion information. It can also reach a fine gradual reduction of the video quality. Many researches are devoted in downsizing transcoding to reduce the computation and improve the video quality [14, 18, 21, 35, 40, 50, 51, 54–66].

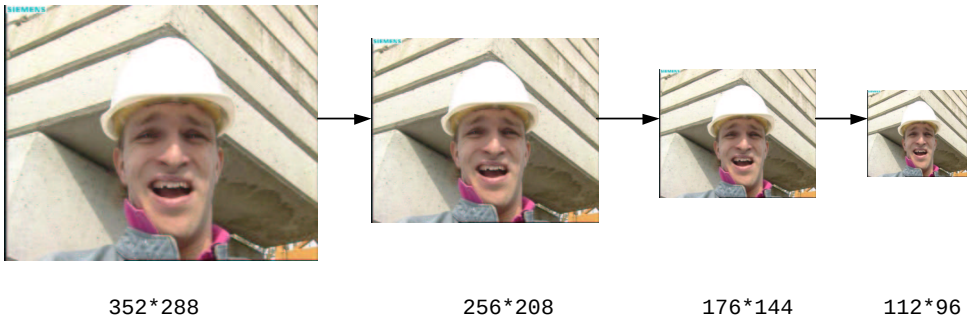


Figure 2.14: Various spatial resolution

2.3.2 Format-Related Transcoding

Another type of transcoding is format-related transcoding. Currently, various video coding standards are available: H.261, H.263, H.264, MPEG-1, MPEG-2, MPEG-4, and so on. In many applications, video coded in one coding standard may need to be converted to another coding standard [23,24,27,34,67–70]. This is because some receiver devices may not have the capability to decode certain type of coding format. For example, video data may be coded with the most recently coding standard, H.264, to get the highest compression ratio. Several

new techniques are used in this standard to improve the compression efficiency. But these new techniques are not recognizable by the former decoder that adopts the previous coding standards. Furthermore, some end devices are designed with low power, low complexity decoding capability. They cannot decode high profile coding format. Under such circumstances, this kind of bitstream must be transcoded to other coding standards according to the requirements of the user.

Apart from transcoding between different coding standards, format conversion is also desirable between the interlaced bitstream and the progressed bitstream. This further increases the requirements of format-related transcoding.

A special case of format-related transcoding is so called transcaling [71–76]. This kind of transcoding is used to convert a non-scalable video into a scalable video, or vice verse, or even conversion between two scalable videos (see Figure 2.15).

Non-scalable video is generally produced at a given target bit rate. When the bit rate is higher than the bandwidth of network, it is very hard to successfully transmit it over networks. When the channel bandwidth is not stable, non-scalable video has to be transcoded to a very low bit rate to accommodate to the lower bound of the bandwidth. However, this results in a waste of bandwidth. The bit rate of the scalable coded video can be reduced by discarding some layers of the video without leading to drift. So, this introduces the need to transcode a non-scalable video to a scalable video (see Figure 2.15(a)). Also, when delivering non-scalable video to multiusers with different bandwidth capabilities, many transcoders are required to transcode the bitstream with different bit rates at the same time. High computation is required in this case. Scalable coding

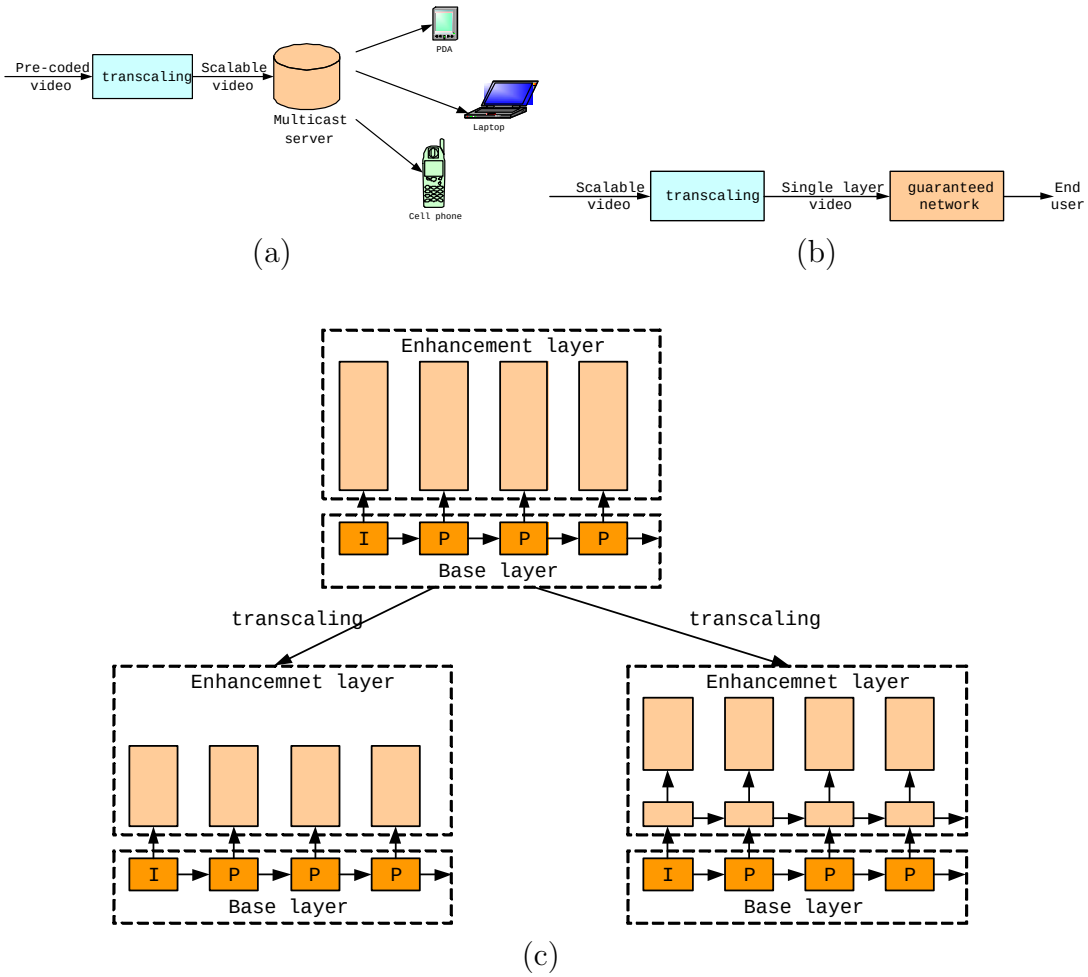


Figure 2.15: Different transcaling operation (a) Scalable video to non-scalable video; (b) Non-scalable video to scalable video; (c) Transcaling between scalable videos

may be a better solution. When the input video is transcoded into a scalable video, only one transcoder is required. This saves the devices and reduces the computation.

However, scalable coding has a lower coding efficiency than non-scalable coding. In a scalable coded bitstream, there is redundant information. When network condition is good, the redundant information may waste the bandwidth and lead to quality degradation. With the same bit rate, single layer coding

presents higher video quality than scalable coding. Transcoding a scalable coded video to a non-scalable coded video is adopted when bandwidth condition is good (see Figure 2.15(b)). This can make full use of the bandwidth and improve the video quality.

Transcaling between scalable bitstreams is used to adjust bit allocation between the base layer bitstream and the enhancement layer bitstreams (see Figure 2.15(c)). This helps to guarantee the base layer video quality and increase the flexibility of enhancement layers. In general, there is an inherent trade-off between the level of scalability and the quality of scalable bitstreams. In other words, the higher the bandwidth varies, the lower the overall video quality of the scalable bitstream will be to support the desired bandwidth range. With transcaling, a scalable video, which covers a given bandwidth range, is mapped into one or more scalable bitstreams covering different bandwidth ranges according to the channel condition.

2.3.3 Error-Related Transcoding

Compression is used to reduce the spatial and the temporal redundancy of the video. In order to reach the highest compression ratio, techniques are developed in the traditional coding system to reduce the redundant information in the video. In this compression process, no transmission error is considered. Dependency between frames is rather high in the generated bitstream, and even dependency between MB's is also high. These kinds of bitstreams are highly sensitive to the channel error. When bit error or package loss happens, it may not only affect the current frame, but also propagate to the following frames. This severely degrades the video quality.

When this kind of bitstreams is transmitted over an error-prone network, it is necessary to reduce the dependency between frames and MB's to generate a more resilient bitstream. As shown in Figure 2.16, error-related transcoding is adopted at the interface of the error-free network and the error-prone network [77–85]. The purpose of error related transcoder is to insert error resilient component to the input bitstream. The transcoded bitstream can be resilient to the error in the network.

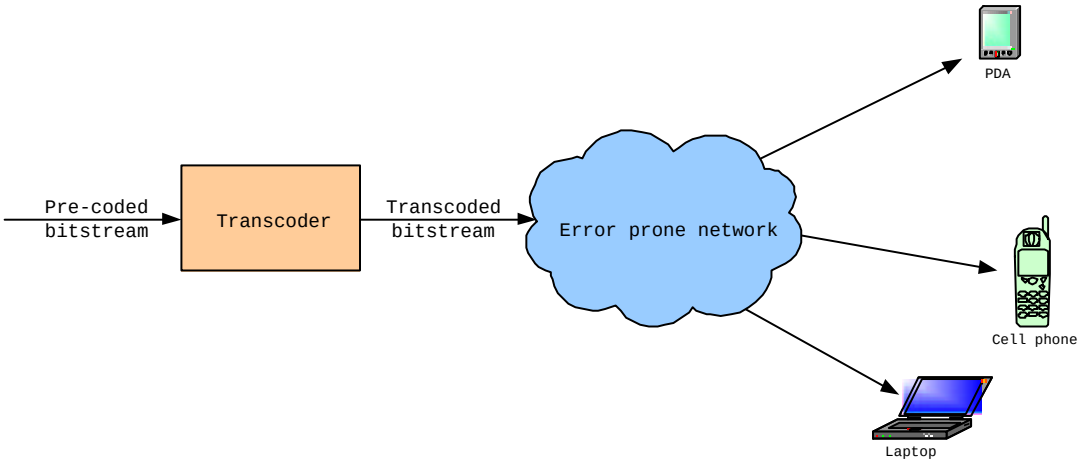


Figure 2.16: Error-related transcoding

Several error resilient tools can be considered in the coding process, e.g. inserting resynchronization marker between different slices, intra MB refreshment, reference picture selection, and so on [86]. These components help to reduce the dependency between frames and MB's.

Channel coding is another way to protect the video free of errors. In channel coding, Forward-Error Correction code (FEC) is used to protect different layers or packets. However, due to the variety of the channel condition, one type of FECs for certain network condition may not be suitable for other network

applications. For example, packet loss is the main error in the wired network. Packet level FEC is designed to provide maximal output video quality. In wireless network, byte level FEC is adopted since bit error is the main error source. A gateway is therefore needed in channel coding level conversion [87]. This is another error-related transcoding application.

2.3.4 Information-Related Transcoding

There are still some other applications of video transcoding, which involve the combination of two and more visual or visual related information to generate a new bitstream. Here we give three examples to illustrate the applications of information-related transcoding.

In practical application, one important issue involves intellectual property management and protection. Actually, the insertion of objects, such as logo, subtitle, and watermark in the compressed domain has faced a growing interest by broadcasting television networks to provide the possibility of inserting copyright information in the pre-encoded bitstreams [88–90]. In this process, additional information is added to the original video (see Figure 2.17(a)). A new bitstream is generated with necessary information included.

Region of interest based transcoding is another type of information-related transcoding. As shown in Figure 2.17(b), the transcoder receives the input bitstream and the instruction from the user. A new bitstream is generated according to the user preferences and the constraints of users display device.

Another application is the multi-point video conferencing [39, 53, 91–95]. Multi-point video conferencing is a natural extension of point-to-point video

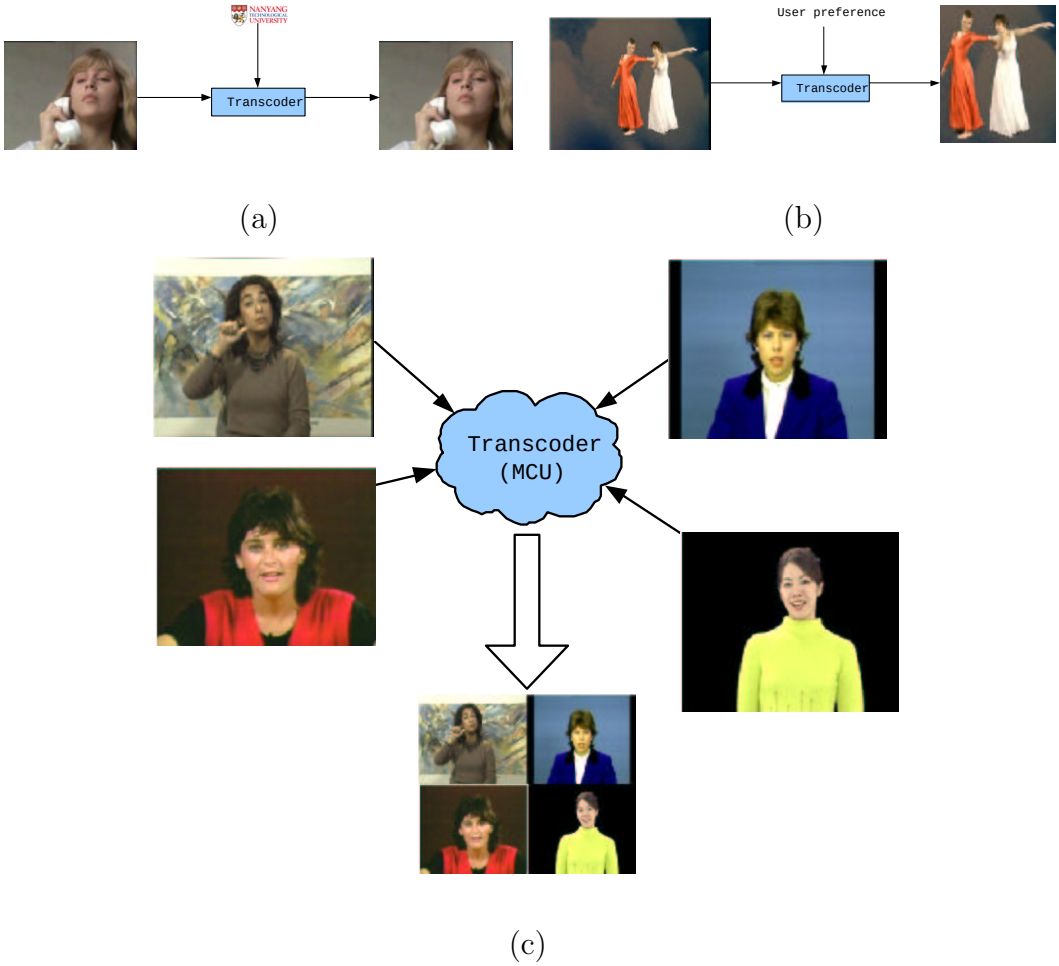


Figure 2.17: Information-related transcoding (a) Logo or watermark insertion; (b) Region of interest; (c) Video conferencing

conferencing. It involves networking, video combining, and multiple coded video signals presenting. In multi-point video conferencing, three or more participants are involved in a teleconference (see Figure 2.17(c)). The conference participants are connected to a Multi-point Control Unit (MCU) in a central office. MCU, which is also referred to as a bridge, is typically used to receive the audiovisual information from the user terminals, process the received signals, and transmit the processed signals to the user terminals. There are two typical ways for an MCU to handle the video signals. In the "switched video" mode, one video

source is selected as the broadcaster and sent to all the terminals except for the broadcaster. In this mode, no transcoding is needed in the MCU except for switching video source. In the "continuous presence" mode, several video signals are combined into one video signal so that participants can view multiple persons simultaneously [91–95].

The same technology can also be used for distance learning, remote collaboration, and video surveillance involving multiple sites.

2.4 Summary

Video transcoding has attracted interests in its diverse application areas. The main purposes of video transcoding are to reduce the computation and improve the video quality. They have to be reached by reusing the information from the input bitstream.

One of most important information to be reused is the MV. As discussed in subsection 2.2.1, reusing the MV's to generate a new series of MV's is an efficient way to reduce the computation. This is only one aspect of MV reusing. In frame skipping transcoding, motion information can be used to detect the scene change. It is a good sign to indicate the importance of frame in temporal rate adjustment. With the consideration of motion information, "jerky" effect can be reduced according to some criteria. This will be further discussed in Chapter 5.

At the same time, there is still other information from the pre-coded bitstream: quantization parameter, bit rate, residue distribution, and so on. These reflect the character of video. They can be used to enhance the coding efficiency

in the transcoding process.

In conclusion, how to reuse the information and how to improve the output video quality according to application requirements are the main interests of video transcoding.

Chapter 3

Block-Based Image/Video Arbitrary Resizing

3.1 Introduction

Today, the communication network connects heterogeneous devices together. Different devices may have different network specifications, and consequently, different demands for video transmission bit rate and quality. To match the bit rate of the video source to the channel constraints and the display screen of end devices, the spatial resolution of the original video needs to be reduced. For example, when the available bandwidth is limited, the spatial resolution of the original bitstream can be downsampled and transmitted via networks, then decoded and upsampled to the original size at the receiver side. Another example is that when display devices of users have different display capabilities. The input bitstream must be resized to accommodate to the screen size of display devices. Image/video resizing has a broad range of applications in multimedia

application environment.

A straightforward approach of downsizing is the bilinear interpolation in the spatial domain. This approach simply calculates the downsized pixel value by interpolating its neighboring pixels. In order to reduce the computation and present a higher picture quality, image scaling in the DCT domain was proposed by using the distributive characteristic of unitary orthogonal transform [32].

In [35], spatial domain bilinear interpolation scheme is converted into DCT domain calculation. Matrix decomposition is used to reduce the computation for matrix multiplication. In [60], the relation between the scaled picture and the original picture is built, and the resizing process is implemented by mapping spatial domain downsizing into its transform domain version. In [56], a fast scheme was proposed to realize resizing directly in the transform domain. The video quality improvement is significant, and the computation is reduced. Following it, a subband DCT based scheme [57] was proposed by extending the algorithm in [56]. In downsizing part, these two methods reach similar result. After that, a method was proposed based on a much bigger block size [61]. All of these methods are designed for 2 to 1 integer resizing.

In practice, due to the versatility of display devices, an arbitrary resizing ratio is expected. Arbitrary resizing enables the entire frame to be displayed in device screen with limited size. To meet the requirements of arbitrary resizing, many schemes have been proposed to resize the original picture with arbitrary ratio. In [63], a scheme was proposed to combine neighboring blocks and generate a much larger block. By zero padding or truncating high frequency coefficients of this big block, a new block is generated. The final result is realized by partitioning this new block into some 8×8 blocks. In [64], L/M-fold

resizing was proposed by upsampling each block to L-fold and then downsizing it by M-fold. In [54], only pixels contributing to the final 8×8 block are extracted. Final result is generated by downsizing these pixels to an 8×8 block. In [62], a scheme was proposed to truncate the original block twice to reduce the computation in arbitrary downsizing, and in [65], an arbitrary resizing was proposed based on binDCT, which is a multiplierless DCT. Although some of these methods can yield a satisfactory result, they give only special cases of the realization. In addition, these schemes are proposed without anti-aliasing consideration. A more general description with theoretical analysis is expected.

In this chapter, two resizing schemes are presented. First, an arbitrary DCT domain downsizing scheme is presented in section 3.2. This scheme utilizes the information of neighboring MB's that contribute to a downsized MB. It can be used to benefit the inter frame resizing. In section 3.3, a more general resizing scheme is proposed based on a two-stage multirate structure.

3.2 Arbitrary DCT Domain Downsizing

3.2.1 Integer Image/Video Downsizing

In spatial domain integer resizing, the most straightforward approach is the bilinear interpolation. This approach simply calculates the downsizing pixel value by interpolating its neighboring pixels (see Figure 3.1). Since the original video is 8×8 blocks, this downsizing operation can be represented in matrix

notation as follows:

$$\mathbf{x} = \sum_{i=1}^{4} \mathbf{h}_i \cdot \mathbf{x}_i \cdot \mathbf{g}_i \tag{3.1}$$

where $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$ are four adjacent 8×8 blocks in the spatial domain, and $\mathbf{h}_i$ and $\mathbf{g}_i$ are the matrices used to extract, downsize, and shift those four adjacent blocks. They are defined in (3.2).

$$\begin{aligned} \mathbf{h}_1 = \mathbf{h}_2 = \mathbf{g}_1^t = \mathbf{g}_3^t = \frac{1}{2} \cdot \begin{bmatrix} \mathbf{P}_{4\times 8} \\ \mathbf{0}_{4\times 8} \end{bmatrix}_{8\times 8} \\ \mathbf{h}_3 = \mathbf{h}_4 = \mathbf{g}_2^t = \mathbf{g}_4^t = \frac{1}{2} \cdot \begin{bmatrix} \mathbf{0}_{4\times 8} \\ \mathbf{P}_{4\times 8} \end{bmatrix}_{8\times 8} \end{aligned} \tag{3.2}$$

where

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{3.3}$$

Many approaches have been proposed to achieve downsizing in the transform domain. By using the distributive characteristic of unitary orthogonal transform, spatial domain operations can be converted to DCT domain manip-

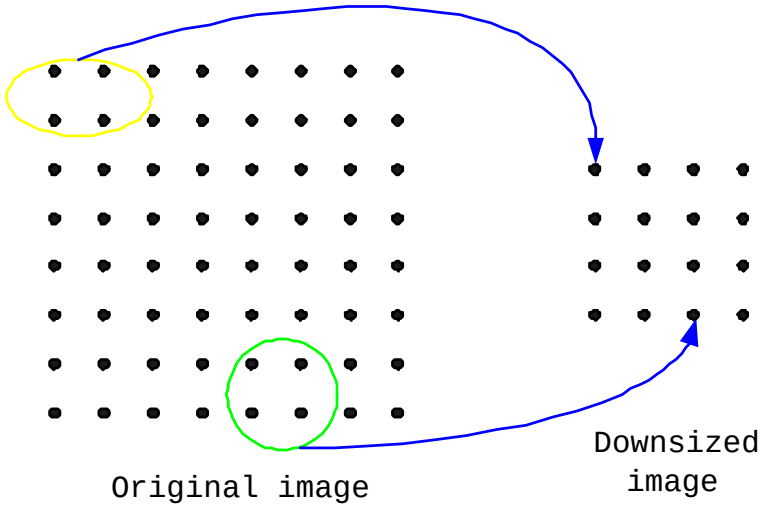


Figure 3.1: Spatial domain integer downsizing

ulations [32]. From (3.1), it can be derived as,

$$\begin{aligned} \mathbf{X} &= DCT(\mathbf{x}) = \sum_{i=1}^{4} DCT(\mathbf{h}_i \cdot \mathbf{x}_i \cdot \mathbf{g}_i) \\ &= \sum_{i=1}^{4} DCT(\mathbf{h}_i) \cdot DCT(\mathbf{x}_i) \cdot DCT(\mathbf{g}_i) = \sum_{i=1}^{4} \mathbf{H}_i \cdot \mathbf{X}_i \cdot \mathbf{G}_i \end{aligned} \tag{3.4}$$

where $\mathbf{X}_i$, $\mathbf{H}_i$, and $\mathbf{G}_i$ are the corresponding DCT domain matrices of $\mathbf{x}_i$, $\mathbf{h}_i$, and $\mathbf{g}_i$.

Some different approaches were also proposed in [56, 57]. However, these approaches can only achieve downsizing by a factor of two, or the downsizing ratio must be an integer. Due to the versatility of display devices, an arbitrary downsizing factor is expected. Next, an arbitrary image/video downsizing method in the DCT domain was proposed. This approach is based on some works mentioned above.

3.2.2 Proposed DCT Domain Arbitrary Downsizing Method

In the spatial domain, for an arbitrary downsizing ratio R, which is defined as the ratio of the original resolution to the desired resolution (note that the horizontal downsizing ratio R_x can differ from the vertical downsizing ratio R_y), more than one pixel in the original frame may contribute to a single pixel in the downsized frame.

As shown in Figure 3.2, one 8×8 output block in the downsized frame can come from as many as $M \times N$ related blocks $\mathbf{a}_{ij}$s in the original frame. $\mathbf{a}_{ij}$s are covered by the supporting area $\mathbf{b}$, which size is $8R_x \times 8R_y$. Therefore, the original frame can be partitioned into these supporting areas. The proposed approach realizes the downsizing in two steps: extracting the supporting area from the original frame, and downsizing it into an 8×8 output block.

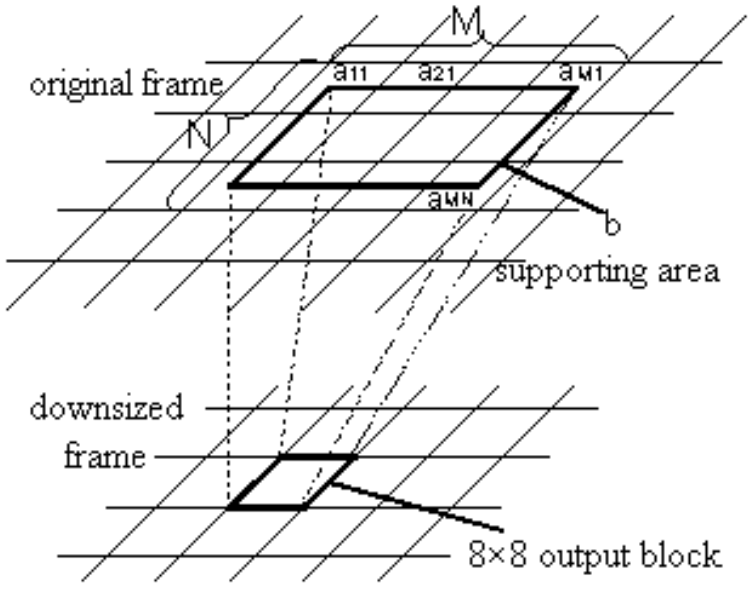


Figure 3.2: Relation between downsized frame and original frame

3.2.2.1 Extracting Supporting Area in the Original Frame

Due to the non-integer downsizing ratio, $\mathbf{a}_{ij}$s could be classified into 9 different cases based on covered locations (see Figure 3.3). They can be fully covered or partially covered by the supporting area. The partially covered blocks have 8 cases, where the overlapped regions locate at the top, bottom, left, right, top left, top right, bottom left, and bottom right portions of the block in the original frame, respectively.

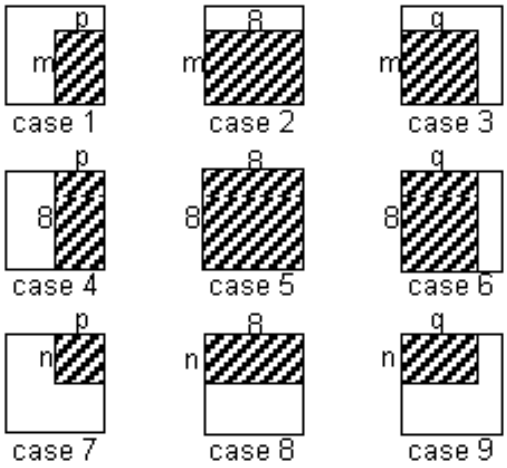


Figure 3.3: Different cases of covered style

In the scenario of integer downsizing, all pixels in the related blocks in the original frame are used to form the output block in the downsized frame. However, when the downsizing ratio is not an integer, not all pixels of these related blocks would be engaged in generating the output block. For instance, if the supporting area covers the bottom right corner of the related block, i.e. case 1 in Figure 3.3, only a portion of the pixels in this block (bottom right $p \times m$ pixels) are used to form the new block.

Assuming that $\mathbf{a}_{ij}$s have already been partially decoded, it is desired to

extract the spatial information of the supporting area $\mathbf{b}$ from $\mathbf{A}_{ij}$s, where

$$\mathbf{A}_{ij} = DCT(\mathbf{a}_{ij}) \qquad i = 1 \; to \; M, j = 1 \; to \; N \tag{3.5}$$

Spatial domain representation can serve as the starting point to build the relation between $\mathbf{b}$ and $\mathbf{A}_{ij}$s. Those related pixels that constitute the supporting area $\mathbf{b}$ are extracted from $\mathbf{a}_{ij}$s. $M \times N$ blocks namely $\hat{\mathbf{a}}_{ij}$ are obtained. The size of $\hat{\mathbf{a}}_{ij}$s may differ and depend on the covered pixels of each related block. For example, for case 6 in Figure 3.3, only the left $q \times 8$ pixels are covered by the supporting area $\mathbf{b}$, then the size of $\hat{\mathbf{a}}_{ij}$ is equal to $q \times 8$. Next, another $M \times N$ block with the size of $8R_x \times 8R_y$ namely $\bar{\mathbf{a}}_{ij}$ is considered. $\bar{\mathbf{a}}_{ij}$s are the zero padding versions of $\hat{\mathbf{a}}_{ij}$s and can be obtained from $\mathbf{a}_{ij}$s as follows:

$$\bar{\mathbf{a}}_{ij} = \mathbf{m}_{ijL} \cdot \mathbf{a}_{ij} \cdot \mathbf{m}_{ijR} \tag{3.6}$$

where $\mathbf{m}_{ijL}$ and $\mathbf{m}_{ijR}$ are the shift matrices used to extract the corresponding matrix $\hat{\mathbf{a}}_{ij}$ from the related block $\mathbf{a}_{ij}$ and pad it into the size of $8R_x \times 8R_y$. Their sizes are $8R_x \times 8$ and $8 \times 8R_y$ respectively.

Then the supporting area $\mathbf{b}$ can be represented by combining $\bar{\mathbf{a}}_{ij}$s as follows:

$$\mathbf{b} = \sum_{i=1}^{M}\sum_{j=1}^{N} \bar{\mathbf{a}}_{ij} = \underbrace{\begin{bmatrix} \hat{\mathbf{a}}_{11} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\bar{\mathbf{a}}_{11}} + \underbrace{\begin{bmatrix} 0 & \hat{\mathbf{a}}_{21} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\bar{\mathbf{a}}_{21}} + \underbrace{\begin{bmatrix} 0 & 0 & \dots & \hat{\mathbf{a}}_{M1} \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\bar{\mathbf{a}}_{M1}} + \cdots + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \hat{\mathbf{a}}_{MN} \end{bmatrix}}_{\bar{\mathbf{a}}_{MN}} \quad (3.7)$$

Due to the distributive property of the DCT, the supporting area $\mathbf{b}$ is extracted in the DCT domain, which is defined as

$$\begin{aligned} \mathbf{B} = DCT(\mathbf{b}) &= \sum_{i=1}^{M}\sum_{j=1}^{N} DCT(\bar{\mathbf{a}}_{ij}) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N} DCT(\mathbf{m}_{ijL} \cdot \mathbf{a}_{ij} \cdot \mathbf{m}_{ijR}) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N} \mathbf{M}_{ijL} \cdot \mathbf{A}_{ij} \cdot \mathbf{M}_{ijR} \end{aligned} \quad (3.8)$$

where $\mathbf{M}_{ijL}$ and $\mathbf{M}_{ijR}$ are the DCT representations of $\mathbf{m}_{ijL}$ and $\mathbf{m}_{ijR}$, respectively.

3.2.2.2 Downsizing the Supporting Area

In a natural image, most of the signal energy is concentrated in the low frequency part in the DCT domain. A reasonable downsizing scheme is to retain only the low frequency components and discard the high frequency components of the

block. Most of the energy of the original block is preserved. As can be seen above, the size of supporting area is $8R_x \times 8R_y$. This supporting area contributes to one 8×8 output block. Discarding the high frequency components and extracting only the low frequency 8×8 part will downsize the block B to 8×8 in the DCT domain as follows:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{I}_8 & 0 \end{bmatrix}_{8\times 8R_x} \cdot \mathbf{B} \cdot \begin{bmatrix} \mathbf{I}_8 \\ 0 \end{bmatrix}_{8R_y \times 8} \tag{3.9}$$

The block $\mathbf{Y}$ is an 8×8 DCT block. No further processing is needed. Then the relation between $\mathbf{A}_{ij}$s and $\mathbf{Y}$ is given by:

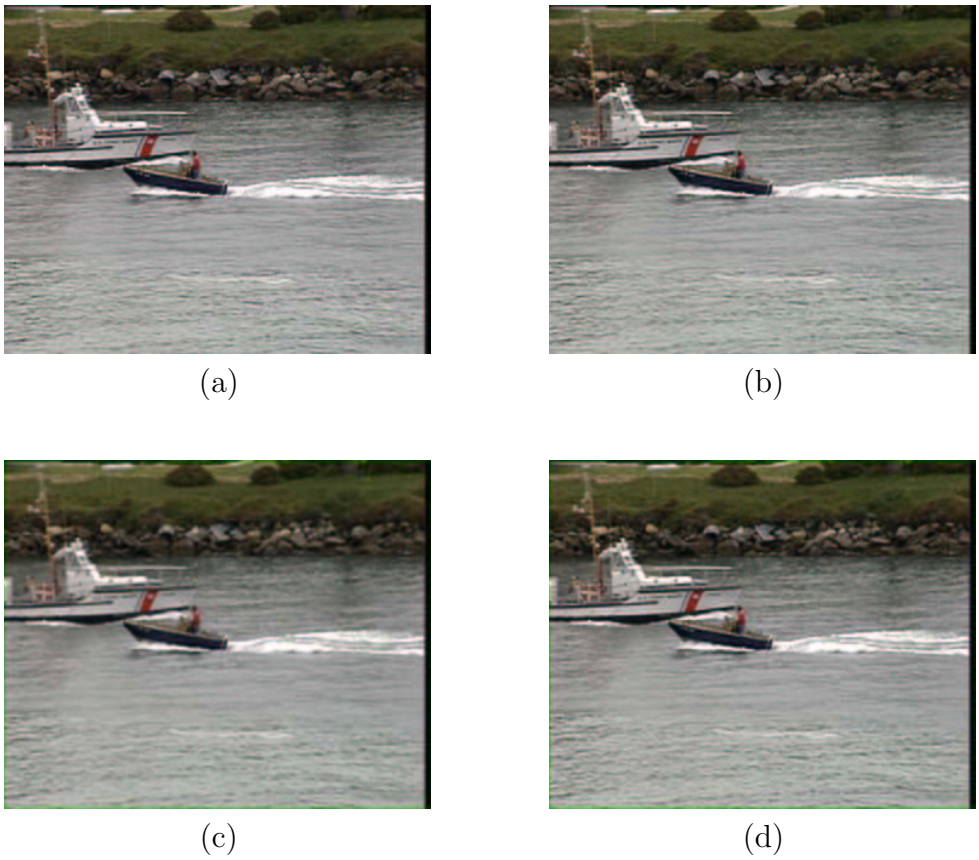
$$\begin{aligned} \mathbf{Y} &= \sum_{i=1}^{M}\sum_{j=1}^{N} \mathbf{H}_{ijL} \cdot \mathbf{A}_{ij} \cdot \mathbf{H}_{ijR} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N} \underbrace{\begin{bmatrix} \mathbf{I}_8 & 0 \end{bmatrix}_{8\times 8R_x} \cdot \mathbf{M}_{ijL}}_{8\times 8} \cdot \mathbf{A}_{ij} \cdot \underbrace{\mathbf{M}_{ijR} \cdot \begin{bmatrix} \mathbf{I}_8 \\ 0 \end{bmatrix}_{8R_y \times 8}}_{8\times 8} \end{aligned} \tag{3.10}$$

Here, downsizing is realized from $8R_x \times 8R_y$ to 8×8 in the DCT domain. The sizes of pre-matrix $\mathbf{H}_{ijL}$ and post-matrix $\mathbf{H}_{ijR}$ are 8×8. These matrices are independent of the input blocks. All possible combinations of them can be pre-computed and stored in the memory. With the look-up-table based implementation, no delay is imposed while processing real time video transcoding.

3.2.3 Experimental Results

In this subsection, experimental results are presented for the proposed arbitrary downsizing algorithm. In the experiments, downsized pictures are upsized to

the original frame size and compared with the original picture. Frame 40 from the video "Coastguard" is shown in Figure 3.4(a). The original frame size is 352×288. This frame is downsized to 256×192, where the horizontal downsizing ratio R_x is $11 : 8$ and the vertical downsizing ratio R_y is $3 : 2$, respectively.



(a) (b)

(c) (d)

Figure 3.4: Coastguard (a) Original; (b) Downsized by the proposed method and upsized by DCT domain zero padding, PSNR is 34.34dB; (c) Downsized by bilinear interpolation and upsized by bilinear interpolation, PSNR is 28.80dB; (d) Downsized by bilinear interpolation and upsized by DCT domain zero padding, PSNR is 30.85dB

For the proposed method, a downsized picture is upsampled to the original size by the DCT domain zero padding scheme. Figure 3.4(b) shows the picture downsized by the proposed method. The PSNR is 34.34dB. Figure 3.4(c)

and Figure 3.4(d) show the pictures downsized by the spatial domain bilinear interpolation. The downsized pictures are upsampled by the spatial domain bilinear interpolation and DCT domain zero padding, respectively. The PSNR is 28.80dB by using spatial domain bilinear interpolation upsizing and 30.85dB by using DCT domain zero padding upsizing. Comparison shows that the proposed DCT domain arbitrary downsizing method can well preserve the most important information of a picture and present a better visual quality. Less information is lost in the downsizing process.

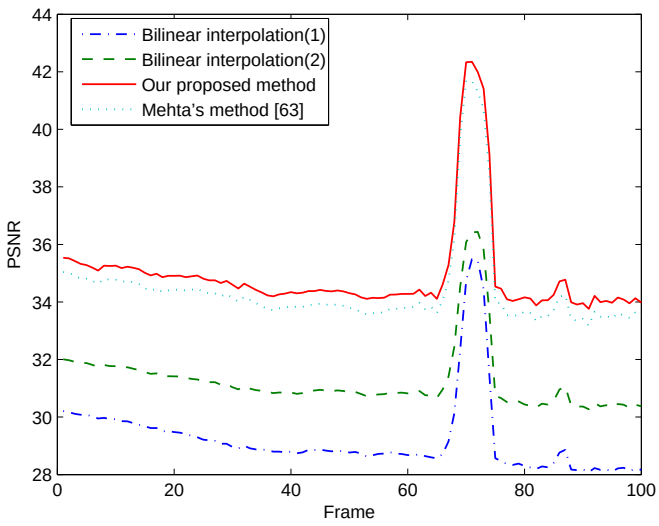


Figure 3.5: The PSNR performances of different downsizing methods on "Coastguard" (Bilinear interpolation downsizing is upsized by bilinear interpolation (1) and DCT domain zero padding (2), respectively.)

Figure 3.5 shows the PSNR comparison of the first 100 frames for the video "Coastguard" by different downsizing approaches. The proposed scheme is compared with bilinear interpolation and Mehta's method [63]. It is clear that there is a significant PSNR improvement in the proposed method.

Some other video sequences are tested to compare the performance. The

Table 3.1: PSNR Comparison for different video sequences downsizing (Bilinear interpolation downsizing is upsized by bilinear interpolation (1) and DCT domain zero padding (2), respectively.)

	Foreman	Suzie	Mobile	Dancer	Coastguard	Flower
Bilinear (1)	32.28	34.83	21.20	35.28	29.24	22.57
Bilinear (2)	32.32	35.05	21.95	36.66	31.27	23.40
Proposed method	36.31	38.35	23.05	39.39	34.94	25.01
Mehta's method [63]	36.02	37.99	22.62	38.89	34.42	24.63

original 100 frames are downsized from 352×288 to 256×192. The downsizing results are shown in Table 3.1, which illustrates that the proposed method outperforms the spatial domain processing.

When compared the computational complexity with the spatial domain method, the computation can be reduced by the proposed method. First, DCT and IDCT can be saved by using DCT domain downsizing. Next, in order to obtain a high quality downsized video in the spatial domain, a long filter length for an ideal low pass filter should be applied to downsize the picture. But the computation will increase depending on the required visual quality. Especially for arbitrary downsizing, the filter matrix may be less sparse. This also involves complicated matrix multiplication, which increases the computation cost in the spatial domain downsizing. Therefore, it is worthwhile to downsize pictures in the DCT domain for a better picture quality.

3.3 Generalized DCT Domain Arbitrary Resizing

In this section, a general description, called two-stage arbitrary resizing (TSAR), is introduced to realize block-based image/video arbitrary resizing. This approach is based on the multistage sampling rate alteration system, and a two-stage structure is adopted with the consideration of computation and flexibility. Considering the property of block-based visual content, some constraints are derived for designing a special filter. These constraints can guarantee the output picture with satisfactory anti-aliasing. In the following, some topics related to TSAR are discussed.

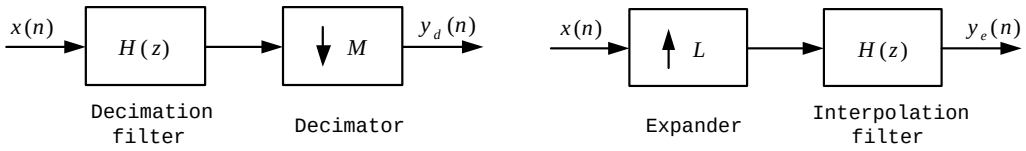
3.3.1 Fundamentals of Multirate System with Multistage Implementation

In a multirate system, rate alteration is realized by the expander or decimator [96, 97]. Figure 3.6(a) shows a complete decimator diagram. A low pass filter is allocated preceding the decimator. It ensures that the signal input to the decimator is bandlimited. This helps to reduce aliasing. The corresponding input-output relation is given by

$$Y_d(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{\frac{1}{M}}W^{-k})H(z^{\frac{1}{M}}W^{-k}) \tag{3.11}$$

where $W = exp(j2\pi/M)$ and $j = \sqrt{-1}$.

Figure 3.6(b) shows a complete expander diagram. An interpolation filter is inserted following the expander to suppress all the images. Similarly, the



(a) (b)

Figure 3.6: Sampling circuit: (a) Downsampling by M; (b) Upsampling by L

input-output relation is shown in (3.12). In image resizing, a single decimator or expander can be used for integer resizing.

$$Y_e(z) = X(z^L)H(z) \tag{3.12}$$

Fractional sampling rate alteration is realized by combining an interpolator with a decimator (see Figure 3.7(a)). This structure realizes an L/M-fold resizing by cascading an L-fold expander with an M-fold decimator. The input-output relation is shown in (3.13). An equivalent form is illustrated in Figure 3.7(b). The corresponding input-output relation is shown in (3.14).

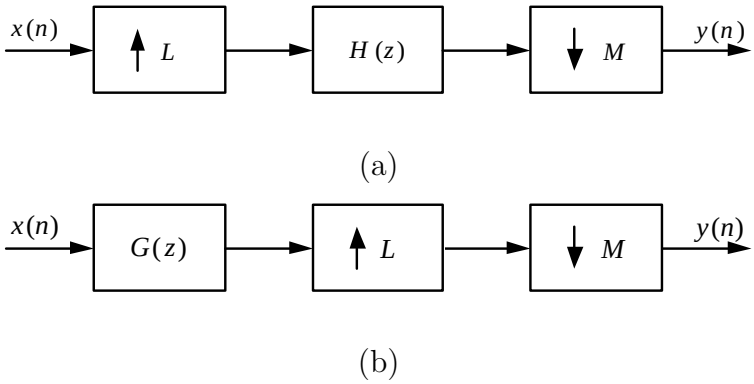


Figure 3.7: Fractional sampling

$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{\frac{L}{M}}W^{-k})H(z^{\frac{1}{M}}W^{-k}) \tag{3.13}$$

$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{\frac{L}{M}}W^{-k})G(z^{\frac{L}{M}}W^{-k}) \tag{3.14}$$

Multistage implementation gives a flexible solution for sampling rate alteration. This structure can also save the computation for some applications. Figure 3.8 shows a three-stage implementation of sampling rate alternation. In multistage implementation, there are two factors to be specified. One is the level of stages, and the other is the parameter selection. The performance of multistage implementation depends on the selection of parameters in this structure. Different parameter sets present different outputs. Suitable selection of the stage level and the parameters is necessary for different applications to reach a compromise between the computation and the flexibility.

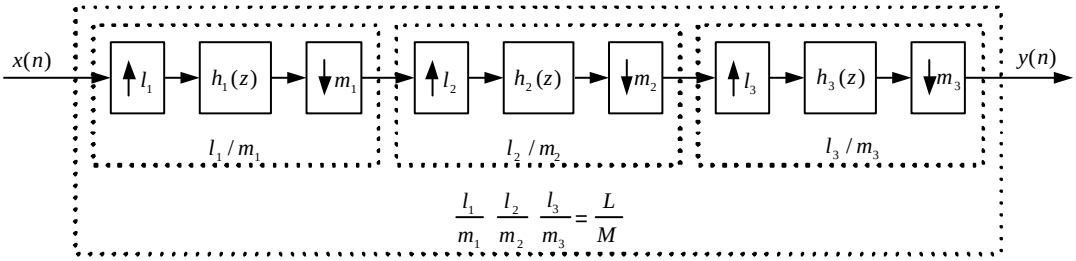


Figure 3.8: Multistage implementation for sampling rate alternation

3.3.2 Two-Stage Realization for Arbitrary Resizing

The multirate system is designed for sampling rate alternation [96, 97]. It can also be used for image resizing. For arbitrary resizing, the resizing ratio is as-

sumed as L/M (L and M are relatively prime integers). In TSAR, a two-stage structure is adopted to realize arbitrary resizing. Although a high stage realization presents higher flexibility, it also introduces high computation when the level of stages increases. Figure 3.9(a) shows a general two-stage structure for arbitrary resizing. In this structure, four parameters are needed to be determined. $L^{'}$ and C_1 are required for the first sampling stage. $M^{'}$ and C_2 are required for the second sampling stage. In the following, the parameter selection for TSAR is introduced.

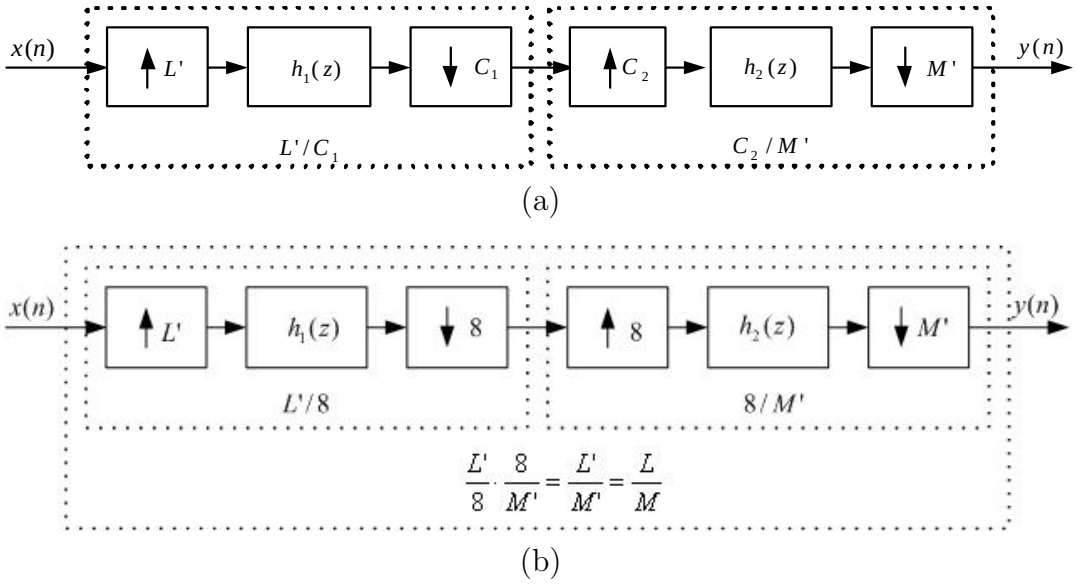


Figure 3.9: Two-stage structure for arbitrary resizing

In video transcoding, the input is a pre-coded bitstream and the available data are 8×8 block based DCT coefficients. From [56] and [61], it is observed that the computation can be saved if one of the parameters in fractional sampling is 8. In a downsampling process, high frequency coefficients of an 8×8 block can be truncated to reach $(8-n)/8$ downsampling (see Figure 3.10(a)), or a big block can be truncated to an 8×8 block to realize $8/(8+m)$ downsam-

pling (see Figure 3.10(b)). Truncating process is done in the transform domain directly. In an upsampling process, an 8×8 block is padded by zeros to realize $(8+m)/8$ upsampling (see Figure 3.11(a)), or a small size block is padded by zeros to an 8×8 block to reach $8/(8-n)$ upsampling in the transform domain (see Figure 3.11(b)).

(a) (b)

Figure 3.10: 8×8 block based downsampling (a) Sampling ratio (8-n)/8; (b) Sampling ratio 8/(8+m)

(a) (b)

Figure 3.11: 8×8 block based upsampling (a) Sampling ratio (8+m)/8; (b) Sampling ratio 8/(8-n)

Based on this observation, two parameters in the two-stage resizing structure are fixed. As shown in Figure 3.9(b), C_1 and C_2 are fixed to a constant value 8. The sampling ratios are $L'/8$ and $8/M'$ in the first and the second stages, respectively. In some applications, the block size of the input signal is not 8×8. The constant parameter C_1 and C_2 may choose other values instead of 8

according to the actual block size. For example, in H.264, the block size is 4×4. When this two-stage structure is considered, 4 will be used instead of 8 as the constant parameter. Moreover, in some applications, the transform function is not DCT. The same structure can still be applied. Only the filter may differ.

As shown in Figure 3.9(b), the total resizing ratio of TSAR is $L^{'}/M^{'}$. The desired resizing ratio is L/M. Then the constraint I is given in (3.15) for the selection of parameter $L^{'}$ and $M^{'}$. If this constraint is satisfied, an input picture is resized to its L/M-fold. This gives a flexible selection of parameters in the realization of arbitrary resizing.

$$Constraint\ I: \qquad L^{'}/M^{'} = L/M. \tag{3.15}$$

In a decimation process, the stretched version of the signal may overlap with its shifted replicas. It is very hard to separate the original signal in the overlapped part. This introduces aliasing when reconstructing the original signal. In order to reduce aliasing, a low pass filter is inserted before the downsampling process to ensure that the signal is bandlimited. This introduces information loss. Upsampling is a process to narrow the frequency band. It helps to reduce aliasing. In TSAR, allocating the upsampling stage before the downsampling stage will reduce aliasing without introducing high information loss. Based on this analysis, the constraint II is given in (3.16) for anti-aliasing.

$$Constraint\ II: \qquad \begin{cases} M^{'} \geq 8 & if\ L < M \\ L^{'} \geq 8 & if\ L > M \end{cases} \tag{3.16}$$

In the downsizing process ($L < M$), there is at least one downsampling stage

in TSAR. It is desired to allocate this downsampling stage in the second stage. $M^{'} > 8$ ensures that one downsampling process is allocated in the second stage. If $M^{'}$ is equal to 8, no resizing is applied in the second stage. In the upsizing process ($L > M$), there exists at least one upsampling process. $L^{'} > 8$ ensures that one upsampling process occurs in the first stage. $L^{'} = 8$ means no resizing is employed in the first stage.

3.3.3 Mathematical Derivation of TSAR

In this part, the relation between the resized picture and the original picture is given. Let us illustrate it with $1-D$ signal in the following three steps. In this process, the input signal is composed of M blocks with the block length of 8. The output signal is composed of L blocks with the block length of 8. The resizing ratio is L/M.

1. In the first sampling stage, the length of the input signal is $8M$. In a downsampling process ($L^{'} < 8$), high frequency coefficients are truncated for each block; in an upsampling process ($L^{'} > 8$), zeros are padded to each block. The output block size is $L^{'}$ and the sampling ratio is $L^{'}/8$. The length of the output signal is $L^{'} \cdot M$. This process is described as,

$$\begin{aligned} \mathbf{X}^{1}_{(L^{'}\cdot M)\times 1} &= \mathbf{P}^{1}_{(L^{'}\cdot M)\times 8M} \cdot \mathbf{X}_{8M\times 1} \\ &= \begin{cases} (\mathbf{I}_M \otimes [\mathbf{I}_8 \ \mathbf{0}_{(L^{'}-8)}]^t) \cdot \mathbf{X}_{8M\times 1} & if\ L^{'} > 8 \\ (\mathbf{I}_M \otimes \mathbf{I}^t_{L^{'}}) \cdot \mathbf{X}_{8M\times 1} & if\ L^{'} < 8 \end{cases} \end{aligned} \tag{3.17}$$

where $\mathbf{X}_{8M\times 1}$ and $\mathbf{X}^{1}_{(L^{'}\cdot M)\times 1}$ are the input signal and the output signal, respectively. $\mathbf{I}$ and $\mathbf{0}$ are the identity matrix and the zero matrix with

the size indicated by their subscripts. $\otimes$ means Kronecker product and $\mathbf{P}^1_{(L'\cdot M)\times 8M}$ serves as a padding or truncating matrix.

2. $\mathbf{X}^1_{(L'\cdot M)\times 1}$ is composed of M blocks with the block size of L'. Because $L'\cdot M = M'\cdot L$, $\mathbf{X}^1_{(L'\cdot M)\times 1}$ is re-grouped into L blocks with the block size of M'. It is realized by applying L' sample IDCT to each block first, then applying M' sample DCT. This process is described by

$$\mathbf{x}^2_{(L'\cdot M)\times 1} = \mathbf{P}^2_{(L'\cdot M)\times(L'\cdot M)} \cdot \mathbf{X}^1_{(L'\cdot M)\times 1} = (\mathbf{I}_M \otimes \mathbf{T}^t_{L'}) \cdot \mathbf{X}^1_{(L'\cdot M)\times 1} \quad (3.18)$$

and

$$\mathbf{X}^3_{(L\cdot M')\times 1} = \mathbf{P}^3_{(L\cdot M')\times(L\cdot M')} \cdot \mathbf{x}^2_{(L'\cdot M)\times 1} = (\mathbf{I}_L \otimes \mathbf{T}_{M'}) \cdot \mathbf{x}^2_{(L'\cdot M)\times 1} \quad (3.19)$$

where $\mathbf{T}^t_{L'}$ and $\mathbf{T}_{M'}$ are L' sample IDCT matrix and M' sample DCT matrix, respectively.

3. In the second step, the re-grouped signal $\mathbf{X}^3_{(L\cdot M')\times 1}$ with length of $L\cdot M'$ is generated. The size of each block is M'. In a downsampling process ($M' > 8$), high frequency coefficients are truncated for each block; in an upsampling process ($M' < 8$), zeros are padded to each block. The sampling ratio is $8/M'$. The length of the output signal is $8L$. This process is described as,

$$\begin{aligned} \mathbf{Y}_{8L\times 1} &= \mathbf{P}^4_{8L\times(L\cdot M')} \cdot \mathbf{X}^3_{(L\cdot M')\times 1} \\ &= \begin{cases} (\mathbf{I}_L \otimes [\mathbf{I}_8\ \mathbf{0}_{(M'-8)}]^t) \cdot \mathbf{X}^3_{(L\cdot M')\times 1} & if\ M' > 8 \\ (\mathbf{I}_L \otimes \mathbf{I}^t_{M'}) \cdot \mathbf{X}^3_{(L\cdot M')\times 1} & if\ M' < 8 \end{cases} \end{aligned} \quad (3.20)$$

Based on these three steps, the output signal is composed of L blocks with the block length of 8. The direct relation between the original picture and the resized picture is shown in (3.21). The equivalent pixel domain realization can be seen in (3.22).

$$\begin{aligned}\mathbf{Y}_{8L\times 1} &= \mathbf{H}_{8L\times 8M}(L^{'},M^{'})\cdot\mathbf{X}_{8M\times 1}\\ &= \mathbf{P}^{4}_{8L\times(L\cdot M^{'})}\cdot\mathbf{P}^{3}_{(L\cdot M^{'})\times(L\cdot M^{'})}\cdot\mathbf{P}^{2}_{(L^{'}\cdot M)\times(L^{'}\cdot M)}\cdot\mathbf{P}^{1}_{(L^{'}\cdot M)\times 8M}\cdot\mathbf{X}_{8M\times 1}\end{aligned} \tag{3.21}$$

$$\begin{aligned}\mathbf{y}_{8L\times 1} &= \mathbf{h}_{8L\times 8M}(L^{'},M^{'})\cdot\mathbf{x}_{8M\times 1}\\ &= (\mathbf{I}_{L}\otimes\mathbf{T}_{8}^{t})\cdot\mathbf{H}_{8L\times 8M}(L^{'},M^{'})\cdot(\mathbf{I}_{M}\otimes\mathbf{T}_{8})\cdot\mathbf{x}_{8M\times 1}\end{aligned} \tag{3.22}$$

Similar concept can be extended to $2-D$ resizing. In (3.23), the input-output relation of $2-D$ resizing is given. Here, the resizing ratio for another dimension is assumed as K/N. The pixel domain relation is given in (3.24). $\mathbf{G}_{8N\times 8K}(K^{'},N^{'})$ can be obtained in a similar way as that of $\mathbf{H}_{8L\times 8M}(L^{'},M^{'})$.

$$\mathbf{Y}_{8L\times 8K}=\mathbf{H}_{8L\times 8M}(L^{'},M^{'})\cdot\mathbf{X}_{8M\times 8N}\cdot\mathbf{G}_{8N\times 8K}(K^{'},N^{'}) \tag{3.23}$$

$$\begin{aligned}\mathbf{y}_{8L\times 8K} &= \mathbf{h}_{8L\times 8M}(L^{'},M^{'})\cdot\mathbf{x}_{8M\times 8N}\cdot\mathbf{g}_{8N\times 8K}(K^{'},N^{'})\\ &= (\mathbf{I}_{L}\otimes\mathbf{T}_{8}^{t})\cdot\mathbf{H}_{8L\times 8M}(L^{'},M^{'})\cdot(\mathbf{I}_{M}\otimes\mathbf{T}_{8})\cdot\mathbf{x}_{8M\times 8N}\\ &\quad\cdot(\mathbf{I}_{N}\otimes\mathbf{T}_{8}^{t})\cdot\mathbf{G}_{8N\times 8K}(K^{'},N^{'})\cdot(\mathbf{I}_{K}\otimes\mathbf{T}_{8})\end{aligned} \tag{3.24}$$

3.3.4 Anti-Aliasing and Computation Analysis

3.3.4.1 Anti-Aliasing Analysis

In this part, the anti-aliasing analysis for TSAR is discussed. Aliasing comes from the overlap of the stretched version of the signal with its shifted replicas in decimation. When a downsampled signal is upsampled to reconstruct the original signal, aliasing may degrade the output quality. In image resizing process, anti-aliasing is necessary. In order to reduce aliasing, a low pass filter is applied to the input signal to ensure that the signal input to the decimator is bandlimited. For different purposes and expectations, the performance of this low pass filter is specified differently.

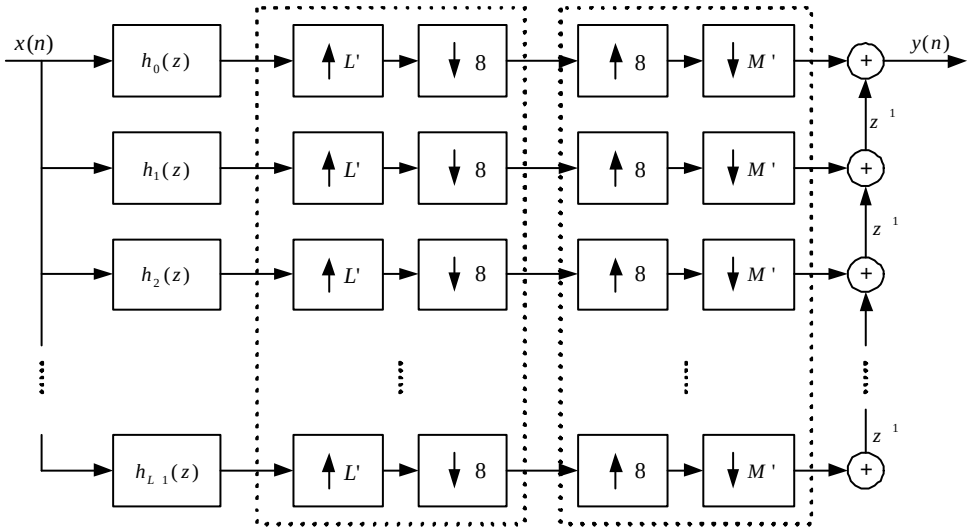


Figure 3.12: Filter bank structure for TSAR

Similar to [56], a filter bank realization of TSAR is shown in Figure 3.12. It is a uniform sampling rate filter bank. Based on this realization, the z-transform description of the output signal is given by using standard results from the multirate signal processing [96,97]. In (3.25), z-transform of the output is given

based on $2-D$ signal description.

$$\begin{aligned}
\mathbf{Y}(z_0,z_1) &= \sum_{l=0}^{L-1}\sum_{k=0}^{K-1} z_0^{-l}\cdot z_1^{-k}\cdot\frac{1}{M}\cdot\frac{1}{N} \\
&\quad \cdot\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\mathbf{X}(W_M^{-m}\cdot z_0^{\frac{L}{M}}, W_N^{-n}\cdot z_1^{\frac{K}{N}})\cdot\mathbf{H}_l(W_M^{-m}\cdot z_0^{\frac{L}{M}})\cdot\mathbf{G}_k(W_N^{-n}\cdot z_1^{\frac{K}{N}}) \\
&= \frac{1}{M}\cdot\frac{1}{N}\cdot\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\mathbf{X}(W_M^{-m}\cdot z_0^{\frac{L}{M}}, W_N^{-n}\cdot z_1^{\frac{K}{N}}) \\
&\quad \cdot\sum_{l=0}^{L-1} z_0^{-l}\cdot\mathbf{H}_l(W_M^{-m}\cdot z_0^{\frac{L}{M}})\cdot\sum_{k=0}^{K-1}\cdot z_1^{-k}\cdot\mathbf{G}_k(W_N^{-n}\cdot z_1^{\frac{K}{N}}) \\
&= \frac{1}{M}\cdot\frac{1}{N}\cdot\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\mathbf{X}(W_M^{-m}\cdot z_0^{\frac{L}{M}}, W_N^{-n}\cdot z_1^{\frac{K}{N}}) \\
&\quad \cdot\mathbf{F}_m^H(W_M^{-m}\cdot z_0^{\frac{L}{M}})\cdot\mathbf{F}_n^G(W_N^{-n}\cdot z_1^{\frac{K}{N}}) \qquad (3.25)
\end{aligned}$$

where $W_M = exp(j2\pi/M)$, $W_N = exp(j2\pi/N)$, $j=\sqrt{-1}$,

$$\mathbf{F}_m^H(z_0) = \sum_{l=0}^{L-1} z_0^{-l\cdot\frac{M}{L}}\cdot\mathbf{H}_l(z_0)\cdot W_M^{-\frac{M}{L}\cdot m\cdot l} \qquad (3.26)$$

and

$$\mathbf{F}_n^G(z_1) = \sum_{k=0}^{K-1} z_1^{-k\cdot\frac{N}{K}}\cdot\mathbf{G}_k(z_1)\cdot W_N^{-\frac{N}{K}\cdot n\cdot k} \qquad (3.27)$$

It is observed that $\mathbf{F}_m^H(z_0)$ and $\mathbf{F}_n^G(z_1)$ have the similar structure. In TSAR, they are separable. For this reason, only the discussion based on one dimensional analysis is given.

For a clearer illustration, an example based on 2/5 downsizing is presented. According to the constraint I, the following different parameter sets to realize 2/5 downsizing are chosen.

- *Case 1.* $L^{'} = 2$ and $M^{'} = 5$;
- *Case 2.* $L^{'} = 4$ and $M^{'} = 10$;
- *Case 3.* $L^{'} = 6$ and $M^{'} = 15$;
- *Case 4.* $L^{'} = 8$ and $M^{'} = 20$;
- *Case 5.* $L^{'} = 16$ and $M^{'} = 40$.

Obviously, case 1 does not satisfy the requirement of the constraint II. It is listed here to show why the constraint II is necessary for anti-aliasing. The filter in Figure 3.12 is defined in (3.22). $h_l(n)$ $(0 \leq l < 8L)$ are $8M$-tap filters. The frequency responses of them for case 2 are shown in Figure 3.13. It can be observed that each of them serves as an equivalent low pass filter.

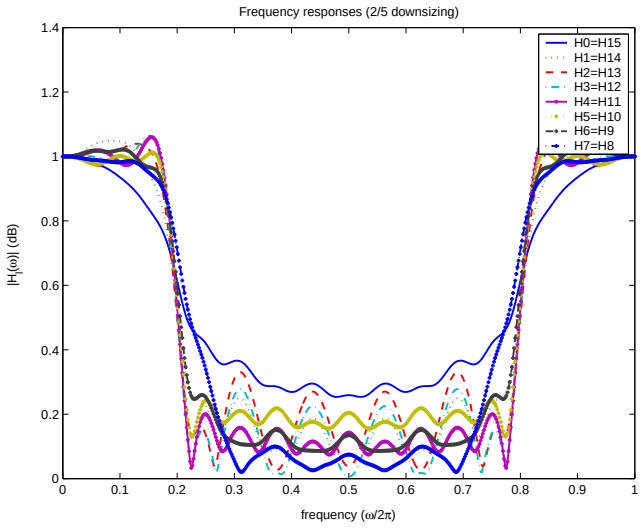
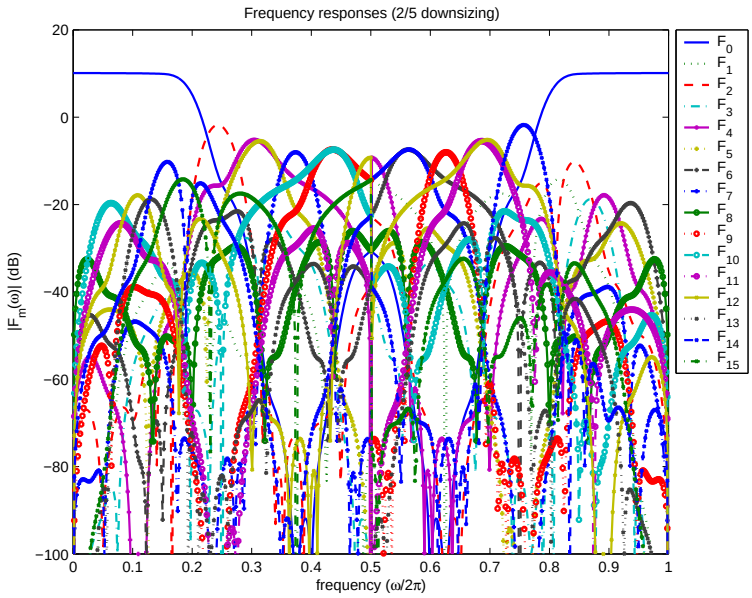
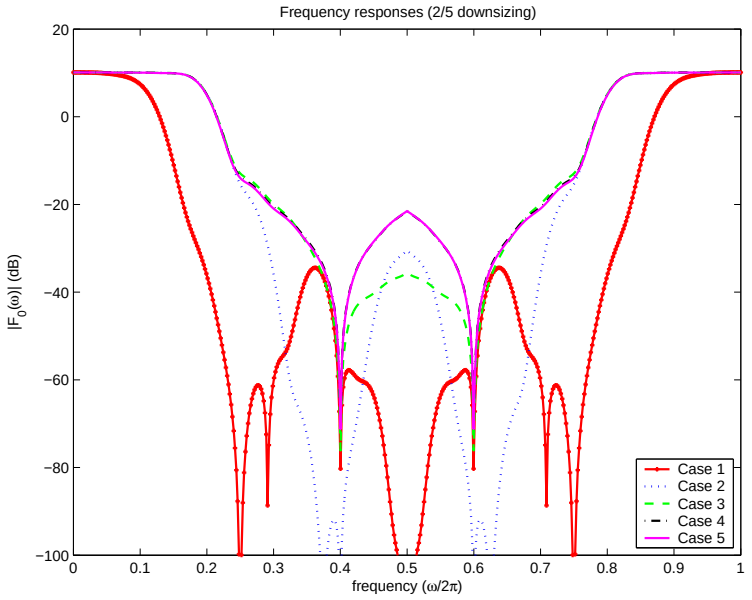


Figure 3.13: Frequency responses of $H_l(\omega)$ (2/5 downsizing)



(a)



(b)

Figure 3.14: Frequency responses (2/5 downsizing) (a) $\mathbf{F}_m(\omega)$ (case 2); (b) $\mathbf{F}_0(\omega)$

Observed from (3.25), $\mathbf{F}_m^H$ and $\mathbf{F}_n^G$ build a direct relation between the input $\mathbf{X}$ and the output $\mathbf{Y}$. Figure 3.14(a) gives the frequency responses of $\mathbf{F}_m^H$ for case 2. It is obvious that $\mathbf{F}_0$ is the dominant signal in the low frequency part. In Figure 3.14(b), only the performances of $\mathbf{F}_0$ between these five cases are compared. For 2/5 downsizing, $\mathbf{F}_0$ is expected to have an ideal low pass property with the pass band from 0 to $2\pi/5$. In Figure 3.14(b), case 2-5 present a better low pass property in $0 \leq \omega \leq 2\pi/5$. The transition region is also narrow.

For case 1, its pass band is around $0 \leq \omega \leq \pi/4$. More information is discarded by this filter. This leads to a bad picture quality when reconstructing the full size picture. In case 1, $M^{'}$ is less than 8. It is an upsampling process in the second stage. This results in $L^{'}/8 < L/M$, which means that the downsampling ratio in the first stage is higher than the overall downsizing ratio. More information is lost in the first stage than necessary.

From the example given above, it can be summarized that if the parameter selection satisfies the constraint II, TSAR can satisfy the anti-aliasing requirement.

3.3.4.2 Computational Complexity Analysis

In the realization of TSAR, two schemes are considered.

The straightforward consideration is applying brute force matrix multiplication as shown in (3.23). In this method, for each output pixel, the number of multiplications required is $8N(M+K)/K$ or $8M(N+L)/L$. The number of additions required is $(8MN+8KN-K-N)/K$ or $(8MN+8LM-L-M)/L$. When parameter M, N, K, and L are small, the computation introduced may also be small.

However, even with some fast matrix multiplication algorithms introduced, the computation of the brute force matrix multiplication is high. The steps described in subsection 3.3.3 with DCT and IDCT can be considered. For simplicity, $1-D$ resizing is discussed here. For an L/M-fold resizing process, the input is $8M$ pixels and the output is $8L$ pixels. It requires M $L^{'}$-sample IDCT and L $M^{'}$-sample DCT. Generally, the smaller $L^{'}$, $M^{'}$, $K^{'}$, and $N^{'}$ are, the lower computation will be involved. With the introducing of some fast DCT algorithms [98, 99], the computation cost can be reduced significantly.

It should be noted that, with these fast DCT algorithms, large sample DCT does not means high computation. According to [99], 7-sample DCT requires 18 multiplications and 24 additions. However, 9-sample DCT requires only 8 multiplications and 34 additions. In TSAR, suitable selection of parameter sets can save the computation by using these fast algorithms. The selection of parameter sets is a tradeoff between the computation and the video quality.

3.3.5 Illustrative Examples

In this part, a few illustrative examples are presented to show the performance of TSAR with the analysis of anti-aliasing performance and output picture quality.

At first, a concept used in experiments is introduced: "resizing set". In the comparison of resizing results, it is needed to resize them back to the original size. In TSAR, an inverse resizing process is available similar to the forward resizing process. In Figure 3.15(a), the parameter set $L^{'}$ and $M^{'}$ is adopted for this two-stage resizing structure. In the inverse resizing process, corresponding parameter set will also be adopted (see Figure 3.15(b)). These two structures together compose a "resizing set". In the following examples, this "resizing set"

is used in the result comparison. The resizing order is downsizing first, and upsizing next to reconstruct the picture to the original size.

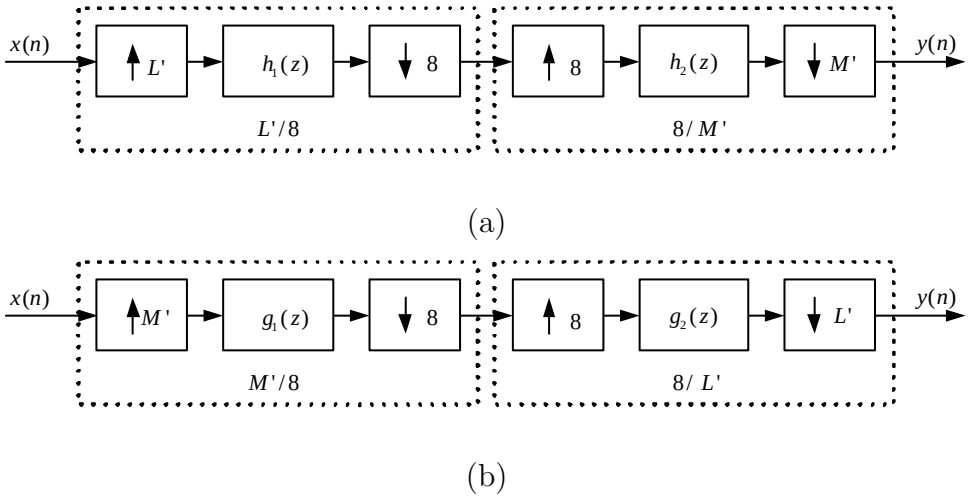


Figure 3.15: Resizing set (a) Forward resizing; (b) Inverse resizing

It is true that, the parameter selection for inverse resizing will affect the final result. This is because inverse resizing is a two-stage structure for arbitrary resizing too. There are downsampling and upsampling in this process. Aliasing and image effect are also introduced. However, it is the optimal solution to reconstruct the downsized video, and has been proved in [100].

3.3.5.1 2/5 Resizing Set

For 2/5 resizing set, the selection of parameter sets has already been introduced in subsection 3.3.4. Corresponding frequency responses are also shown in Figure 3.13 and Figure 3.14. Methods proposed in [63] and [64] are realized by TSAR with parameter sets of case 4 and case 5, respectively.

PSNR comparisons are shown in Table 3.2. It is found that with the increasing of $L^{'}$ and $M^{'}$, the picture quality will not be improved significantly.

Table 3.2: PSNR comparison under 2/5 resizing set

Resizing scheme	L'	M'	PSNR (dB)				
			Lena	Baboon	Barb	Boat	Peppers
case 1	2	5	28.313	21.926	24.828	25.929	27.552
case 2	4	10	33.436	24.199	26.269	30.571	32.052
case 3	6	15	33.602	24.227	26.313	30.675	32.12
case 4	8	20	33.579	24.21	26.3	30.652	32.1
case 5	16	40	33.598	24.22	26.298	30.669	32.114

Case 2-5 have similar performance. For the frequency responses of $\mathbf{F}_0$, case 2-5 present the similar low pass performance. But the computation cost for them may differ. If the lowest computation is required, case 2 presents a satisfactory solution.

Subjective comparison is given in Figure 3.16 and Figure 3.17. Figure 3.16 shows the downsized pictures of "Lena" under different parameter sets, and Figure 3.17 shows the pictures after reconstruction by inverse resizing. Except case 1, all parameter sets present similar picture quality.

3.3.5.2 3/4 and 2/3 Resizing Sets

First, PSNR comparisons are given in Table 3.3 and Table 3.4 for these two resizing sets, respectively. Corresponding parameter sets are specified in the same table. PSNR comparisons confirm the conclusion given above. When the constraint II is satisfied, the qualities of the output picture are similar and acceptable. For 3/4 resizing set, case 2-5 are all acceptable, and for 2/3 resizing set, only case 3-5 satisfy the requirement.

Frequency responses are shown in Figure 3.18 and Figure 3.19. For these two resizing processes, the ideal pass bands of $\mathbf{F}_0$ are $0 \sim 3\pi/4$ and $0 \sim 2\pi/3$, respec-

Table 3.3: PSNR comparison under 3/4 resizing set

Resizing scheme	L'	M'	PSNR (dB)				
			Lena	Baboon	Barb	Boat	Peppers
case 1	3	4	31.921	23.561	26	29.28	30.741
case 2	6	8	42.988	31.5	32.813	39.154	40.418
case 3	9	12	43.404	31.811	33.198	39.488	40.786
case 4	12	16	43.177	31.701	33.071	39.314	40.662
case 5	24	32	42.895	31.572	32.929	39.104	40.509

Table 3.4: PSNR comparison under 2/3 resizing set

Resizing scheme	L'	M'	PSNR (dB)				
			Lena	Baboon	Barb	Boat	Peppers
case 1	2	3	28.313	21.926	24.828	25.929	27.552
case 2	4	6	35.35	25.562	27.063	32.212	33.536
case 3	6	9	40.801	29.526	30.562	36.802	38.364
case 4	8	12	40.784	29.456	30.442	36.775	38.243
case 5	16	24	40.653	29.424	30.428	36.702	38.189

(a)

(b)

(c)

(d)

(e)

Figure 3.16: Lena 2/5 downsizing (a) case 1; (b) case 2; (c) case 3; (d) case 4; (e) case 5

(a) (b) (c) (d) (e) (f)

Figure 3.17: Lena (2/5 resizing set) (a) original; (b) case 1 (28.313db); (c) case 2 (33.436dB); (d) case 3 (33.602dB); (e) case 4 (33.579dB); (f) case 5 (33.598dB)

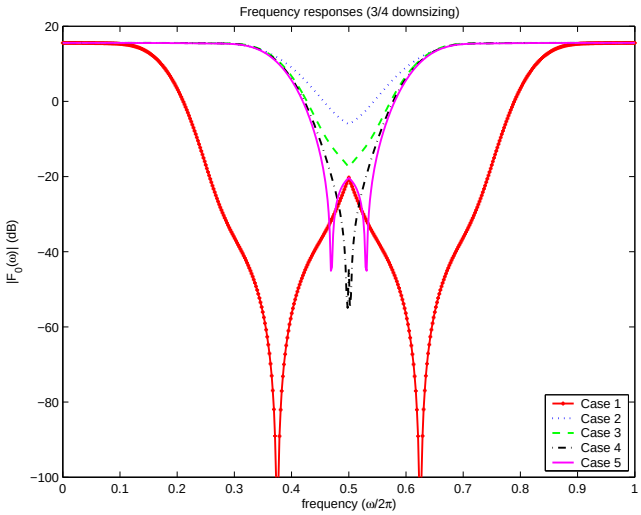


Figure 3.18: Frequency responses of $\mathbf{F}_0(\omega)$ in 3/4 downsizing

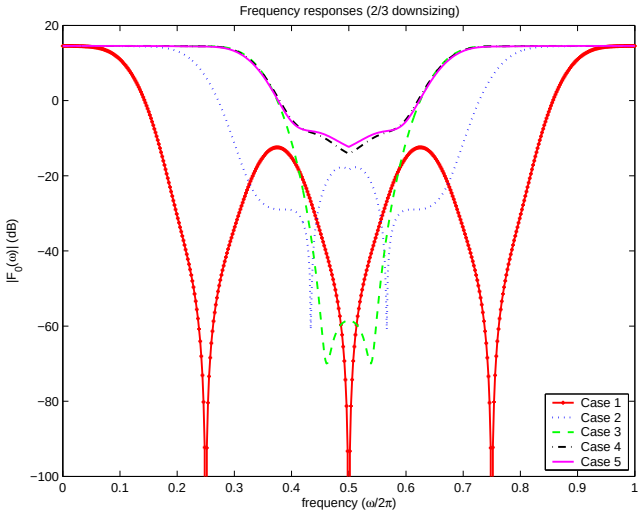


Figure 3.19: Frequency responses of $\mathbf{F}_0(\omega)$ in 2/3 downsizing

tively. Parameter sets that satisfy the constraint II present acceptable low pass performances for anti-aliasing. This coincides with their PSNR performances.

3.3.5.3 1/2 Resizing Set

At last, 1/2 resizing set is discussed. It is a special case for arbitrary resizing. Table 3.5 gives the different parameter sets and the corresponding PSNR results. In these parameter sets, only case 3-5 satisfy the constraint II. It is obvious that methods proposed in [56] and [61] are realized by TSAR with parameter sets of case 3 and case 4, respectively. Case 3-5 have similar PSNR performance and present video quality higher than that of case 1-2.

Table 3.5: PSNR comparison under 1/2 resizing set

Resizing scheme	L'	M'	PSNR (dB)				
			Lena	Baboon	Barb	Boat	Peppers
case 1	1	2	23.535	20.385	22.234	22.212	22.762
case 2	2	4	28.313	21.926	24.828	25.929	27.552
case 3	4	8	35.35	25.562	27.063	32.212	33.536
case 4	8	16	36.229	25.888	27.083	32.837	34.161
case 5	16	32	36.203	25.882	27.082	32.824	34.144

Observing the frequency responses from Figure 3.20, the ideal pass band for $\mathbf{F}_0$ is $0 \sim \pi/2$. Case 4 and case 5 have much better low pass performances. Their transit regions are narrower than that of case 3. This explains why case 4 and case 5 present a little higher PSNR than case 3. Case 3-5 have better low pass performances than case 1-2. This shows again that, a better performance of the low pass property can present a better picture quality.

From the examples shown above, the following conclusion is reached: when the constraint II is satisfied, the output picture quality is better. Furthermore,

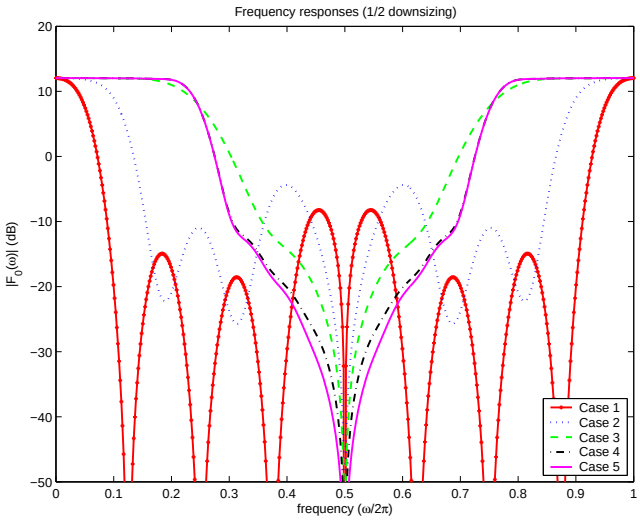


Figure 3.20: Frequency responses of $\mathbf{F}_0(\omega)$ in 1/2 downsizing

a better performance of the low pass property leads to a much better video quality.

3.4 Summary

The DCT domain decimation method discards the high frequency components and retains only the low frequency components of the block. Most of the energy in the original block is preserved. This scheme is adopted in DCT domain downsizing. Based on this technique, two DCT domain resizing schemes are introduced in this chapter.

The advantage of the first scheme lies in that it is a block-by-block processing. The partition of supporting area is the same with that used in the MV re-estimation scheme. This benefits the inter frame downsizing. For inter coded frame, the residue signal is the difference between the current frame and its reference frame. PSNR improvement of this DCT domain method for downsizing

the residue may not be the same as that for downsizing the full frame. This is because the quality of inter coded frame depends on the quality of its reference frame. In [18], a transcoding method is presented, which can re-estimate MV's from the input bitstream for an arbitrary downscaled video. In this method, the original frame is partitioned in the same way as the approach in section 3.2. Combined these two methods, an inter coded frame could be arbitrarily downsized in the DCT domain and the MV's could be re-estimated from the original MV's without re-generating it.

TSAR presents a generalized description for block-based arbitrary resizing. This solution is proposed based on a two-stage structure. All the schemes presented by existing literatures can be realized by this structure with different parameter sets. Moreover, a special constraint on the selection of parameter set is given to guarantee an acceptable picture quality. It is emphasized that, this constraint is used to guarantee the output picture quality with satisfactory anti-aliasing. When this constraint is fulfilled, the picture quality is high. This can be seen from the PSNR comparison between case 1 and case 2-5 in Table 3.2. However, among those parameter sets that satisfy this constraint, i.e. case 2 to case 5 in Table 3.2, picture quality will not increase significantly even when the computation is increased. Illustrative examples and corresponding results listed in the tables show that, TSAR with appropriate parameter sets can provide a flexible solution for arbitrary resizing in the transform domain.

Chapter 4

Frame Size Selection in Arbitrary Downsizing Transcoding

4.1 Introduction

In downsizing transcoding, the frame size of the pre-coded bitstream is reduced. By transmitting a video sequence with a reduced frame size, the overall bit rate is reduced. This is necessary when a large resolution video is to be delivered to users with limited display capabilities, or when the bandwidth cannot accommodate a full frame size bitstream.

Downsizing transcoding can reach a very low bit rate without losing the motion information. To realize frame size reduction, a downsampling filter within the transcoder is required (see Figure 4.1). Its objective is to clearly downsample the incoming video. Reducing the resolution of video ensures a

successful delivery and display of the requested video material without frame skipping.

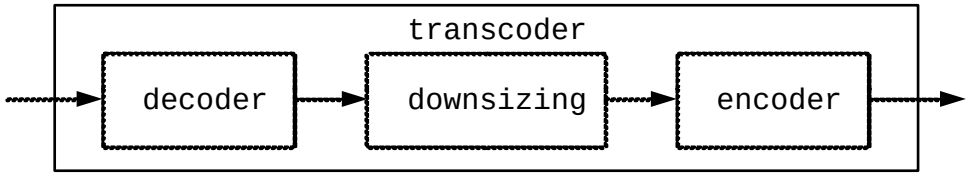


Figure 4.1: Downsizing transcoder

In downsizing transcoding, there are two main purposes: reducing the computation and improving the video quality. For reducing the computation, MV reusing or re-estimation with the related MV refinement is one of the most important tools [14, 17, 18, 21–23, 101]. Furthermore, some downsizing transcoding architectures have been proposed to reduce the computation cost on decoding and re-encoding. In [40], several transcoding architectures are evaluated and drift is analyzed. In addition, an intra refresh architecture was presented to reduce the drift in the open loop architecture. A new architecture was presented in [42] based on a submacroblock motion compensation method. Drift compensation is implemented and motion compensation is processed in the reduced frame size, which further reduces the computation cost. For improving the video quality, one of the main topics is of DCT domain downsizing [56, 57]. By DCT domain downsizing, the video quality can be improved significantly.

Arbitrary downsizing is introduced in downsizing transcoding since it can make full use of the bandwidth and reach a fine gradual reduction of bit rate and video quality. Generally, arbitrary downsizing is used to accommodate to the requirements of users. The other application is that, when the bandwidth is limited, the transcoder can be used to reduce the original frame size to a smaller

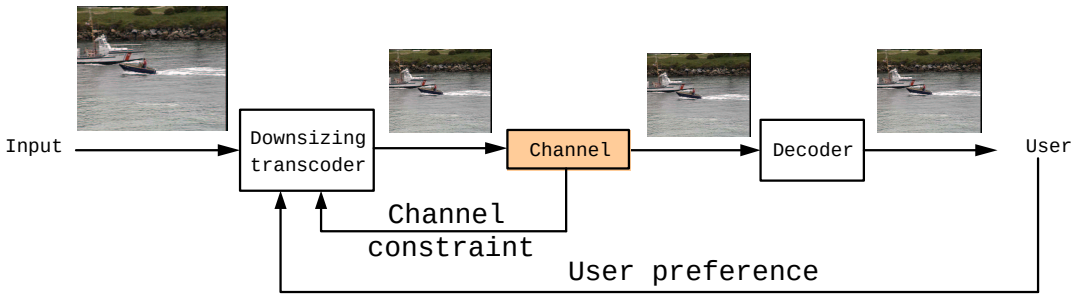
one. In this process, arbitrary downsizing can help to reduce the information loss and make full use of the bandwidth.

When arbitrary downsizing transcoding is adopted to reduce the overall bit rate, one important problem is how to select a suitable frame size for the transcoded bitstream. In [22], a simple model was presented to select the reduced frame size for downsizing transcoding. It simplifies the rate-Q model as a linear relation. Some heuristic parameters are introduced. In practical application, this model is not applicable.

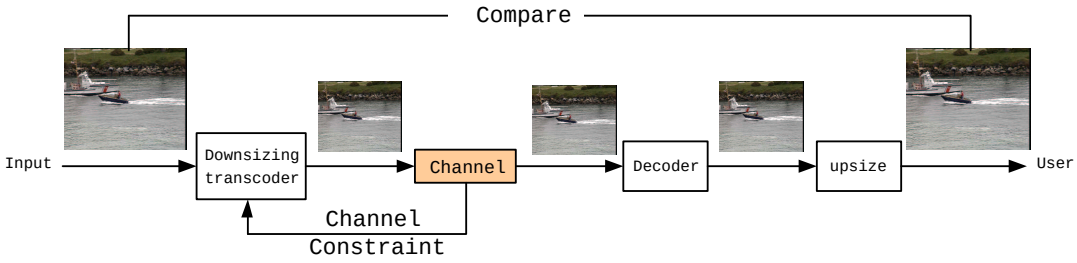
In the following, the frame size selection problem in arbitrary downsizing transcoding will be discussed in details. The rest of this chapter is organized as follows. Arbitrary downsizing transcoding process is analyzed in section 4.2. In section 4.3, some efficient methods are presented to estimate the number of bits generated in downsizing transcoding. Base on these methods, a suitable frame size is determined. At last, experimental results are given in section 4.4, and the conclusion is drawn in section 4.5.

4.2 Arbitrary Downsizing Transcoding

There are two types of application for arbitrary downsizing transcoding. First, different users have different display devices. The size of display devices may not have an integer ratio to the original video. In order to exhibit the complete video in the limited size of display devices, the original bitstream has to be downsized to a reduced frame size (see Figure 4.2(a)). In this process, the transcoder downsizes the pre-coded bitstream to the size required by the user and adopts the rate control according to the bandwidth condition.



(a)



(b)

Figure 4.2: Downsizing transcoding flowchart (a) User preference; (b) Channel constraint

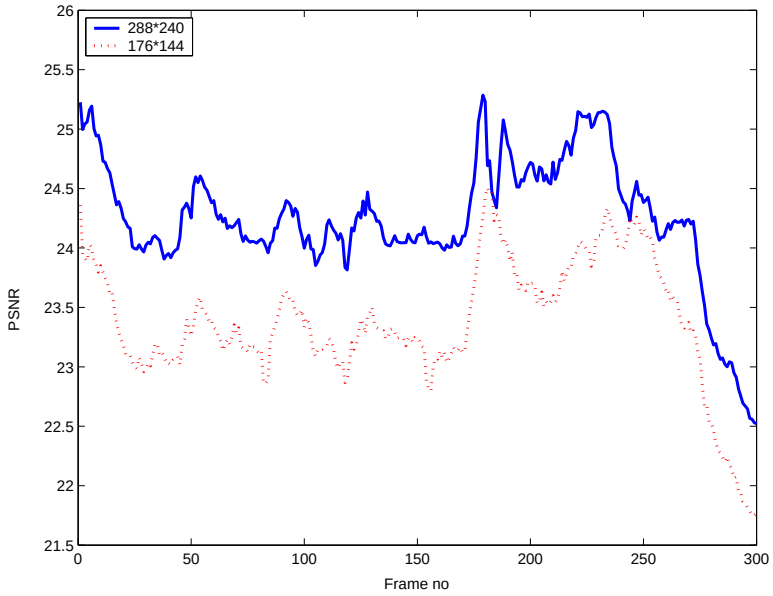
Next, arbitrary downsizing transcoding is necessary when the bandwidth is limited. Transmitting a small size video reduces the overall bit rate. However, the downsizing process may introduce distortion. A high downsizing ratio may result in high distortion. This distortion may sometimes be higher than the distortion introduced by a large quantization parameter. There is a tradeoff between the frame size and the coding parameters.

As shown in Figure 4.2(b), the bitstream in the original size is input to the transcoder. After downsizing transcoding, the frame size is reduced according to the bandwidth requirement. This downsized bitstream is transmitted via bandwidth constrained channel. Lastly, it is decoded and upsized to the original

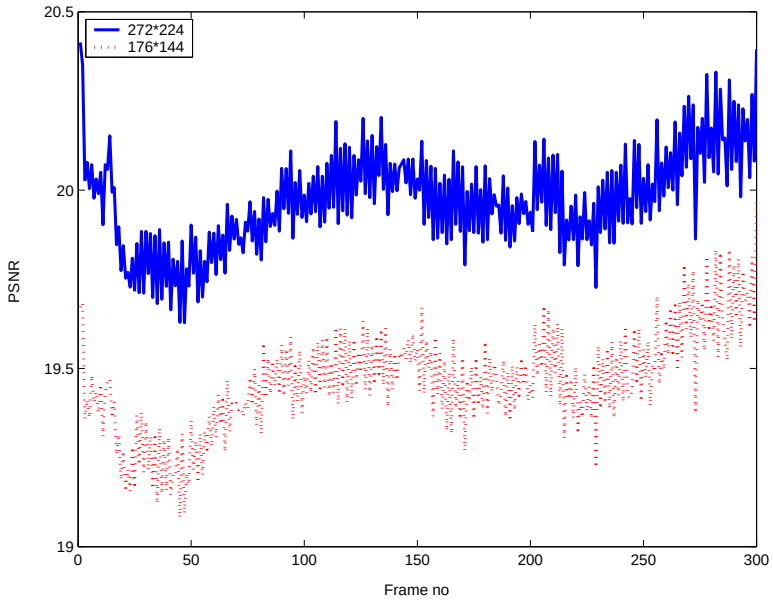
size in the receiver. In this process, there are many options of the reduced frame size. If a larger frame size is selected, a larger quantization parameter will be applied in the coding process. To make full use of the limited bandwidth and provide a high video quality as possible, a suitable frame size is expected for the transcoded bitstream when arbitrary downsizing transcoding is applied.

Traditionally, integer downsizing is adopted since it is easy to realize and no frame size selection is required. However, arbitrary downsizing can present a more flexible frame size than integer downsizing does, and more spatial information is kept when a larger frame size is adopted. Experimental results show that, with the same bit rate, arbitrary downsizing outperforms integer downsizing.

In the simulation, the TMN8 rate control scheme is adopted. Only the first the frame is intra coded, and all the following frames are inter coded. Motion estimation parameters are set to default values. In Figure 4.3(a), the video "Stefan" is pre-coded in the full frame size 352×288 at a high bit rate. With the limited bandwidth, it has to be transcoded to $240kbps$. In order to realize this transcoding without frame skipping, the original video is downsized. In this experiment, it is downsized to 288×240, which is an arbitrary downsizing process, and to 176×144, which is an integer downsizing process, respectively. For arbitrary downsizing transcoding, with a larger frame size, a coarse quantization parameter is applied, while for integer downsizing transcoding, a smaller quantization parameter is adopted. Even with a coarse quantization parameter, arbitrary downsizing transcoding presents a higher video quality. An average improvement of $0.85dB$ over integer downsizing is presented in terms of PSNR. In Figure 4.3(b), similar result is observed. The video "Mobile" is pre-coded in



(a)



(b)

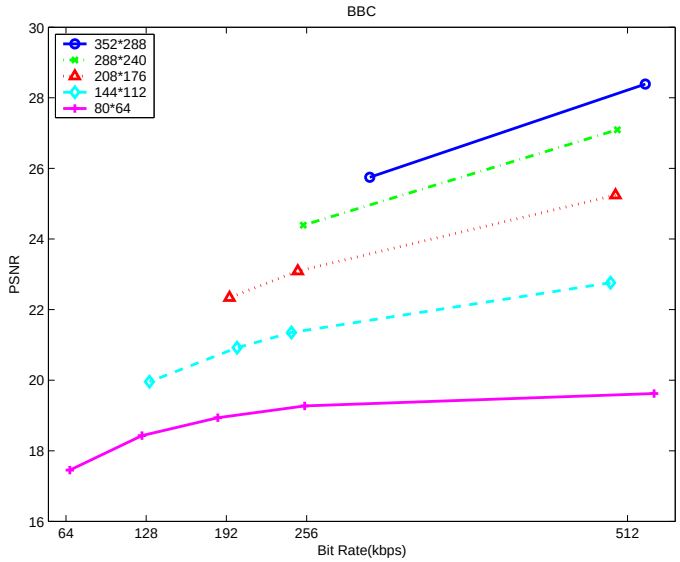
Figure 4.3: Arbitrary downsizing vs. integer downsizing: (a) Stefan; (b) Mobile

the full frame size and to be transcoded to $273kbps$. It is downsized with an arbitrary downsizing ratio to 272×224, and with an integer ratio to 176×144, respectively. Arbitrary downsizing presents an average $0.5dB$ improvement over integer downsizing. Experimental results shown above illustrate the strongpoint of arbitrary downsizing.

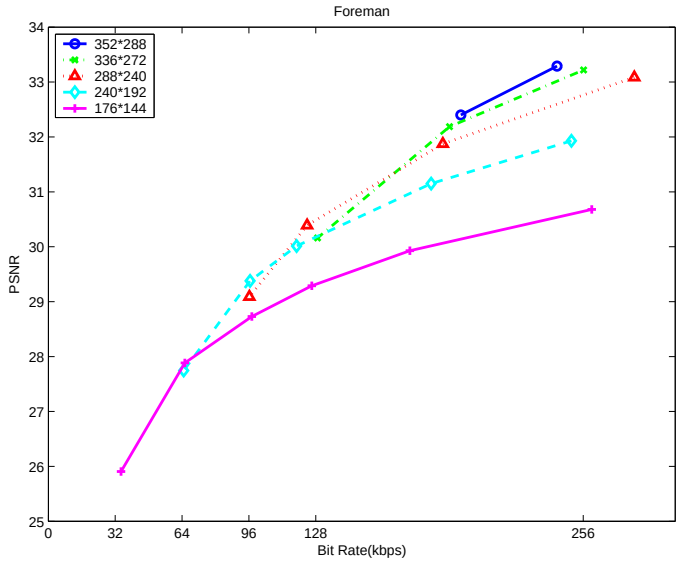
In downsizing process, information will be discarded, which introduces distortion. A high downsizing ratio results in high distortion. Sometimes it may be higher than the distortion introduced by coarse quantization. In the following, some simulation results are shown to illustrate the relation between the frame size and the output quality. In this process, the resizing (downsizing and upsizing) filter introduced in section 3.3 is adopted.

In Figure 4.4(a), the input bitstream is "BBC". It is transcoded to different frame sizes with different bit rates. With the same bit rate, the bitstream with a larger frame size presents a higher video quality. When the frame size is very small (e.g. 80×64), even in a high bit rate, it still cannot generate a high video quality. This shows that, information lost in the resizing process is higher than that lost in the quantization process.

In another example shown in Figure 4.4(b), the test video is "Foreman". The relation between the frame size and the quality is a bit different from that of "BBC". It is observed that, sometimes the bitstream in a larger frame size does not present higher PSNR than that in a smaller frame size. Comparison is given between two cases. For case 1, the frame size of 288×240 is adopted to the transcoded bitstream; for case 2, the frame size is 240×192 for the transcoded bitstream. When the bit rate is larger than 110kbps, case 1 outperforms case 2. With the reduction of the bit rate, the quantization parameter adopted for



(a)



(b)

Figure 4.4: Size vs. PSNR: (a) BBC; (b) Foreman

case 1 increases and reaches a very large value. Most of the AC coefficients in residue are quantized to zero. This introduces serious information loss. With the same bit rate, the quantization parameter adopted for case 2 is still not too large. This results in the output quality of case 1 worse than that of case 2. Similar results can be observed from other frame sizes.

Based on the experiments shown above, the following conclusion is reached: for the bitstream coded with certain frame size, when the bit rate is not too low, it may present higher video quality than when coded with a smaller frame size. Only when the bit rate is too low, coarse quantization is applied, and video quality degrades severely. A larger frame size may present lower quality than a bit smaller frame size does. Similar results have been presented in [102]. In [102], image is coded with JPEG coding standard. In most of the cases, higher image quality is presented by a larger frame size. Only in the very low bit rate case, the bitstream with a smaller frame size presents a bit higher PSNR than that with a larger frame size.

Based on this observation, a scheme is proposed by estimating the bit rate generated by different frame sizes to decide the transcoded frame size. As shown in Figure 4.5, bitstreams in different frame sizes generate the bit rates covering a range of bandwidth. There is a bit rate lower bound (B_{low}) for each frame size. If the bandwidth is lower than B_{low}, the bitstream with certain frame size cannot be transmitted via networks; if the bandwidth is higher than B_{low}, this bitstream can be transmitted successfully. In downsizing transcoding, it is desired to find the bit rate lower bounds (B_{low}'s) generated by different frame sizes. When these lower bounds are available, it is easy to choose qualified frame sizes for downsizing transcoding.

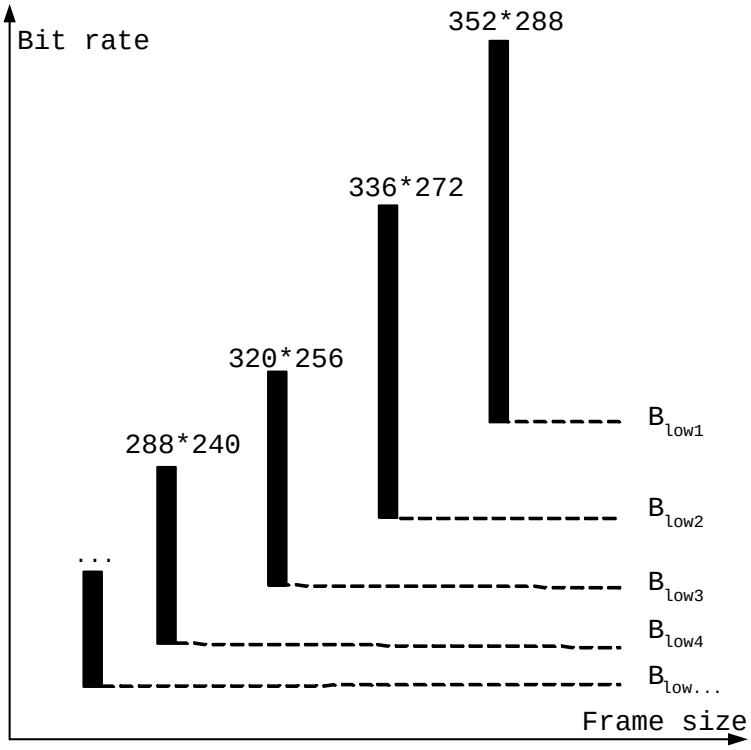


Figure 4.5: Selection of the bit rate lower bound

Among those qualified frame sizes, a larger frame size is desirable according to the observation in the simulation. First, the largest frame size within those qualified candidates is chosen. It can be the most possible frame size for downsizing transcoding. Next, B_{low} of this frame size is checked. When B_{low} is very close to the target bit rate, it means a very coarse quantization parameter is applied. In order to avoid the introduction of block artifact and to present a better visual quality, the secondary frame size in the qualified candidates is selected. Otherwise, the largest frame size among those qualified candidates may be chosen. Although this is not an optimal solution, it is a suboptimal solution for the practical application.

In this work, only the combination of downsizing and re-quantization in the

transcoding process is studied. The highest video quality is expected to meet the requirement of limited bandwidth, at the same time, without introducing any frame skipping. In the following, the detailed description is given on how to estimate B_{low}'s for different frame sizes.

4.3 Bit Rate Estimation

In the analysis of section 4.2, the frame size selection problem is converted to a bit rate estimation problem. It is required to get B_{low}'s of different frame sizes in the selection of frame size. The estimation process is realized by two steps. First, the bit rate in the full frame size after re-quantization is estimated. In this step, bits used for the MV, residue, and header are estimated, respectively. By the re-quantization analysis, the lowest bit rate generated in the full frame size is obtained. Second, the bit rate after downsizing is estimated. Bit rate lower bounds for different reduced sizes are obtained in this step. Detail is discussed as follows.

4.3.1 Re-Quantization Bit Rate in Full Frame Size

In order to get B_{low}'s under different frame sizes after coarse re-quantization, B_{low} of the original frame size is first estimated. Because the bitstream is pre-coded with a fine quantization parameter, statistical information from the incoming bitstream can be reused to estimate the bit rate with a coarse quantization parameter, i.e. B_{low}.

The total number of bits allocated to a frame can be divided into three parts: bits used to code MV, residue, and header as shown in (4.1). In the

re-quantization process, some of them may change with the increasing of quantization parameter, and some may keep constant. In the following, the detailed analysis on how to obtain an accurate estimation of B_{low} in the original frame size is given.

$$R = R_{mv} + R_{residue} + R_{header}. \tag{4.1}$$

4.3.1.1 Bits for MV

As discussed in [21, 103], MV's are computed with the accuracy (Δ) and differentially encoded in raster scan order with lossless entropy coding. The total number of bits allocated to MV's is approximated as

$$R_{mv} \approx \frac{1}{B^2} \cdot \log_2(\frac{4e^2B}{\Delta^2}) + \frac{1}{B^2} \cdot \log_2(\sigma_v^2 \ln(\frac{1}{\tilde{a}})). \tag{4.2}$$

where B is the size of motion compensated block for frame residue, σ_v^2 and $\tilde{a}$ are the average variance and correlation coefficient of MV's for a frame, and e is the natural log. In the coding process, with the MB size B fixed to 16 and $\tilde{a}$ set to 0.998, the second part of (4.2) can be approximated to (4.3).

$$\frac{1}{B^2} \cdot \log_2(\sigma_v^2 \ln(\frac{1}{\tilde{a}})) \approx \frac{1}{256} log_2(0.002 \cdot \sigma_v^2). \tag{4.3}$$

It is noted that, the value of the second part is very small. The first part in (4.2) will dominate and contribute to most of the bit rate. It is also observed that, the first part will not change with the variation of quantization. In another word, the number of bits used to code MV is almost a constant and has little relation to the quantization parameter. In the re-quantization process, with the

same frame size, the number of bits for coding MV's is treated as a constant regardless of the changing of quantization parameter.

4.3.1.2 Bits for Residue

Residue is generated by motion compensation, DCT transform, quantization, and VLC coding. Among them, the quantization process is a lossy process introduced to reduce the bits allocated for residue, and a large quantization parameter will generate fewer bits. Many efforts have been invested in estimating the bits for residue. However, it is very hard to get an accurate estimation before actual coding. In some R-Q models, rate is assumed to have certain relation with the quantization parameter. In general cases, these models cannot accurately build the relation between the rate and quantization parameter for it depends on the texture information of the video. Moreover, the statistical information of residue may vary because of the inter-frame dependency.

In video transcoding, statistical information collected from the pre-coded bitstream can be reused. It helps to reach an accurate rate estimation in the re-quantization process. In following, an efficient scheme is presented to estimate the number of bits generated in full frame size with a coarse quantization parameter.

At first, two transcoding cases are designed (see Figure 4.6). In these transcoding cases, the same pre-coded bitstream is used as the input. This pre-coded bitstream is a fine quantized bitstream with the quantization parameter Q_1.

- *Case 1.*

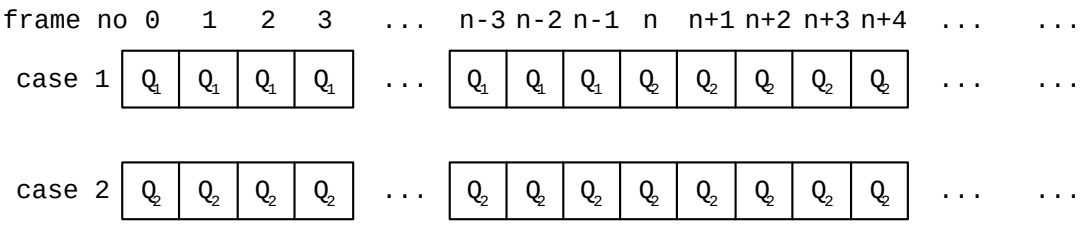


Figure 4.6: Two different transcoding cases

From frame 0 to frame $n-1$, the pre-coded bitstream is transcoded with Q_1. After frame $n-1$, a coarse quantization parameter Q_2 is applied, where Q_2 is much larger than Q_1. Because different quantization parameters are applied, the statistical information is different before and after frame n. However, information collected from the pre-coded bitstream before frame n can be reused since it is a fine quantization process.

- *Case 2.*

  The coarse quantization parameter Q_2 is applied to the pre-coded bitstream throughout the transcoding process. Because Q_2 is larger than Q_1, the reconstructed frame is different from the pre-coded bitstream. The residue collected from the pre-coded bitstream is not suitable to be reused.

In these two cases, case 2 is the expected re-quantization transcoding process. The quantization parameter Q_2 is applied to the entire bitstream, and the overall bit rate is reduced. In this work, Q_2 is the largest quantization parameter available under the coding standard. It is expected to get B_{low} in this coarse quantization process. In order to estimate the bits generated by case 2, the number of bits after frame $n-1$ for case 1 is estimated first, and the relation

of bits between case 1 and case 2 is built.

In case 1, because the coding parameter from frame 0 to frame $n-1$ is the same as that in the pre-coded bitstream, the number of bits generated for each frame is close to the pre-coded bitstream. The residue collected from the pre-coded bitstream can also be reused for transcoding. Here, the residue of frame n ($f_{residue1}(n)$) before quantization can be modelled as follows:

$$f_{residue1}(n) = f(n) - \hat{f}_{Q_1}(n-1). \tag{4.4}$$

where $f(n)$ is the value of frame n and $\hat{f}_{Q_1}(n-1)$ is the reconstructed frame $n-1$ under quantization parameter Q_1. This residue can be assumed to be the same as the residue obtained from the pre-coded bitstream.

In order to get the number of bits to code $f_{residue1}(n)$ when Q_2 is applied, the ρ domain model introduced in [104, 105] is adopted. In [104, 105], the relation between the percentage of zeros among the quantized coefficients (ρ) and the bit rate generated (R_ρ) is built as follows:

$$R_\rho = \theta \cdot (1-\rho). \tag{4.5}$$

where θ is the relation between them, which can be considered as a constant. This shows that, if the number of non-zero coefficients in $f_{residue1}(n)$ under the quantization parameter Q_2 is available, it is easy to obtain the bits for frame n.

Because $f_{residue1}(n)$ is reused from the pre-coded bitstream, DCT coefficients of the residue are easily generated. Considering the new quantization parameter Q_2, the number of non-zero coefficients can be counted. In addition, θ calculated from the pre-coded bitstream can also be reused. In such a way, the number of

bits generated for frame n in case 1 can be calculated.

Next, the relation of bits generated between case 1 and case 2 can be built. In [106], a rate model is introduced as shown in (4.6) to build the relation between the quantization parameter (Q) and the empirical entropy of the quantized coefficient. It is given by

$$H(Q) = \begin{cases} \frac{1}{2} \cdot \log_2(2e^2 \frac{\sigma^2}{Q^2}) & \frac{\sigma^2}{Q^2} > \frac{1}{2e} \\ \frac{e}{\ln 2} \frac{\sigma^2}{Q^2} & \frac{\sigma^2}{Q^2} \leq \frac{1}{2e}. \end{cases} \tag{4.6}$$

In this equation, $H(Q)$ is used as the approximation of bit rate $R(Q)$, and it is a function of the variance of coefficient (σ) and quantization parameter (Q). In low bit rate coding, (4.7) is used as an estimation for the bit rate. This is because when a coarse quantization parameter, e.g., Q = 31 in H.263, is applied, the condition $\frac{\sigma^2}{Q^2} \leq \frac{1}{2e}$ is always met.

$$H(Q) = \frac{e}{\ln 2} \frac{\sigma^2}{Q^2}. \tag{4.7}$$

$$f_{residue2}(n) = f(n) - \hat{f}_{Q_2}(n-1). \tag{4.8}$$

Now, the transcoding cases discussed above can be considered. As these two transcoding cases apply the same quantization parameter Q_2 to frame n, the difference between the bit rates generated by these two cases lies in the difference of residue. In case 1, the residue of frame n is shown in (4.4), and in case 2, the residue of frame n is shown in (4.8). Since the reconstructed frames $\hat{f}_{Q_1}(n-1)$ and $\hat{f}_{Q_2}(n-1)$ are different, residue data and residue variance (σ) are also different. Because Q_2 is applied to frame n in both cases, the number

of bits generated has the following relation:

$$\frac{R_{residue1}}{R_{residue2}} = \frac{e}{\ln 2}\frac{\sigma_1^2}{Q_2^2} / \frac{e}{\ln 2}\frac{\sigma_2^2}{Q_2^2} = \frac{\sigma_1^2}{\sigma_2^2}. \tag{4.9}$$

According to (4.9), the ratio of bit rates between the two cases can be represented as the ratio of coefficient variance. In the coding process, the frame residue is the motion compensated difference between the current frame $f(n)$ and the reconstructed previous frame $\hat{f}_Q(n-1)$. The coefficient variance can be described as

$$\begin{aligned} \sigma^2 &= var[f(n) - \hat{f}_Q(n-1)] \\ &= var[f(n) - f(n-1) + f(n-1) - \hat{f}_Q(n-1)]. \end{aligned} \tag{4.10}$$

where $f(n-1)$ is the previous frame. The first part $(f(n) - f(n-1))$ can be treated as the interpolation error between frames, and the second part $(f(n-1) - \hat{f}_Q(n-1))$ is the quantization error. Assuming that these two parts are independent, (4.10) can be further partitioned into two parts:

$$\sigma^2 \approx var[f(n) - f(n-1)] + var[f(n-1) - \hat{f}_Q(n-1)]. \tag{4.11}$$

The first part is the variance of frame difference. This part is easy to get from the pre-coded bitstream. Under a fine quantization parameter, it is close to the variance of residue after quantization.

The second part is the variance of quantization error. In [105], transform coefficients are assumed to have a generalized Gaussian distribution with standard

deviation σ given by

$$p_{gg}(x) = \left[\frac{\nu\eta(\nu,\sigma)}{2\Gamma(\frac{1}{\nu})}\right] \cdot e^{-\left[\eta(\nu,\sigma)\right]^{\nu}} \tag{4.12}$$

where

$$\eta(\nu,\sigma) = \sigma^{-1}\left[\frac{\Gamma(\frac{3}{\nu})}{\Gamma(\frac{1}{\nu})}\right]^{1/2}, \quad 1 \leq \nu \leq 2. \tag{4.13}$$

When the model parameter ν takes value 2, $p_{gg}(x)$ becomes a Gaussian distribution given by

$$p_g(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}}. \tag{4.14}$$

For a uniform threshold quantizer with a step size q and a dead zone Δ, the second part can be described as,

$$\begin{aligned} var[f(n-1) &- f(n-1,Q)] \\ &= 2\int_0^{\Delta} x^2 p_g(x)dx + 2\sum_{i=0}^{\infty}\int_{iq+\Delta}^{(i+1)q+\Delta}\left[x-(i+\frac{1}{2})q-\Delta\right]^2 p_g(x)dx. \end{aligned} \tag{4.15}$$

In H.263 inter MB coding, Δ and q have the relation with the quantization parameter Q shown as follows:

$$\Delta = 2.5Q. \tag{4.16}$$

$$q = 2Q. \tag{4.17}$$

In (4.14) and (4.15), the only unknown parameter is the standard deviation of transform coefficients σ. As discussed above, under a fine quantization parameter, it is the same as the variance of residue in the pre-coded bitstream.

With the analysis shown above, the proposed approach is outlined as follows. First, the number of bits for frame n in case 1 is estimated by (4.5). Next, the variance of residue can be calculated from (4.11) and (4.15). According to the relation between these two cases stated in (4.9), the number of bits used to code the residue of frame n in case 2 is obtained. Based on this result, the number of bits to code other frames in case 2 is also available.

4.3.1.3 Bits for Header

Header information in each frame is composed of picture layer, Group of Blocks (GOB) layer, MB layer, and block layer. Most of bits are allocated at MB layer. Some parts of bits are almost constant under different quantization parameter; while some will decrease with increasing of the quantization parameter. As defined in H.263 [6], MCBPY is a variable length codeword giving information about the MB type and the coded block pattern for chrominance, and CBPB is a variable length codeword that gives a pattern number signifying those luminance blocks in the MB. With a large quantization parameter applied, bits allocated to these two codewords may reduce. Simulation result is shown in Figure 4.7.

It is observed that, the number of bits allocated to these codewords is reduced with the increasing of quantization parameter. However, this reduction is not considered in other literatures.

The reduction of bits allocated to these codewords varies for different video sequences. The more zero MB's are, the fewer bits allocated to them. Based on

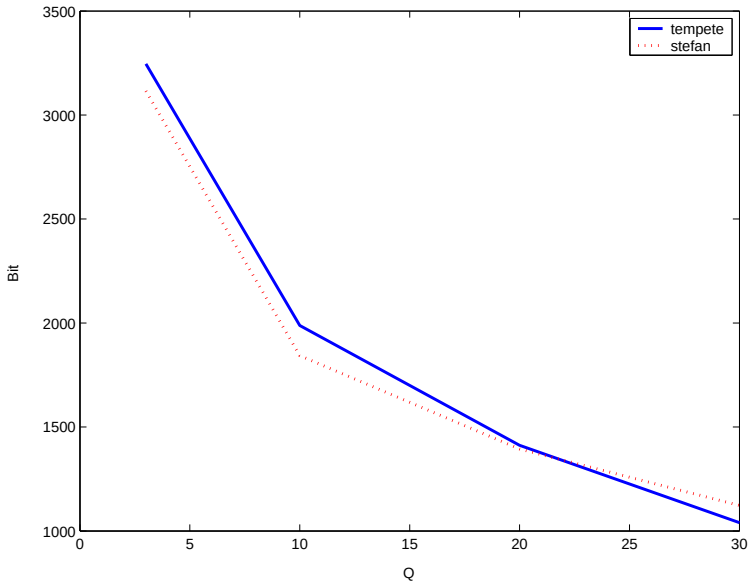


Figure 4.7: Bits for MCBPC and CBPY

the observation, the following model is adopted to estimate the number of bits for MCBPC and CBPY ($B'_{header}(Q_2)$),

$$B'_{header}(Q_2) = \mathcal{F}(B'_{header}(Q_1),\ \sigma^2). \tag{4.18}$$

where $B'_{header}(Q_1)$ is the number of bits for MCBPC and CBPY in the pre-coded bitstream, and σ^2 is the variance of residue. As shown in Figure 4.7, with similar variance and $B'_{header}(Q_1)$, two video sequences produce similar bits for MCBPC and CBPY after re-quantization. A table can be built to describe this relation based on extensive simulation. With look-up-table (LUT), the bits for these two codewords can be roughly estimated for different video sequences.

The other parts of header are seldom affected by the quantization process. They are assumed to be identical to the pre-coded bitstream.

Based on the analysis for these three parts of bits, the relation of bits between the pre-coded bitstream and the transcoded bitstream in full frame size is built. The bit rate lower bound for the original frame size (B_{low}) is available now.

4.3.2 B_{low}'s of Different Frame Sizes

In this part, the relation of bit rates between different frame sizes will be discussed. Generally speaking, the overall bit rate has a linear relation with the frame size. As discussed above, the number of bits allocated to MV and header is only related to the frame size. Assuming that the variance of residue will also not change after downsizing, there is a linear relation between the downsized bit rate and the original bit rate, which is modelled by

$$R_{down} = \frac{S_{down}}{S_{org}} \cdot R_{org} \tag{4.19}$$

where S_{org} and S_{down} are the size of the original bitstream and the transcoded bitstream. This linear relation can be observed from some simulation results (see Figure 4.8).

Based on the analysis above, the re-quantization process and downsizing process can be combined. B_{low}'s of different frame sizes are available as shown in (4.20). With these lower bounds, it is easy to select a suitable frame size for the transcoded bitstream in bandwidth-limited condition based on the scheme presented in section 4.2.

$$B_{low} = \frac{S_{down}}{S_{org}} \cdot (R_{mv} + R_{residue} + R_{header}). \tag{4.20}$$

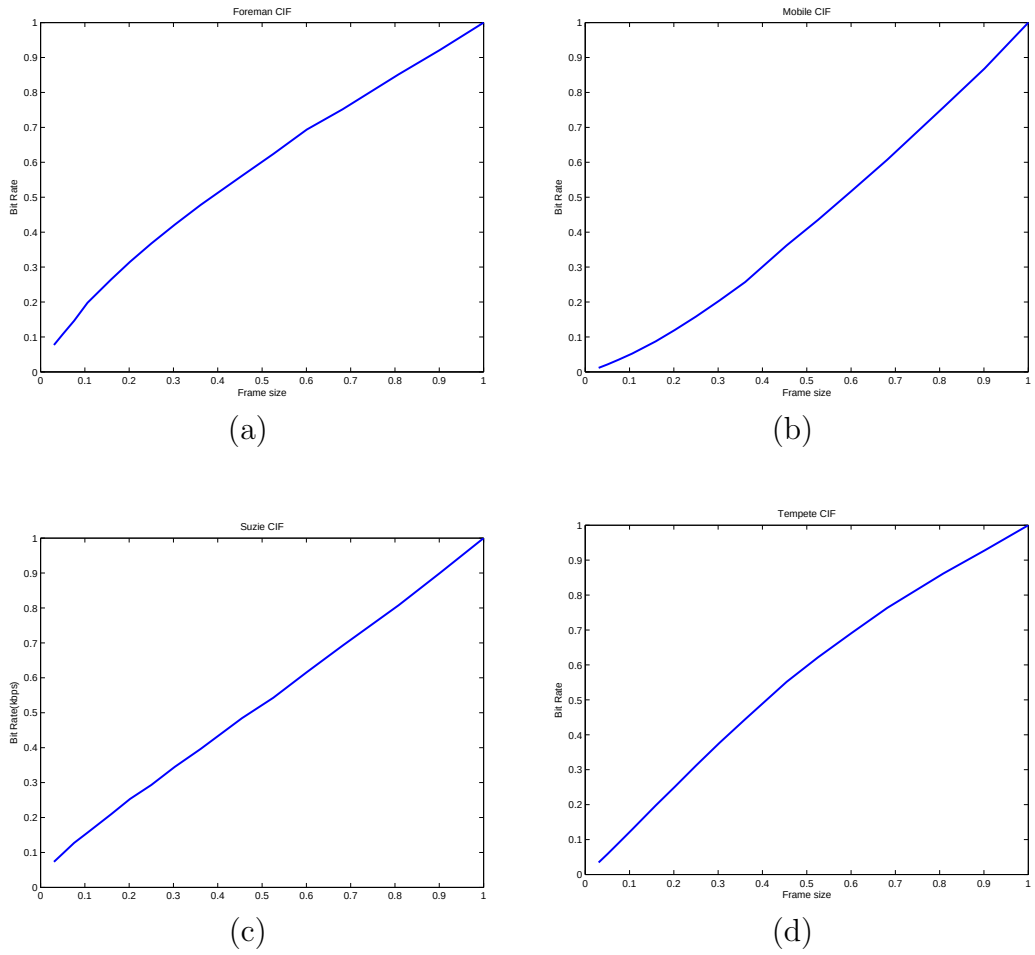


Figure 4.8: Frame size vs. Bit rate. (a) Foreman; (b) Mobile; (c) Suzie; (d) Tempete

4.4 Experimental Results

In the following, experimental results are given to verify the feasibility and accuracy of the proposed method. In Table 4.1, the original frame size is 352×288. The coding parameters are the same as described in section 4.2. These sequences are pre-coded with quantization parameter $Q_1 = 3$. After re-quantization transcoding, the number of bits generated under quantization parameter $Q_2 = 31$ is estimated. It is observed that, the proposed method

presents a satisfactory estimation.

Table 4.1: Estimation of bits allocated to residue

Video sequence	bits for residue	estimated bits	error
Stefan	3858	3454	10%
Mobile	12806	11588	9%
BBC	4463	4924	10%
Tempete	2678	2240	16%
Hall monitor	413	365	11%
Foreman	527	441	16%

For the frame size selection in downsizing transcoding, some experimental results are shown in Table 4.2. The original frame size is 352×288. These sequences are pre-coded with a low quantization parameter. In each test sequence, the statistical information of each frame is approximately the same throughout the sequence. The frame size chosen by the proposed method is presented as "estimated frame size", and the optimal frame size is obtained by exhaustive experiments. The resizing filter introduced in section 3.3 is adopted in the experiments.

Table 4.2: Frame size selection

video sequence	number of frames	target bit rate	estimated frame size	bit rate lower bound	optimal frame size
Bike	300	128kbps	208*176	107k	208*176
Stefan	150	128kbps	240*192	96k	256*208
Foreman	150	128kbps	336*272	108k	336*272
Tempete	260	128kbps	288*240	111k	288*240
Hall monitor	280	128kbps	352*288	101k	352*288

It is observed that, in most of the cases, the frame size chosen by the proposed method is close to the optimal frame size. For the video "Foreman", B_{low} of

frame size 352×288 is $121k$. It is smaller than the target bit rate ($128k$). However, this lower bound is too close to the target bit rate. Secondary frame size 336×272 is chosen for downsizing transcoding. This coincides to the exhaustive experimental result. A better visual quality is presented.

For some sequences, the proposed frame size may be a bit deviated from the optimal one. This is because residue is prone to be affected by different downsizing filters. With different downsizing filters, the reduction of the bit rate is not in a rigorous linear relation with the reduction of frame size.

4.5 Summary

In this chapter, frame size selection for arbitrary downsizing transcoding is discussed. Differing from some traditional schemes (e.g. [107]), which try to maximize the output video quality by exhaustive calculation, the proposed scheme converts a distortion estimation problem to a bit rate estimation problem, which does not need exhaustive simulation.

In order to reduce the artifact introduced by coarse quantization, the proposed scheme chooses a smaller frame size than the fit one. Although the most fit frame size can generate a satisfactory bit rate and higher PSNR, the visual perception may not be as good as a smaller frame size one.

The proposed scheme can be used for real time applications with only a few frames delay. The delay is caused by information collection and scene change detection. Comparing with exhaustive trying scheme, this scheme need not encode and calculate the distortion. This saves the computation significantly. For video sequences with less scene change, this scheme is only employed once

before another scene change happens.

At last, the summarization of the proposed scheme is outlined in the following steps:

1. *Estimate the variance for case 1 and case 2 from (4.11) and (4.15);*
2. *Estimate the bit rate of residue in case 1 by (4.5);*
3. *According to the relation stated in (4.9), estimate the bit rate of residue in case 2 with a coarse quantization parameter;*
4. *Calculate the lower bound for full frame size by (4.1);*
5. *Calculate B_{low}'s for different reduced frame sizes by (4.20);*
6. *Select the frame size for the transcoded bitstream based on the scheme introduced in section 4.2.*

Chapter 5

Frame Skipping Transcoding with Consideration of Motion Information

5.1 Introduction

In the previous two chapters, topics related to downsizing transcoding are discussed. In this chapter, frame skipping transcoding is studied. In video transcoding, a high transcoding ratio may result in an unacceptable picture quality when full frame rate of the incoming bitstream is used. Re-quantization cannot work well in low bit rate transmission because a large quantization factor may introduce block artifact. Furthermore, reducing spatial resolution is operationally complicated. Considering the vision persistence of the human eyes, frame skipping is often adopted as an efficient tool to reduce the overall bit rate. Frame skipping allows allocating more bits to the preserved frames, particularly

those frames containing more important information. In such a way, an acceptable quality for the coded frames can be maintained and limitations imposed by the channel constraints can be satisfied.

Traditionally, frames are skipped according to the buffer constraint [6, 106]. When the buffer occupancy is larger than or equal to a threshold, frames are skipped until the buffer occupancy is below this threshold. However, this simple frame skipping strategy does not consider the motion information and the video quality. The output video may not present a satisfactory effect.

Some other researchers suggested to build the rate-distortion model to determine the frame skipping pattern [50, 51, 108–110]. In [51], a simple relation is built between the spatial and temporal video quality, and temporal rate is determined by making a tradeoff between them. A variable frame rate scheme was introduced in [108], which considers the buffer condition, the desired buffer level, and the frequency coefficients to determine the frames to be skipped. In [109], a rate-distortion model was presented based on a distortion estimation method for the skipped frames. It can reach a tradeoff between the spatial and temporal video quality. However, these model based schemes do not take the motion information into consideration. Some high motion frames may be skipped and the visual quality may be compromised. And in some situations, these models are inaccurate to present the rate-distortion relation. Although the method introduced in [110] can present a good result with joint temporal-spatial bit allocation, the computation cost is rather high for it requires an exhaustive search.

In some papers, the motion activity is considered in the frame skipping schemes [53,111–113]. These schemes consider the motion activity of each frame

and dynamically skip those frames with low motion activity. Several methods were proposed to estimate the motion activity. In [111], the motion activity is estimated by calculating the histogram of difference image (HOD) to detect motion activities in video. The frame skipping style of each group of picture (GOP) is determined by a dynamic frame rate control method. And in [112], motion compensated error (MCE) is selected as the distortion measure to indicate the motion activity. In some other papers, MV's of the previous coded frames are used to predict the motion activity of the current frame [53, 113]. In transcoding, MV's of the current frame are available before the current frame is coded. MV's are adopted as the indication of the motion activity, and frames are skipped when small motion occurs [38, 49].

These schemes that consider the motion activity in the coding process are widely adopted because they aim at reducing motion loss. Therefore, higher temporal quality can be preserved. Especially in high motion case, the generated bitstream can well meet the requirement of human visual perception. However, among these methods, jerky effect in frame skipping is not considered. Jerky effect may introduce a discontinuous motion that affects the visual perception and degrades the overall video quality. Since the evaluation of jerky effect is very complicated, it is seldom considered in existing literatures. In this chapter, the concept of motion change is introduced to evaluate the motion continuity. Combined with the motion activity, these two parts of information can be used to determine the importance of each frame. Based on the motion information, a dynamic frame skipping scheme was proposed to well preserve the motion information in the transcoded bitstreams and let the user enjoy a better video quality without increasing the overall bit rate.

This chapter is organized as follows. Problems existing in the current frame skipping methods are stated in section 5.2. In section 5.3, the analysis of the motion information is given. A dynamic frame skipping algorithm is introduced in section 5.4. Experimental results are presented in section 5.5, and the conclusion is drawn in section 5.6.

5.2 Literature Review

Although the buffer constrained frame skipping scheme is too simple to provide a high video quality, the concept of buffer constraint is adopted in other frame skipping schemes for it can avoid buffer overflow or underflow. In buffer constrained scheme, frames are skipped when the buffer occupancy exceeds a predefined threshold [106, 114]. After the current frame is coded, the number of bits in the buffer is updated as $W = max(W + B' - R/F, 0)$, and frame skipping can be simply described as

$$\begin{aligned} &While\ W > M \\ &\{ \\ &\qquad W = max(W - R/F, 0) \\ &\qquad skip++ \\ &\} \end{aligned} \tag{5.1}$$

where B', M, R, and F are the number of bits for the current coded frame, threshold for frame skipping, target bit rate, and target frame rate, respectively.

The buffer constrained scheme is convenient to be realized. However, it does not consider the motion information of video. The resulted video may lead to

jerky effect. Especially in high motion case, because a large number of bits are used to code the residue, this scheme may skip more frames, or apply a larger quantization factor to the next frame to keep the buffer free from overflow. This leads to more motion information loss in high motion case.

Dynamic frame skipping strategy that considers the motion information is then developed to determine the frames to be skipped. The motion information within bitstream can affect the human visual perception as well as the spatial video quality. It is advisable to consider the motion information in frame skipping transcoding instead of the model based method that considers the rate-distortion relation only.

In video transcoding, MV's of the current frame are available before coding it. The commonly used scheme is considering the MV's of the current frame as the indication of the motion activity [49]. In this approach, the motion activity is defined as the summation of MV values in a picture, which is given by

$$mv_{acc} = \sum(|mv_x| + |mv_y|). \tag{5.2}$$

where mv_x and mv_y are the MV's in x and y direction, respectively.

When the accumulated motion activity after a coded frame exceeds a certain threshold, the current frame should be coded. As seen from [49], there are some drawbacks in this frame skipping scheme that considers the motion activity only. First, it considers only the motion of the current frame. Jerky effect caused by sudden motion change is not taken into account. Second, in low motion case, too many frames are skipped. For example, in the experiment of [49], from frame 209 to frame 223, more than 12 frames are skipped. Although there may be low motion activity in these frames, skipping too many frames may lead to

visual quality degradation. In addition, this will cause buffer underflow.

In [38], a frame skipping transcoding architecture has been developed. With this architecture, a frame skipping scheme was proposed by considering both the motion activity and the transcoding error. This method skips those frames with large transcoding error and small motion activity based on the assumption that more distortion will be introduced by coding those frames with large transcoding error. By discarding those frames, more bits can be saved to code the following frames. However, this scheme neglects the fact that when MV's are accumulated, the transcoding error is hence accumulated. This error will propagate to the following frames and cause severe quality degradation. Furthermore, this scheme is not applicable to other frame skipping transcoding architectures.

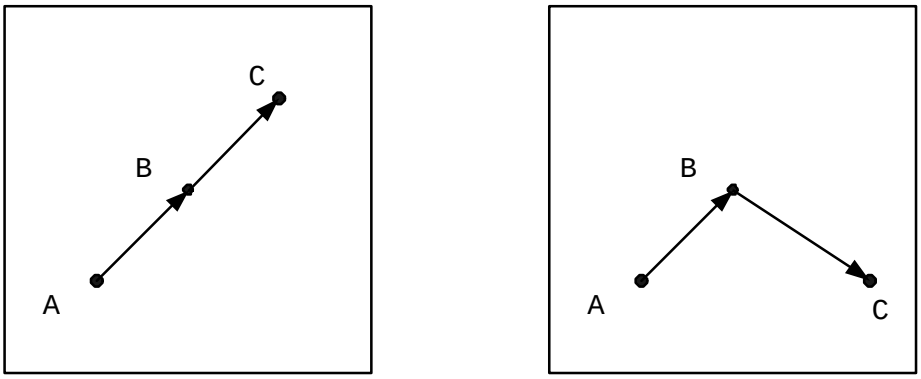
From the analysis shown above, it is observed that frame skipping scheme with the consideration of motion activity is an acceptable solution to determine the temporal rate. But there are still some drawbacks in this scheme. The most important one is that it does not consider the motion continuity. Jerky effect caused by sudden motion change may introduce a discontinuous motion. This affects the visual perception and degrades the overall video quality. In the following section, the analysis of the motion continuity will be presented and one method is proposed to evaluate the motion continuity. A new frame skipping scheme is proposed based on the consideration of both the motion activity and motion continuity information.

5.3 Motion Information Analysis

Motion information is the information related to motion that occurs at a specified time interval. Traditionally, the motion activity is adopted to indicate the motion is fast or slow. However, this is not enough. A simple observation is the swing of the pendulum. It is obvious that the frame in which the pendulum reaches the highest position is very important, as it provides the information of height that the pendulum can reach. If this frame is lost, it is very hard to get this information. If we only consider the motion activity information, the importance of this frame cannot be highlighted.

This example can be illustrated with Figure 5.1. One object moves from A to C via B. From A to B and from B to C, there are the same time intervals. As shown in Figure 5.1(a), an object moves from A to C. The direction and magnitude of the motion are not changed at B. It is a smooth motion. If the motion information at B is unavailable, it is very easy to reconstruct its MV. However in Figure 5.1(b), one motion change is observed at B. Both the direction and magnitude of the motion are changed. This is a discontinuous motion. If the motion information at B is lost, it is hard to reconstruct it. Although the motion activity at B is the same for both cases, there is different motion continuity information at B.

This example shows that, the motion activity information alone cannot represent all the motion information. Some frames in the video are very important for they contain high motion change information. When these frames are skipped, it is very hard to reconstruct the original motion information. In frame skipping transcoding, it is expected to code these frames and preserve more motion information.



(a) (b)

Figure 5.1: Motion continuity: (a) Smooth motion; (b) Abrupt motion

From the analysis above, it is observed that the motion information contained at one frame should consist of two components: motion activity and motion continuity. Motion activity indicates the motion is fast or slow, and motion continuity means the movement is smooth or rough. In frame skipping transcoding, when a frame is skipped, the motion activity information is lost. Also, the motion continuity information is changed. When a frame is coded, the motion activity information is preserved. However, the motion continuity information may be changed due to the previous or next frame skipping. This leads to the motion information loss.

5.3.1 Motion Activity Information

In order to measure the motion activity, several methods are proposed. Apart from the motion activity defined in (5.2), another definition of the motion activity is given in (5.3). Here, it is defined as the square of MV. In this work,

the definition in (5.3) is adopted.

$$mv_{acc} = \sum(mv_x^2 + mv_y^2). \tag{5.3}$$

5.3.2 Motion Continuity Information

The motion continuity information is seldom discussed in the existing literatures. Here, motion change occurring at one frame is applied to evaluate the motion continuity. Motion change is calculated by the difference of MV's between two consecutive frames. Assuming MV's for a pixel in the current frame and the next frame are mv_{cur} and mv_{next} respectively, motion change ∇mv is given by

$$\nabla mv = mv_{next} - mv_{cur}. \tag{5.4}$$

When most of the pixels within one frame present high motion change, it indicates a scene change occurrence. In the following, an experiment is conducted to evaluate the performance of ∇mv via scene change detection.

Several researches have been done in scene change detection. For uncompressed video sequence, the statistical information of frames and frame difference are considered as scene change features [115–119]. Instead of working on the uncompressed video sequence, the approaches for scene change detection in the compressed domain have been developed [48, 116, 119–126]. In [119–123], DC sequences are generated from the compressed image, and statistical information is extracted from these DC sequences. The number of intra MB's has also been proposed as a scene change feature [48, 124]. It shows that more intra MB's

are introduced in those frames directly after the scene change. For MPEG-2 bitstream, the number of interpolated MB's (predicted from both previous and next I-/P-frames) within B-frame is adopted as a scene change feature [125]. Another scene change feature commonly used is the variance of the residue ($var\_residue$). In coding process, high variance indicates a mismatch between the current frame and the previous frame. When scene change occurs, it is very hard to reach good matching in motion estimation process. In [116], the summation of MV values in a picture, MVF, is chosen as a scene change feature. It can represent the motion activity of each frame. In [126], motion smoothness is considered. This value is infinity at each scene change.

In the following experiments, the variance of the residue ($var\_residue$), the summation of MV (MVF), and the number of intra MB's are used for comparison. Here, MVF is one type of motion activity information.

In Figure 5.2, the test video is "Coastguard". One scene change occurs at about frame 69 with a camera suddenly upward and another occurs at about frame 117 with a high motion object disappearance. MVF cannot detect the scene change occurring at about frame 117. For the number of intra MB's, less than 2% MB's are intra coded in the first scene change and another scene change is also missed. For the feature $var\_residue$, one scene change is missed. However, ∇mv can successfully detect all the scene changes in the sequence.

Another test video is "Cact". It has no abrupt scene change. A camera zooms out following by a horizontal shift gradually. As shown in Figure 5.3, MVF shows high fluctuation throughout the video; the number of intra MB's presents little change; $var\_residue$ cannot provide any useful information, while ∇mv presents a gradual increasing before frame 90 and a gradual decreasing

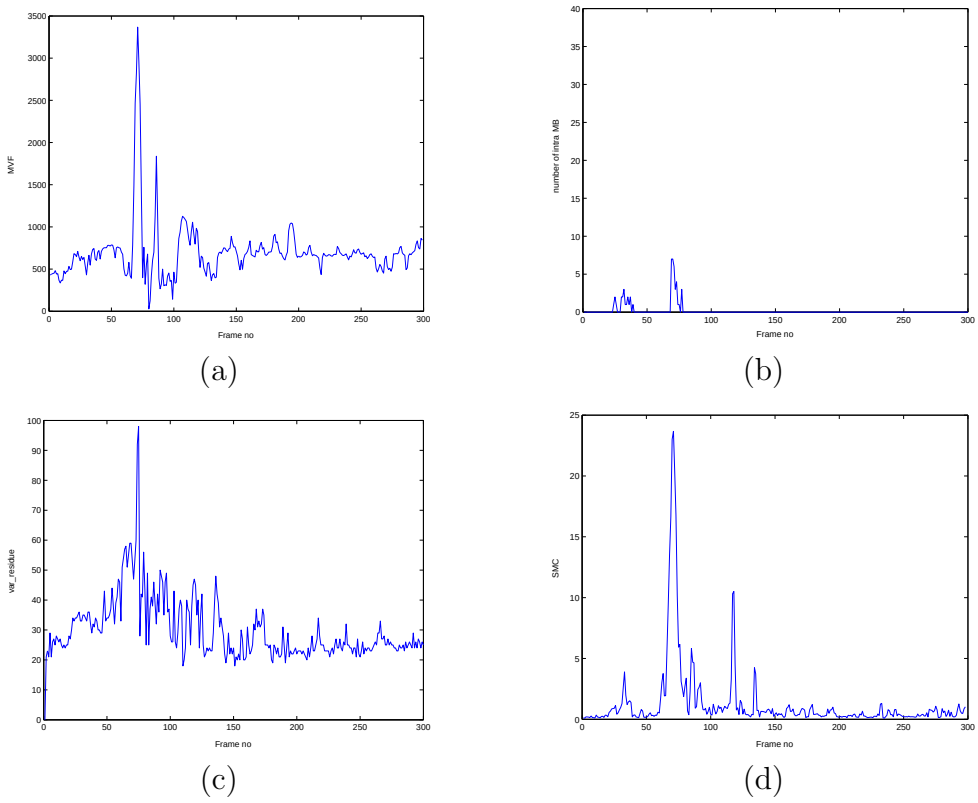


Figure 5.2: "Coastguard": (a) MVF; (b) Number of intra MB's; (c) Var_residue; (d) Motion change information

after frame 90. This provides the tendency of the motion continuity information.

Based on the experimental results shown above, it is obvious that, motion change ∇mv can successfully detect the scene change. It is more efficient than the other scene change features and quite robust in different cases. In addition, it can provide useful information for making decision in practical applications. Therefore, ∇mv provides different information from the motion activity and it is an efficient tool to indicate the motion continuity.

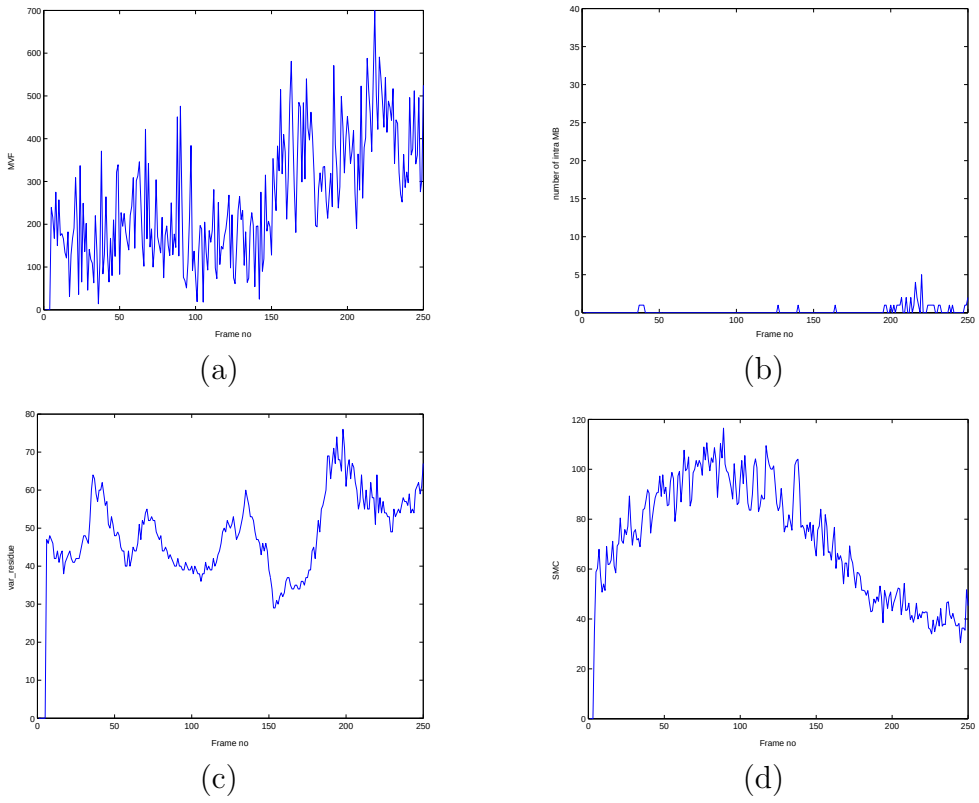


Figure 5.3: "Cact": (a) MVF; (b) Number of intra MB's; (c) Var_residue; (d) Motion change information

5.4 Proposed Frame Skipping Scheme

Based on the above study, a motion information based frame skipping transcoding scheme (MIFST) is proposed in this section. MIFST aims at reducing the overall bit rate and mitigating the difference of motion information between the original bitstream and the transcoded bitstream.

5.4.1 Motion Criterion

First, motion criterion of MIFST is introduced. This criterion is used to reduce the motion loss of the transcoded bitstream. When the motion of the

transcoded bitstream is close to the original bitstream, key motion information of the original video is retained. This benefits the temporal video quality.

It is easy to understand that when one frame is skipped, the motion information of this frame is lost. When this frame is coded, the motion information of this frame may also be lost according to the different skipping modes of the transcoded bitstream. Example is illustrated in Figure 5.4 to show the different motion continuity information at a coded frame.

In Figure 5.4(a), all the frames are coded, the motion change occurring at frame $k+1$ is almost the same as that in the original video. In Figure 5.4(b), frame k is skipped and the other frames are coded. Frame k is directly copied from the previous coded frame and the MV of frame $k+1$ becomes mv_1'. The motion change occurring at frame $k+1$ becomes $mv_2 - mv_1'$. In Figure 5.4(c), only frame $k+2$ is skipped, then the motion change occurring at frame $k+1$ becomes $-mv_1$. In Figure 5.4(d), frame k and frame $k+2$ are both skipped. The motion change occurring at frame $k+1$ is $-mv_1'$. Because frame $k+1$ is coded in these four cases, the motion activity of frame $k+1$ is the same. With different frame skipping modes, there is different motion continuity information at this frame.

This confirms again that, MV's of a single frame cannot be used to represent all the motion information at the current frame interval. The motion activity information and motion continuity information must be evaluated separately according to the skipping mode of the transcoded bitstream. In frame skipping transcoding, both parts of motion information should be taken into account.

When a frame is coded, the motion activity of this frame is preserved; if this frame is skipped, the motion activity information is lost. The loss of the motion

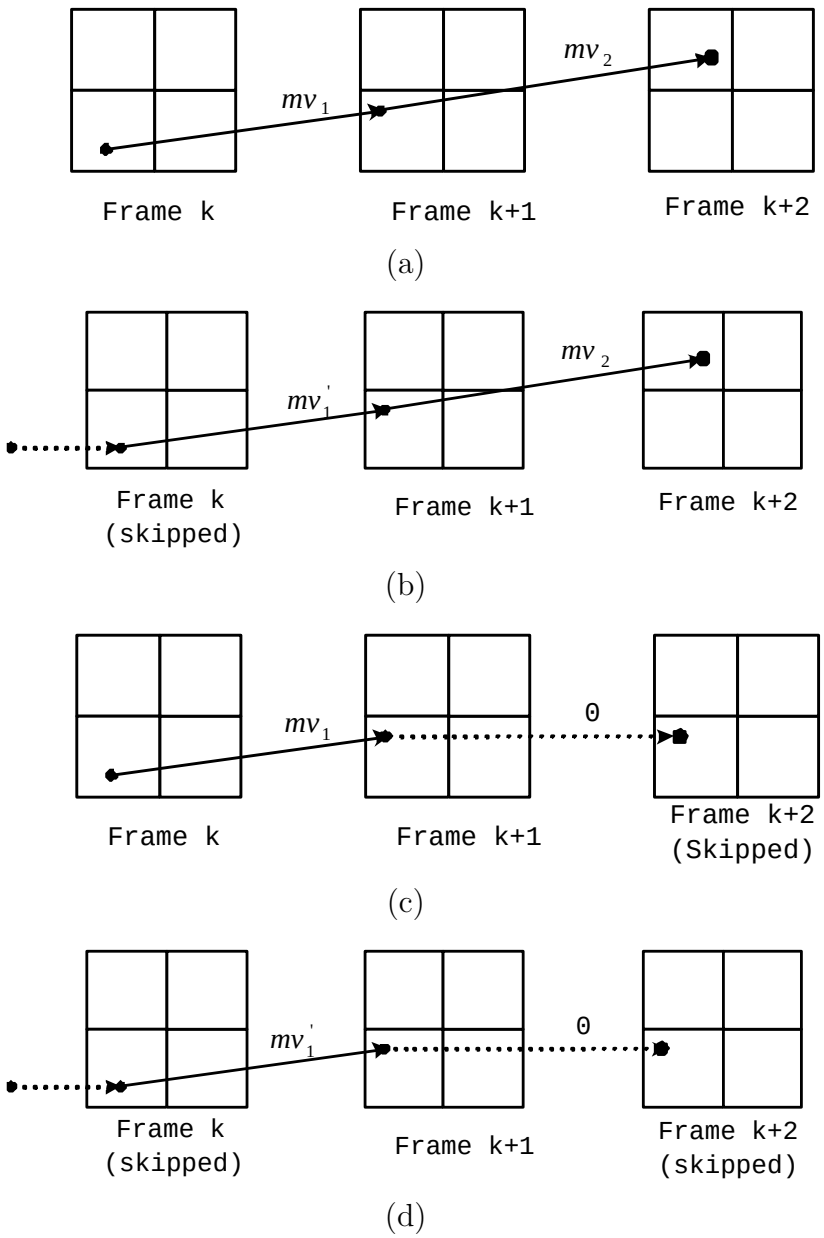


Figure 5.4: Motion change occurring at coded frame k+1. (a) All the frames are coded; (b) Previous frame is skipped; (c) Next frame is skipped; (d) Previous and next frames are skipped

activity information D_MV_{acc} can be described as

$$D\_MV_{acc} = \begin{cases} \sum(mv_x^2 + mv_y^2) & frame\ is\ skipped \\ 0 & frame\ is\ coded \end{cases} \tag{5.5}$$

For the motion continuity information, motion change defined in (5.4) is adopted to evaluate it. In frame skipping transcoding, when one frame is skipped, the reconstructed MV may differ from the original MV. This leads to the difference of motion continuity information between the original video and the reconstructed video. In order to measure this, square difference of motion change (SDMC) is defined to evaluate it, which is given by

$$SDMC = \sum[(\nabla mv_{org,x} - \nabla mv_{trans,x})^2 + (\nabla mv_{org,y} - \nabla mv_{trans,y})^2] \tag{5.6}$$

where $\nabla mv_{org,x}$, $\nabla mv_{trans,x}$, $\nabla mv_{org,y}$, and $\nabla mv_{trans,y}$ are the motion changes of the original video and the transcoded video in x and y directions, respectively. When the motion change of the output video does not comply with the original video, this value may increase, which means the motion continuity is impaired.

Then the difference of motion information between the original video and the transcoded video is defined as

$$MI\_diff = D\_MV_{acc} + w \cdot SDMC. \tag{5.7}$$

where w is a weighting factor that indicates the importance of the motion continuity. In the experiments, its value is set to 1. A small MI_diff means that the motion of the transcoded video is close to the original video. The transcoded

video is desired to present MI_diff as low as possible.

In MIFST, the motion criterion is selected to determine the skipping mode of the transcoded bitstream. All the skipped frames between two coded frames are considered to be in the same GOP with the following coded frame. The average difference of motion information within one GOP ($m(MI\_diff)$) is calculated as follows:

$$m(MI\_diff) = \frac{1}{L+1} \cdot \sum_{i=0}^{L} MI\_diff_i. \tag{5.8}$$

where L is the number of skipped frames and MI_diff_i is the difference of the motion information for each frame within this GOP.

5.4.2 Buffer Constraint

The buffer constraint is also considered in MIFST. Buffer overflow may result in data loss or a larger buffer size; buffer underflow may waste the bandwidth. When only the motion criterion is considered, there may be such cases where the frame rate keeps on increasing or decreasing. Buffer overflow or underflow will be induced.

In MIFST, motion criterion is applied only when the buffer status is under a satisfied condition. If the buffer overflow occurs under the current skipping mode, the number of skipped frames will be forced to increase without considering the motion information loss; if the buffer underflow occurs, the number of skipped frames will be forced to decrease.

5.4.3 Skipping Length Constraint

As defined in 5.4.1, all the skipped frames between two coded frames are considered to be in the same GOP with the following coded frame. The length of GOP changes when the number of skipped frames L changes. In MIFST, L is forced to be less than 6. This is another constraint in MIFST.

Skipping a large number of frames continuously may decrease the visual quality although it can reduce the overall bit rate. The constraint of L is to give a better visual perception and to avoid buffer underflow.

In order to mitigate the jerky effect and to present a smooth motion change when the frame rate is changed, the changing of L is restricted to one level between neighboring GOPs. This means that L can only increase or decrease by one between two consecutive GOPs.

5.4.4 Proposed Scheme

Based on the motion criterion, the buffer criterion, and the skipping length constraint, details of MIFST are illustrated in the following.

After the number of the skipped frames (L) is determined, the corresponding frame is coded. The current $m(MI\_diff)$ is updated. In order to determine L for the following frames, a variable length prediction window is designed to pre-calculate the average difference of motion information for the following frames. The length of the prediction window is determined by the current L as shown in Table 5.1. This prediction window contains two GOPs and the total length is $2(L+1)$. A long prediction window length is used to mitigate the variation of frame rate. The predicted $m(MI\_diff)$ is calculated within this prediction

Table 5.1: Relation between the number of skipped frames, GOP length and prediction window length

Number of skipped frames	GOP length	Prediction window length	Coded frame No.
0	1	2	1, 2
1	2	4	2, 4
2	3	6	3, 6
3	4	8	4, 8
4	5	10	5, 10
5	6	12	6, 12

window.

Let $\triangle$ denote the difference between the predicted $m(MI\_diff)$ and the current $m(MI\_diff)$, the number of skipped frames is adjusted as follows:

$$\begin{cases} \triangle \geq thd & L = L - 1 \\ \triangle \leq -thd & L = L + 1 \end{cases} \tag{5.9}$$

where thd is the threshold.

The process is illustrated in Figure 5.5. At time t, after frame k is coded, MIFST needs to decide how many frames to be skipped for the next interval. According to the current L, the predicted $m(MI\_diff)$ is calculated based on the prediction window length, which is $2(L+1)$. When the predicted $m(MI\_diff)$ is less than the current $m(MI\_diff)$ for a certain threshold, the number of skipped frames is increased from L to $L+1$, and frame $k+L+2$ is coded at time t'. Now, the prediction window length is $2(L+2)$. Because the predicted $m(MI\_diff)$ exceeds the current $m(MI\_diff)$ for a certain threshold, the number of skipped frames is decreased from $L+1$ to L and frame $k+2L+3$ is

coded at time t''.

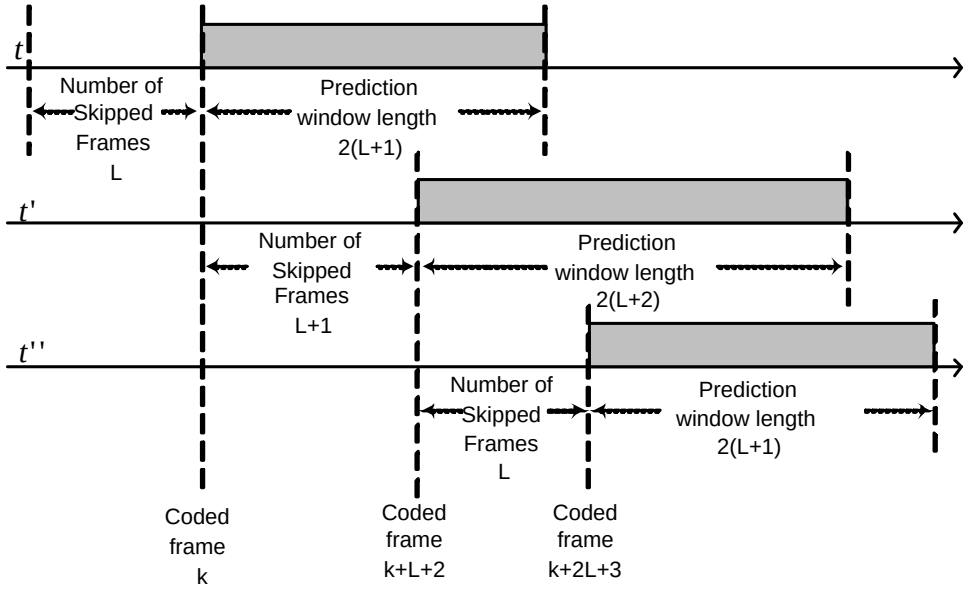


Figure 5.5: Illustration of the proposed frame skipping scheme (MIFST)

The algorithm is outlined as follows:

1. Check the buffer occupancy: if buffer overflow or underflow occurs under the current L, L is increased or decreased by 1, go to the step 5; otherwise, continue.

2. Set the prediction window length as $2(L+1)$. Calculate the predicted $m(MI\_diff)$ within this prediction window.

3. Get the difference between the current $m(MI\_diff)$ and the predicted $m(MI\_diff)$. If the current $m(MI\_diff)$ is larger than the predicted $m(MI\_diff)$ plus a certain threshold, L is increased by 1; if the predicted $m(MI\_diff)$ is larger than the current $m(MI\_diff)$ plus a certain threshold, L is decreased by 1; otherwise, L is unchanged.

4. Check the restriction of L within [0, 5].

5. Skip L frames and code the following frame. Update the current $m(MI\_diff)$.

5.5 Experimental Results

Experiments are conducted to verify the performance of MIFST. This frame skipping method is implemented based on a commonly used cascaded pixel domain transcoder (CPDT) (see Figure 5.6). In this architecture, an extra frame buffer is used to store the input bitstream when calculating the predicted $m(MI\_diff)$. The TMN8-based frame skipping scheme is used for comparison [6]. And, the bit allocation and MB layer rate control are the same in MIFST and the TMN8-based scheme.

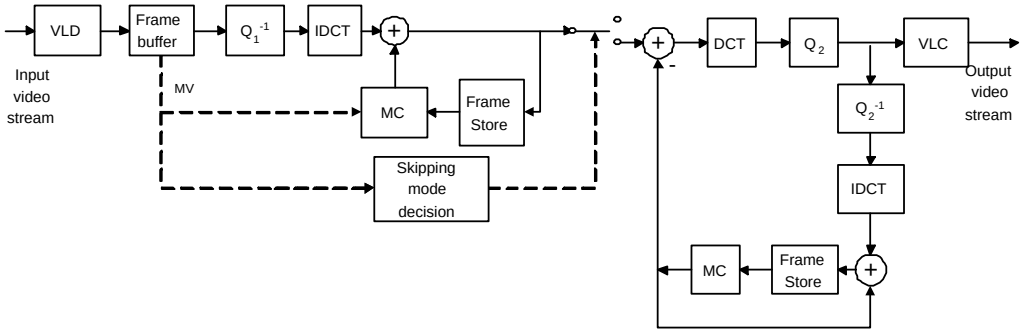


Figure 5.6: Specified cascaded pixel domain transcoder

The video "Coastguard" is taken as an example. This video contains mainly the median motion and a motion outburst around frame 69. The incoming bitstream is in QCIF format and the frame rate is 30 $frames/s$. GOP structure IPPPP... is adopted and only the first frame is intra coded. After transcoding, the target frame rate is 10 $frames/s$ and the bit rate is reduced to 64$kbps$.

Table 5.2: Comparison of coded frames by MIFST and TMN8

	Coded frames
MIFST	60, 63, 66, 68, 69, 71, 74, 77, 80
TMN8	60, 63, 66, 69, 72, 75, 78

Hence the number of coded frames is less than the original video as a result of frame skipping.

In high motion area, it is desired to keep more frames to improve the temporal quality and benefit the human visual perception; in low motion area, visual quality may not decrease significantly even if more frames are skipped. MIFST is designed in such a way, that more frames are coded in high motion area and fewer frames are coded in low motion area. Comparing with the TMN8-based scheme, which treats high motion and low motion cases equally and ignores the motion information of each frame, MIFST codes more frames in high motion area without increasing the overall bit rate. Table 5.2 shows the comparison of the coded frames by two methods from frame 60 to frame 80. In these frames, there is a sudden camera upward movement; therefore motion around frame 69 is high. The TMN8-based method skips almost the same number of frames throughout the video. However, by MIFST, more frames are coded around frame 69. Owing to the prediction window that can well predict the variation of motion information, the number of skipped frames is reduced from 2 gradually and reaches 0 around frame 68. Therefore, there is no sudden change in the frame rate. This result shows that, MIFST preserves more motion information and presents a motion closer to the original video.

Next, the comparison of motion loss between MIFST and the TMN8-based scheme is given. The evaluation of motion loss is given by (5.7). In the

Table 5.3: Comparison on difference of motion information

Sequence	Target bit rate	Difference of motion information		
		TMN8	MIFST	Improvement (%)
Foreman	64kbps	10410	9290	10.8
Coastguard	64kbps	1124	1009	10.2
Stefan	64kbps	26527	24005	9.5
Suzie	32kbps	1887	1530	18.9
News	32kbps	324	320	1.2
Table_tennis	32kbps	4082	3617	11.4

transcoded video of "Coastguard", the overall MI_diff by MIFST is 1009, while by the TMN8-based method it is 1124. MIFST presents 115 units of MI_diff lower than the TMN8-based method does. More results of other video sequences are shown in Table 5.3. It is obvious that, MIFST presents much smaller MI_diff comparing with the method adopted by TMN8. The improvement of MIFST over the TMN8-based method can be calculated as

$$|\frac{MI\_diff_{proposed} - MI\_diff_{TMN8}}{MI\_diff_{TMN8}}| \times 100\%. \tag{5.10}$$

Observed from Table 5.3, MIFST can reduce MI_diff significantly for those sequences with much motion information. For those sequences containing little motion information, the improvement of MI_diff by MIFST is limited. This is because for sequences with little motion information, MIFST performs similar to the TMN8-based approach. Almost the same frame skipping mode is adopted and the motion information loss is similar. In this comparison, the video "Suzie" is a low motion video sequence. However, there is a sudden scene change in it. This is why high improvement by MIFST is presented.

Table 5.4: Comparison of PSNR for difference video sequences

Sequence	Target bit rate	Average PSNR (dB)			
		TMN8	D_MV_{acc}	SDMC	MIFST
Coastguard	64kbps	27.21	27.21	27.25	27.30
Foreman	64kbps	27.59	27.84	27.82	27.87
Stefan	64kbps	20.57	20.69	20.81	20.82
Suzie	32kbps	31.97	32.00	32.01	32.13
News	32kbps	29.79	29.82	29.80	29.86
Table_tennis	32kbps	26.25	26.30	26.30	26.44

By using MIFST, bits saved in low motion area can be used to code more frames in high motion area. Therefore, the overall visual quality can be improved. The easiest and most commonly used way in displaying the skipped frames is to directly copy the previous coded frame. In low motion area, the difference between the consecutive frames is low. Even with more frames skipped, the difference between the reconstructed frame and the original frame is small. Little distortion is introduced. In high motion area, the difference between frames is significant. For those skipped frames, the reconstructed version would be very different from the original one. This introduces high distortion. With MIFST, more frames are coded in this area, thus distortion can be reduced significantly.

With MIFST, spatial visual quality is also improved. Table 5.4 shows the PSNR comparison when the skipped frames are reconstructed by directly copying previous coded frame. MIFST is compared with TMN8-based scheme. The overall performance of MIFST is better than the others. In this table, video quality improvement in terms of PSNR is not significant for the video "News". This is because "News" is a low motion sequence. The motion information loss and the spatial quality are very close by these two methods. For other

sequences, the PSNR improvement by MIFST is about $0.1dB$ to $0.28dB$ higher than the TMN8-based scheme.

Sometimes, the motion compensated interpolation (MCI) is adopted to reconstruct the skipped frames in the transcoded video. MCI can present higher video quality over the duplicating scheme. It is well known that the decoder is unaware of the original motion. When the decoder reconstructs a skipped frame using MCI, it adopts the assumption that the motion between two consecutive coded frames is a linear motion. When the real motion between two consecutive coded frames is a nonlinear motion, the reconstructed frame may differ from the original frame.

When a nonlinear motion appears in the video sequence between consecutive frames, high motion change will be presented. MIFST may code more frames in this area to reduce the motion difference between the transcoded video and the original video. Consequently, little nonlinear motion within the skipped frames is presented by MIFST when compared with the other scheme. This benefits the reconstruction of skipped frames based on the motion compensation interpolation.

In this study, skipped frames in the transcoded video are reconstructed by MCI proposed in [127] and compared with the other scheme. The comparison of the output is given in Table 5.5. The PSNR improvement by MIFST is higher than that by the TMN8-based scheme (see Table 5.6). This proves that MIFST reduces the nonlinear motion and improves the visual quality.

To further illustrate the importance of the motion activity and the motion continuity information, two more schemes are applied that consider the loss of the motion activity and the motion continuity information separately. In

Table 5.5: Comparison of PSNR for difference video sequences With MCI

Sequence	Target bit rate	Average PSNR (dB)			
		TMN8	D_MV_{acc}	SDMC	MIFST
Coastguard	64kbps	29.51	29.52	29.54	29.62
Foreman	64kbps	29.17	29.50	29.52	29.53
Stefan	64kbps	22.00	22.09	22.13	22.32
Suzie	32kbps	32.72	32.81	32.80	32.91
News	32kbps	30.33	30.48	30.38	30.57
Table_tennis	32kbps	27.67	27.71	27.71	27.93

Table 5.6: Comparison of PSNR improvement (in dB)

Sequence	TMN8			MIFST		
	Duplicating (dB)	MCI (dB)	Improve-ment	Duplicating (dB)	MCI (dB)	Improve-ment
Coastguard	27.21	29.51	2.30	27.30	29.62	2.32
Foreman	27.59	29.17	1.58	27.87	29.53	1.66
Stefan	20.57	22.00	1.43	20.82	22.32	1.50
Suzie	31.97	32.72	0.75	32.13	32.91	0.78
News	29.79	30.33	0.54	29.86	30.57	0.71
Table_tennis	26.25	27.67	1.42	26.44	27.93	1.49

the first scheme, only the loss of motion activity (D_MV_{acc}) is considered; in the second scheme, only the loss of motion continuity (SDMC) is taken into account. From the PSNR comparison in Table 5.4 and Table 5.5, it is observed that, these two schemes present higher PSNR when compared with the TMN8-based scheme in any circumstance. However, they present lower PSNR than MIFST. This is because only part of the motion information is considered in these methods.

Another observation is that for some video sequences, the performance of the first scheme is better that of the second scheme, and vice versa for some

other video sequences. It is hard to say which one presents higher visual quality than the other. D_MV_{acc} and SDMC play different roles in the frame skipping schemes to reduce the motion loss. MIFST takes into account both the motion activity and the motion continuity information. The quality improvement of MIFST is the co-operation of these two parts of information.

With the introduction of MCI, the frame by frame PSNR comparison is given in Figure 5.7. In Figure 5.7(a), the PSNR improvement is not clearly observed. Figure 5.7(b) gives the comparison from frame 60 to frame 80, which is a high motion area. It is obvious that MIFST outperforms the TMN8-based scheme significantly.

5.6 Summary

Frame skipping transcoding has been one of the key techniques for video delivery over networks. In this chapter, frame skipping scheme with the consideration of motion information is discussed. The concept of motion change is introduced to evaluate the motion continuity information. Based on the analysis of motion information, MIFST is introduced to reduce the motion difference between the transcoded and the original video.

This scheme has four advantages over the other schemes:

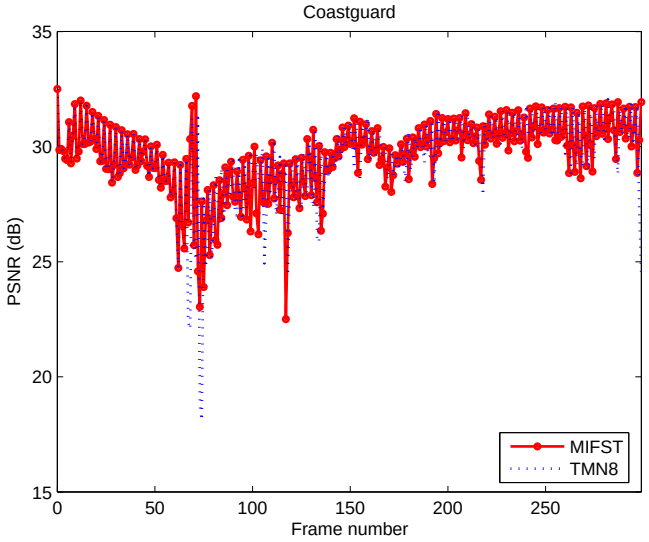
1. Keeping the original motion information. This reduces the jerky effect within the transcoded video;

2. Improving the spatial video quality. This is because the proposed scheme codes more frames in high motion area and less frames in low motion

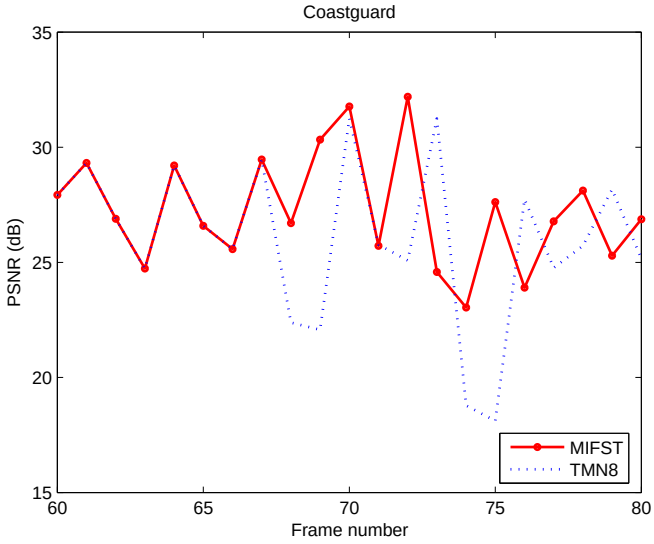
area. Bits saved in coding residue play an important role in enhancing the quality of those preserved frames;

3. Benefiting the reconstruction of skipped frames. Especially for MCI, since the nonlinear motion is reduced, the reconstructed video quality has been improved significantly.

4. Reducing the frame rate variation. This is because the number of the skipped frames is restricted within one level between two consecutive GOP. This further improves the visual quality for human perception.



(a)



(b)

Figure 5.7: Frame by frame PSNR comparison with MCI for "Coastguard": (a) frame 0-299; (b) frame 60-80.

Chapter 6

Enhanced Error Resilient Transcoding

6.1 Introduction

In an error-free network, video content is compressed to reduce the spatial and temporal redundancy without considering the channel error. In order to reduce the bit rate and reach the highest compression ratio, techniques are developed to reduce the redundancy in the video. In the generated bitstream, dependency between frames is rather high. Even correlation between MB's is also high. This kind of bitstream is highly sensitive to the channel error. Unless a dedicated link that can provide a guaranteed quality of service (QoS) is available between the source and the destination, data packets may be lost or corrupted, due to either traffic congestion or bit error that happens in the physical channel.

When bit error or packet loss happens, it not only affects the frame where error happens, but also propagates to the following frames. This results in

serious drift. Figure 6.1 shows the effect of error propagation. One slice is lost in frame 40 of the video "Foreman" (see Figure 6.1(a)). This slice is reconstructed by directly copying the previous frame in the same spatial location. Since all the frames are inter coded, next frame may use the reconstructed previous frame as the reference frame. Although no error happens in the following frames, error propagates from frame 40 to the following frames via motion compensation. It can be observed that drift is serious in Figure 6.1(b)-(d). Video quality is degraded. Therefore, it is important to devise video encoding/decoding schemes that can make the bitstream resilient to transmission errors. It is also prudent to design proper interfacing mechanisms between the codec (encoder and decoder) and the networks, so that the codec can adjust its operations based on the network conditions.

(a) (b) (c) (d)

Figure 6.1: Effect of error propagation

To deliver video content over an error-prone network, video has to be coded in a resilient format to endure the channel error and to facilitate the error concealment scheme in the decoder. Error resilient transcoder is adopted at the interface of the error-free network and the error-prone network. The purpose of error resilient transcoding is to insert error resilient component to the input bitstream. The transcoded bitstream can be more resilient to the errors. It should be noted that, error resilient transcoding is less efficient because it uses

more bits to obtain the same video quality in the absence of any transmission error. The extra bits are used to code the redundancy. The goal of error resilient transcoding is to achieve a maximum gain in error resilience with the smallest amount of redundancy.

Several error resilient tools can be considered in video coding: inserting resynchronization marker between different slices, reversible variable length coding, intra refreshment, multiple description coding, reference picture selection, and so on [86]. Some of the techniques help to prevent error propagation, while others enable the decoder to perform better error concealment upon the detection of errors.

Inserting resynchronization marker and reversible variable length coding (RVLC) are two main schemes used for robust entropy coding [86]. Resynchronization marker can easily be distinguished from other codewords in the bitstream. Inserting resynchronization marker between different slices periodically can help to identify the position where error happens. When error happens, the decoder can resume proper decoding upon the detection of a resynchronization marker. This scheme is used for stopping the spatial error propagation. RVLC is another tool that enables the decoder to decode bitstream from backward direction. Generally, it is assumed that once an error occurs, the decoder discards all the bits until a resynchronization codeword is identified. With RVLC, fewer correctly received bits will be discarded, and the affected area by a transmission error will be reduced.

Intra refreshment and independent segment prediction belong to another group of error resilient tools that can be called as error resilient prediction [86]. Intra refreshment is realized by inserting intra MB's or frames periodically. It

is a way to stop temporal error propagation. Intra MB and frame are coded independently. At the decoder side, they can be decoded without referring to other frames. Thus propagated error cannot affect them. When intra MB refreshment is employed for error resilience purposes, both the number of intra MB's and their spatial location have to be determined. The number of necessary intra MB's is obviously dependent on the quality of the communication network. For independent segment prediction, error propagation is limited by splitting the data into several segments and performing temporal/spatial prediction only within the same segment. In this way, error will only affect the same segment and will not propagate to other segments.

Layered coding and multiple description coding (MDC) are some techniques that will cooperate with channel coding [86]. In layered coding, unequal error protection will be applied to different layers with different importance. The base layer will be transmitted with the highest protection to guarantee a baseline video quality. MDC also codes a source into several substreams, known as descriptions. The resulting descriptions are correlated and have equal importance. Any single description should provide a basic level of quality, and more descriptions together will achieve incremental improvement in visual quality. Since each description provides a certain degree of quality, it must carry fundamental information about the source for independent decoding.

When feedback channel is available, reference picture selection can be employed to choose the "correctly" received frame as the reference frame. This ensures that the reconstructed reference frames at the encoder and the decoder are the same.

Among them, intra refreshment is one of the most commonly used error

resilient tools by inserting intra MB's in the coding process [77, 79, 80, 128, 129]. It reduces the temporal dependency between frames and limits the accumulation of errors in the bitstream. Several papers have addressed intra refreshment in transcoding application [82–85]. These schemes employ the rate distortion (RD) optimized scheme in the intra/inter mode decision for each MB. However, they only consider the channel errors that propagate from previous frames to the current frame. The impact of channel error on the following frames is not taken into account.

Motivated by this, channel error propagation between two consecutive frames is analyzed in this work, and the impact of the propagation is evaluated. An enhanced RD optimized intra/inter mode decision scheme is proposed, which further considers the impact of channel errors that propagate to the next frame. This enhances the robustness of the transcoded bitstream to the packet loss. Considering the availability of information in the next frame, two cases are discussed and detailed schemes are specified for them.

The rest of this chapter is organized as follows. Some preliminaries are presented in section 6.2. In section 6.3, the conventional RD optimized scheme is briefly introduced and the problem is stated. The proposed intra/inter MB mode decision scheme is presented in section 6.4. Experimental results are given in section 6.5 and the conclusion is drawn in section 6.6.

6.2 Preliminaries

In this section, three factors that will affect the overall video quality after the generation of a bitstream are briefly introduced: sender behavior, channel char-

acteristic, and decoder behavior [128].

6.2.1 Sender Behavior

Generally, before a bitstream is transmitted over the networks, video packets are generated. Additional information will be attached to add protection information.

In the encoder, the packetization scheme adopted will affect the decoder distortion of the video. Here, it is assumed that the size of a packet is larger than the size of any MB. Based on this assumption, there are three possible packetization schemes:

- Each packet has the same packet size. In this scheme, an MB may be split into two consecutive packets.
- Each packet contains only one complete MB.
- Each packet contains a complete GOB/slice.

In this work, each packet is assumed to contain one MB or one GOB only. No MB is split into two packets. The loss of each MB comes only from the loss of a packet. Furthermore, in order to make the optimization tractable, no forward error protection (FEC) is adopted. The variation of channel condition has the same impact on each packet.

6.2.2 Channel Characteristic

Channel characteristic is very important to the error resilient scheme adopted. It determines the probability of each packet to be correctly received.

In this work, channel model is characterized as a Bernoulli path model. The Bernoulli path model is a special case of the 2-state Markov chain [128]. Under the Bernoulli path model, the packet loss rate ρ can be obtained by measuring the number of lost packets N_l and correctly received packets N_r at the receiver side by

$$\rho = \frac{N_l}{N_l + N_r}. \tag{6.1}$$

This packet loss rate can be calculated in the receiver side and serve as the feedback information to the encoder. Under the Bernoulli path model, when no FEC protection is adopted, each packet is under the same loss probability.

6.2.3 Decoder Behavior

Decoder behavior is another key factor that will affect the output video quality. In the calculation of distortion, error concealment scheme adopted at the decoder side must be available. In this work, temporal replacement suggested in H.263 standard [6] is adopted. It is described as follows:

- Missing MB's are concealed from the previously decoded picture.
- The reconstructed MV of a missing MB is set to the MV of the MB above the missing one.
- If the MV of the above MB is unavailable, the reconstructed MV is set to zero.
- The MB from the previous picture at the spatial location specified by the reconstructed MV is copied to the location of the missing MB in the

current picture.

6.3 Conventional RD Optimized Scheme

In video coding, the optimization task is to choose the most efficient coded representation in the RD sense. This task is complicated due to the fact that various coding options show varying efficiency at different bit rates and with different scene contents. Intuitively, it is expected to improve the RD performance if the modes can significantly customize the coding for different types of scene statistics, especially if the modes can be applied judiciously to different video sequences and channel constraints. The goal of RD optimization is to optimize the overall fidelity: *minimize distortion* D, *subject to a constraint* R_c, as shown in (6.2).

$$min\ \{D\}, \qquad subject\ to\ R < R_c \tag{6.2}$$

This equation can be solved by the Lagrangian optimization [130]. The Lagrangian formulation of minimization is given by

$$min\ \{J\}, \qquad where\ J = D + \lambda \cdot R \tag{6.3}$$

where cost function J is minimized for a particular value of Lagrange multiplier λ. The Lagrange multiplier optimization can provide significant benefit if it is judiciously applied.

The intra/inter mode decision scheme based on RD optimization is an efficient tool to determine the MB mode based on the cost function defined in (6.3).

In a classical MB mode decision scheme, the cost function of each MB is defined as,

$$J_{MB} = D_{MB} + \lambda_{mode} \cdot R_{MB} \tag{6.4}$$

where D_{MB} and R_{MB} are the source distortion and bits generated by one MB under different coding mode. Each MB can select the coding mode of intra, inter, or even skip. The coding mode that generates the lowest cost will be chosen. In this optimization process, no channel error is considered.

When transmitting this bitstream over an error-prone network, channel error propagation cannot be efficiently stopped. Several papers have addressed this problem. Among them, one scheme that takes the channel errors into account in the RD optimized MB mode decision scheme is introduced in [80]. In [80], the overall expected decoder distortion is incorporated into the conventional RD based framework to dynamically choose MB's to be intra coded. In this method, D_{MB} is redefined as the overall expected decoder distortion given by

$$D_{MB} = \sum_{j \in MB} d_{exp}^{j} \tag{6.5}$$

and

$$\begin{aligned} d_{exp}^{j} &= E\{(f_n^j - \tilde{f}_n^j)^2\} \\ &= (f_n^j)^2 - 2 \cdot f_n^j \cdot E\{\tilde{f}_n^j\} + E\{(\tilde{f}_n^j)^2\}. \end{aligned} \tag{6.6}$$

where f_n^j is the original value of pixel j in frame n, $\tilde{f}_n^j$ is the reconstructed value at the decoder, and $E\{\cdot\}$ is the operation to calculate the expectation

value. In an error-free environment, $\tilde{f}_n^j$ is equal to the encoder reconstruction $\hat{f}_n^j$, whereas in an error-prone environment, $\tilde{f}_n^j$ is a random variable. Its value may vary according to channel condition and error concealment scheme. $E\{\tilde{f}_n^j\}$ and $E\{(\tilde{f}_n^j)^2\}$ are the first and the second moments of the random variable $\tilde{f}_n^j$.

Observed from (6.6), the computation of the expected decoder distortion requires the first and the second moments of the random variable $\tilde{f}_n^j$. In [80], these moments are calculated with recursive optimal per-pixel estimate (ROPE). By ROPE, $E\{\tilde{f}_n^j\}$ and $E\{(\tilde{f}_n^j)^2\}$ are calculated with recursive functions. Two cases are considered depending on whether this pixel belongs to an intra or inter coded MB.

1. Pixel in intra coded MB.

   When the pixel in an intra coded MB is correctly received, its reconstructed values at the decoder ($\tilde{f}_n^j$) and the encoder ($\hat{f}_n^j$) are the same. The probability of this event is $1-\rho$, where ρ is the packet loss rate. However, when this pixel is lost, it is reconstructed according to the error concealment scheme adopted. The possible reconstructed value may be $\tilde{f}_{n-1}^j$ when the reconstructed MV is unavailable, or $\tilde{f}_{n-1}^k$ when the reconstructed MV is available. The probabilities for them are ρ^2 and $\rho(1-\rho)$, respectively.

   Combining these cases, the first and the second moments of $\tilde{f}_n^j$ for a pixel in an intra MB are given by

$$E_I\{\tilde{f}_n^j\} = (1-\rho)\cdot\hat{f}_n^j + \rho^2\cdot E\{\tilde{f}_{n-1}^j\} + \rho(1-\rho)\cdot E\{\tilde{f}_{n-1}^k\} \qquad (6.7)$$

and

$$E_I\{(\tilde{f}_n^j)^2\} = (1-\rho)\cdot(\hat{f}_n^j)^2 + \rho^2\cdot E\{(\tilde{f}_{n-1}^j)^2\} + \rho(1-\rho)\cdot E\{(\tilde{f}_{n-1}^k)^2\}. \quad (6.8)$$

2. Pixel in inter coded MB.

For the pixel in an inter coded MB, the MV of pixel j in frame n is assumed to point to pixel i in frame $n-1$. The residue is $\hat{e}_n^j$, which is given by

$$\hat{e}_n^j = \hat{f}_n^j - \hat{f}_{n-1}^i. \quad (6.9)$$

where $\hat{f}_{n-1}^i$ is the reconstructed value of pixel i of frame $n-1$ at the encoder.

If the pixel in an inter coded MB is correctly received, its reconstructed value at the decoder is given by

$$\tilde{f}_n^j = \hat{e}_n^j + \tilde{f}_{n-1}^i. \quad (6.10)$$

The probability of this event is $1-\rho$. When this pixel is lost, the possible reconstructed value may be $\tilde{f}_{n-1}^j$ when the reconstructed MV is unavailable, or $\tilde{f}_{n-1}^k$ when the reconstructed MV is available. The probabilities for them are ρ^2 and $\rho(1-\rho)$, respectively. This is the same with the intra coded case.

Then, the first and the second moments of $\tilde{f}_n^j$ for a pixel in an inter MB

are given by

$$\begin{aligned} E_P\{\tilde{f}_n^j\} &= (1-\rho)\cdot(\hat{e}_n^j + E\{\tilde{f}_{n-1}^i\}) + \rho^2 \cdot E\{\tilde{f}_{n-1}^j\} \\ &\quad +\rho(1-\rho)\cdot E\{\tilde{f}_{n-1}^k\} \end{aligned} \tag{6.11}$$

and

$$\begin{aligned} E_P\{(\tilde{f}_n^j)^2\} &= (1-\rho)\cdot\{(\hat{e}_n^j + \tilde{f}_{n-1}^i)^2\} + \rho^2 \cdot E\{(\tilde{f}_{n-1}^j)^2\} \\ &\quad +\rho(1-\rho)\cdot E\{(\tilde{f}_{n-1}^k)^2\}. \end{aligned} \tag{6.12}$$

Here, the expected distortion at the decoder is calculated recursively at the encoder. The encoder can exploit this result directly in its coding decision. At the same time, λ_{mode} defined in [130] is not suitable for an error-prone environment. Based on buffer status, λ_{mode} is updated per frame via

$$\lambda_{n+1} = \lambda_n(1 + \frac{1}{5R_{target}}(\sum_{i=1}^{n} R_i - n \cdot R_{target})) \tag{6.13}$$

where R_{target} is the target bit rate, R_i is the number of bits generated by each frame, and n is the frame number.

This optimization scheme based on ROPE is called ROPE-RD. It assumes that, each MB has the equal possibility to be lost. The propagated channel error is incorporated into RD optimization process, and MB's are adaptively intra refreshed. However, it only considers the channel errors that propagate from previous frames. The impacts of channel errors on the following frames are overlooked.

In error resilient transcoding, it is highly desired to take into account the

channel distortion propagated to the following frames in the RD optimized MB mode decision. This is because, in block-based motion compensation, the reference block is not aligned to the block boundary. As shown in Figure 6.2, some pixels in the reference frame are referred to more than once, while some pixels are never referred to. For those pixels that are never referred to, the accumulated channel error will not affect the following frames. Channel distortion propagation is stopped naturally. For pixels referred to more than once, the accumulated channel error may spread and aggravate the distortion propagation. Furthermore, channel error may have different impacts when propagated to different pixels in the following frames.

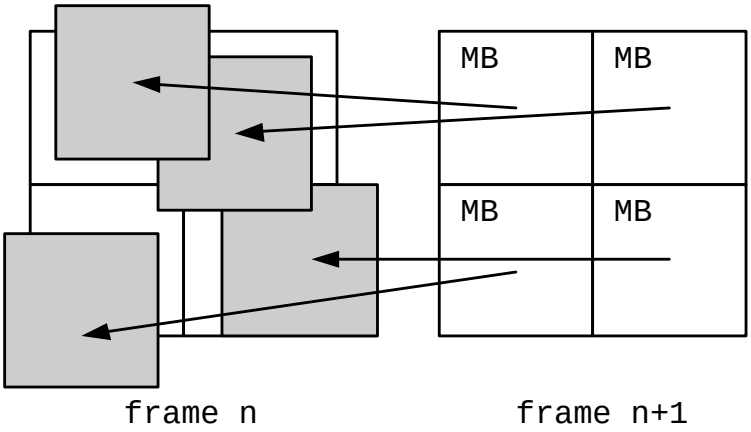


Figure 6.2: Block-based motion prediction

It has been shown that, the channel errors accumulated at the current frame have different impacts on the following frames, which have not been considered by the previous schemes. However, it is unrealistic to consider the impact of channel distortion propagation to all the following frames for it is too complicated. Only considering the propagation between two consecutive frames, an optimal solution for global sequence can be derived. Motivated by the above reasons, a scheme is proposed that further considers the channel distortion prop-

agation in intra refresh transcoding. Details are discussed in the following section.

6.4 Proposed MB Mode Decision Scheme

In the proposed MB mode decision scheme, two cases are specified according to the availability of the next frame information. If the information of next frame is available, the proposed channel distortion propagation based RD optimization (CDP-RD) utilizes the MV and pixel values of the next frame. When the MV and pixel values of the next frame are unavailable, a modified CDP-RD (MCDP-RD) is adopted based on the predicted MV.

6.4.1 CDP-RD MB Mode Decision Scheme

In this subsection, CDP-RD is discussed with consideration of the channel distortion propagation. Distortion in the next frame that is contributed by the channel errors of current frame will be counted into corresponding pixels in the current frame. In CDP-RD, the information of the next frame is available. Transcoder can collect the MV and pixel values from the input bitstream.

In ROPE-RD, D_{MB} is defined as the expected decoder distortion within MB as shown in (6.5). In CDP-RD, this distortion is modified to be

$$D_{MB} = \sum_{j \in MB} (d^j_{exp} + d^j_{ppg}) \tag{6.14}$$

and

$$d_{ppg}^{j} = \sum_{i \in frame\ n+1} d_{ppg}(j,i) \tag{6.15}$$

where $d_{ppg}(j,i)$ is the propagated channel distortion in the next frame that is caused by the channel error of pixel j in the current frame. As stated in section 6.3, channel error accumulated in the current frame has different impacts on the next frame. Adding d_{ppg}^{j} into total distortion takes into account the impacts of channel error propagated to the next frame. Detailed analysis of the propagated channel distortion $d_{ppg}(j,i)$ is given below.

6.4.1.1 Propagated Channel Distortion d_{ppg}^{j}

Each pixel in frame $n+1$ is possible to be affected by the channel distortion from frame n due to motion compensation or error concealment. Then $d_{ppg}(j,i)$ is composed of three parts given as

$$d_{ppg}(j,i) = d_{ppg}^{mc}(j,i) + d_{ppg}^{ec1}(j,i) + d_{ppg}^{ec2}(j,i). \tag{6.16}$$

where $d_{ppg}^{mc}(j,i)$ is the propagated channel distortion due to motion compensation, $d_{ppg}^{ec1}(j,i)$ and $d_{ppg}^{ec2}(j,i)$ are the propagated channel distortions due to error concealment.

1. Propagated channel distortion due to motion compensation ($d_{ppg}^{mc}(j,i)$):

   When the MV's of pixels in frame $n+1$ point to pixel j in frame n, these pixels may be affected by the channel error in $\tilde{f}_{n}^{j}$ through motion compensation. Let f_{n+1}^{i} denote the original value of pixel i in frame $n+1$, which

refers to pixel j in frame n, and $\hat{f}_{n+1}^{i}$ denote its encoder reconstruction, its decoder reconstruction is given by

$$\tilde{f}_{n+1}^{i} = \hat{e}_{n+1}^{i} + \tilde{f}_{n}^{j} \tag{6.17}$$

where

$$\hat{e}_{n+1}^{i} = \hat{f}_{n+1}^{i} - \hat{f}_{n}^{j}. \tag{6.18}$$

According to (6.6), the expected decoder distortion of this pixel is

$$\begin{aligned} \tilde{d}_{n+1}^{i} &= E\{(f_{n+1}^{i} - \tilde{f}_{n+1}^{i})^2\} \\ &= (f_{n+1}^{i})^2 - 2f_{n+1}^{i}E\{\hat{e}_{n+1}^{i} + \tilde{f}_{n}^{j}\} + E\{(\hat{e}_{n+1}^{i} + \tilde{f}_{n}^{j})^2\}. \end{aligned} \tag{6.19}$$

When there is no channel error in $\tilde{f}_{n}^{j}$, i.e. $\tilde{f}_{n}^{j} = \hat{f}_{n}^{j}$, the expected decoder distortion is

$$\hat{d}_{n+1}^{i} = (f_{n+1}^{i})^2 - 2f_{n+1}^{i}(\hat{e}_{n+1}^{i} + \hat{f}_{n}^{j}) + (\hat{e}_{n+1}^{i} + \hat{f}_{n}^{j})^2. \tag{6.20}$$

The difference between these two expected decoder distortions is given as

follows:

$$\begin{aligned}
\Delta d_{n+1}^{i} &= \tilde{d}_{n+1}^{i} - \hat{d}_{n+1}^{i} \\
&= E\{(\tilde{f}_n^j)^2\} - (\hat{f}_n^j)^2 + 2\hat{e}_{n+1}^{i}(E\{\tilde{f}_n^j\} - \hat{f}_n^j) - 2f_{n+1}^{i}(E\{\tilde{f}_n^j\} - \hat{f}_n^j) \\
&= E\{(\tilde{f}_n^j)^2\} - (\hat{f}_n^j)^2 - 2(f_{n+1}^{i} - e_{n+1}^{i} + e_{n+1}^{i} - \hat{e}_{n+1}^{i})(E\{\tilde{f}_n^j\} - \hat{f}_n^j) \\
&= E\{(\tilde{f}_n^j)^2\} - (\hat{f}_n^j)^2 - 2\hat{f}_n^j(E\{\tilde{f}_n^j\} - \hat{f}_n^j) - 2(e_{n+1}^{i} - \hat{e}_{n+1}^{i})(E\{\tilde{f}_n^j\} - \hat{f}_n^j).
\end{aligned} \tag{6.21}$$

It is obvious that this difference is caused by the channel error within $\tilde{f}_n^j$. In (6.21), $e_{n+1}^{i} - \hat{e}_{n+1}^{i}$ is the source coding error. Its expectation is zero. Then the expected value of Δd_{n+1}^{i} becomes

$$\begin{aligned}
E\{\Delta d_{n+1}^{i}\} &= E\{E\{(\tilde{f}_n^j)^2\} - E\{(\hat{f}_n^j)^2\} - 2\hat{f}_n^j(E\{\tilde{f}_n^j\} - E\{\hat{f}_n^j\})\} \\
&\quad -E\{2(e_{n+1}^{i} - \hat{e}_{n+1}^{i})(E\{\tilde{f}_n^j\} - E\{\hat{f}_n^j\})\} \\
&= E\{(\tilde{f}_n^j)^2\} + (\hat{f}_n^j)^2 - 2\hat{f}_n^j E\{\tilde{f}_n^j\}.
\end{aligned} \tag{6.22}$$

The expected value of Δd_{n+1}^{i} is the channel distortion propagated through motion compensation. Then the value of the propagated channel distortion $d_{ppg}^{mc}(j,i)$ is given by

$$d_{ppg}^{mc}(j,i) = (1-\beta)(1-\rho)E\{\Delta d_{n+1}^{i}\} \tag{6.23}$$

where β is the intra refresh rate and ρ is the packet loss rate. For these pixels referring to pixel j in frame n, only when they are inter coded and correctly received, can they be affected by $\tilde{f}_n^j$ through motion compensation. The probability for this is $(1-\beta)(1-\rho)$. In an RD optimized mode

decision scheme, the intra refresh rate is not a constant value. It is not available before fully coding each frame. In (6.23), β can be assumed as the average intra refresh rate of previous coded frames.

For those pixels that do not refer to pixel j in frame n, $d_{ppg}^{mc}(j,i)$ is set to zero.

2. Propagated channel distortion due to error concealment I ($d_{ppg}^{ec1}(j,i)$):

   When some pixels in frame $n+1$ are lost, their reconstructed MV's may point to pixel j in frame n. The corresponding reconstructed value of these pixels becomes $\tilde{f}_n^j$. Channel distortion propagates from $\tilde{f}_n^j$ through error concealment. Similar to (6.19), (6.20), and (6.21), the difference between the expected decoder distortions when $\tilde{f}_n^j$ has or does not have channel error is written as

$$\Delta d_{n+1}^i = -2f_{n+1}^i(E\{\tilde{f}_n^j\} - \hat{f}_n^j) + E\{(\tilde{f}_n^j)^2\} - (\hat{f}_n^j)^2 \tag{6.24}$$

   where f_{n+1}^i is the original value of this lost pixel. The expected value of this difference is given by

$$E\{\Delta d_{n+1}^i\} = E\{(\tilde{f}_n^j)^2\} - (\hat{f}_n^j)^2 - 2f_{n+1}^i(E\{\tilde{f}_n^j\} - \hat{f}_n^j). \tag{6.25}$$

   This difference is caused by the channel error within $\tilde{f}_n^j$. The expected value of Δd_{n+1}^i is the channel distortion propagated through error concealment. Then the value of propagated channel distortion $d_{ppg}^{ec1}(j,i)$ is given by

$$d_{ppg}^{ec1}(j,i) = \rho(1-\rho)E\{\Delta d_{n+1}^i\}. \tag{6.26}$$

Here, the probability that a pixel is lost and its reconstructed MV is available is $\rho(1-\rho)$. This is considered in the calculation of $d_{ppg}^{ec1}(j,i)$. It should be noted that, this distortion is affected by the pixel value f_{n+1}^{i}.

When the reconstructed MV does not point to pixel j in frame n, $d_{ppg}^{ec1}(j,i)$ is set to zero.

3. Propagated channel distortion due to error concealment II ($d_{ppg}^{ec2}(j,i)$):

   Pixel j in frame $n+1$ has a special relation with pixel j in frame n. They have the same spatial location. When pixel j in frame $n+1$ is lost and the reconstructed MV is unavailable, it will be replaced by $\tilde{f}_n^j$ according to the adopted error concealment scheme. This leads to additional channel error propagation. Similar to $d_{ppg}^{ec1}(j,i)$, the propagated channel distortion $d_{ppg}^{ec2}(j,i)$ is given by

$$d_{ppg}^{ec2}(j,i) = \rho^2 E\{\Delta d_{n+1}^{i}\} \tag{6.27}$$

   where

$$E\{\Delta d_{n+1}^{i}\} = \begin{cases} E\{(\tilde{f}_n^j)^2\} - (\hat{f}_n^j)^2 - 2f_{n+1}^{i}(E\{\tilde{f}_n^j\} - \hat{f}_n^j) & i = j \\ 0 & i \neq j \end{cases} \tag{6.28}$$

   In the calculation of $d_{ppg}^{ec2}(j,i)$, the probability, that a pixel is lost and its reconstructed MV is unavailable, is ρ^2.

Adding the propagated channel distortion into the corresponding pixel in frame n, new distortion reflects not only the source coding error and the channel error from previous frames, but also the impact of channel error on the next frame. Table 6.1 summarizes the calculation of propagated channel distortion.

Table 6.1: Relation between pixels in frame $n+1$ with pixel j in frame n

	Relation	Condition	Probability
case 1 $d_{ppg}^{mc}(j,i)$	MV referring to pixel j in frame n	inter coded and correctly received	$(1-\beta)(1-\rho)$
case 2 $d_{ppg}^{ec1}(j,i)$	Reconstructed MV referring to pixel j in frame n	lost and reconstructed MV available	$\rho(1-\rho)$
case 3 $d_{ppg}^{ec2}(j,i)$	In the same spatial location with pixel j in frame n	lost and reconstructed MV unavailable	ρ^2

6.4.1.2 Modified λ_{mode}

Because the distortion model in CDP-RD is different from the conventional RD based model, the optimal value for λ_{mode} may also change. It is observed that the propagated channel distortion considers the intra refresh rate and the average packet loss rate. This new term is introduced in the distortion part in the cost function. To balance the importance of distortion and rate, the initial value of λ_{mode} (λ_0) in (6.13) is modified to consider the propagated channel error

$$\lambda_0^{'} = \lambda_0 \cdot [2 - \beta \cdot (1-\rho)] \tag{6.29}$$

where, β and ρ are the intra refresh rate and the packet loss rate, respectively. As the intra refresh rate is unavailable, we assume its value is equal to the average packet loss rate in the simulation. For the following frame, λ_{mode} is updated in the same way as (6.13). Based on the proposed RD optimized intra/inter mode decision scheme, the robustness of the bitstream can be enhanced.

6.4.1.3 Computational Complexity Analysis

The performance improvement of the CDP-RD MB mode decision scheme is obtained at the cost of computation. Compared with ROPE-RD, additional computation is needed to calculate d^j_{ppg}. In addition to the calculation of d^j_{exp}, about 20 addition/multiplication operations per pixel on average are needed. This additional computation is comparable with the computation of d^j_{exp}, which requires totally 27 addition/multiplication operations per pixel. At the same time, some intermediate results in d^j_{exp} can be reused. Therefore, actual computation can be further reduced.

6.4.2 MCDP-RD MB Mode Decision Scheme

In some applications, even a single frame delay is not acceptable. Or, the input buffer of transcoder is not big enough to accommodate the next frame when transcoding the current frame. This means that the MV's and pixel values of the next frame are not available when transcoding the current frame. However, they are necessary in the CDP-RD MB mode decision scheme. Therefore, information has to be estimated or reconstructed to evaluate the error propagation.

In this subsection, a modified CDP-RD (MCDP-RD) is considered to calculate the propagated channel distortion in intra refresh transcoding. In this method, the MV of the next frame is predicted based on the previous frame information. Also, because the pixel values of the next frame are unavailable, a simplified model is adopted to calculate the propagated channel distortion d^j_{ppg}.

6.4.2.1 Prediction of Motion Vector

Similar to [131, 132], the prediction of MV is based on the motion information in the previous frames. Here, motion information is collected from the input bitstream, not from the transcoded bitstream. This is because that, in the transcoded bitstream, some MB's are coded in intra mode to stop the channel error propagation. It cannot provide sufficient motion information for predicting the future motion. In the input bitstream, most MB's are coded in inter mode except scene change. Motion information is accurate. The prediction of MV includes three steps:

1. Collection of motion information

   When coding frame n, MV's of frame n and frame $n-1$ are needed. As shown in Figure 6.3, for one pixel in frame n, its MV is MV_n. This MV points to a pixel in frame $n-1$. The MV of pixel referred to in frame $n-1$ is MV_{n-1}. These two MV's are used to predict the motion of a pixel in frame $n+1$.

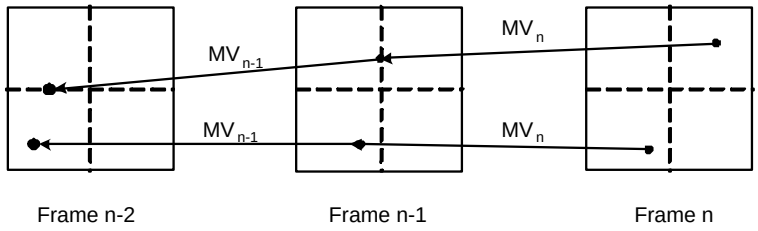


Figure 6.3: Motion vector tracing back

2. Motion projection based on linear model

   In motion prediction process, the motion of each pixel is assumed as a

linear motion. Its motion is based on the model given by

$$MV_{n+1} = MV_n + a_n \cdot \Delta t \tag{6.30}$$

where

$$a_n = (MV_n - MV_{n-1})/\Delta t. \tag{6.31}$$

Here, $a_n = 0$ means that the motion of this pixel is a constant movement; when $a_n < 0$, the motion of this pixel is decelerating; when $a_n > 0$, the motion of this pixel is accelerating. The terms MV_n and MV_{n-1} are the motion information collected from frame n and frame $n-1$ corresponding to a single pixel.

According to this linear model, the calculated MV can be considered as the MV of a pixel in frame $n+1$. Assuming that the pixel in frame n is located at (x_j, y_j), it is projected to the pixel located at $(x_j - MV_{n+1,x}, y_j - MV_{n+1,y})$ in frame $n+1$, whose MV is $(MV_{n+1,x}, MV_{n+1,y})$.

In this motion projection process, there are some special cases:

- *MV_n in frame n is unavailable.*

  This is because that the pixel in frame n is intra coded. In this case, this pixel is skipped without further calculation.

- *MV_{n-1} in frame $n-1$ is unavailable.*

  This is because that MV_n points to a pixel in frame $n-1$ outside the frame boundary, or points to a pixel within an intra MB. In this case, MV_{n+1} is set to be the same value as MV_n.

- *Pixel in frame $n+1$ is unavailable.*

When the calculated pixel position $(x_j - MV_{n+1,x}, y_j - MV_{n+1,y})$ in frame $n+1$ is out of the frame boundary, the motion of this case is not considered.

3. MB based MV modification

   From previous steps, MV's for pixels in frame $n+1$ have been predicted. However, some pixels may not have MV, while some pixels may have more than one MV. So, there may exist different MV's that appear in one MB. In this step, a block-based spatial regulation scheme is designed to assign one MV to all pixels in one MB.

   In block-based coding system, pixels within one MB have the same MV. Hence, in the prediction of MV, it is desirable that all the pixels within one MB have the same MV. After motion projections, pixels within one MB may have different MV's. The MV adopted by most pixels can be selected as the MV for this MB. Then all the pixels within this MB have the same MV.

   There may be some MB's, which do not have MV's assigned. For these MB's, MV's can be obtained by taking the average MV's of their neighboring MB's.

With these three steps, all MV's of frame $n+1$ are reconstructed. They are used to estimate the propagated channel distortion.

6.4.2.2 Propagated Channel Distortion

In (6.25) and (6.28), the pixel value of the next frame is required to calculate the propagated channel distortion. In MCDP-RD, a simplified model is adopted to

estimate the propagated channel distortion when the pixel value is unavailable.

In this simplified model, all MB's in the next frame are assumed to be correctly received. Channel distortion is propagated through motion compensation only. Similar to (6.16) and (6.23), $d_{ppg}(j,i)$ is redefined as

$$\begin{aligned} d_{ppg}(j,i) &= d_{ppg}^{mc}(j,i) \\ &= (1-\beta)E\{\Delta d_{n+1}^{i}\} \end{aligned} \tag{6.32}$$

where $E\{\Delta d_{n+1}^{i}\}$ is defined in (6.22). This is based on the assumption that the packet loss rate is low.

6.4.2.3 Simplified λ_{mode}

As the distortion model in MCDP-RD is different from that in CDP-RD, the initial value of λ_{mode} will also be modified. Same as (6.29), a new initial value is defined as

$$\lambda_0^{'} = \lambda_0 \cdot (2-\beta). \tag{6.33}$$

It is updated in the same way as (6.13).

6.4.2.4 Computational Complexity Analysis

In MCDP-RD, since MV prediction for the next frame is required, 2 more addition/multiplication operations per pixel are needed. At the same time, as a simplified model is adopted to calculate the propagated channel distortion, the computation is saved significantly. It requires only 10 addition/multiplication

operations, which are much fewer than that needed in CDP-RD.

6.5 Experimental Results

To evaluate the performance of CDP-RD and MCDP-RD, extensive experiments have been conducted to compare them with the conventional method (ROPE-RD) [80]. In the experiments, the TMN8 rate control scheme is adopted. GOP structure IPPP... is adopted and only the first frame is intra coded. Each packet contains one GOB. A random packet loss generator is used to drop the packet according to the required packet loss rate (see Figure 6.4).

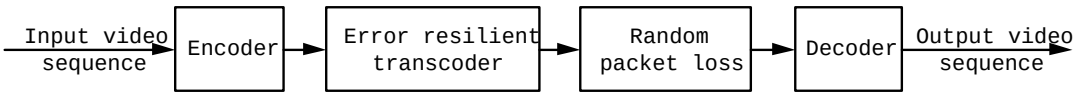
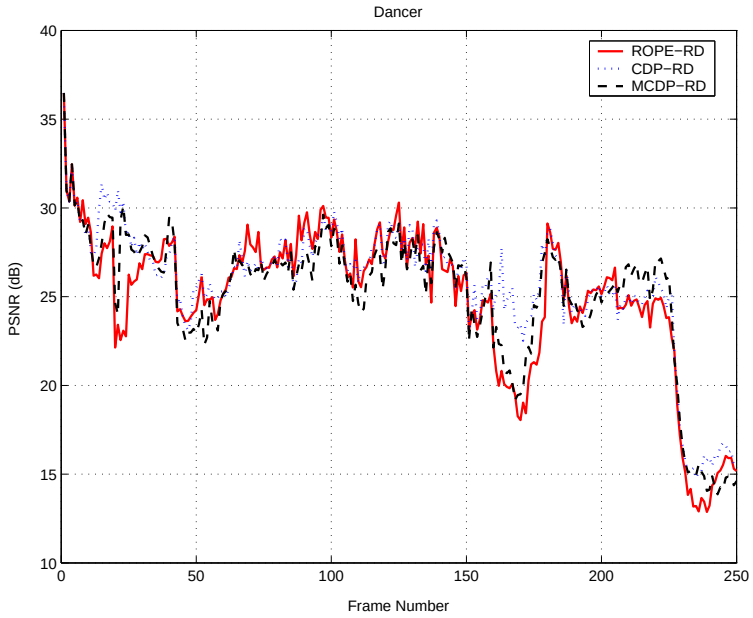
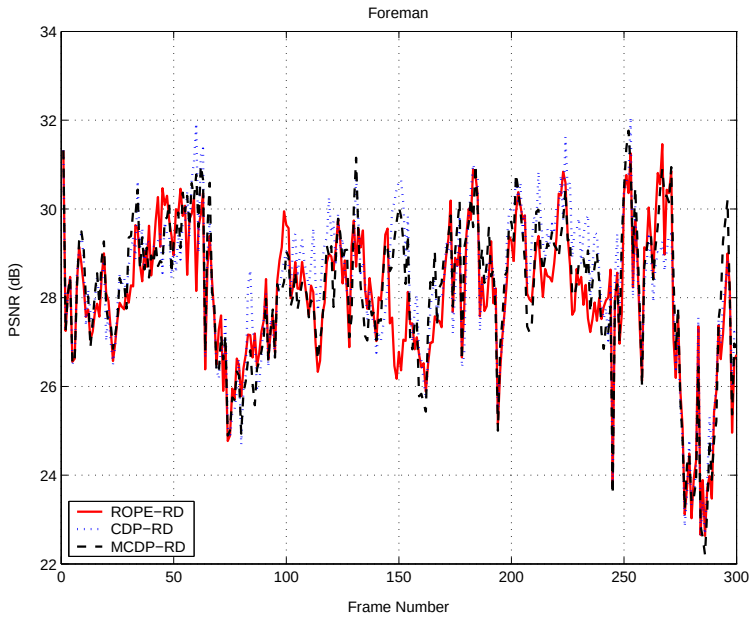


Figure 6.4: Setup for the experiment

First, a frame by frame comparison between the proposed methods and ROPE-RD is given (see Figure 6.5). The test sequences are "Dancer" and "Foreman". "Dancer" is transcoded from 494kbps to 128kbps and "Foreman" is transcoded from 540kbps to 300kbps, respectively. The packet loss rate is 10%. In order to give a fair comparison, exactly the same packets are lost in 3 methods. CDP-RD and MCDP-RD do not always present higher video quality than ROPE-RD through the whole sequence. This is because ROPE-RD is an optimal scheme for single frame. CDP-RD and MCDP-RD try to enhance the robustness not only for the current frame, but also for the following frames. It is an optimal solution for the entire video. For "Dancer", the average PSNR for CDP-RD and MCDP-RD are 25.70dB and 25.08dB respectively. They are higher than that of ROPE-RD, which is 24.96dB. For "Foreman", the average

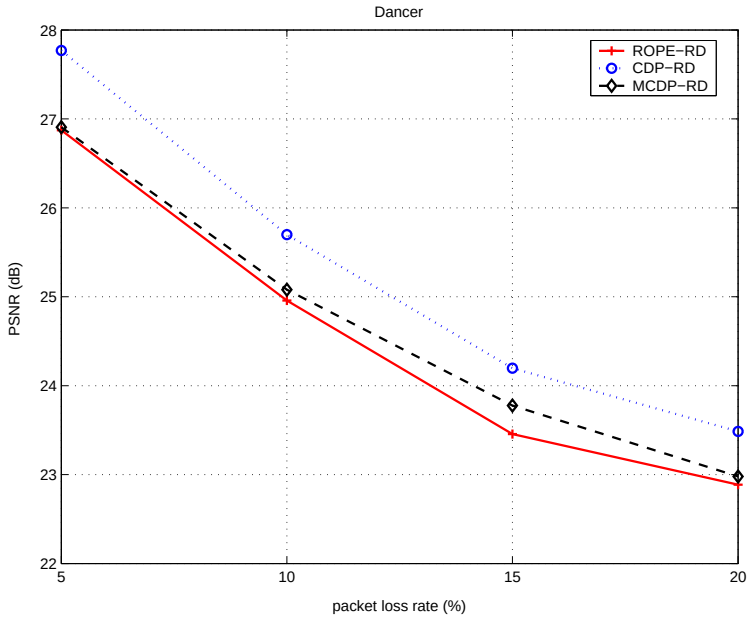


(a)

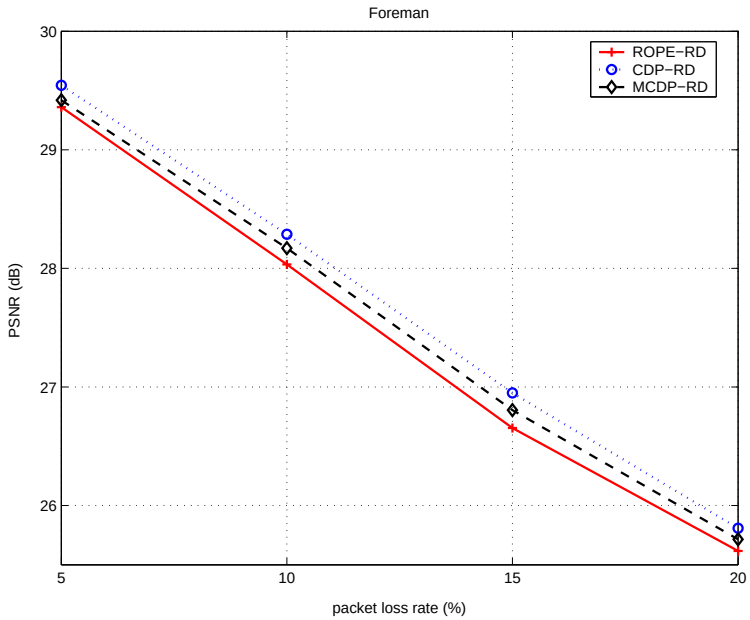


(b)

Figure 6.5: Frame by frame comparison: (a) Dancer; (b) Foreman



(a)



(b)

Figure 6.6: PSNR comparison under different packet loss rate: (a) Dancer; (b) Foreman

PSNR for these three methods are 28.29dB, 28.17dB, and 28.03dB, respectively. In terms of average PSNR, CDP-RD outperforms ROPE-RD. Although MCDP-RD is a simplified scheme with estimation, it also presents improvement over ROPE-RD.

It is noted that, the intra refresh rate per frame is not a constant value in these three methods. This is because in RD optimized intra refreshment, the coding mode of each MB is determined according to the available bandwidth, the packet loss rate, and the cost function applied. For example, in the last experiment with the video "Dancer", average intra refresh rates are 24.8%, 25.0%, and 24.7% for ROPE-RD, CDP-RD, and MCDP-RD, respectively. These average intra refresh rates are very close. However, due to different cost functions that adopted, different MB's are intra refreshed in each frame. These different cost functions result in different performances of error resilient transcoding.

Figure 6.6 shows the error resilient performance of these three methods under different packet loss rates. In Figure 6.6(a), the test video is "Dancer", and in Figure 6.6(b), the test video is "Foreman". The bit rate conversion is the same with that in the previous experiment. Under different packet loss rates, CDP-RD outperforms ROPE-RD in terms of higher PSNR. Even MCDP-RD can present higher PSNR than ROPE-RD.

To further evaluate the performance of these three methods, an experiment is designed to compare them when there is a mismatch between the packet loss rate assumed by the encoder and the actual packet loss rate in the channel. In the previous experiments, the packet loss rate is assumed to be available for the encoder. In practical applications, feedback information may be delayed from the decoder. The packet loss rate assumed by the encoder in its optimiza-

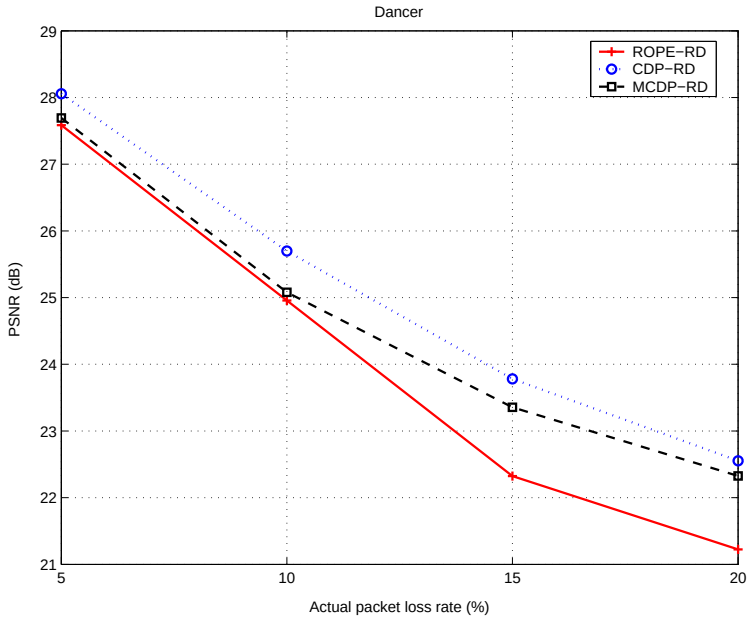


Figure 6.7: Dancer: performance comparison of different methods with mismatch ($\rho = 10\%$)

tion may be different from the actual packet loss rate. It is desirable that the transcoded bitstream is resilient to the mismatch. In this experiment, the estimated packet loss rate is set to 10% in the transcoding optimization. The input bitstream is transcoded based on this value to generate a resilient bitstream. In channel transmission, the actual packet loss rates vary from 5% to 20%. Experimental result is shown in Figure 6.7. When the actual packet loss rate is less than the estimated one, the performance of ROPE-RD is slightly lower than CDP-RD and MCDP-RD. However, with the increasing of the actual packet loss rate, especially when it is larger than the estimated one (10%), the output video quality decreases significantly. This shows that ROPE-RD is not robust enough to endure the variation of channel condition. On the contrary, the performances of CDP-RD and MCDP-RD are better than that of ROPE-RD, especially when the actual channel condition is worse than what is assumed in the transcoder.

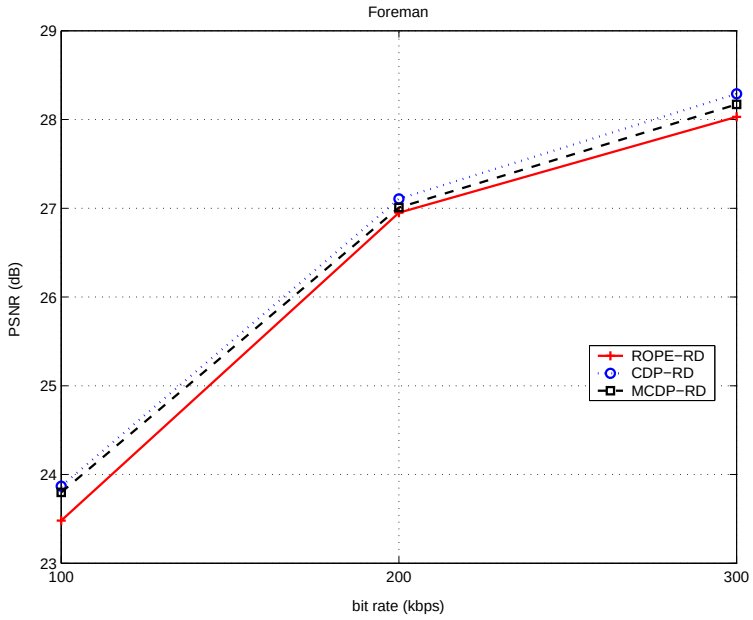


Figure 6.8: Foreman: performance comparison under different bit rate

Next, the performances of these three schemes under different bit rates are compared (see Figure 6.8). The test video is "Foreman". The input bitstream is transcoded to 100kbps, 200kbps, and 300kbps, respectively. The average packet loss rate is set to 10%. As shown in Figure 6.8, CDP-RD and MCDP-RD present better video quality in terms of PSNR than that of ROPE-RD under different target bit rates.

Average PSNR comparison of these methods under different packet loss rates is shown in Table 6.2. CDP-RD presents higher PSNR over the other two methods, and MCDP-RD presents a little improvement over ROPE-RD.

6.6 Summary

In this chapter, error propagation is studied for error resilient transcoding. The proposed scheme utilizes the error propagation to the next frame to evaluate

Table 6.2: Comparison of average PSNR (in dB) between different intra refresh methods

Sequence	Original bit rate	Target bit rate	Method	Packet Loss Rate			
				5%	10%	15%	20%
Dancer	494kbps	128kbps	ROPE-RD	26.87	24.96	23.46	22.89
			CDP-RD	27.77	25.70	24.20	23.49
			MCDP-RD	26.91	25.08	23.78	22.98
Mobile	1800kbps	300kbps	ROPE-RD	21.83	20.44	19.45	18.12
			CDP-RD	21.99	20.54	19.54	18.33
			MCDP-RD	21.84	20.50	19.53	18.15
Salesman	139kbps	128kbps	ROPE-RD	32.39	31.26	30.88	30.30
			CDP-RD	32.57	31.44	30.92	30.54
			MCDP-RD	32.44	31.34	30.90	30.50
Foreman	540kbps	300kbps	ROPE-RD	29.36	28.03	26.65	25.62
			CDP-RD	29.54	28.29	26.95	25.81
			MCDP-RD	29.42	28.17	26.81	25.72
Suzie	269kbps	128kbps	ROPE-RD	31.08	30.44	29.45	28.88
			CDP-RD	31.19	30.58	29.70	29.13
			MCDP-RD	31.13	30.54	29.48	29.12

the distortion of the current frame. It considers not only the channel error from previous frames, but also channel error propagated to the following frames. This gives a global optimization in error resilient transcoding. The application of this concept is applied in the RD optimized MB mode decision to adaptively intra refresh MB's. This enhances the performance of error resilient transcoder, and also improves the robustness of the generated bitstream against the packet loss. In the case that the next frame information is unavailable, an MV reconstruction scheme is proposed to predict the MV of the next frame. This benefits the calculation of channel distortion propagated to the next frame.

Chapter 7

Conclusions and Recommendations

In this thesis, the main purpose is to design quality enhancement algorithms for video transcoding. In previous chapters, the research works for this purpose have been presented. In this chapter, the main development of this thesis is summarized in section 7.1. Several recommendations are discussed for the future work in section 7.2.

7.1 Conclusions

Video streaming over the Internet has evolved as one of the major technical fields in recent years. Due to the variety of user devices and the wide variation of the available bandwidth with different transmission errors over the Internet, it is desirable to introduce an efficient scheme for video adaptation in practical applications.

As an efficient tool to realize video adaptation in real time, video transcoding has been a hot research topic in recent years. The purpose of research is to provide qualified service to end users via transcoding. In this thesis, different transcoding strategies were studied. Several new schemes for transcoding were proposed. Works are briefly concluded in the following.

1. **A literature review of video transcoding scheme was presented**.

   Various techniques have been described for performing video transcoding in video communication environment. It was observed that in the transcoding process, the most important thing is how to reuse the information from the pre-coded bitstream to enhance the video quality of the transcoded bitstream and reduce the computation as possible.

   According to the architecture of transcoding, spatial domain and transform domain transcoding were introduced first. Simplified architectures and the transform domain realization were developed for saving the computation. The most efficient way to save the computation in transcoding, i.e. MV reusing, was discussed in details. This is because motion estimation is the most time-consuming operation in the coding process. Reusing the MV's from the input bitstream can save the computation significantly.

   According to the applications, video transcoding has been classified into four different categories: rate-related transcoding, format-related transcoding, error-related transcoding, and information-related transcoding. A brief introduction of these four types of transcoding has been given and related researches were included.

2. **Arbitrary resizing schemes in the transform domain were proposed**.

Arbitrary resizing is a useful tool in downsizing transcoding. In order to reduce the computation and enhance the video quality, resizing in the DCT domain is preferred. Two schemes were introduced for this purpose.

The first proposed algorithm aims to realize arbitrary resizing especially for the inter frame. Computation can be saved with the look-up-table technique. This algorithm benefits the inter frame downsizing since the block partition style is the same as that of the MV reusing scheme. These two schemes can co-operate well for downsizing transcoding, which helps to reduce the mismatch in the downsizing process.

Next, a more general resizing scheme was developed. A two-stage structure was introduced to realize arbitrary resizing. This enables the resizing process with high flexibility. This scheme can realize all the methods proposed by the other researchers. With the consideration of anti-aliasing, some criteria were proposed to select parameters. These criteria guarantee the baseline video quality in arbitrary resizing.

3. **A frame size selection scheme was designed for arbitrary downsizing transcoding**.

   When arbitrary downsizing is adopted, fine gradual reduction of the bit rate and the video quality becomes feasible. In downsizing transcoding, a suitable reduced frame size is necessary for the transcoded bitstream. Frame size selection for arbitrary downsizing transcoding is a key issue to present a high video quality. Based on some simulation results, the frame size selection problem was converted to the bit rate estimation problem. Furthermore, a bit rate estimation scheme was proposed to estimate the bit rate lower bounds of different frame sizes. With these lower bounds, it

is easy to select a suitable frame size for arbitrary downsizing transcoding.

4. **A frame skipping transcoding scheme was proposed based on motion information consideration**.

   For frame skipping transcoding, one important issue is how to reduce the motion information loss and avoid "jerky" effect. Based on the observation from human visual perception, motion change was introduced to evaluate the motion continuity of the video. The motion continuity information is another aspect of the motion information that should be considered.

   With the combination of the motion continuity and the motion activity information, importance of the frame is evaluated. A frame skipping transcoding scheme based on the motion information consideration was designed to minimize the motion information loss. This enhances the spatial quality of the transcoded bitstream. Specially, this scheme reduces the nonlinear motion in the transcoded bitstream. When the motion compensated interpolation is adopted, the proposed scheme benefits the reconstruction process significantly.

5. **An MB mode decision scheme was designed for error resilient transcoding**.

   Intra refreshment is an efficient tool to reduce the error propagation in the error-prone network transmission. In previous error resilient transcoding scheme, rate distortion optimization was introduced to determine the MB mode adaptively. However, these schemes are the optimal solutions for single frame only, not for the global sequence. With the consideration of error propagation to the next frame, an enhanced MB mode decision scheme was designed for the global sequence. In this scheme, the impact of

channel error in the current frame to the next frame is taken into account. This enhances the robustness of the transcoded bitstream over the error in the channel.

7.2 Recommendations for Future Work

There are many interesting research topics in video transcoding. Although some research topics have already been studied in this thesis, some possible extensions of the work in this thesis may be considered for future research studies.

The first problem to be considered is rate-related transcoding. In order to reduce the overall bit rate, the spatial resolution and temporal resolution of the video could be reduced. How to evaluate the output visual quality between different transcoding schemes is a hard research problem. For example, a CIF bitstream with frame rate 30 $frames/s$ can be transcoded with several different solutions: QCIF format with the original frame rate, CIF format with the frame rate of 10 $frames/s$, or CIF format with the frame rate of 15 $frames/s$. If arbitrary downsizing is adopted, the spatial resolution may have more candidates. For those lower temporal resolutions, simply evaluating the visual quality in terms of PSNR is not an optimal solution due to the vision persistence of the human eyes. In this process, it is very hard to determine the scaling level for different schemes. A simple solution can be described as follows.

When the pre-coded bitstream is transcoded, the motion information of this bitstream can be evaluated. According to some subjective evaluation results, the preferred temporal resolution can be first decided. Based on this pre-defined temporal resolution, the spatial resolution of the transcoded bitstream is deter-

mined. Although in Chapter 4, frame size selection was solved with the proposed scheme, this is only a solution when the frame rate of the transcoded bitstream is the same as the pre-coded bitstream. When the frame rate is changing, the proposed scheme may not be suitable to solve it. This should be solved in future works.

Developing a global rate control strategy that utilizes the joint spatial-temporal transcoding schemes is another interesting topic. This approach should be able to maximize the presentation of diverse video sequences under different channel constraints and user requirements. Especially when the joint source-channel coding is considered, how to adaptively allocate bits for each frame is a crucial step to enhance the output video quality.

Format-related transcoding is also a challenging issue. With the new coding standard developed, format conversion becomes more popular. The main issue in format-related transcoding is how to reduce the computation. For example, in H.264, not only motion estimation is a time consuming process, mode decision also introduces high computation. With the information collected from the input bitstream, some modes can be skipped and mode decision can be early stopped.

Error-related transcoding has also attracted many interests. In Chapter 6, intra refresh transcoding has been studied and an efficient scheme was proposed to adaptively insert intra MB's to generate a more robust bitstream. It is well known that, there are still some other error resilient tools, e.g. resynchronization marker insertion, reference frame selection, and so on. It is expected to incorporate these resilient tools together and present a global optimal solution in coding process.

Author's Publications

Journal papers

1. H. Shu and L.-P. Chau, "A Resizing Algorithm with Two-stage Realization for Video Transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, pp. 248 – 253, February 2007.

2. H. Shu and L.-P. Chau, "The Realization of Arbitrary Downsizing Video Transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 540 – 546, April 2006.

3. H. Shu and L.-P. Chau, "An Efficient Arbitrary Downsizing Transcoding Algorithm for Video Transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 887 – 891, June 2004.

4. H. Shu and L.-P. Chau, "Intra/inter Macroblock Mode Decision for Error-Resilient Transcoding," Revised version submitted to *IEEE Trans. Multimedia.*

5. H. Shu and L.-P. Chau, "Dynamic Frame Skipping Transcoding with Motion Information Considering," Revised version submitted to *IEE Proc. Vision, Image & Signal Processing.*

Conference papers

6. H. Shu and L.-P. Chau, "Generalized Arbitrary Resizing for Video Transcoding," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'06)*, pp. 5271–5274, 2006.

7. H. Shu and L.-P. Chau, "Frame Size Selection in Video Downsizing Transcoding Application," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'05)*, vol. 3, pp. 896–899, 2005.

8. H. Shu and L.-P. Chau, "Variable Frame Rate Transcoding Considering Motion Information," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'05)*, vol. 3, pp. 2144–2147, 2005.

9. H. Shu and L.-P. Chau, "Frame Layer Bit Allocation for Video Transcoding," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'05)*, vol. 3, pp. 4357–4360, 2005.

10. H. Shu and L.-P. Chau, "A New Scene Change Feature for Video Transcoding," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'05)*, vol. 3, pp. 4582–4585, 2005.

11. H. Shu and L.-P. Chau, "Frame-Skipping Transcoding with Motion Change Consideration," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'04)*, vol. 3, pp. 773–776, 2004.

12. H. Shu and L.-P. Chau,"A Fast Arbitrary Downsizing Algorithm for Video Transcoding," *Proc. IEEE Int. Conf. Image Processing (ICIP'03)*, vol. 1, pp. 201–204, 2003.

Bibliography

[1] S.-F. Chang and A. Vetro, "Video Adaptation: Concepts, Technologies, and Open Issues," *Proc. IEEE*, vol. 93, no. 1, pp. 148–158, Jan. 2005.

[2] M. Ghanbari, *Video Coding: An Introduction to Standard Codecs*. The Institution of Electrical Engineers, 1999.

[3] A. H. Sadka, *Compressed Video Communications*. John Wiley & Sons Ltd, 2002.

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

[5] "Video Codec for Audiovisual Services at $P \times 64kbit/s$," *ITU-T Recommandation H.261*, 1993.

[6] "Video Coding for Low Bit Rate Communication," *ITU-T Recommandation H.263*, June 2001.

[7] "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1.5Mbit/s - Part 2: Video," *ISO/IEC 11172*, 1993.

[8] "Information Technology : Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video," *ISO/IEC 13818*, 1995.

[9] "Information Technology - Coding of Audio-Visual Objects - Part 2: Visual," *ISO/IEC 14496*, 1998.

[10] "Advanced Video Coding for Generic Audiovisual Services," *ITU-T RECOMMENDATION H.264*, May 2003.

[11] A. Vetro, C. Christopoulos, and H. Sun, "Video Transcoding Architectures and Techniques: An Overview," *IEEE Signal Processing Mag.*, vol. 20, no. 2, pp. 18–29, Mar. 2003.

[12] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital Video Transcoding," *Proc. IEEE*, vol. 93, no. 1, pp. 84–97, Jan. 2005.

[13] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video Transcoding: An Overview of Various Techniques and Research Issues," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793 – 804, Oct. 2005.

[14] B. Shen, I. K. Sethi, and B. Vasudev, "Adaptive Motion-Vector Resampling for Compressed Video Downscaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 929–936, Sept. 1999.

[15] P. A. A. Assuncao and M. Ghanbari, "Transcoding of Single-Layer MPEG Video into Lower Rates," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 144, no. 6, pp. 377–383, Dec. 1997.

[16] J.-N. Hwang and T.-D. Wu, "Motion Vector Re-estimation and Dynamic Frame-Skipping for Video Transcoding," *Conference Record of the Thirty-Second Asilomar on Signals, Systems and Computers*, vol. 12, pp. 1606–1610, 1998.

[17] K.-D. Seo and J.-K. Kim, "Motion Vector Refinement for Video Downsampling in the DCT Domain," *IEEE Signal Processing Lett.*, vol. 9, no. 11, pp. 356–359, Nov. 2002.

[18] Y. Liang, L.-P. Chau, and Y.-P. Tan, "Arbitrary Downsizing Video Transcoding Using Fast Motion Vector Re-estimation," *IEEE Signal Processing Lett.*, vol. 9, no. 11, pp. 352–355, Nov. 2002.

[19] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion Estimation for High Performance Transcoding," *IEEE Trans Consumer Electron.*, vol. 44, no. 3, pp. 649–658, Aug. 1998.

[20] ——, "Motion Vector Refinement for High-Performance Transcoding," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 30–40, Mar. 1999.

[21] H. Sun, Y.-P. Tan, and Y. Liang, "Fast Motion Vector and Bitrate Re-estimation for Arbitrary Downsizing Video Transcoding," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, pp. 856–859, 2003.

[22] Y.-P. Tan, Y. Liang, and H. Sun, "On the Methods and Performances of Rational Downsizing Video Transcoding," *Signal Processing: Image Communication*, no. 19, pp. 47–65, 2004.

[23] K.-D. Seo and J. k. Kim, "Fast Motion Vector Re-estimation for Transcoding MPEG-1 into MPEG-4 with Lower Spatial Resolution in DCT-domain," *Signal Processing: Image Communication.*, vol. 19, pp. 299–312, 2004.

[24] T. Shanableh and M. Ghanbari, "Heterogeneous Video Transcoding to Lower Spatio-Temporal Resolutions and Different Encoding Formats," *IEEE Trans. Multimedia*, vol. 2, no. 2, pp. 101–110, Jun. 2000.

[25] M.-J. Chen, M.-C. Chu, and C.-W. Pan, "Efficient Motion-Estimation Algorithm for Reduced Frame-Rate Video Transcoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 269–275, Apr. 2002.

[26] W.-J. Lee and W.-J. Ho, "Adaptive Frame-Skipping for Video Transcoding," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 165–168, 2003.

[27] M. Kucukgoz and M.-T. Sun, "Early-Stop and Motion Vector Re-Using for MPEG-2 to H.264 Transcoding," *Proc. SPIE Visual Communications and Image Processing*, vol. 5308, pp. 932–936, 2004.

[28] J. Youn and M.-T. Sun, "Video Transcoding with H.263 Bit-Streams," *Journal of Visual Communication and Image Representation*, vol. 11, pp. 385–403, Dec. 2000.

[29] P. A. A. Assuncao and M. Ghanbari, "A Frequency-Domain Video Transcoder for Dynamic Bit-Rate Reduction of MPEG-2 Bit Streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953–967, Dec. 1998.

[30] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG Bitstreams," *Signal Processing: Image Communication.*, vol. 8, pp. 481–500, 1996.

[31] N. Bjork and C. Christopoulos, "Transcoder Architectures for Video Coding," *IEEE Trans Consumer Electron.*, vol. 44, no. 1, pp. 88–98, Feb. 1998.

[32] S. F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," *IEEE J. Select. Areas Commun.*, vol. 13, no. 1, pp. 1–11, Jan. 1995.

[33] J. Wang and S. Yu, "Dynamic Rate Scaling of Coded Digital Video for IVOD Application," *IEEE Trans Consumer Electron.*, vol. 44, no. 3, pp. 743–749, Aug. 1998.

[34] T. Shanableh and M. Ghanbari, "Transcoding Architectures for DCT-domain Heterogeneous Video Transcoding," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 433–436, 2001.

[35] N. Merhav and V. Bhaskaran, "Fast Algorithms for DCT-domain Image Downsampling and for Inverse Motion Compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 468–476, June 1997.

[36] J. Song and B. L. Yeo, "A Fast Algorithm for DCT-Domain Inverse Motion Compensation Based on Shared Information in a Macroblock," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 5, pp. 767–775, Aug. 2000.

[37] S. Liu and A. C. Bovik, "Local Bandwidth Constrained Fast Inverse Motion Compensation for DCT-Domain Video Transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 309–319, May 2002.

[38] K.-T. Fung, Y.-L. Chan, and W.-C. Siu, "New Architecture for Dynamic Frame-Skipping Transcoder," *IEEE Trans. Image Processing*, vol. 11, no. 8, pp. 886–900, Aug. 2002.

[39] ——, "Low-Complexity and High-Quality Frame-Skipping Transcoder for Continuous Presence Multipoint Video Conferencing," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 31–46, Feb. 2004.

[40] P. Yin, A. Vetro, B. Liu, and H. Sun, "Drift Compensation for Reduced Spatial Resolution Transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 1009–1020, Nov. 2002.

[41] R. Mocry and D. Anastassiou, "Minimal Error Drift in Frequency Scalability for Motion-Compensated DCT Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 392–406, Aug. 1994.

[42] B. Shen, "Submacroblock Motion Compensation for Fast Down-Scale Transcoding of Compressed Video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 1291–1302, Oct. 2005.

[43] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, March 2001.

[44] W. Li and Y. Chen, "Experiment Result on Fine Granularity Scalability," *ISO/IEC JTC1/SC29/WG11, MPEG99/M4473,*, Mar. 1999.

[45] U. Benzler and U. Pestel-Schiller, "Result of Core Experiment on Fine Granularity Scalability for Video (Part2: MC with Drift)," *ISO/IEC JTC1/SC29/WG11, MPEG99/M4847,*, Jul. 1999.

[46] O. Werner, "Requantization for Transcoding of MPEG-2 Intraframes," *IEEE Trans. Image Processing*, vol. 8, no. 2, pp. 179–191, Feb. 1999.

[47] P. A. A. Assuncao and M. Ghanbari, "Optimal Bit Rate Conversion of MPEG-2 Video Bit Streams," *Electronics Letters*, vol. 33, no. 8, pp. 675–677, Apr. 1997.

[48] Z. Lei and N. D. Georganas, "Rate Adaptation Transcoding for Precoded Video Streams," *Proceedings of 10th ACM International Conference on Multimedia*, pp. 127–136, 2002.

[49] J.-N. Hwang, T.-D. Wu, and C.-W. Lin, "Dynamic Frame-Skipping in Video Transcoding," *IEEE Second Workshop on Multimedia Signal Processing.*, pp. 616–621, 1998.

[50] S. Liu and C.-C. J. Kuo, "Complexity Reduction of Joint Temporal-Spatial Bit Allocation Using R-D Models for Video Streaming," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 729–732, 2002.

[51] F. C. M. Martins, W. Ding, and E. Feig, "Joint Control of Spatial Quantization and Temporal Sampling for Very Low Bit Rate Video," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2072–2075, 1996.

[52] H. Shu and L.-P. Chau, "Frame-Skipping Transcoding with Motion Change Consideration," *Proc. IEEE Int. Symp. Circuits and Systems.*, vol. 3, pp. 773–776, 2004.

[53] F. Pan, Z. P. Lin, X. Lin, S. Rahardja, W. Juwono, and F. Slamer, "Content Adaptive Frame Skipping for Low Bit Rate Video Coding," *Proc. IEEE Int. Conf. ICICS-PCM*, vol. 1, pp. 230–234, 2003.

[54] H. Shu and L.-P. Chau, "An Efficient Arbitrary Downsizing Transcoding Algorithm for Video Transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 887 – 891, June 2004.

[55] G. Shen, B. Zeng, Y.-Q. Zhang, and M. L. Liou, "Transcoder with Arbitrarily Resizing Capability," *Proc. IEEE Int. Symp. Circuits and Systems.*, vol. 5, pp. 25–28, 2001.

[56] R. Dugad and N. Ahuja, "A Fast Scheme for Image Size Change in the Compressed Domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 461–474, Apr. 2001.

[57] J. Mukherjee and S. K. Mitra, "Image Resizing in the Compressed Domain Using Subband DCT," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 620–627, July 2002.

[58] W. Zhu, K. H. Yang, and M. J. Beacken, "CIF-to-QCIF Video Bitstream Down-Conversion in the DCT Domain," *Bell Labs Technical Journal*, pp. 21–29, Jul.-Sept. 1998.

[59] B. Shen and S. Roy, "A very Fast Video Spatial Resolution Reduction Transcoder," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 2, pp. 1989–1992, 2002.

[60] Q. Hu and S. Panchnathan, "Image/Video Spatial Scalability in Compressed Domain," *IEEE Trans. Ind. Electron.*, vol. 45, no. 1, pp. 23–31, Feb 1998.

[61] Y. S. Park and H. W. Park, "Design and Analysis of an Image Resizing Filter in the Block-DCT Domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 274–279, Feb. 2004.

[62] Y.-R. Lee, C.-W. Lin, S.-H. Yeh, and Y.-C. Chen, "Low-Complexity DCT-Domain Video Transcoders for Arbitrary-Size Downscaling," *Proc. IEEE 6th Workshop on Multimedia Signal Processing*, pp. 31–34, 2004.

[63] D. Mehta and U. B. Desai, "A Primer To Video Transcoding: Image Transcoding," *Proc. Eighth National Conference on Communications*, pp. 528–532, Jan. 2002.

[64] H. W. Park, Y. S. Park, and S.-K. Oh, "L/M-Fold Image Resizing in Block-DCT Domain Using Symmetric Convolution," *IEEE Trans. Image Processing*, vol. 12, no. 9, pp. 1016–1034, Sept. 2003.

[65] C. L. Salazar and T. D. Tran, "On Resizing Images in the DCT Domain," *Proc. IEEE Int. Conf. Image Processing*, vol. 4, pp. 2797 – 2800, 2004.

[66] J. Xin, M.-T. Sun, B.-S. Choi, and K.-W. Chun, "An HDTV-to-SDTV Spatial Transcoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 998–1008, Nov. 2002.

[67] S. Dogan, A. H. Sadka, and A. M. Kondoz, "Efficient MPEG-4/H/263 Video Transcoder for Interoperability of Heterogeneous Multimedia Networks," *Electronics Letters*, vol. 35, no. 11, pp. 863–864, May 1999.

[68] K.-D. Seo, S.-C. Heob, and J.-K. Kim, "Adaptive Rate Control Algorithm Based on Logarithmic RCQ Model for MPEG-1 to MPEG-4 Transcoding," *Signal Processing: Image Communication.*, vol. 17, pp. 857–875, 2002.

[69] N. Feamster and S. Wee, "An MPEG-2 to H.263 Transcoder," *SPIE International Symposium on Voice, Video, and Data Communications*, Sept. 1999.

[70] T. Shanableh and M. Ghanbari, "Transcoding of Video into Different Encoding Formats," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing.*, vol. 6, pp. 1927–1930, 2000.

[71] H. Radha, M. van der Schaar, and S. Karande, "Scalable Video Transcaling for the Wireless Internet," *EURASIP Journal on Applied Signal Processing- Special Issue on Multimedia over IP and Wireless Networks*, no. 2, pp. 265–279, Feb. 2004.

[72] H. Radha, "TranScaling: A Video Coding and Multicasting Framework for Wireless IP Multimedia Services," *Proceedings of the 4th ACM international workshop on Wireless mobile multimedia.*, pp. 12–22, 2001.

[73] Y.-C. Lin, C.-N. Wang, T. Chiang, A. Vetro, and H. Sun, "Efficient FGS to Single Layer Transcoding," *Int. Conf. Consumer Electronics, Digest of Technical Papers.*, pp. 134–135, 2002.

[74] Y. Liang and Y.-P. Tan, "Methods and Needs for Transcoding MPEG-4 Fine Granularity Scalability Video," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, pp. 719–722, 2002.

[75] E. Barrau, "MPEG Video Transcoding to a Fine-Granular Scalable Format," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 717–720, 2002.

[76] T. Shanableh and M. Ghanbari, "Multilayer Transcoding With Format Portability for Multicasting of Single-Layer Video," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 1–15, Feb. 2005.

[77] G. Cote, S. Shirani, and F. Kossentini, "Optimal Mode Selection and Synchronization for Robust Video Communications Over Error-prone Networks," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 952–965, June 2000.

[78] G. Cote and F. Kossentini, "Optimal Intra Coding of Blocks for Robust Video Communication Over the Internet," *Signal Processing: Image Communication*, vol. 15, pp. 25–34, 1999.

[79] W.-H. J. Chen and J.-N. Hwang, "The CBERC: A Content-based Error-Resilient Coding Technique for Packet Video Communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 8, pp. 974–980, Aug. 2001.

[80] R. Zhang, S. L. Regunathan, and K. Rose, "Video Coding with Optimal Inter/Intra-Mode Switching for Packet Loss Resilience," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 966–976, June 2000.

[81] Z. He, J. Cai, and C. W. Chen, "Joint Source Channel Rate-Distortion Analysis for Adaptive Mode Selection and Rate Control in Wireless Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 511–523, June 2002.

[82] H.-J. Chiou, Y.-R. Lee, and C.-W. Lin, "Error-Resilient Transcoding Using Adaptive Intra Refresh for Video Streaming," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 777–780, 2004.

[83] S. Dogan, A. Cellatoglu, M. Uyguroglu, A. H. Sadka, and A. M. Kondoz, "Error-Resilient Video Transcoding for Robust Internetwork Communications Using GPRS," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 453–464, June 2002.

[84] M. Xia, A. Vetro, H. Sun, and B. Liu, "Rate-Distortion optimized Bit Allocation for Error Resilient Video Transcoding," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 945–948, 2004.

[85] G. Reyes, A. R. Reibman, S.-F. Chang, and J. C.-I. Chuang, "Error-Resilient Transcoding for Video over Wireless Channels," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1063–1074, June 2000.

[86] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error Resilient Video Coding Techniques," *IEEE Signal Processing Mag.*, vol. 17, no. 4, pp. 61– 82, Jul. 2000.

[87] T.-W. A. Lee, S.-H. G. Chan, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Allocation of Layer Bandwidths and FECs for Video Multicast Over Wired and Wireless Networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1059 – 1070, Dec. 2002.

[88] J. Meng and S.-F. Chang;, "Embedding Visible Video Watermarks in the Compressed Domain," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 474 – 477, 1998.

[89] J. Youn, J. Xin, and M.-T. Sun, "Fast Video Transcoding Architectures for Networked Multimedia Applications," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, pp. 25 – 28, 2000.

[90] N. Roma and L. Sousa, "Fast Transcoding Architectures for Insertion of Non-regular Shaped Objects in the Compressed DCT-domain," *Signal Processing: Image Communication.*, vol. 18, pp. 659–683, 2003.

[91] C.-W. Lin, T.-J. Liou, and Y.-C. Chen, "Dynamic Rate Control in Multipoint Video Transcoding," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, pp. 17–20, 2000.

[92] M.-T. Sun, T.-D. Wu, and J.-N. Hwang, "Dynamic Bit Allocation in Video Combining for Multipoint Conferencing," *IEEE Trans. Circuits Syst. II*, vol. 45, no. 5, pp. 644–648, May 1998.

[93] C.-W. Lin, Y.-C. Chen, and M.-T. Sun, "Dynamic Region of Interest Transcoding for Multipoint Video Conferencing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 10, pp. 982 – 992, Oct. 2003.

[94] Q.-F. Zhu, L. Kerofsky, and M. B. Garrison, "Low-delay, Low-complexity Rate Reduction and Continuous Presence for Multipoint Videoconferencing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 666–676, Jun. 1999.

[95] C.-H. Li, H. Lin, C.-N. Wang, and T. Chiang, "A Fast H.264-based Picture-in-Picture (PIP) Transcoder," *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 3, pp. 1691–1694, 2004.

[96] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice Hall, 1993.

[97] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, 2001.

[98] C. W. Kok, "Fast Algorithm for Computing Discrete Cosine Transform," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 757 – 760, Mar. 1997.

[99] G. Bi and L. W. Yu, "DCT Algorithms for Composite Sequence Lengths," *IEEE Trans. Signal Processing*, vol. 46, no. 3, pp. 554 – 562, Mar. 1998.

[100] A. Vetro, H. Sun, P. DaGraca, and T. Poon, "Minimum Drift Architectures for 3-layer Scalable DTV Decoding," *IEEE Trans. Consumer Electron.*, vol. 44, no. 3, pp. 527–536, Aug. 1998.

[101] Y.-P. Tan and H. Sun, "Fast Motion Re-Estimation for Arbitrary Downsizing Video Transcoding Using H.264/AVC Standard," *IEEE Trans. Consumer Electron.*, vol. 50, no. 3, pp. 887–894, Aug. 2004.

[102] A. M. Bruckstein, M. Elad, and R. Kimmel, "Down-Scaling for Better Transform Compression," *IEEE Trans. Image Processing*, vol. 12, no. 9, pp. 1132 – 1144, Sept. 2003.

[103] J. Ribas-Corbera and D. L. Neuhoff, "Optimizing Block Size In Motion-Compensated Video Coding," *Journal of Electronic Imaging.*, vol. 7, no. 1, pp. 155–165, Jan. 1998.

[104] Z. He and S. K. Mitra, "A Unified Rate-Distortion Analysis Framework for Transform Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1221–1236, Dec. 2001.

[105] ——, "A Linear Source Model and a Unified Rate Control Algorithm for DCT Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 970–982, Nov. 2002.

[106] J. Ribas-Corbera and S. Lei, "Rate Control in DCT Video Coding for Low-Delay Communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 172–185, Feb. 1999.

[107] J.-K. Han, S.-M. Kwak, and J. Kim, "Joint Optimization of the Motion Estimation Module and the Up/Down Scaler in Transcoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1303–1313, Oct. 2005.

[108] J.-J. Chen and H.-M. Hang, "Source Model for Transform Video Coder and Its Application-Part II: Variable Frame Rate Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 299–311, Apr. 1997.

[109] J.-W. Lee, A. Vetro, Y. Wang, and Y.-S. Ho, "Bit Allocation for MPEG-4 Video Coding With Spatio-Temporal Tradeoffs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 488–502, June 2003.

[110] S. Liu and C.-C. J. Kuo, "Joint Temporal-Spatial Bit Allocation for Video Coding With Dependency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 15–26, Jan. 2005.

[111] H. Song and C.-C. J. Kuo, "Rate Control for Low-Bit-Rate Video via Variable-Encoding Frame Rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 512–521, Apr. 2001.

[112] Y. Hong and Y.-S. Choe, "Bit Allocation Method for Variable Frame Rated Low Bit Coding Scheme," *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 195–199, 2001.

[113] B. C. Song and K. W. Chun, "A Virtual Frame Rate Control Algorithm for Efficient MPEG-2 Video Encoding," *IEEE Trans Consumer Electron.*, vol. 49, no. 2, pp. 460–465, May 2003.

[114] "Video Coding for Low Bit Rate Communication, version 2," *ITU-T Recommandation H.263*, 1998.

[115] S. Park, Y. Lee, and H. Chang, "A New MPEG-2 Rate Control Scheme Using Scene Change Detection," *ETRI Journal*, vol. 18, no. 2, pp. 61–74, Jul. 1996.

[116] Y.-S. Saw, *Rate-Quality Optimized Video Coding*. Kluwer Academic Publishers, 1999.

[117] B. T. Truong, S. Venkatesh, and C. Dorai, "Scene Extraction in Motion Pictures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 5–15, Jan. 2003.

[118] C.-L. Huang and B.-Y. Liao, "A Robust Scene-Change Detection Method for Video Segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1281–1288, Dec. 2001.

[119] W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull, "A Unified Approach to Scene Change Detection in Uncompressed and Compressed Video," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 769–779, Aug. 2000.

[120] B.-L. Yeo and B. Liu, "A Unified Approach to Temporal Segmentation of Motion JPEG and MPEG Compressed Video," *International Conference on Multimedia Computing and Systems*, pp. 81–88, 1995.

[121] ——, "Rapid Scene Analysis on Compressed Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, pp. 533–544, Dec. 1995.

[122] E. K. Kang, S. J. Kim, and J. S. Choi, "Video Retrieval Based on Scene Change Detection in Compressed Streams," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 3, pp. 932–936, Aug. 1999.

[123] J.-R. Kim, S. Suh, and S. Sull, "Fast Scene Change Detection for Personal Video Recorder," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 3, pp. 683–688, Aug. 2003.

[124] K. Zhang and J. Kittler, "Using Scene-Change Detection and Multiple-Thread Background Memory for Efficient Video Coding," *Electronics Letters*, vol. 35, no. 4, pp. 290–291, Feb. 1999.

[125] W. A. C. Fernando, C. N. Canagarajah, and D. Bull, "Scene Change Detection Algorithms for Content-Based Video Indexing and Retrieval," *Electronics & Communication Engineering Journal.*, pp. 117–126, June 2001.

[126] A. Akutsu, Y. Tonomura, H. Hashimoto, and Y. Ohba, "Video Indexing Using Motion Vectors," *Proc. SPIE: Visual Communication and Image Processing*, vol. 1818, pp. 1522–1530, 1992.

[127] T.-Y. Kuo and C.-C. J. Kuo, "Motion-Compensated Interpolation for Low-Bit-Rate Video Quality Enhancement," *Proc. SPIE Conference on Applications of Digital Image Processing*, vol. 3460, pp. 277–288, July 1998.

[128] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y.-Q. Zhang, and H. J. Chao, "An End-to-End Approach for Optimal Mode Selection in Internet Video Communication: Theory and Application," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 977–995, June 2000.

[129] A. Leontaris and P. Cosman, "Video Compression with Intra/Inter Mode Switching and a Dual Frame Buffer," *Proc. IEEE Data Compression Conference*, pp. 63–72, 2003.

[130] G. J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *IEEE Signal Processing Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[131] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, "Concealment of Whole-Frame Losses for Wireless Low Bit-Rate Video Based on Multiframe Optical Flow Estimation," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 316–329, Apr. 2005.

[132] P. Baccichet, D. Bagni, A. Chimienti, L. Pezzoni, and F. S. Rovati, "Frame Concealment for H.264/AVC Decoders," *IEEE Trans. Consumer Electron.*, vol. 51, no. 1, pp. 227–233, Feb. 2005.