Waseda University Doctoral Dissertation

Low-complexity SVC/AVC Transcoder based on Data Exploitation and Approximation for Videoconferencing

Lei SUN

Graduate School of Information, Production and Systems

Waseda University

June 2013

Acknowledgements

Firstly I would dedicate this dissertation to my beloved families for their support during my PhD period, especially my wife and my one-and-a-half years old son. It's them who gave me the courage to challenge and finish the PhD course.

And I would like to show my great thanks to my supervisor, Professor Takeshi Ikenaga, who helped me a lot in finishing my PhD research and this dissertation. He always provides a lot of useful comments and suggestions for my research, and encourages me to step forward. Without his help, I could not have obtained current achievements and finished this dissertation.

I also owe my thanks to the professors that guide me in improving and finishing this dissertation by providing many useful and remarkable advices, including Professor Seiichi Gohshi, Professor Shinji Kimura, and Professor Takeshi Yoshimura. They also attended my public hearing and provided many helpful comments for the presentation.

Besides, I got many helps from other lab members in both research and daily life. I would like to show great gratitude to them. Professor Zhenyu Liu, Dr. Qin Liu, Dr. Yiqing Huang and Dr. Jia Su were always ready to discuss with me and provide valuable advices. Without their help, I could possibly not finish my PhD course within three years. I am grateful to other former members including Mr. Takahiro Sakayori, Mr. Tianci Huang, Mr. Jingbang Qiu, Mr. Zhewen Zheng, Mr. Shuijiong Wu, Mr. Kodai Kawane, Mr. Tuyoshi Sasaki, Mr. Jin Zhou, Mr. Bingrong Wang, Ms. Chengjiao Guo, Mr. Xiaocong Jin, Ms. Ying Lu, Mr. Jie Leng, Mr. Yoshimasa Iwasaki, Mr. Bin Li and Mr. Wei-jing Chen, for their accompanies. We had a memorizable time at

Kitakyushu. I would also show my thanks to the current lab members including Mr. Xiaolei Yu, Mr. Xiaoyang Yuan, Mr. Lei Gu, Mr. Gaoxing Chen, Mr. Kenta Okuyama, Mr. Yuanche Sun and Mr. Zhenyu Pei. Last but not least, my thanks would go to tokyo memers including Mr. Ryosuke Araki, Mr. Yuhi Shiina, Mr. Yoshiyasu Shimizu, Mr. Hiroyuki Sekiguchi, Mr. Yuichiro Kitao, Mr. Takahiro Suzuki, and Mr. Youhei Mikami. It was a wonderful experience to discuss with them at joint seminars and travel around together.

In addition, I would like to deliver my gratitude to all other friends whose names are not listed here. They supported me in life and study in different ways, which were all so important for me.

At last, I would like to acknowledge the financial support from KDDI Foundation, Global COE project and CREST project.

Abstract

Videoconferencing is an important communication tool in nowadays for companies and other organizations. It allows real-time exchange of video and sound between participants at different locations. Video compression is a key component of videoconferencing applications. In conventional videoconferencing, the widely used H.264/AVC (Advanced Video Coding) standard is adopted to compress the transmitted video. In 2007, Scalable Video Coding (SVC) was standardized as an extension of AVC. SVC enables transmission of a single bit-stream containing multiple subset bit-streams with different spatial resolution, temporal frame rate or quality. The subset bit-streams are organized in layered structure efficiently and can be extracted adaptively according to the receiving terminals with different resolution, performance or network conditions. SVC is expected to be the next-generation video compression technology for videoconferencing.

Although many videoconferencing manufactures have been releasing SVC based products, there still exist many legacy AVC based systems. Due to the different formats, SVC based system cannot communicate with AVC based system. To enable the communication between SVC and AVC based systems, transcoding between SVC and AVC formats is needed. A straightforward solution is to cascade a decoder and an encoder, also named"re-encoding method". It fully decodes the input bit-stream and then re-encodes the decoded pictures, which is easy to implement but consumes intensive computations. For videoconferencing applications which is time-critical, much lower transcoding complexity is desired.

This dissertation is focused on low-complexity SVC/AVC transcoding. To enable two-way communication between SVC and AVC systems, this dissertation targets both SVC to AVC and AVC to SVC transcoding. Targeted scalability includes spatial and quality. Temporal scalability is not used in videoconferencing because backward prediction will cause delay. Besides, SVC quality scalability uses fixed hierarchical coding structure with low coding efficiency. Depending on whether AVC uses same coding structure or not, there are two transcoding approaches for SVC to AVC quality transcoding - homogeneous and heterogeneous transcoding.

The methodologies used in this dissertation include data exploitation and data approximation. Data exploitation means to skip unnecessary processing by utilizing the input data such as mode, motion vector (MV) and residue. Data approximation means to remove computationally heavy components and approximate them with less computation.

The dissertation contents are organized as follows.

In Chapter 1, background of this dissertation is described, including introduction of the videoconferencing, the AVC and SVC coding standards, the necessity and existing works of SVC/AVC transcoding. Furthermore, the target and organization of this dissertation are shown.

In Chapter 2, an AVC to SVC transcoder with spatial scalability is proposed based on coarse-level mode-mapping. The input mode and motion vector are utilized to reduce the SVC encoder complexity. Conventional methods use deterministic mode mapping, i.e., the SVC coding mode is directly decided from the input AVC data. In proposed transcoder, the coding modes of macroblocks within a search range around the co-located area in AVC frame are firstly checked. Only the coding modes that appear in the search range are estimated. Then partition mapping schemes are utilized by allowing more than one candidate. The candidates are estimated and the best one is chosen. Partition mapping schemes are applied adaptively according to the smoothness of the texture in the search range. Simulation results show that proposed

transcoder obtains averagely 82.7% time saving for videoconferencing-alike sequences comparing with the re-encoding method, which is about 2.1 times faster than the representative R. Sachdeva's work [ICIS, 2009]. Besides, the quality loss at same bit-rate for proposed method is only 0.11 dB averagely, while R. Sachdeva's work has about 1 dB quality loss.

In Chapter 3, an SVC to AVC transcoder with spatial scalability is proposed based on hybrid-domain transcoding. Conventional methods are either based on pure pixel- or frequency-domain. Pixel-domain transcoding has better coding efficiency but less speed than frequency-domain transcoding. In proposed transcoder, macroblocks in input SVC frame are divided into two groups - AVC compatible and incompatible macroblocks. AVC-compatible macroblocks are transcoded in frequency domain and incompatible ones are transcoded in pixel domain. The hybrid-domain transcoding results in a drift problem, which is the accumulation of distortion. To solve this problem, a rate-distortion optimization metric emphasizing the importance of prediction pixels is used for I frame, resulting in improved image quality. For following P frames, accumulated distortion is calculated and compensated back to the input signal, thus relieving the drift. Simulation results show that proposed transcoder obtains averagely 96.4% time saving comparing with the re-encoding method, which is 2.1 times faster than the representative H. Liu's work [ICME, 2009]. Besides, the quality loss at same bit-rate for proposed method is 0.1-0.5 dB less than H. Liu's work.

In Chapter 4, an SVC to AVC homogeneous transcoder with quality scalability is proposed based on quantization-domain motion compensation and intra prediction. P. Assuncao's work [ICASSP, 1996] proposed a single loop transcoding architecture which is widely used in later works. In proposed transcoder, transform operations are totally removed and quantized coefficients are used directly, resulting in greatly reduced complexity. For motion compensation, the predictor is approximated by the weighted sum of overlapped macroblocks. The weight is proportional to the overlapped area. For intra prediction, the predictor is approximated

by the weighted sum of neighboring macroblocks. The weight is decided by how many pixels of the macroblock are used for intra prediction. Simulation results show that proposed transcoder obtains averagely 98.1% time saving comparing with re-encoding method, which is 6.5 times faster than the implementation based on single loop. The quality loss at same bit-rate is averagely 0.71 dB.

In Chapter 5, an SVC to AVC heterogeneous transcoder with quality scalability is proposed based on paired mode mapping and MV estimation & refinement. Conventional methods use homogeneous input/output coding structure as shown in Chapter 4 with great time saving. However, the coding efficiency drops a lot. This chapter proposes a heterogeneous transcoder to improve the coding efficiency. Output AVC stream is encoded by IPPP coding structure with multiple reference frames. To reduce the dramatically increased complexity, paired mode mapping method is utilized for the corresponding frame. Not only the input SVC mode but also the most similar mode is examined. The most similar mode pairs are defined according to the intra mode directions. For the non-corresponding reference frame, the MV is derived based on estimation and refinement. Firstly the MV is estimated by conjunction of other known MVs. Then the estimated MV is refined within the neighboring blocks around the reference block. Simulation results show that proposed transcoder obtains averagely 94.3% time saving comparing with the re-encoding method. The quality loss at same bit-rate for proposed transcoder is only 0.048 dB, while homogeneous transcoding has a quality loss of 0.33 dB.

In Chapter 6, the overall dissertation is summarized and the future work is described. With the intention to enable communication between SVC and AVC videoconferencing systems, four works on low-complexity SVC/AVC transcoding for spatial and quality scalability are presented in this dissertation. 82.7%-98.1% time saving is obtained comparing with the re-encoding method. The works in the dissertation is expected to play an important role in a hybrid videoconferencing application.

Contents

1 Introduction **1**

1.1 Videoconferencing 1
1.2 AVC and SVC 2
1.3 Transcoding 3
1.4 Problem & Target 4
1.5 Dissertation Organization 5

2 Coarse level mode mapping based AVC to SVC spatial transcoding **7**

2.1 Introduction 7
2.2 Reference model analysis 8
2.3 Overall transcoding architecture 9
2.4 Proposed lower-layer transcoding schemes 11
2.4.1 ME skipping scheme 11
2.4.2 Probability-profile based mode decision control 14
2.4.3 Coarse-level mode-mapping methods 15
2.4.4 MV refinement scheme 18
2.5 Proposed top-layer transcoding schemes 19
2.5.1 Direct encapsulation 19
2.5.2 Inter-layer prediction utilization 20
2.6 Experimental results 20
2.7 Conclusions 22

3 Drift compensated hybrid-domain SVC to AVC spatial transcoding **25**

3.1 Introduction 25
3.2 Scalable Video Coding 27

3.2.1 Inter-layer Predictions 27
3.2.2 Coding modes in SVC 28
3.3 Hybrid-domain SVC-to-AVC Transcoding 29
3.3.1 Hybrid-domain transcoding 29
3.3.2 Pixel-domain transcoding 31
3.3.3 Quantized transform-domain transcoding 33
3.4 Drift Compensation 34
3.4.1 Drift Analysis 34
3.4.2 Drift Compensation in I frame 36
3.4.3 Drift Compensation in P frame 38
3.5 Overall Transcoding Architecture 39
3.6 Simulation Results 39
3.7 Conclusions 43

4 Drift constrained frequency-domain SVC to AVC quality homogeneous transcoding 47
4.1 Introduction 47
4.2 MGS scalability in SVC 50
4.3 Overall proposed transcoding method 51
4.3.1 Analysis of coding modes in SVC 51
4.3.2 Proposed transcoding method 53
4.4 Non-KEY picture transcoding 53
4.4.1 Quantization-domain single-loop transcoding for IL_Residual MBs 53
4.4.2 Quantization-domain intra prediction for IL_Intra MBs 55
4.4.3 Quantization-domain copy for other MBs 58
4.5 KEY picture transcoding 59
4.6 Simulation results 60
4.7 Conclusions 63

5 Mode mapping and MV conjunction based SVC to AVC quality heterogenous transcoding 67
5.1 Introduction 67
5.2 Proposed 3-stage transcoder 68

5.2.1 SVC to AVC mode mapping 69
5.2.2 Optimized MV conjunction 71
5.2.3 Hadamard-based early termination 73
5.2.4 Overall scheme 74
5.3 Simulation results 76
5.4 Conclusions 78

6 Conclusions and Future Works **81**

Publications **91**

List of Figures

1.1 Timeline for video coding standards 2
1.2 A hybrid video conferencing scenario 4

2.1 Intuitive mode & partition comparison for akiyo sequence 11
2.2 Proposed transcoder 13
2.3 Search range and search order in VGA sequence. 14
2.4 INTER (non-SKIP) mode profile for akiyo sequence, frame 37. 15
2.5 Direct mapping method. 16
2.6 Candidate mapping method. 17
2.7 Priority mapping method. 17
2.8 R-D curves comparison for top layer. 23

3.1 SVC coding structure with spatial scalability. 27
3.2 Different transcoding domains. 29
3.3 Hybrid-domain transcoding 30
3.4 Transcoding in pixel domain and quantized transform domain. 31
3.5 Accuracy ratio for mode mapping 32
3.6 MV refinement. 33
3.7 Pixel-domain transcoding. 33
3.8 Quantized transform-domain transcoding. 34
3.9 Drift problem in proposed transcoder. 36
3.10 Prediction pixels. 37
3.11 Error Compensation. 37
3.12 Overall transcoding architecture. 39
3.13 Subjective comparisons. (akiyo sequence, DC: drift compensation) . . 43

3.14 Subjective comparisons. (bus sequence, DC: drift compensation) . . . 44
3.15 RD curves comparison. 45

4.1 Motion reuse (MR) transcoding architecture 48
4.2 Single-loop (SL) transcoding architecture. 48
4.3 Simplified single-loop (SSL) transcoding architecture. 49
4.4 Open-loop (OL) transcoding architecture. 49
4.5 Coefficients partitioning. 50
4.6 Hierarchical-P with KEY pictures. (GOPSize = 4) 51
4.7 Proposed transcoding method. 52
4.8 Quantization-domain single-loop (QDSL) transcoding architecture. . . 54
4.9 Quantization-domain MC. 55
4.10 Quantization-domain intra prediction (QDIP) transcoding architecture. 56
4.11 Intra 16x16 prediction. 57
4.12 Intra 4x4 mode 3 prediction. 57
4.13 Intra 4x4 prediction. 59
4.14 Base layer copy. 60
4.15 Single-loop based BL-copy architecture. 60
4.16 R-D curves comparison. 64

5.1 Hierarchical-P SVC to hierarchical-P AVC transcoding. 68
5.2 Hierarchical-P SVC to IPPP AVC transcoding. 69
5.3 SVC to AVC mode mapping. 70
5.4 MGS layer prediction structure. 72
5.5 Overall proposed scheme. 74
5.6 R-D curves comparison. 79

List of Tables

2.1 Computational complexity distribution (QP = 20). 9
2.2 Mode percentage difference (frame 37, QP = 20). 10
2.3 INTER partition percentage difference (frame 37, QP = 20). 10
2.4 Mode correlation for co-located MBs (frame 37, QP = 20). 12
2.5 Partition correlation for co-located MBs (frame 37, QP = 20). 12
2.6 Performance comparison of proposed transcoder (lower-layer). 19
2.7 Performance comparison of proposed transcoder (top-layer). 19
2.8 Overall performance of proposed transcoder (top-layer + lower-layer). 20

3.1 Coding modes in SVC 28
3.2 Experimental configurations 40
3.3 Time saving comparison. ("360p": 640x360) 41
3.4 MB mode type ratio. 42

4.1 Coding modes in SVC 52
4.2 Mode 3-7 predictions. 58
4.3 Experimental configurations 61
4.4 Computational time comparisons. 63
4.5 Coding efficiency comparisons. 65

5.1 Group classification of SVC modes 70
5.2 Most similar intra 4x4 directions 71
5.3 Experimental configurations 76
5.4 Performance comparison. 77

LIST OF TABLES

Acronyms

AVC: advanced video coding
AZB: all-zero block
BDBR: Bjøntegaard bit-rate
BDPSNR: Bjøntegaard peak signal to noise ratio
BL: base layer
CABAC: context-adaptive binary arithmetic coding
CAVLC: context-adaptive variable length coding
CBR: constant bit-rate
CIF: common intermediate format
CLMM: coarse-level mode-mapping
CU: coding unit
DCT: discrete cosine transform
DPCM: differential pulse-code modulation
EL: enhancement layer
fps: frame per second
GOP: group of pictures
HEVC: high efficiency video coding
ILP: inter-layer prediction
ISO/IEC: International Organization for Standardization/International Electrotechnical Commission
ITU-T: International Telecommunication Union Telecommunication Standardization Sector
JM: joint model
JSVM: joint scalable video model
MB: macroblock

ACRONYMS

MC: motion compensation
MCP: motion compensated prediction
MCU: multipoint control unit
ME: motion estimation
MPEG: moving picture experts group
MV: motion vector
PPRDO: prediction-pixel based RDO
PSNR: peak signal to noise ratio
PU: prediction unit
QCIF: quarter common intermediate format
QDC: quantization-domain copy
QDIP: quantization-domain intra prediction
QDSL: quantization-domain single-loop
QVGA: quarter video graphics array
R-D: rate-distortion
RDO: rate distortion optimization
RL: reference layer
SAD: sum of absolute difference
SATD: sum of absolute transformed differences
SNR: signal to noise ratio
SR: search range
SVC: scalable video coding
TU: transform unit
VBR: variable bit-rate
VCEG: video coding experts group
VGA: video graphics array
VLC: variable length coding

Chapter 1

Introduction

1.1 Videoconferencing

The term videoconferencing is rooted in two Latin words: videre, or "I see," and conferre, or "to bring together." Videoconferencing enables people to share visual information to overcome distance as a barrier to collaborative work. It allows real-time exchange of digitized video images and sounds between conference participants at two or more separate sites. Videoconferencing has been an important tool for companies and other organizations. Popular videoconferencing manufactures include Polycom, Radvision, Lifesize and Vidyo.

Video communications bring together multiple persons or multiple groups into single multi-site meetings. They link two people through dissimilar computers, videophones, or other communications-enabled devices. Video communications occur as point-to-point and multipoint events. Point-to-point videoconferencing links participants in two sites; multipoint videoconferencing links more than two sites. The device that links three or more locations in a single conference is a multipoint control unit (MCU). MCU is in charge of bit-stream adaptation such as bit-rate reduction and Hollywood square generation.

Due to the large size of uncompressed video, video compression technology becomes a key component of videoconferencing. In nowadays the video coding standards used in videoconferencing usually include H.261, H.263, MPEG-1, MPEG-2, H.264/AVC and H.264/SVC. Among them, AVC is currently most poplar, and SVC

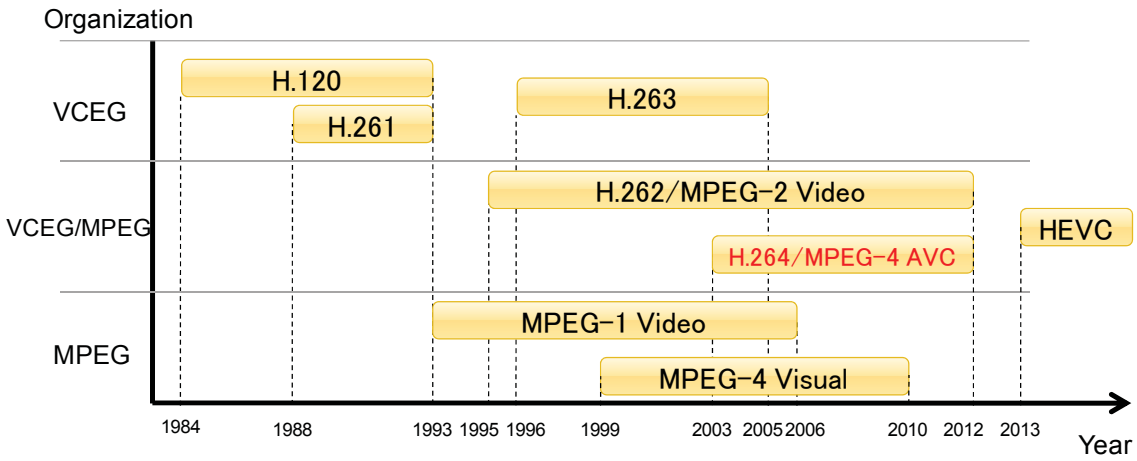


Figure 1.1: Timeline for video coding standards

is going to be the next generation video compression standard used in videoconferencing.

1.2 AVC and SVC

There are mainly two organizations in charge of the development of international video coding standards. One is the ITU-T Video Coding Experts Group (VCEG), and the other one is ISO/IEC Moving Picture Experts Group (MPEG). The video coding standard timeline is shown in Figure 1.1. Several coding standards were/are being developed jointly by VCEG and MPEG, including the popular H.264/AVC coding standard.

The Scalable Video Coding extension of the H.264/AVC standard provides the ability to adapt to diverse client capabilities and requirements, which enables transmission of one bitstream containing multiple subset bitstreams [1, 2]. These subset streams are organized in layered structure and can be extracted adaptively according to user requirements. SVC provides 3 kinds of scalabilities: spatial (resolution) scalability, temporal (frame rate) scalability and quality (SNR, Signal-to-Noise Ratio) scalability. It is a good solution for video broadcasting and video conferencing which involve multiple terminals with different processing capabilities and network

conditions. Performance evaluations of SVC and its key technologies can be found in [3, 4, 5, 6, 7].

1.3 Transcoding

Though SVC achieves good coding efficiency benefitting from the inter-layer predictions [6], it is impossible for every existing or under-developing system to support SVC codec. There are a lot of legacy systems or terminals which do not support SVC standard, and newly developed ones may choose another coding standard due to the system characteristic limitations. These systems or terminals are potential participants in a future video conferencing application. In order to communicate with such kind of user ends, transcoding is needed for SVC-based systems. Now let's assume a multiparty video conferencing scenario, as shown in Figure 1.2. Part A is a new video conferencing system based on SVC standard, and part C is a personal pda user with limited network bandwidth who supports only AVC codec for processing small size frames. Part B is a legacy multipoint control unit (MCU) [8] based system which also supports only H.264/AVC standard. Assume that a desktop PC in B sends a frame to a mobile in A, or a notebook PC in A sends a frame to a pda in C, the receivers may be unable to decode or even receive. Thus, transcoding between AVC and SVC is needed. Note that B probably can not transcode between AVC and SVC since SVC is later standardized than AVC. Therefore, as a solution to provide backward compatibility, the gateway for system B should integrate the functionality of transcoding between AVC and SVC.

A simple and straightforward solution for transcoding is the cascaded re-encoding architecture [9], which fully decodes the input bitstream and then re-encodes. It usually requires high computational cost. In earlier works on transcoding, the majority of interest focused on 2 directions: homogeneous transcoding (same coding standard for input & output bitstreams) [9] and heterogeneous transcoding (different coding standards for input & output bitstreams) [10, 11]. Though SVC has a layered structure, the AVC/SVC transcoding is more of a homogeneous transcoding due to SVC's AVC-compatible single layer encoding, except for the inter-layer predictions. Most conventional works focus on single layer transcoding for bit-rate reduction [12, 14, 15, 16, 17, 18], spatial/temporal resolution reduction [18, 19, 20, 27, 33], CBR

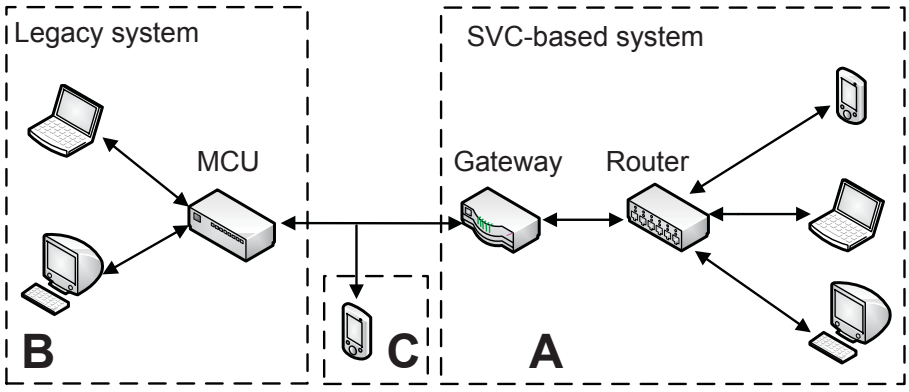


Figure 1.2: A hybrid video conferencing scenario

(constant bit-rate) and VBR (variable bit-rate) conversion, error-resilience transcoding and so on [9]. Newly developed works include AVC/SVC temporal transcoding [38], quality transcoding [23, 24, 25, 26], and SVC-to-AVC spatial transcoding [30]. AVC-to-SVC spatial transcoding has not been fully investigated in existing literatures except for [39] which integrates some existing techniques and achieves about 2/3 time reduction. However, the PSNR (Peak Signal-to-Noise Ratio) drops about 1 dB at the same bit-rate compared with the re-encoding method for many cases, due to the introduced inter-layer prediction, non-optimal mode decision and proportional MV scaling.

1.4 Problem & Target

Since the cascaded re-encoding approach requires great computation, faster transcoding method is needed. This dissertation is focused on low-complexity SVC/AVC transcoding. To enable two-way communication between SVC and AVC systems, this dissertation targets both SVC to AVC and AVC to SVC transcoding. Targeted scalability includes spatial and quality. Temporal scalability is not used in video-conferencing because backward prediction will cause delay. Besides, SVC quality scalability uses fixed hierarchical coding structure with low coding efficiency. Depending on whether AVC uses same coding structure or not, there are two kinds of transcoding approaches for SVC to AVC quality transcoding - homogeneous and het-

erogeneous transcoding. Roughly for all works over 90% time reduction is achieved in this dissertation, without significant coding efficiency loss. Details will be described in following chapters.

1.5 Dissertation Organization

This dissertation is focused on low-complexity transcoding between SVC and AVC formats based on data reuse and data approximation methodologies. ata exploitation means to skip unnecessary processing by utilizing the input data such as mode, motion vector (MV) and residue. Data approximation means to remove computationally heavy components and approximate them with less computation. Drift problem is incurred due to the approximation and it is constrained by techniques like compensation or prevention schemes. The dissertation contents are organized as follows.

In Chapter 2, a low-complexity AVC to VC transcoder with spatial scalability is presented based on coarse-level mode-mapping (CLMM). Different from the conventional deterministic mode mapping methods, a novel mode-mapping strategy is proposed at a "coarser" level based on the context of the co-located area in input AVC frame. Three sub-schemes for mode mapping are proposed by allowing more than one candidate. With CLMM as the key technology, the whole AVC to SVC spatial transcoder is composed by following steps. First, mode skipping schemes are performed, including motion estimation (ME) skipping and probability-based mode control. ME skipping scheme skips the mode which is unlikely to be selected. After that mode control is applied to further reduce the amount of modes by recording the mode percentage profiles. Second, mode mapping is performed based on CLMM schemes. The resulting candidate partitions are examined and the best one is chosen. Finally, motion vector (MV) refinement is applied in order to further reduce the complexity. Motion search is performed in a relatively small range for homogeneous region.

In Chapter 3, a low-complexity SVC to AVC transcoder with spatial scalability is proposed based on hybrid-domain transcoding with drift compensation. Conventional transcoding approaches are based on either pure pixel- or pure frequency-domain. In proposed transcoder, MBs are classified into two types and data reuse

methods are applied accordingly in different domains. In the pixel domain, only mode and motion informations are reused. In the frequency domain, residual information is also reused. This means a lack of encoding loop in the transcoder, resulting in unsynchronized predictors and causing drift problem. Compensation techniques are proposed for I frame and P frame accordingly. In I frame, a rate-distortion metric considering the importance of edge pixels is proposed. The intra prediction accuracy is improved in I frame, and thus ensure better quality of following P frames. In P frames, accumulated drift error is calculated and added back to the input signal.

In Chapter 4, a frequency-domain SVC to AVC homogeneous transcoder with quality scalability is proposed based on quantization-domain motion compensation and intra prediction. Transform operations are totally removed, resulting in extremely fast transcoding. Approximation schemes for both motion compensation and intra prediction are proposed. However, drift problem occurs due to the unsynchronized MCP loops. To constrain the drift error, the "KEY" pictures with lowest temporal layer id are transcoded using drift-free transcoding. Thus drift propagation is limited within every two KEY pictures.

In Chapter 5, a mode-mapping and MV conjunction based SVC to AVC transcoder with quality scalability is proposed. Comparing with the scheme in Chapter 4, better coding efficiency is obtained. The key proposal is the realization of mode/motion reuse for heterogeneous coding structures. The input SVC bitstream is hierarchical-P structured while AVC encoded bitstream is IPPP structured with multiple reference frames. First, the SVC mode is mapped to AVC encoder by proposed mapping strategy. Then the MV information is reused to estimate the MV of non-corresponding reference frame. Finally, an all zero block (AZB) check is performed to early terminate the encoding process.

In Chapter 6, the overall dissertation is summarized and future work is described. Four works were done targeting at low-complexity transcoding between SVC and AVC. Proposed works achieves 89.6%-97.4% time saving without significant coding efficiency loss, and this research is expected to play an important role in a hybrid videoconferencing application.

Chapter 2

Coarse level mode mapping based AVC to SVC spatial transcoding

2.1 Introduction

For reduced resolution transcoding, [19, 20, 27, 33] proposed approaches in the DCT(Discrete Cosine Transform)-domain, which basically achieve larger time reduction compared with pixel-domain approaches. However, drift problem occurs and should be compensated, which needs additional calculation effort and decreases the overall gain. The resultant PSNR drops a lot, averagely about 0.5-0.9 dB for [19], 0.3-0.4 dB for [20], 0.7-1.6 dB for [33] and 0.5-1.5 dB for [27]. Since in video conferencing systems, the quality is usually expected to remain as much as possible, these DCT-domain approaches are not suitable. [18] and [29] are pixel-domain transcoding methods. In [18] the authors utilize 4-to-1 MV mapping with refinement which involves no sub-macroblocks (H.263), and the complexity reduction is claimed to be 23% averagely with approximately 0.7 dB PSNR loss. [29] presents a mode-mapping based downscaling transcoding method. Though refinement through an MV-based block merging scheme is possible, the PSNR still drops about 1-4 dB according to the gap of R-D curves in simulations due to the underlying proportional mapping.

This chapter investigates AVC-to-SVC spatial transcoding and proposes a coarse-level mode-mapping based low-complexity transcoding architecture for video conferencing. For SVC lower-layer encoding, an ME skipping scheme based on the mode

distribution of input AVC stream is adopted for saving unnecessary ME calculations. The search range for ME skipping scheme is determined through an adaptive way. A following probability-profile based mode control method is applied for further mode skipping. Then, for non-skippable MBs, 3 coarse-level mode-mapping methods are presented with different tradeoff between coding efficiency and computational complexity, and the adaptive usage of these methods are also explained. Finally, MV refinement is introduced for further time reduction after mode decision. For SVC top-layer encoding, 2 schemes with different focus on quality or bit-rate are discussed.

This chapter is organized as follows. Section 2.2 analyzes the reference model, i.e. re-encoding method. Overall architecture and algorithms are described in Section 2.3, and Section 2.4 & 2.5 explain the proposed methods in detail for lower layers and top layer respectively. Experimental results are given in Section 2.6, followed by conclusions in Section 2.7.

2.2 Reference model analysis

The cascaded pixel-domain re-encoding architecture for AVC-to-SVC spatial transcoding [39] is selected as the starting point of proposed transcoder and serves as a reference model. 3 procedures are involved: input AVC bitstream decoding, downsampling and SVC encoding (with adaptive inter-layer predictions). Table 2.1 shows the time cost distribution in re-encoding model for 8 test sequences which will be specified in Section 2.6. As expected, the most time-consuming part is the SVC encoding procedure, which involves motion estimations. AVC decoding and downsampling procedures are trivial in computation cost compared with SVC encoding. Complexity reduction in SVC encoding is necessary on the road towards low-delay transcoding for video conferencing. Reduction in top layer is simple since R-D (Rate-Distortion) optimized information from AVC bitstream is available. The follows paragraph discusses the possible solutions for time reduction in reduced-resolution transcoding.

Though more general and statistical analysis is preferred, we only give some representative data to show the trend of motion data correlation between original frame and reduced-resolution frame. Table 2.2 shows the mode percentage difference

Table 2.1: Computational complexity distribution (QP = 20).

Sequence	AVC decoding	Downsampling	SVC encoding
akiyo	2.29%	1.89%	95.82%
panzoom2	2.45%	1.63%	95.92%
vidyo1	2.16%	1.71%	96.13%
vidyo3	2.17%	1.63%	96.20%
bus	2.42%	1.16%	96.42%
football	2.38%	1.19%	96.43%
flower_garden	2.49%	1.13%	96.38%
cheer_leaders	2.26%	0.92%	96.82%

for AVC frame and SVC downsized frame at a randomly selected frame in dyadic transcoding. Here the INTER refers to non-SKIP inter modes, and it holds for the rest of this chapter. It can be inferred from the table that AVC-coded frame and corresponding downsized SVC-coded frame have similar mode distribution. Therefore, mode information reuse is possible. Table 2.4 and Table 2.5 show the average number of MBs within co-located region (4 MBs in total) in input AVC frame which have the same mode or partition as SVC low-resolution MB. We can see that the mode tends to be similar for co-located positions while the partition tends to be very different. Based on this observation, the mode skipping schemes in Section 2.4.1 and Section 2.4.2 are proposed. Besides, although the mode distributions are similar, the MB partitions are usually loosely coupled as shown in Figure 2.1 which shows an example for mode and partition comparison. Table 2.3 shows the percentage difference for INTER partitions which contains large fluctuations for many cases. Therefore, conventional methods based on proportional partition mapping is not suitable. To address this problem, the mode mapping and MV refinement schemes are proposed in Section 2.4.3 and Section 2.4.4.

2.3 Overall transcoding architecture

Figure 2.2 shows the proposed transcoder. The solid lined blocks stand for conventional modules and the slashed blocks stand for proposed methods. Although the proposed methods in following sections are applicable to multiple layers, here

Table 2.2: Mode percentage difference (frame 37, QP = 20).

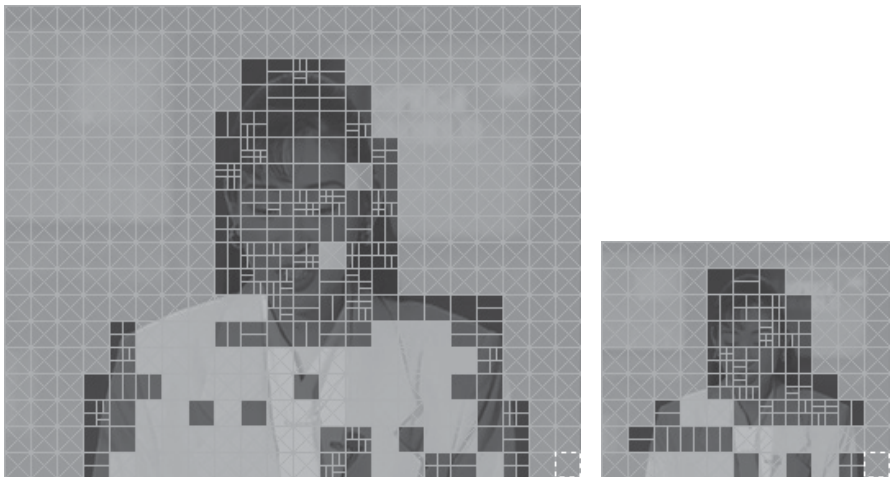
Sequence	INTRA	SKIP	INTER
akiyo	+0.0%	+8.1%	-8.1%
panzoom2	-1.0%	+10.1%	-9.1%
vidyo1	-0.2%	-1.7%	+1.9%
vidyo3	+0.2%	-5.5%	+5.3%
bus	-2.3%	+3.8%	-1.5%
football	-1.5%	+2.8%	-1.3%
flower_garden	-0.6%	+5.2%	-4.6%
cheer_leaders	-0.3%	-4.8%	+5.1%

Table 2.3: INTER partition percentage difference (frame 37, QP = 20).

Sequence	16×16	16×8	8×16	8×8
akiyo	-7.7%	-10.9%	+10.2%	+8.4%
panzoom2	-3.8%	-7.6%	-7.8%	+19.2%
vidyo1	-8.8%	+4.0%	-1.6%	+6.4%
vidyo3	-15.1%	-1.7%	-1.4%	+18.2%
bus	+0.1%	-0.1%	+4.5%	-4.5%
football	-12.9%	-14.8%	-11.4%	+39.1%
flower_garden	-5.3%	-2.2%	-5.6%	+13.1%
cheer_leaders	-8.8%	-7.7%	-4.2%	+20.8%

for simplicity and clarity we only show the 2-layer structure. Both motion compensation (MC) and downsampling are processed in pixel domain. The marks E, Q, T, E_1, E_2 stand for entropy coding, quantization, transformation, top-layer entropy coding and lower-layer entropy coding respectively.

For the lower-layer ME, 4 proposed schemes are applied - 2 mode skipping schemes, 1 mode mapping scheme(incl. 3 sub-schemes) and 1 MV mapping scheme. First, the ME skipping scheme is applied with the intention to skip unlikely mode types which is described in Section 2.4.1. The following profile-based mode control method (Section 2.4.2) is utilized if more than 2 candidate mode types remain after ME skipping. Then, the coarse-level mode-mapping method is applied for INTER MBs which are not skipped by previous steps. Section 2.4.3 presents 3 such kind of methods and explains the adaptive usage of the 3 methods. And at last, MV refinement is applied for further time reduction.



(a) cif@AVC (b) qcif@SVC

Figure 2.1: Intuitive mode & partition comparison for akiyo sequence, frame 37. (light grey: SKIP, dark grey: INTER (non-SKIP))

A is a switch which changes the top-layer strategy according to the network condition. If the bandwidth is enough, the upper routine described in Section 2.5.1 with well-preserved top-layer quality will be selected. Otherwise, the lower routine described in Section 2.5.2 will be adopted for lower bit-rate with degraded top-layer quality.

2.4 Proposed lower-layer transcoding schemes

2.4.1 ME skipping scheme

Although downsampling methods cannot completely construct the relation between high-resolution and low-resolution frames, we notice that there is a rough rule between them, that is the similar mode distribution. If the lower-layer MB is mode X (INTER, INTRA or SKIP; no concern about sub-partitions), then the input AVC frame co-located region (consisting of β^2 MBs and β is the scaling factor) is probably also mode X. This is usually the case except for some irregular MBs caused by downsampling losses. Based on this rule, an ME skipping scheme is proposed.

Table 2.4: Mode correlation for co-located MBs (frame 37, QP = 20).

I	Sequence	INTRA	SKIP	INTER
II	akiyo	-	3.6	3.1
	panzoom2	-	1.2	3.5
	vidyo1	-	3.2	2.8
	vidyo3	0.8	3.2	3.2
	bus	2.0	1.0	3.7
	football	3.4	0.8	3.1
	flower_garden	-	3.0	3.8
	cheer_leaders	2.9	1.1	3.4

I: The mode of SVC low-resolution MB

II: The number of same-mode MBs within 4 co-located MBs

("-" means no such a mode in low-resolution frame)

Table 2.5: Partition correlation for co-located MBs (frame 37, QP = 20).

I	Sequence	16×16	16×8	8×16	8×8
II	akiyo	1.3	0.3	0.3	0.3
	panzoom2	1.6	0.5	0.6	0.0
	vidyo1	1.2	0.7	0.5	0.4
	vidyo3	1.5	0.6	0.6	0.7
	bus	1.8	0.6	0.6	0.0
	football	1.8	0.5	0.9	0.0
	flower_garden	1.2	0.8	0.4	1.4
	cheer_leaders	0.9	0.5	0.7	0.9

I, II: same definition as Table 4

First an adaptive search range in high-resolution frame is defined. The lower bound for the search range is $\beta \times \beta$ (co-located MBs). Assume that the top-layer resolution is $M \times N$, then the upper bound is set to $U \times U$ according to Eq.(2.1) & (2.2).

$$SR\_Width = round\left(\sqrt{\frac{M}{16} \times \frac{N}{16} \times \alpha}\right) \tag{2.1}$$

$$U = \beta \times round\left(\frac{SR\_Width}{\beta}\right) \tag{2.2}$$

The round(.) operator calculates the nearest integer for the parameter. α is the

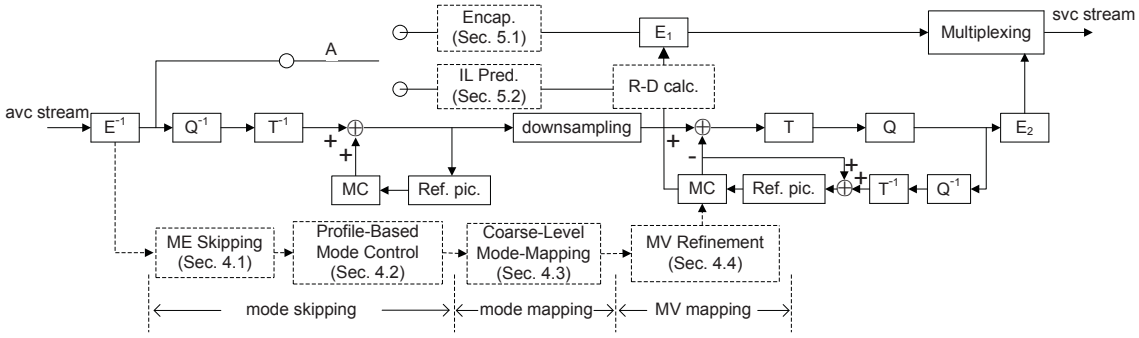


Figure 2.2: Proposed transcoder. (Encap.: Encapsulation, IL Pred.: Inter-Layer Prediction)

percentage of MBs in search range over entire frame. Eq.(2.1) calculates the search range width measured in terms of MBs, and Eq.(2.2) maps the value to multiples of scaling factor in order to make the search range symmetrical to co-located region. Too large α decreases overall time reduction and too small value will lead to non-statistical result. Through vast experiments over different sequences, we found that generally 0.04 gave good performance. In dyadic transcoding ($\beta = 2$), the upper bound is 4×4 for CIF size, 6×6 for VGA size and 12×12 for 720p size when α is set to 0.04. Figure 2.3 shows an example for VGA sequence. The grey region shows the co-located MBs and the numbers mean the search order (from 1 to 36 with increasing distance to center - decreasing relevance to lower-layer MB).

The search range is adapted MB by MB between lower bound and upper bound according to the homogeneity of previous MB. If the search range of previous MB contains only one type of mode, it is considered smoothed and the current MB will decrease the search range by [-2, -2] (e.g. 6×6 -> 4×4). Otherwise, it is considered detailed and the search range will be enlarged by [+2, +2]. Of course, the search range will not across the boundaries.

Then check the modes of these MBs in the search range in predefined order. If SKIP, INTER or INTRA exists, estimate this mode respectively. On the contrary, if some mode does not exist in the search range, then skip the estimation for this mode. This scheme works efficiently when some mode is concentrated in limited areas, which means the other modes may be skipped by proposed scheme.

26	25	24	23	22	21
27	10	9	8	7	20
28	11	1	2	6	19
29	12	3	4	5	18
30	13	14	15	16	17
31	32	33	34	35	36

Figure 2.3: Search range and search order in VGA sequence.

2.4.2 Probability-profile based mode decision control

If more than 2 modes out of SKIP, INTER and INTRA are not skipped by the method in Section 2.4.1, a scheme for further mode selection is adopted by maintaining a profile of mode percentages for high-resolution frame. This method tries to "catch up with" the percentage profile of high-resolution frame.

Assume that we are processing an MB in lower layer. Let N'_X be the number of mode X in high-resolution frame up to current co-located MBs, N''_X be the lower-layer number of mode X over all previous MBs. ΔN is the maximum allowed difference between high-resolution frame and scaled lower layer in terms of MBs and $\Delta\hat{N}$ is the actual difference as shown in Eq.(2.3). β is the scaling factor and M_X calculated by Eq.(2.4) is a signed measurement for the lower-layer deviation from top layer, ranged from -1 to 1. P''_X denotes the probability of mode X for current MB in lower layer and Eq.(2.5) maps its range to [0, 1]. ΔN is typically set to 20 in our dyadic experiments which means the maximum allowed deviation for lower-layer is 5 MBs ($20/2^2$).

$$\Delta\hat{N} = N'_X - \beta^2 \times N''_X \tag{2.3}$$

$$M_X = \begin{cases} -1 & , \Delta\hat{N} \leq -\Delta N \\ 1 & , \Delta\hat{N} \geq \Delta N \\ \Delta\hat{N}/\Delta N & , \text{otherwise} \end{cases} \tag{2.4}$$

$$P_X'' = \frac{M_X + 1}{2} \tag{2.5}$$

For each existing mode, a lottery function based on the estimated probability P_X'' is applied to judge whether to skip that mode. If all the modes are skipped, the mode with largest probability will be selected.

Figure 2.4 shows an example for INTER mode. The horizontal axis denotes the lower-layer MB number and the vertical axis is the percentage of INTER-coded MBs. The straight line shows the profile for high-resolution frame and the dotted line shows the profile for lower layer by proposed method. It can be seen that the mode distribution is well mirrored.

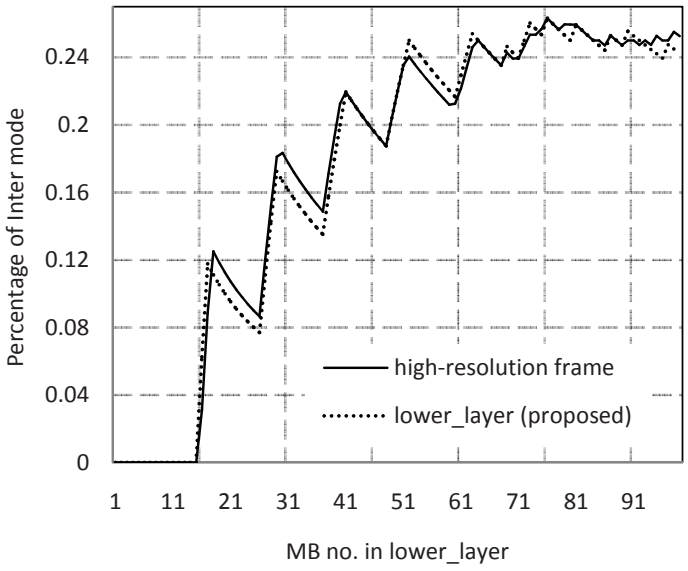


Figure 2.4: INTER (non-SKIP) mode profile for akiyo sequence, frame 37.

2.4.3 Coarse-level mode-mapping methods

Although there is a similar mode distribution rule between high-resolution and low-resolution frames, this is not the case for partitions and MVs of INTER MBs. They have rarely proportional relations. Therefore, lower layer should not be mapped by scaling partition and MV directly [29]. Instead, we find another coarse-level rule

that if lower-layer MB has few details, AVC co-located MBs usually also have few details. On the contrary, if lower-layer MB has many details, AVC co-located MBs probably also have many details (but not have to be proportional). Based on this rule, 3 mode mapping methods for INTER estimation with different tradeoff between coding efficiency and complexity are explained in the following paragraphs, and an adaptive usage is proposed to achieve an optimal combination of the 3 methods.

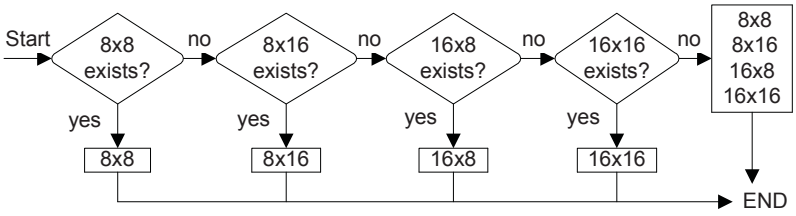


Figure 2.5: Direct mapping method.

The first method is the direct-mapping method, which is a 4-to-1 mapping as shown in Figure 2.5. This method first checks the co-located MBs to see if 8×8 mode (no concern about sub-partitions) exists. If it exists, stop the procedure and estimate 8×8 (incl. sub-partitions) only. Otherwise, continue to check 8×16, 16×8 and 16×16 similarly. If no mode is selected at last, all INTER modes will be estimated.

Candidate-mapping method is the second approach which performs ME for candidate modes only. This method selects co-located MBs' modes as candidates for lower-layer encoding, as shown in Figure 2.6. It checks 8×8, 8×16, 16×8 and 16×16 sequentially in co-located MBs. If some mode exists, it will be added to estimation list. Otherwise, it will not be added. When the procedure finishes, only the modes in estimation list will be estimated. If no mode exists in the estimation list at last, all INTER modes will be estimated.

Another method is the priority-mapping method, which performs ME based on priority. The priory is defined as: 8×8 > {8×16, 16×8} > 16×16. This is because the complexity and uncertainty of detailed MB is larger than smooth one. This method checks from 8×8 to 16×16 as shown in Figure 2.7. If some mode exists, estimate all modes with larger (or equal) priority. Similarly, if no INTER mode exists at last, all modes will be estimated.

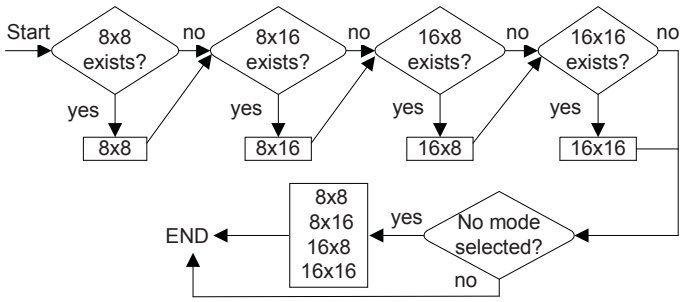


Figure 2.6: Candidate mapping method.

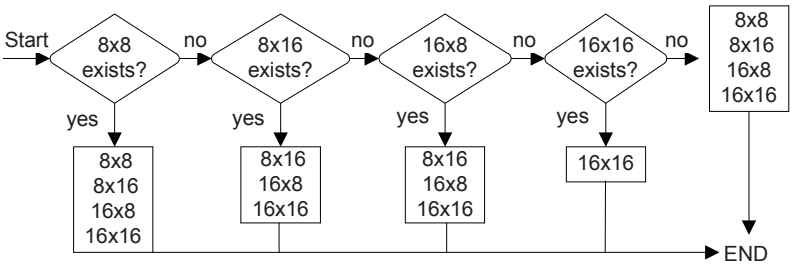


Figure 2.7: Priority mapping method.

These methods reuse the AVC stream information at a coarser level which involves no sub-partitions due to the irregularity of sub-partitions. Table 2.6 in Section 2.6 shows that the direct mapping method has the largest complexity reduction while priority mapping method achieves the best coding efficiency, and candidate mapping method performs moderately.

In proposed transcoder, they are combined adaptively according to the homogeneity of current search range. 3 levels are defined: level 1 with less than 1/3 SKIP or INTER_16×16 modes in current search range, level 2 with less than 2/3 but more than 1/3 SKIP or INTER_16×16 modes, level 3 with more than 2/3 SKIP or INTER_16×16 modes. Level 1 is the most detailed, so the most accurate priority-mapping method is adopted. Level 2 is moderately detailed and candidate-mapping method is selected. Direct-mapping method is used in level 3 which is the least detailed.

2.4.4 MV refinement scheme

Some conventional works focused on MV refinement based on nearly whole-frame MV mapping [19, 31], which turned out to be inaccurate and caused efficiency loss. However, MV refinement is expected to be more efficient in homogeneous area compared with detailed area since less MVs are involved. Detailed area which leads to more MVs will increase the complexity and uncertainty for MV mapping. In proposed transcoder, MV refinement is only applied for MBs whose co-located MBs are all SKIP or INTER_16×16. Before applying the MV refinement another check is executed - the MV diversity of co-located MBs. Eq.(2.6) calculates the arithmetic average MV among co-located MBs. β is the scaling factor and MV_i_x & MV_i_y represent the horizontal and vertical components for i_{th} MB respectively. Eq.(2.7) calculates the diversities of horizontal and vertical MV components by summing the absolute difference (SAD) between the MVs of co-located MBs and the average MV.

$$\begin{cases} \overline{MV\_x} = \frac{1}{\beta^2} \sum_{i=0}^{\beta^2-1} MV_i\_x \\ \overline{MV\_y} = \frac{1}{\beta^2} \sum_{i=0}^{\beta^2-1} MV_i\_y \end{cases} \tag{2.6}$$

$$\begin{cases} SAD_x = \sum_{i=0}^{\beta^2-1} |MV_i\_x - \overline{MV\_x}| \\ SAD_y = \sum_{i=0}^{\beta^2-1} |MV_i\_y - \overline{MV\_y}| \end{cases} \tag{2.7}$$

Then the SAD values are compared with pre-defined thresholds, as shown in Eq.(2.8). If Eq.(2.8) holds, the MV refinement is applicable. Otherwise, it will not be performed. Smaller thresholds will constrict the applicable rate while larger thresholds will result in worse coding efficiency. In our experiments, the thresholds Th_x and Th_y are both set to 4 since small thresholds give no harm anyhow.

$$\begin{cases} SAD_x \leq Th_x \\ SAD_y \leq Th_y \end{cases} \tag{2.8}$$

$$\begin{cases} MV\_scaled\_x = \frac{1}{\beta}\overline{MV\_x} \\ MV\_scaled\_y = \frac{1}{\beta}\overline{MV\_y} \end{cases} \tag{2.9}$$

If the MV refinement is feasible, INTER_16×16 is chosen as the lower-layer MB mode. The scaled average MV calculated by Eq.(2.9) will be used for lower-layer ME, and it is used as the starting point for motion search with a much smaller search

Table 2.6: Performance comparison of proposed transcoder (lower-layer).

Sequence	direct			candidate			priority			adaptive		
	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
akiyo	+6.45	-0.34	-71.4	+5.21	-0.28	-70.2	+2.38	-0.13	-68.3	+3.50	-0.17	-70.0
panzoom2	+8.54	-0.42	-66.9	+6.82	-0.33	-63.7	+3.34	-0.16	-62.0	+3.40	-0.16	-63.7
vidyo1	+11.50	-0.46	-73.6	+7.13	-0.29	-72.2	+4.46	-0.19	-70.6	+5.14	-0.21	-72.1
vidyo3	+8.51	-0.39	-73.6	+4.45	-0.21	-72.0	+2.92	-0.14	-70.3	+3.28	-0.17	-72.7
bus	+6.75	-0.43	-44.5	+3.28	-0.22	-37.2	+2.46	-0.16	-32.8	+2.71	-0.18	-34.7
football	+5.72	-0.34	-37.4	+4.56	-0.27	-33.4	+3.69	-0.22	-29.1	+3.89	-0.23	-32.4
flower_garden	+5.70	-0.36	-56.3	+4.05	-0.26	-50.7	+3.55	-0.24	-46.9	+3.75	-0.25	-49.4
cheer_leaders	+4.35	-0.35	-35.1	+3.42	-0.27	-29.5	+2.58	-0.20	-22.1	+2.85	-0.22	-24.0

Criterions: C1: BDBR(%), C2: BDPSNR(dB), C3: $\Delta time(\%)$

Table 2.7: Performance comparison of proposed transcoder (top-layer).

Sequence	Direct Encapsulation			Inter-Layer Prediction		
	Δ bit-rate (%)	Δ Y-PSNR (dB)	Δ time(%)	Δ bit-rate (%)	Δ Y-PSNR (dB)	Δ time(%)
akiyo	+7.49	+1.510	-96.4	+1.11	-0.078	-85.9
panzoom2	+11.60	+1.488	-97.9	+0.98	-0.063	-84.0
vidyo1	+8.97	+1.364	-97.4	+0.64	-0.160	-86.9
vidyo3	+5.41	+1.419	-97.6	+1.52	-0.133	-87.1
bus	+7.94	+1.577	-96.5	+3.60	-0.087	-77.1
football	+4.10	+1.555	-96.6	+0.65	-0.115	-81.5
flower_garden	+8.05	+1.653	-94.6	+1.13	-0.085	-77.8
cheer_leaders	+4.34	+1.477	-97.5	+0.38	-0.170	-80.3

range compared to original search window. In our experiments, the refinement search range is selected as [-2, +2] for both horizontal and vertical components.

2.5 Proposed top-layer transcoding schemes

2.5.1 Direct encapsulation

One approach for top-layer transcoding is to transmit the top-layer bit-rates directly without full decoding and re-encoding. The top-layer AVC bitstream is first VLC (Variable Length Coding) decoded and then VLC re-coded using SVC encoder. There is no quality loss since no quantization is needed. After the VLC re-coding, the top-layer bitstream is multiplexed with lower-layer bitstream which is generated using proposed lower-layer schemes. This behaviour is actually very similar to simulcast transmission except that the final bitstream formats are different. In simulcast, the bitstreams are transmitted as 2 AVC streams while in direct encapsulation method the bitstreams are multiplexed into SVC format. [6] points out that the SVC coding tools are less efficient for spatial scalability especially for simple and slow-motion scenes, which is often the case in video conferencing applications. Therefore, direct encapsulation is recommended for video conferencing if the bandwidth is sufficient.

Table 2.8: Overall performance of proposed transcoder (top-layer + lower-layer).

Sequence	Direct Encapsulation		Inter-Layer Prediction	
	Δ bit-rate (%)	Δ time(%)	Δ bit-rate (%)	Δ time(%)
akiyo	+6.69	-91.1	+1.59	-82.7
panzoom2	+9.96	-91.1	+1.46	-79.9
vidyo1	+8.20	-92.3	+1.54	-84.0
vidyo3	+4.98	-92.6	+1.87	-84.1
bus	+6.89	-90.3	+3.42	-72.9
football	+4.06	-90.2	+1.30	-76.6
flower_garden	+7.19	-90.1	+1.65	-75.0
cheer_leaders	+4.04	-90.2	+0.87	-74.7

2.5.2 Inter-layer prediction utilization

As another choice, the inter-layer predictions can be utilized when the bandwidth is crucial. In direct encapsulation method, R-D costs for different modes which the inter-layer predictions need can not be obtained since there is no ME performed in top layer. As shown in Figure 2.2, we only re-calculate the R-D cost according to the mode and MV got from the AVC bitstream, which have been R-D optimized already. It should be noticed that the source sequence can not be obtained on the transcoder side, so we use the decoded sequence instead as the input for R-D cost calculation in SVC encoder, just like the re-encoding method. The inter-layer motion and intra predictions act the same as original SVC encoder while residual prediction is only performed for the corresponding mode and MV got from AVC frame.

Although the transcoder processing speed decreases a little, the overall complexity is still kept very low since ME is not performed. The drawback for this scheme is the degraded quality due to a second-time quantization loss. It is recommended for bandwidth-crucial applications.

2.6 Experimental results

In this section, proposed methods are applied to some sequences and the results are shown. All experiments are performed on an Intel Core 2 (2.67GHz) computer with 2.0GB RAM and software implementation is based on JM (Joint Model) 17.2

and JSVM (Joint Scalable Video Model) 9.18. JSVM's AVC compatible decoder and down-converter are used for AVC decoding and downsampling processes respectively. 8 sequences are examined with 2-layer dyadic spatial scalability. Akiyo, panzoom2, football and bus are CIF to QCIF transcoding at 30 fps; flower_garden and cheer_leaders are VGA to QVGA transcoding at 30 fps; vidyo1 and vidyo3 are 720p to 360p (640×360) transcoding at 60 fps. Akiyo, panzoom2, vidyo1 and vidyo3 are sequences similar to video-conferencing scenes with still background, slow motions or camera motions, while the other 4 are complex and detailed ones. For each sequence 150 frames are tested with the GOP structure of IPPP. The search range is 16 for CIF-to-QCIF transcoding and 32 for the rest. In experiments the QPs (Quantization Parameter) for AVC encoder and transcoder are set to same values, and QPs are selected as 20, 24, 28 and 32. Other parameters are carefully selected to insure the comparability between proposed transcoder and the reference model.

Table 2.6 shows the lower layer gains for the 3 methods as well as the adaptive usage, compared with re-encoding model. The methods in Section 2.4.1, 2.4.2 and 2.4.4 are applied and mode-mapping method is switched between the 4 methods described in Section 2.4.3. It can be seen that adaptive method achieves almost the same time reduction as candidate method, while obtaining comparable coding efficiency to priority method. It should be noticed that proposed adaptive method achieves 69.6% time reduction averagely for the top 4 sequences, which are video-conferencing similar scenes. For the following 4 sequences, the time reduction is only 35.1% averagely.

Table 2.7 shows the top-layer results for the 2 top-layer methods. The adaptive method in Sec.4.3 is selected as lower-layer transcoding method along with other schemes (Section 2.4.1, 2.4.2 & 2.4.3). The bit-rate data contains only top-layer bit-rate and the lower-layer bit-rate is not included. To fairly compare the top layer quality, the Y-PSNR between original sequence (user side, encoded by AVC and sent to transcoder) and the reconstructed sequence after transcoding are calculated, since it's meaningless to calculate the Y-PSNR between decoded sequence (transcoder side, decoded and used as SVC encoder input) and reconstructed sequence which is identical to the original one in case of direct encapsulation.

Table 5.4 shows the overall results. The lower-layer scheme is fixed - Section 2.4.1, 2.4.2, 2.4.4 and adaptive method in Section 2.4.3. Both the overall bit-rate

increments and overall time reductions are shown for the 2 top-layer methods.

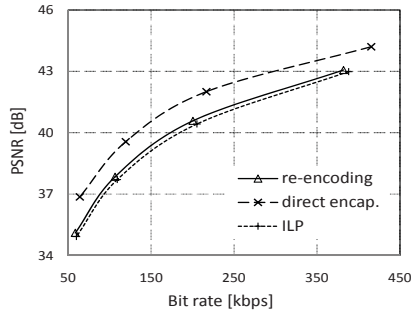
The direct encapsulation method gains averagely 91% overall time reduction for tested sequences with 6.69% overall bit-rate increase and 1.34 dB top-layer quality increment. The time reduction for top 4 sequences is 1.6% larger than the lower 4 sequences. The merit for this method is the significant time reduction and the well-preserved top-layer quality since there is no second-time encoding.

By contrast, the ILP approach keeps the overall bit-rate low while still obtaining 78.7% overall time reduction averagely. It decreases the bit-rate increment to 1.71%. The time reduction for lower 4 sequences decreases by 7.9% compared with top 4 sequences. It is suitable for applications with limited network bandwidth. The main drawback is the degraded top-layer quality due to re-quantization loss which is a little worse than re-encoding method.

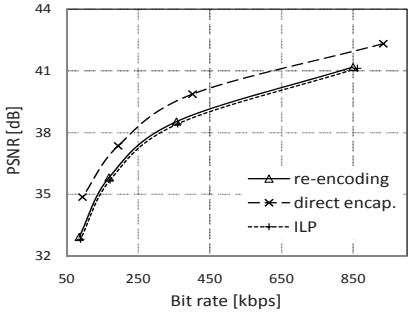
Figure 2.6 shows the top-layer R-D curves for re-encoding method and proposed transcoder with 2 different top-layer methods (Section 2.5.1 & Section 2.5.2). The X-axis shows the overall bit-rate since in ILP the lower-layer bit-rate is required for top-layer decoding. It would be unfair if we only compare the top-layer bit-rates. The Y-axis shows the Y-PSNR for top layer. We can see that the direct encapsulation method achieves best coding efficiency while ILP method is slightly worse than re-encoding method. Direct encapsulation method achieves higher quality and lower complexity compared with re-encoding method, while the overall bit-rate increases. The degree of quality increment is much higher than bit-rate increase, resulting in higher coding efficiency. ILP method achieves lower complexity compared with re-encoding method, however, the quality degrades a little and the bit-rate increases a little, causing the coding efficiency to decrease.

2.7 Conclusions

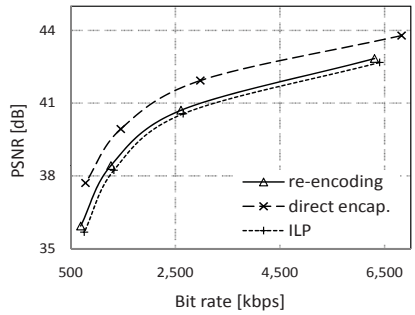
This chapter proposes a low-complexity AVC to SVC spatial transcoder based on coarse-level mode mapping for video conferencing systems. For lower layer (with reduced picture size) transcoding, 2 mode skipping methods are first applied as described in Section 2.4.1 and Section 2.4.2. Then a coarse level mode-mapping method is applied which adaptively selects different sub-schemes described in Section 2.4.3, followed by an MV refinement scheme for special cases for further time



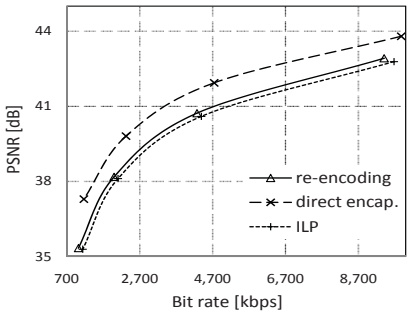
(a) akiyo



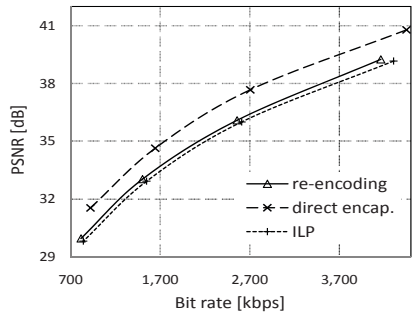
(b) panzoom2



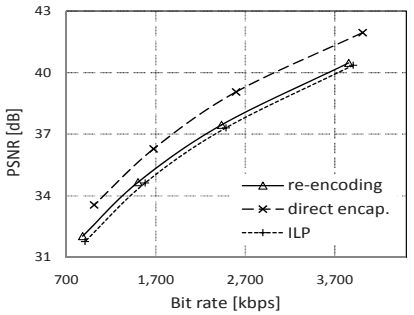
(c) vidyo1



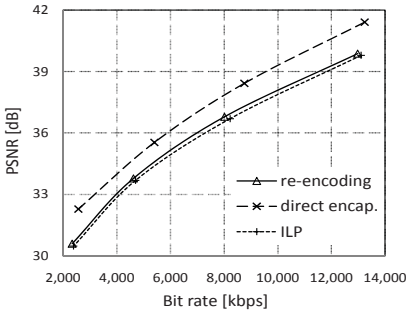
(d) vidyo3



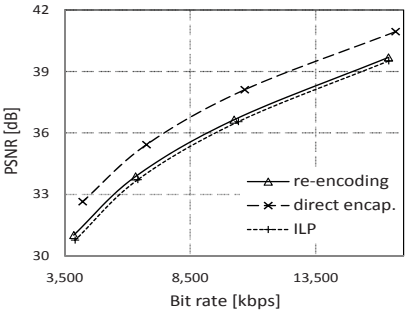
(e) bus



(f) football



(g) flower_garden



(h) cheer_leaders

Figure 2.8: R-D curves comparison for top layer.

reduction. And for the top layer (with identical picture size to AVC frame), 2 schemes are possible according to the network condition. Section 2.5.1 depicts the direct encapsulation method which is suitable when the bandwidth is sufficient, and Section 2.5.2 shows another approach which utilizes the inter-layer predictions of SVC for reducing the bit-rate. Simulation results show that direct encapsulation method achieves significant time reduction with much higher coding efficiency than re-encoding method, since no second-time quantization is involved. The ILP method achieves lower bit-rate than direction encapsulation when the QP is the same, while the time reduction reduces by 12.3% averagely. The coding performance of ILP method is slightly worse than re-encoding method.

Chapter 3

Drift compensated hybrid-domain SVC to AVC spatial transcoding

3.1 Introduction

Due to the continuous progress in video coding standards, transcoding has been an important task for bitstream adaptation or communication between different standards. A straightforward transcoding solution is the full re-encoding method. It fully decodes the input bitstream and then re-encodes the decoded pictures, consuming intensive computations. In previous works on transcoding, how to reduce the complexity without significant efficiency loss is usually the most concerned issue. Basic transcoding approaches are explained in [9, 32] for bit-rate reduction, resolution reduction, format conversion and so on. These transcoding techniques mainly fall into two categories, i.e., pixel-domain transcoding and transform-domain transcoding. The pixel-domain approaches carry out the main transcoding operations (motion compensation, downsampling, etc.) on decoded pictures. Representative works based on pixel-domain transcoding include [10, 12, 18, 19, 29]. The transform-domain approaches carry out the main operations directly on transform coefficients. Representative works based on transform-domain transcoding include [13, 14, 16, 20, 33, 34, 35]. Generally the transform-domain transcoding achieves more time reduction, but less coding efficiency than the pixel-domain transcoding due to the drift problem.

3. DRIFT COMPENSATED HYBRID-DOMAIN SVC TO AVC SPATIAL TRANSCODING

SVC standard provides an SVC-to-AVC rewriting scheme [36]. However, it only supports quality scalability. It cannot be applied to spatial scalability because up-sampling of quantized coefficients would be required, which is very difficult. Recent representative works on SVC/AVC transcoding support the SVC scalability in terms of temporal [37, 38], quality [23, 24, 25, 26] and spatial [30, 39, 40]. In literatures [23, 24, 25, 26] the authors propose a fast AVC-to-SVC quality transcoding method based on transform-domain approaches and achieve more than 99% time reduction compared with the full re-encoding method. For AVC-to-SVC temporal transcoding, the pixel-domain methods give good performance in both time reduction and coding efficiency [37, 38], since large portion of the motion data can be reused directly. A pixel-domain AVC-to-SVC spatial transcoder is described in [39] which is mainly based on motion data adaptation and refinement. In [40] we further proposed a fast and efficient AVC to SVC transcoding architecture based on pixel-domain mode-mapping. SVC-to-AVC temporal transcoding is very nature by simple syntax adaptations. SVC-to-AVC quality transcoding is not an urgent issue since bitstream rewriting can be used instead of transcoding by constraining the SVC terminals to support this functionality. In [30], a pixel-domain fast mode decision method is proposed for SVC-to-AVC spatial transcoding. The original motion data from the input SVC bitstream are utilized to speed up the AVC encoder mode decision process. MBs are classified into three types and treated with different mode-mapping strategies. Only the deduced modes need to be estimated, and the reference pictures & MVs are both reused. This work achieves averagely 94.4% time reduction and slightly higher coding efficiency compared with the full re-encoding method.

In this chapter, a fast SVC-to-AVC spatial transcoder is proposed. Coding efficiency is improved comparing with reference [30]. The input to our transcoder is SVC-encoded bitstream with multiple resolution layers, and the output is AVC-encoded bitstream with highest resolution. The target device to install our transcoder is usually a specific router or gateway in a videoconferencing system. This chapter proposes a hybrid-domain transcoding architecture by combining the pixel-domain transcoding and transform-domain transcoding. The input SVC MBs are classified according to their mode types and processed in different domain. Drift is caused by proposed transcoder, which is solved by compensation techniques.

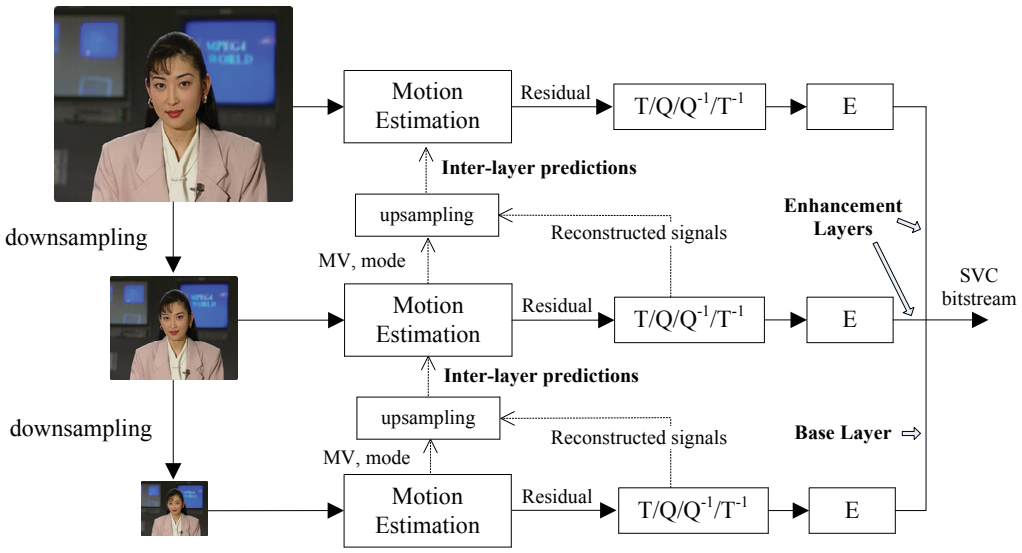


Figure 3.1: SVC coding structure with spatial scalability.

The rest of this chapter is organized as follows. Section 3.2 briefly describes the SVC standard and the related coding modes. Section 3.3 explains the details of proposed hybrid-domain transcoder, and the resulted drift problem is solved in Section 3.4. Section 3.5 shows the overall transcoding structure with drift compensation. Simulation results are given in Section 3.6, and conclusions are drawn in Section 3.7.

3.2 Scalable Video Coding

3.2.1 Inter-layer Predictions

SVC adopts a layered coding structure. Figure 3.1 shows an example of SVC coding structure with 3 spatial layers, defined by the standard. Each layer corresponds to one particular spatial resolution. The coding tools within each layer is identical to AVC, but a new tool named inter-layer prediction is introduced between layers. The inter-layer predictions try to reduce the higher-layer bit-rate by exploring the lower-layer information. There are three kinds of inter-layer predictions, i.e., inter-layer intra, residual and motion predictions. These inter-layer predictions will be denoted

as IL_Intra, IL_Residual and IL_Motion predictions hereafter.

IL_Intra prediction predicts the higher-layer signal using the upsampled lower-layer reconstructed signal. The lower layer is usually called the "reference layer (RL)" and higher layer is called "enhancement layer (EL)". The RL is encoded first, and then the RL reconstructed signal is upsampled and used as the predictor for EL signal. The residual is transmitted after transform, quantization and entropy coding. IL_Residual prediction tries to predict the residual data generated by normal INTER prediction. The upsampled RL reconstructed residual is used as the predictor. The resulted second residual is transmitted after transform, quantization and entropy coding. IL_Motion prediction tries to reduce the size of motion data for INTER coded MBs, such as coding mode and MV. The upsampled RL mode and MV are utilized to predict the EL motion data. For more details, the reader is referred to [1].

3.2.2 Coding modes in SVC

Besides the three inter-layer predictions explained in previous section, SVC also inherits the conventional AVC modes (INTRA and INTER). The IL_Intra prediction is totally independent from the AVC INTRA or INTER modes, while the IL_Residual and IL_Motion predictions are additional refinements based on AVC INTER mode. Thus the coding modes in SVC are shown in Table 3.1. It is also possible that IL_Residual and IL_Motion both exist for an INTER MB. In such case, it is considered as IL_Residual. For short, "INTER with IL_Residual" and "INTER with IL_Motion" will be denoted as IL_Residual and IL_Motion hereafter.

Table 3.1: Coding modes in SVC

Inherited modes	Newly introduced modes
INTRA INTER without ILP	IL_Intra INTER with IL_Residual INTER with IL_Motion

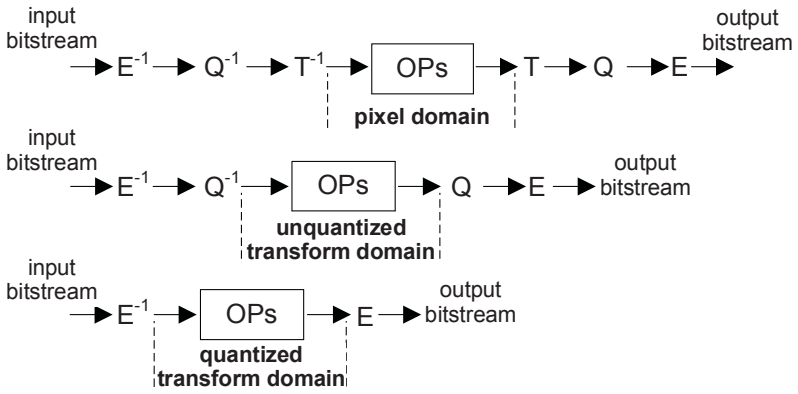


Figure 3.2: Different transcoding domains.

3.3 Hybrid-domain SVC-to-AVC Transcoding

In this section, a hybrid-domain transcoding architecture is proposed for SVC-to-AVC spatial transcoding. The transcoding approaches in the pixel domain and the transform domain are explained respectively.

3.3.1 Hybrid-domain transcoding

As mentioned in the Introduction section, the transcoding domains mainly include the pixel domain and the transform domain. The pixel domain is also called "spatial domain" in some literatures. The transform domain is also called "frequency domain", "DCT (discrete cosine transform) domain" or "compressed domain" sometimes. The following discussion is based on the popular hybrid DPCM/DCT (DPCM - differential pulse-code modulation) coding structure [41], which includes motion compensation, DCT transform, quantization and entropy coding.

The transform domain is further divided into two types, i.e., unquantized transform domain and quantized transform domain. Together with the pixel domain, the three kinds of domains are shown in Figure 3.2. Here the marks E, Q, T stand for entropy coding, quantization and DCT transform respectively. The superscript "-1" represents the inverse process. The "OPs" stands for the main transcoding operations, such as motion compensation, downsampling, logo/watermark insertion, and so on. The top sub-figure in Figure 3.2 shows the pixel-domain transcoding, which

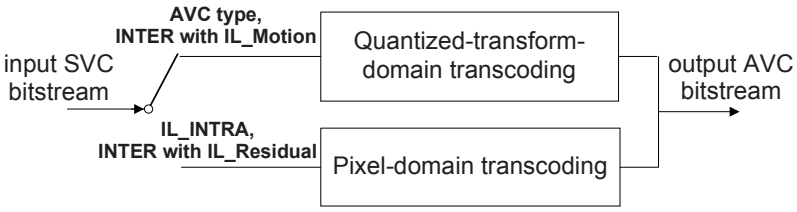


Figure 3.3: Hybrid-domain transcoding

operates after inverse transform and before forward transform. The full re-encoding method is a special case of the pixel-domain transcoding. Most pixel-domain approaches try to utilize the motion data from the input bitstream to accelerate the encoding process. The transform domain is divided into unquantized and quantized transform domains, shown as the middle and bottom sub-figures in Figure 3.2. Most conventional works on transform domain fall into the "unquantized transform domain" category. In quantized transform-domain transcoding there is no re-quantization, and quantized transform coefficients are used directly.

In this chapter, we propose a hybrid-domain transcoding architecture by combining the pixel-domain and quantized transform-domain transcoding. Conventional methods adopt pixel-domain or unquantized transform-domain transcoding, which dequantizes the input bitstream and then re-quantizes it. This will cause coding efficiency loss for every MB. Our transcoder divides the MBs into 2 groups. For one group, pixel-domain transcoding is applied. For the other group, transform-domain transcoding is utilized, which has lower complexity and higher coding efficiency since there is no re-quantization loss.

In Figure 3.3 the AVC-type MBs and INTER with IL_Motion MBs are processed in quantized transform domain. The residual generation of these MBs in SVC encoding is identical to AVC encoding, therefore the residual could be reused in the form of quantized transform coefficients. Thus the re-quantization loss is avoided and transcoding time is reduced. On the contrary, the residuals of IL_Intra and INTER with IL_Residual MBs are generated in a different way comparing with AVC encoding. The residuals cannot be reused. For these MBs the pixel-domain transcoding is applied. Details are explained in following subsections.

3.3.2 Pixel-domain transcoding

Figure 3.4 shows a more intuitive version of proposed transcoder. If the input MBs are coded with IL_Intra mode or INTER mode with IL_Residual prediction (the shadowed MBs), a mode mapping based fast re-encoding method is applied. The mode decision in AVC encoding process is accelerated by deducing the mode from the input SVC bitstream. Exhaustive motion estimation is not needed and thus the transcoding time is saved greatly. The following paragraphs describe the proposed mode mapping method for IL_Intra and INTER mode with IL_Residual prediction, respectively.

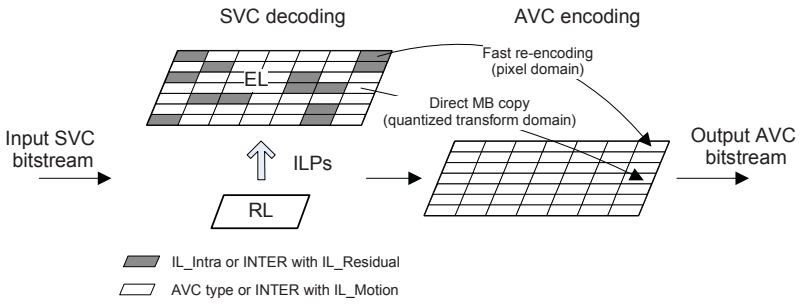


Figure 3.4: Transcoding in pixel domain and quantized transform domain.

The IL_Intra prediction directly uses the lower-layer reconstructed MB for prediction, implying a high correlation between enhancement layer and reference layer. IL_Intra prediction is an special coding mode in SVC which is independent from conventional INTER or INTRA modes. No INTER partition or INTRA prediction direction information is transmitted in enhancement layer except a signal which means that IL_Intra prediction is used. In our transcoder, the reference-layer INTRA prediction partition along with the INTRA prediction direction are used as the AVC encoding mode.

For the IL_Residual predicted MB, the mode mapping depends on whether IL_Motion prediction is used. If it is used, the enhancement layer transmits no mode information except MV difference if it exists. The enhancement-layer INTER partition and MVs are reconstructed from the reference layer, i.e., upsampled reference-layer partition and MVs. If some MV difference is transmitted in enhancement layer,

it is added back to the upsampled reference-layer MV. The final reconstructed INTER partition and MVs are directly used as the AVC encoding mode. If IL_Motion prediction is not used, mode information is transmitted in the enhancement layer. Also, reference layer transmits the mode information which is independent from the enhancement layer. The enhancement-layer mode together with the upsampled reference-layer mode are provided as candidate modes for AVC encoding. Through rate-distortion optimization, better one will be selected as the final mode.

Figure 3.5 shows the accuracy ratio for proposed mode mapping method. The full re-encoding method is selected as the ground truth. If the predicted mode is the same as the mode encoded by the full re-encoding method, it is considered accurate. The average accuracy for IL_Intra and IL_Residual are 86% and 79% respectively. The impact on coding efficiency loss is expected to be very small.

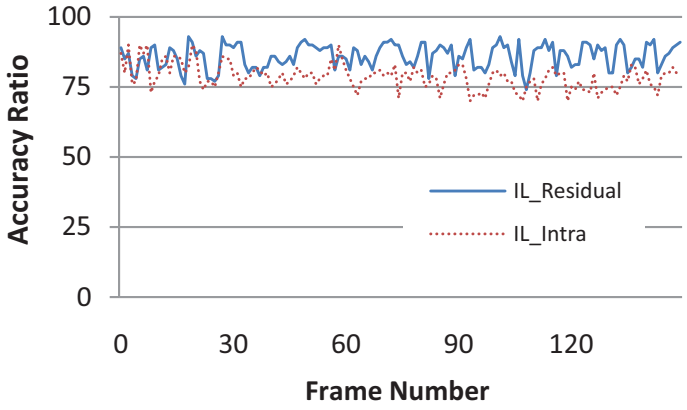


Figure 3.5: Accuracy ratio for mode mapping. (akiyo sequence, 2 spatial layers, 150 frames)

However, through experiments we find that if the MVs are reused as it is for IL_Residual predicted MBs, the coding efficiency drops a lot. Instead of using the MVs directly, a further motion search is applied after the mode mapping in proposed transcoder. As shown in Figure 3.6, the end-point of the input SVC bitstream (reconstructed) MV is set as the center of the refined search area, and the search range is greatly reduced. In our experiments, the refined search range is selected as [-2,+2] for both horizontal and vertical directions.

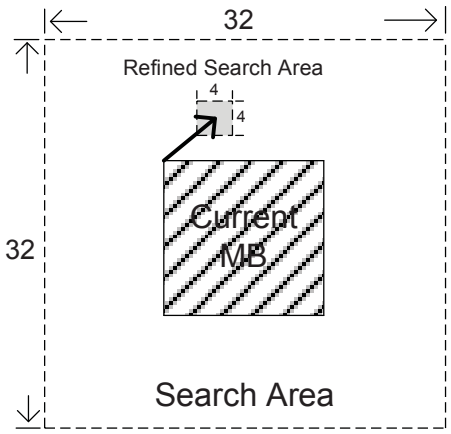


Figure 3.6: MV refinement.

The overall pixel-domain transcoding scheme is shown in Figure 3.7. The previously described mode mapping methods are first applied according to the MB type. Then for IL_Residual predicted MBs, the MV refinement scheme is applied.

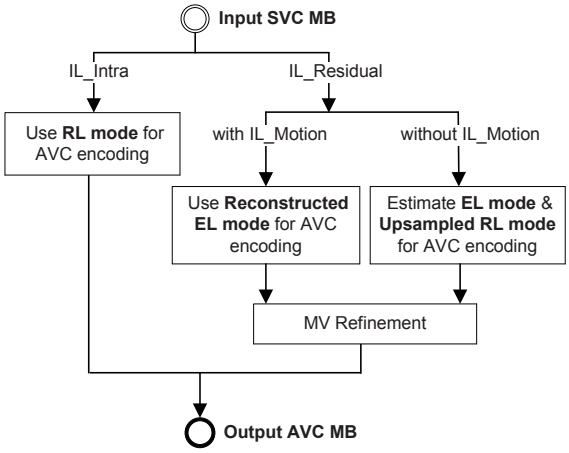


Figure 3.7: Pixel-domain transcoding.

3.3.3 Quantized transform-domain transcoding

In Figure 3.4, if the MBs are coded with AVC-type modes or INTER mode with IL_Motion prediction (the unshadowed MBs), the SVC MB data are copied directly

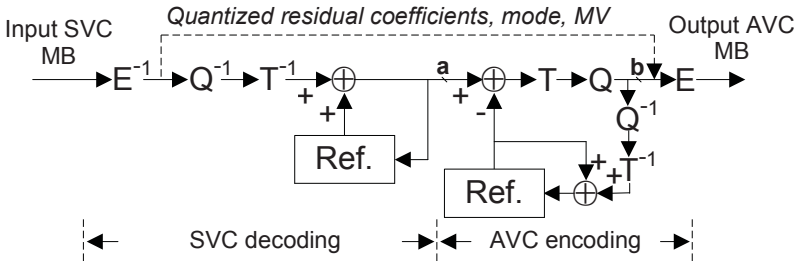


Figure 3.8: Quantized transform-domain transcoding.

into the the AVC MB. The residual is copied in the form of quantized transform coefficients. The motion data and other side information are also copied directly. Figure 3.8 shows the transcoding path in the quantized transform domain, where "Ref." stands for the reference picture buffer. The quantized residual and motion data are extracted after entropy decoding in SVC decoding. They are copied into AVC encoding at the position before entropy coding. Thus, there is no quality loss for the residual data since it is not re-quantized. The operations between point **a** and point **b** are not performed. Thus transcoding time is greatly reduced. The full SVC decoding path is still remained for two purposes. First, it is needed to decode other MB which uses the current MB as a reference in SVC decoding. Second, the decoded MB is stored into the AVC reference picture buffer for predictions in AVC encoding.

3.4 Drift Compensation

The proposed hybrid-domain transcoding introduces a so-called "drift" problem. In this section, the drift problem in the proposed transcoder is analyzed and solved with compensation techniques.

3.4.1 Drift Analysis

In the transcoding field, "drift" refers to the error accumulation through continuous P-pictures. It is usually caused by the mismatched prediction signals between

encoder and decoder. The prediction errors are accumulated through INTER predictions and become larger and larger. We know that in motion-compensated video coding, the MB is predicted by a similar MB, as shown in Equation (3.1). The resulted residual is then quantized and transmitted. At the decoder side, the dequantized residual signal is added back to the prediction, as shown in Equation (3.2). Here only the lossy quantization operations are shown and other non-lossy operations are omitted. In AVC coding standard, the "$Prediction''$ is kept exactly the same for encoder and decoder by a reconstruction loop at the encoder. Combine the two equations we get Equation (3.3), which shows that the encoding error is only the residual quantization error, denoted as ΔQ_ERR. Residual quantization error does not accumulate through INTER predictions.

$$Residual = Orignal\_MB - Prediction \tag{3.1}$$

$$Decoded\_MB = Q^{-1}[Q(Residual)] + Prediction \tag{3.2}$$

$$\begin{aligned} Orignal\_MB - Decoded\_MB &= Residual \\ -Q^{-1}[Q(Residual)] &= \Delta Q\_ERR \end{aligned} \tag{3.3}$$

In the proposed transcoder, the quantized-transform-domain transcoding insures that the residual is directly copied without quantization. The residual quantization error in quantized transform domain is zero. However, the mode and MV are also reused. The correct prediction MB based on this MV should be the decoded MB in SVC decoding (left "Ref" in Figure 3.8). A mismatch would occur if the prediction MB is coded in the pixel domain, which re-encodes this MB with certain quantization loss. On the other hand, the MBs processed in quantized transform domain are not reconstructed in AVC encoding. Instead, the decoded MBs in SVC decoding are used as AVC encoding references. They could be different and cause a prediction error, denoted as ΔP_ERR. Equations (3.4)-(3.6) show the encoding errors in proposed transcoder. The "$Prediction$" and "$Prediction'$" stand for prediction signals at AVC encoder (transcoder side) and AVC decoder (receiver side) respectively. The errors consist of two parts, i.e. prediction error and quantization error. Our proposed transcoder decreases the quantization error but introduces the

prediction error. Prediction error can be accumulated through INTER predictions, as shown in Figure 3.9. The accumulated prediction errors cause great distortion and even visual artifacts with a large GOP (group of pictures) size.

$$\Delta P\_ERR = Prediction - Prediction^{'} \tag{3.4}$$

$$\Delta Q\_ERR = Residual - Q^{-1}[Q(Residual)] \tag{3.5}$$

$$\begin{aligned} Orignal\_MB - Decoded\_MB = \Delta P\_ERR \\ +\Delta Q\_ERR \end{aligned} \tag{3.6}$$

To solve the drift problem, we deal with 2 aspects. First, the quality of I frame is important since I frame is the first reference frame of a video sequence. A new RDO criterion is proposed for I frame to improve the quality. Second, for the following P frames, we compensate the drift error frame by frame. Details are explained in following subsections.

3.4.2 Drift Compensation in I frame

I frame includes two kinds of modes, that is, IL_Intra and (conventional) INTRA. IL_Intra is processed in pixel domain and INTRA is processed in quantized transform domain. According to the analysis in previous section, the INTRA MBs processed in quantized transform domain is the source of the drift. In this section, a prediction-pixel based RDO (PPRDO) method is proposed for IL_Intra MBs in I frame, to

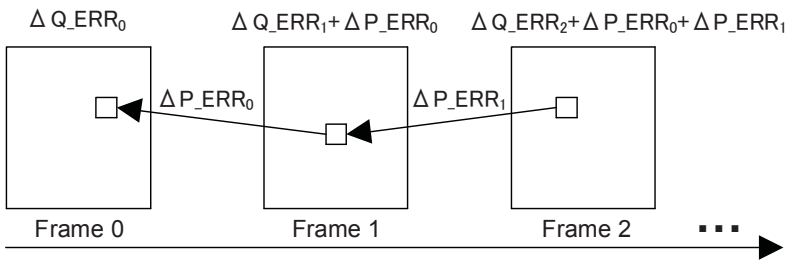


Figure 3.9: Drift problem in proposed transcoder.

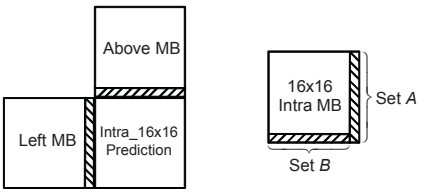


Figure 3.10: Prediction pixels.

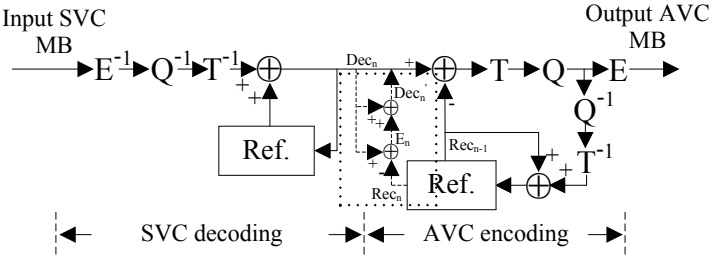


Figure 3.11: Error Compensation.

improve the accuracy of INTRA MBs. Instead of using the conventional RDO metric, we propose a weighted RD (rate-distortion) calculation metric to consider the importance of pixels which might be used for predictions of following INTRA MBs.

The shadowed pixels in the left figure of Figure 3.10 show the neighboring pixels used for INTRA_16x16 prediction, and the right figure shows the 31 pixels in an MB which might be used for predictions of following MBs. The RD cost is calculated as shown in Equations (3.7)-(3.9). The set A in Equation (3.7) and set B in Equation (3.8) represent the indexes of right-most column pixels and the rest 15 bottom-line pixels in the right figure of Figure 3.10, respectively. Equation (3.9) calculates the RD cost by combining the overall distortion with prediction-pixel distortion.

$$A = \{15 + j * 16 | j = 0, 1, ..., 15\} \tag{3.7}$$

$$B = \{16 * 15 + k - 1 | k = 1, ..., 15\} \tag{3.8}$$

$$\begin{aligned} RD\_COST = (1 - \phi) * Dist(all\ pixels) \\ +\phi * s * Dist(A \cup B) + \lambda * R \end{aligned} \tag{3.9}$$

Here $Dist(X)$ calculates the distortion for pixels in set X. ϕ denotes the weight ratio of prediction pixels and the value of ϕ is in the range of [0,1]. Equation (3.9) is equivalent to standard RD calculation when ϕ equals 0, and only the distortion of prediction pixels is considered when ϕ is equal to 1. Via the comprehensive experiments, we set ϕ to a fixed value: 0.5, which generally achieves good results. s is a scaling factor due to the unequal pixel numbers of all_pixel set (16x16) and $A \cup B$

set (31), which is set to 8 ($\approx$ 16x16/31). λ is the Lagrangian parameter and R is the number of bits for current MB, defined by the standard. Though the above statement illustrates the INTRA_16x16 case, it can be easily extended to INTRA_4x4 by updating the prediction-pixel sets and the scaling factor s.

In the proposed transcoder, RD_COST in Equation (3.9) is used as the RD optimization criterion for IL_Intra MBs in I frame. The coding efficiency of these MBs may probably drop a little. But the following conventional INTRA MBs will have higher quality, and hence decrease the drift.

3.4.3 Drift Compensation in P frame

As analyzed in Section 3.4.1, INTER predictions causes drift. In P frame there are three kinds of modes using INTER prediction, i.e., INTER without ILP, INTER with IL_Residual and INTER with IL_Motion. For these MBs an error compensation (EC) method is proposed.

In proposed scheme, MCP (motion-compensated prediction) loop is executed twice. In the first loop, the switch is connected to connector A and Dec_n is the input. $Dec_n^{'}$ is generated during the first loop, and it is used as the input to the second loop by connecting the switch to connector B. The dotted rectangular part in Figure 3.11 illustrates the error compensation method. This method is performed after AVC encoding, i.e., after the reconstruction MB Rec_n is generated. The accumulated error is first calculated and then compensated. Dec_n represents the decoded MB by SVC decoding. The error between Rec_n and Dec_n is calculated by Equation (3.10). The resulted error is then added back to Dec_n, as calculated in Equation (3.11). At last, the calculated signal $Dec_n^{'}$ is motion compensated, transformed, quantized and entropy coded. Note that at this time motion estimation is not performed. The calculated mode and MV by AVC encoder is reused and therefore no heavy burden is introduced.

$$E_n = Dec_n - Rec_n \tag{3.10}$$

$$Dec_n^{'} = 2 * Dec_n - Rec_n \tag{3.11}$$

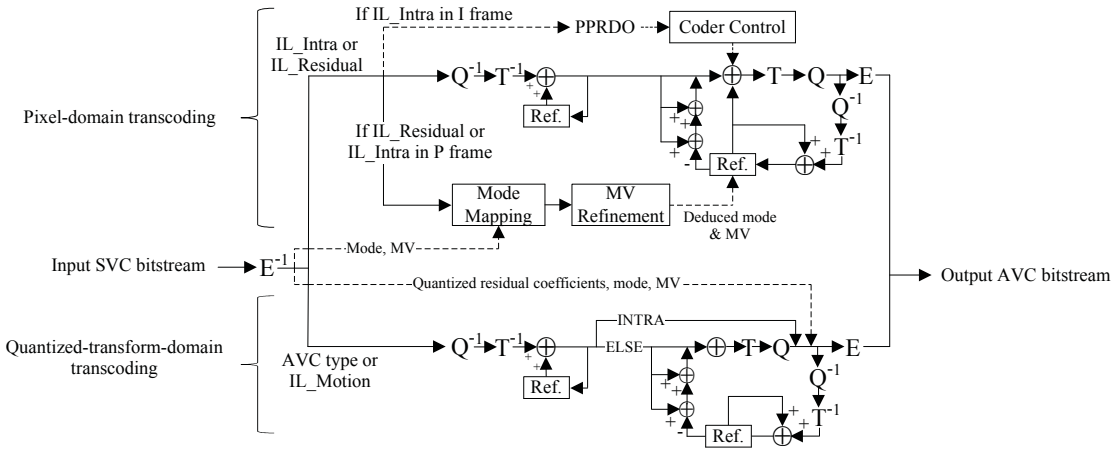


Figure 3.12: Overall transcoding architecture.

3.5 Overall Transcoding Architecture

Combining all the methods described in Section 3.3 and Section 3.4, Figure 3.12 shows the overall proposed transcoding architecture. The upper part shows the pixel-domain transcoding, and the lower part shows the quantized-transform-domain transcoding as described in Section 3.3. The drift compensation methods explained in Section 3.4 are also integrated. In the first loop of the 2-loop procedure, the switches are connected to $A_i(i = 1, 2)$ connectors. B_i connectors are connected at the second loop. The dashed line stands for data transfer, and the PPRDO criterion is considered a kind of data transfer here.

3.6 Simulation Results

In this section, the proposed transcoder is applied to several representative sequences and the results are shown. Software implementation is based on the SVC reference software JSVM (Joint Scalable Video Model). Eight sequences are examined with 2-layer dyadic spatial scalability. Akiyo, panzoom2, football and bus are CIF & QCIF resolutions at 30 fps; flower garden and cheer leaders are VGA & QVGA resolutions at 30 fps; vidyo1 and vidyo3 are 720p & 360p (640360) resolutions at 60 fps. For each sequence 150 frames are tested with the GOP structure of IPPP (only the first

frame is coded as I picture). In our experiments, the QPs (quantization parameters) for input SVC encoder and transcoder are set to the same value, which are selected as 20, 24, 28 and 32. The main parameters are shown in Table 3.2. All experiments are performed on an Intel Core 2 (2.67GHz) computer with 2.0GB RAM.

Table 3.2: Experimental configurations

Parameters	SVC encoding	AVC encoding
Software Version	JSVM 9.18	JSVM 9.18
AVCMode	0	1
FramesToBeEncoded	150	150
SymbolMode	CABAC	CABAC
Enable8x8Transform	disabled	disabled
CodingStructure	IPPP	IPPP
NumRefFrames	1	1
SearchMode	4 (FastSearch)	4 (FastSearch)
SearchRange	16 for CIF/QCIF, 32 for the rest	16 for CIF, 32 for the rest
Quantization Parameter	20/24/28/32	20/24/28/32
Loop Filter	enabled	enabled
NumLayers	2 (scaling factor = 2)	-
Inter-layer Prediction	2 (adaptive)	-
AVCRewriteFlag	0 (disabled)	-

Table 3.3 shows the time saving of reference work [30] and the proposed transcoder, compared with the re-encoding method. Our proposed transcoder achieves averagely 96.4% time reduction relative to the re-encoding method. The speed-up compared with the re-encoding method and [30] are 28 times and 2 times averagely. For CIF sequences, the processing speed is almost real-time based on our pure software implementation. Besides, the upper four sequences in Table 3.3 are videoconferencing-like sequences which are simple and slow. The other four sequences are complex and fast sequences. The proposed transcoder gains more time saving for the upper four sequences than the lower four sequences. It saves averagely 98.1% time for the upper sequences, 3.4% larger than the lower four sequences. Proposed transcoder gains more time reduction for AVC type and IL_Motion coded MBs since they are

transmitted directly in the quantized transform domain. Table 3.4 shows the mode ratio of different test sequences. It shows that for videoconferencing sequences, much more MBs are coded in AVC type or IL_Motion modes which results in more time reduction. Therefore, our proposed transcoder is expected to be suitable for videoconferencing applications. With non-optimized pure software implementation on an desktop PC, only 14 to 312 ms (milliseconds) is needed to transcode one frame. For videoconferencing applications the minimum acceptable delay is usually 200 ms [44], including the network transmission time. With software optimization, more powerful processing unit and certain hardware support, usually over 10 times speed-up is possible. Thus, acceptable delay can be achieved by proposed transcoder for even 720p resolution.

Table 3.3: Time saving comparison. ("360p": 640x360)

Sequence	Re-encoding	Reference [30]		Proposed transcoder		
	time (s)	time (s)	time saving	time (s)	time saving	time/frame (ms)
akiyo (cif+qcif)	163.2	8.6	94.7%	2.1	98.7	14.0
panzoom2 (cif+qcif)	181.7	13.4	92.6%	4.4	97.6	29.3
vidyo1 (720p+"360p")	1626.2	110.6	93.2%	29.3	98.2	195.3
vidyo3 (720p+"360p")	1645.4	85.6	94.8%	33.0	98.0	220
bus (cif+qcif)	197.3	15.6	92.1%	12.2	93.8	81.3
football (cif+qcif)	206.9	19.4	90.6%	11.6	94.4	74.7
flower_garden (vga+qvga)	619.3	55.7	91.0%	21.7	96.5	144.7
cheer_leaders (vga+qvga)	766.7	74.4	90.3%	46.8	93.9	312.0
Average	-	-	92.4%	-	96.4	-

To show the effectiveness of our proposed drift compensation scheme, subjective comparisons are shown in Figure 3.13 & 3.14. The upper figures show the results when drift compensation is off, and lower figures show the results when drift compensation is on. These subjective comparisons show that the drift error is invisible with our drift compensation scheme on.

To illustrate the coding efficiency, RD curves are shown in Figure 3.15. The results of four methods are shown, i.e., direct encoding, proposed transcoder, reference work [30] and the re-encoding method. All the PSNR calculation is using the original sequence (at the sender side) as the calculation reference. The "direct encoding" method means to encode the original sequence directly, and then the encoded bitstream is decoded by an SVC decoder. The scenario for this method is that the original sequence is encoded at the sender side, and then the encoded bitstream is transmitted to the receiver client without transcoding. The receiver side is supposed

Table 3.4: MB mode type ratio.

Sequence	Group I (%)	Group II (%)
akiyo	89.2	10.8
panzoom2	83.6	16.4
vidyo1	91.1	8.9
vidyo3	90.3	9.6
bus	57.5	42.5
football	71.7	28.3
flower_garden	65.0	35.0
cheer_leaders	49.2	50.8

Group I: AVC type & IL_motion, Group II: IL_Residual & IL_Intra

to support SVC decoding. "Direct encoding" method is the ideal case which would not happen if the client doesn't support SVC decoding. The other three methods are real solutions for SVC-to-AVC spatial transcoding. Figure 3.15 shows the comparison. The direct encoding method achieves the highest coding efficiency than the other three methods, since it is the ideal case. The re-encoding method achieves the worst coding efficiency. The gap between ideal direct encoding method and the re-encoding method is about 1-2 dB. Reference work [30] is slightly better than the re-encoding method, about 0.1-0.2 dB higher. The proposed method additionally gains about 0.1-0.5 dB than the referece work [30].

As explained in Section 3.4, the trade-off in quantized-transform-domain transcoding is between the quantization error and the prediction error. Our method eliminates the quantization error for quantized-transform-domain MBs and introduces the prediction error. In an extreme case, if the input SVC bitstream consists of all pixel-domain MBs (i.e. no quantized-transform-domain processing), the resulted coding efficiency should be a little worst (due to fast mode decision) than the re-encoding method. In another extreme case, if the input SVC bitstream consists of all quantized-transform-domain MBs, the resulted coding efficiency should be exactly the same the "direct encoding". That is to say, the proposed transcoder favors quantized-transform-domain MBs (AVC type or IL_Motion). As shown in Figure 3.15, our performance for the eight sequences is better than the re-encoding method, which means in common cases the coding efficiency gain by eliminating quantization error can overwhelm the loss by prediction error.

Fast mode decision (FMD) methods for H.264/AVC could be an alternative

(a) frame 3, DC off (b) frame 10, DC off (c) frame 30, DC off

(d) frame 3, DC on (e) frame 10, DC on (f) frame 30, DC on

Figure 3.13: Subjective comparisons. (akiyo sequence, DC: drift compensation)

approach for fast SVC-to-AVC transcoding. However, they generally decrease the coding efficiency and do not gain as much time reduction as our work, because the input bitstream information is not explored. Therefore, comparisons are not shown here. Representative FMD results can be found in [42, 43].

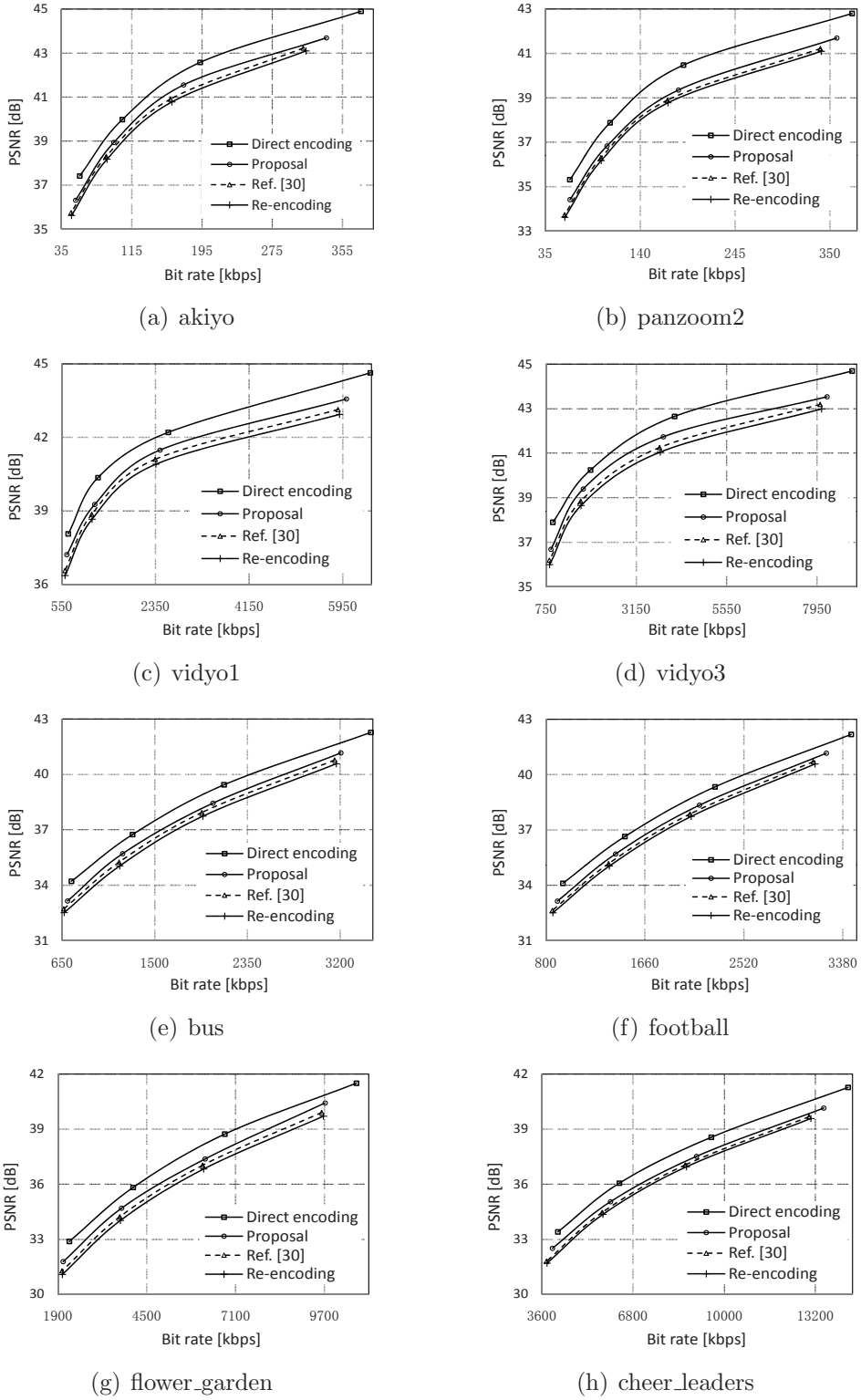
3.7 Conclusions

This chapter proposes a low-complexity SVC-to-AVC spatial transcoder based on a hybrid-domain architecture. The input bitstream MBs are transcoded in different domains according to their mode types. Transcoding approaches in the pixel domain and the quantized transform domain are explained. A drift problem is introduced by the proposed hybrid-domain transcoding architecture. The cause of the drift is analyzed and the drift problem is solved by compensation techniques. Experiments show that the proposed transcoder can speed up 28 times the re-encoding method with even higher coding efficiency. The proposed transcoder is expected to play an importance role in a hybrid videoconferencing application.

(a) frame 5, DC off (b) frame 15, DC off (c) frame 40, DC off

(d) frame 5, DC on (e) frame 15, DC on (f) frame 40, DC on

Figure 3.14: Subjective comparisons. (bus sequence, DC: drift compensation)



(a) akiyo (b) panzoom2

(c) vidyo1 (d) vidyo3

(e) bus (f) football

(g) flower_garden (h) cheer_leaders

Figure 3.15: RD curves comparison.

3. DRIFT COMPENSATED HYBRID-DOMAIN SVC TO AVC SPATIAL TRANSCODING

Chapter 4

Drift constrained frequency-domain SVC to AVC quality homogeneous transcoding

4.1 Introduction

For single layer transcoding, many works have been done. Literatures [10, 12, 18, 19, 29, 31, 45, 46] are based on the motion reuse (MR) transcoding architecture, as shown in Figure 4.1. The modes and motion vectors (MVs) of input bitstream are utilized and refined to accelerate the motion estimation (ME) process of encoder. Motion reuse for bit-rate reduction transcoding is examined in [12, 18]. In [18, 19, 29], mode and MV mapping strategies are proposed for resolution reduction transcoding in the context of different coding standards. The authors of [31] propose an MV refinement scheme for frame-rate reduction transcoding. Literatures [10, 45, 46] further analyze motion reuse for heterogeneous transcoding (format conversion). All these works are based on the motion reuse architecture, which only accelerates the ME part of the re-encoding method. The speed-up is restricted by existence of all other components such as DCT transforms. In [47] the authors merge the decoder and encoder MCP (motion-compensated prediction) loops under the assumption that motion data are the same for decoder and encoder, referred as Single-Loop (SL) transcoding architecture (Figure 4.2). This architecture is free of

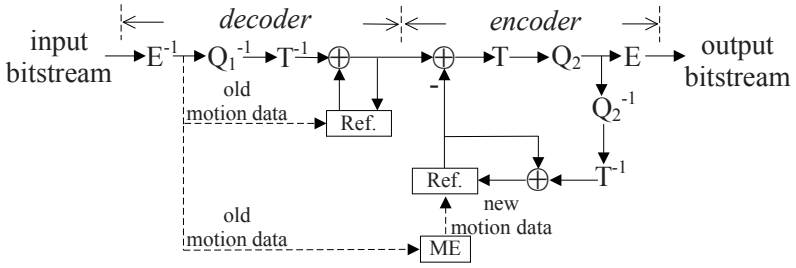


Figure 4.1: Motion reuse (MR) transcoding architecture. (E: entropy coding, $Q_i(i = 1, 2)$: quantization, T: DCT transform, $Ref.$: reference picture buffer, ME: motion estimation, superscript "-1": inverse process.)

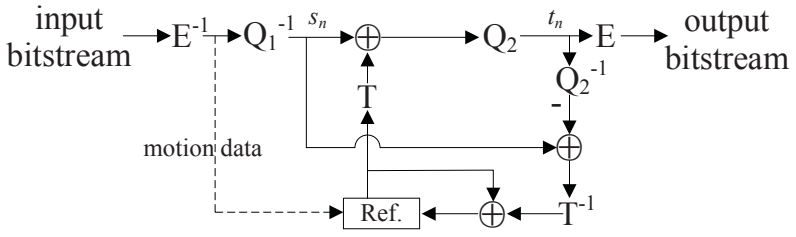


Figure 4.2: Single-loop (SL) transcoding architecture.

drift (error propagation), and one inverse transform and one picture buffer are reduced. [33, 48, 49, 50] are extensions based on SL architecture. A further accelerated transcoding architecture is proposed by [51], namely Simplified Single-Loop (SSL) transcoding architecture (Figure 4.3). Transforms are totally removed and motion compensation is directly performed on DCT transform coefficients (denoted as MC-DCT). MC-DCT needs floating-point matrix multiplication which is quite costly and diminishes the speed gain. Matrix approximation is possible for acceleration but introduces drift. Another common known architecture is the open-loop (OL) transcoding architecture (Figure 4.4), for which the drift problem is quite severe. The later three architectures (SL,SSL,OL) are often referred as frequency-domain (or DCT-domain, transform-domain) transcoding methods since there is no transform operations on the main route. Relatively, the re-encoding and MR architectures are usually mentioned as pixel-domain transcoding methods.

Transcoding between SVC and AVC has not been thoroughly investigated yet. In

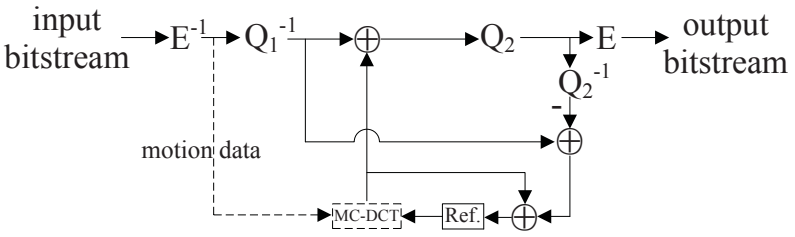


Figure 4.3: Simplified single-loop (SSL) transcoding architecture.

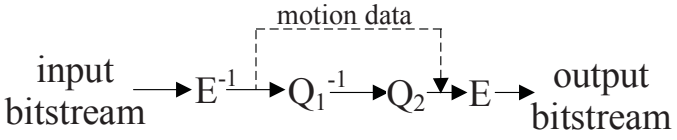


Figure 4.4: Open-loop (OL) transcoding architecture.

[23] the authors propose a fast AVC-to-SVC quality transcoding method based on SL and OL architecture and achieve significant time reduction compared with the full re-encoding method. MR architecture based solutions for AVC-to-SVC temporal transcoding are described in [37, 38]. Motion data adaptation and refinement for AVC-to-SVC spatial transcoding is proposed in [39, 40] based on MR architecture. The authors of [30] propose a fast mode decision method for SVC-to-AVC spatial transcoding, also based on MR architecture.

In this chapter, an ultra-low-delay SVC-to-AVC MGS transcoding architecture is proposed. Significant speed-up is achieved by proposed three fast transcoding methods for MBs (macroblocks) with different coding modes in non-KEY pictures. KEY pictures are transcoded with drift-free single-layer architecture by reusing the base layer motion data. Thus drift will not propagate beyond KEY pictures.

The rest of this chapter is organized as follows. Section 4.2 describes the MGS scalability in SVC. Section 4.3 shows the overall architecture for proposed frequency-domain transcoding. Non-KEY picture and KEY picture transcoding are explained in Section 4.4 and Section 4.5 respectively. Section 4.6 gives the simulation results and conclusions are drawn in Section 4.7.

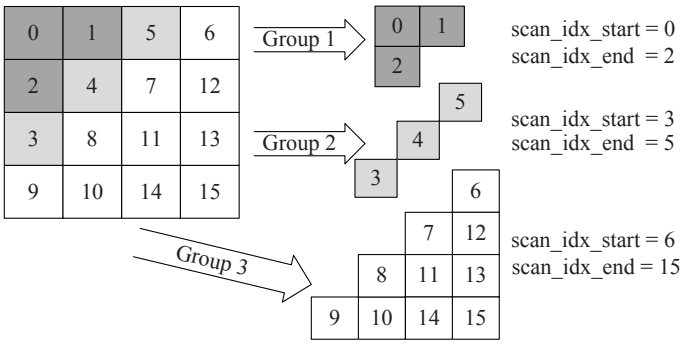


Figure 4.5: Coefficients partitioning.

4.2 MGS scalability in SVC

MGS includes two main features, i.e., coefficients partitioning and KEY picture concept. Coefficients partitioning allows to distribute the transform coefficients among several NALUs (Network Abstraction Layer Units). Figure 4.5 shows an example of coefficient partitioning. The left 4x4 matrix represents transform coefficients after 4x4 transform, and the numbers are zigzag scanning indices. These coefficients are divided into three groups, and each group corresponds to one NALU containing coefficients from *scan_idx_start* to *scan_idx_end*. Up to 16 NALUs are possible, and by discarding several of them flexible packet-based quality scalability is provided. In transcoding, it is very easy to parse the input NALUs and reassemble the transform coefficients.

Another feature is the KEY picture concept, which is based on hierarchical prediction structure. Figure 4.6 shows a hierarchical-P prediction structure for MGS encoding defined by standard (highly delayed B pictures are rarely used in videoconferencing [44]). TID represents the temporal layer ID. Grey-colored pictures are KEY pictures ($TID = 0$), which only use other KEY pictures for prediction. Base layer KEY picture is predicted from previous base layer KEY picture and MGS layer KEY picture is predicted from current base layer picture. Non-KEY picture ($TID > 0$) is predicted by the MGS layer of nearest previous picture with smaller TID. Such prediction structure can constrain the drift due to discarded packets within a GOP (Group Of Pictures).

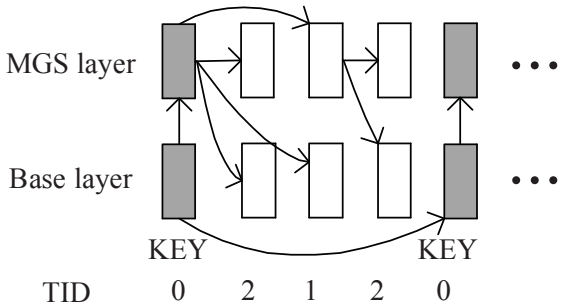


Figure 4.6: Hierarchical-P with KEY pictures. (GOPSize = 4)

4.3 Overall proposed transcoding method

4.3.1 Analysis of coding modes in SVC

SVC introduces inter-layer prediction (ILP) schemes while inheriting the AVC coding modes (INTER and INTRA). Three kinds of ILPs are introduced to explore the correlation between base layer (BL) and enhancement layer (EL) in SVC encoding. They are inter-layer residual, inter-layer intra and inter-layer motion predictions. These inter-layer predictions will be denoted as IL_Residual, IL_Intra and IL_Motion predictions hereafter.

IL_Intra prediction predicts the original enhancement layer input picture using the upsampled base layer reconstructed picture. IL_Intra prediction only occurs when the co-located position in base layer is coded with constrained INTRA prediction (see [53]). The base layer reconstructed picture is upsampled and used as the predictor for enhancement layer input picture. The resulted residual is transmitted after transform, quantization and entropy coding. IL_Residual prediction tries to predict the residual data generated by INTER prediction. The first residual generated by normal INTER prediction is predicted by the upsampled base layer reconstructed residual signal, and the resulted second residual is transmitted after transform, quantization and entropy coding. IL_Motion prediction tries to reduce the size of motion data for INTER coded MBs. The upsampled base layer mode and MV information is utilized to predict the enhancement layer motion data. More descriptions about ILPs can be found in [1].

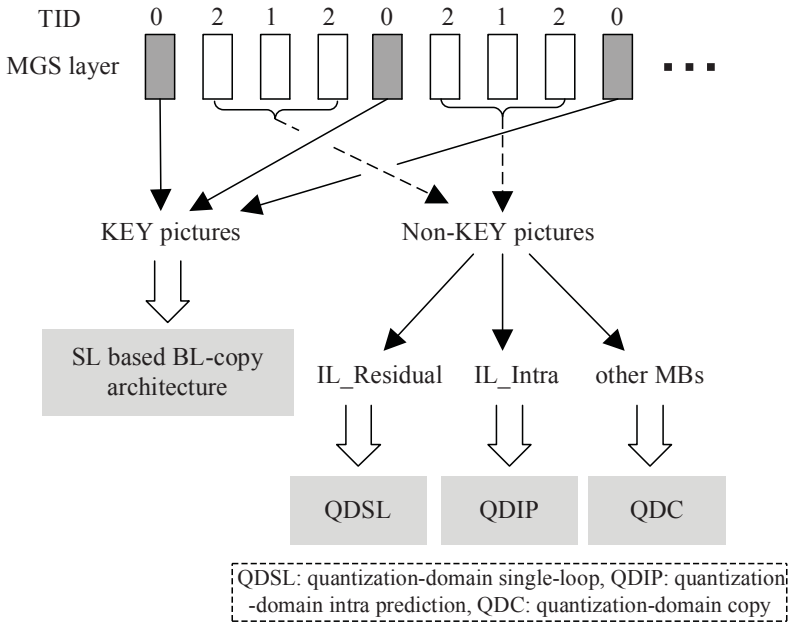


Figure 4.7: Proposed transcoding method.

The IL_Intra prediction is totally independent from the AVC INTRA or INTER modes, while the IL_Residual and IL_Motion predictions are additional refinements based on AVC INTER mode. Thus the coding modes in SVC are shown in Table 1. It is also possible that IL_Residual and IL_Motion both exist for an INTER MB. In such case, it is considered as IL_Residual. For short, "INTER with IL_Residual" and "INTER with IL_Motion" will be denoted as IL_Residual and IL_Motion hereafter.

Table 4.1: Coding modes in SVC

Inherited modes	Newly introduced modes
INTRA INTER without ILP	IL_Intra INTER with IL_Residual INTER with IL_Motion

4.3.2 Proposed transcoding method

Figure 4.7 shows the overall proposed SVC-to-AVC MGS transcoding method. The input MGS layer frames are first divided into KEY pictures and non-KEY pictures. Non-KEY pictures are further divided according to MB types and transcoded in "quantization domain", which will be described in Section 4.4 and can be considered as a special case of "frequency domain". IL_Residual MBs in non-KEY pictures will be transcoded by a quantization-domain single-loop architecture (Section 4.4.1). IL_Intra MBs are transcoded with a quantization-domain intra prediction architecture (Section 4.4.2). And finally, other type MBs are transcoded by quantization-domain copy method (Section 4.3). KEY pictures are transcoded by the drift-free single-loop based BL-copy architecture (Section 4.5). Details of proposed schemes are explained in Section 4.4 and 4.5.

4.4 Non-KEY picture transcoding

The hierarchical prediction structure is maintained for non-KEY picture transcoding, and predictions will not beyond GOP boundaries. MBs in non-KEY pictures are transcoded differently according to their mode types.

4.4.1 Quantization-domain single-loop transcoding for IL_Residual MBs

In this subsection, a special frequency-domain single-loop transcoding architecture is derived for IL_Residual transcoding. Let's start from the drift-free SL architecture as shown in Figure 4.2. Two signals s_n and t_n corresponding to the input and output signals of the MCP loop are shown. The relation between them is shown in (4.1).

$$t_n = Q_2(s_n + T(MC(T^{-1}(s_{n-1} - Q_2^{-1}(t_{n-1})))))) \tag{4.1}$$

Here the $MC(.)$ represents the motion compensation operation corresponding to the bottom addition symbol in Figure 4.2. By assuming a distributive property for quantization (which is actually not true; same implication for following "assuming"), Equation (4.1) is modified to Equation (4.2).

$$t_n = Q_2(s_n) + Q_2(T(MC(T^{-1}(s_{n-1} - Q_2^{-1}(t_{n-1})))))) \tag{4.2}$$

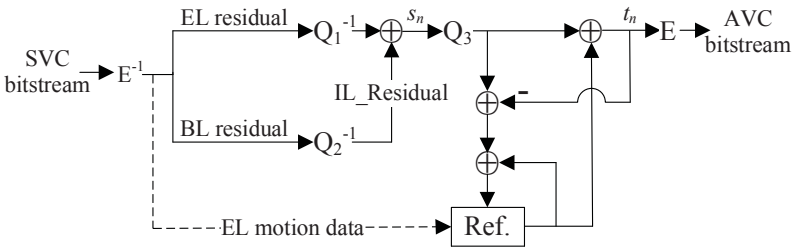


Figure 4.8: Quantization-domain single-loop (QDSL) transcoding architecture.

Then by assuming a commutative property between DCT transform and motion compensation, Equation (4.2) is further modified to Equation (4.3), also based on the fact that DCT transform is a lossless operation.

$$t_n = Q_2(s_n) + Q_2(MC(s_{n-1} - Q_2^{-1}(t_{n-1}))) \tag{4.3}$$

By assuming the commutative property between quantization and motion compensation, Equation (4.3) is changed to Equation (4.4).

$$t_n = Q_2(s_n) + MC(Q_2(s_{n-1} - Q_2^{-1}(t_{n-1}))) \tag{4.4}$$

Finally, by applying the previously assumed distributive property of quantization operation, Equation (4.5) is obtained.

$$t_n = Q_2(s_n) + MC(Q_2(s_{n-1}) - t_{n-1}) \tag{4.5}$$

Note that the second term of Equation (4.5) implies a motion compensation on quantized transform coefficients, while in SL and SSL architectures motion compensation is performed on pixel values and unquantized transform coefficients. The first term is easily obtained by quantizing the input signal s_n. A corresponding architecture based on Equation (4.5) for IL_Residual transcoding is shown in Figure 4.8. This architecture is named "quantization-domain" single-loop (QDSL) transcoding to emphasize the difference from existing frequency-domain architectures. Proposed QDSL architecture eliminates DCT transforms and further reduces one inverse quantization component comparing with the SSL architecture.

Figure 4.9 illustrates the motion compensation method in quantization domain. MB_{cur} is the current MB to be coded and MB_{ref} is the reference MB. Dotted lines

are aligned MB boundaries. If the motion vector points to an intersection of dotted lines, e.g. the intersection near the word "MB_1", then the prediction signal can be easily decided as the quantized transform coefficients of MB_1. When the MV does not point to an intersection, the prediction is composed by weighted sum of several related MBs. $MB_i(i = 1..4)$ are MBs overrode by MB_{ref}. The right sub-figure in Figure 4.9 enlarges the overrode area consisting of 4 regions. The areas of these regions are denoted by $Area_i(i = 1..4)$. Let $Coeff(MB_i)$ be the quantized coefficients matrix of MB_i. The prediction signal is generated by Equation (4.6). Partition or sub-partition motion compensation is done in a similar way.

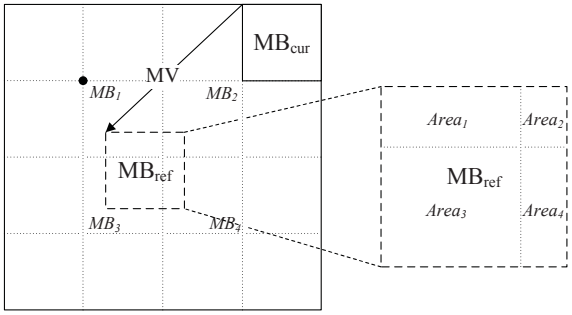


Figure 4.9: Quantization-domain MC.

$$PRED = \frac{\sum_{i=1}^{4}[Area_i \times Coeff(MB_i)]}{Area_1 + Area_2 + Area_3 + Area_4} \tag{4.6}$$

The false assumptions and MC approximations introduce errors which might propagate through inter predictions, namely "drift" in transcoding field. However, this drift problem is diminished by drift-free transcoding of KEY pictures (at GOP boundaries). Frames within a GOP do not use frame outside this GOP as a reference, and thus error propagation will be constrained within GOPs. Details of KEY picture transcoding are explained in Section 4.5.

4.4.2 Quantization-domain intra prediction for IL_Intra MBs

Similar to the derivation in previous subsection, Equation (4.7) can be obtained for Figure 4.2 in the context of intra prediction. Here the $I\_PRED(.)$ is the intra

prediction operation. The second term implies quantization-domain intra prediction (QDIP). Figure 4.10 gives the corresponding transcoding architecture. Since IL_Intra does not transmit intra prediction mode information in EL, the BL motion data is reused.

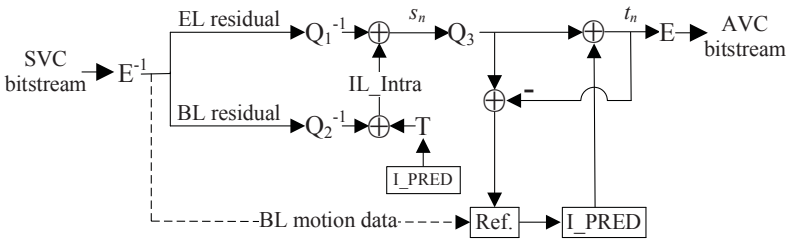


Figure 4.10: Quantization-domain intra prediction (QDIP) transcoding architecture.

$$t_n = Q_2(s_n) + I\_PRED(Q_2(s_{n-1}) - t_{n-1}) \tag{4.7}$$

In the pixel domain, neighboring pixels are used to form an intra prediction. But in quantization-domain where coefficients are concentrated in upper-left corner, extracting corresponding coefficients for those neighboring pixels is difficult. In proposed transcoder, QDIP is accomplished by approximating the prediction signal using neighboring 4x4 blocks. Figure 4.11 shows the intra 16x16 prediction in quantization domain. In the top-left sub-figure, $B_i(i = 1..8)$ are the neighboring 4x4 blocks containing quantized coefficients. When intra 16x16 prediction mode is vertical or horizontal, the prediction is formed by extending neighboring blocks along the prediction direction, shown as the bottom-left and top-right sub-figures. For other modes (DC/plane), the prediction shown as bottom-right sub-figure is formed by averaging the vertical and horizontal predictions. $B_{ij}(i = 1..4, j = 5..8)$ is defined in Equation (4.8).

$$B_{ij} = (B_i + B_j)/2 \tag{4.8}$$

Intra 4x4 predictions are processed as Figure 4.13. B_{cur} is the current 4x4 block to be predicted and $B_i(i = 1..4)$ are neighboring blocks. $(X, A..L)$ are positions used for intra prediction in pixel domain. For mode 0, 1 or 8, the pixels used for intra

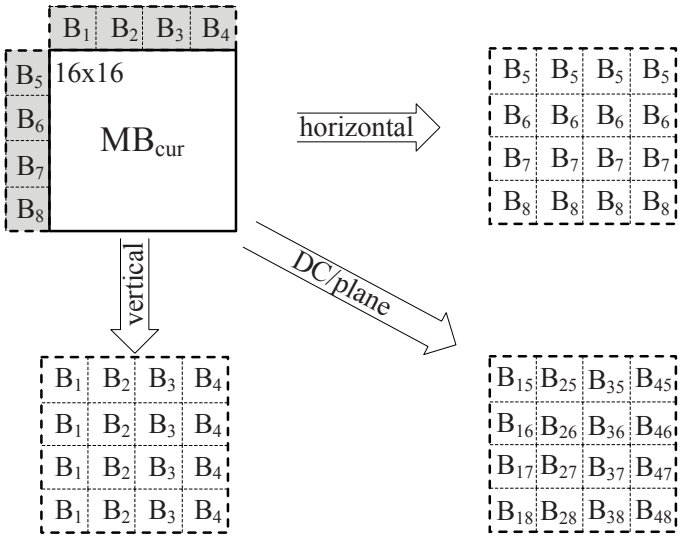


Figure 4.11: Intra 16x16 prediction.

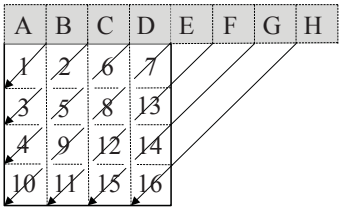


Figure 4.12: Intra 4x4 mode 3 prediction.

4x4 prediction belong to one particular neighboring block. This block is selected to approximate the prediction signal. For mode 2 which uses the mean value of $(A..B, I..L)$, the average of B_2 and B_4 is selected as the prediction. For the rest modes, a weighted average of neighboring blocks is formed as the prediction. The weight depends on how much one block contributes to the prediction signal. For example, Figure 4.12 shows the mode 3 prediction (each square corresponds to one pixel position), where there are totally 7 predictor values. Table 4.2 shows the corresponding positions using these predictors. Function $Con(.)$ represents the contribution of block B_i to each predictor. It equals the sum of weights of B_i pixels used for the predictor, multiplied by the number of blocks using this predictor. For example, in mode 3 the predictor $(C + 2D + E)/4$ uses C, D in B_2 and their weights

Table 4.2: Mode 3-7 predictions.

mode 3	Pixel no.	1	2,3	4,5,6	7,8,9,10	11,12,13	14,15	16	-	-	-
	Predictor	$\frac{A+2B+C}{4}$	$\frac{B+2C+D}{4}$	$\frac{C+2D+E}{4}$	$\frac{D+2E+F}{4}$	$\frac{E+2F+G}{4}$	$\frac{F+2G+H}{4}$	$\frac{G+2H+H}{4}$	-	-	-
	Con(B_2)	1	2	9/4	1	0	0	0	-	-	-
	Con(B_3)	0	0	3/4	3	3	2	1	-	-	-
mode 4	Pixel no.	7	6,13	2,8,14	1,5,12,16	3,9,15	4,11	10	-	-	-
	Predictor	$\frac{B+2C+D}{4}$	$\frac{A+2B+C}{4}$	$\frac{X+2A+B}{4}$	$\frac{A+2X+I}{4}$	$\frac{X+2I+J}{4}$	$\frac{I+2J+K}{4}$	$\frac{J+2K+L}{4}$	-	-	-
	Con(B_1)	0	0	3/4	2	3/4	0	0	-	-	-
	Con(B_2)	1	2	9/4	1	0	0	0	-	-	-
	Con(B_4)	0	0	0	1	9/4	2	1	-	-	-
mode 5	Pixel no.	1,9	2,12	6,14	7	3,11	5,15	8,16	13	4	10
	Predictor	$\frac{X+A}{2}$	$\frac{A+B}{2}$	$\frac{B+C}{2}$	$\frac{C+D}{2}$	$\frac{I+2X+A}{4}$	$\frac{X+2A+B}{4}$	$\frac{A+2B+C}{4}$	$\frac{B+2C+D}{4}$	$\frac{X+2I+J}{4}$	$\frac{I+2J+K}{4}$
	Con(B_1)	1	0	0	0	1	1/2	0	0	1/4	0
	Con(B_2)	1	2	2	1	1/2	3/2	2	1	0	0
	Con(B_4)	0	0	0	0	1/2	0	0	0	3/4	1
mode 6	Pixel no.	1,8	2,13	6	7	3,12	5,14	4,15	9,16	10	11
	Predictor	$\frac{X+I}{2}$	$\frac{I+2X+A}{4}$	$\frac{X+2A+B}{4}$	$\frac{A+2B+C}{4}$	$\frac{I+J}{2}$	$\frac{X+2I+J}{4}$	$\frac{J+K}{2}$	$\frac{I+2J+K}{4}$	$\frac{K+L}{2}$	$\frac{J+2K+L}{4}$
	Con(B_1)	1	1	1/4	0	0	1/2	0	0	0	0
	Con(B_2)	0	1/2	3/4	1	0	0	0	0	0	0
	Con(B_4)	1	1/2	0	0	2	3/2	2	2	1	1
mode 7	Pixel no.	1	2,4	6,9	7,12	14	3	5,10	8,11	13,15	16
	Predictor	$\frac{A+B}{2}$	$\frac{B+C}{2}$	$\frac{C+D}{2}$	$\frac{D+E}{2}$	$\frac{E+F}{2}$	$\frac{A+2B+C}{4}$	$\frac{B+2C+D}{4}$	$\frac{C+2D+E}{4}$	$\frac{D+2E+F}{4}$	$\frac{E+2F+G}{4}$
	Con(B_2)	1	2	2	1	0	1	2	3/2	1/2	0
	Con(B_3)	0	0	0	1	1	0	0	1/2	3/2	1

are 1/4 & 2/4. Number of blocks using this predictor is 3 and thus the contribution of B_2 for this predictor is $(1/4 + 2/4) \times 3 = 9/4$. The total contributions for B_2 and B_3 are 25/4 & 39/4, and thus the prediction is formed by $(25/4 \times B_2 + 39/4 \times B_3)/(25/4 + 39/4) = (25 \times B_2 + 39 \times B_3)/64$. The predictions for other modes are calculated similarly. The contributions for modes 3-7 are shown in Table 4.2, and the final results are shown in Figure 4.13.

The proposed intra prediction approximations will introduce errors. However, as explained in Section 4.1, these errors can be constrained within GOPs by drift-free transcoding of KEY pictures (Section 4.5).

4.4.3 Quantization-domain copy for other MBs

For MBs with other coding modes (INTRA/INTER without ILP/IL_Motion), a quantization-domain copy (QDC) method is applied. Different from IL_Residual or IL_Intra, for these modes the residual generation process in SVC is identical to AVC encoding. The input MB is entropy decoded only, and the quantized residual coefficients along with the motion data are copied into AVC bitstream directly. In case of IL_Motion, the motion data need to be reconstructed first, which is very easy. Note that no re-quantization is performed here, and thus the residual is kept accurate.

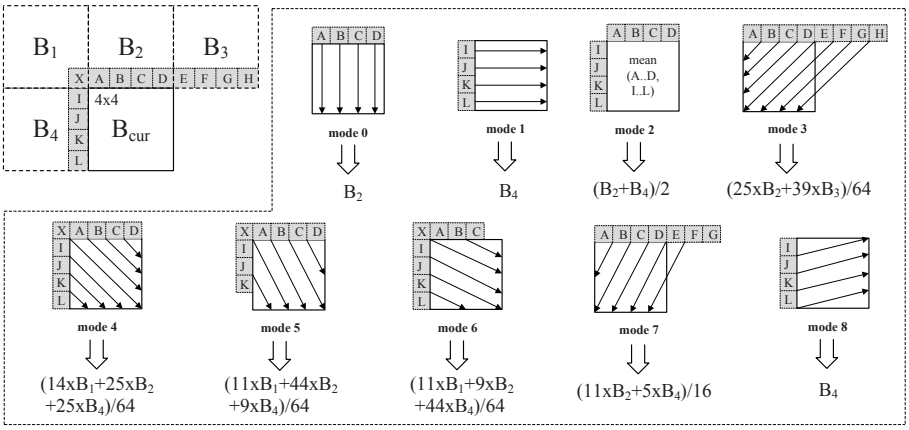


Figure 4.13: Intra 4x4 prediction.

4.5 KEY picture transcoding

To constrain the propagation of errors caused by the false assumptions in QDSL deduction and MC/intra prediction approximations , KEY pictures are transcoded based on drift-free single-loop architecture. In hierarchical-P prediction structure, MGS layer KEY pictures are predicted by base layer KEY pictures. However, after transcoding the base layer pictures will be all discarded. Thus motion re-estimation is necessary for these KEY pictures since a new reference picture must be selected. In proposed transcoder, the previous MGS layer KEY picture is selected as the reference picture (Figure 4.14). Two merits can be obtained by such selection. Firstly, the prediction structure will remain as a single-layer hierarchical-P structure, by which the temporal scalability is kept. Secondly, due to the high correlation between MGS layer and base layer, the motion data from base layer prediction can be reused for MGS layer.

The above discussion solves INTER MBs in MGS layer (INTER, IL_Residual, IL_Motion). For IL_Intra MB, the base layer prediction mode of current frame is reused since there is no mode information transmitted in MGS layer. For INTRA MB, the MGS layer mode is directly reused. The proposed single-loop based BL-copy architecture is shown in Figure 4.15. The EL mode is checked to decide which motion data to be utilized. Note that EL does not need inverse transform or motion compensation. Partial decoding is also performed for BL decoding. Only MBs used

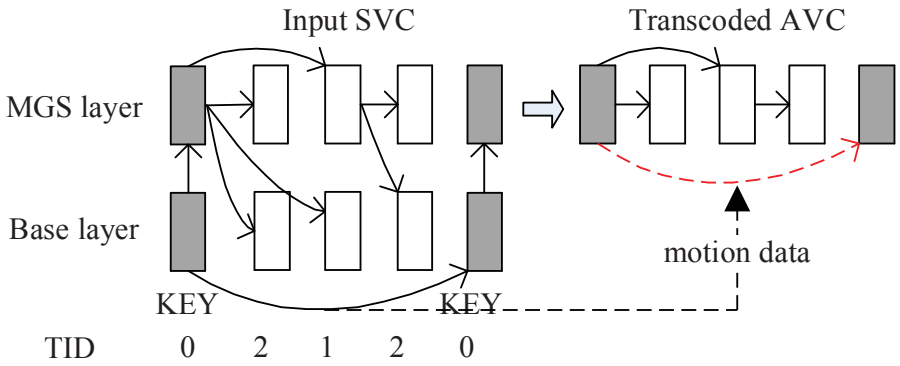


Figure 4.14: Base layer copy.

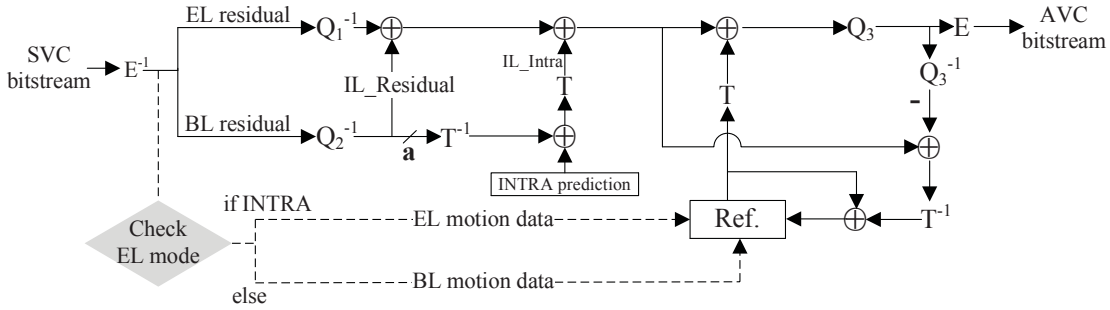


Figure 4.15: Single-loop based BL-copy architecture.

for IL_Residual and IL_Intra predictions will be decoded. Besides, decoding of MBs used for IL_Residual prediction stops at position **a** since BL reconstruction signal is not needed.

4.6 Simulation results

In this section, the proposed transcoder is applied to several publicly available sequences and the results are shown. Software implementation is based on the SVC reference software JSVM (Joint Scalable Video Model). 12 sequences are encoded with 3-layer MGS scalability, and then the encoded bitstreams are transcoded into AVC format with highest MGS layer quality. *Akiyo*, *panzoom*2, *football* and *bus* are CIF (352x288) sequences. *cheer_leaders* and *flower_garden* are VGA (640x480) se-

quences. *vidyo*1, *vidyo*3, *vidyo*4, *FourPeople*, *parkrun* and *SlideEditing* are 720p (1280x720) sequences. For each sequence 150 frames are tested. In our experiments, the QPs (quantization parameters) for input MGS bitstream encoding (SVC encoder) is set as 28 & 24 for base layer & MGS layer, and transcoder (AVC encoder) QPs are selected as 20, 24, 28 and 32. The main configuration parameters are shown in Table 4.3. All experiments are performed on an Intel Core 2 (2.67GHz) computer with 2.0GB RAM.

Table 4.3: Experimental configurations

Parameters	SVC encoding	AVC encoding
Software Version	JSVM 9.18	JSVM 9.18
AVCMode	0	1
FramesToBeEncoded	150	150
SymbolMode	CABAC	CABAC
Enable8x8Transform	disabled	disabled
CodingStructure	Hierarchical-P	Hierarchical-P
NumRefFrames	1	1
SearchMode	4 (FastSearch)	4 (FastSearch)
SearchRange	16 for CIF/VGA, 32 for 720p	16 for CIF/VGA 32 for 720p
Quantization Parameter	28 for BL, 24 for EL	20/24/28/32
Loop Filter	enabled	enabled
DisableBSlices	1 (B-slice disabled)	1 (B-slice disabled)
GOPSize	4	4
MGSVectorX(X=0,1,2)	3,3,10	-
InterLayerPred	2 (adaptive)	-
AVCRewriteFlag	0 (disabled)	-

Besides the proposed method, 4 methods are used for comparison - re-encoding, motion reuse (MR), single loop (SL) and open loop (OL). The implementation of re-encoding and OL methods are straightforward. The MR method is implemented based on the mode mapping schemes in reference [40]. The SL method is implemented based on reference [47].

Table 4.4 shows the computational time comparisons. Three criteria are shown, i.e., total transcoding time (C1), time saving (C2) and time per frame (C3). The re-encoding method is selected as the comparison base, and the time saving for other methods is calculated by comparing with re-encoding method. The bolded figures in Table 4.4 show the time saving for our proposal relative to the re-encoding method, as well as the processing time per frame. Time saving ranges from 96.3% up to 98.4%, and the average time saving is 97.4% corresponding to a 38.5 times speed-up. Processing time per frame ranges from 490 ms down to 21 ms. Comparing with MR and SL methods, proposed transcoder achieves averagely 11.2 and 4.8 times speed-up respectively. OL method is about 3.3 times faster than proposed method. Besides, the upper six sequences in Table 4.4 are videoconferencing-like sequences with simple background, slow movement or PTZ (panning, tilting & zooming) camera motions. The other six sequences are complex or fast sequences which rarely appear in videoconferencing. The proposed transcoder gains more time saving for the videoconferencing sequences than the other sequences. It saves averagely 98.1% time for the upper 6 sequences and 96.7% time for the lower 6 sequences, i.e., proposed transcoder is 1.74 times faster for videoconferencing sequences than the other sequences. The reason is that videoconferencing sequences usually contain much AVC INTER coded MBs (without ILP) which will be copied directly (Section 4.4.3). With non-optimized pure software implementation on an desktop PC, down to 21 ms and 247 ms delay can be achieved for videoconferencing sequences with CIF and 720p resolutions respectively. For videoconferencing applications the minimum acceptable delay is usually 200 ms [44], including the network transmission time. With software optimization, more powerful processing unit and certain hardware support, negligible single-digit delay can be achieved by proposed transcoder for even 720p resolution.

Table 4.5 shows the BD (Bjøntegaard Delta [52]) performance for MR, SL, OL & proposed method, all comparing with the re-encoding method. The two criteria C4 & C5 are BDBR (BD bit-rate) & BDPSNR (BD peak signal-to-noise ratio) respectively. The bolded figures in Table 4.5 show the coding efficiency performance of proposed transcoder, resulting averagely 18.5% BDBR increase and 1.02 dB BDPSNR loss. This amount of coding efficiency loss is usually acceptable for transcoding applications and does not introduce disturbing visual artifacts. This performance

Table 4.4: Computational time comparisons.

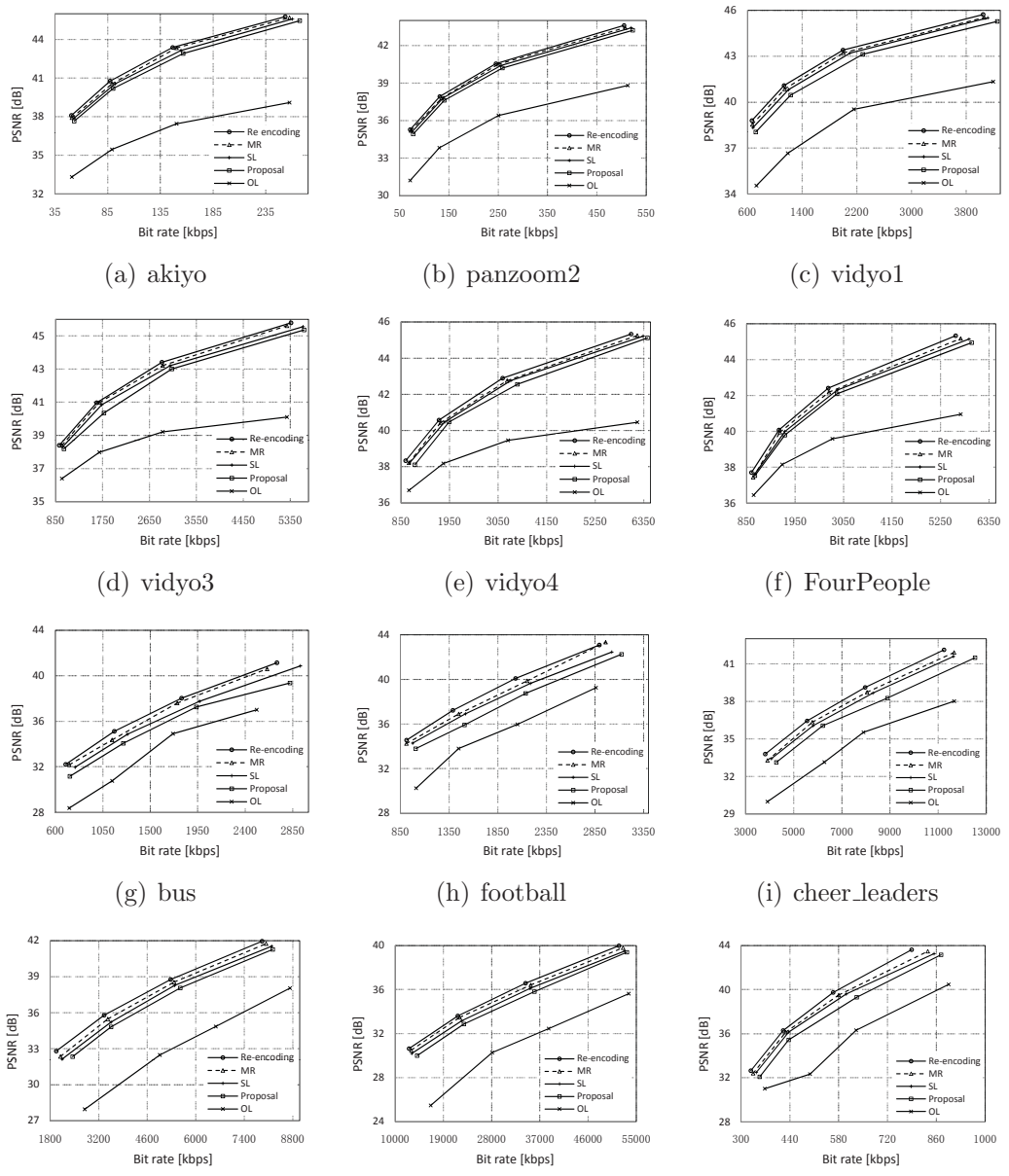
Sequence	Re-encoding	Motion Reuse			Single Loop			Open Loop			Proposal		
	C1	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
akiyo	194.1	52.8	72.8	352	23.5	87.9	157	1.5	99.2	10	3.1	**98.4**	**21**
panzoom2	202.5	54.3	73.2	362	23.8	88.2	159	1.2	99.4	8	3.6	**98.2**	**24**
vidyo1	1942.4	563.9	71.0	3759	242.3	87.5	1615	14.2	99.3	94	3.9	**97.9**	**266**
vidyo3	1957.6	576.1	70.6	3840	244.4	87.5	1629	13.7	99.3	91	41.0	**97.9**	**273**
vidyo4	1948.1	570.0	70.7	3800	243.9	87.4	1626	14.7	99.2	98	42.2	**97.8**	**281**
FourPeople	1936.3	567.7	70.7	3784	247.7	87.2	1651	15.8	99.2	105	37.1	**98.1**	**247**
bus	230.3	62.3	72.9	415	26.2	88.6	175	1.9	99.1	13	6.9	**97.0**	**46**
football	239.8	74.5	69.0	497	33.2	86.2	221	2.0	99.2	13	8.8	**96.3**	**59**
cheer_leaders	738.0	227.6	69.2	1517	97.6	86.8	651	6.9	99.1	46	25.2	**96.6**	**168**
flower_garden	668.8	189.2	71.7	1261	80.3	88.0	535	5.7	99.1	38	20.3	**97.0**	**135**
parkrun	2209.3	666.5	69.8	4443	281.2	87.3	1875	18.3	99.2	122	73.5	**96.7**	**490**
SlideEditing	1940.2	572.7	70.5	3818	250.8	87.1	1672	17.5	99.1	117	65.9	**96.6**	**439**
average	-	-	71.0	-	-	87.5	-	-	99.2	-	-	**97.4**	-

Criteria: C1: time(s), C2: time saving(%), C3: time/frame(ms)

is a little worse than the SL method while MR method has the best performance among the 4 methods. OL method performs worst with mostly over 50% BDBR increase and over 3 dB BDPSNR loss. Results in Table 4.5 show that proposed transcoder performs better for videoconferencing sequences. The BDBR increase is similar for top 6 sequences and lower 6 sequences, but the average BDPSNR loss for videoconferencing sequences is only 0.71 dB while the average loss for other sequences is 1.33 dB. Similar to the computation time issue, the reason lies in the large percentage of AVC INTER coded MBs which will be processed as specified in Section 4.4.3. The quality for these MBs is well preserved. In order to give intuitive comparisons, Figure 4.16 is provided which shows the R-D (rate-distortion) curves for tested sequences. The results of 5 methods are shown, i.e., re-encoding, motion reuse, single-loop, proposal and the open-loop methods. It is obvious that for all sequences the re-encoding method performs best (topmost curve), following by MR, SL and proposal curves sequentially with similar small gaps. OL method is much worse than the other 4 methods, mostly 3 to 4 dB lower.

4.7 Conclusions

This chapter proposes an ultra-low-delay SVC-to-AVC MGS transcoder in frequency domain. The KEY frames of input MGS bitstream are transcoded using a single-loop based BL-copy transcoding architecture. Non-KEY frames are transcoded according the MB mode types using very fast quantization-domain transcoding methods. The resulted error is restricted within GOPs and no visual artifacts are introduced.



(a) akiyo (b) panzoom2 (c) vidyo1

(d) vidyo3 (e) vidyo4 (f) FourPeople

(g) bus (h) football (i) cheer_leaders

(j) flower_garden (k) parkrun (l) SlideEditing

Figure 4.16: R-D curves comparison.

Table 4.5: Coding efficiency comparisons.

Sequence	Motion Reuse		Single Loop		Open Loop		Proposal	
	C4	C5	C4	C5	C4	C5	C4	C5
akiyo	+4.8	-0.23	+9.9	-0.46	+282.6	-5.73	**+16.3**	**−0.73**
panzoom2	+4.5	-0.19	+8.1	-0.33	+173.9	-4.16	**+14.9**	**−0.60**
vidyo1	+6.1	-0.23	+11.8	-0.44	+190.7	-4.41	**+24.4**	**−0.87**
vidyo3	+4.2	-0.17	+9.8	-0.40	+174.6	-3.77	**+19.9**	**−0.76**
vidyo4	+6.8	-0.25	+10.4	-0.38	+158.8	-3.21	**+18.7**	**−0.66**
FourPeople	+5.9	-0.26	+10.3	-0.43	+94.2	-2.78	**+15.5**	**−0.62**
bus	+6.2	-0.42	+14.6	-0.91	+61.6	-3.66	**+19.1**	**−1.19**
football	+7.5	-0.52	+13.5	-0.93	+73.9	-4.04	**+25.4**	**−1.73**
cheer_leaders	+6.2	-0.47	+10.4	-0.76	+63.8	-4.02	**+17.4**	**−1.22**
flower_garden	+7.2	-0.48	+12.9	-0.84	+120.5	-5.90	**+19.8**	**−1.23**
parkrun	+4.6	-0.29	+10.2	-0.64	+109.5	-5.19	**+17.3**	**−1.07**
SlideEditing	+4.3	-0.53	+7.0	-0.82	+49.2	-5.01	**+13.8**	**−1.56**
average	+5.7	-0.34	+10.77	-0.61	+129.4	-4.32	**+18.5**	**−1.02**

Criteria: C4: BDBR(%), C5: BDPSNR(dB)

Simulation results show that proposed method gains 38.5 times speed-up comparing with the re-encoding method with acceptable coding efficiency loss. Besides, experiments show that proposed transcoder performs better for videoconferencing sequences. The proposed transcoder is expected to play an importance role in a hybrid videoconferencing application.

4. DRIFT CONSTRAINED FREQUENCY-DOMAIN SVC TO AVC QUALITY HOMOGENEOUS TRANSCODING

Chapter 5

Mode mapping and MV conjunction based SVC to AVC quality heterogenous transcoding

5.1 Introduction

Recent representative works on SVC/AVC transcoding support the SVC scalability in terms of temporal [37, 38], quality [23] and spatial [30, 39, 40]. For AVC-to-SVC temporal transcoding, the pixel-domain methods give good performance in both time reduction and coding efficiency [37, 38], since large portion of the motion data can be reused directly. In literatures [23] the authors propose a fast AVC-to-SVC quality transcoding method based on transform-domain approaches and achieve more than 99% time reduction compared with the full re-encoding method. A pixel-domain AVC-to-SVC spatial transcoder is described in [39] which is mainly based on motion data adaptation and refinement. In [40] we further proposed a fast and efficient AVC to SVC transcoding architecture based on pixel-domain mode-mapping. SVC-to-AVC temporal transcoding is very nature by simple syntax adaptations. In [30], a pixel-domain fast mode decision method is proposed for SVC-to-AVC spatial transcoding. The original motion data from the input SVC bitstream are utilized to speed up the AVC encoder mode decision process. Macroblocks (MBs) are classified into three types and treated with different mode-mapping strategies. SVC-to-AVC quality transcoding has not been thoroughly invested yet.

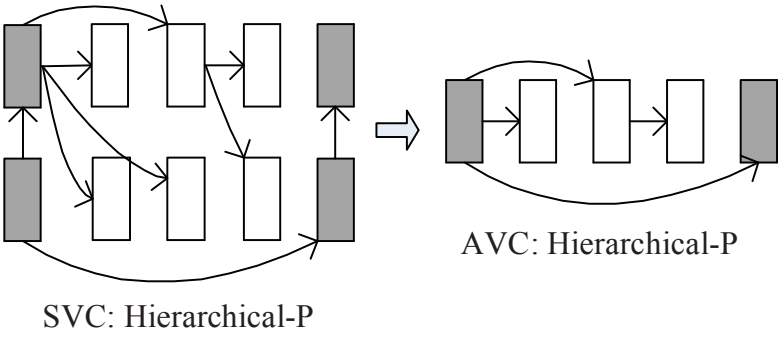


Figure 5.1: Hierarchical-P SVC to hierarchical-P AVC transcoding.

In this chapter, we propose a 3-stage fast transcoding method for SVC to AVC conversion with MGS scalability, targeting videoconferencing applications. In videoconferencing, B-picture is rarely utilized due to its high latency. Instead, hierarchical-P [44] structure is used for MGS-scalable SVC encoding. Mode and motion information can be fully utilized if it is transcoded into AVC bitstream with the same coding structure. However, the coding performance is very poor as will be revealed in simulation section. To improve the coding efficiency, we transcode SVC bitstream into IPPP structured AVC with multiple reference frames. In the first stage, mode decision is accelerated by proposed SVC-to-AVC mode mapping scheme. In the second stage, INTER motion estimation is accelerated by an optimized MV conjunction method to construct the MV predictor, and the search range is reduced. In the last stage, hadamard-based AZB detection is utilized for early termination of motion estimation process.

The rest of this chapter is organized as follows. Section 5.2 shows the proposed 3-stage transcoding method. Simulation results are given in Section 5.3 and conclusions are drawn in Section 5.4.

5.2 Proposed 3-stage transcoder

Figure 5.1 & 5.2 shows two possible transcoding approaches depending on the encoded AVC coding structure. Both hierarchical-P coding structure and IPPP coding structure are defined by the standard. Figure 5.1 shows the hierarchical-P transcod-

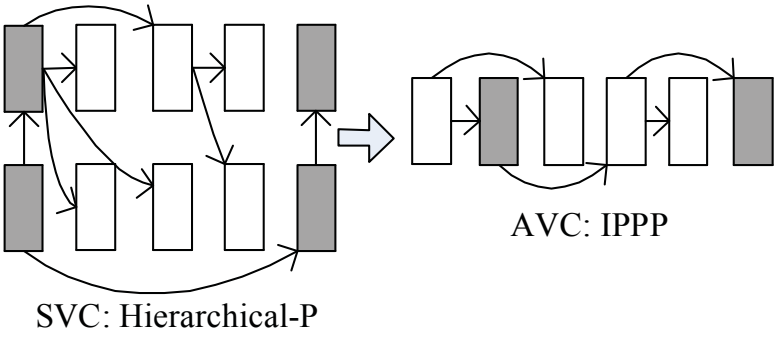


Figure 5.2: Hierarchical-P SVC to IPPP AVC transcoding.

ing, and Figure 5.2 shows the IPPP transcoding. In hierarchical-P transcoding, the AVC coding structure follows the same coding structure with SVC MGS layer. Thus most of the input SVC mode/motion data are reusable, resulting in very fast transcoding. However, the coding efficiency is much lower than IPPP transcoding, which allows to achieve optimal result among multiple reference frames as shown in Figure 5.2. The MGS layer of hierarchical-P structured SVC bitstream is transcoded into IPPP structured AVC bitstream. To obtain higher coding efficiency, our transcoder is based on IPPP transcoding. It consists of 3 stages, as explained in the following subsections.

5.2.1 SVC to AVC mode mapping

SVC introduces inter-layer predictions (ILP) while inheriting the conventional AVC modes (INTER and INTRA). Three kinds of ILPs are used to explore the correlation between layers, i.e., inter-layer residual, intra and motion predictions. These ILPs will be denoted as IL_Residual, IL_Intra and IL_Motion predictions hereafter.

IL_Intra prediction predicts the original enhancement layer input picture using the upsampled base layer reconstructed picture. IL_Residual prediction predicts the residual data generated by conventional INTER prediction. IL_Motion prediction tries to reduce the size of mode/motion data for INTER coded MBs. Detailed explanations can be found in [1].

Among the three ILPs, IL_Residual and IL_Motion predictions are additional refinements based on AVC INTER mode, while IL_Intra prediction is totally inde-

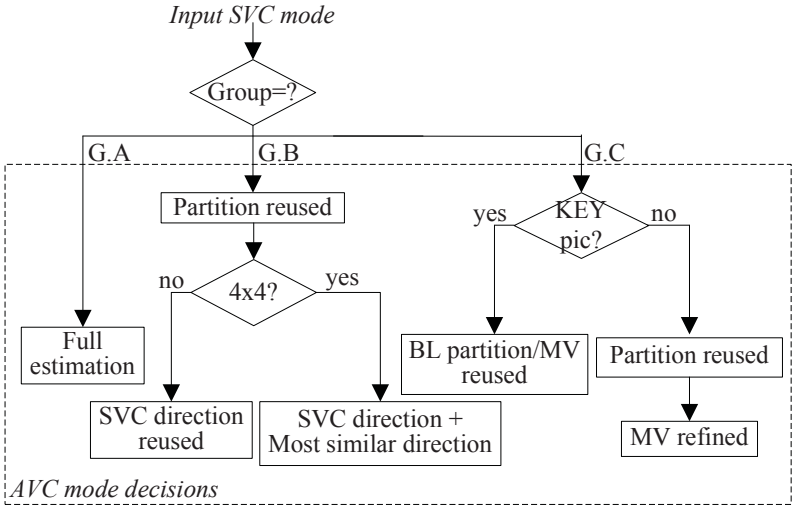


Figure 5.3: SVC to AVC mode mapping.

pendent from conventional AVC predictions. The coding modes in SVC are classified into three groups, as shown in Table 5.1. Group A includes IL_Intra only, for which no mode/motion information is transmitted in MGS layer. Group B includes conventional AVC INTRA mode only, for which intra partition and prediction mode are transmitted. Group C includes conventional AVC INTER mode, IL_Residual and IL_Motion predictions, for which inter partition and motion information are transmitted.

Table 5.1: Group classification of SVC modes

Groups	Included modes
G.A	IL_Intra
G.B	INTRA
G.C	INTER, IL_Residual, IL_Motion

These modes are mapped into AVC encoding process as shown in Figure 5.3, and details are described in following paragraphs.

■ For group A: full motion estimation is executed since there is no mode/motion information available to reuse.

■ For group B: since SVC follows the same INTRA coding process as AVC,

Table 5.2: Most similar intra 4x4 directions

Intra 4x4 direction	Most similar direction
DC	-
vertical/vertical-right	vertical-right/vertical
horizontal/horizontal-up	horizontal-up/horizontal
diagonal down-left/vertical-left	vertical-left/diagonal down-left
diagonal down-right/horizontal-down	horizontal-down/diagonal down-right

the optimized intra partition and prediction mode are reused in AVC encoding. To improve the accuracy, additional most similar mode are also examined if the intra partition is 4x4 (except DC mode). As shown in Table 5.2, the pairs with most similar directions are vertical/vertical-right, horizontal/horizontal-up, diagonal down-left/vertical-left and diagonal down-right/horizontal-down.

■ For group C: similar to group B, the optimized inter partition and motion vectors are reused for the **corresponding** reference picture. To improve the accuracy, motion vectors are used as the motion search start point, with a search range of [-2,+2] for both x- and y-components. One special situation is that when the current picture is a KEY picture, MGS layer pictures are not used as references. In such case, the corresponding reference picture is set to be the MGS layer of previous KEY picture, and the mode/motion data are set to the corresponding base layer data.

5.2.2 Optimized MV conjunction

For group C modes in previous subsection, the partition/MV information is deduced from input SVC bitstream only for **corresponding** reference picture with same coding structure. However, multiple reference pictures are used in proposed transcoder. For the other reference pictures, an optimized MV conjunction scheme is applied to construct the motion vectors. 16 MVs are deduced for each reference picture, corresponding to the 16 4x4 blocks in current MB.

If the current picture is a KEY picture, the corresponding reference picture is set to be the MGS previous KEY picture and mode/motion data are copied from base layer. Thus for MGS layer every picture is “connected” with nearest previous

KEY picture, as shown in Figure 5.4. For example, the frame i is connected to $i-3$ through $i-1$.

Let frame i in Figure 5.4 be the current picture to be coded. The current MB in frame i is first divided into 16 4x4 sized sub-blocks. For each MB in frame i, let the $MVj_{(i,i-1)}$ be the MV of 4x4 sub-block j $(0 \leq j < 16)$ regarding reference picture $i-1$. The pointed location of $MVj_{(i,i-1)}$ in reference picture $i-1$ is checked, and the MV of the block occupying this location is recorded as $MVj_{(i-1,i-3)}$. If this block contains no MV (INTRA or IL_Intra coded), then zero MV is recorded. For each sub-block j $(0 \leq j < 16)$, the MV regarding previous KEY picture (frame $i-3$) $MVj_{(i,i-3)}$ is calculated by MV conjunction as Equation (5.1).

$$MVj_{(i,i-3)} = MVj_{(i,i-1)} + MVj_{(i-1,i-3)} \tag{5.1}$$

To calculate MVs for frame $i-2$, firstly MVs between frame i and $i-2$ are estimated as Equation (5.2). Then the 4x4 block pointed by $MVj'_{(i,i-2)}$ is found. Together with 8 surrounding 4x4 blocks, an optimization process is performed to find the block which forms nearest MV with $MVj_{(i,i-3)}$ by connecting frames i and $i-3$ via $i-2$. Equation (5.3) shows the optimization problem statement. The optimal 4x4 sub-block k is chosen to be the reference block in frame $i-2$.

$$MVj'_{(i,i-2)} = MVj_{(i,i-1)} + 1/2 \times MVj_{(i-1,i-3)} \tag{5.2}$$

$$\begin{aligned} &\underset{k}{\text{minimize}} \quad |MVj_{(i,i-3)} - MVj'_{(i,i-2)} - MVk_{(i-2,i-3)}| \\ &\text{subject to} \quad 0 \leq k < 9 \end{aligned} \tag{5.3}$$

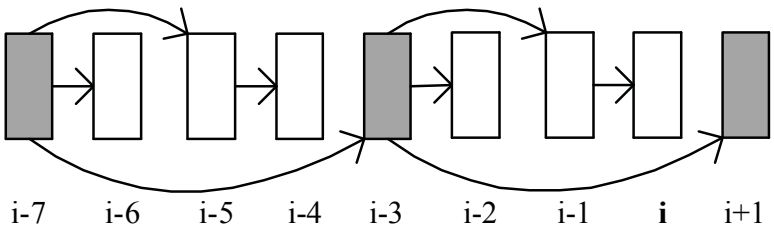


Figure 5.4: MGS layer prediction structure.

For frames $i-4$ and $i-6$, similar optimization is performed except the estimation weight is adapted according to the frame distance with frame i. Equations (5.4) & (5.5) show the estimations, and Equations (5.6) & (5.7) show the corresponding optimization problem statement.

$$MVj'_{(i,i-4)} = MVj_{(i,i-3)} + 1/4 \times MVj_{(i-3,i-7)} \tag{5.4}$$

$$MVj'_{(i,i-6)} = MVj_{(i,i-3)} + 3/4 \times MVj_{(i-3,i-7)} \tag{5.5}$$

$$\begin{aligned} \underset{k}{\text{minimize}} \quad & |MVj_{(i,i-3)} + MVj_{(i-3,i-7)} - MVj'_{(i,i-4)} \\ & -MVk_{(i-4,i-7)}| \\ \text{subject to} \quad & 0 \leq k < 9 \end{aligned} \tag{5.6}$$

$$\begin{aligned} \underset{k}{\text{minimize}} \quad & |MVj_{(i,i-3)} + MVj_{(i-3,i-7)} - MVj'_{(i,i-6)} \\ & -MVk_{(i-6,i-7)}| \\ \text{subject to} \quad & 0 \leq k < 9 \end{aligned} \tag{5.7}$$

The frames $i-5$ and $i-7$ can be obtained easily by conjuncting other MVs, as shown in Equations (5.8) & (5.9). Thus for each reference frame, MVs for all 4x4 sub-blocks can be obtained.

$$MVj_{(i,i-5)} = MVj_{(i,i-4)} + MVj_{(i-4,i-5)} \tag{5.8}$$

$$MVj_{(i,i-7)} = MVj_{(i,i-3)} + MVj_{(i-3,i-7)} \tag{5.9}$$

INTER prediction is then performed based on these MVs. All the INTER partitions are examined, and the MV for each sub-partition is formed by averaging the 4x4 block MVs within this sub-partition. Also a search range of [-2,+2] is applied to refine the coding performance.

5.2.3 Hadamard-based early termination

In [54] we proposed a hadamard-transform based all zero block detection method for AVC. It is utilized in our transcoder to early terminate motion estimation. The

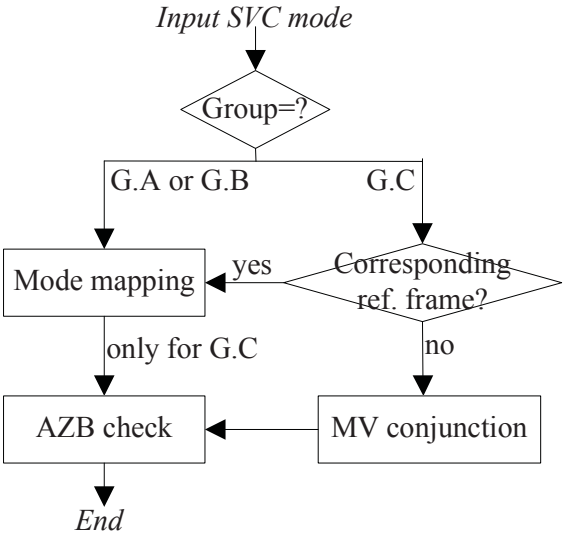


Figure 5.5: Overall proposed scheme.

hadamard transform coefficients are compared with a threshold. If all the coefficients are smaller than the threshold, the block is considered to be an AZB. The threshold is defined as Equation (10), where $qp\_rem = QP\%6$, $qp\_bits = QP/6 + 15$, $qp\_const = (1 << qp\_bits)/6$ and $quant\_coeff$ is the scaling matrix defined by AVC standard.

$$Th = \frac{2^{qp\_bits} - qp\_const}{quant\_coef[qp\_rem][0][0]} \tag{5.10}$$

5.2.4 Overall scheme

Overall proposed scheme is shown in Figure 5.5. Depending on the mode of input SVC bitstream MB, proposed sub-schemes are applied accordingly. Note that for group C, actually multiple reference frames are examined at "$corresponding\ ref.\ frame$?" phase. To better illustrate this matter, pseudo code of proposed scheme is shown as Algorithm 1 for reference.

Algorithm 1 Proposed transcoding scheme

```
if (mode == IL_Intra) then
  Full motion estimation;
else if (mode == INTRA) then
  mode == DC ? DC : (paired modes);
else
  Set search range = 2;
  Set motion data = MGS layer data;
  ME for corresponding reference frame
  AZB check;
  if (isAZB == true) then
    return;
  end if
  for (rest reference frames) do
    MV conjunction;
    ME for current reference frame;
    AZB check;
    if (isAZB == true) then
      return;
    end if
  end for
end if
```

Table 5.3: Experimental configurations

Parameters	SVC encoding	AVC encoding
Software Version	JSVM 9.18	JSVM 9.18
AVCMode	0	1
FramesToBeEncoded	150	150
SymbolMode	CABAC	CABAC
Enable8x8Transform	disabled	disabled
CodingStructure	Hierarchical-P	IPPP
NumRefFrames	-	5
SearchMode	4 (FastSearch)	4 (FastSearch)
SearchRange	16 for CIF/VGA, 32 for 720p	16 for CIF/VGA 32 for 720p
SearchFunc	hadamard	hadamard
Quantization Parameter	28 for BL, 24 for EL	20/24/28/32
Loop Filter	enabled	enabled
DisableBSlices	1 (B-slice disabled)	1 (B-slice disabled)
GOPSize	4	4
MGSVectorX(X=0,1,2)	3,3,10	-
InterLayerPred	2 (adaptive)	-
AVCRewriteFlag	0 (disabled)	-

5.3 Simulation results

In this section, the proposed transcoder is applied to several publicly available sequences and the results are shown. Software implementation is based on the SVC reference software JSVM (Joint Scalable Video Model). 12 sequences are encoded with 3-layer MGS scalability, and the highest MGS layer is transcoded into AVC format. *Akiyo*, *panzoom*2, $football$ and *bus* are CIF (352x288) sequences. $cheer\_leaders$ and $flower\_garden$ are VGA (640x480) sequences. *vidyo*1, *vidyo*3, *vidyo*4, $FourPeople$, *parkrun* and *SlideEditing* are 720p (1280x720) sequences. For each sequence 150 frames are tested. In our experiments, the QPs (quantization parameters) for input MGS bitstream encoding (SVC encoder) is set as 28 & 24 for base layer & MGS layer, and transcoder (AVC encoder) QPs are selected as 20, 24, 28 and 32. The main configuration parameters are shown in Table 5.3. All experiments are performed on an Intel Core 2 (2.67GHz) computer with 2.0GB RAM.

Table 5.4: Performance comparison.

Sequence	Hierarchical-P transcoding			Proposal		
	BDBR(%)	BDPSNR(dB)	Δ time(%)	BDBR(%)	BDPSNR(dB)	Δ time(%)
akiyo	+6.69	-0.41	-96.1	+0.59	-0.038	-94.7
panzoom2	+8.81	-0.44	-95.3	+1.09	-0.052	-93.2
vidyo1	+4.12	-0.22	-96.2	+0.47	-0.033	-94.5
vidyo3	+4.94	-0.35	-95.9	+0.59	-0.062	-93.4
vidyo4	+3.83	-0.23	-96.6	+0.49	-0.043	-95.7
FourPeople	+4.65	-0.32	-95.6	+0.48	-0.057	-94.3
bus	+21.29	-1.42	-87.3	+2.05	-0.131	-82.1
football	+18.43	-1.07	-88.7	+1.88	-0.122	-80.5
flower_garden	+15.29	-0.72	-91.6	+1.79	-0.085	-88.3
cheer_leaders	+11.36	-0.67	-93.7	+1.28	-0.103	-89.5
parkrun	+10.63	-0.61	-90.0	+1.67	-0.090	-84.3
SlideEditing	+16.13	-0.97	-90.2	+1.47	-0.125	-85.0
average	+10.51	-0.62	-93.1	+1.15	-0.078	-89.6

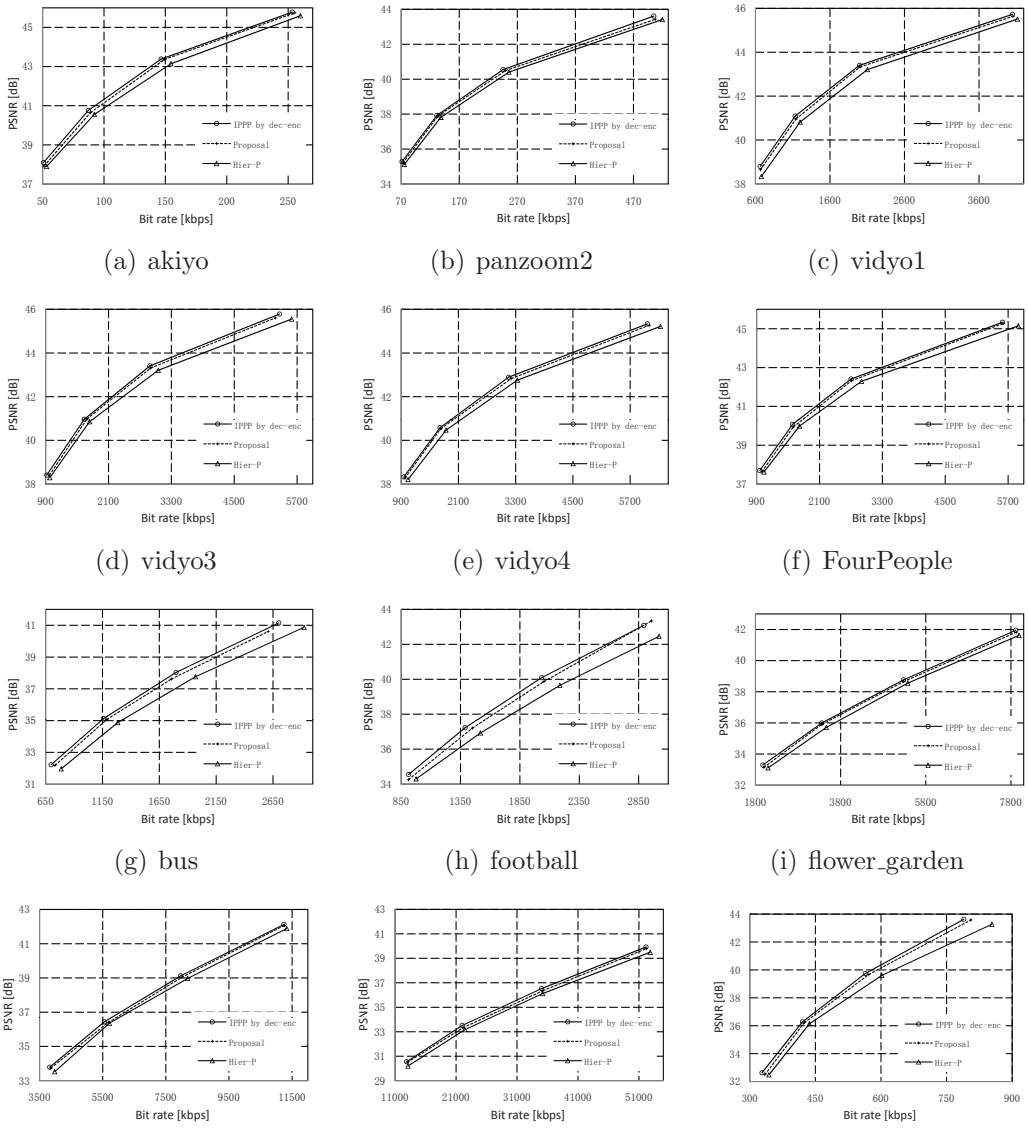
Table 5.4 shows the coding performance and time cost comparisons. Cascaded decoder-encoder IPPP transcoding method is selected as the basis for comparison. Besides our proposed transcoder, the hierarchical-P transcoding with full mode/motion reuse are also examined. Bjøntegaard Delta are used as the coding performance evaluation metric. The results show that proposed transcoder achieves very similar coding efficiency to IPPP transcoding. Only 1.15% BDBR increase and 0.078 dB BDPSNR decrease are found averagely, which is much better than the hierarchical-P transcoding. Time saving of our proposal ranges from 80.5% up to 95.7%, and the average time saving is 89.6% corresponding to a 9.6 times speed-up. Hierarchical-P transcoding is 1.5 times faster than our proposal, but with significant coding efficiency loss. Besides, the upper six sequences in Table 5.4 are videoconferencing-like sequences with simple background, slow movement or PTZ (panning, tilting & zooming) camera motions. The other six sequences are complex or fast sequences which rarely appear in videoconferencing. The proposed transcoder gains more time saving for the videoconferencing sequences than the other sequences. It saves averagely 94.3% time cost for the upper 6 sequences and only 85.0% for the lower 6 sequences, i.e., proposed transcoder is 2.6 times faster for videoconferencing sequences than the other sequences. The reason is that non-videoconferencing sequences contain many details and large portion of INTRA modes which deteriorates the effect of our proposed mode-mapping and MV conjunction schemes.

To illustrate the coding performance comparison intuitively, Figure 5.6 shows the

R-D curves where the coding efficiency is improved a lot by our proposal comparing with hierarchical-P transcoding. The R-D curves of proposed method are very near the ideal case of re-encoding by cascaded decoder-encoder.

5.4 Conclusions

This chapter proposes a 3-stage SVC-to-AVC MGS transcoder. In the first stage, mode decision is accelerated by proposed SVC-to-AVC mode mapping scheme. In the second stage, INTER motion estimation is accelerated by an optimized motion vector conjunction method to predict the MV with a reduced search range. In the last stage, hadamard-based all zero block detection is utilized for early termination. Results show that proposed transcoder achieves similar coding efficiency to optimal result, and the time saving is averagely near 90%.



(a) akiyo (b) panzoom2 (c) vidyo1

(d) vidyo3 (e) vidyo4 (f) FourPeople

(g) bus (h) football (i) flower_garden

(j) cheer_leaders (k) parkrun (l) SlideEditing

Figure 5.6: R-D curves comparison.

5. MODE MAPPING AND MV CONJUNCTION BASED SVC TO AVC QUALITY HETEROGENOUS TRANSCODING

Chapter 6

Conclusions and Future Works

In this dissertation, low-complexity transcoding techniques between H.264/SVC and H.264/AVC are proposed. The targeting application is videoconferencing. The target is to enable communication for hybrid videoconferencing scenarios with both SVC and AVC systems. The proposals in this dissertation are expected to be ready for industrial videoconferencing applications. The whole dissertation is divided into 4 main parts.

Firstly, a low-complexity H.264/AVC to H.264/SVC transcoder with spatial scalability is proposed based on coarse-level mode-mapping (CLMM). A novel one-to-many mode-mapping strategy is proposed at a coarser level based on the information of co-located macroblock (MB) in base layer. The whole AVC to SVC spatial transcoder is composed by following steps. First, mode skipping schemes are performed, including motion estimation (ME) skipping and probability-based mode control. Second, mode-mapping is performed based on CLMM schemes. Finally, motion vector (MV) refinement is applied in order to further reduce the complexity. Simulation results show that proposed SVC to AVC spatial transcoder achieves up to 92.6% time saving with insignificant coding efficiency loss.

Secondly, a low-complexity H.264/SVC to H.264/AVC transcoder with spatial scalability is proposed based on hybrid-domain transcoding with drift compensation. In proposed transcoder, MBs are classified into two types and optimal data reuse methods are applied accordingly in different domains. In the pixel domain, only mode and motion informations are reused. In the frequency domain, residual information is also reused. This architecture results in unsynchronized predictors

and causing drift problem. Compensation techniques are proposed for I frame and P frame accordingly. Simulation results show that proposed transcoder achieves averagely 96.4% time saving and 0.1-0.5 dB quality gain comparing with a representative conventional work.

Thirdly, a frequency-domain H.264/SVC to H.264/AVC transcoder with quality scalability is proposed targeting at ultra low delay. The key proposal is the approximation scheme of motion compensated prediction (MCP) at quantized coefficients level. Using the approximation, transform operations are avoided, resulting in extremely fast transcoding. To constrain the drift error, the KEY picture sequence with lowest temporal layer id is transcoded using drift-free schemes. The time saving is 97.4% averagely, with a 21 ms and 247 ms delay for CIF and 720p sequences respectively.

Finally, a mode-mapping and MV conjunction based H.264/SVC to H.264/AVC transcoder with quality scalability is proposed. Comparing with the scheme in the third part, better coding efficiency is obtained. The key proposal is the realization of mode/motion reuse for heterogeneous coding structures. The input SVC bitstream is hierarchical-P structured while AVC encoded bitstream is IPPP structured. Simulation results show that proposed transcoder achieves averagely 89.6% time saving with only 0.078 dB quality loss.

Three main future works are remaining to be done. First, fast and efficient AVC to SVC quality transcoding approaches will be examined. Second, after finishing AVC to SVC quality transcoding, system integration will be completed by combining all the works. Third, subjective evaluations are needed to verify the true visual effects of each proposal for practical applications, and trade-off between complexity and performance should be made based on not only objective but also subjective measures.

Bibliography

[1] H. Schwarz, and D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103-1120, 2007. 2, 28, 51, 69

[2] H. Schwarz and M. Wien, "The Scalable Video Coding Extension of the H.264/AVC Standard [Standards in a Nutshell]", IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 135-141, 2008. 2

[3] H. Choi, K. Lee, S.J. Bae, J.W. Kang and J.J. Yoo, "Performance Evaluation of the Emerging Scalable Video Coding", IEEE International Conference on Consumer Electronics (ICCE), , pp. 1-2, Las Vegas, NV, 2008. 3

[4] T. Oelbaum, H. Schwarz, M. Wien and T. Wiegand, "Subjective Performance Evaluation of the SVC Extension of H.264/AVC", 15th IEEE International Conference on Image Processing (ICIP), pp. 2772-2775, San Diego, CA, 2008. 3

[5] E.D. Jang, J.G. Kim, T.C. Thang, and J.W. Kang, "Adaptation of Scalable Video Coding to packet loss and its performance analysis" 12th International Conference on Advanced Communication Technology (ICACT), vol. 1, pp. 696-700, Phoenix Park, 2010. 3

[6] X. Li, P. Amon, A. Hutter, and A. Kaup, "Performance Analysis of Inter-Layer Prediction in Scalable Video Coding Extension of H.264/AVC", IEEE Transactions on Broadcasting, vol. 57, no. 1, pp. 66-74, 2011. 3, 19

[7] C.A. Segall and G.J. Sullivan, "Spatial Scalability Within the H.264/AVC Scalable Video Coding Extension", IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1121-1135, 2007. 3

[8] M.H. Willebeek-LaMair, D.D. Kandlur, and Z.Y. Shae, "On Multipoint Control Units for Videoconferencing", 19th Conference on Local Computer Networks (LCN), pp. 356-364, Minneapolis, MN, 1994. 3

[9] A. Vetro, C. Christopoulos, and H. Sun, "Video Transcoding Architectures and Techniques: An Overview", IEEE Signal Processing Magazine, vol. 20, no. 2, pp. 18-29, 2003. 3, 4, 25

[10] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to Lower Spatio-Temporal Resolutions and Different Encoding Formats", IEEE Transactions on Multimedia, vol. 2, no. 2, pp. 101-110, 2000. 3, 25, 47

[11] S. Li, L. Li, T. Ikenaga, S. Ishiwata, M. Matsui, and S. Goto, "Content-Based Complexity Reduction Methods for MPEG-2 to H.264 Transcoding", IEICE Transactions on Information and Systems, vol. E90-D, no. 1, pp. 90-98, 2007. 3

[12] H. Sun, W. Kwok, and J.W. Zdepski, "Architectures for MPEG Compressed Bitstream Scaling", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 2, pp. 191-199, 1996. 3, 25, 47

[13] S.F. Chang and D.G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video", IEEE Journal on Selected Areas in Communications, vol. 13, no. 1, pp. 1-11, 1995. 25

[14] P.A.A. Assuncao and M. Ghanbari, "Post-processing of MPEG2 Coded Video for Transmission at Lower Bit Rates", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 4, pp. 1998-2001, Atlanta, GA, 1996. 3, 25

[15] D.G. Morrison, M.E. Nilsson, and M. Ghanbari, "Reduction of the bit-rate of compressed video while in its coded form", 6th International Workshop on Packet Video, pp. 392-406, Portland, OR, 1994. 3

[16] P.A.A. Assuncao and M. Ghanbari, "A Frequency-Domain Video Transcoder for Dynamic Bit-Rate Reduction of MPEG-2 Bit Streams", IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 8, pp. 953-967, 1994. 3, 25

[17] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG bitstreams", Signal Processing: Image Communication, vol. 8, pp. 481-500, 1996. 3

[18] N. Bjork and C. Christopoulos, "Transcoder Architectures for Video Coding", IEEE Transactions on Consumer Electronics, vol. 44, no. 1, pp. 88-98, 1998. 3, 7, 25, 47

[19] B. Shen, I.K. Sethi, and B. Vasudev, "Adaptive Motion-Vector Resampling for Compressed Video Downscaling", IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 6, pp. 929-936, 1999. 3, 7, 18, 25, 47

[20] W. Zhu, K.H. Yang, and M.J. Beacken, "CIF-to-QCIF Video Bitstream Down-Conversion in the DCT Domain", Bell Labs Technical Journal, vol. 3, no. 3, pp. 21-29, 1998. 3, 7, 25

[21] P. Yin, A. Vetro, B. Liu and H. Sun, "Drift Compensation for Reduced Spatial Resolution Transcoding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 11, pp. 1009-1020, 2002. 3, 7, 25, 48

[22] R. Garrido-Cantos, J. De Cock, J. Luis Martinez, S. Van Leuven, and P. Cuenca, "Motion-based Temporal Transcoding from H.264/AVC-to-SVC in Baseline Profile", IEEE Transactions on Consumer Electronics, vol. 57, no. 1, pp. 239-246, 2011. 4, 26, 49, 67

[23] J. De Cock, S. Notebaert, and R. Van de Walle, "Transcoding from H.264/AVC to SVC with CGS Layers", IEEE International Conference on Image Processing (ICIP), vol. 4, pp. IV-73-IV-76, San Antonio, TX, 2007. 4, 26, 49, 67

[24] J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle, "Advanced Bitstream Rewriting from H.264/AVC to SVC", IEEE International Conference on Image Processing (ICIP), pp. 2472-2475, San Diego, CA, 2008. 4, 26

[25] J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle, "Transcoding of H.264/AVC to SVC with Motion Data Refinement", IEEE International Conference on Image Processing (ICIP), pp. 3673-3676, Cairo, 2009. 4, 26

[26] J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle, "Architectures for Fast Transcoding of H.264/AVC to Quality-Scalable SVC Streams", IEEE Transactions on Multimedia, vol. 11, no. 7, pp. 1209-1224, 2009. 4, 26

[27] J. De Cock, S. Notebaert, K. Vermeirsch, P. Lambert, and R. Van de Walle, "Efficient Spatial Resolution Reduction Transcoding for H.264/AVC", IEEE International Conference on Image Processing (ICIP), pp. 1208-1211, San Diego, CA, 2008. 3, 7

[28] R. Sachdeva, S. Johar, and E. Piccinelli, "Adding SVC Spatial Scalability to Existing H.264/AVC Video", IEEE/ACIS International Conference on Computer and Information Science (ICIS), pp. 1090-1095, Shanghai, 2009. 4, 8, 26, 49, 67

[29] P. Zhang, Y. Liu, Q. Huang, and W. Gao, "Mode Mapping Method for H.264/AVC Spatial Downscaling Transcoding", IEEE International Conference on Image Processing (ICIP), vol. 4, pp. 2781-2784, Singapore, 2004. 7, 15, 25, 47

[30] H. Liu, Y. Wang, Y. Chen, and H. Li, "Spatial transcoding from Scalable Video Coding to H.264/AVC", IEEE International Conference on Multimedia and Expo (ICME), pp. 29-32, New York, NY, 2009. 4, 26, 40, 41, 42, 49, 67

[31] J. Youn, M. Sun, and C. Lin "Motion Vector Refinement for High-Performance Transcoding", IEEE Transactions on Multimedia, vol. 1, no. 1, pp. 30-40, 1999. 18, 47

[32] I. Ahmad, X. Wei, Y. Sun, and Y.Q. Zhang, "Video Transcoding: An Overview of Various Techniques and Research Issues", IEEE Transactions on Multimedia, vol. 7, no. 5, pp. 793-804, 2005. 25

[33] P. Yin, M. Wu, and B. Liu, "Video transcoding by reducing spatial resolution", IEEE International Conference on Image Processing (ICIP), vol. 1, pp. 972-975, Canada, 2000. 3, 7, 25, 48

[34] T. Shanableh and M. Ghanbari, "Transcoding architectures for DCT-domain heterogeneous video transcoding", IEEE International Conference on Image Processing (ICIP), vol. 1, pp. 433-436, Greece, 2001. 25

[35] P. Yin, A. Vetro, B. Liu, and H. Sun, "Drift Compensation for Reduced Spatial Resolution Transcoding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 11, pp. 1009-1020, 2002. 25

[36] A. Segall and J. Zhao, "Bit-stream rewriting for SVC-to-AVC conversion", IEEE International Conference on Image Processing (ICIP), pp. 2776-2779, San Diego, 2008. 26

[37] A. Dziri, A. Diallo, M. Kieffer, and P. Duhamel, "P-Picture Based H.264 AVC to H.264 SVC Temporal Transcoding", International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 425-430, Greece, 2008. 26, 49, 67

[38] R. Garrido-Cantos, J. De Cock, J. Luis Martinez, S. Van Leuven, and P. Cuenca, "Motion-based Temporal Transcoding from H.264/AVC-to-SVC in Baseline Profile", IEEE Transactions on Consumer Electronics, vol. 57, no. 1, pp. 239-246, 2011. 4, 26, 49, 67

[39] R. Sachdeva, S. Johar, and E. Piccinelli, "Adding SVC Spatial Scalability to Existing H.264/AVC Video", IEEE/ACIS International Conference on Computer and Information Science (ICIS), pp. 1090-1095, Shanghai, 2009. 4, 8, 26, 49, 67

[40] L. Sun, J. Leng, J. Su, Y. Huang, H. Motohashi, and T. Ikenaga, "Low-Complexity Coarse-Level Mode-Mapping Based H.264/AVC to H.264/SVC Spatial Transcoding for Video Conferencing", IEICE TRANSACTIONS on Information and Systems, vol. E95-D, No. 5, pp. 1313-1323, 2012. 26, 49, 61, 67

[41] I.E.G. Richardson, "H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia", John Wiley & Sons, pp. 72-82, 2003. 29

[42] Y. Huang, Q. Liu, and T. Ikenaga, "Macroblock feature and motion involved multi-stage fast inter mode decision algorithm in H.264/AVC video coding", IEEE International Conference on Image Processing (ICIP), pp. 1041-1044, Egypt, 2009. 43

[43] Y.H. Huang, T. Ou, and H. Chen, "Fast decision of block size, prediction mode and intra block for H.264 intra prediction", IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 8, pp. 1122-1132, 2010. 43

[44] D. Hong, M. Horowitz, A. Eleftheriadis, and T. Wiegand, "H.264 Hierarchical P Coding in the Context of Ultra-Low Delay,Low Complexity Applications", Picture Coding Symposium (PCS), pp. 146-149, Nagoya, 2010. 41, 50, 62, 68

[45] G. Escribano, H. Kalva, P. Cuenca, L. Barbosa, and A. Garrido, "A Fast MB Mode Decision Algorithm for MPEG-2 to H.264 P-Frame Transcoding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 2, pp. 172-185, 2008. 47

[46] Q. Tang and P. Nasiopoulos, "Efficient Motion Re-Estimation with Rate-Distortion Optimization for MPEG-2 to H.264/AVc Transcoding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 2, pp. 172-185, 2010. 47

[47] P. Assuncao and M. Ghanbari, "Post-Processing of MPEG2 Coded Video for Transmission at Lower Bit Rates", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 4, pp. 1998-2001, 1996. 47, 61

[48] S. Notebaert, J. Cock, K. Wolf, and R. Walle, "Requantization Transcoding of H.264/AVC Bitstreams for Intra 4x4 Prediction Modes", Proceedings of Pacific-rim Conference on Multimedia (PCM), vol. 4261, no. 1, pp. 808-817, Hangzhou, 2006. 48

[49] J. Cock, S. Notebaert, P. Lambert, D. Schrijver, and R. Walle, "Requantization Transcoding in Pixel and Frequency Domain for Intra 16x16 in H.264/AVC", Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS), vol. 4179, no. 1, pp. 533-544, Berlin, 2006. 48

[50] J. Cock, S. Notebaert, P. Lambert, D. Schrijver, and R. Walle, "A Novel Hybrid Requantization Transcoding Scheme for H.264/AVC", Proceedings of International Symposium on Signal Processing and Its Applications (ISSPA), pp. 1-4, Sharjah, 2006. 48

[51] P. Assuncao and M. Ghanbari, "A Frequency-Domain Video Transcoder for Dynamic Bit-Rate Reduction of MPEG-2 Bit Streams", IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 8, pp. 953-967, 1998. 48

[52] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", ITU-T document VCEG-M33, USA, 2001. 62

[53] H. Schwarz, T.Hinz, D. Marpe, and T. Wiegand, "Constrained Inter-layer Prediction for Single-loop Decoding in Spatial Scalability", IEEE International Conference on Image Processing (ICIP), vol. 2, pp. 870-873, Italy, 2005. 51

[54] Z. Liu, L. Li, Y. Song, S. Li, S. Goto and T. Ikenaga, "Motion Feature and Hadamard Coefficient-Based Fast Multiple Reference Frame Motion Estimation for H.264", IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 5, pp. 620-632, 2008. 73

Publications

Journals (with review)

[1] **Lei Sun**, Zhenyu Liu, and Takeshi Ikenaga, "Low-complexity Hybrid-domain H.264/SVC to H.264/AVC Spatial Transcoding with Drift Compensation for Videoconferencing", IEICE Transactions on Fundamentals, Nov. 2013. (to appear)

[2] **Lei Sun**, Zhenyu Liu, and Takeshi Ikenaga, "A Drift-Constrained Frequency-Domain Ultra-Low-Delay H.264/SVC to H.264/AVC Transcoder with Medium-Grain Quality Scalability for Videoconferencing", IEICE Transactions on Fundamentals, Vol. E96-A, No. 6, pp. 1253-1263, June 2013.

[3] **Lei Sun**, Jie Leng, Jia Su, Yiqing Huang, Hiroomi Motohashi, Takeshi Ikenaga, "Low-Complexity Coarse-Level Mode-Mapping Based H.264/AVC to H.264/SVC Spatial Transcoding for Video Conferencing", IEICE Transactions on Information and Systems, Vol. E95-D, No. 5, pp. 1313-1323, May 2012.

[4] Jia Su, Yiqing Huang, **Lei Sun**, Shinichi Sakaida, Takeshi Ikenaga, "Content Based Coarse to Fine Adaptive Interpolation Filter for High Resolution Video Coding", IEICE Transactions on Fundamentals, Vol. E94-A, No. 10, pp. 2013-2021, Oct. 2011.

International Conferences (with review)

[1] Gaoxing Chen, Zhenyu Pei, **Lei Sun**, Zhenyu Liu, and Takeshi Ikenaga, "Fast Intra Prediction for HEVC based on Pixel Gradient Statistics and Mode Refinement",

China Summit and International Conference on Signal and Information Processing (ChinaSIP), Beijing, China, July 2013. (to appear)

[2] Xiaoyang Yuan, Lei Gu, **Lei Sun**, and Takeshi Ikenaga, "Local-Threshold 2D-Tophat Cell Segmentation for the Two-Photon Confocal Microscope Image", International Conference on Machine Vision Applications (MVA), pp. 455-458, Kyoto, Japan, May 2013.

[3] **Lei Sun**, Zhenyu Liu, and Takeshi Ikenaga, "A Mode-Mapping and Optimized MV Conjunction based MGS-scalable SVC to AVC IPPP Transcoder", IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1648-1651, Beijing, China, May 2013.

[4] **Lei Sun**, Zhenyu Liu, and Takeshi Ikenaga, "A Low-complexity Quantization-domain H.264/SVC to H.264/AVC Transcoder with Medium-Grain Quality Scalability", 19th International Conference on Multimedia Modeling (MMM), pp. 336-346, Huangshan, China, Jan. 2013.

[5] **Lei Sun**, Zhenyu Liu, and Takeshi Ikenaga, "A Videoconferencing-oriented Hybrid-domain H.264/SVC to H.264/AVC Spatial Transcoder", 13rd Pacific-rim Conference on Multimedia (PCM), pp. 129-141, Singapore, Dec. 2012.

[6] **Lei Sun**, Zhenyu Liu, and Takeshi Ikenaga, "A Pixel-domain Mode-mapping based SVC-to-AVC Transcoder with Coarse Grain Quality Scalability", 21st International Conference on Pattern Recognition (ICPR), pp. 939-942, Tsukuba, Japan, Nov. 2012.

[7] Wei-Jing Chen, **Lei Sun**, Lei Gu, Zhen-yu Liu, and Takeshi Ikenaga, "A Low Complexity ALF based on Inter-Channel Correlation between Chroma and Luma in HEVC", 14th IASTED International Conference on Signal and Image Processing (SIP), Honolulu, USA, Aug. 2012.

[8] **Lei Sun**, Jie Leng, Jia Su, Yiqing Huang, Hiroomi Motohashi, and Takeshi Ikenaga, "Video Conferencing Oriented Low-Complexity Coarse-Level Mode-Mapping Based H.264/AVC to H.264/SVC Spatial Transcoding", APSIPA Annual Summit and Conference 2011 (ASC 2011), Xi'an, China, Oct. 2011.

[9] Jia Su, Yiqing Huang, **Lei Sun**, Shinichi Sakaida, and Takeshi Ikenaga, "Low Complexity Quadtree based All Zero Block Detection Algorithm for HEVC", APSIPA Annual Summit and Conference 2011 (ASC 2011), Xi'an, China, Oct. 2011.

[10] Jia Su, Yiqing Huang, **Lei Sun**, Shinichi Sakaida, and Takeshi Ikenaga, "Coarse to Fine Adaptive Interpolation Filter for High Resolution Video Coding", IEEE International Conference on Multimedia and Expo (ICME), pp. 1-6, Barcelona, Spain, July 2011.

[11] Jie Leng, **Lei Sun**, Takeshi Ikenaga, and Shinichi Sakaida, "Content Based Hierarchical Fast Coding Unit Decision Algorithm For HEVC", International Conference on Multimedia and Signal Processing (CMSP), pp. 56-59, Guilin, China, May 2011.

[12] Bingrong Wang, **Lei Sun**, Jia Su, and Takeshi Ikenaga, "Complicated Scene Retrieval Using Block Voting Mechanism and Weak Feature Selection Based on Bag-of-Features", 4th International Conference on New Trends in Information Science and Service Science (NISS), pp. 287-292, Gyeongju, Korea, May 2010.

[13] **Lei Sun**, Bingrong Wang, and Takeshi Ikenaga, "Real-time Non-rigid Object Tracking Using CAMShift with Weighted Back Projection", 10th International Conference on Computational Science and Its Applications (ICCSA), pp. 86-91, Fukuoka, Japan, Mar. 2010.

Standardization contributions

[1] K. Kawamura, T. Yoshino, S. Naito, **L. Sun**, T. Ikenaga, "Description of scalable video coding technology proposed by KDDI", JCTVC-K0052, 11th HEVC meeting, Shanghai, Oct. 2012.

Patents

[1] Y. Huang, H. Motohashi, T. Ikenaga, **L. Sun**, and J. Leng, "Image conversion equipment", Japanese patent, No. 2011-146349, pending.

PUBLICATIONS

[2] Y. Huang, H. Motohashi, T. Ikenaga, **L. Sun**, and J. Leng, "Image conversion equipment", Japanese patent, No. 2011-146364, pending.