國立臺灣大學電機資訊學院電子工程學研究所

博士論文

Graduate Institute of Electronics Engineering

College of Electrical Engineering & Computer Science

National Taiwan University

Doctoral Dissertation

# 多視角視訊編碼技術之演算法和硬體架構及系統設計之研究

# Multiview Video Coding: Algorithms, VLSI Architectures, and System Design

丁立夫

Li-Fu Ding

指導教授：陳良基 博士、簡韶逸 博士

Advisor: Liang-Gee Chen and Shao-Yi Chien, Ph.D.

中華民國 97 年 12 月

December, 2008

國立臺灣大學 · 敦品勵學 · 愛國愛人

# 中文摘要

三維立體視訊能夠藉由同時將不同視角的影像傳送至左右兩眼，進而提供完整的場景實境效果。在一個三維視訊系統中，由於有大量的資料量必須做有效的壓縮處理，多視角視訊編碼因此伴演了很重要的角色。在本篇論文中，我們將從演算法、硬體架構及系統設計等三個不同的層面來研究多視角視訊編碼技術。

在演算法的相關研究中，我們針對三維視訊編碼系統中的預測核心部分以及彩度校正部分分別提出了快速演算法。所謂的預測核心，包含了程動估計和位移估計兩種運算，分別可以消除時間和空間上的多餘資料量。然而，預測核心需要大量的運算量，造成軟硬體設計上的困難，因此，我們提出了內容感知預測演算法降低運算複雜度。運用位移估計先在不同視角間找到相對應的方塊其內容包含同樣物件內容，我們可以從已編碼後的視角影像中，有效地取得編碼資訊，如此可以在大多數的視角影像頻道中節省 98.4—99.1%的運算量，伴隨而來的代價只有 0.03—0.06dB 的 PSNR 下降量。此外，我們也針對了不同視角間亮度及彩度不對稱的問題，提出位移補償線性回歸的亮度及彩度校正演算法，同時同覆利用了移動向量的資訊，提升了 0.4dB 的編碼效率。

在多視角視訊編碼中，預測核心向來是運算複雜度最高的部分，如此的運算量，無法利用現今的處理器達成即時編碼的需求，因此利用硬體加速有其必要性。在論文的第二部分，我們分別針對雙視角以及多視角視訊編碼系統提出了兩個預測演算法及其對應的硬體架構。首先，我們提出了聯合預測演算法應用在雙視角視訊編碼系統中，以增加編碼效

率以及降低運算量。聯合預測演算法利用了雙視角影像的特性，成功降低了 80%的運算量，同時也增進了影像的品質。我們針對此演算法所設計以及實作的晶片大小約 2.13×2.13 $mm^2$，包含了 137K 個邏輯閘以及 20.75Kbits 的內嵌記憶體。與傳統的架構相較，此架構只使用了 11.5%的內嵌記憶體以及 3.3%的處理單元即可達到相同的規格。接著，我們將研究的廣度由雙視角擴展為多視角系統，我們改進了移動估計中的向量中心預測演算法，以增進移動向量的準確度。該演算法節省了 96%的運算複雜度，同時視訊品質只降低了 0.045dB。所提出的演算法的搜尋範圍在水平與垂直方向最大可支援到[-256, +255]/[-256, +255]，因此可以滿足四倍大高解析度影像的需求。我們以快取處理的概念設計相對應的硬體架構，利用快取的機制可以有效地節省 39%的外部記憶體頻寬。我們所實作的晶片只需 230K 個邏輯閘以及 8KB 的記憶體，在 300MHz 的操作頻率下，可以即時的處理四倍大高解析度的視訊影像。

接著，論文的第三部分針對多視角編碼系統提出了頻寬分析的方法以及單晶片編碼器的硬體架構。首先，我們針對各種不同且複雜的多視角編碼結構提出了一套新的系統頻寬分析方法，用圖論中的「次序限制」處理多視角編碼結構中影像間的預測相依性。根據此分析，我們即可針對不同的編碼結構達到硬體資源的最適分配。接下來，我們提出了多視角視訊編碼器的系統架構，可應用於未來三維立體電視和四倍高畫質解析度電視中。其中包含了一個新的八級巨集區塊管線架構以及系統排程，並且搭配快取式預測核心，能夠即時的處理單視角 4096×2160p、三視角 1920×1080p、到七個視角 1280×720p 的編碼運算。此外，為了維持高解析度影像的畫質，本架構可支援的搜尋範圍是傳統架構的 4—64 倍

大。與傳統架構相比，本架構可以節省 79%的系統頻寬和 94%的內嵌記憶體。我們所實作的晶片利用 CMOS 90 奈米的製程，大小為 11.46 mm$^2$，藉由新的系統排程、高度的平行化、演算法最佳化、以及模組化的時脈閘控等技術，本晶片能夠達到 1 億 1200 萬像素/秒的運算能力。本晶片為世界上第一顆多視角視訊單晶片編碼器。

簡而言之，我們對多視角編碼技術的貢獻主要有三個方向。我們所提出演算法可運用在多視角編碼系統的核心；此外，我們也針對雙視角以及多視角視訊編碼的預測核心率先提出了硬體架構以及晶片；並且，整合了論文前兩部分，我們所提出的多視角視訊編碼系統為世界上最前瞻的設計。利用我們提出的多視角編碼技術，三維立體視訊便能夠實現在許多實際的生活應用中，我們由衷地希望我們的研究成果能對人類生活的便利性帶來劃時代的突破。

國立臺灣大學
敦品勵學 · 愛國愛人

# Multiview Video Coding: Algorithms, VLSI Architectures, and System Design

*Advisor: Liang-Gee Chen and Shao-Yi Chien*

*Graduate student: Li-Fu Ding*

*December 2008*

*Department of Electrical Engineering*

*National Taiwan University*

*Taipei, Taiwan, R.O.C.*

國立臺灣大學 · 敦品勵學 · 愛國愛人

# Contents

**III System Analysis and Architecture Design of 4096×2160p**

# List of Figures

國立臺灣大學 · 敦品勵學 · 愛國愛人

# List of Tables

# Abstract

Three-dimensional (3D) video can provide a complete scene structure to human eyes by transmitting several views simultaneously. In 3D video systems, multiview video coding (MVC) plays a critical role because huge amount of data make compression necessary. In this dissertation, MVC is studied at three different levels: algorithm level, VLSI architecture level, and system design level. According to the research relevance between hardware-oriented algorithms and VLSI architectures, several proposed algorithms and their corresponding architectures are discussed together. Therefore, this dissertation is divided into three parts: efficient MVC algorithms, algorithm and architecture co-design of MVC prediction core, and system analysis and architecture design of MVC encoder.

In the first part of this dissertation, two new fast algorithms are developed for prediction core and color correlation. The prediction core mainly consists of motion estimation (ME) and disparity estimation (DE) to remove temporal and inter-view redundancy, respectively. We develop content-aware prediction algorithm (CAPA) for computational complexity reduction. By utilizing DE to find corresponding blocks between different views, the coding information can be effectively shared and reused from the coded view channel. It can save 98.4–99.1% computational complexity of ME in most view channels with negligible quality loss of only 0.03–0.06 dB in PSNR. In addition, we also develop an illuminance and chrominance correlation algorithm to deal with the color mismatch between views. The proposed algorithm is based on motion compensated linear regression which reuses the motion information from the encoder and provides better coding gain by up to 0.4 dB.

Prediction core is always the most computation-intensive part in an MVC system. Therefore, hardware acceleration is necessary for real-time processing requirement. In the second part of this dissertation, two prediction algorithms and their corresponding VLSI architectures are proposed for stereo and MVC, respectively. First, we propose joint prediction algorithm (JPA) for high coding efficiency and low computational complexity. JPA utilizes the characteristics of stereo video and successfully reduces about 80% computational complexity while enhances the video quality. A corresponding architecture of JPA is then designed and implemented on a 2.13×2.13$mm^2$ die with only 137K logic gates and 20.75 Kbits on-chip SRAM. The JPA architecture is an area-efficient design because only 11.5% on-chip SRAM and 3.3% processing elements are used compared with conventional architectures. Second, we extend the design space from stereo video to MVC systems. A predictor-centered prediction algorithm is modified to enhance the motion vector (MV) accuracy. 96% computational complexity can be saved with the penalty of quality loss of only 0.045dB. The proposed algorithm supports search range up to $[-256,+255]/[-256,+255]$ in horizontal/vertical directions, so it can fulfill the requirement of quad full high-definition (QFHD) videos. A corresponding architecture is then presented based on the cache processing concept. The proposed cache architecture saves 39% off-chip memory bandwidth with a rapid prefetching algorithm. The implementation results show that the proposed cache-based prediction architecture requires little hardware cost: only 230K logic gates and 8KB on-chip SRAM (2.1×2.1$mm^2$ with 90 *nm* process) are required to process QFHD videos in real-time at the working frequency of 300 MHz.

The third part of this dissertation describes system analysis and architecture design of MVC encoder. The accumulated know-how and design experience of the previous parts are integrated. First, a new system bandwidth analysis scheme is proposed for various and complicated MVC structures. The precedence constraint in the graph theory is adopted for deriving the processing order of frames in an MVC system. The suitable hardware resource allocation can be easily determined with the proposed analysis scheme. Second, the system architecture of MVC encoder for 3D/QFHD applications is presented. An eight-stage macroblock pipelined architecture with proposed system scheduling and cache-based prediction core supports real-time processing from one-view 4096×2160p to seven-view 720p videos. In addition,

the search range is four to sixty-four times larger than the previous work to maintain HD video quality. The proposed architecture saves 79% system memory bandwidth and 94% on-chip SRAM. A prototype chip is implemented on a 11.46mm$^2$ die with 90nm CMOS technology. The 212M pixels/s throughput and 407M pixels/Watt are achieved by proposed system scheduling, high degree of parallelism to reduce memory access, algorithmic optimization, and module-wise clock gating, etc. The proposed architecture is the worldwide first reported MVC single-chip encoder.

In brief, we believe that with the MVC technologies proposed in this dissertation, 3D video processing can be realized in many real applications. We sincerely hope that our research contributions can create a new era for digital multimedia life.

國立臺灣大學 · 人為國愛 · 學勵品敦

# Chapter 1

# Introduction

## 1.1 Trends of 2D and 3D Television Systems

From the 1930s, the monochrome TV started to transmit monochrome images. Only the transmission of images make people crazy. At that time, the contents in the television are mainly pantomimes. They do not need sounds, but they can still lead one into wonderland. The monochrome TV broadcast continued for a while. In 1946, RCA announced a new standard – NTSC, which is backward compatible to the monochrome television and with 525 scan lines. The frame rate is 29.97 frame per second (fps). NTSC standard is still in use until today. Its history is around 60 years. In 1950s, the transistor technology became more mature. Then the transistors replaced the traditional vacuum tubes, and TVs are miniaturized and popularized. More and more families brought TVs to their house. From 1954, the RCA company officially started the broadcasting of color TV programs. In 1963, the PAL standard was also formulated in Europe. The scan lines of the PAL standard raised to 625 lines, but its frame rate reduced to 25 fps. The camcorder is evolving, too. From the earliest large vacuum tube camcorder to 1970s, NASA successfully developed the CCD technique to reduce the volume of the camcorder, and no more film was needed. The data can be recorded in the removable media, and this eased the shooting of the TV programs. Various kinds of TV programs were possible. There's no need to shoot in the film studio anymore. In Taiwan, the color TV program is broadcasted since 1969. There are 40 years for TV to evolve from monochrome to color, but people

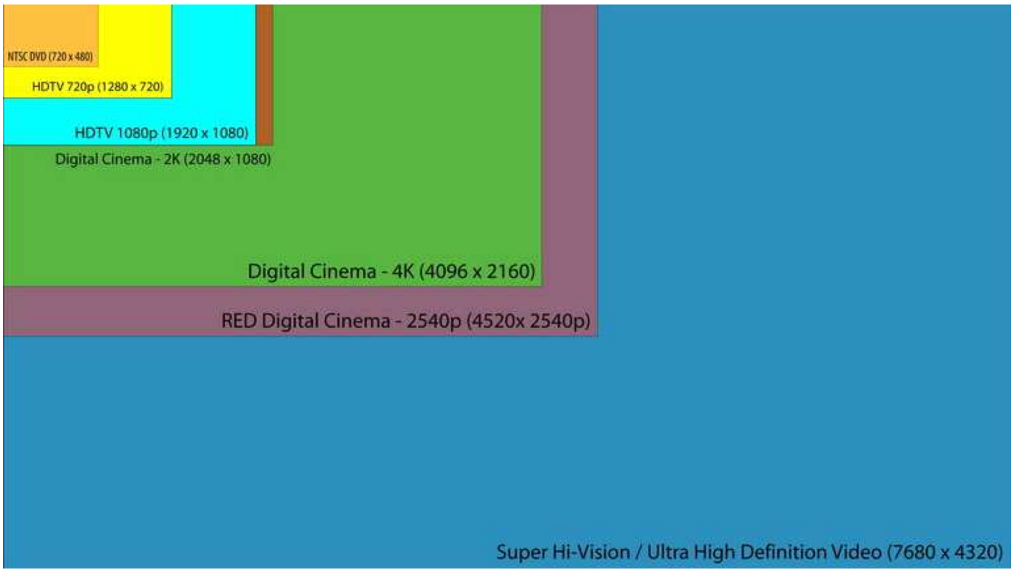

NTSC DVD (720 x 480)
HDTV 720p (1280 x 720)
HDTV 1080p (1920 x 1080)
Digital Cinema - 2K (2048 x 1080)
Digital Cinema - 4K (4096 x 2160)
RED Digital Cinema - 2540p (4520x 2540p)
Super Hi-Vision / Ultra High Definition Video (7680 x 4320)

Figure 1.1: Relative size of digital TV format.

still pursues more real vision.

In 1981, NHK proposed the HDTV standard. The number of the scan lines is up to 1125. It was an acme of the vision. However, it still costs twenty years to set HDTV into action. The cathode ray tube are replaced by TFT-LCD. TFT-LCD panels not only reduce the volume of the TV but also raise the resolution to the extent. In 2007, display panels with Full HD resolution are everywhere, and the resolution of the camcorders is raised, too. Furthermore, “quad HDTV,” which has a resolution of a resolution of 3840×2160 or 3840×2160, displays four times the number of pixels of the highest HDTV standard resolution, 1080p. Quad Full High Definition (QFHD) is the next step in high-resolution display technology, and won’t start shipping until 2015. The only step higher in television technology is Ultra High Definition Video. The relative size digital TV format is illustrated in Fig. 1.1, and the specifications and description of HDTV is listed in Table 1.1.

In addition to HDTV, another way to bring human the complete scene perception is three-dimensional broadcast TV (3DTV). 3DTV technology utilize the one of the significant human depth cue, stereo parallax, provides different perspective views to the left and right eye simultaneously. There are several kinds of 3D displays which have been commercialized, as shown in Fig. 1.2. More views are provided, more vivid scene structures are sensed by the viewers. 3D TV is believed by many to

Table 1.1: Specifications of HDTV [4]

| Video Format Supported | Native Resolution (W×H) | Pixels (Advertised Megapixels) | Aspect Ratio (X:Y) | | Description |
|---|---|---|---|---|---|
| | | | Image | Pixel | |
| 720p<br>1280×720 | 1024×768<br>XGA | 786,432 (0.8) | 16:9 | 4:3 | Typically a PC resolution XGA; also exists as a standardized "HD-Ready" TV on the Plasma display with non-square pixels. |
| | 1280×720 | 921,600 (0.9) | 16:9 | 1:1 | Typically one of the PC resolutions on WXGA, also used for 750-line video, as defined in SMPTE 296M, ATSC A/53, ITU-R BT.1543, Digital television, DLP and LCOS projection HDTV displays. |
| | 1366×768<br>WXGA | 1,049,088 (1.0) | 683:384<br>(Approx 16:9) | 1:1<br>Approx | Typically a TV resolution WXGA; also exists as a standardized HDTV displays as (HD Ready 720p,1080i), TV that used on LCD HDTV displays. |
| 1080i<br>1920×1080 | 1280×1080 | 1,382,400 (1.4) | 32:27<br>(Approx 16:9) | 3:2 | Non-standardized "HD Ready", TV. Used on HDTV Plasma display with non-square pixels. |
| 1080p<br>1920×1080 | 1920×1080 | 2,073,600 (2.1) | 16:9 | 1:1 | A standardized HDTV displays as (HD Ready 1080p) TV, that used on LCD HDTV displays. Used for 1125-line video, as defined in SMPTE 274M, ATSC A/53, ITU-R BT.709. |
| 2160p<br>3840×2160 | 3840×2160 | 8,294,400 (8.3) | 16:9 | 1:1 | Quad HDTV for DCI Cinema 4k standard format, (Currently, there is no HD Ready 2160p Quad HDTV format until 2015). |

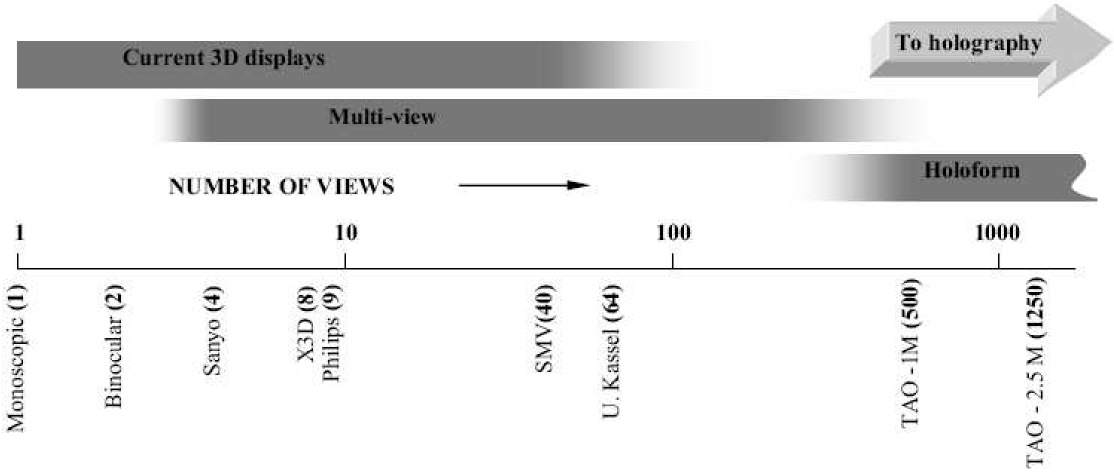


Figure 1.2: Statistics of view number of current commercialized 3D displays.

be the next logical development towards a more natural and life-like visual home entertainment experience. As shown in 1.3, 3DTV system is a huge infrastructure compared with conventional 2D video processing chain. 3D video capturing and display are the key components differ from 2D video capturing and display. 3D video content contains much larger amount of data than 2D video, so the video CODEC and transmission of 3D video are the main challenges. The “3D playback” represents the arbitrary view selection by users. It is a special functionality supported by a free-view-point TV (FTV) system. A complete 3D TV system should aim to fulfill important requirements:

- Backwards-compatibility to todays digital 2D color TV.

- Low additional storage and transmission overhead.

- Support for autostereoscopic, single and multiple user 3D displays.

- Flexibility in terms of viewer preferences on depth reproduction

- Simple way to produce sufficient, high-quality 3D content.

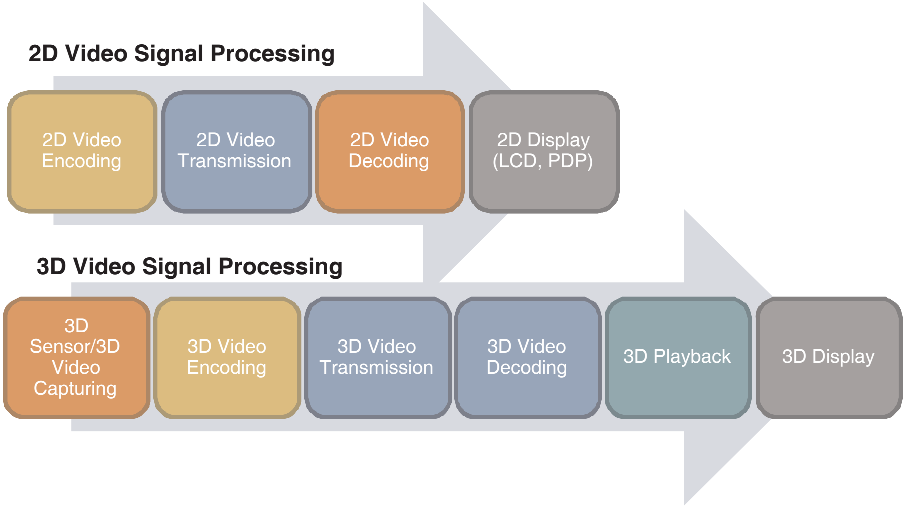


Figure 1.3: The signal processing pipeline of 2D and 3D video, respectively.

## 1.2 Development of 2D and 3D Video Coding Standards

### 1.2.1 2D Video Coding: From MPEG-1, H.261 to MPEG-4/H.264

Before the introduction of 3D video coding standards, 2D video coding standards are reviewed. Video compression plays an important role in digital TV system due to limited storage capacity and transmission bandwidth. Therefor, several video coding standards are defined for storage and broadcasting applications since 1990. The coding efficiency is the most critical consideration when developing new video coding standards. The evaluation of main video coding standards defined in the past years is shown in Fig. 1.4. From H.261 [5], MPEG-1 [6], MPEG-2 [7], H.263 [8], H.263+ [9], H.263++ [10], MPEG-4 [11], H.26L to H.264/AVC [12], these video standards all focus on the improvement of compression efficiency. Compared to MPEG-4 [11], H.263 [10], and MPEG-2 [7], the H.264/AVC baseline profile saves 39%, 49% and 64% bit rate respectively [13].

With the progress in video sensor, storage device, communication system, and display device, the style of multimedia applications changes and so does the direction of video coding standards. Therefore, H.264/AVC High Profile [14] and High Fidelity Extension [15] further address the need of high definition and high qual-

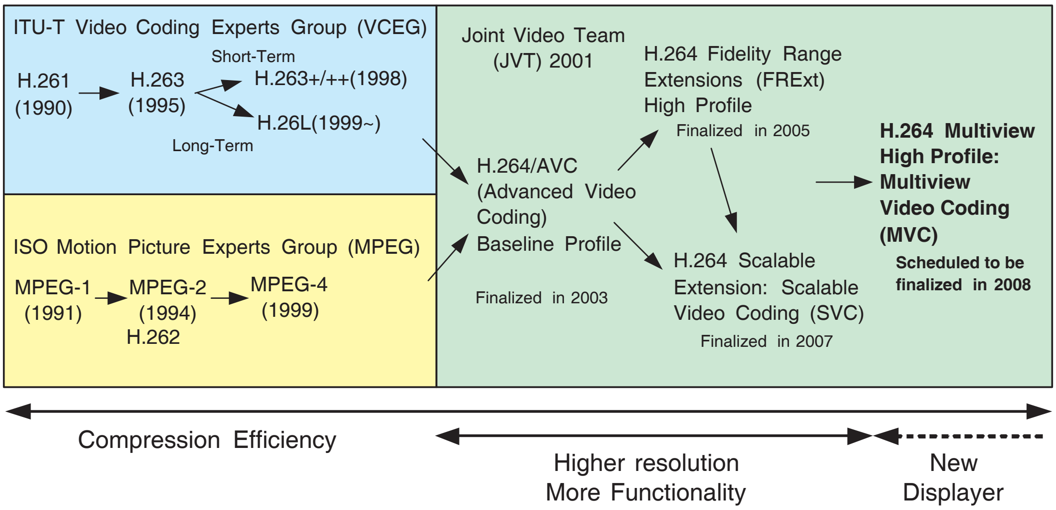
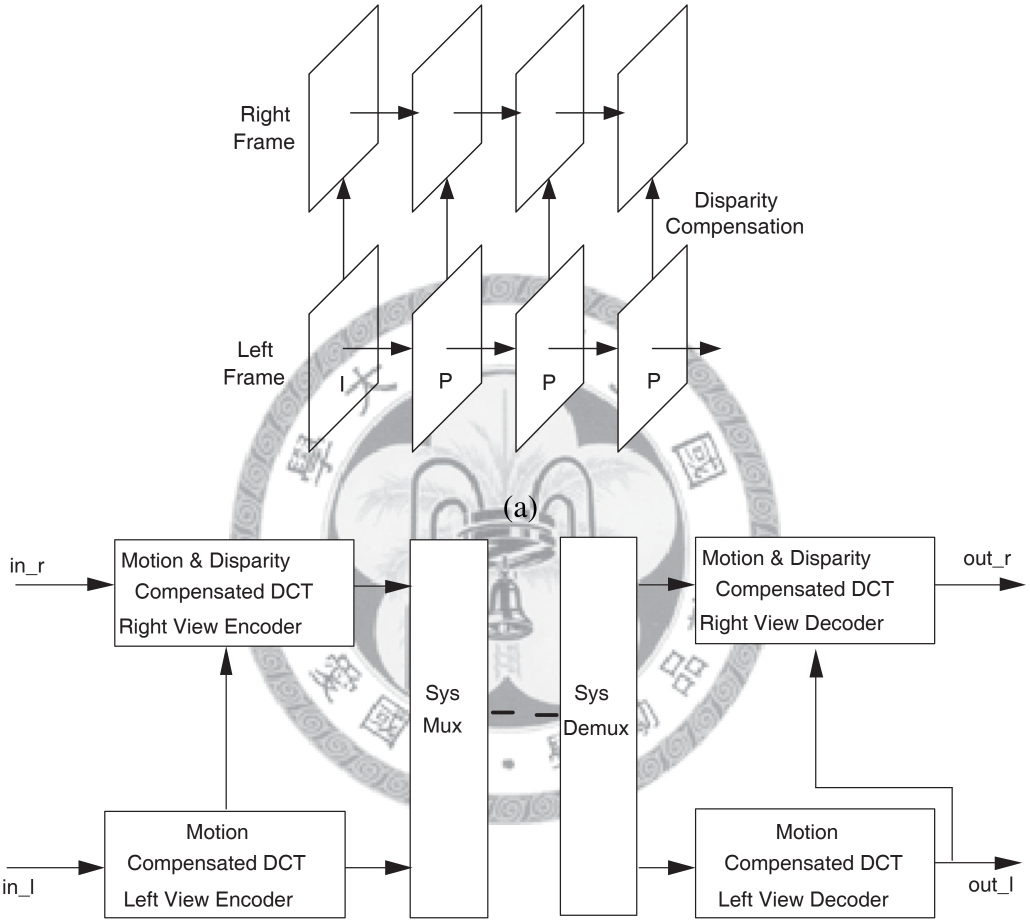

ITU-T Video Coding Experts Group (VCEG)
H.261 (1990)
H.263 (1995)
Short-Term
H.263+/++(1998)
H.26L(1999~)
Long-Term
ISO Motion Picture Experts Group (MPEG)
MPEG-1 (1991)
MPEG-2 (1994) H.262
MPEG-4 (1999)
Joint Video Team (JVT) 2001
H.264/AVC (Advanced Video Coding) Baseline Profile
Finalized in 2003
H.264 Fidelity Range Extensions (FRExt) High Profile
Finalized in 2005
H.264 Scalable Extension: Scalable Video Coding (SVC)
Finalized in 2007
H.264 Multiview High Profile: Multiview Video Coding (MVC)
Scheduled to be finalized in 2008
Compression Efficiency
Higher resolution More Functionality
New Displayer

Figure 1.4: The evaluation of video coding standards.

ity. On the other hand, the variety of communication systems also provide more and more wired or wireless channels and add more consumer electronics with different platform into multimedia systems. Thus, the scalability and functionality now become the necessity for video standards to support various demands. Under such environment, the H.264/AVC scalable extension (Scalable Video Coding, SVC) [16] is established and finalized in 2007. An unified scalable bitstream is only encoded once, but it can be adapted to support various multimedia applications with different specifications from mobile phone, personal computer to HDTV.

### 1.2.2 MPEG-2 Multiview Profile (MVP)

MPEG-2 Multiview Profile (MVP) [17] is the first developed standard which describes the bitstream conformance of stereoscopic video. Figure 1.5 illustrates the two-layer coding structure and Codec reference model. In the coding structure, the "temporal scalability" supported by MPEG-2 Main Profile is utilized in MPEG-2 MVP. That is, a view is encoded as the base layer, and the other one is encoded as the enhancement layer. The frames in the enhancement layer is predicted via disparity compensation. In the codec reference model, in addition to the residue which is generated by the difference between the original frame and the motion/disparity compensated frame, the other parts follow the MPEG-2 coding flow. The concept of MPEG-2 MVP is simple, and the only additional functional block is processing



(a)

(b)

Figure 1.5: MPEG-2 Multi-view Profle (MVP). (a) Two-layer coding structure. (b) Codec reference model.

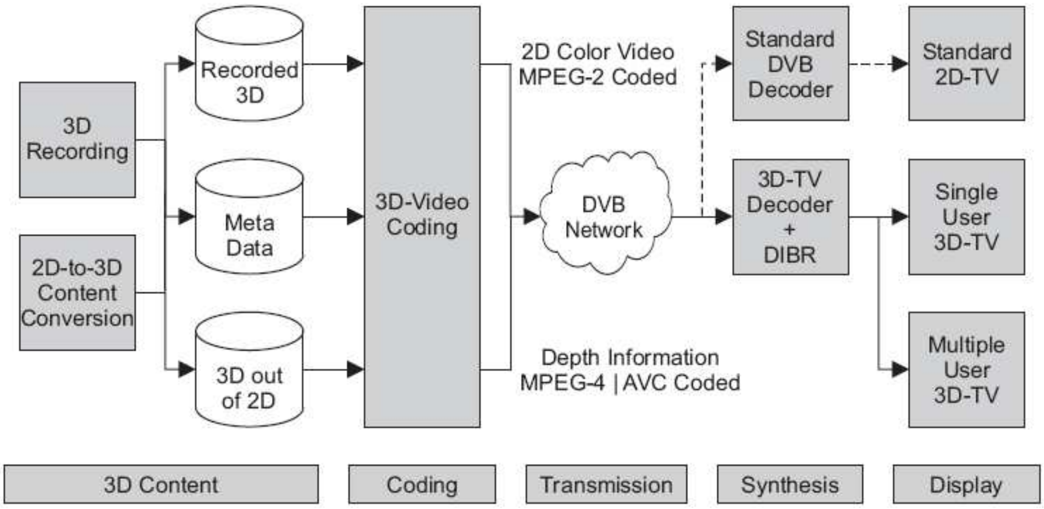


Figure 1.6: The ATTEST signal processing and data transmission chain consisting of five different functional building blocks: 1) 3D content generation; 2) 3D video coding; 3) Transmission; 4) "Virtual" view synthesis; 5) 3D display [1].

unit of syntax element in entropy coding. However, MPEG-2 MVP can not provide the good coding efficiency because the prediction tool defined in MPEG-2 is poor compared with the state-of-the-art coding standard such as H.264/AVC [18].

### 1.2.3 MPEG-C Part3/Advanced Three-Dimensional Television System Technologies (ATTEST)

A 3D broadcasting TV system has been proposed by the European Information Society Technologies (IST) project "Advanced Three-Dimensional Television System Technologies" (ATTEST). Fig. 1.6 shows the system diagram of ATTEST. From this 3D data representation, one or more "virtual" views of a real-world scene can then be generated in real-time at the receiver side by means of so-called depth image-based rendering (DIBR) techniques. The required backwards compatibility to today's 2D digital TV infrastructure is achieved by compressing the video and depth data with existing standards. In ATTEST, the frame data is encoded as MPEG-2 conformance bitstream because of the compatibility with the standard of Digital Video Broadcasting (DVB). The depth map is encoded with H.264/AVC. Therefore, minor overhead of transmission bandwidth ($<$ 20%) is achieved. Besides, MPEG Group have also

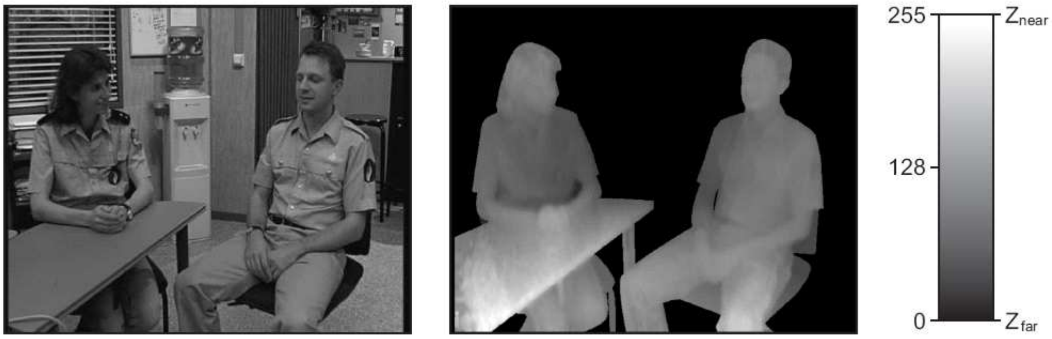


Figure 1.7: MPEG-C Part 3/ATTEST data representation format consisting of: 1) Regular 2D color video in digital TV format; 2) Accompanying 8-bit depth-images with the same spatial-temporal resolution [1].

developed the standard, the so-called MPEG-C Part3, to define the bitstream format of the depth information. The depth images are encoded with 8-bit and the same spatial-temporal resolution as color video, as shown in Fig. 1.7.

Although MPEG-C Part3 and ATTEST belong to two different standards, the motivation is similar. In the receiver side, the decoder decodes the color video and depth maps, and then DIBR renders arbitrary views from the decoded data. This scheme enables 3D video with limited free-view point-video. The benefit of ATTEST is that whether the kind of users' TV is, the bitstream can fit every user's requirement. The application-oriented consideration speeds up the standardization of MPEG-C Part3 and ATTEST. However, the technology of depth map generation is no mature enough. It directly causes the quality degradation of the rendered virtual views in the receiver side.

### 1.2.4 H.264/AVC Multiview High Profile

Multiview video is composed of multiple view channels, as shown in Fig. 1.8. The multiview video coding (MVC) is necessary due to the huge amount of data in an MVC system. Figure 1.9 illustrates the overview of an MVC system. The multiview video is captured by a camera array, and followed by the MVC encoder to execute the data compression for transmission or storage. In the decoder side, reconstructed multiview video can be displayed on various display such as currently commercialized

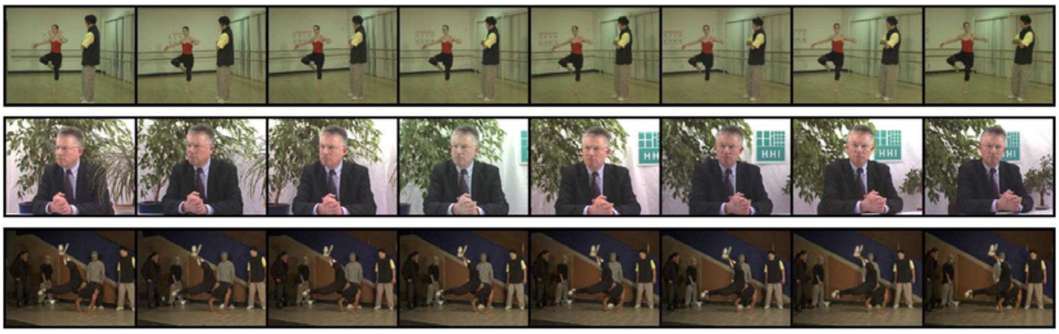


Figure 1.8: Multiview video data set is composed of two to several view channels.

HDTV, developed stereo and multiview 3DTV. Since 2001, MPEG has been working on the exploration of 3D Audio-Visual (3DAV) technology. A lot of tools are already available within MPEG standards, but some of them are incomplete or inefficient for the 3DAV application scenarios. For that reason, MPEG established an Ad-hoc Group (AhG) that has been investigating the topic [19]. The multi-view video coding (MVC) is currently being developed as an extension of the ITU-T Recommendation H.264 — ISO/IEC International Standard ISO/IEC 14496-10 advanced video [20]. The reference model, Joint Multiview Model (JMVM), describes several non-normative encoder issues. There are some new coding tools different from conventional H.264/AVC, such as hierarchical bi-directional frame prediction, illumination compensation, and motion skip mode, etc. In addition to the hierarchical B-frame prediction which is already adopted as an coding tool in H.264/AVC Multiview High Profile, the other issues are still discussed and tested in another new reference model, JMVC.

## 1.3 Research Topics and Contributions

Figure 1.10 shows the main components of MVC. MVC consists of pre-processing, inter-view/temporal/spatial redundancy reduction, and post-processing. Because of the natural characteristics of camera sensor response, the color mismatch between views usually exists. Therefore, the pre-processing stage for color correlation is necessary. Inter-view and temporal redundancy can be removed by disparity estimation (DE) and motion estimation (ME), respectively. Besides, spatial redundancy can be

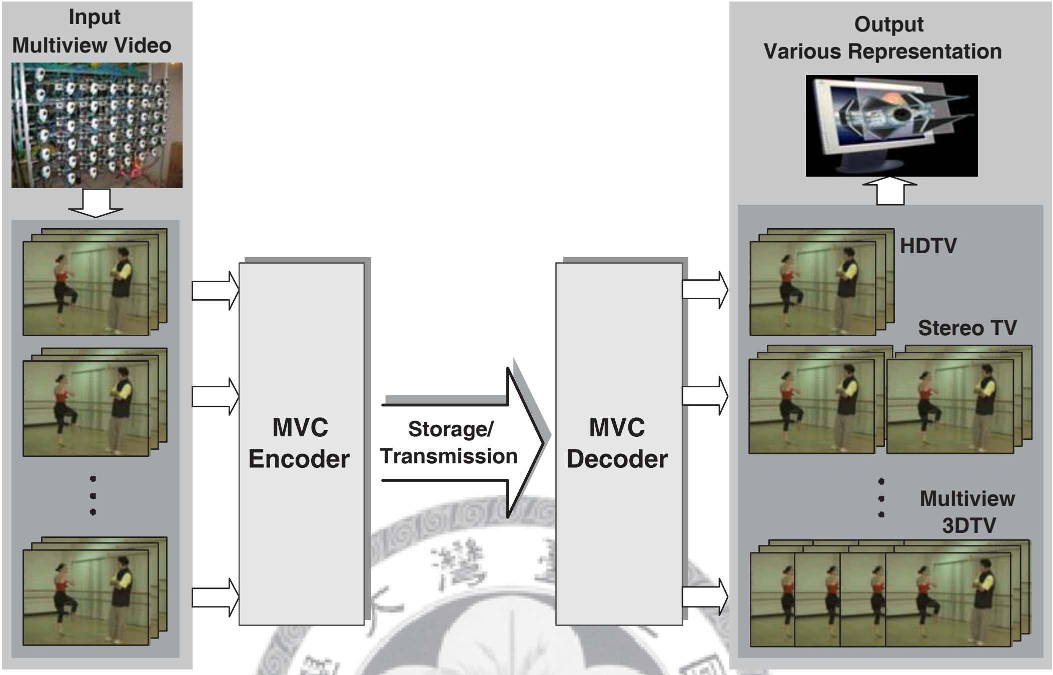


Figure 1.9: Overview of an MVC system.

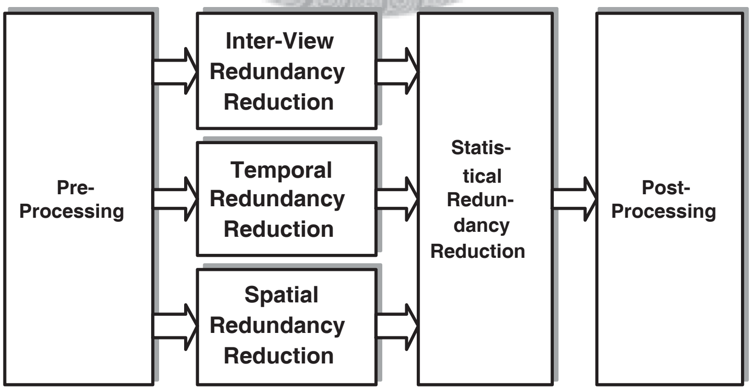


Figure 1.10: MVC consists of pre-processing, inter-view/temporal/spatial redundancy reduction, and post-processing.

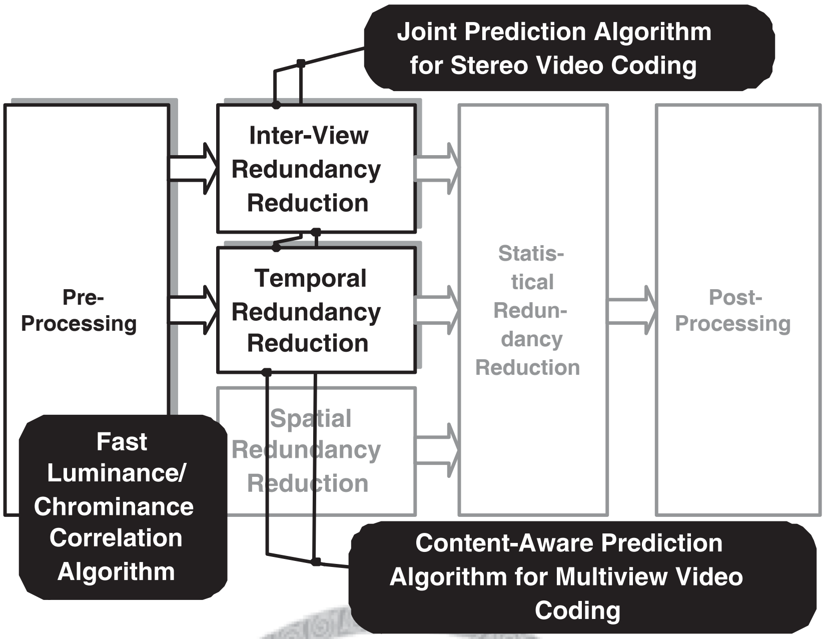

Joint Prediction Algorithm for Stereo Video Coding
Pre-Processing
Inter-View Redundancy Reduction
Temporal Redundancy Reduction
Spatial Redundancy Reduction
Statis-tical Redun-dancy Reduction
Post-Processing
Fast Luminance/ Chrominance Correlation Algorithm
Content-Aware Prediction Algorithm for Multiview Video Coding

Figure 1.11: Research contributions at MVC algorithm level.

reduced by intra prediction in H.264/AVC. The statistical redundancy is removed by entropy coding, and post-processing is achieved by deblocking filter to enhance the subjective quality.

In this dissertation, several research topics of MVC are covered at three different level: algorithm level, VLSI architecture level, and system level. The main target of this research is to enhance the coding efficiency and realize real-time processing capability of MVC systems. At algorithm level, three fast prediction algorithms are developed for stereo video coding and MVC. At VLSI architecture level, two prediction core architecture are proposed and implemented for the most computation-consuming part in stereo and MVC systems. At system level, system design issues are taken into consideration. A memory bandwidth analysis method is presented. Finally, VLSI design of MVC encoder system is proposed. The research contributions are described in the following subsections.

### 1.3.1 Algorithms of Multiview Video Coding

The MVC research is started from developing new prediction algorithms. Figure 1.11 shows the research issues at algorithm level. To calibrate the color mismatch between

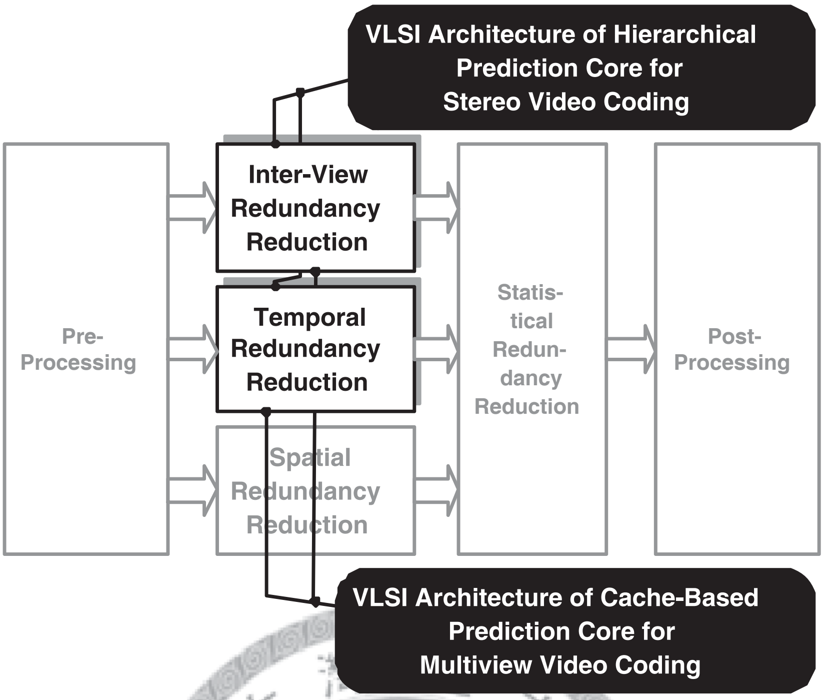

VLSI Architecture of Hierarchical Prediction Core for Stereo Video Coding
Pre-Processing
Inter-View Redundancy Reduction
Temporal Redundancy Reduction
Spatial Redundancy Reduction
Statis-tical Redun-dancy Reduction
Post-Processing
VLSI Architecture of Cache-Based Prediction Core for Multiview Video Coding

Figure 1.12: Research contributions at MVC VLSI architecture level.

input views, we propose a fast luminance and chrominance correlation algorithm. With two to three iteration, the video quality is effectively enhanced, so is the coding efficiency. In addition, for the most computation-intensive part, DE and ME, two fast prediction algorithms are proposed to remove inter-view and temporal redundancy. Joint prediction algorithm is designed for stereo video coding, and it not only reduces huge computational complexity but also enhance the video quality with new coding tools. On the other hand, Content-aware prediction algorithm is proposed for MVC. By utilizing the correlation between views, this algorithm efficiently reduces the ME computation for most view channels while maintains the quality.

### 1.3.2 VLSI Architectures of Multiview Video Coding

Figure 1.12 illustrates the research issues at architecture level. 3D video consists of high-resolution frames in each view channel for 3DTV application. Therefore, the ultra high computation complexity make the real-time processing of encoding hardly realized. We implement the VLSI architectures of prediction cores for stereo and MVC, respectively. The hierarchical prediction core can achieve real-time require-

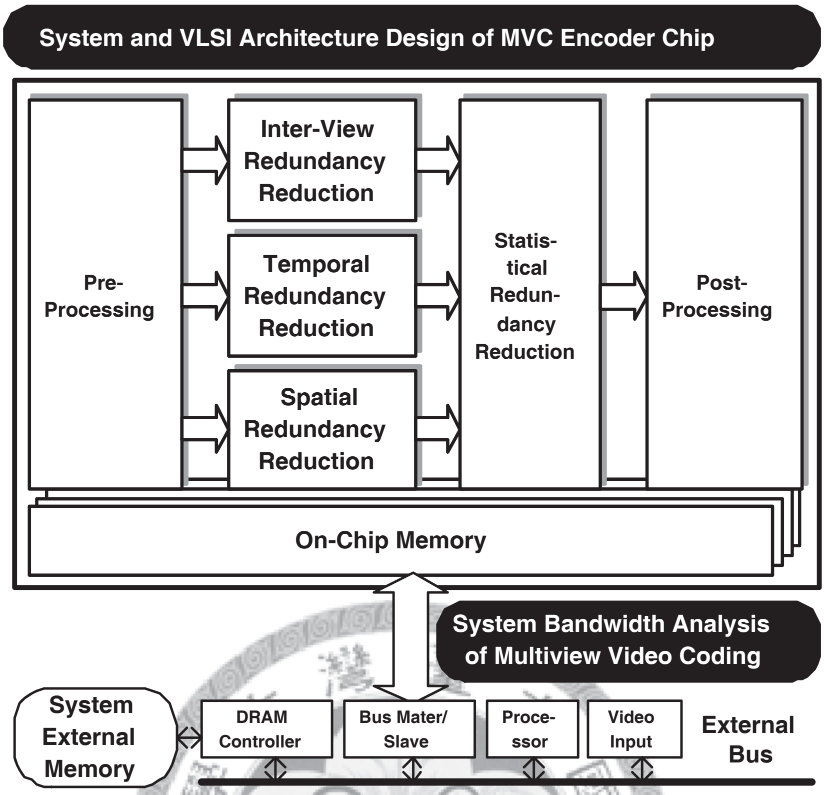

System and VLSI Architecture Design of MVC Encoder Chip
Pre-Processing
Inter-View Redundancy Reduction
Temporal Redundancy Reduction
Spatial Redundancy Reduction
Statis-tical Redun-dancy Reduction
Post-Processing
On-Chip Memory
System Bandwidth Analysis of Multiview Video Coding
System External Memory
DRAM Controller
Bus Mater/ Slave
Proce-ssor
Video Input
External Bus

Figure 1.13: Research contributions at MVC system level.

ment for stereo D1 videos, and the cache-based prediction core can encode multiview HDTV videos in real time. Both architectures are designed according the developed algorithms. Therefore, the inter-view and temporal redundancy are successfully removed with proposed area-efficient VLSI architectures.

### 1.3.3 System Design of Multiview Video Coding

The research of MVC algorithms and architectures are integrated and taken into consideration of system design issues, as shown in Fig. 1.13. There are several design issues to implement the MVC encoder chip, including large on-chip memory requirement and external memory bandwidth. We propose a system bandwidth analysis method to evaluate the system bandwidth for various coding structures. Furthermore, the first MVC single-chip encoder is presented. This chip supports the real-time encoding for one-view 4096×2160, three-view 1920×1080 to seven-view 4096×2160 videos. With this technique, the design challenges for implementing the encoder chips with specifications lower than quad HDTV applications are overcome.

## 1.4 Dissertation Organization

This dissertation is organized as follows. Three parts are included in this dissertation. Part I introduces efficient algorithms for MVC. A fast MVC prediction algorithm which utilizes the content-aware concept is proposed in Chapter 2. After overcoming the design challenges of the most critical prediction part, Chapter 3 presents a fast color correlation algorithm to enhance the quality of input multiview videos. After that, Part II discusses the issues of algorithm and architecture co-design of prediction core for 3D and quad high definition video coding. In Chapter 4, a fast prediction algorithm with new coding tools is proposed to reduce the computational complexity and enhance the coding efficiency simultaneously. To meet the real-time requirement of quad full HD (QFHD) video coding, a prediction-centered fast algorithm is proposed in Chapter 5. Based on this fast prediction algorithm, he VLSI architecture of cache-based prediction core is designed and introduced in Chapter 6. In Part III, the system analysis and architecture design of 4096×2160p multivew video single-chip encoder are developed. Chapter 7 describes the a system bandwidth analysis method with precedence constraint for MVC. Then, the design and implementation issues of MVC encoder chip is presented in Chapter 8. Finally, Chapter 9 summarizes this dissertation, and some future research directions for MVC are proposed.

國立臺灣大學 · 愛國 · 愛人 · 敦品勵學

# Part I

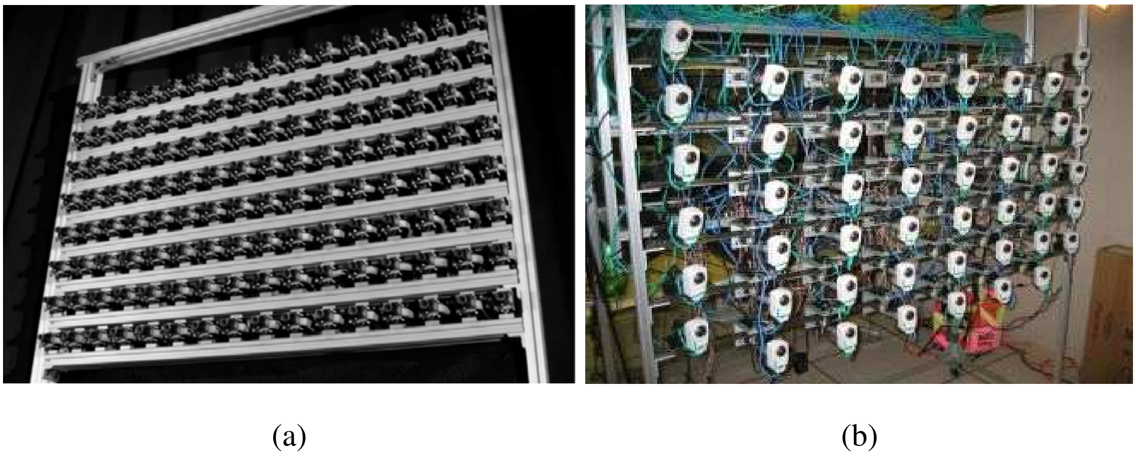
# Efficient Algorithms for Multiview Video Coding

國立臺灣大學 · 敦品勵學 · 愛國愛人

# Chapter 2

# Content-Aware Prediction Algorithm with Inter-View Mode Decision for Multiview Video Coding

3-D video will become one of the most significant video technologies in the next-generation television. Due to the ultra high data bandwidth requirement for 3-D video, effective compression technology becomes an essential part in the infrastructure. Thus multiview video coding (MVC) plays a critical role. However, MVC systems require much more memory bandwidth and computational complexity relative to mono-view video coding systems. Therefore, an efficient prediction scheme is necessary for encoding. In this chapter, a new fast prediction algorithm, content-aware prediction algorithm (CAPA) with inter-view mode decision, is proposed. By utilizing disparity estimation (DE) to find corresponding blocks between different views, the coding information, such as rate-distortion cost, coding modes, and motion vectors, can be effectively shared and reused from the coded view channel. Therefore, the computation for motion estimation (ME) in most view channels can be greatly reduced. Experimental results show that compared with the full search block matching algorithm (FSBMA) applied to both ME and DE, the proposed algorithm saves 98.4–99.1% computational complexity of ME in most view channels with negligible quality loss of only 0.03–0.06 dB in PSNR.



(a) (b)

Figure 2.1: Multiple camera arrays that have been built. (a) 128-camera array [2]. (b) Self-configurable camera array [3].

## 2.1 Introduction

Multiview video can provide users with a sense of complete scene perception by transmitting several views to the receivers simultaneously. It can give users a vivid information about the scene structure. Moreover, it can also provide the capability of 3D perception by respectively showing two of these frames to the eyes. With the technology of 3D-TV [21][22] and free viewpoint TV (FTV) [23][24][25] getting more and more mature, multiview video coding (MVC) draws more and more attention. Besides, some multiple camera arrays have also been proposed for 3D video applications [2][3] as shown in Fig. 2.1. In recent years, JVT/MPEG 3D auido/video (3DAV) group has worked toward the standardization for MVC [26], which also advances the multiview video applications. From the discussion in JVT/MPEG 3DAV meetings, the developed coding scheme for multiview video settings mainly uses H.264/AVC with exploiting temporal and inter-view dependencies [27]. That is, many coding tools of MVC in the related research area are based on the hybrid coding scheme and highly related to H.264/AVC [28].

Although MVC is an emerging technology, huge amount of video data and ultra high computational complexity make it difficult to be realized. An H.264/AVC encoder requires computing power of about 1.3 tera-operations/second (TOPS) on a general-purpose processor to encode single-view HDTV720p videos (1280 × 720,

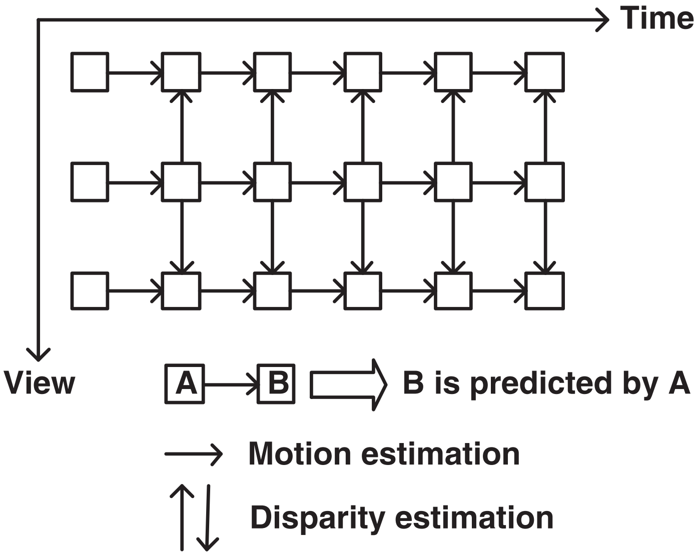


Figure 2.2: Illustration of an example of a three-view coding structure. The white blocks represent frames, and the arrows represent prediction directions.

30 frames/second) in real time [29]. Different from mono-view video coding, disparity estimation (DE) is also utilized to reduce inter-view redundancy in MVC. Many DE algorithms have been proposed [30][31][32][33] to enhance the quality of the depth map for view synthesis or other intelligent video processing. Taking coding efficiency into consideration, block-based DE, like motion estimation (ME), is more appropriate for MVC because it has better compatibility with the existing video coding standards. Consequently, the prediction part, which consists of ME and DE, becomes the most computationally intensive part in an MVC system. Taking a three-view coding structure shown in Fig. 2.2 as an example, an instruction profiling of this coding structure is analyzed, as shown in Table 2.1. It shows that the prediction part occupies 95% computational complexity in an MVC system. The proportion is even higher in some MVC systems with more complex coding structures. Therefore, ultra high computational complexity is a critical design challenge for MVC, especially in the prediction part.

In an MVC system, ME removes the temporal redundancy while DE removes the inter-view redundancy. Because of the setup structure of multiple cameras, there is close relation between motion vectors and disparity vectors in neighboring frames.

Table 2.1: Instruction Analysis of an MVC Encoder with the Coding Structure Shown in Fig. 2.2

| Functions[a] | MIPS[b] | Percentage |
|---|---:|---:|
| Integer-pel ME | 229180.6 | 75.4% |
| Integer-pel DE | 38386.8 | 12.6% |
| Fractional-pel ME/DE | 21396.6 | 7.0% |
| Others[c] | 15183.1 | 5.0% |
| Total | 304147.1 | 100.0% |

[a] The encoding parameters are QVGA, 30 frames/s, ME with search range of [-32, +31] in both vertical and horizontal directions, DE with search range of [-32, +31]/[-8, +7] in the horizontal/vertical direction, and QP=20.

[b] MIPS stands for million instructions per second.

[c] Other modules include Lagrangian mode decision, intra prediction, variable length coding, transform & quantization, and deblocking filter, and so on.

The correlation is shown in Fig. 4.7. It can be described as [34][35]

$$DV_{k-1} + MV_R = MV_L + DV_k. \tag{2.1}$$

By utilizing the correlation between motion and disparity fields, some new coding methods for MVC have been proposed [36][37][38]. Guo et al. have proposed an inter-view motion model to model the temporal motions at different views [36]. “Inter-view direct mode” has been introduced to enhance the coding efficiency. Although the target of 4–6% bit-rate saving is achieved, the complexity is increased due to additional global DE. On the other hand, according to the correlation between views, another kind of redundancy called “computational redundancy” exists in addition to temporal and inter-view redundancies. Based on this concept, a fast prediction algorithm has been proposed to save the computation of ME for stereo video coding in our previous work [37]. However, the coding structures in MVC are more complex than that in stereo video coding. Besides, the previous work cannot deal with variable-block-size ME and complex mode decision. Moreover, Lai et al. have proposed predictive fast motion and disparity search. They track along the first estimated field (disparity/motion field) to get candidate vectors for the other field (motion/disparity field) [38]. Great computational complexity is saved with quality

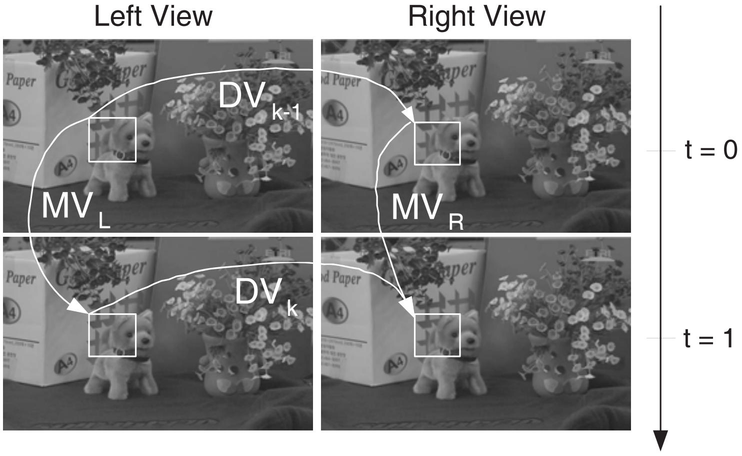


Figure 2.3: The relation between disparity vectors and motion vectors.

loss of 0.1–0.2 dB in PSNR. However, the ME with variable block size is not taken into consideration in their predictive search as well. In summary, all of the previous work only reuses motion or disparity vectors from other views or time slots. There are still some inter-view coding information, such as rate-distortion cost and coding modes. They can be further adopted for complex mode decision and complexity reduction.

In this chapter, a new fast prediction algorithm, content-aware prediction algorithm (CAPA) with inter-view mode decision, is proposed for MVC. Based on the fact that the video contents are highly related between view channels, the proposed algorithm greatly reduces computational complexity while maintains video quality. The remainder of the chapter is organized as follows. Section 2.2 describes the analysis of macroblock prediction modes in MVC. Next, the proposed CAPA is presented in Section 3.2. Section 5.5 shows the simulation results. Finally, Section 7.5 summarizes this chapter.

## 2.2 Analysis of Macroblock Prediction Modes in MVC

First, the coding structure of MVC is introduced. Many coding structures have been evaluated [39]. However, there is no unique coding structure which is appropriate for every video sequence. The selection of coding structures highly relies on the video contents and the corresponding camera setup. Figure 2.4 shows the illustration of

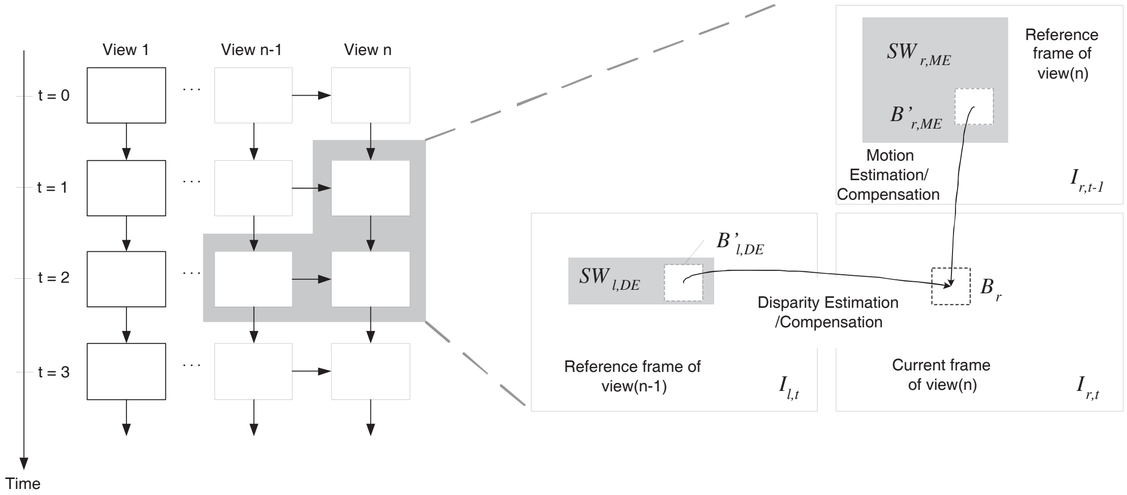

View 1
View n-1
View n
t = 0
t = 1
t = 2
t = 3
Time
SW r,ME
B' r,ME
Reference frame of view(n)
Motion Estimation/ Compensation
I r,t-1
B' l,DE
SW l,DE
Disparity Estimation /Compensation
B r
Reference frame of view(n-1)
I l,t
Current frame of view(n)
I r,t

Figure 2.4: Illustration of an MVC structure. The arrows represent the prediction directions, and the gray regions are the search windows for $B_r$.

one coding structure, where the prediction directions of ME and DE are represented by arrows. For convenience of interpretation, the view channel $n-1$ is regarded as the left channel, and the view channel $n$ is regarded as the right channel. There are two types of compensated blocks. They are the motion-compensated blocks and the disparity-compensated blocks, which are illustrated as $B'_{r,ME}$ and $B'_{l,DE}$ in Fig. 2.4, respectively. According to Lagrangian mode decision, the best type of compensated blocks is selected. For each macroblock in the current frame, the costs of ME and DE are computed by

$$Cost_{ME} = \min_{B'_{r,ME} \in SW_{r,ME}(B_r)} \{ \sum_{(k,k') \in (B_r, B_{r,ME})} |I_{r,t}(k) - I_{r,t-1}(k')| + \lambda \cdot Rate \}, \qquad (2.2)$$

$$Cost_{DE} = \min_{B'_{l,DE} \in SW_{l,DE}(B_r)} \{ \sum_{(k,k') \in (B_r, B'_{l,DE})} |I_{r,t}(k) - I_{l,t}(k')| + \lambda \cdot Rate \}, \qquad (2.3)$$

where $Cost_{ME}$ and $Cost_{DE}$ are the minimum costs of motion-compensated and disparity-compensated blocks, respectively. $B_r$ is the current block in the right channel. $B'_{r,ME}$ is a block of the reference frame in the right channel. $B'_{l,DE}$ is a block of the reference frame in the left channel. $SW_{r,ME}(B_r)$ and $SW_{l,DE}(B_r)$ are the search windows for the current block $B_r$. After ME and DE, the best matched blocks in the two search windows can be derived. Then the final prediction mode can be decided by selecting the one with lower cost.

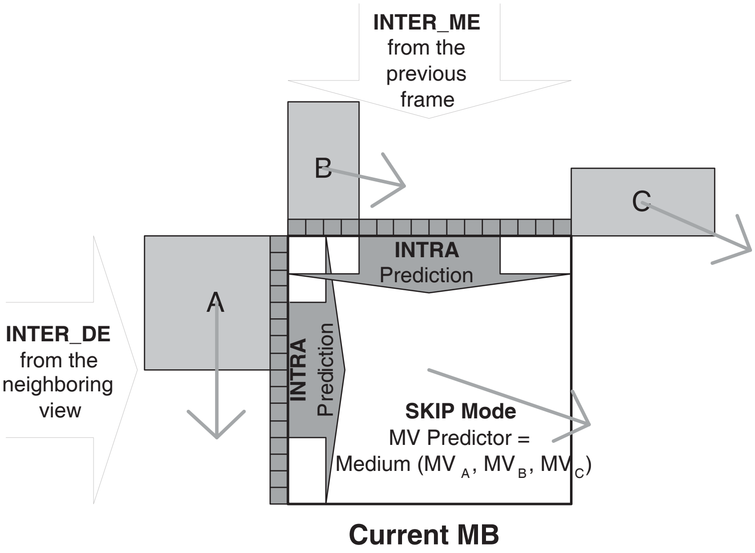
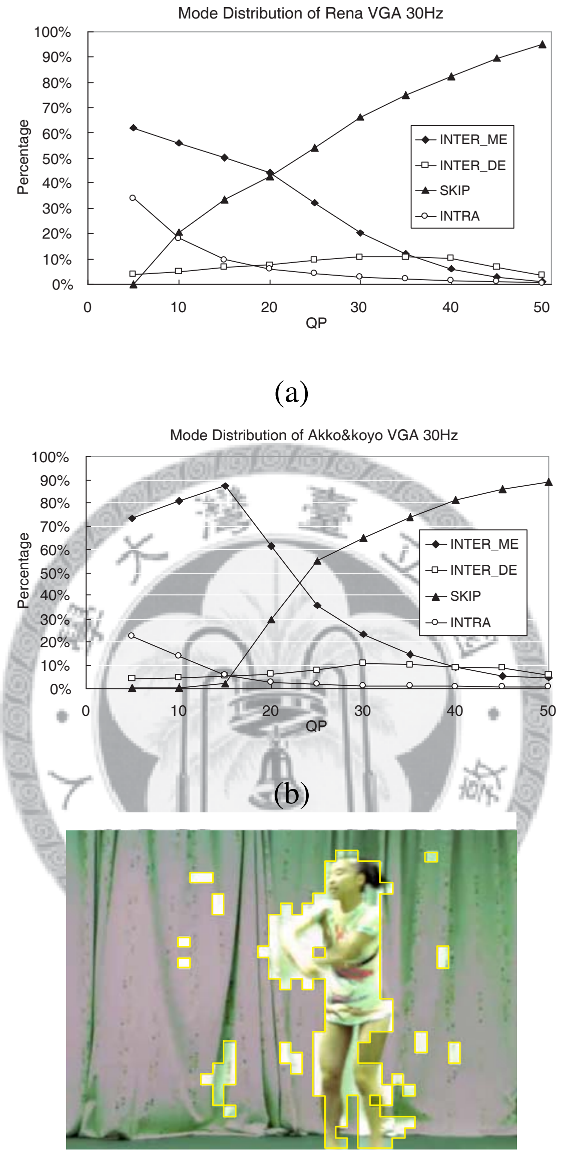


Figure 2.5: The current macroblock can be predicted by various prediction modes. These prediction modes are classified into four categories: INTER_ME, INTER_DE, INTRA, and SKIP modes.

There are several prediction modes defined in H.264/AVC standard. In our analysis of mode distribution, the prediction modes are classified into four categories, that is, INTER_ME, INTER_DE, INTRA, and SKIP modes. As shown in Fig. 2.5, the current macroblock can be predicted by ME from the reference frame in the same view channel, where INTER_ME mode can remove temporal redundancy. On the other hand, INTER_DE mode can remove inter-view redundancy by DE from the reference frame in the neighboring view channel. If the inter prediction cannot predict well, INTRA mode can predict the current macroblock by utilizing boundary pixels in the neighboring macroblocks. Moreover, SKIP mode utilizes the motion vector predictor to predict the current macroblock without performing inter prediction. It not only reduces the computational complexity but also saves the coding bits for motion vectors. The mode decision between INTER_ME and INTER_DE is closely related to video contents [37]. Therefore, the mode classification can reflect the features of video contents.

According to the classification, Fig. 2.6 shows the mode distribution with various quantization parameters (QPs). It shows that the distribution of INTER_ME and SKIP mode has larger variation with various QPs. SKIP mode is the dominant



(a)

(b)

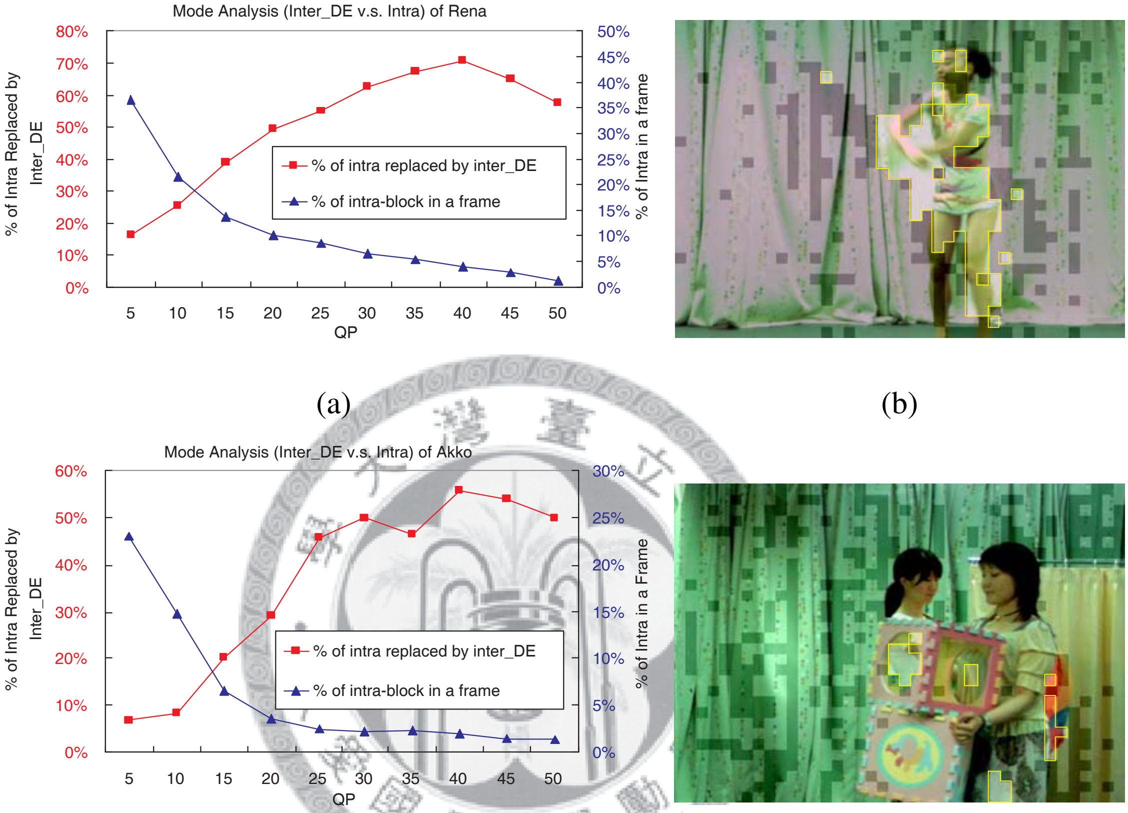
(c)

Figure 2.6: Mode distribution analysis of two test sequences. Four main prediction modes are analyzed for sequences (a) “Rena” and (b) “Akko&Kayo.” (c) Illustration of the compensated block types. The highlighted blocks with yellow boundaries are predicted by INTER_DE mode.

mode at lower bit-rates (high QPs), while INTER_ME and INTRA modes are dominant at higher bit-rates (low QPs). The distribution is similar to that in mono-view video coding. The main difference between mono- and multiview video coding is INTER_DE mode, which is used in 5–10% macroblocks in a frame. The percentage of INTER_DE-mode macroblocks relies on the video contents. As shown in Fig. 2.6 (c), the moving objects are usually predicted by INTER_DE because INTER_ME can not predict well in the areas.

It is observed that certain types of macroblocks which are originally encoded by INTRA mode in mono-view video coding are encoded by INTER_DE mode in MVC. Figure 2.7 shows the statistics of the ratio that INTRA mode is replaced by INTER_DE mode after applying DE. In the cases of median and low bit-rates, over 50% macroblocks which are originally INTRA-coded in mono-view video are instead predicted by INTER_DE mode in MVC. The video contents of these macroblocks usually contain objects with fast motion or occlusions, as shown in Fig. 2.7 (b) and (d). In summary, in an MVC system, the most general types of video contents are predicted by INTER_ME and SKIP modes, while INTRA mode can predict the macroblocks which contain more homogeneous or textural video contents. In addition, some complex video contents with fast moving objects or occlusions can be coded with INTER_DE mode.

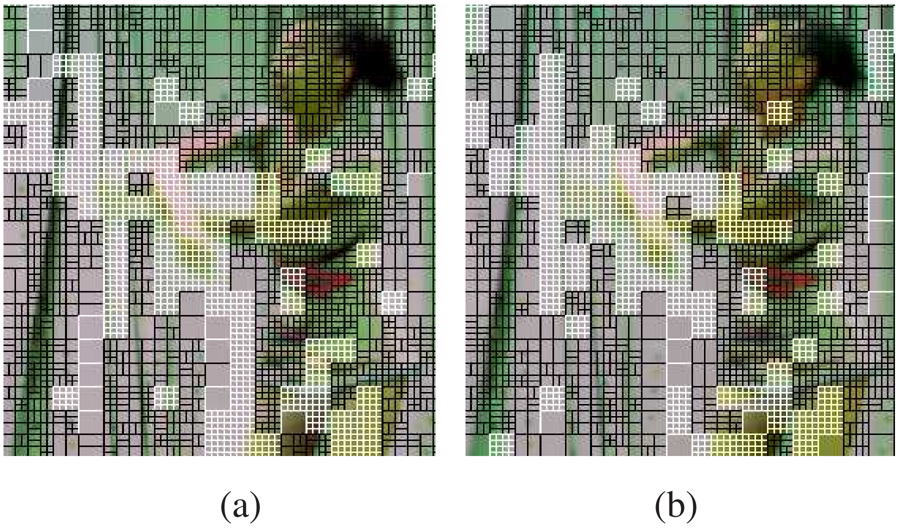
It is also observed that the mode distribution of two views are very similar. In mono-view video coding, there are many coding tools adopted to extract data redundancies and remove them. When the video contents are extended from single view to multiple views, another data redundancy appears. On the conditions that cameras are setup with close parallelized structure, the video contents of different views are usually similar. This inter-view similarities exist not only in the video contents but also in the prediction modes. An example is shown in Fig. 2.8. Two views are encoded separately by an H.264/AVC encoder. The macroblock partition is marked on the reconstructed frames. Black and white blocks represent inter- and intra-predicted blocks respectively. It shows that the inter-view correlation is high. In other words, if the correlation is successfully explored, the computational complexity of the prediction part in MVC can be greatly reduced.



(a) (b)

(c) (d)

Figure 2.7: Statistics of ratio that INTRA mode is replaced by INTER_DE mode after applying DE. Disparity compensated macroblocks which are originally coded with INTRA mode in mono-view video coding are highlighted in subjective views (areas with yellow boundaries). Dark macroblocks are coded with INTRA mode in both cases of mono- and multi-view video coding. (a) Mode analysis of “Rena.” (b) Coding mode illustration of (a) with $QP = 10$. (c) Mode analysis of “Akko&Kayo.” (d) Coding mode illustration of (c) with $QP = 5$.



(a) (b)

Figure 2.8: Illustration of macroblock partition after variable-block-size ME. Two views are independently encoded without DE. (a) Left view. (b) Right view.

The analysis of macroblock prediction modes for MVC is summarized as follows.

- SKIP mode provides good coding performance and requires little computational complexity in both mono- and multi-view video coding.
- INTRA mode tends to be replaced by INTER_DE mode in MVC when the current macroblock contains objects with fast motion or occlusions.
- In most cases, the video contents are similar between view channels. It results in the similar mode distribution between view channels. Therefore, there exists computational redundancy, and the inter-view information can be obtained with DE to predict the coding information. An efficient prediction algorithm with content-aware functionalities can effectively save the unnecessary computation.

## 2.3 Proposed Content-Aware Prediction Algorithm (CAPA) With Inter-View Mode Decision

According to the analysis in the previous section, the content-aware prediction algorithm (CAPA) with inter-view mode decision is proposed to save unnecessary computational load by exploiting the correlation between view channels. By utilizing various features of video contents in the coded view channels, macroblock coding modes and their corresponding motion vectors can be predicted with the aid of DE and the coding information of neighboring views. Therefore, ME computational complexity can be greatly reduced. In this section, the system architecture of the multiview hybrid coding system is introduced first, followed by the details of the proposed algorithm.

### 2.3.1 System Architecture

The block diagram of the multiview video encoder with the proposed CAPA is shown in Fig. 8.4. The encoder adopts the coding tools defined in H.264/AVC standard. Input views are classified into two types of view channels, the primary channel and the secondary channel. A view channel is regarded as a primary channel if no reconstructed frames in other view channels are used for reference when performing mode decision. Therefore, there are no DE operations in primary channels. The coding flow of a primary channel is identical to the flow of mono-view video coding. The block engine includes quantization, transform, and deblocking filter, etc. After the Lagrange mode decision, the coding information, including the rate-distortion costs, the optimum macroblock coding mode, and the corresponding motion vectors of the primary channel are stored for the proposed CAPA. The main difference between the primary and secondary channels is the CAPA part, which contains DE, twin-MB selection, inter-view mode decision, and content-aware ME. Each of them is introduced in the following subsections. In CAPA, DE is performed prior to ME. The purpose of performing DE first is to extract the correlation between views, and the coding information of the corresponding neighboring coded view can be retrieved. With the corresponding coding information, the inter-view mode decision part decides the most probable coding mode for the current macroblock. The most probable coding mode is one of the modes described in Section 2.2, that is, INTER_ME, INTER_DE, SKIP, or INTRA mode. If INTER_ME is chosen as the most probable coding mode by inter-view mode decision, it means ME is further required for better coding efficiency of the current macroblock, and thus proposed content-aware ME

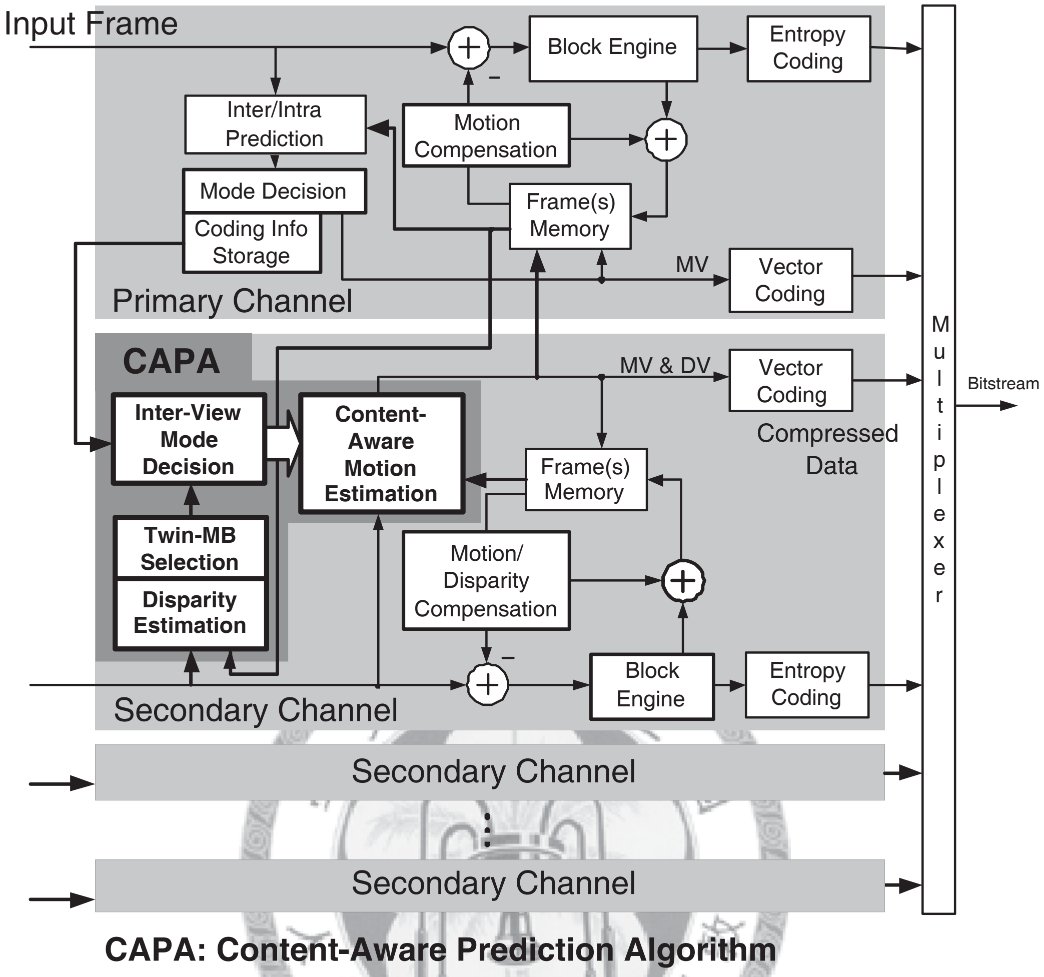


Figure 2.9: Block diagram of the multiview video encoder with the proposed CAPA.

is performed. Content-aware ME is a predictor-centered ME algorithm, and it also utilizes the inter-view coding information. Note that, the numbers of primary and secondary channels are decided according to the coding structure. The numbers of secondary channels are normally much more because DE can effectively enhance coding efficiency [27]. After all views are encoded, the compressed bitstream of each channel is assembled and transmitted.

### 2.3.2 Disparity Estimation and Selection of Twin-Macroblock

DE is performed between the reference frame in the primary channel and the current macroblocks in the secondary channel. The minimum rate-distortion cost, $Cost_{DE}$, is decided by (2.3). The macroblocks in the primary channel are overlapped by the corresponding best matched block indicated by a disparity vector. And among the macroblocks the one with the largest overlapped area is called the "twin-macroblock," as shown in Fig. 2.10. To predict the most probable coding mode for the current macroblock in a secondary channel, it is required to retrieve the related coding information from the twin-macroblock in the coded primary channel. As illustrated in Fig. 2.10, when performing DE, the corresponding best matched block indicated by the disparity vector is derived by

$$\widehat{B_p} = \arg \min_{B'_p \in SW_p(B_s)} \{ \sum_{(k,k') \in (B_s, B'_p)} |I_s(k) - I_p(k')| \}, \tag{2.4}$$

where $B_s$ represents the current macroblock in a secondary channel $I_s$, and $B'_p$ represents the search candidates located in the search window in the primary channel $I_p$. Therefore, MB2 is regarded as the corresponding twin-macroblock of the current macroblock in Fig. 2.10. Note that the rate part of the block-matching cost is not considered here for deriving the twin-macroblock in the primary channel. However, the best disparity compensated block for coding can still be searched by Lagrangian mode decision at the same time without additional computation. The coding information of the twin-macroblock is then stored for the following inter-view mode decision and content-aware ME.

### 2.3.3 Inter-View Mode Decision

After the selection of a twin-macroblock, the coding information of the twin-macroblock, which includes the rate-distortion cost, the optimum macroblock coding mode, and the corresponding motion vectors, is retrieved. The purpose of inter-view mode decision is to choose the most probable coding mode among INTER_ME, INTER_DE, SKIP, and INTRA modes. SKIP mode is a useful and simple coding tool in H.264/AVC, where the motion vector predictor is adopted for the current macroblock to generate a compensated block. Therefore, the ME computation of a macroblock can be entirely saved if SKIP mode can be pre-decided. SKIP mode is also effective in the multiview video encoder. On the other hand, INTRA mode is chosen by a lot of macroblocks in a frame in the condition of high bit-rate, as shown in Fig. 2.6. Therefore, if INTRA mode can be pre-decided, many computation operations for ME can be saved by utilizing the correlation between views. In short, the unnecessary computation for ME

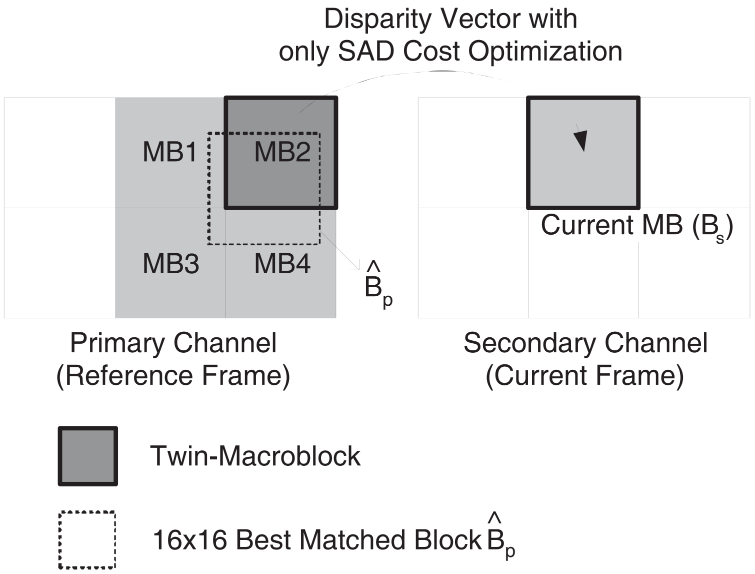


Figure 2.10: A twin-macroblock is the macroblock in the primary channel which is overlapped by the corresponding best matched disparity compensated block with the largest overlapped area. In this case, MB2 is the twin-macroblock.

can be saved if SKIP, INTRA, or INTER_DE mode is chosen. Therefore, inter-view mode decision can be regarded as an early termination scheme for the following ME.

Figure 4.18 shows the decision and data flow of the proposed inter-view mode decision. Solid lines and grey blocks represent the decision flow, and dotted lines represent the data flow. First, intra prediction is performed, and the optimum rate-distortion cost among several modes in intra prediction, $Cost_{INTRA}$, is derived. It is followed by the cost computation of SKIP mode, and $Cost_{SKIP}$ is then derived. $Cost_{INTRA}$ and $Cost_{SKIP}$ are compared with $Cost_{DE}$, which is derived from DE. If $Cost_{SKIP}$ or $Cost_{INTRA}$ is the smallest, the mode of the twin-macroblock is checked. If the twin-macroblock is also coded by the same mode, it implies that it has high possibility to be the best coding mode for the current macroblock. Then the rate-distortion cost of the twin-macroblock, $Cost_{TMB}$, is compared with $Cost_{SKIP}$ or $Cost_{INTRA}$. Finally, the current macroblock is predicted by SKIP/INTRA mode if $Cost_{SKIP}/Cost_{INTRA}$ is still smaller than $Cost_{TMB}$. Otherwise, the current macroblock is assigned to INTER_ME mode and the following content-aware ME is performed.

On the other hand, if $Cost_{DE}$ is the smallest among three coding modes, the data flow is different from the other cases. According to the analysis introduced in Section

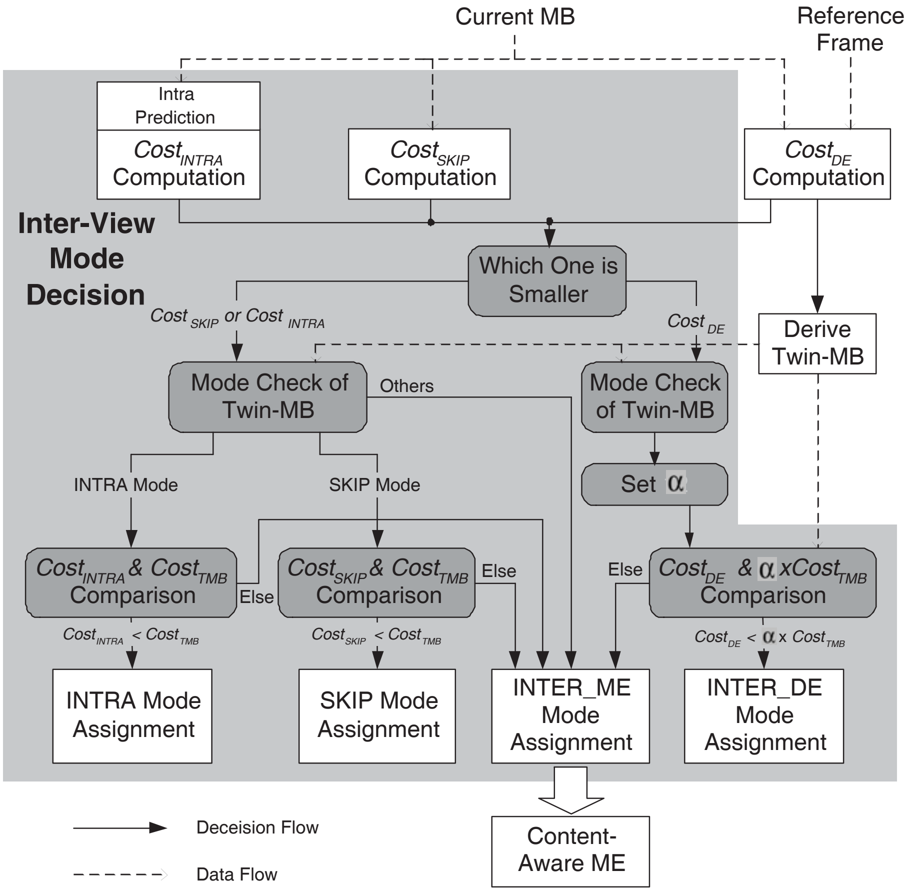


Figure 2.11: Decision and data flows of inter-view mode decision.

2.2, macroblocks contain objects with fast motion tends to be encoded by INTER_DE mode. In addition, INTRA mode tends to be replaced by INTER_DE mode in MVC when the current macroblock contains objects with fast motion or occlusions. Based on these two concepts, the coding modes of the twin-macroblock and the neighboring coded macroblocks are utilized and checked. A parameter, $TMB_{INTRA}$, is defined as follows,

$$TMB_{INTRA} = \begin{cases} 1, & \text{if the twin-macroblock is INTRA-coded,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

$TMB_{INTRA}$ shows whether the twin-macroblock is INTRA-coded or not. Then INTER_DE mode assignment is described by the following equation,

$$Mode = \begin{cases} INTER\_DE, & \text{if } Cost_{DE} < \alpha \times Cost_{TMB}\ , \\ INTER\_ME, & \text{otherwise.} \end{cases} \tag{2.6}$$

In the above equation, a parameter set, $\alpha \in \{\alpha_0, \alpha_1, ..., \alpha_7\}$, is defined to adjust the threshold of the early termination of ME, as shown in Table 2.2. $NMBC_n$ stands for the count of neighboring macroblocks which are encoded by INTER_DE mode. The neighboring macroblocks are the left, top, and top-right macroblocks relative to the current macroblock. Therefore, $NMBC_n \in \{0, 1, 2, 3\}$. The value of $\alpha$ relies on $TMB_{INTRA}$ and $NMBC_n$. $\alpha$ is bigger in the case that the twin-macroblock is coded by INTRA mode. Similarly, the more neighboring macroblocks coded by INTER_DE mode are, the bigger $\alpha$ is. In our simulation, the values of $\alpha_0$, $\alpha_1$,..., $\alpha_7$ are empirically chosen between 0.1 and 2.0, that is, $\{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\} = \{0.4, 1.0, 1.5, 2.0, 0.1, 0.7, 1.2.1.7\}$. Therefore, the coding mode can be decided after inter-view mode decision. If the current macroblock is assigned to INTRA, SKIP, or INTER_DE mode, the following ME operation can be skipped. Otherwise, content-aware ME is performed.

### 2.3.4 Content-Aware Motion Estimation

To further reduce the computational complexity of ME in the secondary channels, content-aware ME is proposed. As shown in Fig. 2.12, there are seven macroblock partition types according to their block sizes such as 16 × 16, 8 × 8, and 4 × 4, etc.

Table 2.2: Definition of the Parameter Set α

| $TMB_{INTRA}$ | $NMBC_n$ | $\alpha$ |
|---|---|---|
| $TMB_{INTRA}$ $= 1$ | 0 | $\alpha_0$ |
| | 1 | $\alpha_1$ |
| | 2 | $\alpha_2$ |
| | 3 | $\alpha_3$ |
| $TMB_{INTRA}$ $= 0$ | 0 | $\alpha_4$ |
| | 1 | $\alpha_5$ |
| | 2 | $\alpha_6$ |
| | 3 | $\alpha_7$ |

$$\alpha_0 < \alpha_1 < \alpha_2 < \alpha_3, \alpha_4 < \alpha_5 < \alpha_6 < \alpha_7;$$
$$\alpha_0 > \alpha_4, \alpha_1 > \alpha_5, \alpha_2 > \alpha_6, \alpha_3 > \alpha_7.$$

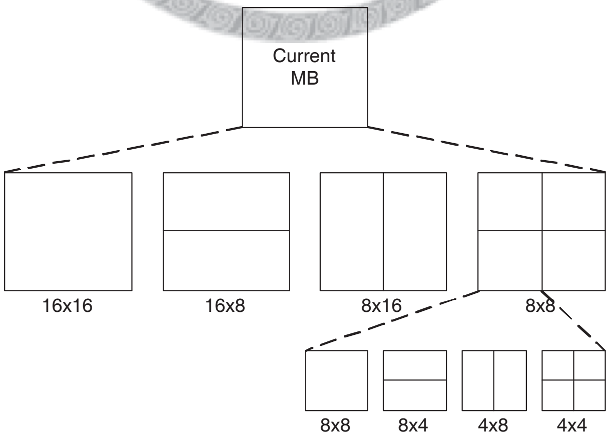
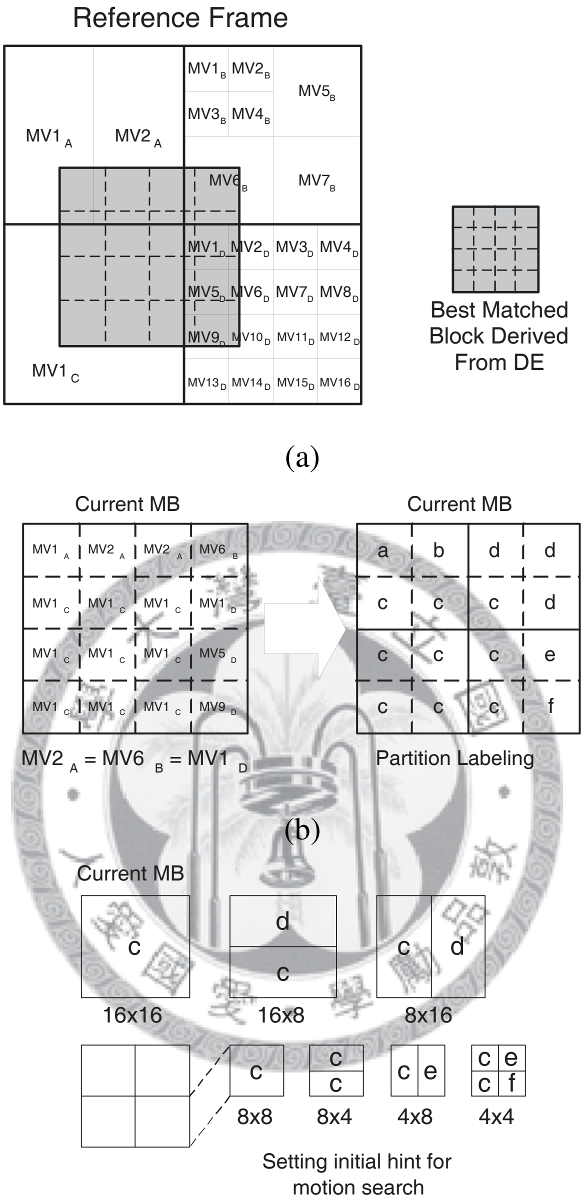


Figure 2.12: The macroblock partition rule defined in H.264/AVC. It follows the hierarchical and symmetric manner.



(a)

(b)

(c)

Figure 2.13: Illustration of inter-view motion vector prediction. (a) The coding information of the reference frame to be used is stored in the memory. The 16 × 16 area of the best matched block derived from DE is split into sixteen 4 × 4 sub-blocks. (b) Each sub-block is assigned a motion vector. (c) An initial guess of the motion vector, CAPA motion vector predictor, is set for each macroblock type according to partition labeling.

in H.264/AVC [40]. Content-aware ME is proposed to predict the macroblock partition and the corresponding motion vectors of the current macroblock. The proposed algorithm consists of two parts, inter-view motion vector prediction and motion vector refinement with small search range. The illustration of inter-view motion vector prediction is shown in Fig. 2.13. After the frame in the primary channel is encoded, the coding information is stored in the memory. The location of the best matched block has already been derived from DE shown as the grey area in Fig. 2.13 (a). The grey area is split into sixteen $4 \times 4$ sub-blocks. Each sub-block covers a $4 \times 4$ area in the reference frame, and then it is assigned with the motion vector of the $4 \times 4$ area in the coded reference frame. Note that if the $4 \times 4$ area contains more than one different motion vectors, the assigned motion vector is the motion vector of the coded sub-block with the largest overlapped area by the best matched block. To prevent prediction error propagation, there are not any early termination and fast prediction schemes applied in the primary channel, which means all kinds of cost must be calculated in the primary channel. Besides, no matter what kind of mode is selected for coding, the best inter prediction mode and its corresponding motion vectors are stored in the primary channel. Therefore, if the covered macroblock in the reference frame is predicted by INTRA or SKIP mode, the macroblock partition and its corresponding motion vectors are still available for the proposed algorithm.

After each $4 \times 4$ sub-block is assigned a motion vector, the process of “partition labelling” begins. The sub-blocks with the same motion vectors are assigned to the same label, as the labels “a, b, c, d, e, f” shown in Fig. 2.13 (b). Next, an initial guess of the motion vector, which is called “CAPA motion vector predictor,” is set for each marcoblock type according to partition labelling. An example is shown in Fig. 2.13 (c). To provide a good initial guess of a motion vector, the most representative value should be chosen. For example, when setting the CAPA motion vector predictor for a $16 \times 16$ block, the value of the label which appears the most times is chosen as an initial guess, which is “c” in Fig. 2.13 (c).

In addition to the initial guess provided from inter-view coding information, the motion vectors of the left, top, top-right neighboring macroblocks, and zero motion vector, are also adopted as the initial guesses to enhance the coding efficiency. That

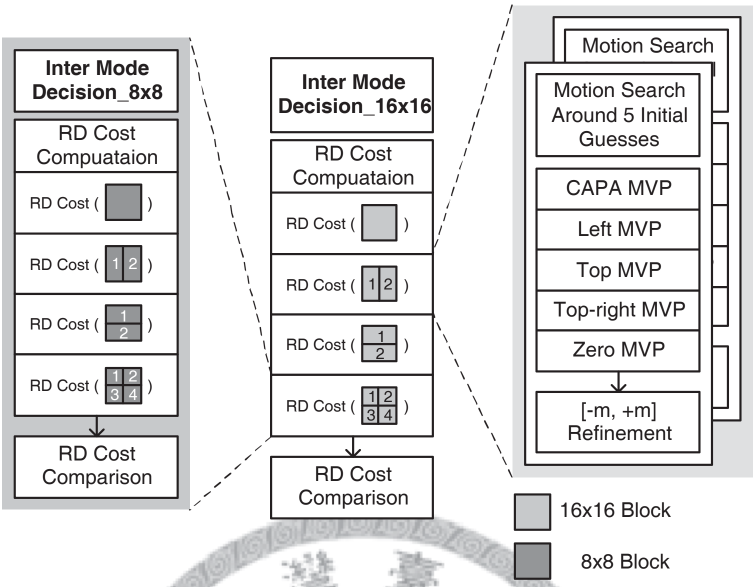
Table 2.3: Multiview Video Sequences for the Experiments

| Sequences | Frame size | ME search range [x, y] | DE search range [x, y] | Coding structure |
|---|---|---|---|---|
| Akko&Kayo | 640 × 480 | [±32, ±32] | [±32, ±8] | Fig.2.15 (a) |
| Ballroom | 640 × 480 | [±32, ±32] | [±32, ±8] | Fig.2.15 (b) |
| Exit | 640 × 480 | [±32, ±32] | [±32, ±8] | Fig.2.15 (c) |
| Rena | 640 × 480 | [±32, ±32] | [±32, ±8] | Fig.2.15 (d) |

is, for each macroblock type, there are five initial guesses of motion vectors. The optimum initial guess is chosen by Lagrange mode decision, and then the refinement with a small search range is performed around the optimum initial guess. Figure 2.14 shows the data flow and the corresponding pseudo codes of searching the optimum motion vectors. Because the proposed algorithm is a predictor-centered ME, the required search candidates are much less than that for full search block matching algorithm (FSBMA). Therefore, computational complexity can be greatly reduced.

## 2.4 Simulation Results

The proposed algorithm is implemented by modifying the MVC-configuration in JSVM4.5 [41]. Sequences “Akko&Kayo,” “Ballroom,” “Exit,” and “Rena,” with size 640×480 are tested. They are standard sequences released by JVT/MPEG 3DAV Group [27]. Two- and three-view channels of these sequences are chosen for simulation. The four coding structures adopted in our experiments are shown in Fig. 2.15. The grey blocks represent the frames in the primary channel, and the white blocks represent the frames in the secondary channels. The test condition is shown in Table 2.3. The search ranges of DE and ME are both [-32, +32] in the horizontal direction, while the search range of DE is [-8, +8] in the vertical direction. The search range of DE is not a square because the cameras to capture these sequences are parallel-structured [27]. Thus the candidate blocks can be assumed in a belt-shape region [42]. Note that in the simulation of ME complexity, the index “search candidate per macroblock” is adopted to make the data independent of various hardware platforms.

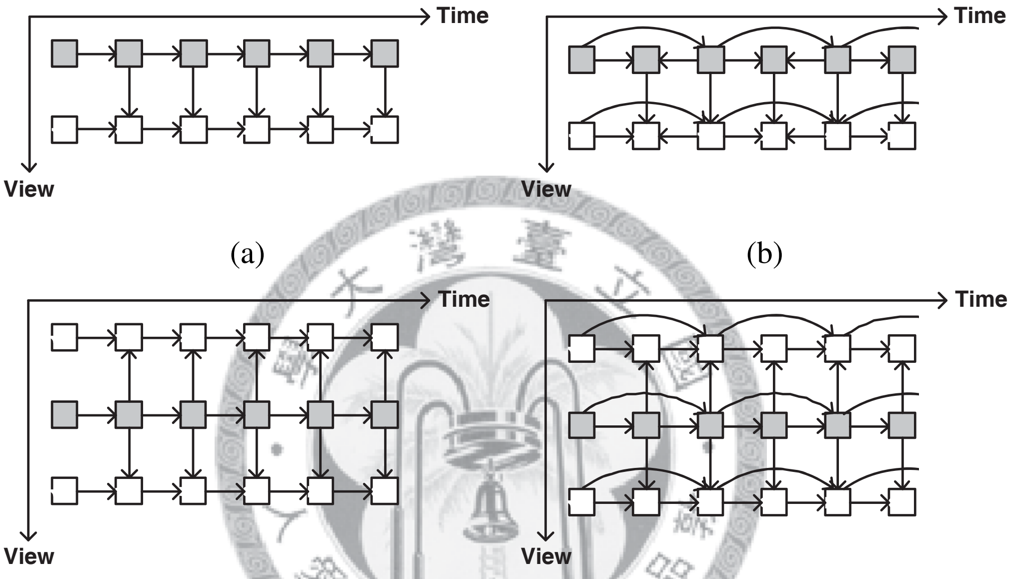


(a)

```
Inter_Mode_Decision_16x16 ( Current MB )
{
        RD_Cost ( □ ) = Motion_Search ( 16x16_block )
        RD_Cost ( 1|2 ) = Motion_Search ( 8x16_block(1) ) + Motion_Search ( 8x16_block(2) )
        RD_Cost ( 1/2 ) = Motion_Search ( 16x8_block(1) ) + Motion_Search ( 16x8_block(2) )
        Loop ( 8x8_sub_block (n), n = 1 ~ 4 )
                Inter_Mode_Decision_8x8 ( 8x8_sub_block (n) )
        RD_Cost ( ⊞ 1 2 3 4 ) = Motion_Search ( 8x8_block(1) ) + Motion_Search ( 8x8_block(2) )
                         + Motion_Search ( 8x8_block(3) ) + Motion_Search ( 8x8_block(4) )
        Compare_RD_Cost ( □ ◫ ⊟ ⊞ )
        Return Optimum_MV&Cost
}
Motion_Search ( MB partition type )
{
        Loop ( Initial_guess(n), n = 1 ~ 5 )
        {
                Cost_computation()
                // n = 1  MV from CAPA motion vector predictor
                // n = 2  MV of left block
                // n = 3  MV of top block
                // n = 4  MV of top-right block
                // n = 5  Zero MV
        }
        Refinement ( Optimum_initial_guess )
        {
                Loop ( Refine_range [-m, +m] )
                        Cost_computation()
        }
        Return Optimum_MV&Cost ( MB partition type )
}
```

(b)

Figure 2.14: (a) Data flow and (b) pseudo codes of searching the optimum motion vectors. Note that the motion search is performed for each macroblock type.



(a)

(b)

(c)

(d)

Figure 2.15: Four coding structures for experiments. The grey blocks represent the frames in the primary channel, and the white blocks represent the frames in the secondary channels. (a) Two views with IPPP structure. (b) Two views with IBPBP structure. (c) Three views with IPPP structure. (d) Three views with IBPBP structure.

Table 2.4: Complexity Reduction and Quality Drop of Inter-view Mode Decision

| Sequences | QP | Complexity Reduction | Quality Drop |
|---|---|---|---|
| Akko&Kayo | 45 | 20.4% | –0.017 dB |
| Ballroom | 30 | 61.8% | –0.032 dB |
| Exit | 35 | 73.4% | –0.035 dB |
| Rena | 40 | 47.5% | –0.015 dB |

Table 2.5: Distribution of the Initial Guess Which Is Chosen for Further Refinement

| Sequence | Akko&Kayo | Ballroom | Exit | Rena |
|---|---|---|---|---|
| CAPA MV predictor | 81.6% | 88.9% | 81.7% | 86.2% |
| Left MV predictor | 13.4% | 9.1% | 9.5% | 9.7% |
| Top MV predictor | 2.6% | 1.0% | 2.2% | 1.6% |
| Top-right MV predictor | 0.8% | 0.4% | 1.3% | 0.7% |
| Zero MV predictor | 1.6% | 0.6% | 5.3% | 1.8% |

### 2.4.1 Quality and Complexity Analysis of Inter-View Mode Decision

Table 2.4 shows the statistics of complexity reduction and PSNR degradation of the proposed inter-view mode decision. Only the computational complexity of ME in the secondary channel is considered. The computational complexity and coding performance of FSBMA are regarded as references for comparison. It shows that 20.4–73.4% computation for ME is saved with only 0.015–0.035 dB quality loss in PSNR. It indicates that 20.4–73.4% macroblocks are assigned to INTRA, SKIP, or INTER_DE mode, so the following ME is then skipped. Therefore, it can be claimed that the proposed inter-view mode decision effectively executes the mode assignment for each macroblock.

Table 2.6: Quality and Complexity Analysis for Various Refinement Ranges

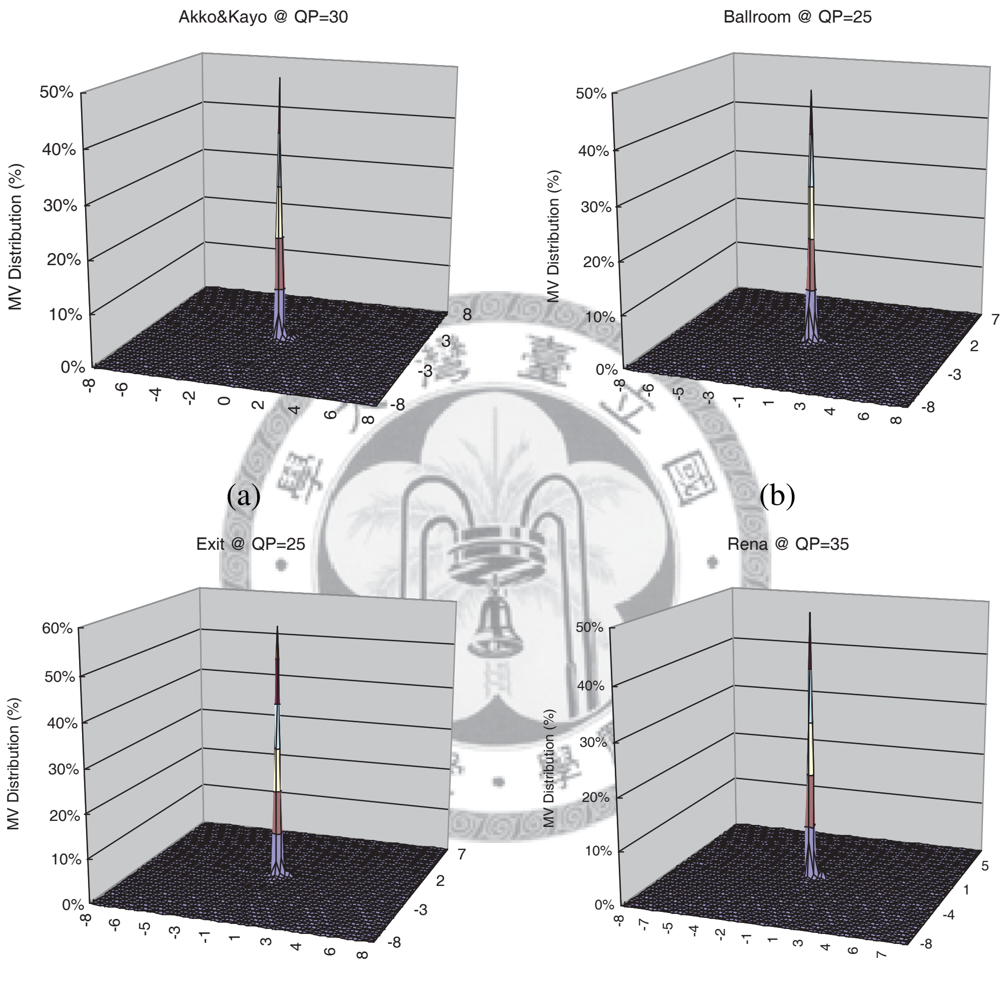
| Refinement Range | PSNR Drop of Sequences (dB) | | | | Complexity Comparison | | |
|---|---|---|---|---|---|---|---|
| | Akko&<br>Kayo | Ballroom | Exit | Rena | Search Candidates<br>for CAPA | Search Candidates<br>for FSBMA | Complexity<br>Ratio |
| [±1, ±1] | –0.051 | –0.088 | –0.055 | –0.119 | 91 | 29575 | 0.3% |
| [±2, ±2] | –0.038 | –0.061 | –0.052 | –0.073 | 203 | 29575 | 0.7% |
| [±4, ±4] | –0.027 | –0.048 | –0.044 | –0.058 | 595 | 29575 | 2.0% |
| [±8, ±8] | –0.024 | –0.031 | –0.030 | –0.057 | 2051 | 29575 | 6.9% |

### 2.4.2 Quality and Complexity Analysis of Content-Aware Motion Estimation

The proposed content-aware ME is a predictor-centered ME algorithm. The optimum initial guess is chosen among five initial guesses, and then the refinement with a small search range is performed around the initial guess. Table 2.5 shows the distribution of the initial guess which is chosen for further refinement. Over 80% macroblocks choose CAPA motion vector predictor for further refinement. It indicates that the proposed algorithm can provide an accurate initial guess. Table 2.6 shows the analysis of quality and complexity with various refinement ranges. The larger the refinement range is, the less PSNR degradation is. It shows that [±4, ±4] refinement keeps PSNR drop within 0.05 dB in average. The motion vector distribution with [±8, ±8] refinement range is shown in Fig. 2.16. Most motion vectors are located within [±2, ±2] range. Therefore, [±2, ±2] and [±4, ±4] are appropriate choices of refinement ranges. In addition, no matter which refinement range is chosen, the required complexity is always much less than that of FSBMA.

### 2.4.3 Rate-Distortion Performance of Content-Aware Prediction Algorithm

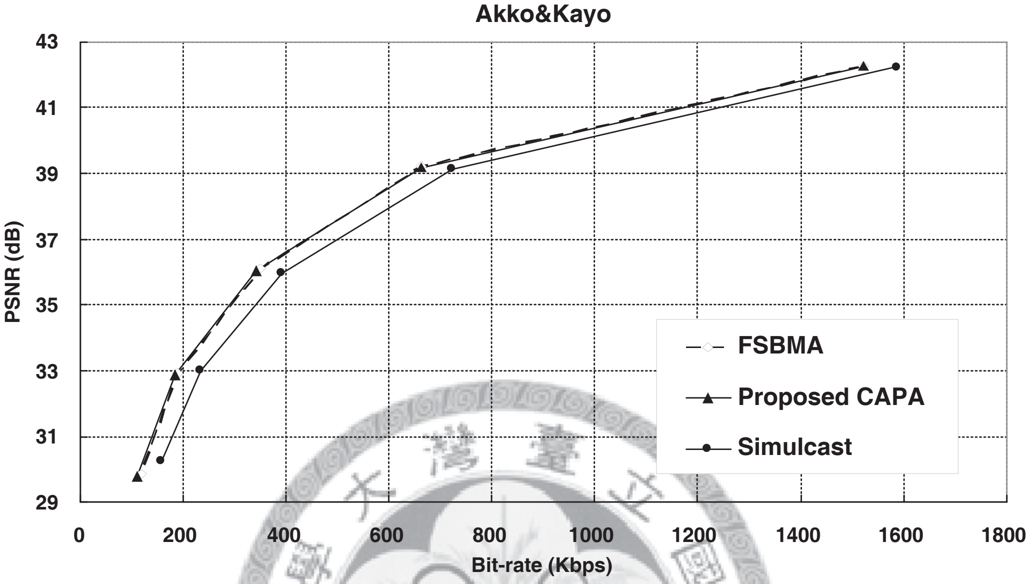
The proposed CAPA consists of inter-view mode decision and content-aware ME. It is compared with multicast coding and simulcast coding. Multicast coding means FSBMA is applied to both ME and DE in the coding structures. On the other hand, in simulcast coding, each view channel is encoded independently without DE. Rate-distortion performance of only secondary channels are compared because the ME parts in the primary channels in all cases are implemented with FSBMA. Moreover, the refinement range is [±4, ±4]. The rate-distortion performance is shown in Fig. 4.13. It shows that there is almost no quality difference between FSBMA and CAPA, and CAPA provides coding gain of 0.09–1.44 dB over simulcast coding. The comparison of quality and complexity among three coding schemes is shown in Table 2.7. Compared with multicast coding, CAPA reduces 98.4–99.1% ME computation in secdonary channels with PSNR drop of only 0.03–0.06 dB. In the previous work



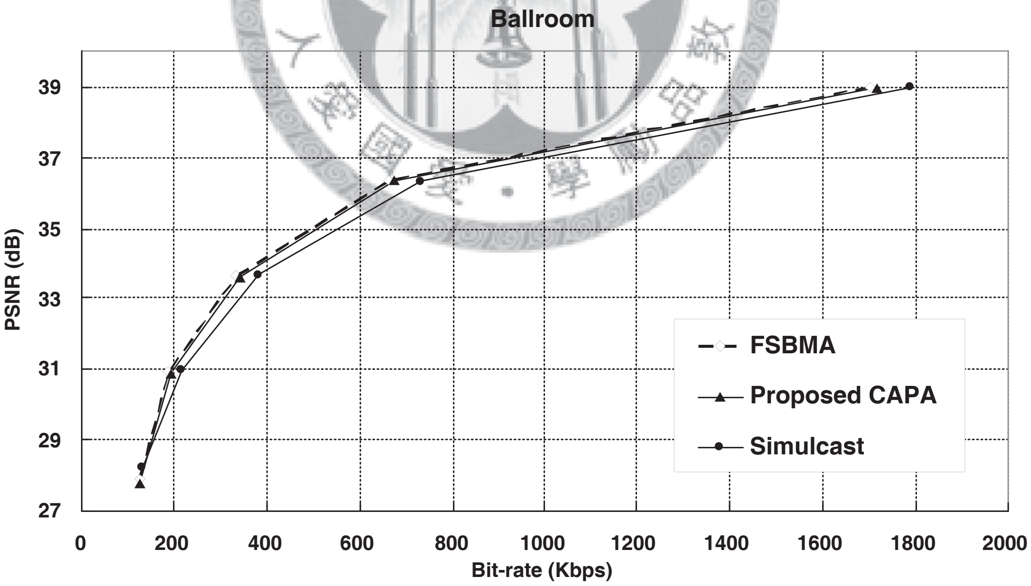
(a) (b)

(c) (d)

Figure 2.16: Motion vector distribution of various test sequences. (a) “Akko&Kayo.” (b) “Ballroom.” (c) “Exit.” (d) “Rena.”

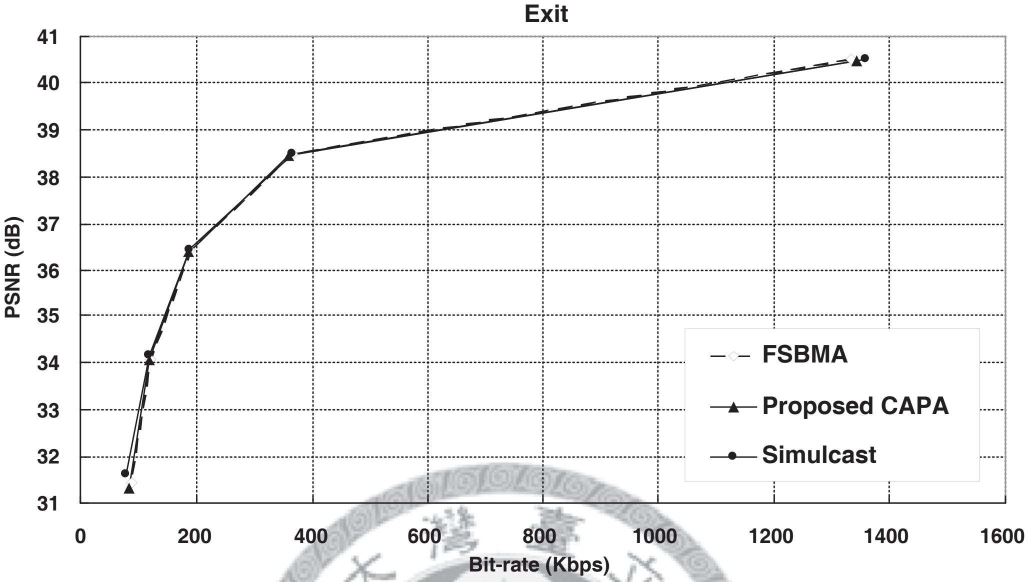


(a) Akko&Kayo

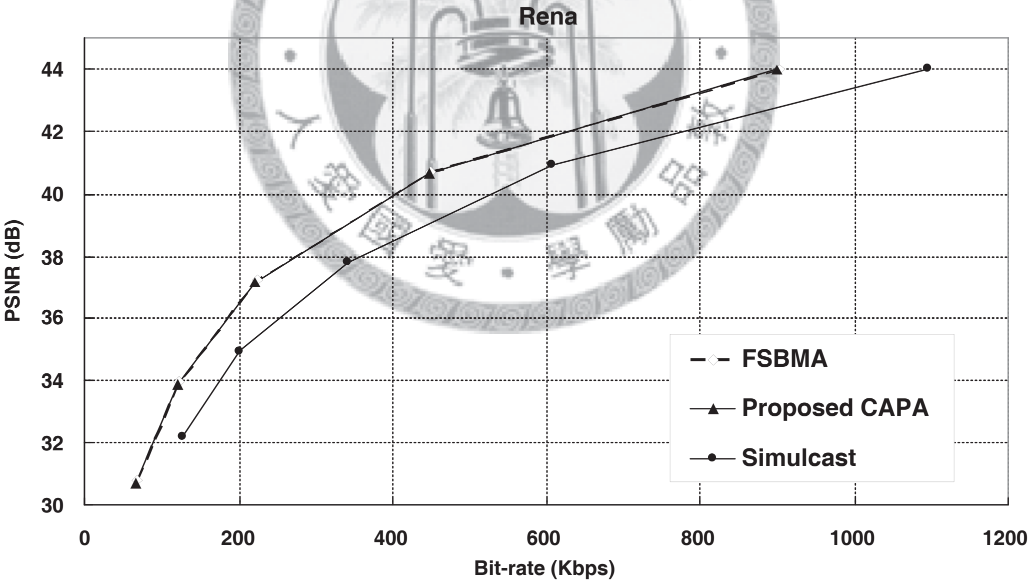


(b) Ballroom

國立臺灣大學 1928



(a) Exit



(d) Rena

Figure 2.17: Comparison of rate-distortion performance among proposed CAPA, FSBMA, and simulcast coding. (a) “Akko&Kayo.” (b) “Ballroom.” (c) “Exit.” (d) “Rena.”

[37][38], 80% computational complexity is reduced with quality loss of 0.1 dB in [37], and about 95% computational complexity can be reduced with quality loss of 0.1–0.2 dB in [38]. Compared with them, the proposed CAPA effectively removes more computational redundancy while maintains the coding performance.

In addition, taking the computation of ME in the primary channel and DE into consideration, 43.6–56.2% complexity can be saved. Note that because the computational complexity of DE is 25% of that of ME in our simulation, the total computational complexity of the proposed algorithm is also much less than that of simulcast coding, and only 51.9–64.1% computational complexity is required. The degree of reduction of the total computational complexity depends on the view numbers. Therefore, it means that the redundant ME computation can be effectively removed by the proposed algorithm. With the proposed CAPA, computational complexity can be greatly saved, and near-FSBMA quality is maintained. That is, the inter-view correlation can be effectively exploited by the proposed algorithm, and then the computational redundancy is removed.

## 2.5 Summary

This chapter presents an MVC encoder structure with an efficient fast prediction algorithm for the prediction part in MVC. Content-aware prediction algorithm (CAPA) with inter-view mode decision is proposed to overcome the design challenge of ultra high computational complexity in MVC. Based on the concept of high inter-view correlation between views and the feature of mode distribution different from that in mono-view video coding, unnecessary ME computation can be early terminated by inter-view mode decision. Moreover, accurate initial guesses are provided by content-aware ME. Only small refinement ranges, such as [±2, ±2] and [±4, ±4], are sufficient for maintaining comparable quality to FSBMA. The proposed algorithm effectively reduces 98.4–99.1% computational complexity for ME in most view channels with negligible quality loss of 0.03–0.06 dB in PSNR. Compared with simulcast coding, the proposed algorithm provides coding gain of 0.09–1.44 dB with only 51.4–64.1% computational complexity. It indicates that the computational re-

Table 2.7: Comparison of Quality and Complexity Reduction Ratio Between the Proposed Algorithm, Multicast Coding with FSBMA, and Simulcast Coding with FSBMA

| Sequences | Compare with Multicast with FSBMA | | | Compare with Simulcast with FSBMA | |
|---|---|---|---|---|---|
| | PSNR Drop | ME Complexity Reduction | Total Complexity Reduction | PSNR Gain | Total Complexity Ratio |
| Akko&Kayo | –0.03 dB | 98.4% | 44.5% | +0.65 dB | 64.1% |
| Ballroom | –0.06 dB | 98.7% | 43.6% | +0.47 dB | 63.7% |
| Exit | –0.04 dB | 99.1% | 56.2% | +0.09 dB | 51.4% |
| Rena | –0.06 dB | 98.7% | 55.8% | +1.44 dB | 51.9% |

dundancy is effectively removed.

There are still some extensions of the proposed algorithm. After adopting the proposed CAPA in MVC, DE will become the most computation-consuming part. Note that the proposed algorithm is orthogonal to other fast ME search algorithms such as three-step search [43], four-step search [44], and diamond search [45], etc. Therefore, appropriate fast prediction algorithms for DE is also worth developing. In addition, the proposed algorithm can also be further adopted in more complex coding structures, such as hierarchical bi-directional prediction, and eight- and sixteen-view structures. The required number of primary channels in a given coding structure is also an important research issue. Less primary channels reduce total computational complexity burden while result in quality degradation. Moreover, the proposed CAPA effectively reduces most computational complexity, while it can also provide an accurate MV predictor to enhance bit-rate savings for MV coding. They are challenging research topics and also belong to our future work.

# Chapter 3

# Fast Luminance and Chrominance Correction Based on Disparity Compensated Linear Regression for Multiview Video Coding

Luminance and chrominance correction (LCC) is important in multiview video coding (MVC) because it provides better rate-distortion performance when encoding video sequences captured by ill-calibrated multiple cameras. In this chpater, a robust and fast LCC algorithm is proposed. This algorithm is based on disparity compensated linear regression which reuses the motion information from the encoder. We adopt the linear weighted prediction model in H.264/AVC as the LCC model. In the experimental results, the proposed LCC algorithm outperforms basic histogram matching method up to 0.4dB with only few computational overhead and zero external memory bandwidth. Therefore, the dataflow of this method is suitable for low bandwidth/low power VLSI design for future multi-view applications.

## 3.1 Introduction

Luminance and chrominance correction (LCC) is an important part in an MVC system due to the camera parameter mismatch which occurs between views, as shown in

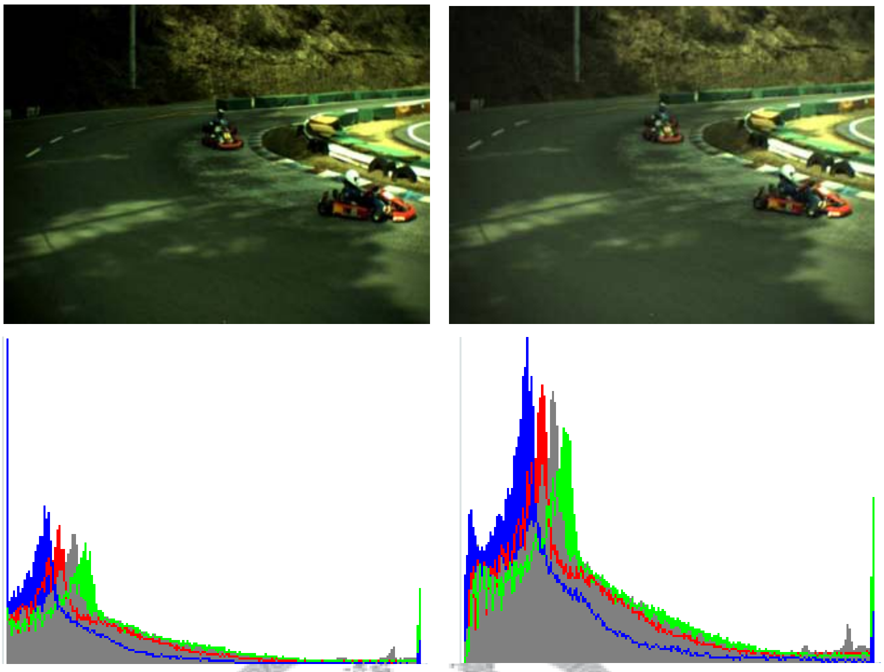


Figure 3.1: Image pair with mismatched luminance and chrominance, the "Race" camera#3 and #4.

Fig. 3.1. To utilize maximal inter-view correlation, the image sources from different cameras should be adjusted to the same tone and illumination before the estimation and compensation steps. For all the various coding tools developed in this decade, weighted prediction (WP) [46] is suitable in this scheme. The original usage of WP is to compensate the luminance and chrominance difference between current frame and the temporal reference frames, such as flashes, fade-in/fade-out, and changed lighting conditions. It can be easily adapted for the inter-view reference frames because the underlying representation method of the frame reference structure is similar. Furthermore, the extent of the luminance/chrominance distortion of inter-view reference frames is averagely higher than that of temporal references, so this coding tool can have better use in this application.

According to many research results based on this coding tool, the global (frame-wise) linear model of WP defined in AVC is sufficient in most cases [47][48]. In this model, there are two parameters for each reference frame that describe the slope (ratio) and intersection (offset) of the linear transform formula. Defining a model is one thing, but finding the best parameters in the model is another. There exists a trade-off between the simplicity of the searching algorithm and the accuracy of the model. The simplest way to get the linear model parameters is implemented in the reference software, JSVM [49]. In this algorithm, the luminance/chrominance DC values (frame average) of the current frame and the reference frames are calculated, and the parameters are derived from the ratio or difference of these DC values. Because this naive method cannot get an accurate model and has few coding gain, some other methods are proposed to achieve higher accuracy and robustness, like linear regression of co-located pixels [50], linear regression of motion compensated pixels [48], histogram matching [51][52], and translational global disparity compensated histogram matching [47]. The methods listed above more or less get better model parameters, but they either neglect those non-global motions [47, 50, 51, 52] or have high computational complexity [48]. Since the motion estimation (ME), or equivalently the disparity estimation (DE), is an essential step in all the video encoders, we can leverage this information without much additional effort to retrieve the relationship of paired pixels between current and reference frames, so the matching assumption of linear regression method can be fulfilled.

The rest of this chapter is organized as follows. The conventional and proposed algorithms are described in section 3.2, and the experimental results are shown in section 5.5. Finally, section summarizes this chapter.

## 3.2 Proposed Algorithm

### 3.2.1 Conventional Encoding Process with Weighted Prediction

To illustrate the proposed algorithm clearly, the conventional coding process with WP is explained first below. As the video coding standard only defines the behavior of the decoder, the description of decoder below is mandatory, but the behavior of the encoder introduced below is only one of many possible implementation methods.

According to the video coding standard, the parameters of the linear correction model for WP consist of an integer offset and a fixed point fractional ratio for each reference frame, and the bit-width of the parameters can be adjusted according to the needed precision and the range of value. The WP parameters are embedded in the header of each frame, and each reference frame has its own WP parameters. When decoding a macroblock (MB), the luminance and chrominance of the referenced MB should be adjusted according to the WP parameters of the frame in which the MB resides. The WP parameters of the two reference frames are considered altogether to make a blended model for the averaged prediction. For the encoder, the parameters of the linear correction model for WP are determined before encoding process. After that, ME/DE and motion compensation (MC)/disparity compensation (DC) are conducted. In the ME/DE stage, the current block is reversely transformed according to the linear correction model before doing the block matching algorithm (BMA) rather than transforming the pixels in the searching area in order to save some computation while keeping reasonable accuracy. In the MC/DC stage, the best-matching block is transformed before compensation, like what should be done in the decoder side.

### 3.2.2 Problem Definition

The disparity compensation with WP can be expressed as

$$I_{comp}(x,y) = mI_{ref}(x',y') + b, \tag{3.1}$$

where $I_{comp}$ is the compensated frame, which should be close to the current frame, $I_{cur}$. The frame-wise parameters in LCC model are $m$ and $b$ in 3.1. $I_{ref}$ is the reference frame, and (x'-x, y'-y) is the disparity vector (DV) of pixel (x, y) in $I_{cur}$. Ideally, the LCC parameters should be determined in order to minimize the difference between $I_{comp}$ and $I_{cur}$. Sum of absolute difference (SAD) is one possible criterion, and the expression is written as

$$(m,b) = \arg\min_{m,d} \sum_{(x,y)} |I_{comp}(x,y) - I_{cur}(x,y)|. \tag{3.2}$$

The exhaustive search is not feasible because the solution domain includes all the possible disparity vectors (DVs) in each MB and the two LCC parameters, so the complexity is the complexity of DE times the possible range of LCC parameters.

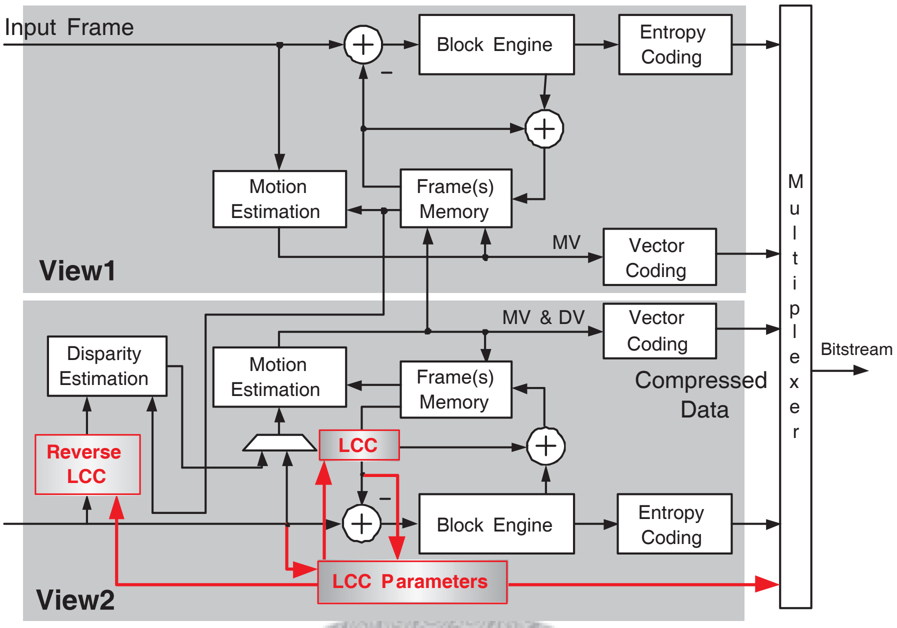


Figure 3.2: Proposed MVC system with color correlation components.

### 3.2.3 Linear Regression With Reused Disparity Information

Fig. 3.2 shows the proposed MVC system with color correlation components. The process is decomposed into two phases. The first phase is DE and color correlation, and the second phase is update of LCC model for the next iteration. The disparity information is used from the DE stage in the normal coding loop. In this way, the only additional computation is the color correlation. We use linear regression over all the disparity-estimated pixel pairs to calculate the color correlation. In order to find the LCC model between $I_{cur}$ and $I_{ref}$, $m$ and $b$ can be calculated after linear regression is done on all the pixel pairs $(I_{cur}(x,y), I_{ref}(x'',y''))$. It should be noted that the DV used in encoding $(x'-x, y'-y)$, assuming it is true motion, could be different from the DV used to estimate the LCC model $(x''-x, y''-y)$. The closer the two different DVs are, the more accurate LCC model could be got. If we refine the model iteratively, the two kinds of DVs can gradually become closer. The algorithm flow is illustrated in Fig. 3.3.

The original usage of the mono-view WP is to compensate the changing illumination. The WP parameters in the mono-view coding changes rapidly and the parameters in different frames are generally not correlated very much, so they should be separately estimated in each frame. However, as the mismatch between a camera

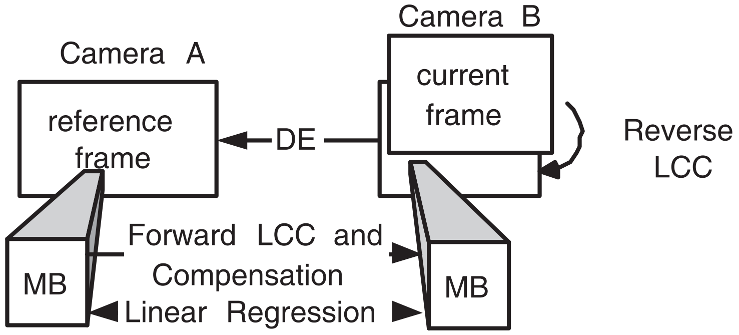


Figure 3.3: Algorithm flow of linear regression with reused disparity information.

pair is relatively stable over time, the LCC model only changes gradually between the same view pairs, so it is wasteful to calculate the model right from the beginning with huge computation resource like the original algorithm used for mono-view WP. Alternatively, the computation in the proposed algorithm is highly reduced and amortized to multiple iterations spread in consecutive frames, so only a low-cost refinement is performed in an iteration. Furthermore, it keeps the ME/DE module intact and reuses the motion/disparity information. Although the R-D efficiency could be worse than doing dedicated ME/DE as [48], the computational cost is largely reduced and the correction model still converges to the optimal value after reasonable iterations. The more biased the tone between different views, the less accurate the MVs can be estimated in the first iteration, thus the color correlation calculated from the pixel pairs could be affected either. As a result, more iterations are needed if the color distortion is severe. The coding flow is described in the pseudo code in Fig. 3.4. In the coding flow, the LCC model is fixed in the ME/DE and MC/DC stage. After the motion information is generated, the LCC model is updated for the next iteration.

We express the $i_{th}$ frame in $j_{th}$ view as $V_jF_i$, and the LCC model estimated at time $i$ to predict view $j$ from view $k$ is *LCC*($V_jF_i$, $V_kF_i$). The proposed algorithm uses *LCC*($V_jF_i$, $V_kF_i$) as the initial value in the refinement process for *LCC*($V_jF_{i+1}$, $V_kF_{i+1}$), since the LCC model between the same view pair in consecutive time slots should be similar. When encoding frame $V_jF_{i+1}$, each blocks are reversely transformed by the initial model *LCC*($V_jF_i$, $V_kF_i$). In the estimation stage, the reversely transformed current block are compared with many candidate blocks in the reference

```
(m, b) □ model from the previous iteration
for each current block {
  reversely transform the current block $B^{cur}$ by (m, b)
  perform DE and get matched block $B^{ref}$
  accumulate linear regression registers with ($B^{cur}$, $B^{ref}$) pair
  compensate $B^{comp}$ with forwardly transformed block $B^{ref}$
}
(m, b) □ new linear model from linear regression
```

Figure 3.4: Pseudo-code of the proposed algorithm.

frame $V_kF_{i+1}$, and the best matching block pair between $V_kF_{i+1}$ and $V_kF_{i+1}$ are found. The best matching block in $V_kF_{i+1}$ is forwardly transformed to be the compensated block. The model *LCC*($V_jF_i$, $V_kF_i$) is adjusted by the statistical data gathered from the motion/disparity information, and then saved as *LCC*($V_jF_i$, $V_kF_i$) to be used at the next time prediction view $j$ from view $k$. There is a single-frame latency between the estimation and the application of LCC models, so the performance can be degraded when the LCC model changes rapidly.

To sum up, the proposed algorithm is an iterative refinement process of the model parameters. When encoding a frame with the previously calculated LCC model, the pixel pairs in matched blocks are fed in the linear regression module. When all the blocks are finished, the LCC model can be refined according to the result of the linear regression, and the new model is used as the initial model when encoding frames from the same view next time.

### 3.2.4 Computational Cost and Bandwidth Overhead

Compared with the conventional encoder, the proposed algorithm only additionally requires the linear regression computation, which consists of three multiplications and five additions per pixel per color channel per reference frame, and five multiplications, three subtractions, and two divisions per color channel per reference frame. This computational cost is negligible for a video encoder compared with ME/DE.

Table 3.1: Comparison of Computational Complexity With Three Algorithms

| Algorithm | Computational<br>Complexity | R-D<br>Performance |
|---|---|---|
| Histogram-based | 15.5MIPS | lowest |
| Proposed algorithm | 124.4MIPS | higher |
| Linear regression with dedicated DE | 2073.6MIPS† | highest |

† Assuming 100 search candidates for DE.

Table 3.2:

| Algorithm | Computational<br>Complexity | R-D<br>Performance |
|---|---|---|
| Histogram-based | 15.5MIPS | lowest |
| Proposed algorithm | 124.4MIPS | higher |
| Linear regression with dedicated DE | 2073.6MIPS | highest |

The linear regression is done at the same time when the best-matched block is loaded in the compensation stage, so it causes no bandwidth overhead to the external memory, or equivalently higher cache hit-rate in software model. In the perspective of VLSI design, low external bandwidth leads to low power design, and the hardware area cost is only several adders and multipliers, and around 150 bits of registers. When encoding a video sequence in D1 size, 30 fps, the three kinds of algorithms can be compared as Table 3.2.

## 3.3 Simulation Result

### 3.3.1 Environment Setup

JSVM3.5 is modified to support our LCC model refinement algorithm. Several multiview video sequences are encoded to compare the R-D efficiency of multiple LCC model searching algorithms, and some R-D curves are shown in section 3.3. The prediction structure shown in Fig. 3.5 is used in order to focus on the efficiency of

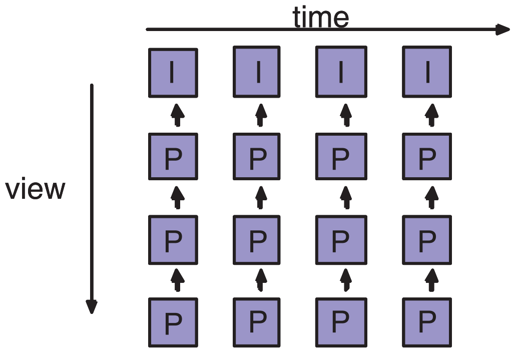


Figure 3.5: IPPP coding structures in the experiment.

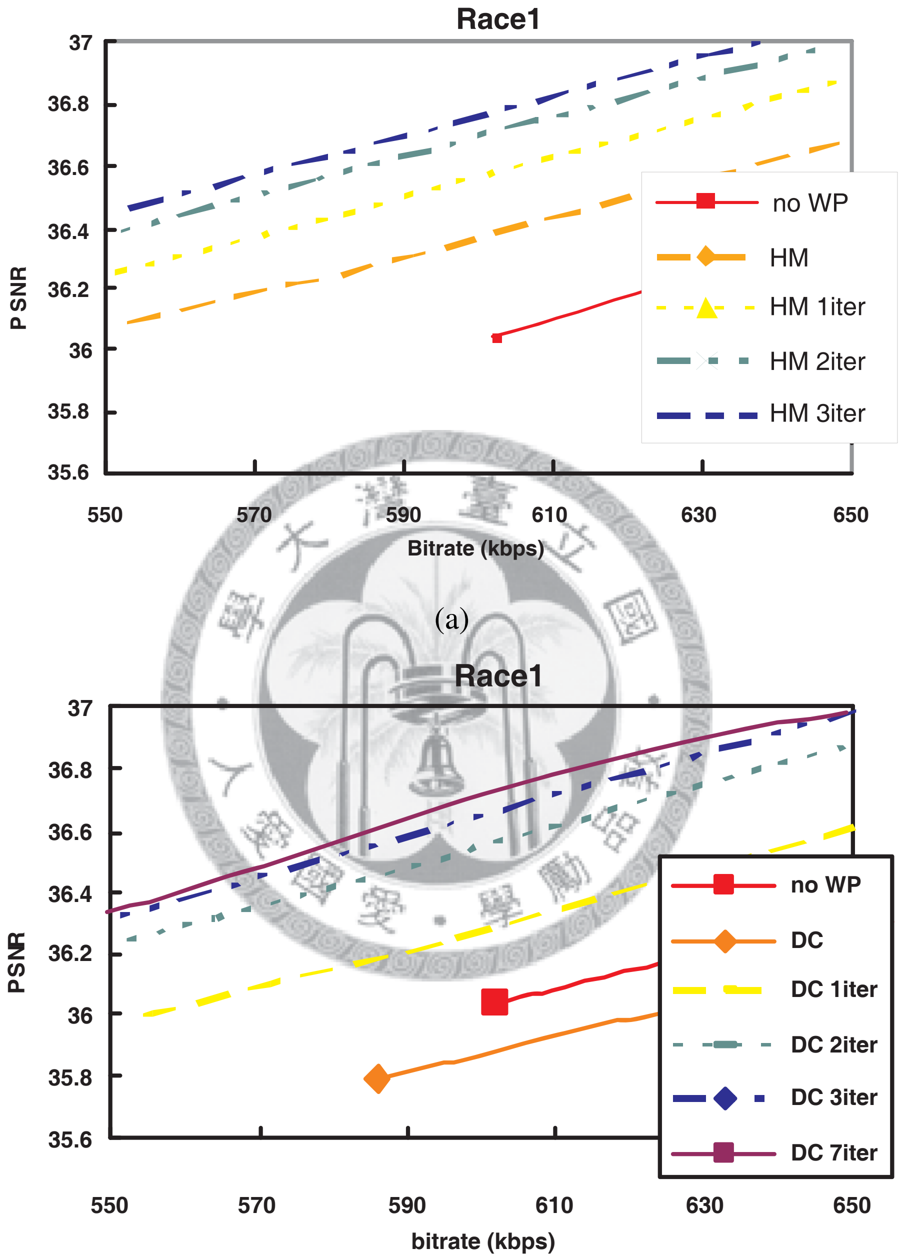
inter-view prediction.

### 3.3.2 Initial Correction Models

In this experiment, two initial LCC model values are used. One of them is the default algorithm of the WP implemented in JSVM3.5, which uses the ratio of the DC values of the current frame and the reference frame as the slope of the linear model and set the intersection to zero. Another is histogram matching, which minimizes the area of absolute difference of the accumulated histograms of current frame and reference frame by tuning the LCC model.

### 3.3.3 Simulation Result

The experiment result is shown below. The label “no WP” means the sequence is encoded without WP, ”DC” means the initial value of the correction model is calculated by the ratio of DC values, and “HM” means the initial value is from histogram matching. The label ended with “n iter” means n refinement iterations of proposed algorithm are done after the initial model. Fig. 3.6 shows that the PSNR increases with number of iterations. If the initial value is as good as HM model, then the correction model would converge after only three iterations. Even if the initial value is worse than ”no WP” like ”DC”, the proposed algorithm can still improve it to nearly the same PSNR as that using ”HM” as initial value, and the difference is smaller than 0.1dB. So it can be concluded that the initial model is not absolutely needed and $m = 1, b = 0$ can be used. A worse initial model only increases the number of itera-



(a)

(b)

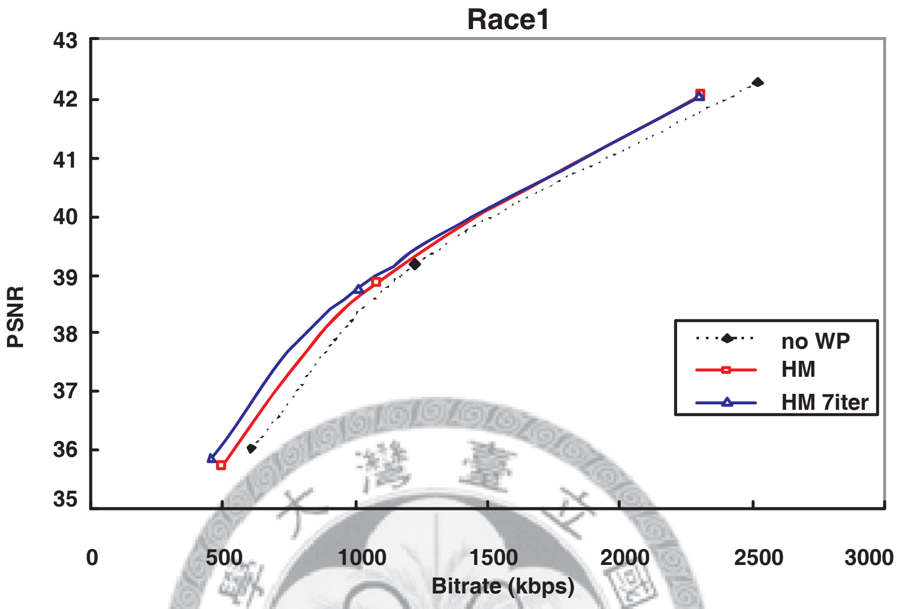
Figure 3.6: (a) Magnified RD curve of Race1, HM as initial value. (b) Magnified RD curve of Race1, DC as initial value.

tions required to converge to steady state, but it doesn't severely affect the quality of that steady state.

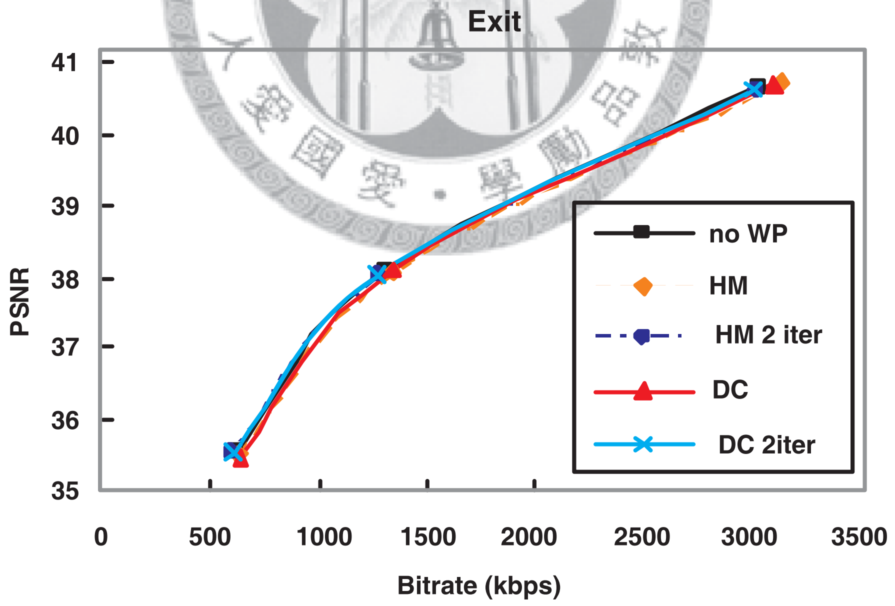
Compared with the HM algorithm, the proposed algorithm has coding gain from 0.1 to 0.4dB in different bitrate ranges in Fig. 3.7. The inter-view mismatches in sequence Exit are lower, so WP is not very helpful. However, from Fig. 3.7 (b), the coding gain of HM is negative, while that of proposed algorithm is still positive, and this shows the robustness of our method. In Fig. 3.7 (c) and Fig. 3.7 (d), we can see that the color distortion in sequence Ballroom and Breakdancers is even lower. No matter how many iterations are applied after the ”DC” initial condition, the R-D curve is still lower than that of ”HM”. This seems contradictory to the previous claim, but the difference is less than 0.05dB.

## 3.4 Summary

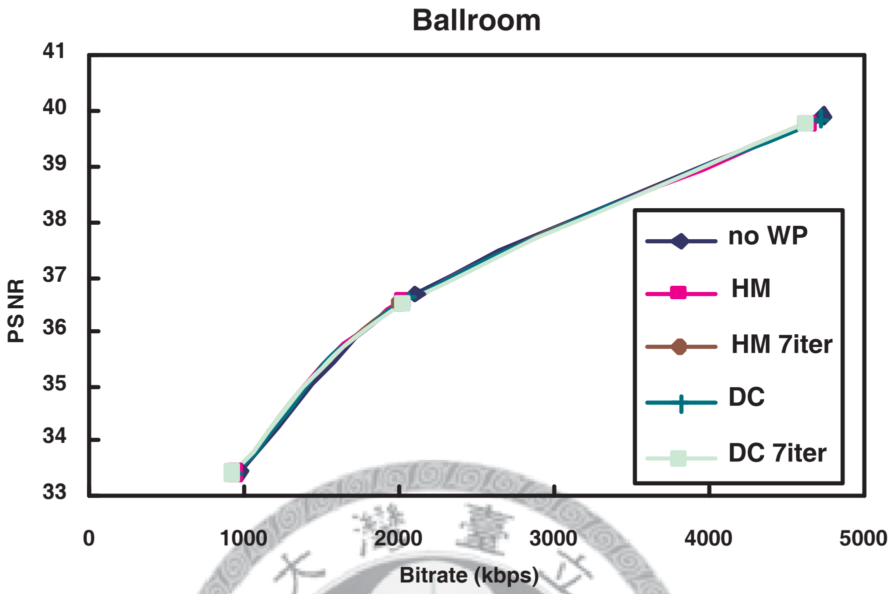
Luminance and chrominance correction for MVC can boost the R-D curve considerably when the images from the camera array are not perfectly adjusted. The distortion of luminance and chrominance is modeled as a linear transform, and the global WP model in H.264/AVC is used in our coding system. The proposed algorithm amortizes the computation of multiple ME/DE-color correlation refinement iterations to multiple frames in the same view, and reuses the disparity information from the encoder to calculate accurate LCC models with negligible computation overhead. The dataflow is also suitable for low power VLSI implementations because it requires zero additional external memory bandwidth. The coding gain of the proposed LCC model refinement algorithm compared with HM model is up to 0.4 dB. Due to the higher robustness of the proposed method, the coding gain for those sequences with little mismatch is generally positive, while that of HM is sometimes negative. Besides coding efficiency, the LCC model embedded in the encoded bitstream can be further utilized by the decoder to do the post-processing if homogeneous tone and illumination is preferred for the quality of subjective view.


Race1
PSNR
43
42
41
40
39
38
37
36
35
0
500
1000
1500
2000
2500
3000
Bitrate (kbps)
no WP
HM
HM 7iter

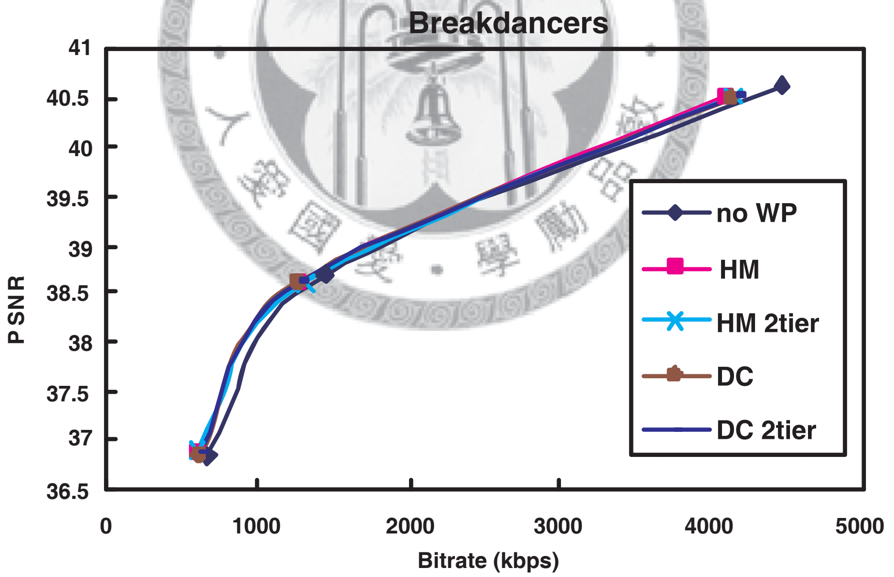
(a)


Exit
PSNR
41
40
39
38
37
36
35
0
500
1000
1500
2000
2500
3000
3500
Bitrate (kbps)
no WP
HM
HM 2 iter
DC
DC 2iter

(b)



(c)



(d)

Figure 3.7: Comparison of rate-distortion performance among proposed algorithm, HM, and w/o any algorithms. (a) "Race." (b) "Exit." (c) "Ballroom." (d) "Breakdancer."

國立臺灣大學 · 愛國 · 愛學 · 勵學 · 敦品 · 人

# Part II

# Algorithm and Architecture Co-Design of Prediction Core for 3D/Quad High Definition Video Coding

國立臺灣大學
敦品勵學・愛國愛人

# Chapter 4

# Joint Prediction Algorithm and Architecture for Stereo Video Hybrid Coding Systems

Among the 3D video technologies, stereo video systems are considered to be realized first in the near future. Stereo video systems require double bandwidth and more than twice computational complexity relative to mono-video systems. Thus an efficient coding scheme is necessary for transmitting stereo video. In this chapter, a new structure of prediction core in stereo video coding systems is proposed from algorithm level to hardware architecture level. The joint prediction algorithm (JPA), which combines three prediction schemes, is proposed for high coding efficiency and low computational complexity. It makes the system outperform MPEG-4 temporal scalability and simple profile by 2–3 dB in rate-distortion performance. Besides, JPA also utilizes the characteristics of stereo video and successfully reduces about 80% computational complexity. Then a new hardware architecture of prediction core based on JPA and modified hierarchical search block matching algorithm (HSBMA) is proposed. With special data flow, no bubble cycles exist during the block matching process. The proposed architecture also adopts near-overlapped candidates reuse scheme (NOCRS) to save the heavy burden of data access. Besides, both on-chip memory requirement and off-chip memory bandwidth can be reduced by the proposed new scheduling. Compared with the hardware requirement for the implementation of full search block matching algorithm (FSBMA), only 11.5% on-chip SRAM and 3.3% processing elements are needed with a tiny PSNR drop. It is area efficient while maintaining high stereo video quality and processing capability.

## 4.1 Introduction

Stereo video can provide the users with a sense of depth perception by showing two frames to each eye simultaneously. It can give users a vivid information about the scene structure. With the technology of 3D-TV getting more and more mature [21], stereo and multi-view video coding draw more and more attention. In recent years, MPEG 3D auido/video (3DAV) group has worked toward the standardization for multi-view video coding [26], which also advances the stereoscopic video applications. Although stereo video is attractive, the amount of video data and the computational complexity are doubled. A good coding system is first required to solve the problem of huge data with limited bandwidth. In a mono-video coding system, motion estimation (ME) requires the most computational complexity [53]. By comparison, computational loading is even heavier in stereo video coding systems due to additional ME and disparity estimation (DE). Therefore, an efficient prediction scheme is required to overcome these problems. Moreover, it is preferred that the proposed video encoding system can be easily integrated by the existing video standards.

Some stereo video coding systems have been proposed. Stereo video coding can be supported by temporal scalability tools of existing standards, such as the MPEG-2 multi-view profile (MVP) [17], where a view is encoded as the base layer, and the other one is encoded as the enhancement layer. This approach does not have good coding efficiency [18]. The I3-D [32] is a famous approach, in which the texture information is collected in a synthetic view, and the depth information is recorded in a disparity map. It has good coding efficiency and compatibility with MPEG-4 standard. However, additional operations for extracting disparity maps and synthesizing stereo views are required in the encoder and the decoder, respectively, which are not the building blocks of conventional video coding systems. A mesh-based and block-based hybrid approach is proposed by Wang *et al*. [33]. However, it needs additional preprocessing for segmentation to prevent matching failure around the object boundary [54]. In addition, the computational complexity is very high.

DE is the core of stereo video coding systems. The DE algorithms of the previous systems can be roughly classified into three categories: pixel-based, mesh-based, and block-based. Pixel-based algorithms, such as dynamic programming [30][31], and mesh-based algorithms [33], can generate more precise disparity or depth maps than block-based algorithms do. Because of the feature of non-crossing order of vectors, the mesh-based algorithms have good view-synthesis ability. The main disadvantage of pixel-based and mesh-based algorithms is that they cannot be compatible with the existing video coding standards [55]. An ultra high computational complexity is usually required for this approach. Besides, a segmentation step is usually needed for more accurate estimation in the mesh-based algorithms. On the other hand, the main advantage of block-based algorithms is that they have much better compatibility with the existing standards.

In this chapter, a new stereo video coding system with joint prediction algorithm (JPA) and its architecture are proposed. For better compatibility, the system is based on the hybrid coding scheme. Block-based algorithms are adopted for ME and DE, which are combined as the prediction core with JPA. To improve the coding efficiency and reduce the computational complexity, JPA is composed of three coding tools, joint block compensation, MV-DV (motion vector-disparity vector) prediction, and mode pre-decision. To meet the real-time constraint, the hardware architecture is designed based on the modified hierarchial search block matching algorithm (HS-BMA) and the joint block compensation scheme. A new scheduling and bandwidth-reduction scheme is proposed for improving the hardware utilization.

The rest of the chapter is organized as follows. Section 4.2 describes the proposed stereo video hybrid coding system with proposed JPA. Next, the analysis and algorithms of BMA are presented in 4.3. Section 4.4 describes the proposed prediction core architecture. Finally, Section 7.5 summarize this chapter.

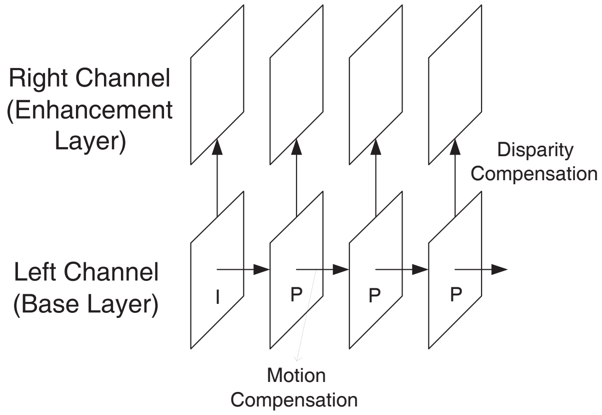


Figure 4.1: Base-layer/enhancement-layer scheme of the proposed system. The base layer is encoded with MPEG-4 Simple Profile (SP) encoder.

## 4.2 Proposed Joint Prediction Algorithm

For the purpose of compatibility, the coding system adopts a base-layer/enhancement-layer scheme, as shown in Fig. 4.1. The left view is set as the base layer, and the right view is set as the enhancement layer. The base layer is encoded with MPEG-4 Simple Profile (SP) encoder [56]. The block diagram of the proposed stereo video encoder is shown in Fig. 4.2. The main differences between the left channel and the right channel are disparity estimation, joint block generation, and other functional blocks such as mode pre-decision and MV-DV prediction, which will be introduced later. Note that reference frames from left and right channels are both reconstructed. After encoding, the left compressed data, M and L, and the right compressed data of a small amount, N and R, are transmitted.

In the stereo video coding system, the prediction is the most important part. It is not only computationally intensive but also critical for the coding efficiency. Motion/disparity estimation/compensation are the key operations in the prediction core. Figure 4.3 illustrates the prediction directions and the search windows (SWs) of two reference frames for ME and DE, respectively. For the current block in the right channel at $t = 1$, in addition to ME with $SW_{r,ME}$, there exists another way to find good prediction, that is, DE with $SW_{l,DE}$. ME can remove the temporal redundancy. On the other hand, DE can remove the inter-view redundancy [57]. Therefore, the frames in the right channel have more than one choice to find their best matching

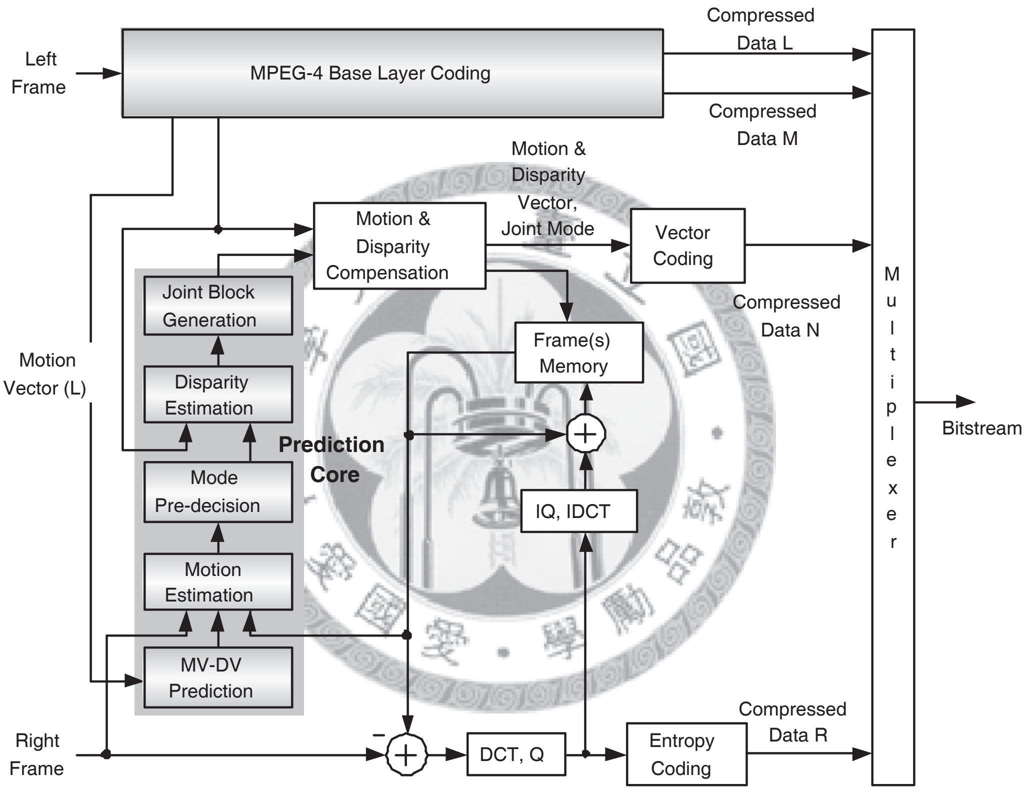


Figure 4.2: Block diagram of the proposed stereo video encoder. The system is based on the hybrid coding scheme.

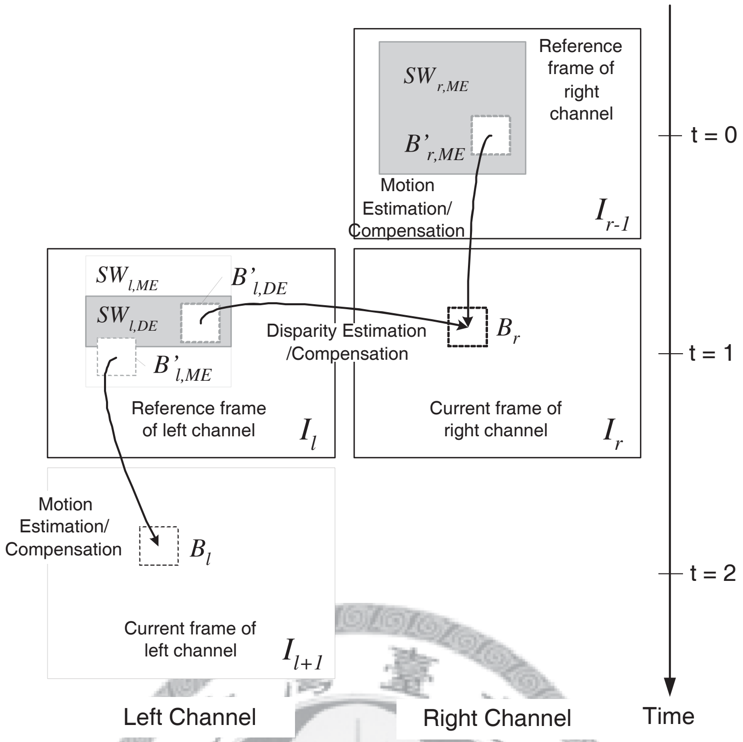

Reference frame of right channel
$SW_{r,ME}$
$B'_{r,ME}$
Motion Estimation/ Compensation
$I_{r-1}$
t = 0
$SW_{l,ME}$
$B'_{l,DE}$
$SW_{l,DE}$
$B'_{l,ME}$
Disparity Estimation /Compensation
$B_r$
t = 1
Reference frame of left channel
$I_l$
Current frame of right channel
$I_r$
Motion Estimation/ Compensation
$B_l$
t = 2
Current frame of left channel
$I_{l+1}$
Left Channel
Right Channel
Time

Figure 4.3: Illustration of the prediction directions and the SWs of two reference frames for ME and DE, respectively. The gray regions are the SWs for $B_r$.

blocks.

In order to improve the coding efficiency and reduce the computational complexity in the right channel, JPA is proposed and based on three prediction schemes. Firstly, a block is compensated not only by the block of left or right reference frames but also by the combination of them according to different types of content of it. Secondly, the properties of stereo video are considered for the accurate motion vector prediction to reduce the computational complexity of ME. Thirdly, the computational complexity of DE is reduced by the proposed mode pre-decision scheme. Based on these three schemes, in this section, the details of JPA are shown in the subsections.

## 4.2.1 Joint Block Compensation

### Joint Block

In ME and DE steps of the right channel, the current block has two reference frames, as shown in Fig. 4.3. The gray region is the SW of a reference frame. Note that the

SW of the left reference frame for DE is not a square because cameras are assumed to be parallel-structured, so the candidate blocks are only on the belt of the region [42]. There are mainly two types of compensated blocks in the right channel. They are the motion compensated (MC) block and the disparity compensated (DC) blocks which are illustrated as $B'_{r,ME}$ and $B'_{l,DE}$ in Fig. 4.3, respectively. MC block often occurs in the background because of its zero or slow motion. Since DE is usually not able to predict well in the case of occlusions between left and right frames, these blocks will also be compensated by this type of blocks. On the other hand, DC block often occurs in the moving objects because of their deformation during motion. In this case, DC blocks usually have better prediction capability.

However, when a frame is divided into several blocks, there is a high probability that a block may contain more than one type of video objects. For example, a block may contain both moving objects and background in it. In this case, neither the MC block nor the DC block can predict well. As a result, a new type of compensated blocks, the joint block, is proposed. Figure 4.4 shows the illustration of the proposed joint block generation and compensation. After ME and DE, the best matching blocks in the two SWs are derived. Then the joint block generation step starts. They are the linear combination of the MC and the DC blocks. By the specified weighting parameters, several different joint block candidates are generated.

According to the criterion of sum of absolute difference (SAD), the best type of compensated block is selected. For each macroblock (MB) of the current frame, the distortion of the three types of blocks are computed by

$$D_{motion} = \min\{\sum_{t\in B_r, t'\in B'_{r,ME}} |I_r(t) - I_{r-1}(t')|\,|_{B'_{r,ME}\in SW_{r,ME}(B_r)}\}, \tag{4.1}$$

$$D_{disparity} = \min\{\sum_{t\in B_r, t'\in B'_{l,DE}} |I_r(t) - I_l(t')|\,|_{B'_{l,DE}\in SW_{l,DE}(B_r)}\}, \tag{4.2}$$

$$\begin{aligned} D_{j_n} = \quad & \min\{\textstyle\sum_{t\in B_r, t'\in B'_{l,DE}, t''\in B'_{r,ME}} |I_r(t) - [W_n\cdot I_l(t') + W'_n\cdot I_{r-1}(t'')]| \\ & |_{W_n+W'_n=1}\}, \end{aligned} \tag{4.3}$$

where $D_{motion}$ and $D_{disparity}$ are the minimum SADs of MC and DC blocks, respectively. $B_r$ is the current block in the right channel. $B'_{r,ME}$ is the reference block

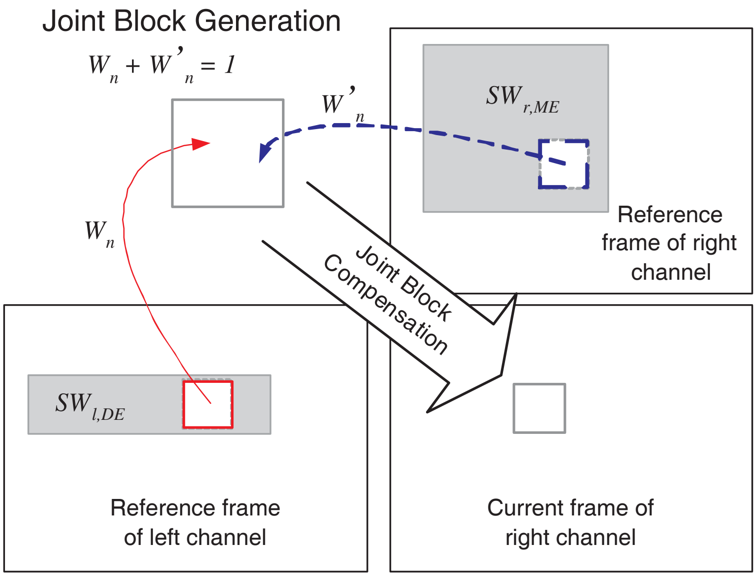


Figure 4.4: Proposed joint block generation and compensation. The joint block is the weighted sum of the MC and the DC blocks.

in the right channel. $B'_{l,DE}$ is the reference block in the left channel. $SW_{r,ME}(B_r)$ and $SW_{l,DE}(B_r)$ are the SWs in the right and left reference frames of the block $B_r$, respectively. The proposed joint block is then generated as the weighted sum of the two blocks. $W_n$ and $W'_n$ are complementary weighting functions that describe the weighting parameters. In (4.3), $D_{j_n}$ is derived. Finally, the mode decision is described as

$$Mode = \arg\min_{mode}\{D_{motion}, D_{disparity}, D_{j_1}, ..., D_{j_n}\}. \tag{4.4}$$

In our stereo video encoder, the modes are compressed by the arithmetic coding process. Note that the proposed algorithm is applied on the luminance domain. When performing the joint block compensation, the chrominance data are compensated by the same set of vectors, as in the existing hybrid coding standards [56] [58].

### Selection of Joint Block Weighting Parameters and Patterns

There are infinite weighting parameters or patterns which can be used for joint block generation. In the previous works, the MB is only compensated by the MC and the DC blocks with fixed weighting parameters [59][60]. In our experiment, seventeen modes are considered. As shown in Table 4.1, nine modes of joint blocks are gen-

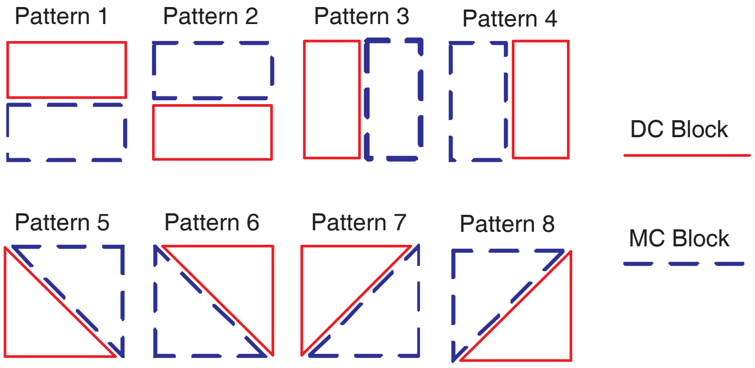


Figure 4.5: Selection of joint block patterns. 8 patterns are chosen for experimental analysis.

erated by the weighted sum of MC and DC blocks. Five stereo video sequences are tested and averaged. For example, The value of $W_n$, 0.625, means the pixels in the joint block are the sum of the pixels of the DC block multiplied by 0.625 and the pixels of the MC block multiplied by 0.375. Note that the zero $W_n$ makes the joint block mode as the same as using ME. $W_n = 1$ means the joint block use DE only for prediction. On the other hand, there are also eight kinds of joint blocks generated by the combination of complementary shapes, as shown in Fig. 4.5. In Table 4.1, it shows that after mode decision, 92% of blocks choose the joint block modes formed by the weighting parameters. Only 8% blocks choose the combination of complementary shapes to form the joint blocks. The reason is that the edges of patterns used are very sharp and straight, while the shape of an object is not so in general cases. Therefore, the distortion of these kinds of joint blocks is often large.

Figure 4.6 shows the rate-distortion performance of various number of joint block modes. Five, nine, and seventeen modes are taken into consideration. Curve I contains 9 weighting modes and 8 pattern combination modes listed in Table 4.1. Curve II and III contain only weighting modes without pattern combination modes. The performance of Curve I and II are similar. The reason is that although joint block compensation with more modes has better prediction ability and significantly reduce bitrate for encoding residue, it has the penalty for encoding mode information. As a result, we decide to choose the weighting parameters to generate all the joint blocks in the stereo video coding system.

Table 4.1: Statistics of the Probability of Various Joint Block Patterns

| $W_n$ | Percentage | Pattern combination | Percentage |
|---|---|---|---|
| 0.000 | 23.1% | Pattern1 | 2.1% |
| 0.125 | 2.8% | Pattern2 | 1.5% |
| 0.250 | 2.4% | Pattern3 | 0.1% |
| 0.375 | 14.5% | Pattern4 | 0.3% |
| 0.500 | 18.8% | Pattern5 | 1.6% |
| 0.625 | 11.5% | Pattern6 | 0.2% |
| 0.750 | 8.5% | Pattern7 | 0.7% |
| 0.875 | 5.6% | Pattern8 | 1.5% |
| 1.000 | 4.8% | | |

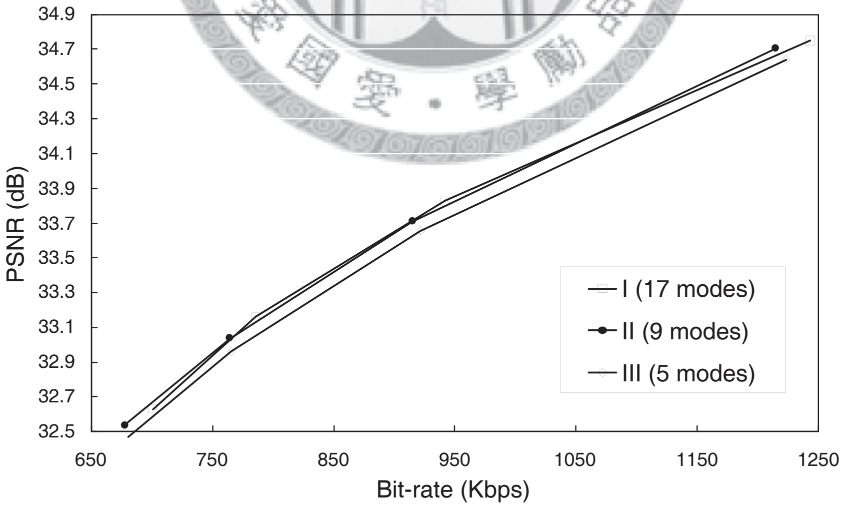


Figure 4.6: Rate-distortion performance of various number of joint block modes. Curve I contains both weighting modes and pattern combination, whereas curve II and III contains only weighting modes.

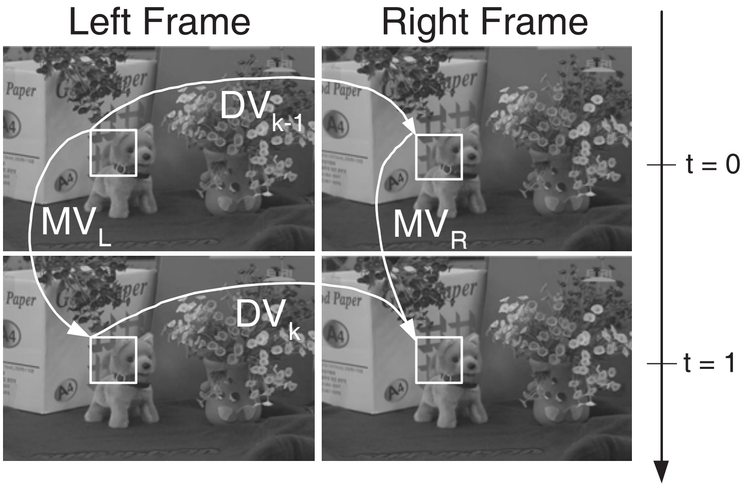


Figure 4.7: The relation between DVs and MVs. If parallel-setup cameras are unchanged, DVs of an object in different time slots are almost the same.

### 4.2.2 MV-DV Prediction

In general stereo video coding systems, ME and DE are the key operations. However, compared with mono-video systems, additional ME and DE of the right channel greatly increase the computational burden. Therefore, the MV-DV prediction scheme is proposed.

#### Correlation between DVs and MVs

The correlation is shown in Fig. 4.7. It can be described as [34]

$$DV_{k-1} + MV_R = MV_L + DV_k. \tag{4.5}$$

If parallel-setup cameras are unchanged, DVs of an object in different time slots are almost the same. Therefore,

$$DV_{k-1} \approx DV_k \Rightarrow MV_R \approx MV_L. \tag{4.6}$$

According to the correlation, $MV_L$ is used as the predictor of $MV_R$. Because of the parallel-setup camera structure, there is an global horizontal displacement between left and right channels, which is called the global disparity. In order to find the predictors, the global disparity should be derived first because of the relation between

MVs and DVs introduced above. Here, we use a simple way to find the global disparity rather than complex global motion estimation (GME) scheme [61]. That is, DE is performed with the statistic process in the first P-frame in the right channel. Since the background usually occupies the largest area, the disparity that occurs most frequently is set as the initial global disparity. The global disparity is dynamically updated due to unexpected conditions such as scene change and moving background. The details will be introduced in the next subsection. The background detection is determined by

$$F_{diff}(N) = \sum_{t \in B_{N,r}, t' \in B_{N,r-1}} |I_r(t) - I_{r-1}(t')|, \tag{4.7}$$

$$Background(N) = \begin{cases} true, & \text{if } MV_N(x,y) = (0,0) \text{ OR} \\ & \quad F_{diff}(N) < Threshold \\ false, & \text{otherwise.} \end{cases}, \tag{4.8}$$

where $B_{N,r}$ and $B_{N,r-1}$ are the $N-th$ blocks in $I_r$ and $I_{r-1}$, respectively. $Background(N)$ is the state of the $N-th$ block in the right frame. $MV_N(x,y)$ is the MV of the $N-th$ block. The $Threshold$ of $F_{diff}(N)$ shown here is empirically chosen for the additional criterion. According to the background information, the disparity vectors of these background blocks are counted in the statistical analysis. Then, the global disparity vector, GD, is derived as

$$GD = \arg\max_{DV}\{\ Num(DV)\ \}, \tag{4.9}$$

where $Num(DV)$ is the histogram of DV. For the first P-frame in the left channel, background detection scheme is used to find $GD$. Before ME in the right channel, the corresponding block in the left frame of the current block in the right frame can be found by use of $GD$. Then the MV of the corresponding block is used as the predictor of the current block. $MV_R$ is derived within a small SW to reduce computation. However, the DVs of background are usually smaller than those of foreground. If the SAD is larger than a empirically chosen threshold, the block is viewed as a foreground block. Its search range extends adaptively to find a better MV. Next, a more precise GD is fed back to the system. To avoid error propagation, $GD$ can

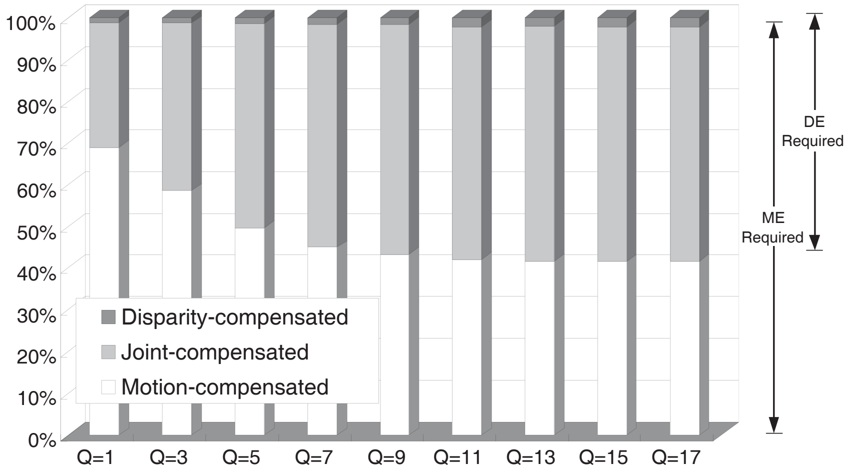


Figure 4.8: Percentage of block prediction types of sequence “Race2.”

be updated after every M frames, where M is a flexible parameter. In the proposed stereo video coding system, the MVs in the left channel and the DVs between two channels are coded separately by checking the vector differences in the self-defined look-up tables. The MVs in the right channel are then predicted by these two vectors in the above-mentioned way.

### 4.2.3 Mode Pre-decision

In addition to the reduction of ME complexity, a new method is proposed for computational complexity reduction of DE. In our experiments shown in Fig. 4.8, 40%–70% blocks in the right frame are motion-compensated, 25%–60% blocks are joint-compensated, whereas only about 5% blocks are disparity-compensated. From the above analysis, over 95% blocks must perform ME, while only 30%–60% blocks must perform DE. Thus an unnecessary DE could be skipped to reduce computational complexity. From our analysis, the MV-predicted blocks often have zero motion, such as blocks in the background, or have slow motion caused by moving cameras. An example is shown in Fig. 4.9. The highlighted blocks are DV-predicted. It shows that moving objects, such as soccer players or the ball, are usually DV-predicted while the other blocks composed of the background are usually MV-predicted. By use of these properties, mode pre-decision scheme is applied after the minimum SAD

國立臺灣大學 愛國 學 勵品 教

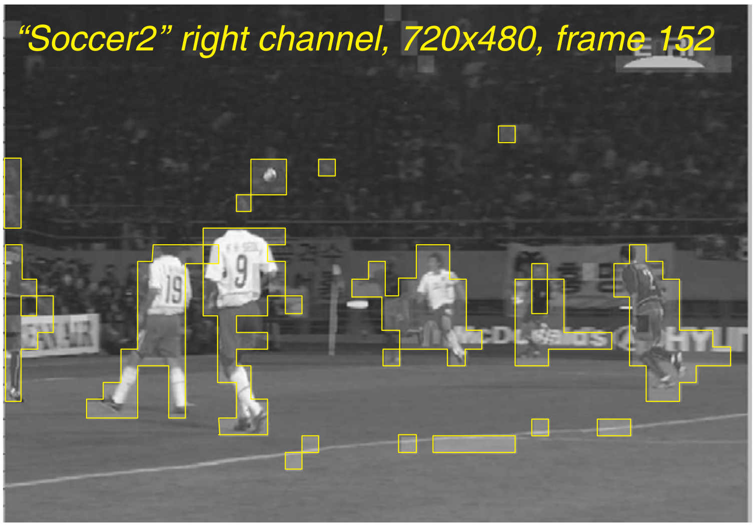


Figure 4.9: The subjective view of statistics of compensated block types. The highlighted blocks are DV-predicted. It shows that moving objects, such as soccer players or the ball, are usually DV-predicted.

of ME, $SAD_{ME}$,

$$Skip = \begin{cases} true, & \text{if } F_{diff} < Threshold_1 \text{ AND} \\ & \quad SAD_{ME} < Threshold_2 \\ false, & \text{otherwise.} \end{cases} \tag{4.10}$$

If *Skip* is $true$, the block is usually MV-predicted. Then DE is skipped, and the computational complexity is reduced. The $Threshold_1$ and $Threshold_2$ shown above are similar with the $Threshold$ in equation (8). They are sequence dependent and closely-related with rate-distortion condition. For example, in our simulation, we set 600 as $Threshold_1$ and 1200 as $Threshold_2$ in the sequence "Soccer2." Table 4.2 shows the hit rate and PSNR drop of five test sequences. The average hit rate is about 87% and the PSNR drop is about 0.02 dB.

### 4.2.4 Experimental Analysis and Comparison

#### Improvement of Coding Efficiency

The proposed system is compared with MPEG-4 SP [62] and temporal scalability profile (TSP) encoder [63]. Rate-distortion performance of only right channels (en-

Table 4.2: Hit Rate and PSNR Drop of Mode Pre-decision

| Sequences | Hit rate | PSNR drop (dB) |
|---|---|---|
| Soccer2 | 89.72% | 0.025 |
| Puppy | 94.30% | 0.018 |
| Golf | 90.15% | 0.023 |
| Flamenco | 85.36% | 0.046 |
| Race2 | 77.32% | 0.096 |

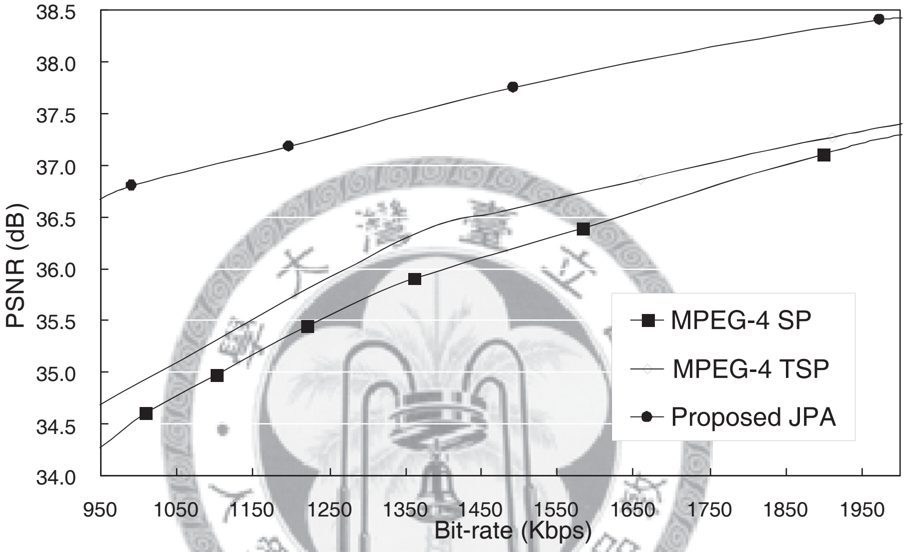


Figure 4.10: Rate-distortion curve of sequence “Soccer2.”

hancement layer) are compared because the left channels are all encoded by MPEG-4 SP. The performance of the left channels are similar with the right channels also encoded by MPEG-4 SP because of the similar video contents of two channels. Stereo video sequence “Race2” (320×240, 30 fps) and “Soccer2” (720×480, 30 fps) are taken as test sequences.

Figure 4.10 shows the comparison between the proposed algorithm, MPEG-4 TSP, and MPEG-4 SP. The proposed joint prediction scheme is 3 and 2 dB better than MPEG-4 SP and TSP, respectively. It shows that the joint block compensation scheme successfully reduces more redundancy. Figure 4.11 shows the performance of different coding tools. Without the joint prediction scheme (curve 1), the PSNR degradation is serious, as in the MPEG-4 SP. After the DE operation is turned on (curve 2), the coding efficiency is improved, just like the effect of multiple reference

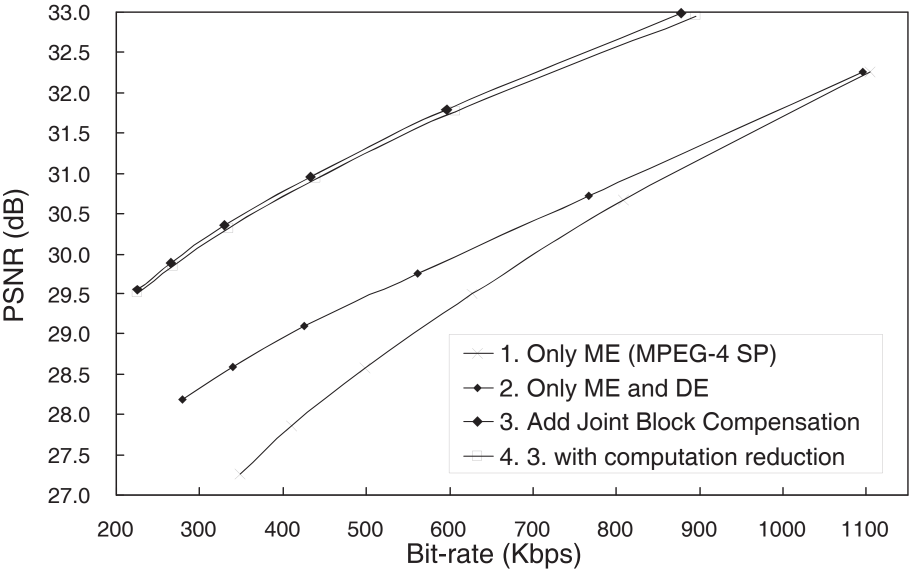


Figure 4.11: Rate-distortion curve of sequence “Race2.”

frames. When joint block compensation scheme is applied (curve 3), there is a 3 dB gain on coding efficiency. Besides, after applying MV-DV prediction scheme and mode pre-decision scheme (curve 4), not only the video quality is maintained but also most of the computational complexity can be reduced, which will be introduced in the next subsection.

### Reduction of Computational Complexity

Table 4.3 shows the reduction of search points. Note that every search point contains 256 substraction and addition operations. In our experiments, the search ranges of ME and DE are [±32, ±16] and [±32, ±8] for 320×240 sequences, and are [±64, ±32] and [±64, ±16] for 720×480 sequences, respectively. The proposed algorithm reduces about 80% computational complexity with negligible quality degradation. From Fig. 4.12, we can see that if the search range is reduced from ±16 to ±2, the PSNR degradation is only about 0.1dB, while both the computational complexity of ME and DE are greatly reduced.

Table 4.3: Search Points Reduction

| Sequences | Race2 | Flamenco | Golf | Soccer2 | Puppy |
|---|---|---|---|---|---|
| None | 2560 | 2560 | 2560 | 10240 | 10240 |
| MV-DV prediction | 1088 | 1088 | 1088 | 4352 | 4352 |
| Mode pre-decision | 1574 | 1160 | 1850 | 4204 | 3004 |
| Combination of 2 schemes | 609 | 400 | 706 | 1787 | 1277 |
| Search point reduction ratio | 76.20% | 84.37% | 72.41% | 82.54% | 87.52% |

國立臺灣大學 敦品勵學 愛國愛人

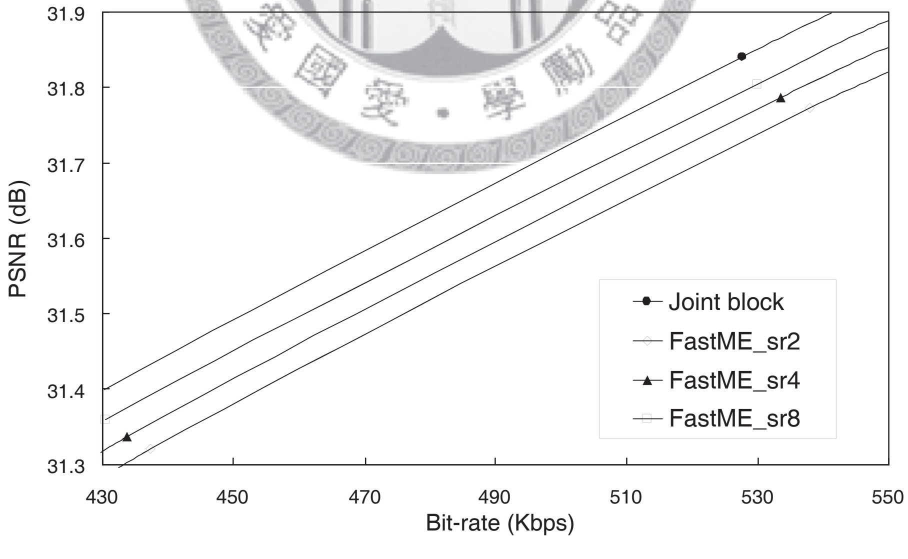


Figure 4.12: Rate-distortion curve of fast algorithm with various search ranges of sequence “Race2.”

## 4.3 Block Matching Algorithm for ME/DE

ME is a key unit in hybrid video coding systems. In the proposed stereo video coding system, additional ME and DE are required when encoding frames in the right channel. It increases the design challenges in large on-chip memory, memory bandwidth, and computational complexity. Due to the high hardware cost of FSBMA architecture, it is not suitable for the hardware architecture design of the prediction core. There are several kinds of fast motion estimation algorithms, such as three step search [43], four step search [44], diamond search [45], hexagon-based search [64], and hierarchical search [65]. Among those fast algorithms, hierarchical search block matching algorithm (HSBMA) can reduce not only the computational complexity but also the requirement of on-chip memory. Therefore, HSBMA is adopted with further improvement.

### 4.3.1 Modified Hierarchical ME/DE Block Matching Algorithm

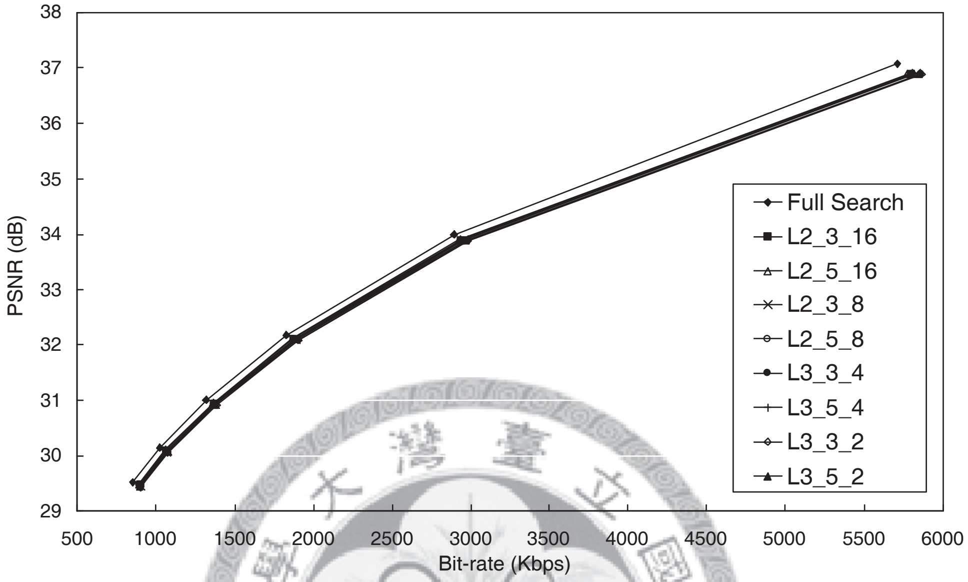
Compared with FSBMA, conventional HSBMA suffers from the problem of quality degradation [66]. Then error propagation will make the MVs not the best. To prevent this situation, the multiple candidates scheme is adopted. That is, several motion vector candidates are chosen after performing the coarser level block matching process (BMP). Take a 3-level HSBMA for example, level-2 is defined as the coarsest level, and level-0 is defined as the finest level. Three motion vectors with smaller SADs are first chosen in level-2 BMP. Then three level-1 BMPs begins. After that, only three rather than nine better MVs are chosen in these three level-1 BMPs. Then three level-0 BMPs starts. Finally, the best MV is chosen in these three SWs of level-0 BMPs.

Usually, more candidates chosen in the coarser levels and larger SWs in the refinement levels can provide better video quality. However, these are accompanied with some side effects. The computational complexity and the system memory bandwidth increase rapidly with more candidates chosen and larger SWs. To find the suitable combination of the number of candidates chosen and the search range, we did an experimental analysis focused on the rate-distortion and system bandwidth.

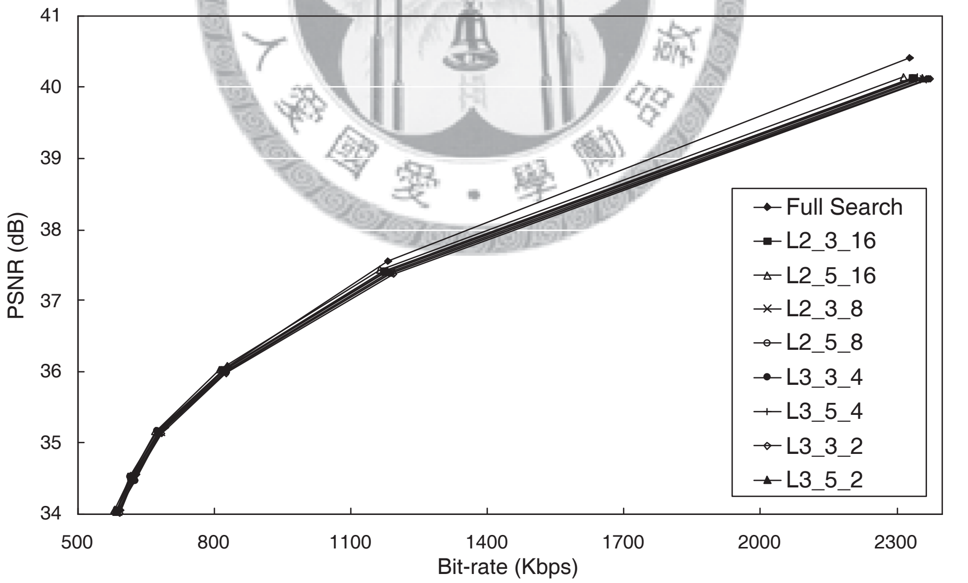
The D1 (720×480) size sequences are tested, and the search ranges are [-64, +63] in the horizontal direction and [-32, +31] in the vertical direction. Figure 4.13 shows the rate-distortion performance of various HSBMAs with different levels, different number of candidates, and different refinement ranges. For example, L_3_5_2 means 3 levels, 5 candidates refinement, and 5×5 refinement range are selected. Four video sequences are tested. Compared to FSBMA, we observed that the video quality is acceptable. However, the bandwidth requirement of data access is very much different between different cases. Table 4.4 shows the bandwidth requirement for various specifications. The reference frames are in the off-chip frame buffer. The bottom SW represents the SW of level-2 that is required to be loaded from the off-chip frame buffer. On the other hand, the top SW represents the SW of level-0. Take the algorithm "HS2" for example, a current block contains 256 pixels, thus it spends 256 bytes of memory bandwidth per block. The memory bandwidth of loading a bottom SW is reduced by utilizing level-C data reuse scheme [67] whatever the adopted algorithm is. However, when loading the mid or top SW, the regular data reuse scheme can not be applied. In this case, $(16+32)\times(16+32)\times5=$ 11520 bytes of memory bandwidth is required for the Top SW of every block. It shows that the bandwidth requirement of HSBMA is much higher than that of FSBMA. The main reason is that finer level cannot be applied with effective data reuse scheme, such as level-C data reuse scheme of FSBMA. The situation is more serious when the number of candidates chosen is five. From this table, HS7 is a suitable choice.

### 4.3.2 Near-Overlapped Candidates Reuse Scheme (NOCRS)

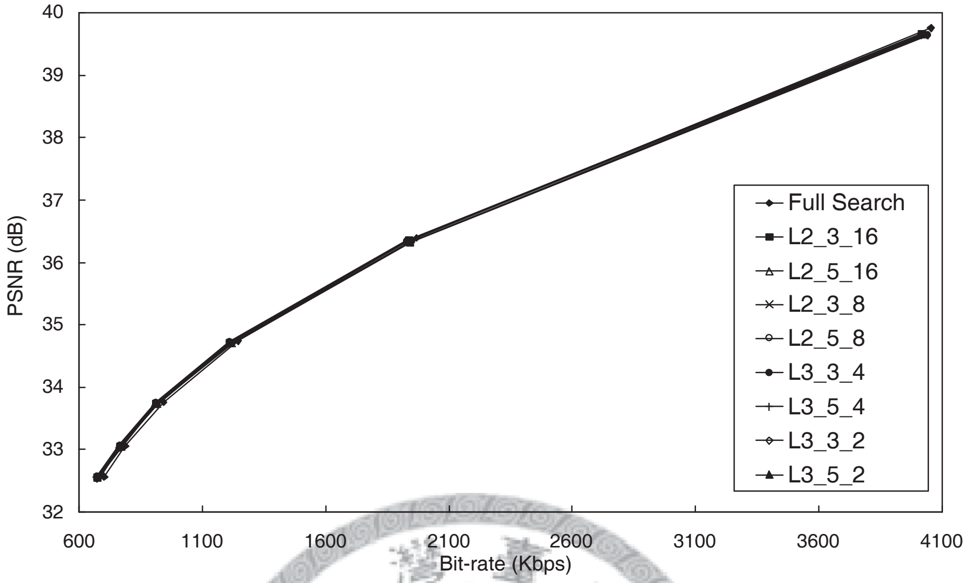
Still, the problem of bandwidth requirement is not fully solved. SW data cannot be reused effectively in the refinement levels, level-1 and level-0 BMPs. It will cause serious overhead on bus bandwidth. However, the experimental analysis shows that MVs of the best three candidates are usually very close. It means that the SWs of them in the next refinement level are partially overlapped. Therefore, we propose the near-overlapped candidates reuse scheme (NOCRS) to reduce the bandwidth requirement. After finding the best three candidates during level-2 and level-1 BMPs, three MVs will be checked by calculating their differences mutually. If the differences



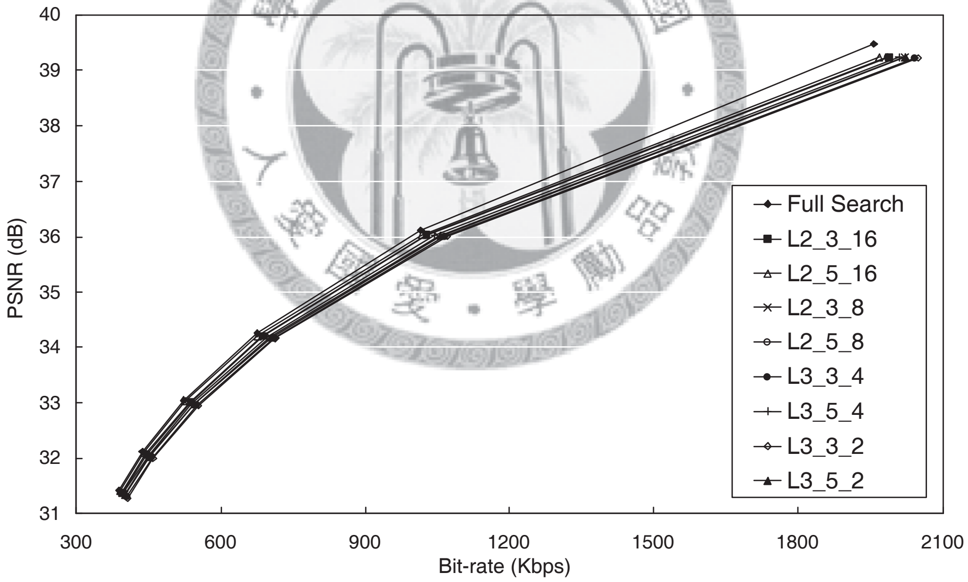
(a) Wendy



(b) Angel



(c) Toshiba



(d) Taxi

Figure 4.13: Rate-distortion performance of four test sequences (a) “Wendy,” (b) “Angel,” (c) “Toshiba,” (d) “Taxi.” Various levels, number of candidates, and refinement ranges are tested. For example, L_3_5_2 means 3 levels, 5 candidates refinement, and 4×4 refinement range are selected.

Table 4.4: Off-chip Memory Bandwidth Requirement of Various HSBMA

| Algorithm | FS | HS1[a] | HS2[b] | HS3[c] | HS4[d] | HS5[e] | HS6[f] | HS7[g] | HS8[h] |
|---|---|---|---|---|---|---|---|---|---|
| Current block (Byte/block) | 256 | 256 | 256 | 256 | 256 | 256 | 256 | 256 | 256 |
| Bottom SW (Byte/block row) | 6400 | 4000 | 4000 | 4000 | 4000 | 4000 | 4000 | 4000 | 4000 |
| Mid SW (Byte/block) | 0 | 0 | 0 | 0 | 0 | 768 | 1280 | 1200 | 2000 |
| Top SW (Byte/block) | 0 | 6912 | 11520 | 3072 | 5120 | 1728 | 2880 | 1200 | 2000 |
| Total bandwidth (MByte/sec) | 64.8 | 252.5 | 458.3 | 132 | 211 | 109.7 | 174 | 106 | 167.8 |

[a] 2 levels, 3 candidates refinement, [-16, +16] refine search range

[b] 2 levels, 5 candidates refinement, [-16, +16] refinement range

[c] 2 levels, 3 candidates refinement, [-8, +8] refinement range

[d] 2 levels, 5 candidates refinement, [-8, +8] refinement range

[e] 3 levels, 3 candidates refinement, [-4, +4] refinement range

[f] 3 levels, 5 candidates refinement, [-4, +4] refinement range

[g] 3 levels, 3 candidates refinement, [-2, +2] refinement range

[h] 3 levels, 5 candidates refinement, [-2, +2] refinement range

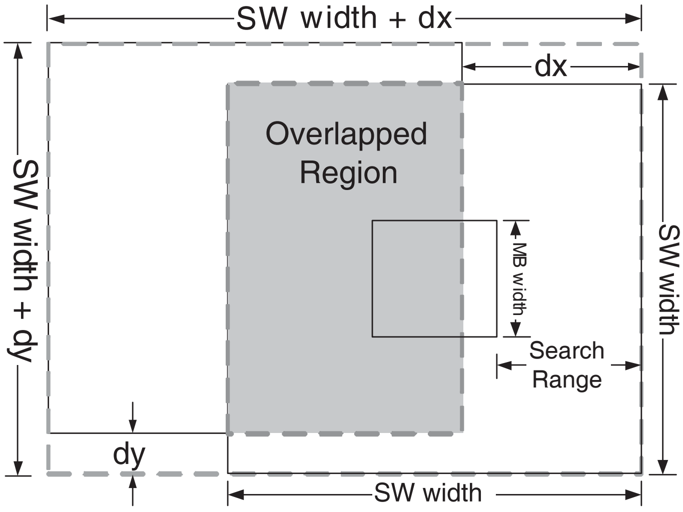


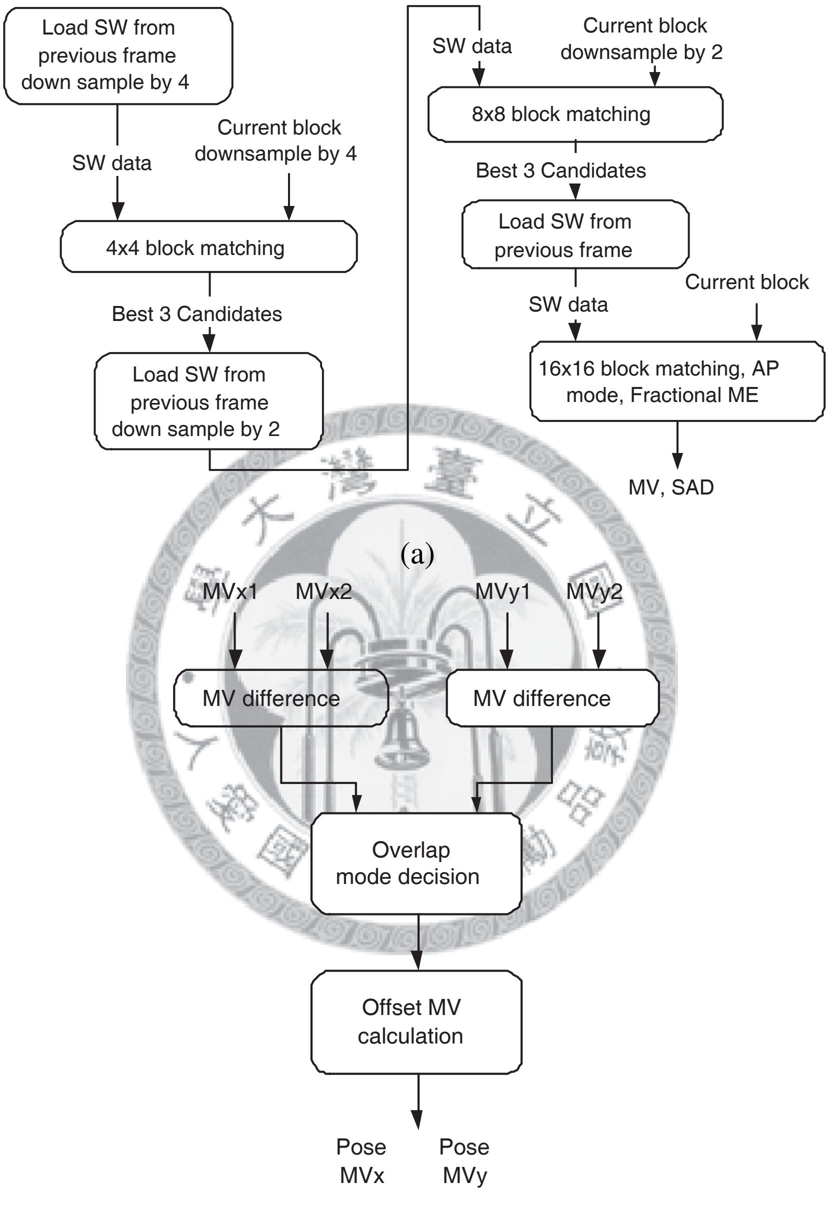
Figure 4.14: The union of overlapped SWs. The SWs in the next refinement level are partially overlapped.

are smaller than a threshold, the overlapping condition is satisfied. The threshold is statistically analyzed according to the designer's specifications. We set 4 and 2 as the thresholds for motion vector differences in the x and y direction, respectively, in the simulation. The union of two or even three SWs is loaded only once from the off-chip frame buffer, as shown in Fig. 4.14.

In summary, Fig. 4.15 shows the flow chart of HSBMA with NOCRS. NOCRS not only reduces off-chip memory bandwidth successfully but also avoids unnecessary computation on duplicated search candidates in two separate SWs. Table 4.5 shows the system bandwidth requirement of three ME/DE algorithms. After NOCRS is applied, 35.5% system bandwidth can be saved. Although FSBMA still requires less system bandwidth by regular data reuse scheme, for example, level-C data reuse scheme, the proposed HSBMA with NOCRS has much less on-chip memory requirement and computational complexity.

## 4.4 Prediction Core Architecture

The overall architecture of the prediction core is shown in Fig. 4.16. There are nine main units: control unit, reference shift register network (RSRN), current register set (CRS), current mux network (CMN), 128-PE adder tree, comparison tree (CT),


Load SW from previous frame down sample by 4
SW data
Current block downsample by 4
4x4 block matching
Best 3 Candidates
Load SW from previous frame down sample by 2
SW data
Current block downsample by 2
8x8 block matching
Best 3 Candidates
Load SW from previous frame
SW data
Current block
16x16 block matching, AP mode, Fractional ME
MV, SAD
MVx1
MVx2
MVy1
MVy2
MV difference
MV difference
Overlap mode decision
Offset MV calculation
Pose MVx
Pose MVy

(a)

(b)

Figure 4.15: Data flow of the proposed ME/DE algorithm. (a) Flow of proposed HS-BMA with NOCRS. (b) Flow of near overlapped candidates reuse scheme (NOCRS).

Table 4.5: System Bandwidth Requirement of Three BMAs

| Data loaded from<br>off-chip frame buffer | FSBMA | HSBMA<br>without NOCRS | HSBMA<br>with NOCRS |
|---|---|---|---|
| Current frame | 9.9 | 9.9 | 9.9 |
| SW for 4×4 BMP | 0 | 3.4 | 3.4 |
| SW for 8×8 BMP | 0 | 46.4 | 29.5 |
| SW for 16×16 BMP | 55 | 46.4 | 31 |
| Reconstruct frame | 9.9 | 9.9 | 9.9 |
| DS Reconstruct frame | 0 | 7.4 | 7.4 |
| Total bandwidth (MByte/sec) | 74.8 | 123.4 | 91.1 |

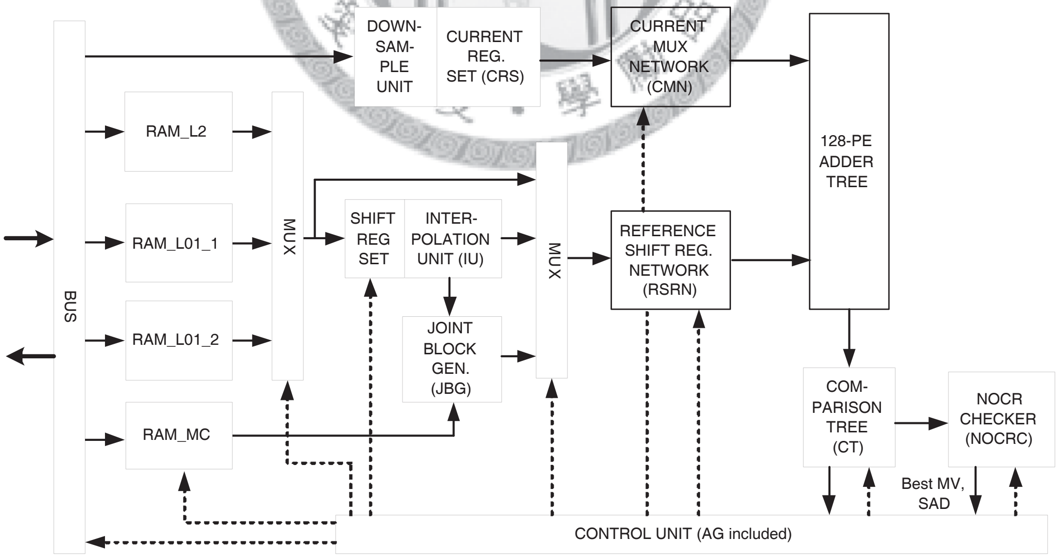


Figure 4.16: Architecture of the prediction core. The architecture performs the prediction task in both channels.

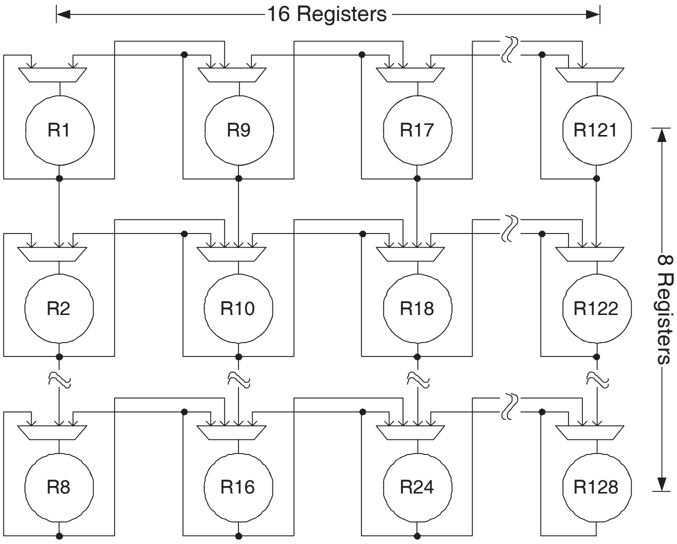


Figure 4.17: RSRN. it is composed of a set of reconfigurable shift register array so that the pixels can shift downward, leftward, and right ward.

NOCR checker (NOCRC), interpolation unit (IU), and joint block generator (JBG). The CMN outputs three kinds of current blocks for three-level BMPs. The RSRN is composed of a reconfigurable shift register array. After data loading of SW is finished, RSRN starts to fetch data from the on-chip memory. Meanwhile, 128-PE adder tree generates SADs. Then CT compares these SADs in one cycle. The best three candidate MVs are chosen for refinement. NOCRC checks the degree of overlapping and outputs the post-processed MVs to the address generator (AG) in the control unit, which decides when to begin the next level BMP. IU generates sub-pixels in the half pixel refinement process. JBG generates joint blocks for mode decision for improving the coding efficiency of the stereo video. The detailed architecture will be described in the following. Furthermore, data reuse scheme and memory organization are also shown. Besides, a new scheduling is proposed to reduce the demand of on-chip memory and off-chip memory bandwidth [68].

### 4.4.1 RSRN

To achieve the design goal of hierarchical BMP with only one hardware resource, RSRN is composed of a reconfigurable shift register array, which consists of 128 8-bits registers, as shown in Fig. 4.17. It has high reconfigurability and can reconfigure

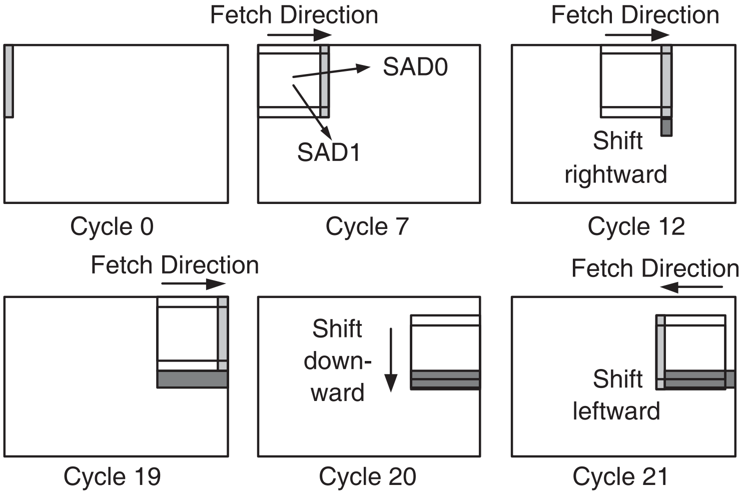


Figure 4.18: Data flow of level-1 BMP. Level-2 and level-0 BMP have the similar flows with level-1 BMP.

臺灣大學 國立

to shift downward, leftward, and rightward. After the SW are loaded from off-chip to the on-chip memory, one column of SW pixels are fetched to RSRN every cycle. There are no bubble cycles when the search position is changed in the vertical position. An example of the detailed data flow is shown in Fig. 4.18, which is the data flow of level-1 BMP with the maximum search range [-6, +6] in our design, and the SW is 20 words×20 bits. When BMP starts, RSRN fetches one column of SW pixels at each cycle. At cycle 7, all the candidate block data of search positions (-6, -6) and (-6, -5) are stored in RSRN. Thus SAD0 and SAD1 are generated at cycle 7. Then, two SADs are generated in each cycle. At cycle 12, two additional pixels must be pre-fetched in additional registers to avoid bubble cycles during the reconfiguration step. At cycle 19, these additional sixteen pixels are ready to input to RSRN. Then the RSRN shifts downward, and two SADs of search position (6, -4) and (6, -3) are generated without any bubble cycles. At cycle 21, RSRN changes the connection configuration again and shifts leftward. In this way, all the bubble cycles can be avoided besides the initial cycles, so the utilization is near 100%. Level-2 and level-0 BMP have the similar flows with level-1 BMP.

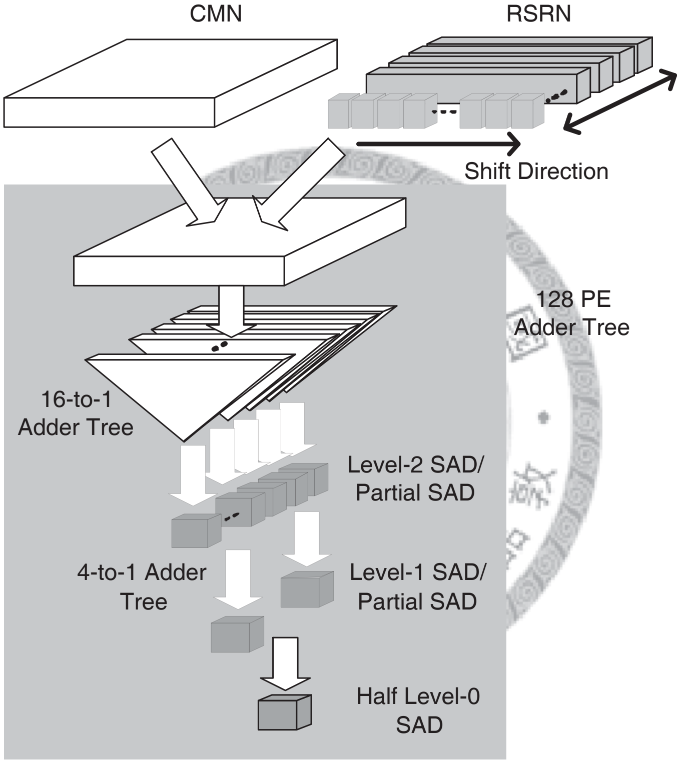


Figure 4.19: 128-PE adder tree. The pixels of the current block and the reference candidates are fetched into 128-PE adder tree every cycle.

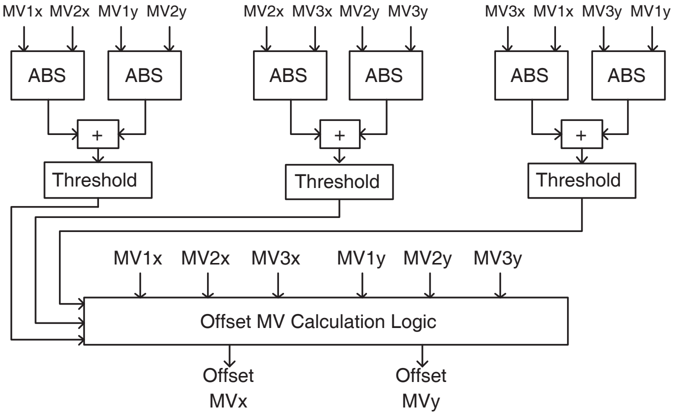


Figure 4.20: NOCRC. The MV differences are calculated mutually, and then the overlapping condition is decided.

### 4.4.2 128-PE Adder Tree

Figure 4.19 illustrates 128-PE adder tree. The pixels of the current block and the reference candidates are fetched into 128-PE adder tree every cycle. Except for several cycles in the beginning for data preparing, SADs of eight candidate blocks in level-2 BMP, two candidate blocks in level-1 BMP, or half candidate blocks in level-0 BMP can be derived. Then the CT compares the SADs and chooses the proper candidates for the next level BMP.

### 4.4.3 NOCRC

The architecture of NOCRC is shown in Fig. 4.20. The best three MVs chosen by CT are inputted to the NOCRC after level-2 and level-1 BMPs. The MV differences are calculated mutually, and then the overlapping condition is decided. For example, if three outputs of threshold units are all logic 1, it means SW of next level should be loaded only once rather than three times. It can effectively reduce over 35% unnecessary data access from off-chip. Furthermore, it also reduces unnecessary computation and saves processing cycles.

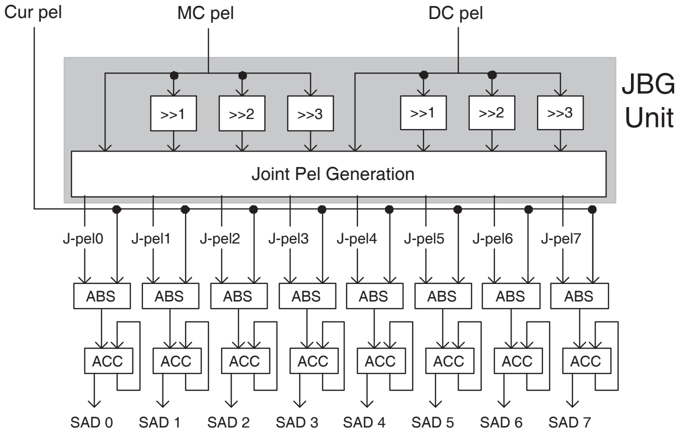


Figure 4.21: One of the sixteen JBG units. Every JBG unit generates one column of all the joint block in one cycle.

### 4.4.4 JBG

When the ME of the right channel is finished, the best candidate block must be stored for the joint block generation step. The best candidate block is loaded into on-chip RAM_MC, as shown in Fig 4.16. After DE of the right channel is finished, mode decision for the joint block starts. Figure 4.21 shows one of the sixteen JPG units in JBG. Only adders are used to generate weighted sum in the joint pel generation. Every JBG unit generates one column of all the joint block in one cycle. After sixteen cycles, eight SADs of joint block candidates are generated. Then they are inputted to the CT to choose the best SAD, and the best mode is derived as well.

### 4.4.5 Memory Organization and Data Reuse Scheme

Figure 4.16 shows that SRAM_L2 stores the level-2 SW data for the left and right channel. RAM_L01_1 and RAM_L01_2 store the refinement SW data for level-1 and level-0 BMPs. Since there might be 1 to 3 SWs for block matching, RAM_L01_1 and RAM_L01_2 are accessed with ping-pong mode to store SW data from the off-chip buffer. RAM_MC buffers the best ME candidate block in the right channel, which is used for joint block generation. In the proposed architecture, only 20.75 Kbits on-chip SRAM are required, which is only 11.5% requirement compared with FSBMA.

Because of the regular data access of level-2 BMP, level-C reuse scheme is applied for the SW loading in level-2 BMP. The disadvantage of conventional HSBMA [44] is that the SW required for refinement level (level-1, level-0) cannot be reused due to its irregular flow. It increases the data access burdens. However, the proposed NOCRS effectively solves this problem. In other words, data reuse scheme is also applied in level-1 and level-0 BMP to save bandwidth.

### 4.4.6 Proposed Scheduling for Stereo Video Coding System

The proposed scheduling is modified from our prior stereo video system [68] for hardware implementation consideration. The original frame-based scheduling of the prediction engine is shown in the upper part of Fig. 4.22. It shows that ME in the left channel cannot start until MVs and DVs of all the blocks of a frame in the right channel are derived. However, the SW for DE in Fig. 4.3 is enclosed by the search window for ME in the left frame. In the original scheduling, $SW_{l,DE}$ is loaded twice from the off-chip frame buffer. Therefore, this wastes bus bandwidth.

This problem can be solved by the new scheduling, as shown in the lower part of Fig. 4.22. Before DE of $B_r$, $SW_{l,ME}(B_l)$ is loaded from off-chip instead of $SW_{l,DE}(B_r)$. After DE and joint block mode decision are done, ME of $B_l$ in the left channel starts. No loading process is needed for ME of $B_l$. On-chip memory for $SW_{l,DE}(B_r)$ is shared with that for $SW_{l,ME}(B_l)$. The proposed scheduling reduces both the requirements of off-chip memory bandwidth and loading cycles. Moreover, 23% on-chip memory can be saved.

### 4.4.7 Chip Implementation

The proposed prediction core architecture was verified by the VLSI implementation. The processing capability is listed as follows: 720×480 frame size and 30 frames per second (fps) both in the left and right channels. In the ME case, the search range is [-64, +63] in the horizontal direction and and [-32, +31] in the vertical direction. While in the DE case, the search range is [-64, +63] in the horizontal direction and [-16, +15] in the vertical direction. The die photograph is shown in Fig. 4.23. There are three groups of on-chip single-port SRAM on the chip. The core size is 2.13×2.13

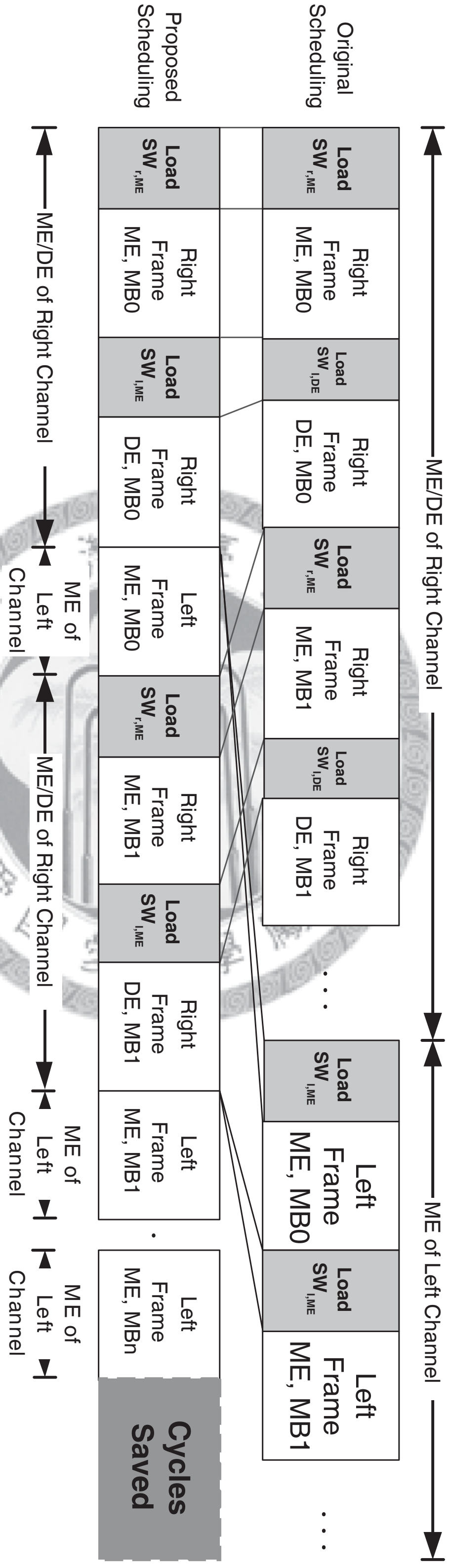


Figure 4.22: Proposed scheduling of the stereo video system. No loading process is needed for ME in the left channel.

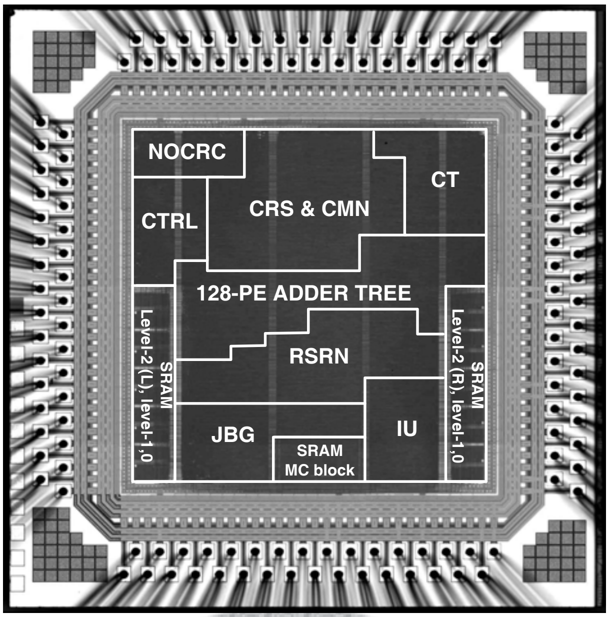


Figure 4.23: Die photograph of the prediction core design.

$mm^2$. The detailed chip features are shown in Table 6.3.

### 4.4.8 Comparisons

So far there is no other architecture of prediction core for stereo video systems. The proposed architecture can also perform ME. Table 4.7 shows the comparison of FSBMA and the proposed HSBMA. A search point means one SAD calculation process of a MB. Since the computational complexity of search points in different levels of hierarchical search algorithms is not the same, they are normalized first. Compared with FSBMA, only 11% of on-chip memory is required, and the computational complexity is also greatly reduced. The number of PEs of HSBMA architecture is 3.3% of FSBMA architecture. Figure 4.24 shows the rate-distortion performance of FSBMA and proposed HSBMA with NOCRS. Three video sequences are tested by simulation with hardware description language (HDL). Compared with FSBMA, the proposed algorithm maintains good objective video quality.

Table 4.8 shows the comparison with the previous HSBMA architecture [65]. Although the hardware cost such as logic gate count and on-chip memory are similar, the proposed architecture provides more functionalities, such as DE and joint block compensation for the stereo video prediction, with less PE and system bandwidth

Table 4.6: Chip Specifications

| | |
|---|---|
| Technology | TSMC 1P6M 0.18*um* |
| Package | 128 CQFP |
| Core size | $2.13 \times 2.13 mm^2$ |
| Logic gate count | 137,838 (2-input NAND gate) |
| On-chip memory | 20.75 Kbits |
| Maximum frequency | 100 MHz |
| Power supply | 1.8 V |
| Power consumption | 95.85 mW @ 100 MHz |
| Search range | ME: horizontal [-64, +63], vertical [-32, +31]<br>DE: horizontal [-64, +63], vertical [-16, +15] |
| Processing capability | 30 D1(720x480) frames/sec in left and right channels simultaneously, including 2 ME and 1 DE operations |

Table 4.7: Algorithm Comparison

| Algorithm | FSBMA | HSBMA with NOCRS |
|---|---|---|
| On-chip memory | 180 Kbits | 20.75 Kbits |
| search points/MB[a] | 8192 | 100 - 234 |
| PE requirement | >4096 | 128 |
| Quality drop | 0 | <0.2 dB |
| Bandwidth | 74.8 MB/s | 91.1 MB/s |

[a] The search point is normalized because of different computational complexity in various BMPs.

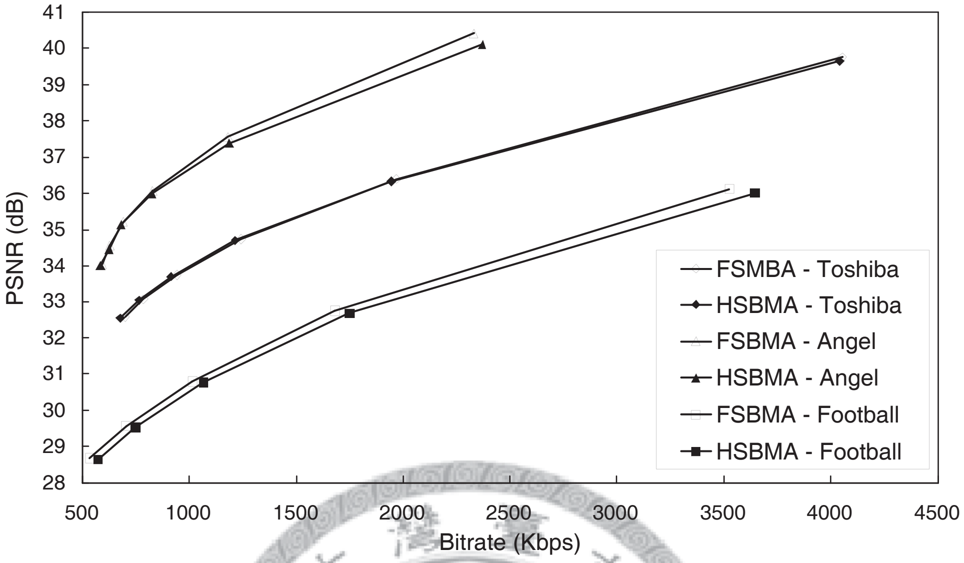


Figure 4.24: Comparison of rate-distortion between proposed HSBMA and FSBMA.

Table 4.8: Architecture Comparison

| Architecture | [65] | This work |
|---|---|---|
| Area | 140K | 137K |
| No. of PEs | 256 | 128 |
| Memory | 2.5 KBytes | 2.6 KBytes |
| Bandwidth[a] | 125 MBytes/sec [b] | 91.1 MBytes/sec |
| Frequency | 54 MHz | 81MHz |
| Function | ME | 2 ME, 1 DE, joint block compensation |

[a] Only mono-channel ME in a P-frame is considered

[b] The system bandwidth requirement is estimated

requirement. The improvement results from the proposed NOCRS, scheduling, and efficient reconfigurability of the architecture. Besides, it can be easily integrated into mono or stereo video coding systems because of its various functionalities.

## 4.5 Conclusion

This chapter presents an efficient structure of the prediction core in stereo video coding systems from algorithm level to hardware architecture level. A stereo video hybrid coding system with the joint prediction scheme is designed for the purpose of overcoming the design challenges of poor coding efficiency and high computational complexity. Compared with MPEG-4 TSP and SP, the coding efficiency is improved by 2–3 dB. Besides, 80% of the computational complexity is reduced. Moreover, a hardware-oriented stereo video prediction algorithm and the associated hardware architecture for the prediction core are also presented. Compared with FSBMA, the proposed HSBMA greatly reduces hardware cost, while it still maintains good video quality. With NOCRS, the problem of critical memory bandwidth requirement can be solved by checking the overlap degree of the three SWs in the refinement levels. The hardware architecture is co-designed with the proposed algorithm with a set of reconfigurable shift register array and its related circuits, which can be configured for all the scan directions for three-level BMPs. Moreover, the proposed scheduling not only reduces cycles for loading data from off-chip frame buffer but also eliminates on-chip memory for level-2 of DE. It shows that the architecture is area-efficient and has both good processing capability and functionality.

There are still some extensions in the proposed stereo video hybrid coding system. There might be another suitable pattern combination of the joint block, such as gradual blending modes and H.264 block modes, for better coding efficiency. Mode decision can be further optimized by the Lagrange multiplier. The rate-control algorithms for stereo video coding can also be further explored. On the other hand, MV-DV prediction and mode pre-decision can be implemented in the architecture for higher area efficiency. They are challenged research topics and also belongs to our future work.

# Chapter 5

# Analysis and Algorithm Design of High-throughput Prediction Core for 3D/Quad HDTV Videos

Among all the functional blocks in video encoding, the most resource hungry part is inter prediction [69]. The module handling inter prediction is called prediction core in this dissertation. Inter prediction refers to the process of generating predicted pixel values from previously decoded frames, which are called reference frames. These frames could be forward or backward in displaying time, since the coding order is different from displaying order. Inter prediction retrieves displaced blocks from the decoded frames, and the displacement is specified by the motion vectors (MVs), which are estimated by the motion estimation (ME). In this chapter, an efficient predictor-centered ME algorithm is proposed for high definition (HD) videos. The proposed algorithm is designed for the VLSI architecture which is introduced in Chapter 6 and Chapter 8. With the concept of motion information preserving, the proposed algorithm saves 96% computational complexity while maintains the quality with only 0.013dB loss. In addition, the proposed algorithm is suitable for mapping to the VLSI architecture and overcomes the design challenges of large oh-chip memory and high external memory bandwidth.

## 5.1 Introduction

In video coding systems, motion estimation (ME) is an essential part to reduce the temporal redundancy in video sequences. It consumes most of the computation in the system, and the coding efficiency greatly relies on the effectiveness of the ME algorithm. Therefore, reducing the computation without sacrificing the rate-distortion (RD) performance is the target of such systems.

Modern ME algorithms generally consist of two parts: refining center decision and refinement algorithms. The first part chooses some motion vectors (MVs) to be the predictors of refining center, mostly from highly correlated sources like the spatial and temporal neighboring macro-blocks (MBs) or blocks of other sizes if variable block-size is supported. Algorithms like PMVFAST [70] and EPZS [71] focus on this part. After that, the predictors of refining center are evaluated by block-matching algorithm, and then refinement occurs around the best refining center. Finally, the best matched block with lowest Lagrange cost is derived. Algorithms like diamond search [45], hexagonal search [64], three-step-search [43], four-step-search [44], and exhaustive search fall into this category. Because the refinement algorithms only do the block-matching around the refining center, the search range can be greatly reduced if the refining center are close enough to the optimal solution, hence the ME computation is lowered to an acceptable level. In modern video coding standards like MPEG-4 and H.264/AVC, there are numerous block sizes. Exhaustive search of all these block sizes would require enormous computation, so we need to reduce the computation even more. Consequently, the accuracy of the refining center becomes more critical. With the introduction of variable block-size, a new type of refining center predictor exploiting the correlation between different blocks within an MB emerges. We call them intra predictor in contract to the traditional inter predictor. These newly added predictor and their related interaction also makes the design of refining center decision algorithm more complex.

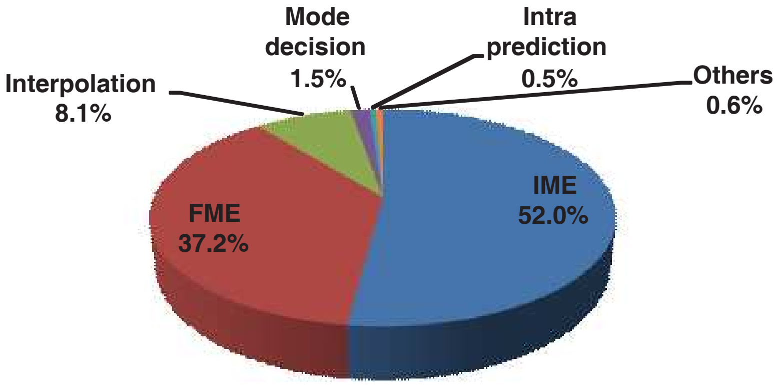


Figure 5.1: Run-time profiling of H.264 encoding

## 5.2 Design Challenges

In ultra high-throughput prediction core design, several challenges make it extremely difficult to implement such a system. These challenges include intensive computation, and the trade-off between external memory bandwidth and on-chip memory size. In the following subsections, each design challenge is discussed.

### 5.2.1 High Computational Complexity

Fig. 5.1 shows the run-time profiling of H.264 reference software encoding a HD video sequence. We can observe that two major parts in the prediction core, integer motion estimation (IME) and fractional motion estimation (FME), contribute more than 89% of total computation, and IME alone needs more than half the total computation. The computation of prediction core also depends on the video resolution. In the log-log plot of IME complexity vs. video resolution in Fig. 5.2, we can see that full search algorithm needs $O(n^2)$ growth of computation, where n is for the area of video frame. Hierarchical search is a fast search algorithm widely adapted in high definition (HD) video encoders, especially in hardware implementations [72]. However, it can only provide a constant factor of computation reduction, and the reduction ratio is not high enough. A high-end quad-core CPU designed by Intel, QX9770, supports 60 Giga instruction per second (GIPS). However, for $4k \times 2k$ res-

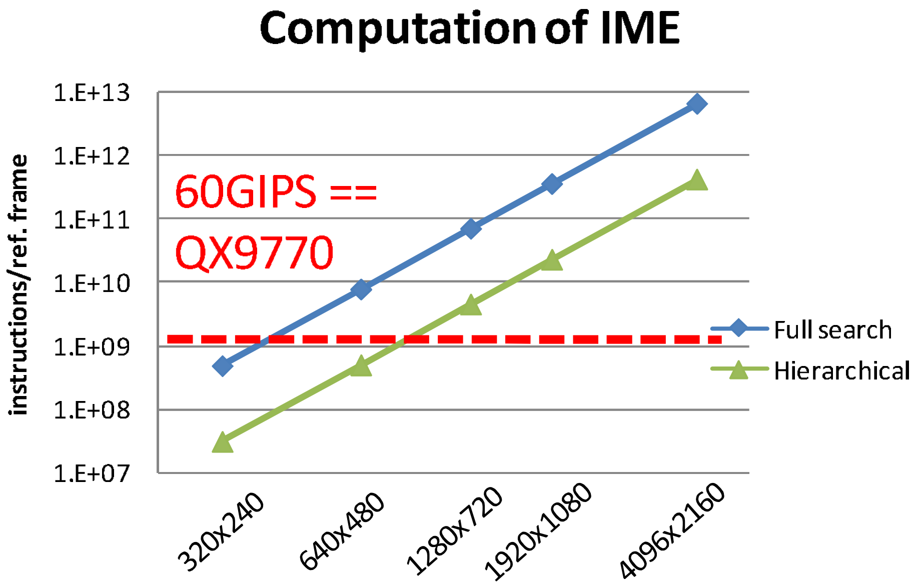


Figure 5.2: Computational complexity of IME

olution, the computation of IME is almost four orders of magnitude higher than the horse power of QX9770. Single instruction multiple data (SIMD) techniques can boost the performance for an order of magnitude on modern processors [73, 74], and even though the power efficiency of application-specific IC (ASIC) designs is around two orders of magnitude better than a general-purposed CPU, the performance gap is not bridged. From this analysis, we can conclude that it is fairly reasonable to optimize the prediction core and reduce the required computation.

### 5.2.2 High External Memory Bandwidth and Large On-Chip Memory Size

Assuming we want to encode a video sequence with $4096 \times 2160p$ resolution, 24 frames per second, and we use $\pm 256 \times \pm 128$ search range, level C data reuse scheme [75], bi-directional frames (B-frames), then the on-chip SR memory would contain $(256 \times 2 + 16) \times (128 \times 2 + 16) \times 2$ pixels, which occupies 281 KB of on-chip SRAM, and the external memory bandwidth would be $4096 \times 2160 \times ((128 \times 2 + 16)/16) \times 2 \times 24(pixels/s) = 6.72(GB/s)$. Nevertheless, for a high-end SoC system running at 200 MHz with a fairly wide 128-bit memory bus, the throughput can only achieve 3.2 GB/s at 100% bus utilization. As we can see, the bus bandwidth budget

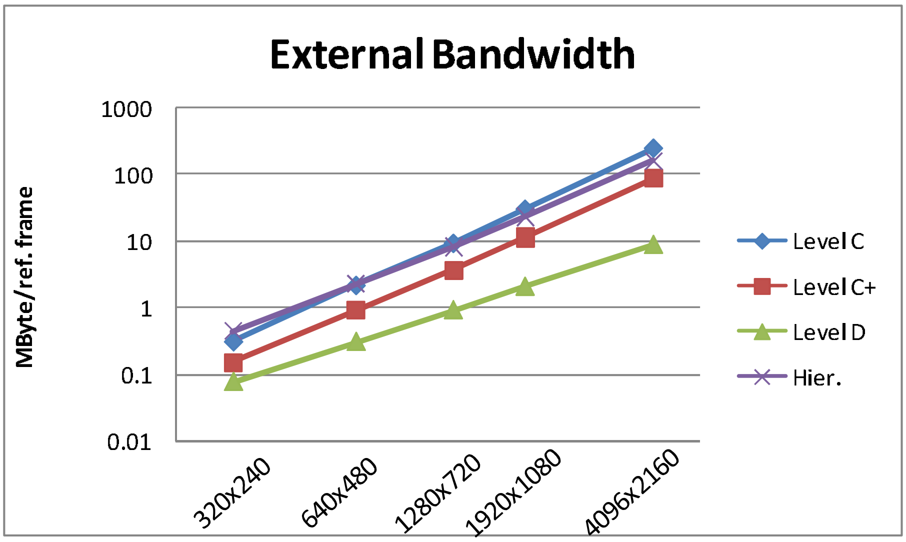


Figure 5.3: External bandwidth of different data reuse schemes of search range buffer

is tight even on a high-end SoC system.

When the frame resolution changes, it is reasonable to adjust the search range proportionally. If we fix the search range to 10% of frame in each direction, for example, $\pm 192 \times \pm 108$ search range for $1920 \times 1080p$ resolution, then the log-log plot of external bandwidth and on-chip SRAM size vs. frame size is shown in Fig. 5.3 and Fig. 5.4. In these figures, 4 different data reuse schemes, level C, level C+ [76], level D, and hierarchical search, are compared. In TSMC 90nm technology, SRAM with capacity of 62.5KB is equivalent to a million gates in area. For $4k \times 2k$ applications, The bandwidth is higher than what a high-end embedded system can support, and the area of on-chip SRAM for reference frame buffer is unreasonably large. Previous work shows that the SR utilization is only 30% on CIF video, and it decreases to 15% on D1 video [77]. That is to say, many data read to the SR buffer are never used. Further investigation reveals the trend of low utilization still applies to HD video. However, if we try to save on-chip memory by directly shrinking the search range, the rate-distortion (RD) performance would be greatly hurt. Therefore, a smarter strategy to reduce on-chip memory usage without sacrificing the coding efficiency is desirable. Moreover, when the video resolution gets higher, the ratio of (cache size/level C buffer size) can be smaller. From the work in [77], if we want a reasonable RD performance, cache size is $1/3$ of level C buffer size for D1 video,

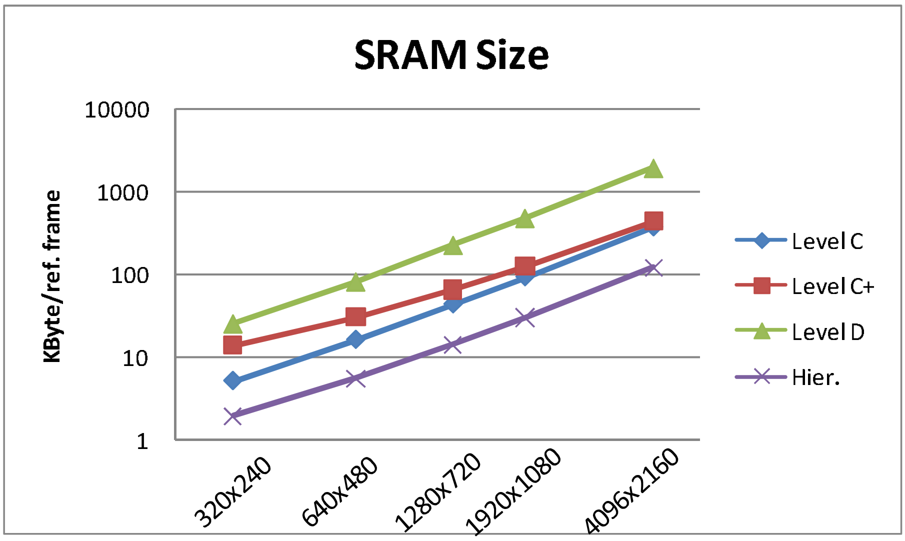


Figure 5.4: On-chip SRAM size of different data reuse schemes of search range buffer

and the ratio is 2/3 for CIF video. From this trend, we can expect the ratio be smaller on HD video. As a result, it makes more sense to utilize cache-based architecture on HD video coding systems.

## 5.3 Predictor-Centered Search Block Matching Algorithm

Since we tend to cut down the computation by limiting the search range and decreasing searching candidates, low-quality predictors could make the refinement trapped in the local minimum, thus resulting in low R-D performance. In order not to be trapped in local minimum and increase the robustness of the whole ME algorithm, multiple predictors are necessary. The predictors are classified into two types, inter and intra predictors.

### 5.3.1 Inter Predictor

The source of inter predictors are from the spatial and temporal neighbors outside the current MB. Generally, the predictors include the motion vector predictor (MVP)

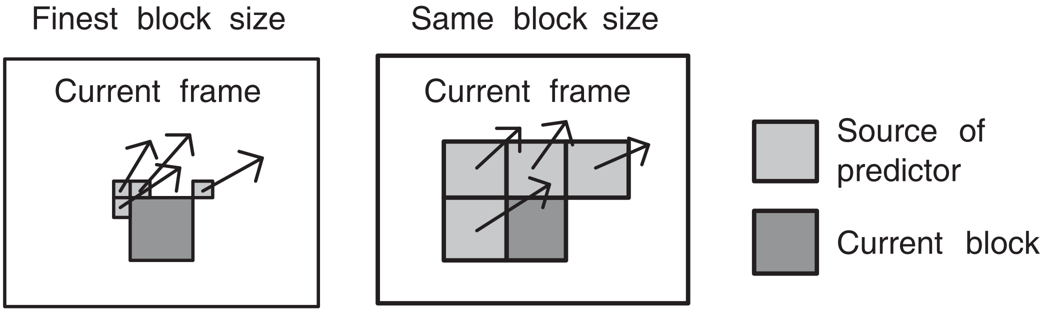


Figure 5.5: Block sizes of inter predictors.

defined in the coding standard, which is the median of MVs from left, top, and top-right blocks. In addition, these three MVs from left, top and top-right blocks, zero vector and some more spatial and temporal neighbors may also be considered. When variable block-size is enabled, the concept of MVs from neighbor blocks becomes obscure because an MB can be split into several blocks with different MVs for each one. As for the MVP defined in H.264/AVC, the three MVs are from the finest block sizes. However, the selection of inter predictors is not defined in the standard, so it is up to the encoder.

Two schemes are chosen for comparison. One is to use the MVs from the finest block size available, and this is similar to the process of MVP selection defined in H.264/AVC. The other one is to use the best-matching MV from the neighbor blocks with the same size. In order to support this algorithm, several additional MV fields with different granularities must be stored in the encoder. These two schemes are illustrated in Fig. 5.5, where ME is performed on a 16×16 block, and variable block-size supports down to 4×4 size.

### 5.3.2 Intra Predictor

The intra predictors are from other sub-blocks within the same MB. In order to achieve better computational efficiency, only the sources with higher correlation is used, so the selection is generally hierarchical. For example, a 4×4 block doesn't use an MV of an 8×8 block not covering it. When applied in ME, intra predictors can be used in any block sizes, but they are not available for the very first block evaluated in this MB, which is often the 16×16 block. Figure 5.6 shows that when the

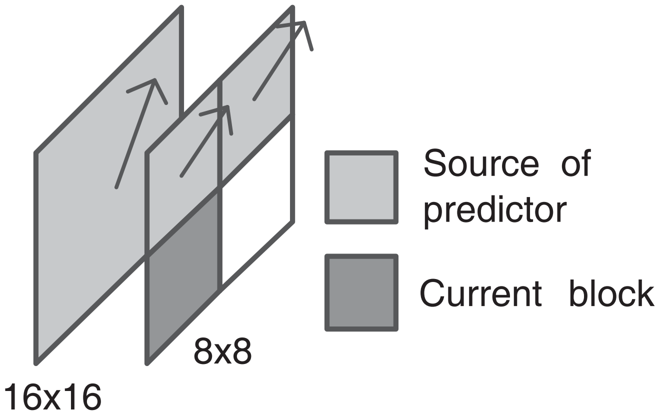


Figure 5.6: Intra predictors.

third 8×8 block is being evaluated, the result of 16×16 block and previous two 8×8 blocks can be used as predictors. For those blocks which are not evaluated first in current MB, both inter-MB and intra-MB predictors are available. However, the gain from multiple predictors might saturate if the number of predictors are too large. As a result, whether the inter-MB predictors can be skipped is a design issue.

## 5.4 Proposed Algorithms

According to the analysis above, the design challenge includes high computation and high bandwidth. As a result, it is unavoidable to modify the prediction core and make it more efficient and memory-friendly. We developed several algorithms for the prediction core to reduce computation requirement, consume less internal and external memory bandwidth, and access the external memory more regularly while keep near-lossless rate-distortion (RD) performance. These techniques are illustrated in the following subsections.

### 5.4.1 Motion Information Preserving

The MVP defined in the coding standard is derived from the MV field that is available at the decoder. As a result, when an MB is intra-coded, its motion information is not encoded, and no MV is available at the decoder. However, if the MV pointing to the best matched block is stored at the encoder, even if the intra mode wins the inter/intra mode decision, the MV can still be used as a RC predictor for neighbor MBs. This would not break the compatibility with the standard because the RC predictor only affects the ME quality. This way, motion information is reused rather than just being discarded. This technique can be integrated in the inter-MB predictors but not the intra predictors because either all or none of the sub-blocks are intra-coded. In short, the main difference is not to clear the MVs even if the block is intra-coded.

### 5.4.2 Data Dependency and Regularity of Access Pattern

According to the algorithms described in Section 8.1, the refining center depends on the ME result of other blocks, so the access pattern on the reference frames is harder to predict. This causes no problem on modern computers. However, on some other computing platforms without luxuriously huge data cache and high working frequency, parallelism and memory de-coupling become more essential. Taking dedicated hardware, which resides on the other end of the agility spectrum, for example, zero-vector-centered full search pattern is usually adopted because of its extremely regular access pattern and low data dependency. For platforms in-between, such as GPGPU, ASIP, VLIW, etc, more irregularity and data dependencies are tolerable. For more types of computing devices, we need to remove some of the data dependencies and make the access pattern more regular in order to enable high parallelism, while preserving the R-D performance.

By adopting adder tree architecture for calculation of sum of absolute difference (SAD), partial results of SAD can be reused with almost no additional cost [78]. For example, if we do the block-matching on a 16×16 block, all the SAD values of smaller partitions can be obtained for free. One way to leverage the free partial SAD is to share the inter-MB predictors among sub-blocks, so that the SAD values of different sub-blocks at these same locations can be reused. In order to maximize this reusing, MV field with 16×16 granularity is used for predictors. In Fig. 5.7, two 8×8 blocks are being evaluated, and two granularities of the MV field are shown for comparison. We can observe that when the granularity is 16×16, all the sub-blocks share the same predictors, while only part of the predictors can be reused if the granularity is 4×4. In contract to the situation described in Section 5.3.2, whether

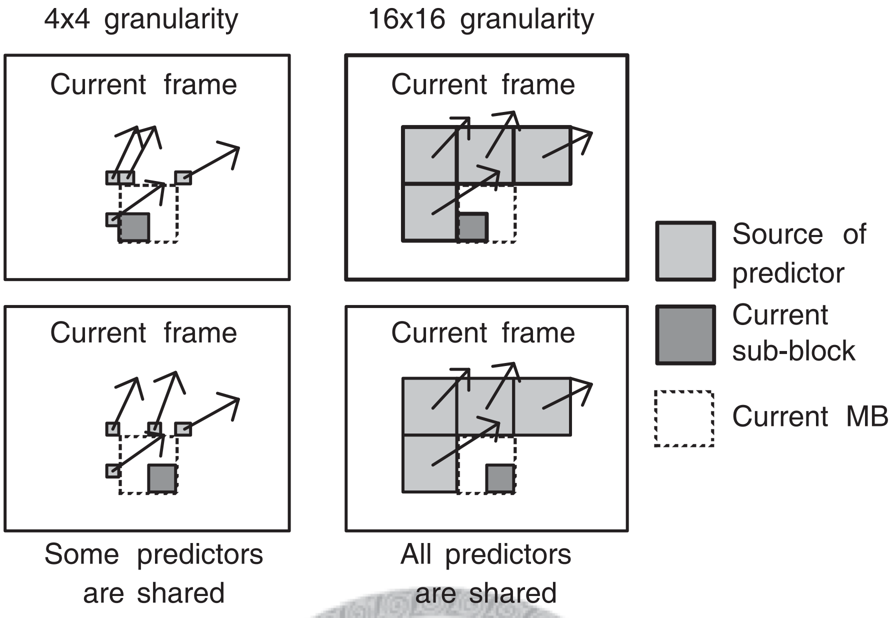


Figure 5.7: Granularity of inter predictors.

intra predictors can be skipped is the issue in this scheme.

The proposed refinement center prediction algorithm uses zero-vector, the MVP of the 16×16 block defined in the coding standard, and three inter-MB predictors from the MV field in 16×16 granularity. No intra-MB predictor is used, and all the sub-blocks share the same MVPs. This algorithm takes advantage of reusing inter-MB predictors and skipping intra-MB predictors, so the data dependency and access regularity are less harsh. Therefore, it is more applicable on a wider range of computing platforms.

### Issues of On-Chip Memory Access

Hardware-oriented full search pattern generally choose zero-vector as the refinement center for regularity and simplicity [75]. A smarter method is to choose the motion vector predictor (MVP) defined in the coding standard as the refinement center because MVP should be closer to the optimal motion vector than zero-vector. A further optimization could include several motion vectors to be the hints of refinement center, and they are mostly from highly correlated sources like the spatial and temporal neighboring macro-blocks (MBs) or blocks of other sizes if variable block-size is supported. After that, the hints of refinement center are evaluated by block-matching

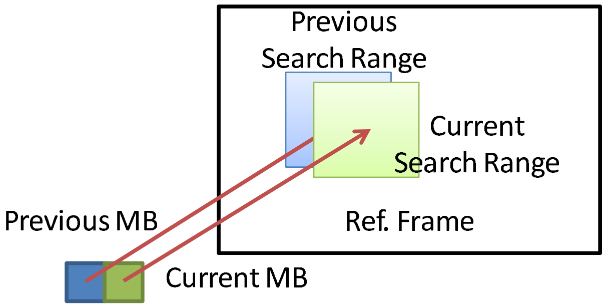


Figure 5.8: Overlapped search range of adjacent macro-blocks

algorithms, and then the one with lowest Lagrange cost is chosen as the refinement center.

If the refinement center is not fixed, the memory access pattern might be unpredictable and discontinuous, thus harms the efficiency of memory system [79, 80]. In Fig. 5.8, a typical situation of irregular access pattern with non-fixed center of refinement is shown. In this figure, the overlapped region of search ranges belonging to adjacent macro blocks may vary. If we want to save on-chip memory capacity, frequent reloading could cause irregular external memory access; otherwise, if we want to avoid redundant reloading, it is also hard to decide which part to keep on-chip. This issue will be address in the following sections.

### 5.4.3 Variable-Block-Size Data Reuse

In a software implementation of IME, it is easy to adopt algorithms with lots of data dependencies without huge overhead. This property is derived from the sequential nature of processors and sophisticated general-purposed cache system.

For a single MB, a typical software-based ME algorithm is:

1. For all the variable block-size (VBS) partitions, do the following steps.

2. Take zero-vector, MVP defined in H.264/AVC, and the final MVs from spatial and temporal neighboring MBs that have already done ME as hints of refinement center. If the current block partition type is sub 16-by-16, then MVs from other sub-blocks within the current MB can also be used.

3. Evaluate the Lagrange cost of all the hints; choose the one with lowest cost as the refinement center.

4. Do a certain type of fast search around the refinement center. Find the location with minimum cost within refinement range.

5. Refine the final result by doing the fractional motion estimation (FME) around the minimum point.

In the given algorithm, the hints of a sub-block depend on the IME refinement results of other sub-blocks. This data dependency prevents further data reuse and makes memory access pattern more unpredictable. As a result, we propose a memory-friendly ME algorithm.

The duration from the time of knowing where to access to the time of actually needing the data is the allowable latency for the memory system. The longer this duration is, the more robust it is to the memory access delay. In order to construct a memory-friendly algorithm, we need to know where to access as early as possible. As for the data flow, we break the dependency between sub-blocks within the same MB, so the access pattern is more regular. The refinement center merely depends on the result of the 16-by-16 MB, and all the sub-blocks just reuse the partial sum of absolute difference (SAD) values. The access pattern would be almost identical to that of the software-based ME without enabling variable block sizes (VBS), but the RD performance would not be harmed as much, which can be verified in the simulation results.

### 5.4.4 Cache as Search Range Buffer

When doing ME using the predictor-centered algorithm, the refinement range can be much smaller. As a result, the data reuse schemes created for full-search [75] is no longer suitable because the utilization rate of the buffer is lowered. If only the ”hot zone” of the reference buffer is kept on chip while the rest of the data is loaded from external memory when needed, the capacity requirement of on-chip memory can be greatly decreased. In addition, if the cache controller is well-designed, uncessary external memory bandwidth can also be saved. In Fig. 5.9, we replace the level-C

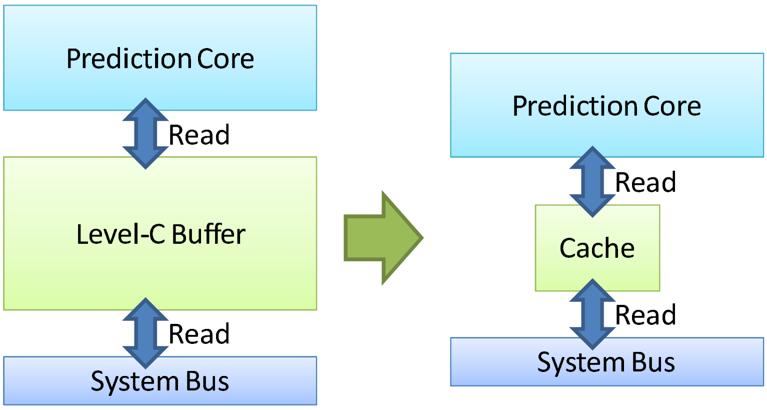


Figure 5.9: Replacing reference frame buffer in level-C reuse scheme with cache

Table 5.1: Parameter Setting of The Encoder

| | |
|---|---|
| GOP structure | IBPBP |
| Total frames | 33 |
| Quantization parameter (Qp) | 25 to 35 |
| Entropy coder | CABAC |
| ME refinement algorithm | Full search |

reference frame buffer with a cache system, so that the advantage described above can be applied to our design. The detailed architecture design of the cache system is introduced in the next chapter.

## 5.5 Simulation Result

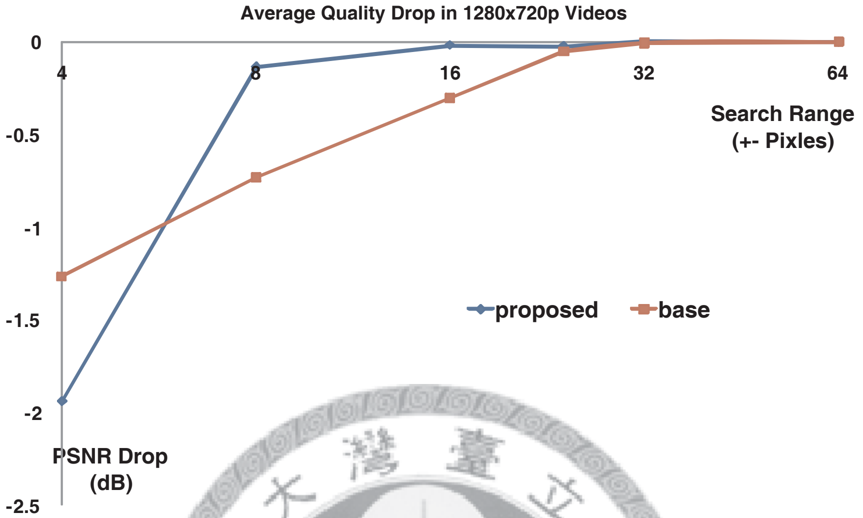
In order to only focus on the effect of changing resolutions and eliminate the bias caused by intrinsic characteristics difference of sequences, we use the same set of video sequences and resize them to a series of smaller sizes. All the testing sequences are originally in 1920×1080p at 25fps. When comparing the R-D performance, we interpolate and use the PSNR difference at the same bitrate as the metric. The quality evaluation is based on the average of all these clips. The detailed encoding parameters are shown in Table 5.1.

The proposed prediction algorithm is compared with the widely used base algorithm, which uses zero-vector, the MVP defined in the coding standard, and the three MVs from which the MVP is derived. This is applied to all the block sizes, so they contain both inter-MB and intra-MB predictors. Multiple H.264/AVC related reference software including JM, JSVM, and JMVM use this base algorithm in their fast ME algorithms.

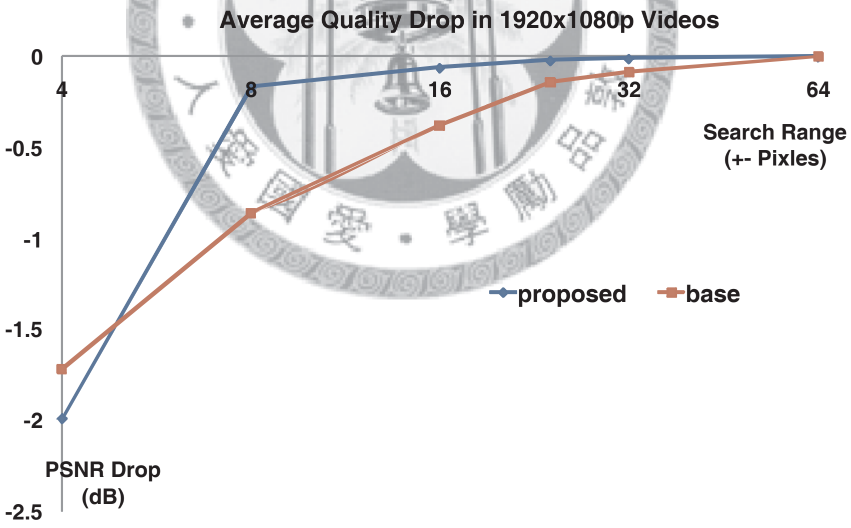
### 5.5.1 Analysis of Refinement Range

The evaluation methodology here is to show the degradation versus various search range settings. When the search range in the refinement algorithm is reduced, the quality degrades. The less the quality degrades, the better the algorithm is. The setting with the largest search range is used as the pivot, so the PSNR drop is defined as zero at that point. In Fig. 5.10, the proposed algorithm outperforms the base algorithm in moderate search ranges. In 1920×1080p resolution, the proposed method is 0.7dB better on average at ±8-pixel search range, and the quality of the proposed method with ±8-pixel search range is comparable to the base method with ±24-pixel search range, which covers nine times the search area. The quality converges when the search range is high enough because the center of refinement doesn't matter that much. When the search range is set to a tiny value, the base algorithm performs better because the MV propagates faster. As the video resolution gets smaller, the quality difference between the two algorithms decreases.

In order to get a better understanding of the phenomenon, the quality evaluation of a particularly dynamic scene in video sequence ”Tractor” is shown in Fig. 5.11, and its block type distribution is shown in Fig. 5.12. From Fig. 5.11, we can observe that as a video clip with larger motion, the quality degradation is more sensitive to the search range. The smaller gray-scale images in Fig. 5.12 display the block type, where each pixel denotes an MB in the frame. In Fig. 5.12 (b), the search range is larger in order to illustrate how well it can be with a larger search range. The intra blocks only locate in frame border, upper-left corner, and object boundaries, and each of them represents an individual phenomenon. The intra blocks in frame borders are inevitable because the theoretically matched block does not exist in the reference frame. In the upper-left corner, the predictors begin at zero-vector. If the optimal



(a)



(b)

Figure 5.10: Quality of various refinement ranges in (a) 1280×720p and (b) 1920×1080p videos.

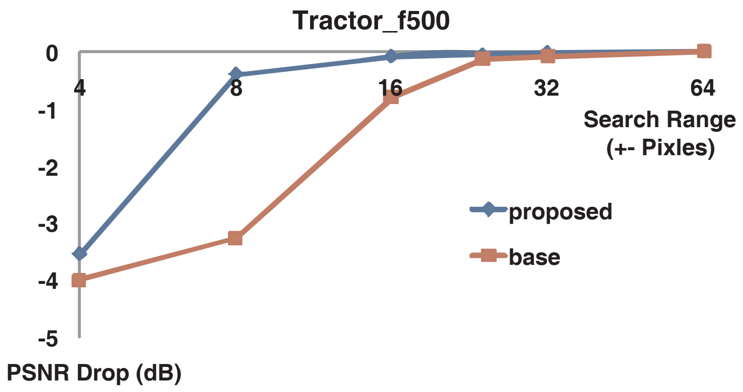


Figure 5.11: Quality evaluation of sequence “Tractor” beginning at frame 502.

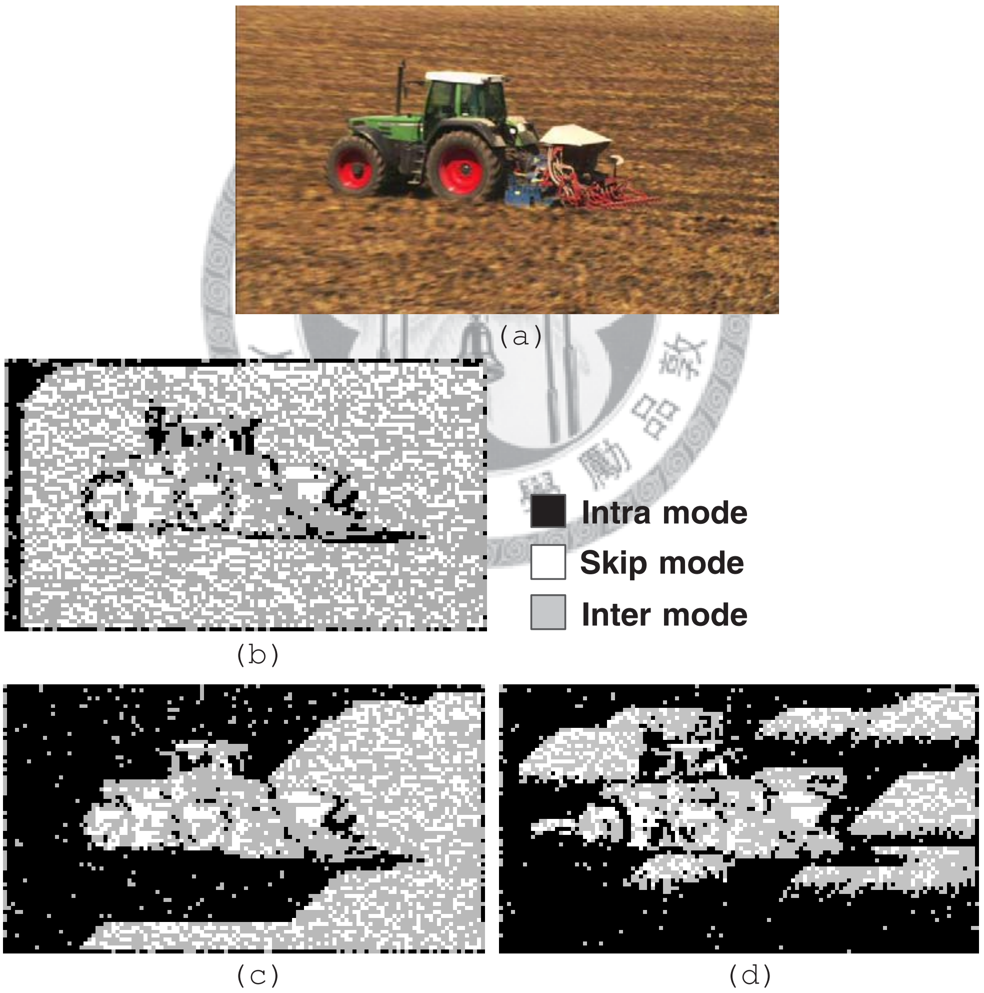


Figure 5.12: Block type distribution at Tractor frame 502. (a) Subjective view of the frame. (b) Proposed algorithm with search range ±16. (c) Base algorithm with search range ±16. (d) Proposed algorithm with search range ±4.

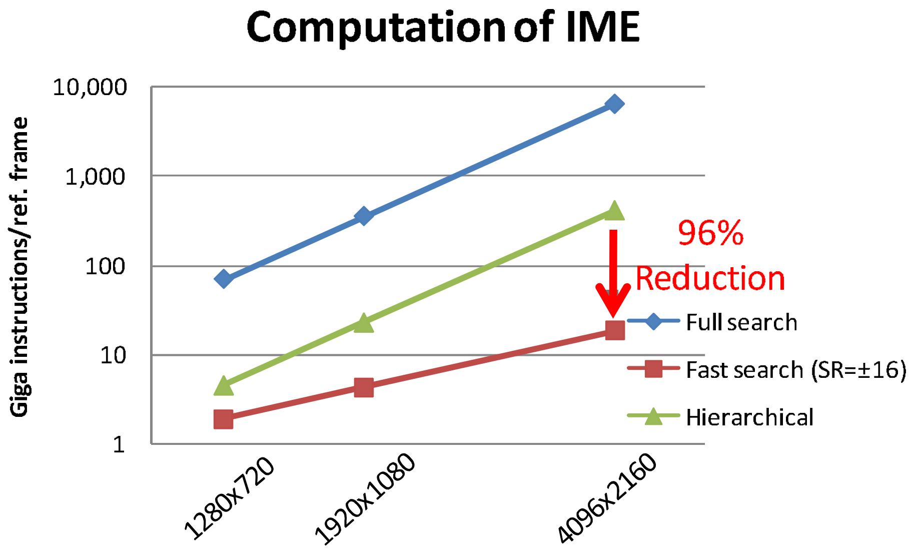


Figure 5.13: Computation reduction of predictor-centered fast IME algorithm

MV is longer than the search range, intra mode would win the mode decision before the MV locks in the optimal motion. However, the best matched MV can still keep propagating to the neighbor MBs as predictors as said in Section 5.4.1. Boundaries between objects of different motion are places where predictors are less helpful. In Fig. 5.12 (c), the upper-left intra corner grows enormously because in the base algorithm, predictors are unavailable when the neighboring block is encoded in intra mode. In Fig. 5.12 (d), the search range is set to a tiny value, and the locked-in MVs get lost after a few rows because the variation of MV distribution is larger than the search range. So even if the prediction algorithm is accurate, adequate search range is still essential. From Fig. 5.12, we can conclude that using proposed algorithm with moderate search range, optionally combined with larger search range around the upper-left corner is the most efficient algorithm. The doubt about granularity of predicting MV field and the effectiveness of combining inter-MB and intra-MB predictors are also resolved. That is, as long as the inter-MB predictors are accurate enough, coarse-grained MV field suffice, and intra-MB predictors are not essential.

### 5.5.2 Reduction of Computational Complexity

With proposed fast IME algorithm and VBS data reuse, the IME computation is largely reduced. Fig. 5.13 is a log-log plot of computation per reference frame vs.

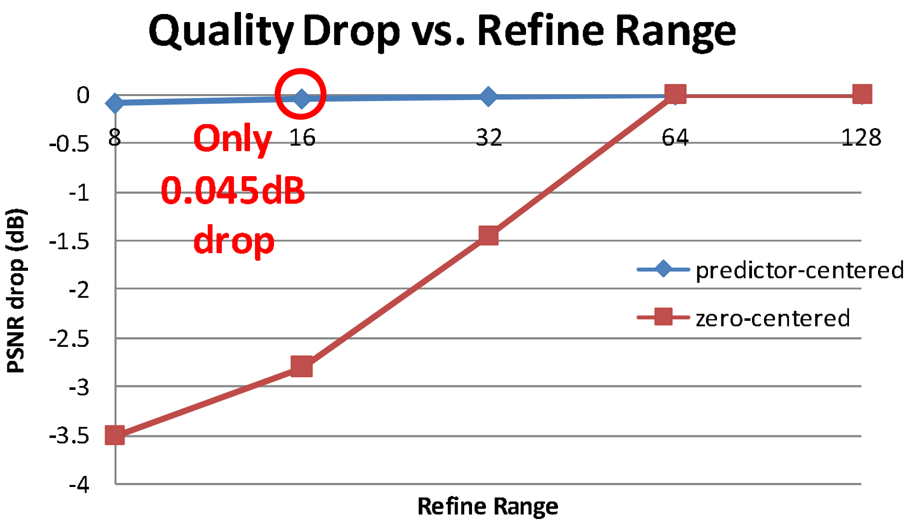


Figure 5.14: Rate-distortion quality drop of predictor-centered fast IME algorithm in worst case

frame resolution. Compared with hierarchical search, the computation reduced by 96% in $4k \times 2k$ resolution. Computation reduction generally comes with a price, rate-distortion quality, which can be objectively measured as PSNR value. If the refinement center is not chosen well, the quality drops quickly with the reduction of refinement range because the optimal motion vector falls outside the refinement range. From Fig. 5.14, the quality does drop quickly when the refinement center is fixed at zero vector. However, with the proposed algorithm, the PSNR drop is only 0.045 dB in the worst case when the refinement range is ±16 pixels. In our experiments, the average PSNR drop is only 0.013 dB.

## 5.6 Summary

Designing a prediction core that supports $4k \times 2k$ video resolution with real-time performance is challenging. The challenge comes from four aspects: huge computation, high bandwidth, and large on-chip memory size. The computation of IME exceeds the computing capability of a high-end quad-core CPU by $10^4$ times, the external memory bandwidth saturates a high performance DDR2-800 throughput, and the on-chip memory costs millions of equivalent gate count.

In this chapter, an efficient predictor-centered algorithm for the prediction core is presented. In order to increase the robustness and accuracy, the proposed algorithm preserves more motion information as predictors. Moreover, it uses the same coarse-grained inter-MB predictors for all the sub-blocks inside an MB, resulting in loose data dependency, better SAD computation reuse, and regular access pattern on reference frames. When followed by a refinement algorithm with different search ranges, the quality drop is much less than the base algorithm used in numerous reference software when the search range shrinks. In 1080p video resolution, the proposed method outperforms the base method by 0.7dB when the search range is ±8-pixel, and the quality drop of the proposed method with ±8-pixel search range is similar to that of the base method with ±24-pixel search range, so the computation and bandwidth are greatly saved. This refining-center decision algorithm is orthogonal to the following refinement algorithms, therefore further candidate-reducing techniques can be utilized as well.

國立臺灣大學 · 愛國 · 愛學 · 勵學 · 敦品

國立臺灣大學・敦品勵學・愛國愛人

# Chapter 6

# Analysis and Architecture Design of Cache-Based Prediction Core for 3D/Quad HDTV Videos

In memory hierarchy of computer architecture, cache system refers to the memory layer closer to the computing device, which can bridge the fast computing device and the slow memory down in the hierarchy. It compensates the long latency and low throughput of the memory system, hence plays an important role in performance [81]. For video encoders, the heaviest burden of memory traffic comes from the access to the reference frames when doing motion estimation. Traditionally, in hardware design of video encoder, a portion of data in reference frames is kept in scratch pad memory, generally implemented as on-chip SRAM, so that all the accesses to the reference frames can be redirected to the SRAM, resulting in smaller latency and lower off-chip memory bandwidth. However, with the growth of video resolution, the search range of motion estimation grows proportionally, so does the size of on-chip SRAM. Given the situation that there is only a small portion in the search range is highly likely to be accessed, keeping all the pixels falling within the search range on chip is not a sane design. Instead, using a cache as the search range buffer can reduce the required capacity at expense of cache miss penalty. If the degraded performance due to the cache miss penalty is tolerable, then replacing the all-inclusive search range buffer with a cache system is desirable.

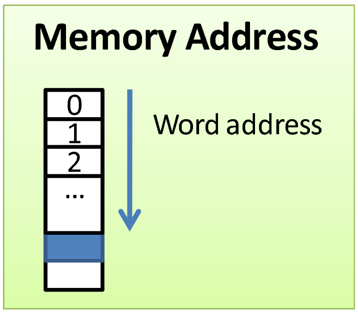


Figure 6.1: Generic memory address

In this chapter, a cache system dedicated for quad full high definition video encoder is proposed. In the first section, the concept of generic cache system is introduced, followed by a cache organization for 2D structure. After that, the design challenge, proposed prefetching algorithm and the hardware architecture are described in detail.

## 6.1 Introduction to Generic Cache System

In computer systems, the input and output of memory-mapped devices can be indexed and projected to a memory map, and a memory word can be uniquely addressed by a word address as shown in Fig. 6.1.

A memory word can be stored in a cache, and that particular word can be addressed in a slightly different addressing scheme, illustrated in Fig. 6.2. In a general set-associative cache, we need a line address to specify which line the word lies in. After tag matching, a set index can be obtained to indicate which set the data resides, or in the other case, the data doesn't exist on cache. Generally a cache line contains multiple cache words. As a result, after getting the cache line containing the data, a word index is needed to point out the particular desired word.

Since there are two different addressing schemes for memory and cache, we need a method to translate memory address into cache address. Assuming we use a 32-bit physical address in a byte addressable architecture, the physical memory address can be translated into byte index, word index, line address, and tag as shown in Fig. 6.3. Assuming a cache word contains $2^b$ bytes, then byte index contains $b$ bits. Similarly,

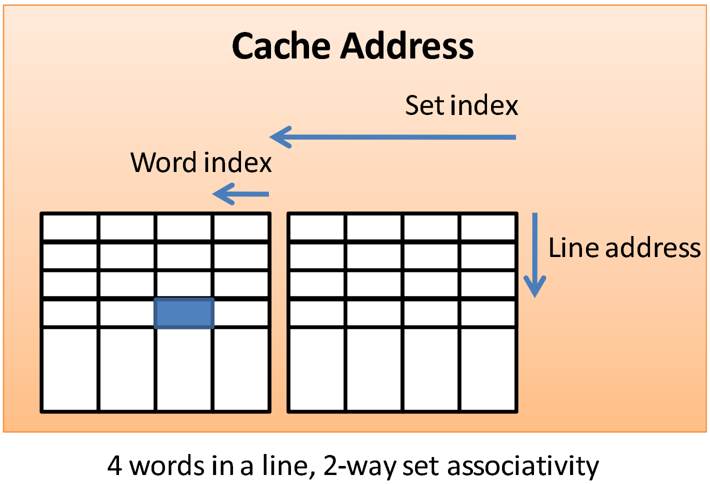


Figure 6.2: Generic cache address

if a cache line contains $2^w$ words, then the word index contains $w$ bits. When the number of lines in a set is $2^l$, the line address is a $l$-bit field. The rest part of the physical address is defined as the tag.

The address translation described above assumes the parameters byte-per-word, word-per-line, and the number of lines in a set are all power-of-two, which is the most widely used case because of its simplicity. Were it not the case, then the address translation is somewhat more complex. Assuming byte-per-word is $B$, word-per-line is $W$, and the number of lines in a set is $L$, then the translation formula is as follows:

$$byte\_index = address\%B \tag{6.1}$$

$$word\_index = (address/B)\%W \tag{6.2}$$

$$line\_address = ((address/B)/W)\%L \tag{6.3}$$

$$tag = ((address/B)/W)/L \tag{6.4}$$

The region of memory-map sharing the same tag is defined as a tag modulus as shown in Fig. 6.4. From the formula above, a tag modulus contains $W \times L$ cache words.

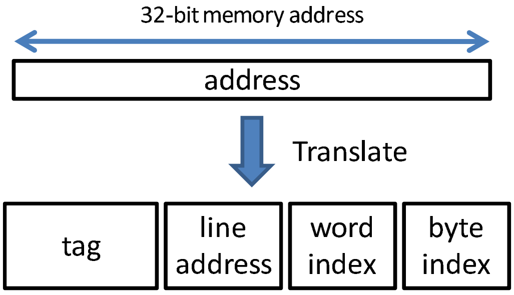


Figure 6.3: Address translation of generic cache system

國立臺灣大學 人學國榮學勵品敦

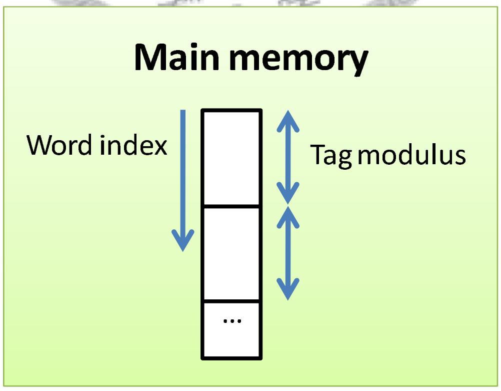


Figure 6.4: Tag modulus of generic cache system

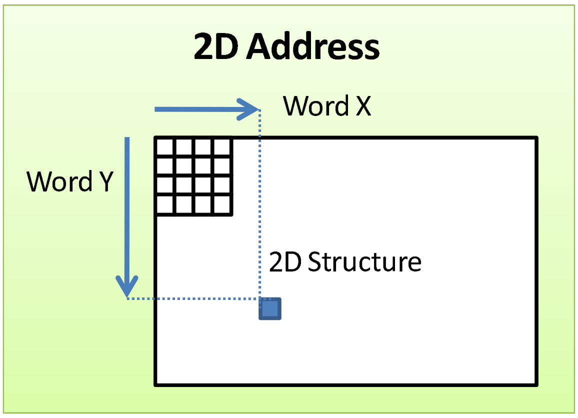


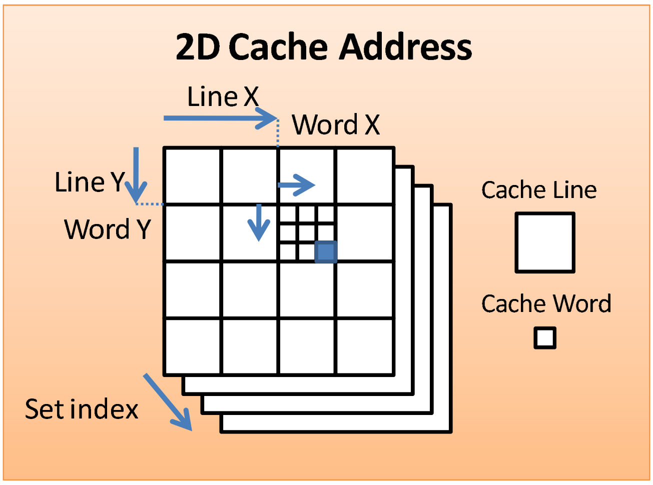
Figure 6.5: 2D memory address

### 6.1.1 Two Dimensional Cache Organization

One of the advantages of cache system is the exploitation of data locality. If a word is accessed, then the whole line containing the word is brought to the cache. Once the following accesses request the other words in the same line, then it is a cache hit. Improving data locality is one of the programming techniques to increase cache hit rate and decrease conflict misses. The basic idea of spatial cache locality optimization is to use a data structure that put data with high correlation of access pattern together, so that the previous accesses benefit the following ones.

In software design of video encoding, the reference frames can be packed in a block-based data structure, so the inherent two dimensional characteristic of reference frames can be exploited in the pixel level. In hardware design of video encoding, this property can be further utilized by exposing it to the architecture. As we are designing a dedicated cache system for reference frame, a 2-D structure, it can be beneficial to keep the 2-D nature inside the cache system.

In contrast to the general memory map shown in Fig. 6.1, a block, i.e. a 2-D word, can now be addressed by a pair of word addresses, word X and word Y, in a 2-D structure shown in in Fig. 6.5. Similar to the generic memory system, when a 2-D memory word is stored in a 2-D cache, a different address scheme is used. Cache address scheme in Fig. 6.2 is modified to Fig. 6.6, which uses a pair of line addresses and word indexes instead of one. As a result, the cache word, cache line, and tag modulus also become two dimensional. For example, the cache line contains $3 \times 3$



3x3 words in a line, 4-way set associativity

Figure 6.6: 2D cache address

國立臺灣大學

cache words in the case of Fig. 6.6.

The two different addressing schemes between 2-D memory and 2-D cache can be translated as shown in Fig. 6.7. Assuming the X and Y coordinate of the 2-D memory are pixel-addressed, they can be translated into pixel index pair, word index pair, line address pair, and the tag. In the most general case, we assume the pixel-per-word is $P_x \times P_y$, word-per-line is $W_x \times W_y$, and the number of lines in a set is $L_x \times L_y$, then the translation formula is as follows:

$$
\begin{aligned}
pixel\_index_x &= X\%B_x && (6.5)\\
pixel\_index_y &= Y\%B_y && (6.6)\\
word\_index_x &= (X/B_x)\%W_x && (6.7)\\
word\_index_y &= (Y/B_y)\%W_y && (6.8)\\
line\_address_x &= ((X/B_x)/W_x)\%L_x && (6.9)\\
line\_address_y &= ((Y/B_y)/W_y)\%L_y && (6.10)\\
tag_x &= ((X/B_x)/W_x)/L_x && (6.11)\\
tag_y &= ((Y/B_y)/W_y)/L_y && (6.12)\\
tag &= \{tag_x, tag_y\} && (6.13)
\end{aligned}
$$

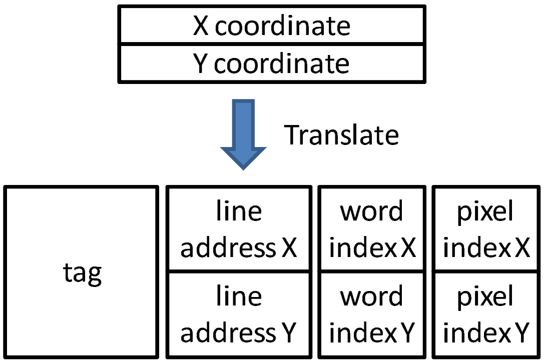


Figure 6.7: Address translation of 2D cache system

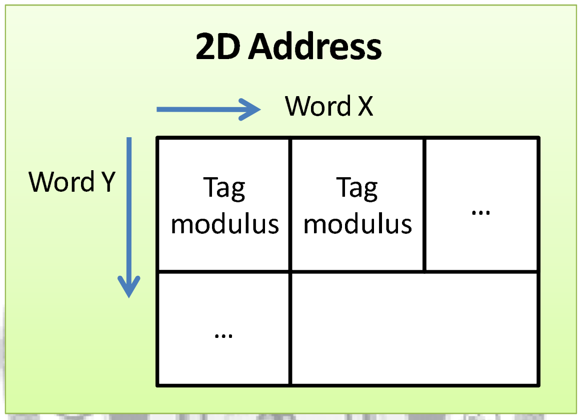


Figure 6.8: Tag modulus of 2D cache system

Since there are only one tag needed, $tag_x$ and $tag_y$ can be concatenated bitwisely to form the tag. When the cache parameters pixel-per-word, word-per-line, and the number of lines in a set happens to be power-of-two, then the division and modulus operations in the formula above can be simplified as bit-field separation. The definition of tag modulus in 2-D cache remains the same, the region of memory sharing the same tag. In this case, the region is rectangular like in Fig. 6.8.

## 6.2 Design Challenges

The criterion to allow replacing an all-inclusive search range buffer by a cache system is that the degraded performance caused by cache miss penalty is tolerable. As a result, how to design a cache system with miss rate and miss penalty low enough is

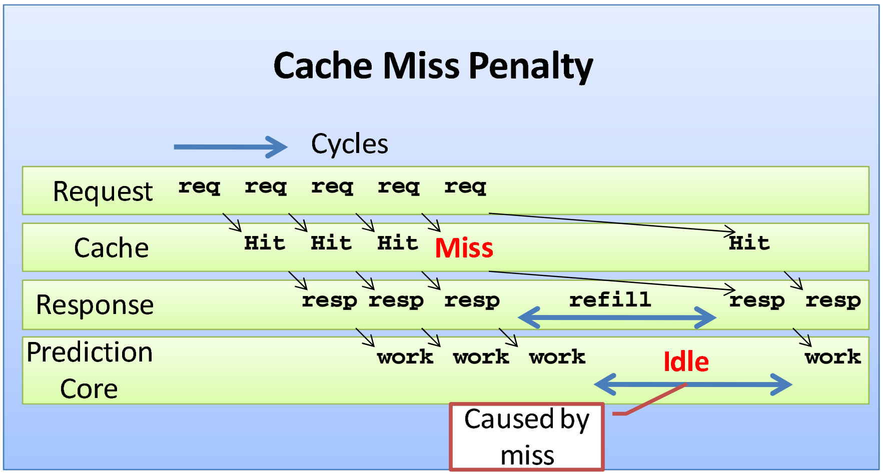


Figure 6.9: Cache miss penalty

the biggest design challenge. In the following sections, cache miss penalty and data prefetching overhead are covered, and the performance evaluation of the prediction core is shown.

### 6.2.1 Cache Miss

When the computing element sends a data request to the cache, but the cache doesn't have it in the internal memory, then it is called a cache miss. The probability of cache miss is called cache miss rate, and the idling cycles of the computing element caused by cache miss is called miss penalty. The average access time ($T_{access}$) can be expressed in the following formula, where $T_{hit}$ stands for the required time for cache hit, $p_{miss}$ for cache miss rate, and $penalty_{miss}$ for miss penalty.

$$T_{access} = T_{hit} + p_{miss} \times penalty_{miss}$$

In order to improve the cache performance, we need to reduce the miss rate and miss penalty. In Fig. 6.9, we can see that the first three requests from the prediction core get cache hits, and the requested data return to the prediction core smoothly, resulting in full throughput. However, the fourth request gets a cache miss, and the cache system needs to load the missing cache line from external memory and refill it to the internal RAM. As a result, the request returns after a period of time, and during that time, the prediction core is starving for data, resulting in wasted idling cycles.

### 6.2.2 Overhead of Data Prefetching

Data prefetching is to get the required data ahead of the actual data request. This is a well-known technique to decrease miss rate, especially compulsory miss rate, which refers to the miss rate cause by the first time of access.

In the literature of computer architecture, cache prefetching algorithm can be categorized into hardware prefetching and software prefetching [82]. In a simple one-block-lookahead technique, when the processor asks for line $N$, the cache can prefetch line $N+1$. Some works have extended this to strided pattern [83, 84]. This is done in hardware, hence the name. This method is suitable for instructions or very regular data structure like those used in scientific computing because the access pattern is more predictable. However, when the access pattern is not that regular, hardware prefetching is of less use. In this case, programmers can use some special instructions to tell the cache where to do the data prefetching.

Both hardware and software cache prefetching come with some overhead. First, the cache system generates more traffic, and memory contention leads to longer latency. If the prefetched data is not utilized later, the increased external traffic is totally wasted. Second, the prefetched data should be stored somewhere inside the cache. If it is stored in an additional hardware, ”stream buffer”, then there are overhead in hardware area. If it is directly stored in the internal RAM, which is more suitable for software prefetching, then a cache line must be evicted to spare the space. Once the evicted data is more useful then the prefetched data, then it is called cache pollution. Third, if the processor or the cache system cannot handle data request and the prefetching instruction at the same time, a cycle allocated to do data prefetching is a cycle that cannot be used to handle data request. This can decrease cache throughput, and in memory-bounded situations, this directly affects the performance.

As for the hardware design of the prediction core, the access pattern of motion estimation is not exactly regular, so a method similar to software prefetching is used in our hardware design. The overhead of prefetch instructions in software prefetcing is translated to area cost in the hardware design. In such a hardware system, the cache throughput is almost always kept at the peak rate. Consequently, if prefetching occupies a cycle that could have been allocated to data request, the performance of

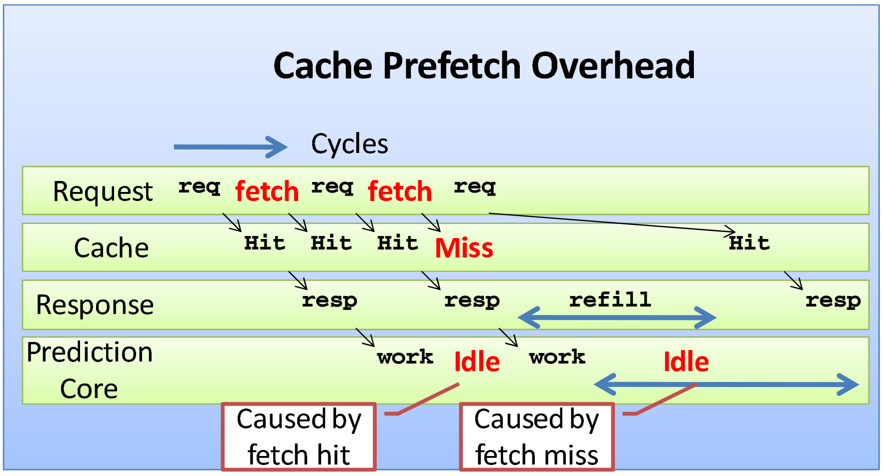


Figure 6.10: Overhead of cache prefetching

the prediction core can be affected, like memory-bounded programs on a processor. For example, in Fig. 6.10, the cycle between the first and second request is used to do data prefetching, as a result, there is an idling cycle in the prediction core. Since the prefetching algorithm is similar to software prefetching, the prefetched data is directly stored in the internal RAM. Therefore, cache pollution can happen and should be avoided.

## 6.3 Access Pattern and Throughput Requirement

### 6.3.1 Access Pattern of IME

The shifting reference buffer of IME is shown in Fig. 6.11. A candidates macroblock is of size $16 \times 16$. Combining $4 \times 4$ such consecutive candidates forms a buffer of size $19 \times 19$. After two-by-two down-sampling, the reference buffer becomes $10 \times 10$, and this is the size used in the proposed design. The reference buffer can be shifted in three directions in order to support snake scan order, which is illustrated in Fig. 6.12. In this figure, assuming the search range is of size $14 \times 14$, it can be scanned in 9 cycles by using snake scan order shown in the upper right part. The initial 5 cycles are used to fill the reference buffer, so that the upper-left corner can be evaluated. In the following 8 cycles, the access pattern are shown in the figure, where the blank rectangle represents the required new data, and the solid square is

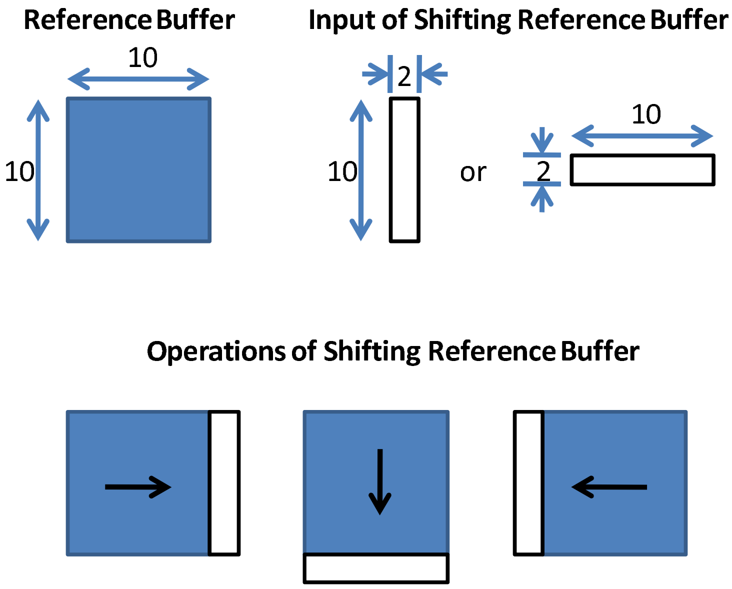

Reference Buffer
10
10
Input of Shifting Reference Buffer
2
10
or
10
2
Operations of Shifting Reference Buffer

Figure 6.11: Shifting reference buffer in IME

for the reference buffer.

From this access pattern, the cache system needs to provide a $10 \times 2$ or a $2 \times 10$ rectangular block of pixels per cycle for the IME engine. Consequently, the proposed cache system uses a $2 \times 2$ block as the cache word, and 5 words per cycles is the required throughput. It should be noted that the access pattern must be word-aligned, i.e. the X and Y coordinate must be even numbers in our case.

### 6.3.2 Access Pattern of FME

It would be more efficient if the required throughput of FME engine can match that of IME engine. Therefore, we determine the cache throughput according to the requirement of IME, and try to match FME with the existing cache throughput. The basic reference buffer for our proposed FME is a $10 \times 10$ buffer, and depending on the even-odd property of the X and Y coordinates, four cases of access pattern can be inducted as shown in Fig. 6.13. Given the throughput and access pattern determined, the reference buffer of FME can be filled in the patterns shown in Fig. 6.14. For the odd-odd case, the eighth request can be omitted without greatly affecting the quality of FME because it only contributes to a single pixel. This missing pixel can be compensated by the median of its neighboring three pixels.

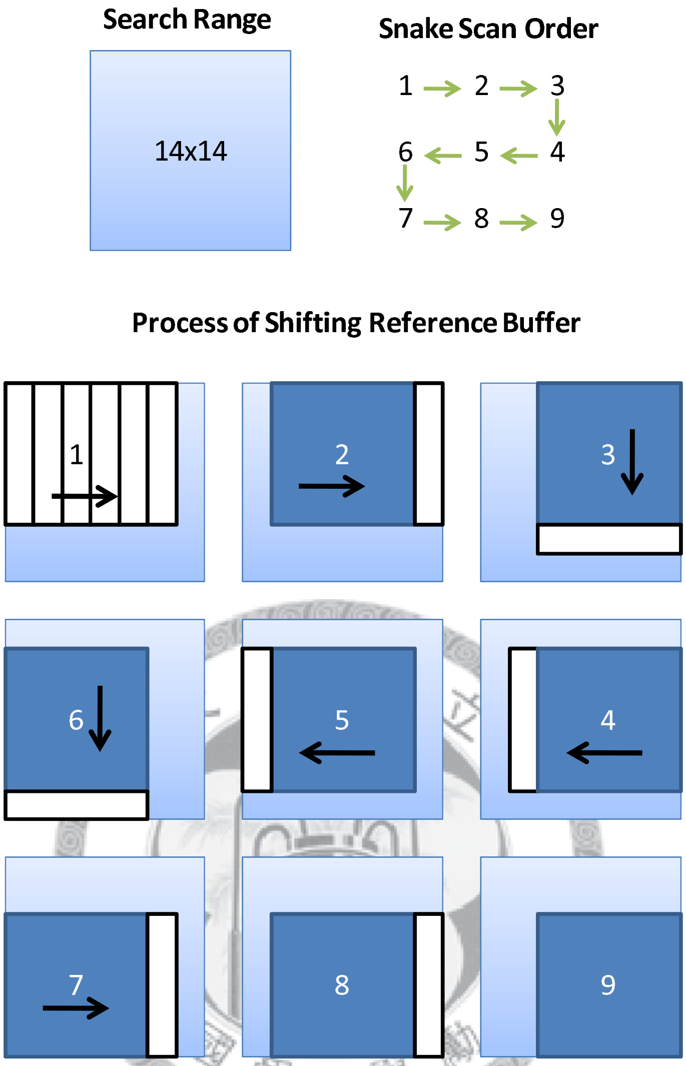


Figure 6.12: Snake scan of reference buffer in IME

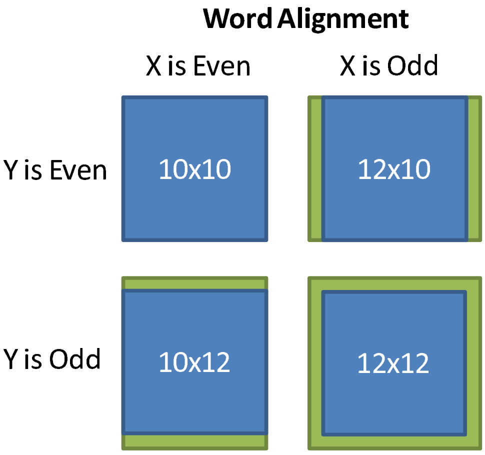


Figure 6.13: Cache word alignment of reference buffer in FME

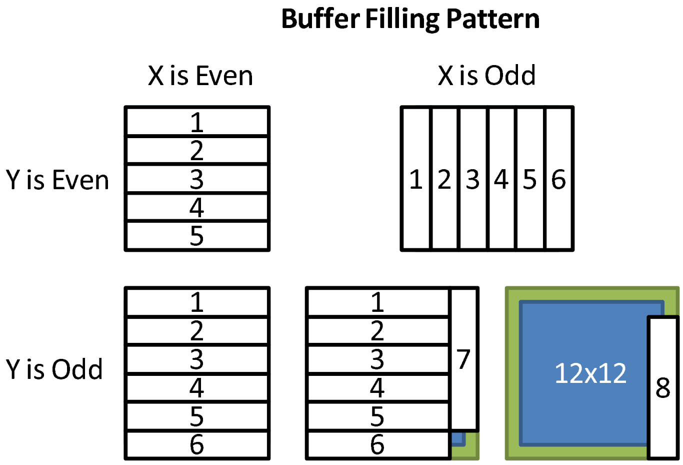


Figure 6.14: Filling pattern of reference buffer in FME

## 6.4 Proposed Prefetching Algorithm

### 6.4.1 Rapid Prefetching Pattern

In the proposed fast IME algorithm described in Chapter 5, there are two kinds of reading patterns. One is the hints of the refinement center, and the other is the refinement range. In Fig. 6.15, the hint and refinement range are shown in solid blocks, and these access patterns needs different sizes of cache lines. The mission of data prefetching is to bring all these needed cache lines to the cache before they are actually read, so the access pattern of data prefetching needs to cover all the needed cache lines. In the figure, refinement range covers $4 \times 4$ cache lines, so 8 prefetching requests are needed to cover the $4 \times 4$ cache lines. Similarly, a hint covers $2 \times 2$ cache lines, so only 2 prefetching requests are needed to cover the $2 \times 2$ cache lines. Given that the proposed fast IME algorithm uses 6 hints and 1 refinement range for a reference frame, there are totally $(6 \times 2 + 8) \times 2 = 40$ prefetching requests when processing B-frames. Compared with $(6 \times 5 + 96) \times 2 = 252$ reading requests, 84% of requests are reduced.

As for the access pattern of proposed FME architecture, a $8 \times 8$ block needs reference frame pixels that covers from 2 to 3 cache lines shown in Fig. 6.16. As a result, from 2 to 5 prefetching requests are needed to cover these cache lines. Assuming the distribution of the FME support region is uniform, the expected number

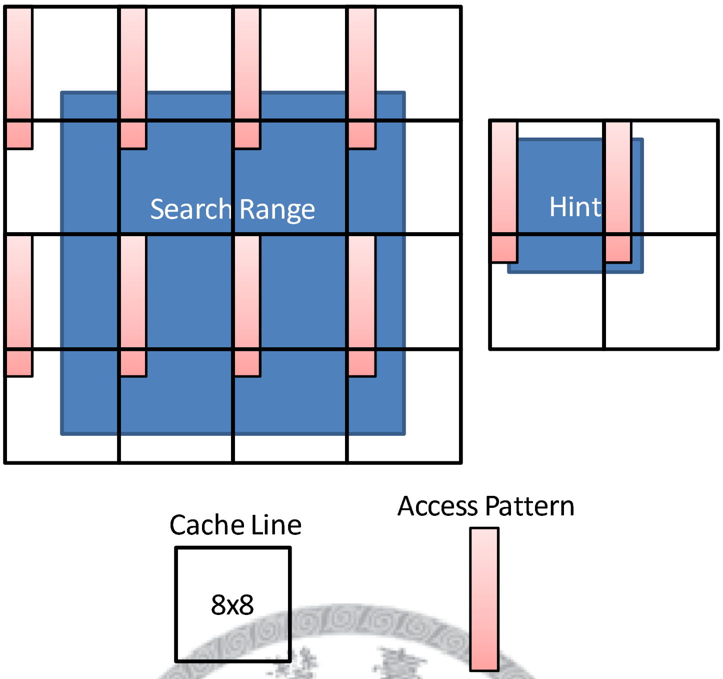


Figure 6.15: Access pattern of IME cache prefetching

of prefetching requests is 2.26 for a $8 \times 8$ block. When processing a B-frame, the expected number of prefetching requests for a MB is $2.26 \times 4 \times 4 \times 3 = 108.5$, which is only 36% of reading requests, i.e. 64% reduction is achieved. However, even with this rapid prefetching pattern, there are not enough cycles to do prefetching after the normal processing. Assuming IME data prefetching is done after FME, then $300 + 40 = 340$ reading and prefetching requests are needed, leaving $360 - 340 = 20$ cycles for cache refilling, which is far from enough. Meanwhile, FME data prefetching is done after IME, so $252 + 108.5 = 360.5$ reading and prefetching requests are needed, which even exceeds the number of cycles in a MB pipeline stage. To sum up, techniques other than rapid prefetching pattern should be adopted in order to squeeze all the operations within a MB pipeline stage.

### 6.4.2 Concurrent Reading and Prefetching

Following the analysis in 6.4.1, concurrent reading and prefetching is proposed. When data prefetching can be processed in the same cycle as reading, data prefetching can be started as early as the MB pipeline stage starts. As a result, when IME

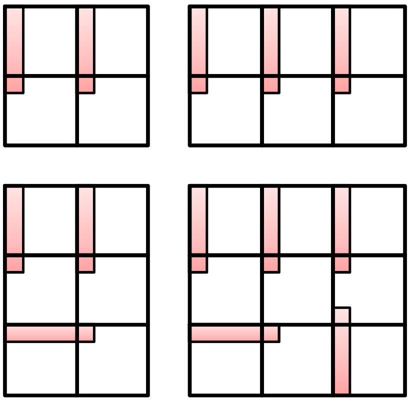


Figure 6.16: Access pattern of FME cache prefetching

data prefetching is started with FME, then $360 - 40 = 320$ cycles are left for cache refilling operations. Meanwhile, when FME data prefetching is started with IME, there are $360 - 108.5 = 251.5$ cycles for refilling.

Nevertheless, if data prefetching runs concurrently with normal data reading, the prefetched data must be stored somewhere within the cache. When the evicted cache line is required by future reading request, then cache pollution happens. Many widely-used replacement policies like pseudo-random, least-recently-used, least-frequently-used, not-most-recently-used, etc cannot prevent the cache pollution from happening. Therefore, a replacement policy that guarantees no cache pollution is helpful to reduce the miss rate of reading.

### 6.4.3 Priority-based Replacement Policy

In order to eliminate the cache miss rate caused by cache pollution of data prefetching, a priority-based replacement policy is proposed. In original data prefetching algorithm described in 6.4.2, the prefetching starts with normal reading, resulting the possibility of cache pollution. Now a locking mechanism is introduced as shown in Fig. 6.17. Before doing the data prefetching, all the data needed by the following reading within this MB pipeline stage are locked first. The data locking is done similarly to the data prefetching, the only difference is that data locking uses higher priority to protect the data from overwritten. After locking is done, all the cache lines touched by locking are labeled with priority bit. When eviction happens due to data

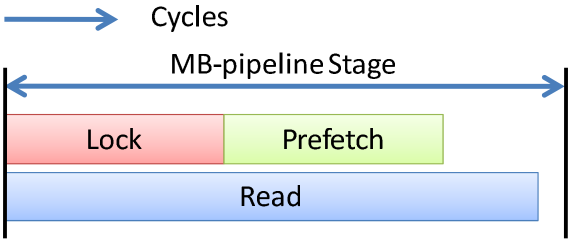


Figure 6.17: Timing diagram of cache read/lock/prefetch operations

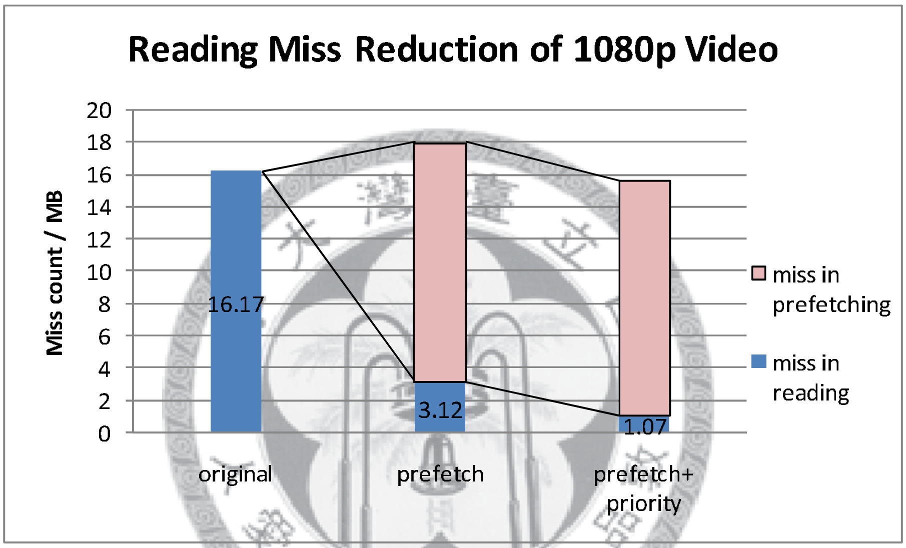


Figure 6.18: Average cache reading miss rate reduction by data prefetching and priority-based replacement policy of 1080p video sequences

prefetching, only the cache line with lower priority can be chosen. As a result, the cache lines needed by future reading requests are protected from eviction, thus cache pollution is not possible.

The experimental result of encoding video sequences with resolution of 1080p and 2160p are shown in Fig. 6.18 and Fig. 6.19. From the results, we can see that after applying data prefetching and priority-based replacement policy, the cache miss rate in data reading is reduced by 15.1 folds and 14.3 folds for these two resolutions, achieving 99.81% and 99.79% hit rate respectively.

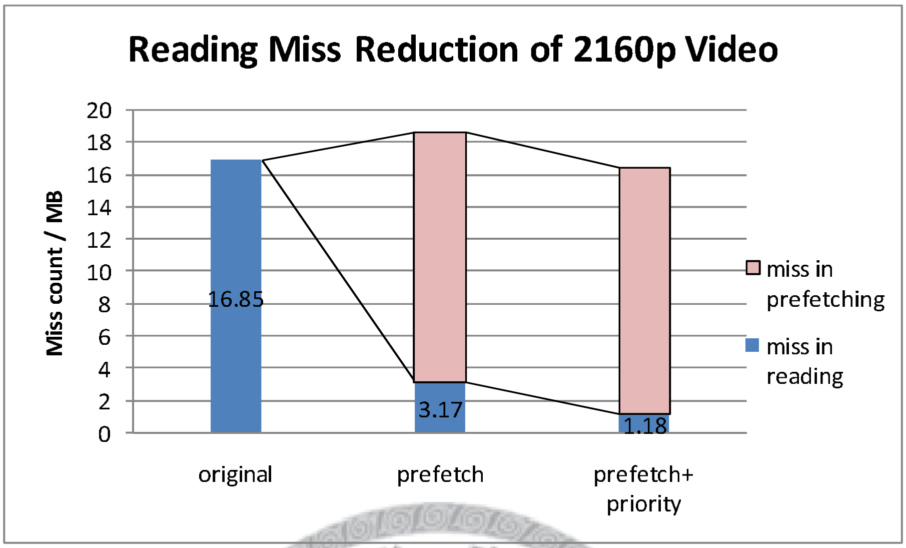
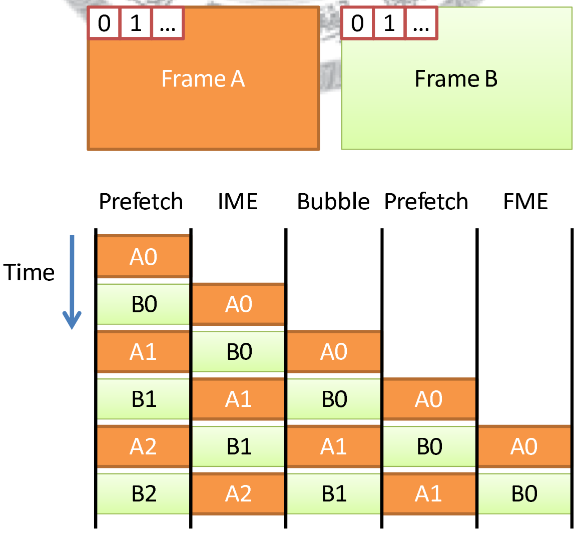


Figure 6.19: Average cache reading miss rate reduction by data prefetching and priority-based replacement policy of 2160p video sequences



| Prefetch | IME | Bubble | Prefetch | FME |
|---|---|---|---|---|
| A0 | | | | |
| B0 | A0 | | | |
| A1 | B0 | A0 | | |
| B1 | A1 | B0 | A0 | |
| A2 | B1 | A1 | B0 | A0 |
| B2 | A2 | B1 | A1 | B0 |

Figure 6.20: Interleaved MB pipeline stages in prediction core

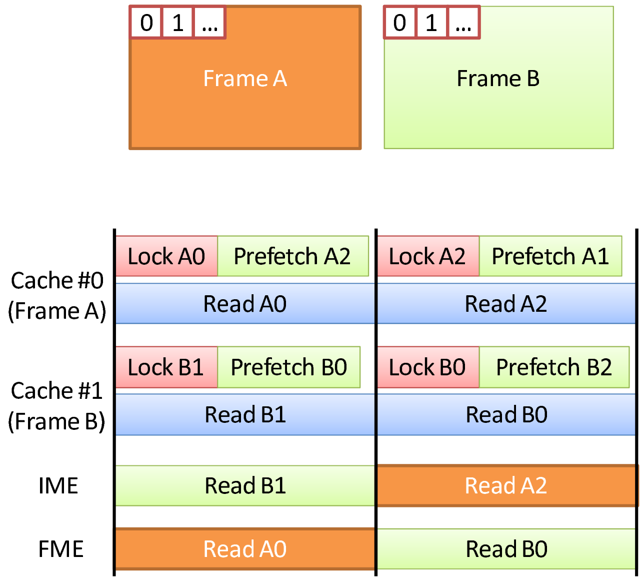


Figure 6.21: Timing diagram of cache operations with MB interleaving

國立臺灣大學

### 6.4.4 Integration with Macro-Block Pipeline

The operation introduced in Section 6.4.3 and Fig. 6.17 is what happens within a single MB pipeline stage. When integrated with the MB interleaving pipeline architecture of the whole prediction core, the timing diagram should be modified as shown in Fig. 6.20. In the 4-stage MB pipeline architecture, the prediction core contains two of these stages, IME and FME [85]. With the introduction of data prefetching, there are two more prefetching stages for IME and FME respectively. In order to make sure that IME and FME handles different current frames, a bubble stage is inserted. Fig. 6.21 shows the close-up look of the cache operations.

## 6.5 Proposed Architecture for Cache Design

The prefetching algorithm described in section 6.4 needs the underlying hardware architecture to support all the operations, and the constraints given by the prediction core and the whole encoder should be considered as well. The requirements include high data throughput, zero penalties on cache line split, non-blocking refilling, concurrent reading and prefetching, and high working frequency.

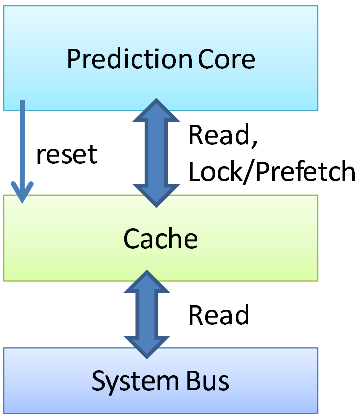


Figure 6.22: Generic system diagram of prediction core with cache

### 6.5.1 Overall Architecture

Fig. 6.22 shows a generic system diagram of prediction core with cache system as reference frame buffer. In the proposed double current frame data flow with interleaving MB pipeline architecture, the system diagram becomes what is shown in Fig. 6.23. The two major computing modules in the prediction core, IME and FME, are connected to the two cache cores with a $2 \times 2$ switching network. The locking and prefetching operations for IME and FME are done in other helping modules, and they are connected to the switching network as well. There are two routing paths of the 2-by-2 switching network, shown in Fig. 6.24. The routing path switches mode after processing an MB.

Inside the cache core, the simplified conceptual data flow can be shown in Fig. 6.25. The prediction core sends a reading request to the cache core, and then the frame-based address in the request is resolved to the internal cache-based addressing scheme. When a cache address is resolved, the bookkeeping logic checks the bookkeeping states to see whether the requested data is on chip. If yes, then it is a cache hit. On a cache hit, the bookkeeping logic sends a reading request to the data memory, and the returned data is then forwarded back to the prediction core. If the data is not on chip, then it is a cache miss. On a cache miss, the bookkeeping logic sends reading requests to the external memory bus for the missing cache line. When the missing cache line returns, the data memory is refilled, and the data can then be forwarded back to the prediction core.

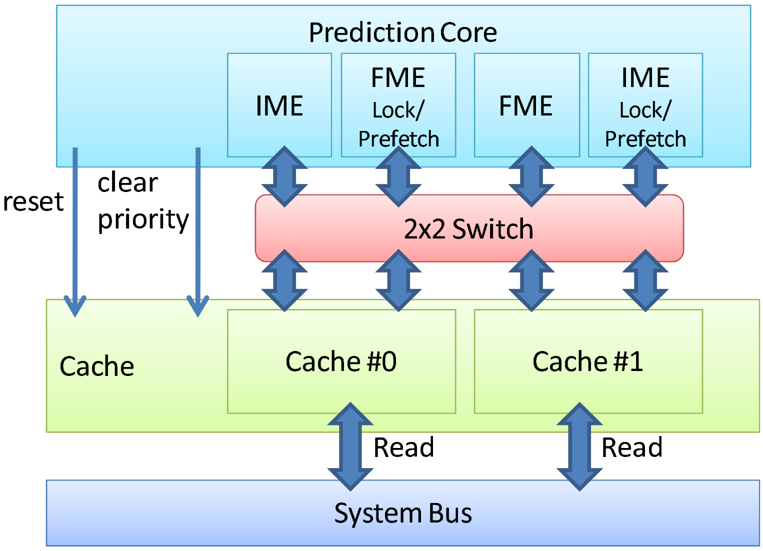


Figure 6.23: System diagram of proposed prediction core with cache

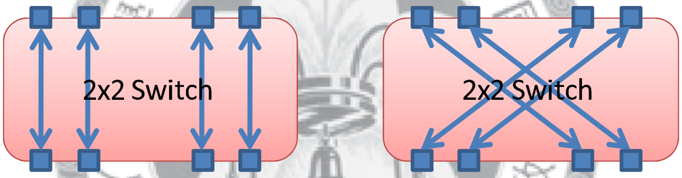


Figure 6.24: Connecting configurations of 2-by-2 switching network for MB interleaving

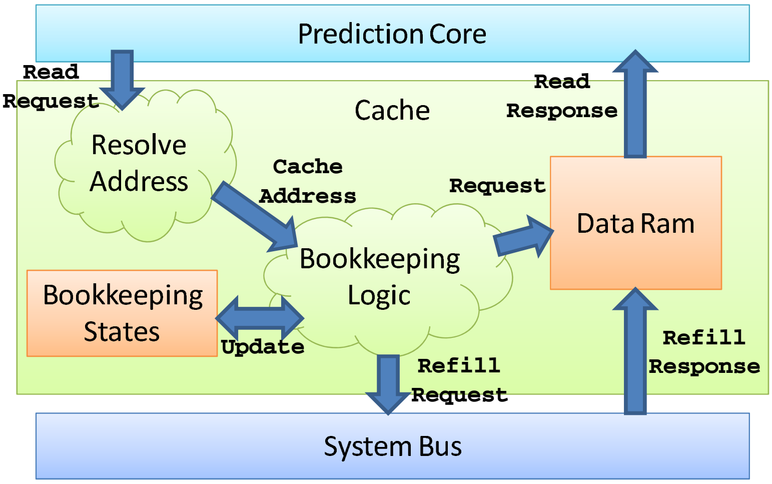


Figure 6.25: Conceptual data flow of cache system

### Address Translation and Bank Assignment

From Fig. 6.25 and the description above, we can see that the first step in the cache data flow is to translate the frame-based address scheme to the cache-based address scheme. The overview of a 2D cache is already covered in section 6.1.1. Fig. 6.26 depicts the whole translation process. The frame-based address scheme contains the X and Y coordinate in the unit of pixel, and it also contains a frame ID since a cache core stores multiple frames. The first step is to translate it to three parts: cache tag address, tag, and cache data address. The cache tag address consists of X and Y cache line address, and it can be used to specify a tag set in the tag RAM, which is included in the bookkeeping states block in Fig. 6.25. A tag set is an array of tags belonging to the cache lines with the same line address in different cache sets. The tag can be matched against the tag set to determine which cache set the data resides in, and if nothing in the tag set matches, then it is a cache miss. After the cache set is calculated, it can be combined with the X and Y word address, and they form the cache data address. The cache data address is then translated to the SRAM address to address the data memory. The SRAM address is composed of a bank ID and a word index. The assignment of bank ID and the word index is introduced below.

The bank ID assignment in cache data memory is shown in Fig. 6.26. There are several constraints to meet. In order to fulfill the 5-word-per-cycle throughput requirement of the prediction core, the data memory should contain at least 5 banks. To avoid bank conflict between the concurrently accessed words, the ladder bank assignment should also be used. Furthermore, the tag modulus is of a torus topology because the address is the modulus of a continuous frame field. The bank ID should comply with this topology; otherwise, if the concurrently accessed words cross the tag modulus boundary, the bank ID may conflict. Moreover, the tag modulus is composed by a 2-D array of cache lines, so the size of the tag modulus should be a multiple of cache line size. Combining these three constraints, the final bank ID assignment can be determined. Assuming a cache line contains $4 \times 4$ cache words, and 5 banks are used, then the dimension of tag modulus should be multiple of $LCM(4,5) = 20$. Alternatively, if 6 memory banks are used, which is still valid because $6 > 5$, the dimension of tag modulus should be multiple of $LCM(4,6) = 12$.

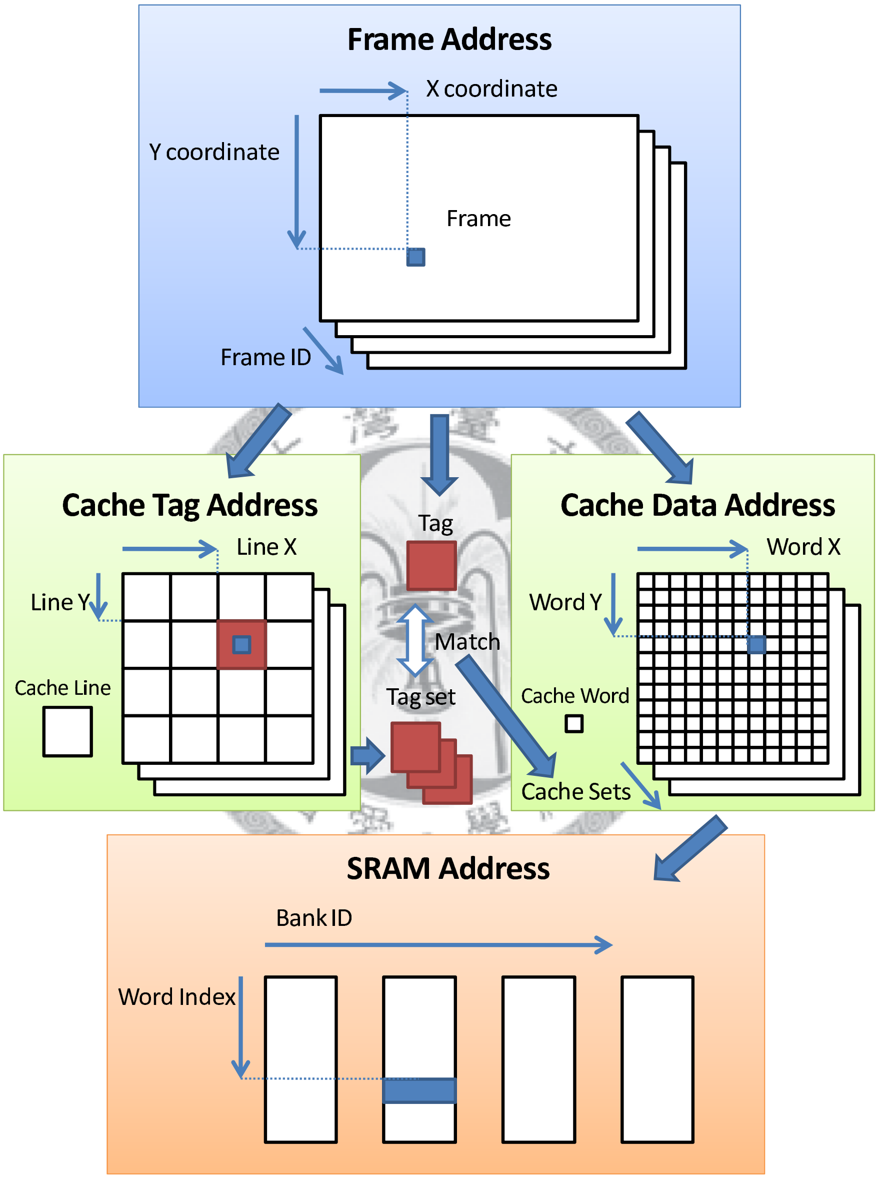


Figure 6.26: Address translation between different addressing schemes

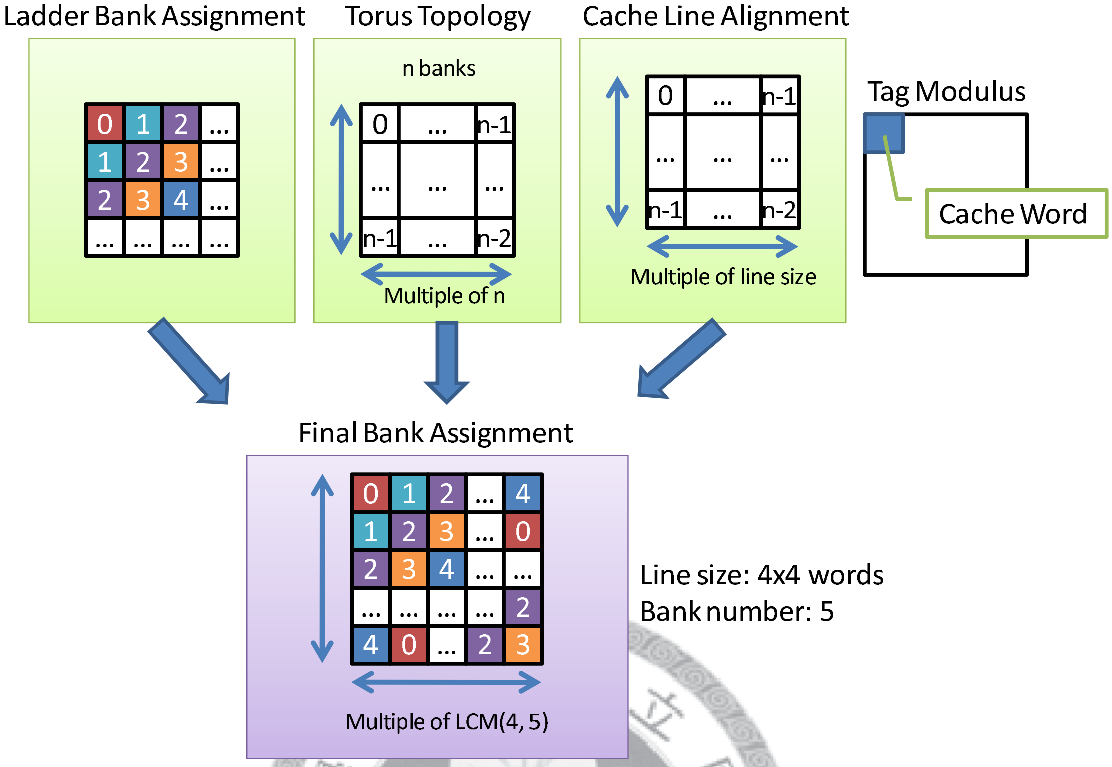


Figure 6.27: Bank ID assignment in cache data SRAM

The designer is free to choose these parameters as long as all the constraints are complied. In our proposed design, we use 5 banks and choose 20 × 20 words as the size of tag modulus.

As for the bank address, it can be illustrated in Fig. 6.28, where the tag modulus is chosen to be 20 × 20 words. The address is a simple raster scan order with $1 \times number\_of\_banks$ words as the constructing unit. Multiple cache sets can then be concatenated in the memory map.

### 6.5.2 Non-blocking Cache Refill

In a blocking cache, when cache miss happens, all the operations stall and wait for the refilling to complete. This architecture is easy to implement, but the performance is poorer. A non-blocking cache can handle hit when the previous miss is not refilled, which is called hit-after-miss. A more aggressive non-blocking cache can handle miss after miss, or essentially hit after several misses, which is called miss-after-miss. If a cache miss is the first miss in that particular cache line, then it is called a

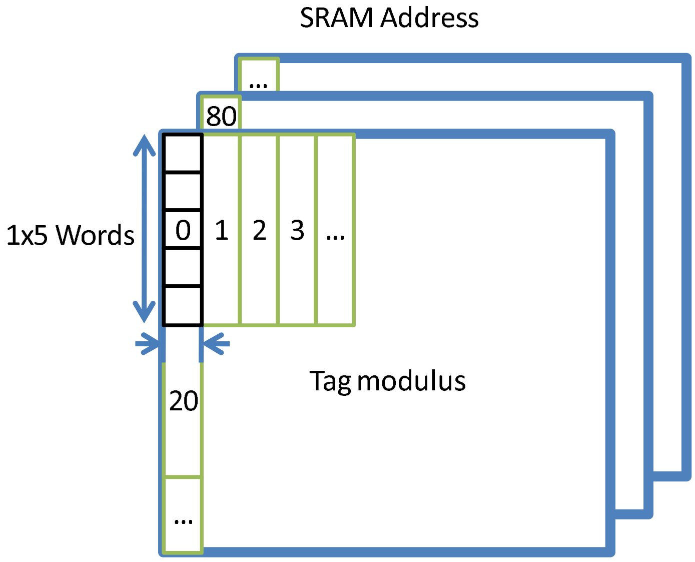


Figure 6.28: Bank address assignment in cache data SRAM

primary miss; otherwise, it is called a secondary miss. It is intuitive that a secondary miss should wait for its primary miss to finish the refilling instead of firing another refilling request, or the bandwidth caused by the duplicated refilling is wasted. As a result, in a miss-after-miss scenario, if the latter miss is a secondary miss, the reading request should be put in a waiting queue, but not firing a refilling request; on the other hand, if the latter miss is a primary miss, then another refilling request should be fired. According to the maximum number of supported on-the-fly primary misses, the non-blocking cache architectures can be further categorized. In our proposed architecture, up to three on-the-fly primary misses and consecutive four on-the-fly misses are supported.

Before explaining the non-blocking cache architecture, a blocking version is illustrated first. Fig. 6.29 shows a blocking cache architecture in a cloud diagram, where the cloud-like shapes represent rules, the ladder-shaped icons are FIFOs, and rectangular blocks are modules. The reading request comes from the prediction core and goes through FIFO reqQ to the address resolving rule. The resolved cache address is then sent to the bookkeeping logic through the addrQ. The bookkeeping logic queries the bookkeeping states and check if it is a cache hit. On a cache hit, the bank rotating amount is sent through readQ, and the reading request method on data RAM is called. When the rule ”Reply” gets both the signal in readQ and the

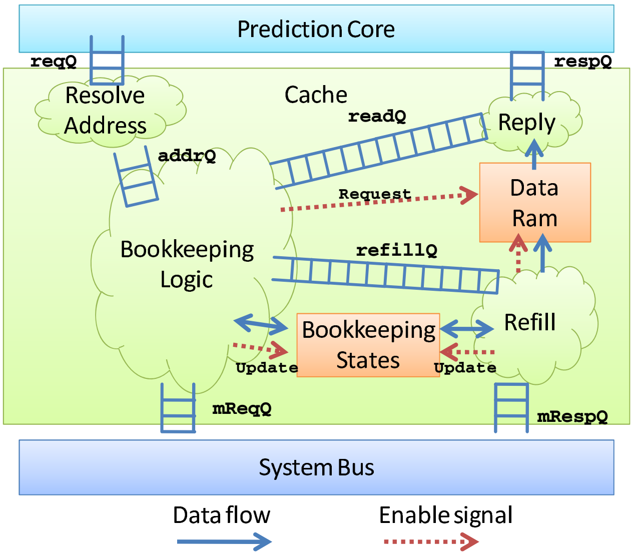


Figure 6.29: Architecture of blocking cache

returned data from the data RAM, it does the bank rotation according to the rotating amount specified in readQ, and then forward the rotated data back to the prediction core through respQ. On a cache miss, a refilling request is sent to the external memory bus through mReqQ, and almost everything else stalls until the rule ”Refill” gets the returned data from mRespQ. When the rule ”Refill” get the data, it updates the bookkeeping logic that the cache line is now valid, do bank rotation according to the parameter from refillQ, and write the data to the data RAM. Since the refilling is complete, the bookkeeping logic continue to read the data RAM as in cache hit.

Non-blocking cache architecture is shown in Fig. 6.30. There could be several hazards if things are not done right, including duplicated refilling as described above, refill-before-read hazard, and read-before-refill hazard. Refill-before-read hazard happens when refilled results overwrite a cache line that is still needed by previous cache hit. Similarly, the read-before-refill happens when a reading happens before the required data has finished refilling. In order to support hit-after-miss and miss-after-miss correctly without duplicated refilling, one more field in bookkeeping is added to track if a certain cache line is being refilled. If a miss happens and the field is on, the the bookkeeping logic knows it is a secondary miss, and refilling is not required.

Before explaining the technique used to resolve the refill-before-read and read-

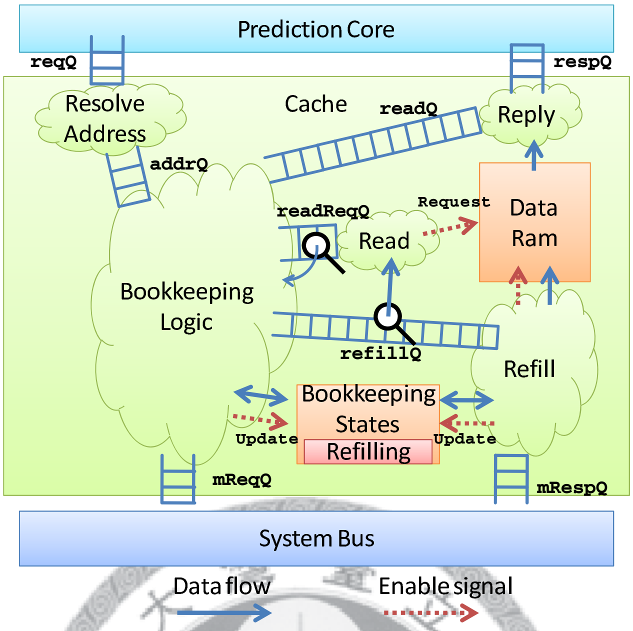


Figure 6.30: Architecture of non-blocking cache

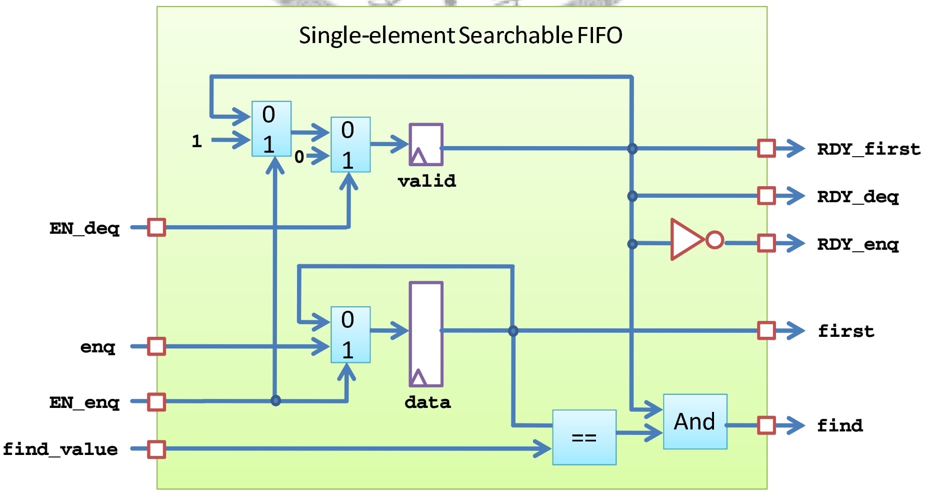


Figure 6.31: Schematic circuit of one-element searchable FIFO

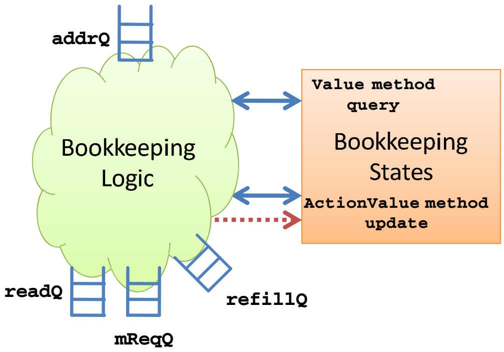


Figure 6.32: Bookkeeping logic expressed in a single rule

before-refill hazards, the concept of searchable FIFO should be explained first. A seachable FIFO provides a port that receives a data element, searches for that element within the FIFO itself, and returns whether it is found. The schematic circuit of one-element searchable FIFO is shown in Fig. 6.31. As we have the ability to check if a specific entry exists in a FIFO, the refill-before-read hazard can be eliminated by using seachable FIFO in readReqQ, so that if a cache line is waiting to be read, i.e. exists in readReqQ, the bookkeeping logic do not fire refilling request for that cache line. In the similar concept, the read-before-refill hazard can be removed by using searchable FIFO in refillQ. If a cache line has not finished being refilled, it can be searched in refillQ. In this case, the rule ”Read” stalls if the first element of readReqQ can be found in refillQ. With these mechanisms, a non-blocking cache supporting multiple primary misses can be implemented without any hazards.

### 6.5.3 Concurrent Reading and Prefetching

According to the analysis in section 6.4.2, it is necessary for the underlying cache architecture to support concurrent reading and prefetching requests. As the cache architecture is implemented in GAA methodology, the proposed method is better illustrated with cloud diagrams.

#### Rule Splitting and Rule Concurrency

The original bookkeeping logic is implemented in a single rule, which can be represented in a cloud diagram in Fig. 6.32. However a Value method ”query” and an

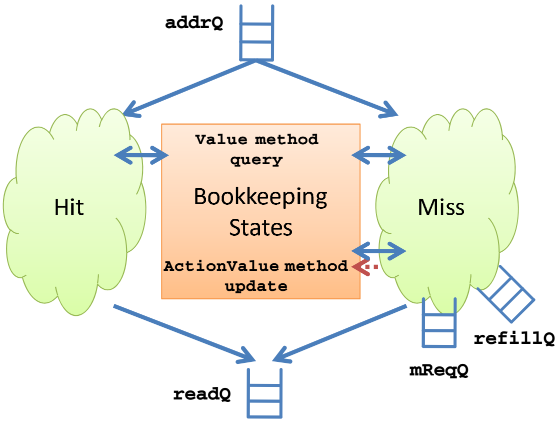


Figure 6.33: Bookkeeping logic expressed in split hit/miss rules

ActionValue method ”update” are invoked in that rule. Since the method query are scheduled to be appeared firing earlier than update in the same cycle, two copies of the same bookkeeping logic would conflict with each other, preventing parallelism. As a result, the rules must be rewritten in another form.

Cache hit and cache miss are mutually exclusive conditions for the bookkeeping logic. Therefore, the original rule can be rewritten as two rules handling the two situations respectively, which is illustrated in Fig. 6.33. In this diagram, the ”Hit” rule only invokes the ”query” method, while the ”Miss” rule invokes both the ”query” and ”update” methods. After rule scheduling, these rules would be scheduled as ”Hit before Miss”. Although rules ”Hit” and ”Miss” are mutually exclusive in this case, it is helpful when we need to duplicate the bookkeeping logic into two copies.

### Double Channel Architecture

Since we need to handle reading and prefetching requests concurrently, the bookkeeping logic must be duplicated to double the processing capability, namely the double channel architecture. However, if the rule is simply duplicated as shown in Fig. 6.34, then the reading rule conflicts with prefetching rule as analyzed in section 6.5.3. Continuing the analysis in section 6.5.3, if bookkeeping logic is split into ”Hit” and ”Miss” rules, and they are duplicated to ”ReadHit”, ”ReadMiss”, ”PrefetchHit”, and ”PrefetchMiss” rules, then the rule scheduling result would be ”{ReadHit, PrefetchHit} before {ReadMiss conflict PrefetchMiss}”. As a result, the

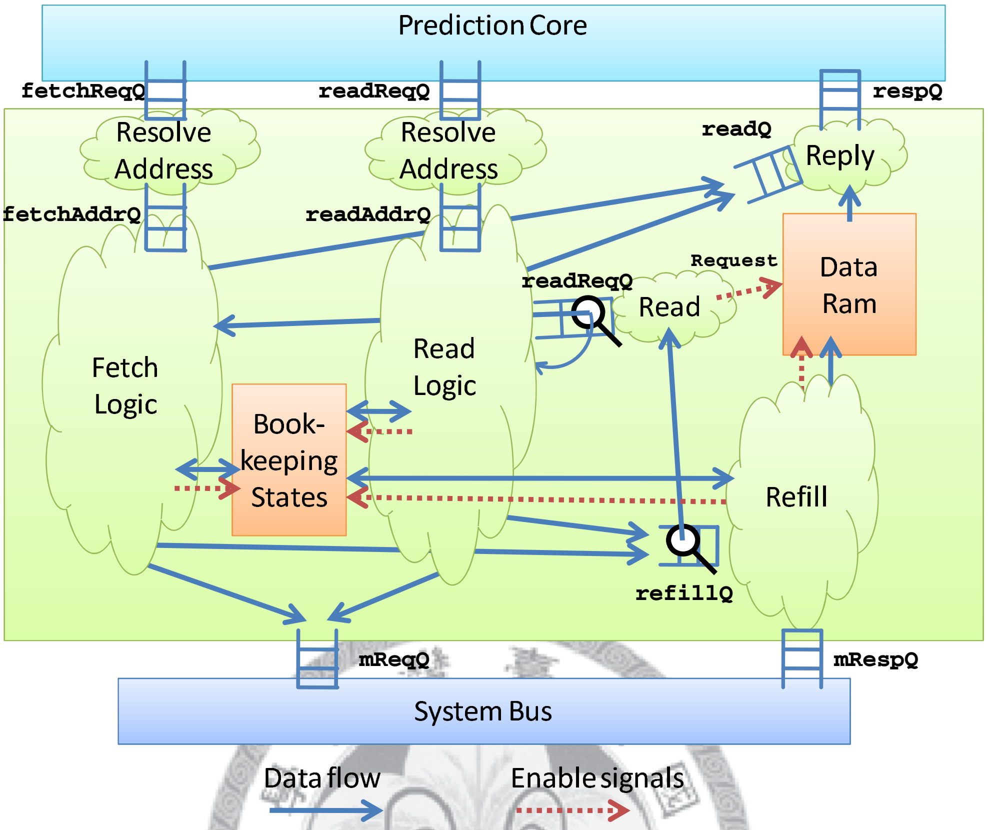


Figure 6.34: Architecture of non-blocking cache with concurrent reading and prefetching

two channels do not conflict unless both channels get cache misses. Since the probability of having misses in both channels is rare, the throughput would be almost the same as two independent processing channels.

## 6.6 Cache Profiling

The cache profiling is conducted on numerous video sequences in 1080p and 2160p resolution. All the figures in this section show the refilling bandwidth in the unit of size of frame per reference frame, which means how many equivalent frames are read when doing ME on a reference frame. The refilling bandwidth are shown in Fig. 6.35 and Fig. 6.36. A sequence with particularly high refilling bandwidth, pedestrian area, is picked up to show the configurations of cache vs. the refilling bandwidth in Fig. 6.37. Meanwhile, a sequence with particularly low refilling bandwidth, station2

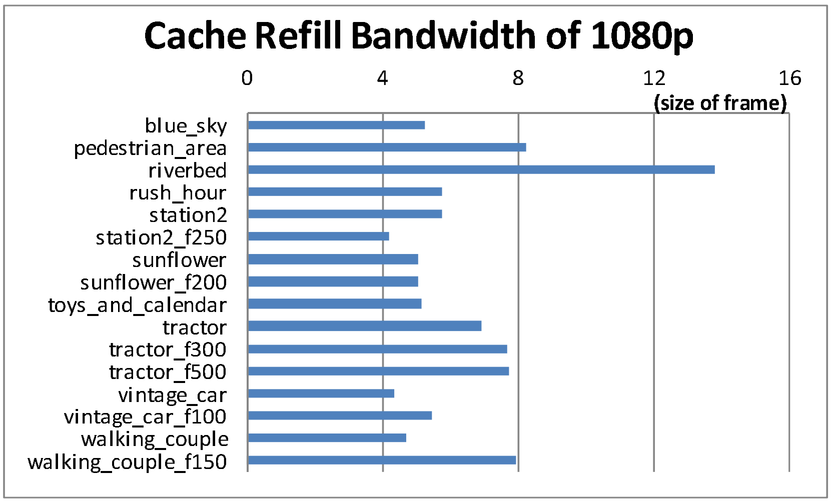


Figure 6.35: Cache refilling bandwidth of 1080p video sequences

from frame #250, is shown in Fig. 6.38.

## 6.7 Summary

In this chapter, a cache design for super high definition H.264 encoder is proposed. It is used as the reference frame buffer and achieves lower on-chip memory capacity and external bandwidth at the same time. Consequently, the silicon cost of the chip and the power consumption are reduced. An ordinary cache used in computer architecture is not suitable for this goal because of the limited throughput and the penalty of cache line splitting. For the high performance prediction core, low cache miss rate, light cache miss penalty, reduced overhead of data prefetching, zero line split penalty, and high throughput requirement must be satisfied. Data prefetching can lower the cache miss rate, but the overhead is not satisfying. With proposed rapid prefetching algorithm, the overhead is lowered by 84% and 64% for IME and FME data prefetching respectively. Furthermore, the cache pollution caused by data prefetching is also eliminated by the proposed replacement policy. Applying the MGAA methodology on the hardware architecture, a non-blocking 4-way cache with 300 MHz working frequency is proposed. This architecture lowers the cache miss rate by 15 folds, a 93% reduction, compared to traditional design.

The architecture proposed in ISSCC ’08 [87] is used as an anchor. It supports up

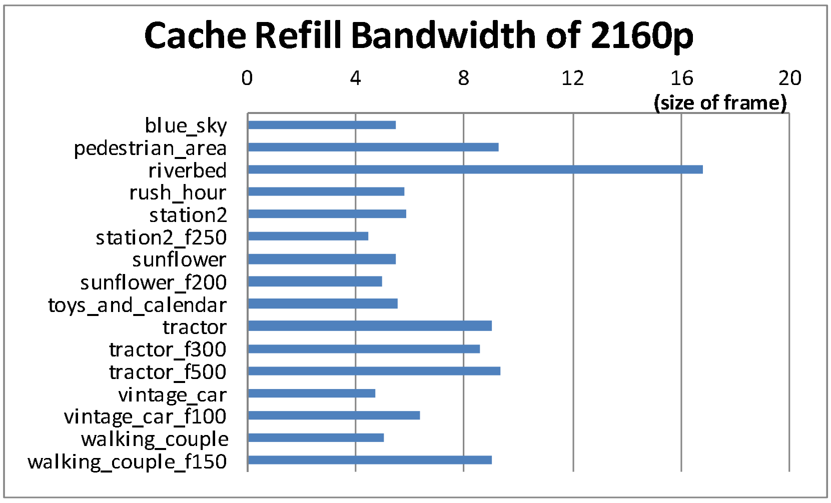


Figure 6.36: Cache refilling bandwidth of 2160p video sequences

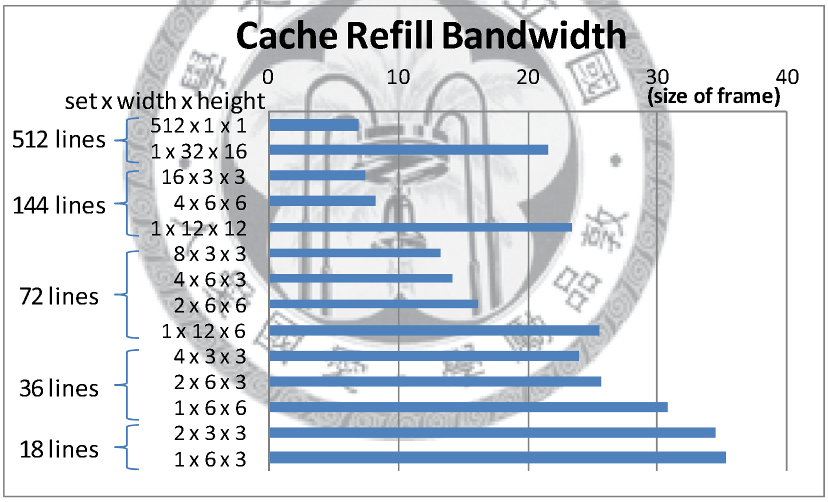


Figure 6.37: Cache refilling bandwidth of a dynamic scene (pedestrian_area)

Table 6.1: Comparison of normalized (per reference frame) on-chip memory capacity

| | 720p | 1080p | $4096 \times 2160p$ |
|---|---|---|---|
| VLSI Symp.'07 [86] | N.A. | 40.9 KB +<br>8.0 MB DRAM | N.A. |
| ISSCC'08 [87] | N.A. | 8.72 KB | 37.2 KB (scaled) |
| Proposed | 7.0KB | | |

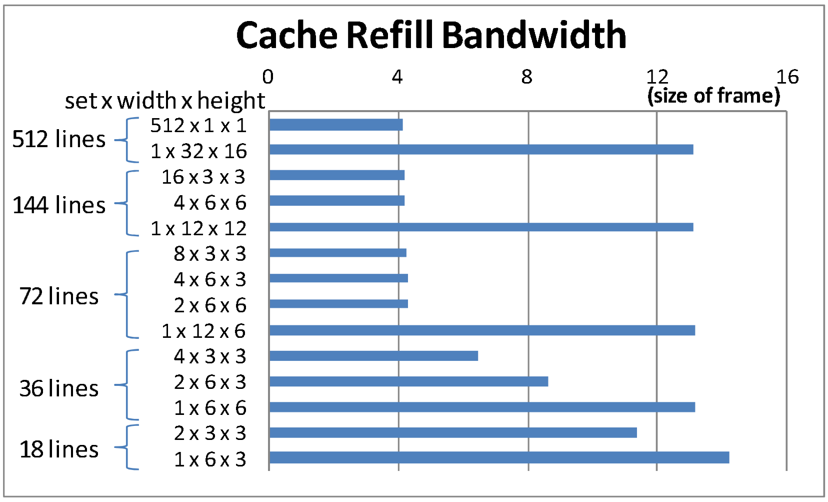


Figure 6.38: Cache refilling bandwidth of a static scene (station2_f250)

Table 6.2: Comparison of normalized (per reference frame) external memory bandwidth

| | 720p | 1080p | $4096 \times 2160p$ |
|---|---|---|---|
| VLSI Symp.'07 [86] | N.A. | 2.07 MB | N.A. |
| ISSCC'08 [87] | N.A. | 19.0 MB | N.A. |
| Proposed | 4.70 MB | 11.6 MB | 51.6 MB |

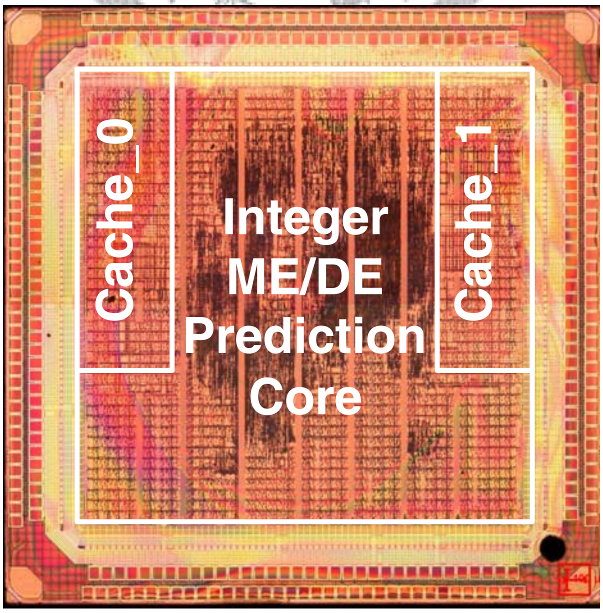


Figure 6.39: Die photo of prototype chip

Table 6.3: The specifications of the prototype chip design

| | |
|---|---|
| Technology | TSMC 90 nm CMOS LOGIC Low Power<br>LowK Cu 1P9M 1.2&2.5V |
| Package | COB256 S1 |
| Core Size | 2.1 mm × 2.1 mm |
| Logic Gate Count | 230K (NAND2 gate) |
| SRAM capacity | 8 KBytes |
| Working Frequency | 300 MHz |
| Power | 265mW @ 300 MHz, 1.2V |
| Throughput | $4096 \times 2160p$, 24 fps quad HD in H.264/AVC format<br>Three-view 1080p or seven-view 720p, 30 fps in MVC format |

to 1920 × 1080 resolution, uses 17.44KB of on-chip SRAM as the reference frame buffer, and requires 19.0 MBytes per reference frame. The proposed cache architecture uses 6.4KB of on-chip SRAM. Combined with the cache tag memory and control overhead, they totally contribute to chip area equivalent to 14KB of SRAM, which is at least 20% smaller than anchor, and the control overhead of on-chip SRAM in the anchor design is not considered. If the anchor design is directly scaled to the maximum resolution supported in the proposed design, then their on-chip SRAM would grow proportionally to the frame size, which is 4.27 times larger. Under this circumstance, the proposed design is at least 82% smaller than the anchor design in the same resolution. As for external bandwidth, the proposed design uses 4.70, 11.6, and 51.6 MBytes per reference frame in 720p, 1080p, and $4096 \times 2160p$ resolutions. Compared with the anchor design in the 1080p resolution, the proposed architecture uses 39% less external bandwidth. The complete comparison of the memory capacity is listed in table 6.1, and the external memory bandwidth can be found in table 6.2. In order to fairly compare works supporting different profiles, data are normalized to a per-reference-frame basis, so works only supporting baseline profile do not take advantage.

The proposed architecture is implemented in an prototype ASIC design with specifications shown in table 6.3, and the die photo is shown in Fig. 6.39. The prototype chip is fabricated with 90*nm* technology. Only 8KB SRAM is used. The power consumption is 265*mW* operating at 300MHz. The maximum throughput of 4096×2160p video is achieved with only 230K logic gate count. It shows the area-efficient feature of the proposed architecture.

# Part III

# System Analysis and Architecture Design of 4096×2160p Multivew Video Single-Chip Encoder

國立臺灣大學
敦品勵學 ‧ 愛國愛人

# Chapter 7

# System Bandwidth Analysis of Multiview Video Coding with Precedence Constraint

Multiview video coding (MVC) systems require much more bandwidth and computational complexity relative to mono-view video systems. Thus, when designing a VLSI architecture for MVC systems, the hardware resource allocation is a critical issue. In this chapter, we propose a new system bandwidth analysis scheme for various and complicated MVC structures. The precedence constraint in the graph theory is adopted for deriving the processing order of frames in a MVC system. In addition, current block centric scheduling (CBCS) and search window centric scheduling (SWCS) are proposed for MVC bandwidth analysis. By adopting data reuse schemes, several design points are explored with the aid of the proposed analysis scheme. The suitable hardware resource allocation can be easily determined.

## 7.1 Introduction

In an MVC system, Motion estimation (ME) and disparity estimation (DE) are the major components. They dominate the greater part of the computational complexity and memory bandwidth in the system. The large computational complexity is due to a lot of candidate blocks to be matched, and the huge memory bandwidth results from loading the data of candidate blocks. The instruction profiling shows that 2.76 tera-operations/s (TOPS) of computational loading and 4.25 tera-bytes/s (TB/s) of memory access are required for real-time encoding single-view SDTV videos [88]. The required computational loading and memory access for MVC systems are even much larger. The challenge of large computational complexity can be solved by parallel processing skills or fast prediction algorithms. However, the system memory bandwidth is limited in a VLSI hardware system. In tradition, the data of the current macroblock (MB) and the search window (SW) are loaded from system memory and then buffered in on-chip SRAMs or registers. The system memory bandwidth can be reduced by local data reuse schemes. Some data reuse strategies have been proposed with different tradeoffs between system bandwidth and local memory size [89][90]. In addition, a frame-level data reuse scheme has been proposed to reduce more memory bandwidth for multiple-reference-frame ME [88]. However, as the design space extends from mono-view to multiview video systems, the demand for system bandwidth, on-chip and off-chip buffers increases with an order. Various coding structures for MVC are required for different applications, which greatly increase the design challenge of the system design. Thus the previous data reuse schemes for mono-view video systems no longer efficiently support MVC. In this chapter, a new system analysis scheme with precedence constraint is proposed for MVC systems. It utilizes the relation between SWs for ME and DE and combines the previous data reuse schemes. With the aided of the precedence constraint, the most suitable scheduling and resource allocation for every coding structures can be systematically derived.

The rest of this chapter is organized as follows. The previous data reuse schemes for mono-view video coding systems is briefly introduced in Section 7.2. In Section 7.3, the proposed system analysis scheme with precedence constraint is described. The performance evaluation and discussion are shown in Section 7.4. Finally, Section 7.5 summarizes this chapter.

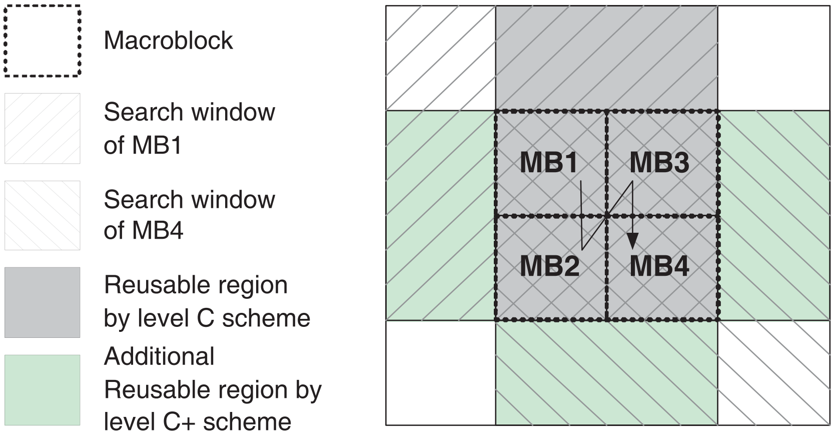

Macroblock
Search window of MB1
Search window of MB4
Reusable region by level C scheme
Additional Reusable region by level C+ scheme
MB1
MB3
MB2
MB4

Figure 7.1: Illustration of the level-C+ data reuse scheme.

## 7.2 Previous Data Reuse Schemes for Mono-View Video Coding Systems

Data reuse is an important concept and is usually adopted in most VLSI designs for ME in video coding systems. Many data reuse schemes have been proposed, and they can be generally classified into two categories, intra-frame and inter-frame data reuse. Intra-frame data reuse schemes utilize the characteristic that the SWs of the neighboring MBs overlap each other to save the memory bandwidth. With different trade-off judgement between system bandwidth and on-chip memory size, they can be classified into four schemes and indexed from level-A to level-D [89]. Level-A scheme requires the smallest on-chip memory size and the highest external bandwidth, while level-D scheme has the largest on-chip memory size and the lowest external bandwidth. Among four schemes, the level-C scheme is often adopted because it is more suitable to be implemented with the current VLSI technology. To enhance the scalability of data reuse and fully utilize the hardware resource, Chen *et al*. [90] propose the level-C+ data reuse scheme. As shown in Fig. 7.1, it not only fully reuse the overlapped SWs in the horizontal direction, but also partially reuse the overlapped SWs in the vertical direction. It inserts many design choices between the design choices of level-C and level D scheme. The system bandwidth can be further reduced with a little overhead of oh-chip memory size.

On the other hand, inter-frame data reuse schemes, such as single reference

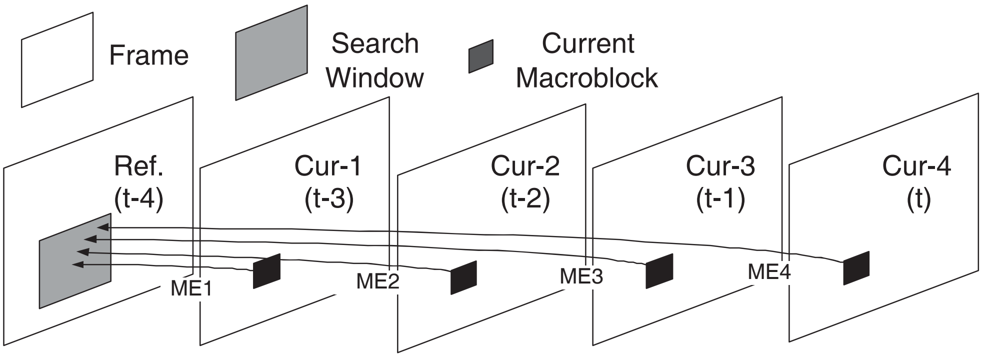


Figure 7.2: The concept of SRMC scheme. It exploit the frame-level data reuse for MRF-ME.

frame multiple current MBs scheme (SRMC) [88], reuse SW data in the frame-level when performing multiple-reference-frame ME. The concept of the SRMC scheme is shown in Fig. 7.2. The current MBs located in the same positions in their corresponding frames have an identical SWs in a reference frame. Therefore, only single SW memory is required. The system bandwidth can also be further reduced. To achieve inter-frame data reuse, the ME procedure for MEs have to be rescheduled, that is, ME for one current MB in different reference frames are processed at different time slots.

## 7.3 Proposed System Analysis Scheme with Precedence Constraint

MVC is a challenged task due to the fact that various coding structures and different number of view channels. The processing scheduling and resource allocation greatly effect the architecture performance of MVC. The previous data reuse schemes for mono-view video systems are not sufficient for MVC. In this section, a system analysis scheme with precedence constraint is proposed to derive the suitable processing scheduling and the hardware resource allocation systematically. Before introducing the proposed analysis method, the system architecture of a MVC system is defined first. Then, the intra-inter-view data reuse scheme with precedence constraint, its corresponding analysis, and the case studies are described.

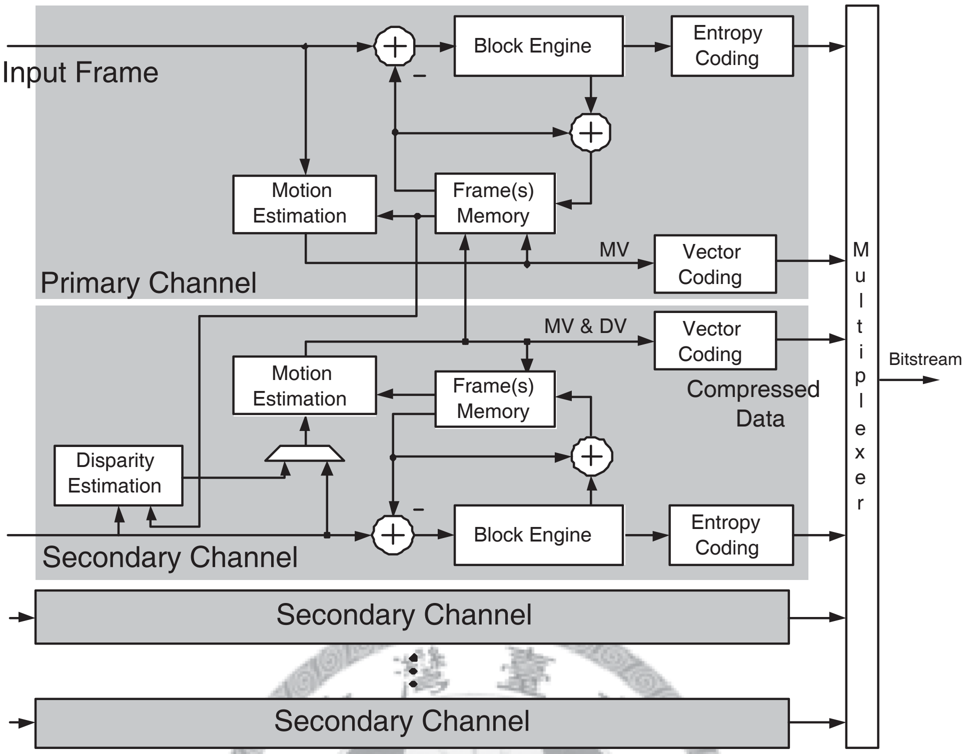


Figure 7.3: Block diagram of the proposed multiview video encoder.

### 7.3.1 System Architecture of Multiview Video Coding Systems

Figure 8.4 shows the block diagram of the proposed multiview video encoder, which is based on the hybrid coding scheme. There are two kinds of view channels, the primary channel and the secondary channel. They are both encoded with H.264/AVC. There is no DE operation in the primary channel. The number of primary and secondary channels depends on the coding structure. The block engine includes quantization, transform, and deblocking filter, etc.. After encoding, the compressed bitstream of each channel is transmitted. In addition, the hardware system architecture is defined in Fig. 7.4. It consists of three part, the multiview video encoding engine, the system memory, and the processor. Most of the system bandwidth are required in the ME and DE parts. The busy communication between the system memory and the SW buffers make the bandwidth loading of the system bus a critical issue.

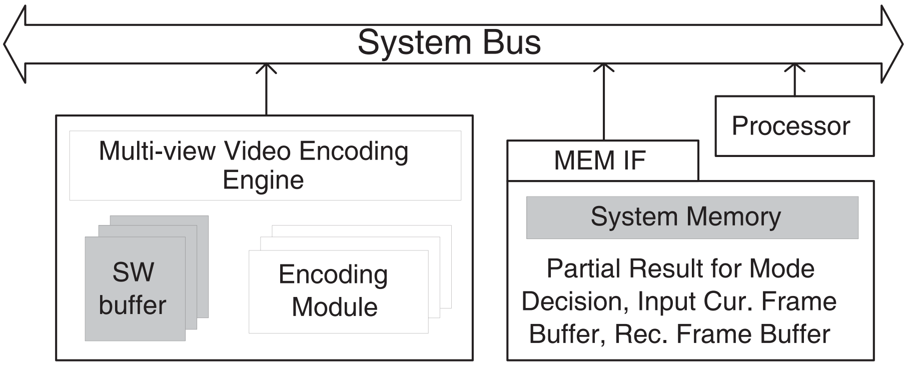

System Bus
Multi-view Video Encoding Engine
SW buffer
Encoding Module
MEM IF
Processor
System Memory
Partial Result for Mode Decision, Input Cur. Frame Buffer, Rec. Frame Buffer

Figure 7.4: Hardware system architecture for a MVC system.

### 7.3.2 System Bandwidth Analysis with Precedence Constraint

**Precedence Constraint**

To cope with the complicated processing order of frames in a MVC system, we found that there exists the data dependency between frames, that is, a current frame cannot be encoded until its reference frames are encoded. Therefore, the precedence constraint, which is a concept in the graph theory, is adopted to interpret the data dependency betweens the frames. Each frame can be regarded as a vertex $v_i$ with the sequence order $S(v_i)$. Each prediction arrow can be regarded as an edge $e_{ij}$ with weight $d(e_{ij})$ between two vertices. Therefore, a constraint graph $G(V,E)$ can be constructed with the following criterion,

$$d(e_{ij}) = \begin{cases} -\infty, & \text{if there is no edge connected from } v_i \text{ to } v_j \\ 1, & \text{otherwise.} \end{cases} \tag{7.1}$$

$$S(v_j) = \max\{S(v_i) + d(e_{ij}), 1\}. \tag{7.2}$$

Figure 7.5 shows an example of the precedence constraint applied on a stereo video coding structure. The first frames in both view channels are intra-coded. Thus no data dependency exists between them, and their vertex values are assigned 1. There is only one edge connected to $v_3$, so $S(v_3)$ is assigned $S(v_1) + d(e_{13}) = 2$. The other vertex values are defined with the same rule. Therefore, the processing order of the frames can be derived.

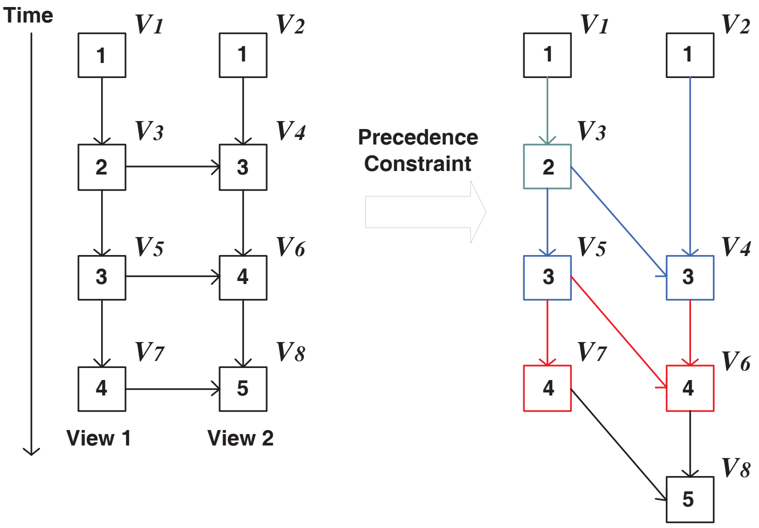


Figure 7.5: Illustration of the precedence constraint applied on a stereo video coding structure.

### System Bandwidth and Memory Analysis Scheme

Although the processing order of the frames in a MVC system is derived, the processing order of prediction arrows is not designed yet. The prediction is carried out by the limited hardware resource, such as processing elements (PEs) and on-chip buffers. To make the analysis more systematically, two kinds of scheduling, current block centric scheduling (CBCS) and search window centric scheduling (SWCS), are proposed for convenience for the analysis. In CBCS, each current block and its corresponding SWs are loaded from system memory for ME or DE. The prediction for the next MB will not start until the mode decision of this current block is finished. Therefore, each current frame is loaded only once from system memory. CBCS is a common scheduling with a simple data flow. However, in some MVC structures, a reconstructed frame may be accessed several times from system memory if it is required for predicting several current frames. It wastes much system memory bandwidth. In contrast to CBCS, each reconstructed frame taken as a reference frame is loaded only once in SWCS. When a SW is loaded, its corresponding current blocks are also loaded for cost computation. However, the mode decision of these current blocks are not finished if they have other reference frames. Therefore, the partial result for mode decision is needed to be stored in the on-chip or off-chip memory.

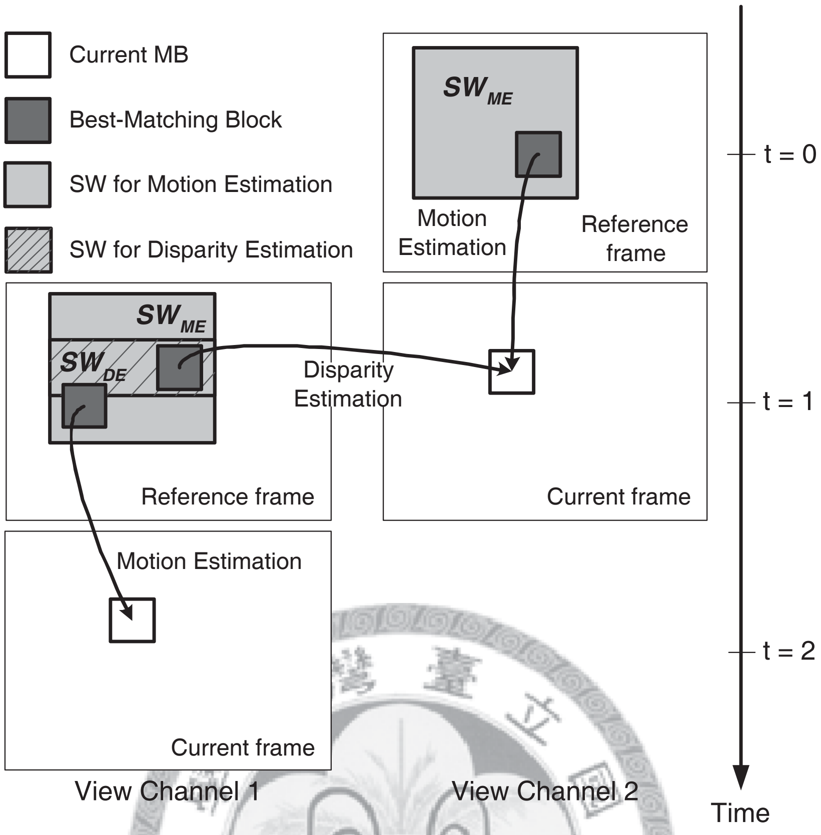


Figure 7.6: Inter-view data reuse scheme can efficiently save the bandwidth for loading SWs for DE.

Storing in the on-chip memory is not a suitable choice usually due to the requirement of much silicon area. The little penalty of SWCS is the little quality loss due to incomplete MV predictor generation. The SRMC scheme belongs to this scheduling. Take Fig. 7.6 as an example, the proposed inter-view data reuse scheme for our prior stereo video system [37] can be extended for MVC. With SWCS, $SW_{ME}$ is first loaded for the current block in view channel 2 for DE. Then, the current block in view channel 1 is loaded for ME. Therefore, the required on-chip memory and bandwidth for $SW_{DE}$ are saved.

The choice of CBCS or SWCS for a MVC system greatly effect the performance of system architecture, especially system bandwidth. Whether CBCS or SWCS is chosen, the system bandwidth can be described as

$$BW_{SYSTEM} = BW_{ME} + BW_{DE} + BW_{PR} + BW_{BE}, where \tag{7.3}$$

$$BW_{ME} = n(ref_{ME}) \times BW_{SW_{ME}}, \tag{7.4}$$

$$BW_{DE} = n(ref_{DE}) \times BW_{SW_{DE}}, \tag{7.5}$$

$$BW_{PR} = n(Cost_{PR}) \times BW_{Cost_{PR}}. \tag{7.6}$$

$$BW_{BE} = (n(Ori) + n(Rec)) \times BW_{frame}, \tag{7.7}$$

The system bandwidth is composed of four parts, $BW_{ME}$, $BW_{DE}$, $BW_{PR}$, and $BW_{BE}$. $BW_{SW_{ME}}$ and $BW_{SW_{DE}}$ are the required bandwidth for loading SWs for ME and DE, respectively. They depend on whether the intra-frame data reuse scheme is adopted. $BW_{BE}$ is the required bandwidth for the block engine introduced in Section 7.3.1. $n(ref_{ME})$, $n(ref_{DE})$, $n(Ori)$, and $n(Rec)$ are the frequencies of loading or transfering $SW_{ME}$, $SW_{DE}$, original frames, and reconstructed frames, respectively, through the system bus in every time slot. $BW_{PR}$ is the required bandwidth for sending or loading the partial results of cost from the system memory.

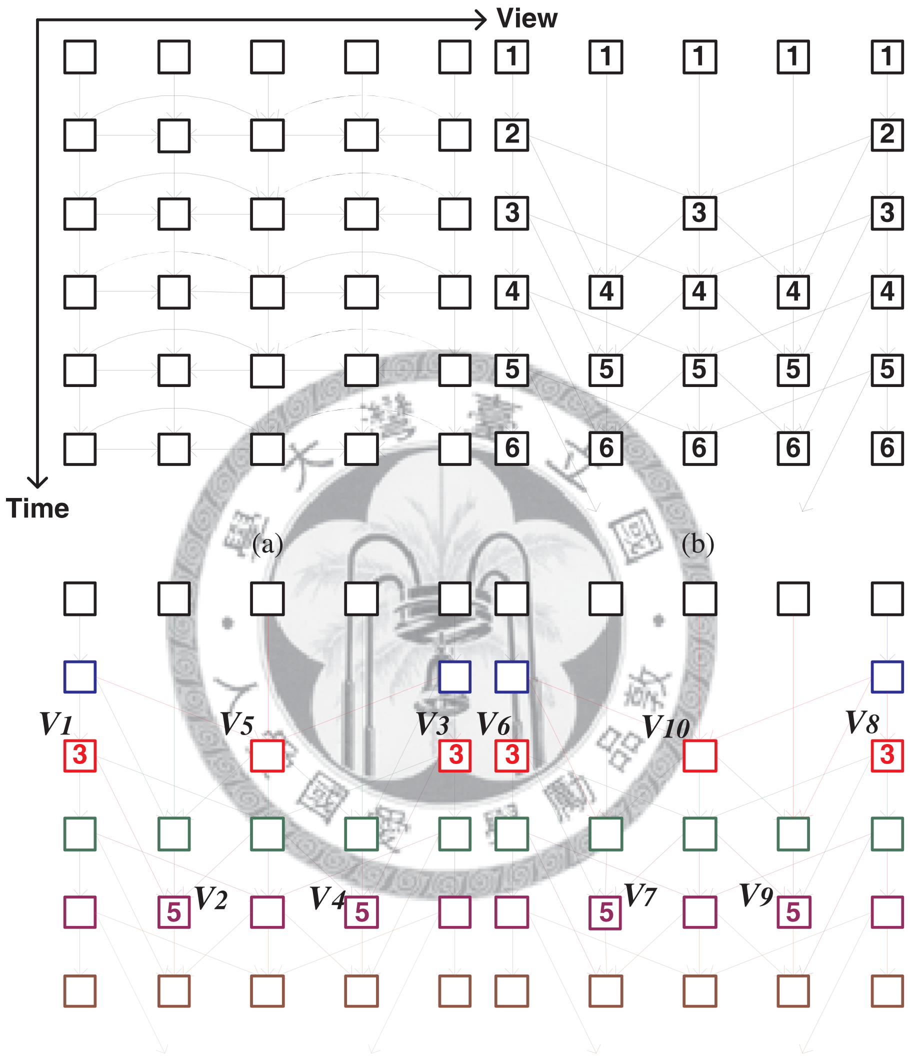
In addition, for any two vertices $v_i$, $v_j$ connected by an edge $e_{ij}$, the distance between $v_i$ and $v_j$ is defined by

$$D(v_i, v_j) = v_j - v_i. \tag{7.8}$$

$D(v_i, v_j)$ is either equal to 1 or bigger than 1. In CBCS, $n(Ori)$ and $n(Rec)$ are equal to the number of view channels regardless of $D(v_i, v_j)$. $n(ref_{ME})$ and $n(ref_{DE})$ depend on the coding structures. $n(Cost_{PR})$ is equal to zero because the mode decision can be finished immediately without storing partial results in CBCS. In the case of SWCS, if $D(v_i, v_j) > 1$, it means that for a SW, its corresponding current frames have different vertex values. According to the precedence constraint, the partial results of cost is needed to be stored in the system memory. Thus $n(Ori)$ and $n(Cost_{PR})$ increase. Usually, $n(ref_{DE})$ in CBCS is bigger than that in SWCS. It exists the trade-off between loading SWs and storing partial results of cost. To make the analysis more comprehensive, a design example is shown in the next section.

## 7.4 Case Studies and Performance Evaluation

Fig. 7.7 shows a design example of the proposed system bandwidth analysis with precedence constraint. The coding structure consists of five view channels. First,



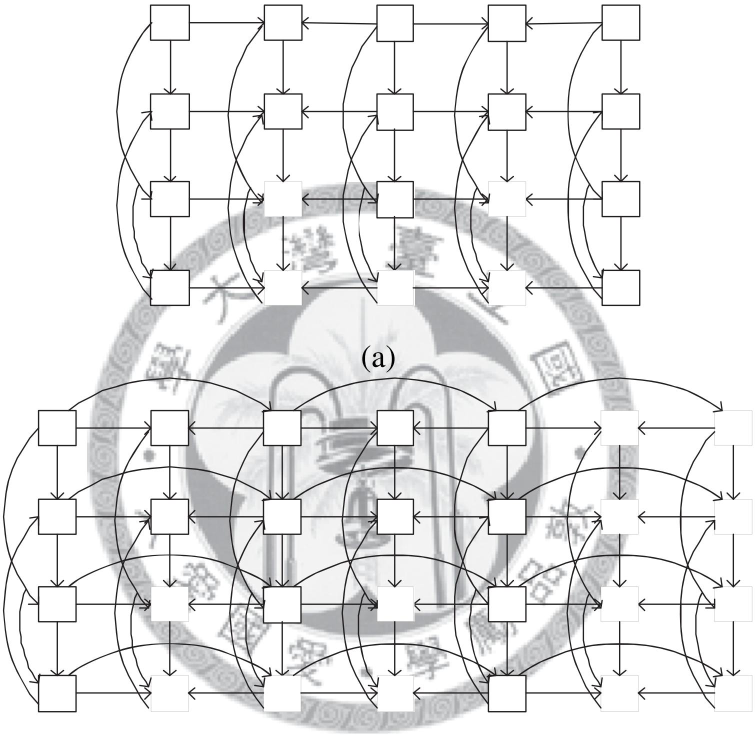
(a) (b)

(c) (d)

Figure 7.7: Design example of system bandwidth analysis for 5-view MVC systems. (a) The coding structure. (b) The processing order of frames are derived by the proposed method. (c) CBCS applied. (d) SWCS applied.

the vertex values, which are regarded as the processing order, are derived by the proposed method. Then, the frames are arranged according to the processing order as shown in Fig. 7.7 (c). Figure 7.7 (c) and (d) shows two schedulings. The prediction arrows with the same color means that these operations can be executed at the same pipeline stage if the required SWs are ready. FSBMA and level-C data reuse scheme is adopted. In the case of CBCS, $n(Ori) = n(Rec) = 5$. It means five original frames and five reconstructed frames are transmitted through the system bus in a time slot. Among the prediction arrows with the same color, there are five ME prediction arrows and four DE prediction arrows. Thus $n(ref_{ME}) = 5$ and $n(ref_{DE})$ can be assigned 4. However, some frames, such as $v_5$, have three corresponding current frames. The SWs for the current frames overlap, so the bandwidth is reduced. $n(ref_{DE}) = 4 - 2 = 2$ and $n(Cost_{PR}) = 0$. With SWCS applied in Fig. 7.7 (d), $D(v_6, v_7) = D(v_8, v_9) = 2 > 1$, so the mode decision of $v_7$ and $v_9$ can not be finished in a time slot. It also means the current frames of $v_7$ and $v_9$ have to be load twice. Thus $n(Ori) = 5 + 2 = 7$,$n(Rec) = 5$, and $n(Cost_{PR}) = 2 \times 2$. The multiplier 2 for $n(Cost_{PR})$ represents the data are sent off-chip for storage and loaded on-chip for final mode decision. After all the parameters are derived, the system bandwidth can be calculated.

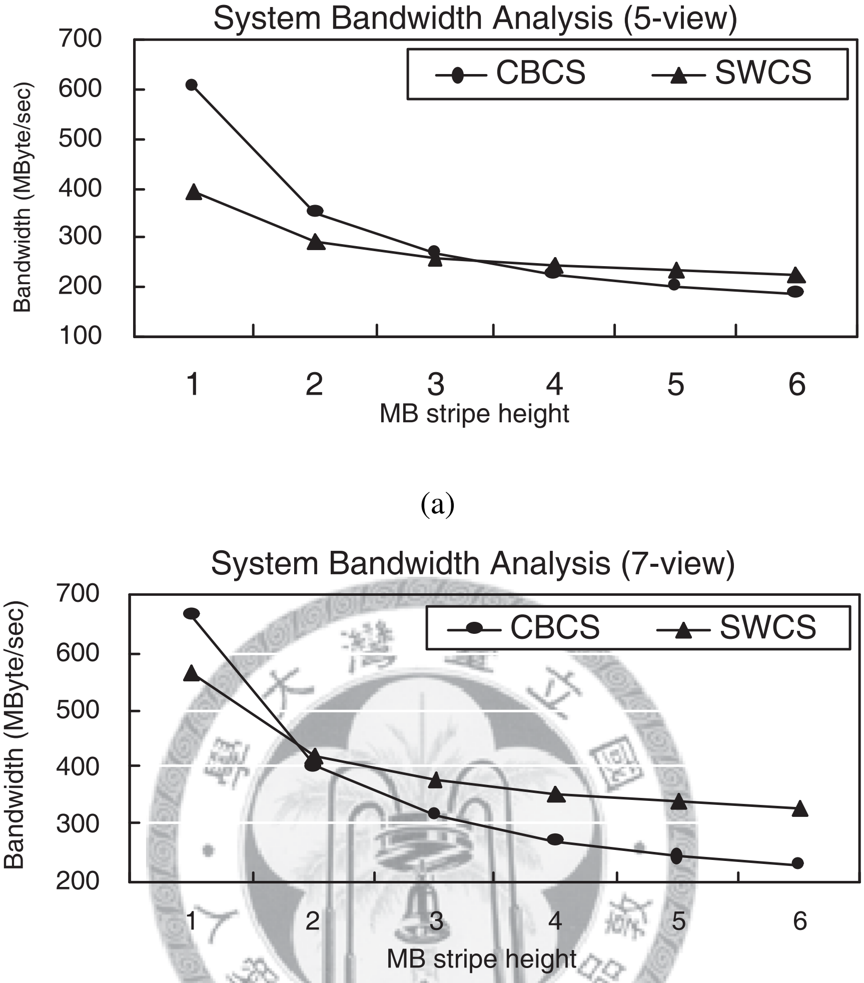
The proposed analysis method can support more complicated MVC structures. Two MVC structures with 720 × 480 frame size and 30 fps, as shown in Fig. 7.8, are verified. FSBMA is adopted for ME and DE. The ME/DE search range is [–64, +63]/[–64, +63] in the horizontal direction and and [–64, +63]/[-16, +15] in the vertical direction. Level-C+ scheme is adopted for scalable bandwidth analysis by adjusting the reusable MB stripe in the vertical direction. The analysis charts for two schedulings are shown in Fig. 7.9. MB stripe height represents the degree of partial data reuse in the vertical direction. When MB stripe is equal to 1, the level-C+ scheme is simplified to become the level-C scheme. With the increase of the MB stripe height, two curves intercepts with each other. The bandwidth requirement is lower in CBCS with large MB stripe height. The reason is that in CBCS, $n(ref_{DE})$ is usually bigger than that in SWCS. Thus the bandwidth requirement for $SW_{DE}$ is higher. However, with the increase of the reusable ME stripe height, $BW_{SW_{DE}}$ is



(a)

(b)

Figure 7.8: Two MVC structures for system bandwidth analysis. (a) Five views with IBBP structure in the time domain. (b) Seven views with IBBP structure in the time domain and hierarchical B structure in the inter-view domain.



(a)

(b)

Figure 7.9: Bandwidth analysis charts for CBCS and SWCS. (a) and (b) are the analysis of MVC structures in Fig. 8 (a) and (b), respectively.

getting lower. In addition, $n(Cost_{PR})$ and $n(Ori)$ are the overhead in the SWCS. They can not be reused by adopting the level-C+ scheme. The trade-off between the system bandwidth and the required on-chip memory can be easily observed from the analysis. Therefore, the proposed analysis scheme provides effective quantitative design selection for MVC systems.

## 7.5 Summary

This chapter presents a new bandwidth analysis scheme for various MVC structures. The concept of precedence constraint in the graph theory is adopted to derive the processing order in a MVC structure. In addition, two schedulings in MVC are proposed for systematical analysis. With the combination of the level-C+ data reuse scheme, several design points can be derived. Hardware resource allocation can be systematical defined with the trade-off between system bandwidth and on-chip memory.

國立臺灣大學
敦品勵學 · 愛國愛人

# Chapter 8

# System Architecture of a 4096×2160p Multivew Video Single-Chip Encoder

Multiview video coding (MVC) plays an important role in a 3D video system. In addition, the resolution of HDTV is increasing to present more vivid perception for users. Computational complexity of dozens of TOPS make VLSI solution become necessary. However, a large amount of external memory bandwidth, on-chip SRAM size, and complex MVC prediction structures are three main design challenges of implementation of MVC hardware architecture. In this chapter, the first MVC single-chip encoder is proposed for H.264/AVC Multiview Extension and High Profile for 3D and quad-full HD (QFHD) TV applications. A 4096×2160p multiview video encoder chip is implemented on a 11.46$mm^2$ die with 90$nm$ CMOS technology. An eight-stage macroblock pipelined architecture with proposed system scheduling and cache-based prediction core supports real-time processing from one-view 4096×2160p to seven-view 720p videos. The 212M pixels/s throughput is 3.4 to 7.7 times higher than the state-of-the-art encoder chips. The 407Mpixels/W power efficiency is achieved, and 94% on-chip SRAM size and 79% external memory bandwidth are saved.

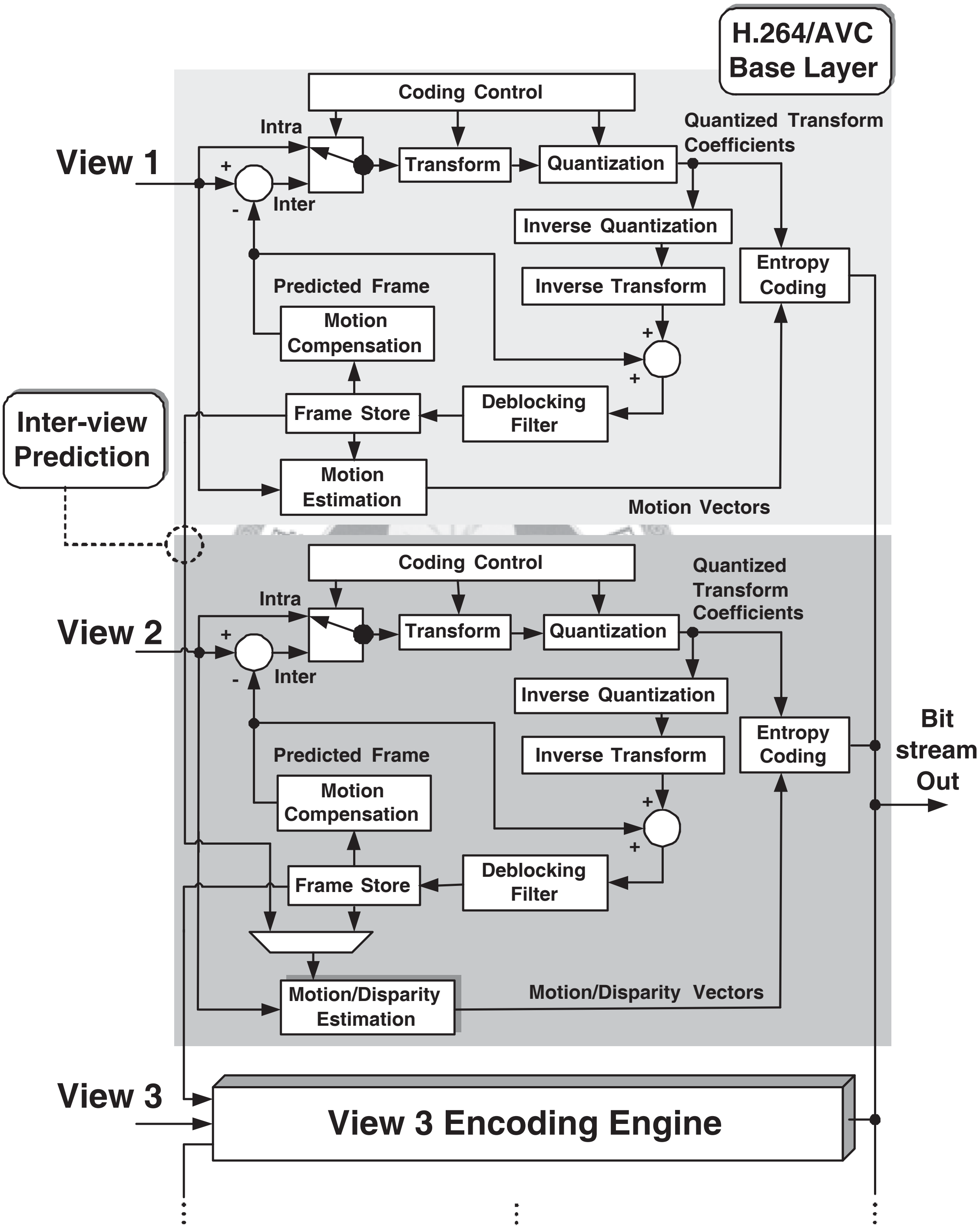


Figure 8.1: Block diagram and data flow of an MVC encoder system.

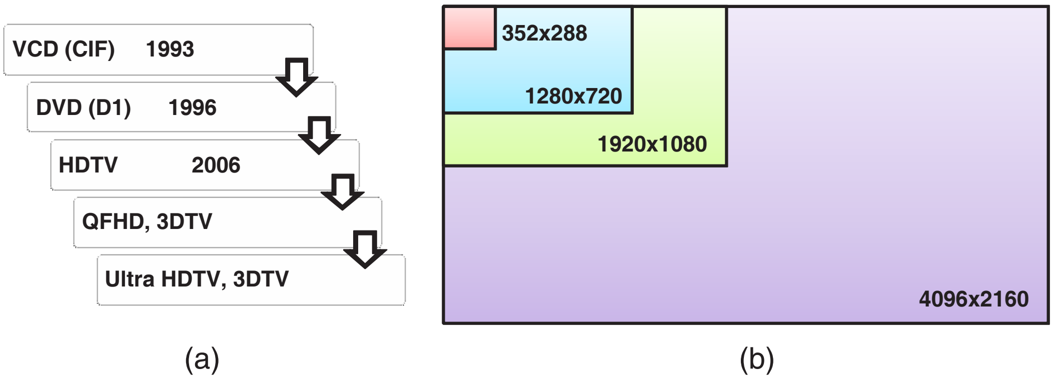


Figure 8.2: (a) Evolution of video resolution and (b) Illustration of relative size of video resolutions.

## 8.1 Introduction

To provide more vivid perception, TV resolution is getting higher and higher. In addition, 3D video becomes emerging because it can present immersive and complete scenes. Therefore, multiview video coding (MVC) is currently being developed as an extension of H.264/AVC [91]. The block diagram of H.264/AVC Multiview Extension system is illustrated in Fig. 8.1. H.264/AVC High Profile is adopted as the base layer. The most significant feature which differs from original H.264/AVC standard is inter-view prediction, which is also called disparity estimation (DE). DE can effectively exploit the inter-view redundancy and saves 20% to 30% of bit rates. Output bistream of each views are assembled and then transmitted. The bitstream format is compatible with H.264/AVC, so a single-view H.264/AVC decoder can decode the the base layer. However, DE and motion estimation (ME) require ultra high computation and memory access. To encode a 3-view 1080p video, 82.4TOPS computing power and 54.6TB/s memory access are required with a full search algorithm. Moreover, view scalability is a critical functionality to deal with various coding structures of 3D video.

There are multiple dimensions to improve the quality of video contents, such as higher resolution, multiple viewing channels, higher frame rate, wider dynamic range, etc. Historically, Fig. 8.2 shows the video resolution evolves with time. The angular resolution of naked eye is around 30 angular seconds [92]. Assuming the viewing distance is equal to the width of the screen, resolution of $8k \times 4k$ pixels

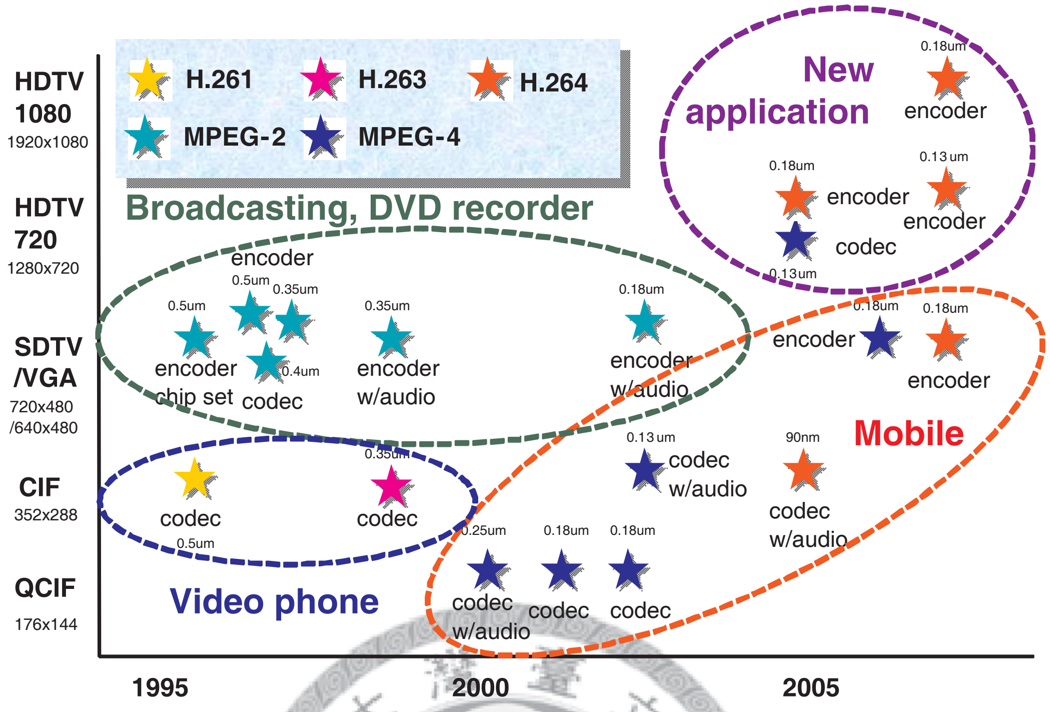


Figure 8.3: Video compression VLSI research trend in international academic conferences (ISSCC and SOVC) since 2005.

is fine enough to match human vision system. Further improvements should come from other dimensions. However, the best video resolution in main stream devices in only 1920 × 1080 [87]. Figure 8.3 shows the Video compression VLSI research trend in international academic conferences (International Solid State Circuit Conference, ISSCC and Symposium on VLSI, SOVC) since 2005. The resolution grows from QCIF to full HD format. It leaves a large room to improve. Table 8.1 shows the features of the state-of-the-art encoder chips. These works progress from 720p, Baseline Profile [78] to recent 1080p, High Profile [93]. However, they are still far from the long-term goal of ultimate $8k \times 4k$ resolution.

There are three challenges to design an efficient MVC encoder chip.

- Encoding high-definition (HD) multiview video requires high processing capability.

- Conventional macroblock (MB) pipelining and scheduling cannot deal with various MVC structures.

- With 3D and quad HDTV specifications, conventional ME architectures re-

Table 8.1: Feature of the-state-of-the-art single-chip encoders

| Prior art<br>(ISSCC05,07,08) | Supported<br>Resolution | Supported<br>Profile | Operating<br>Frequency | MB pipe-<br>lining | Cycles/MB<br>pipeline |
|---|---|---|---|---|---|
| Huang [78] | 1280×720 | Baseline | 108MHz | 4-stage | 1049 |
| Chang [94] | 1280×720 | Baseline | 108MHz | 4-stage | 1049 |
| Lin [93] | 1920×1080 | High | 145MHz | 3-stage | 626 |

quire 2.9Mb on-chip SRAM and 13.8GB/s external memory bandwidth, which is far beyond 6.4GB/s supported by DDR2-800 at 100% utilization.

Above issues cannot be solved effectively by the state-of-the-art H.264/AVC encoder chips [78, 94, 93]. In this chapter, the design and implementation of an MVC encoder chip for 3D/QFHD TV applications is proposed. To deal with high processing requirement and various MVC structures, a new system scheduling scheme, view-parallel macroblock-interleaved (VPMBI) scheme, is proposed. In addition, all the computation core in their corresponding MB pipeline stages are designed according to the VPMBI scheduling. The proposed MVC encoder architecture greatly reduces the high external memory bandwidth and large oh-chip SRAM size. In the meanwhile, the encoding rate of 4096×2160 resolution, 24 frames per second, is achieved.

The remainder of the chapter is organized as follows. Section describes the system architecture and the VPMBI scheduling. Next, the detailed architectures and analysis of critical computation cores are presented from Section to Section 8.5. Section shows the implementation results. Finally, Section 8.7 concludes this chapter.

## 8.2 Proposed View-Parallel Macroblock-Interleaved Scheduling

The system architecture is shown in Fig. 8.4. The encoder contains seven kinds of computation cores for integer ME/DE (IMDE) and fractional ME/DE (FMDE), intra

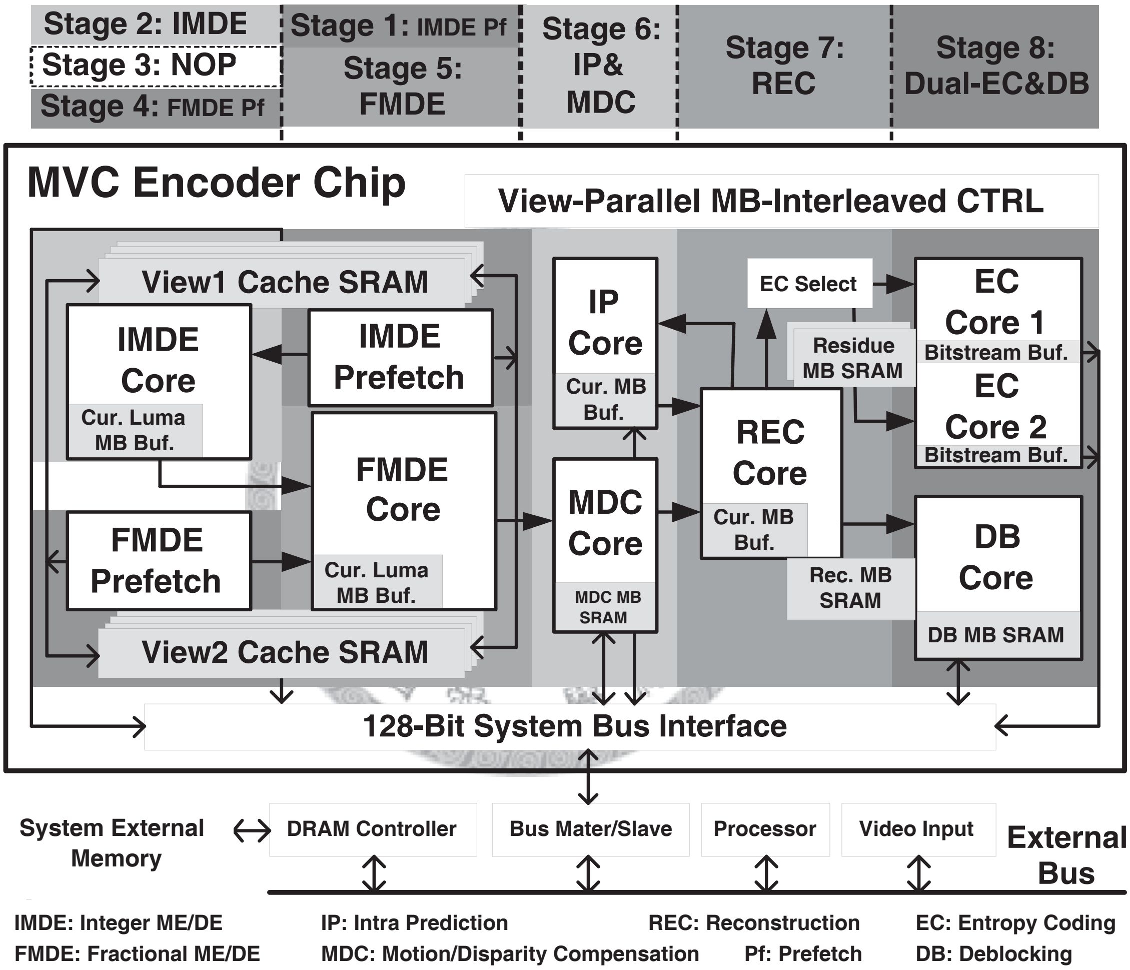


Figure 8.4: Block diagram of the proposed MVC encoder system.

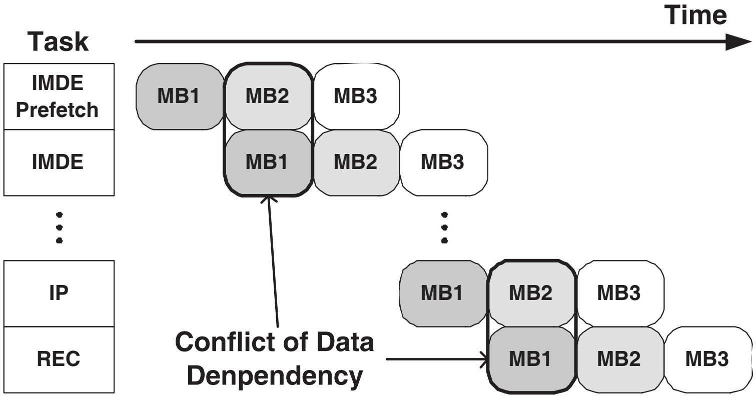


Figure 8.5: Conflict of data dependency occurs in the conventional MB pipelining.

prediction (IP), motion and disparity compensation (MDC), reconstruction (REC), entropy coding (EC), and deblocking filter (DB). There are only 350 cycles in an MB pipeline stage at the highest specification (4096×2160p/24fps/1 view@280MHz), where the conventional three- or four-stage MB pipelining [78, 94, 93] containing 600 to 1000 cycles in a pipeline stage is not feasible. Therefore, the eight-stage MB pipelining is proposed instead of simply raising the degree of parallelism. In the proposed system, the inter-prediction part is split into five MB pipeline stage, and the rest part is split into three ME pipeline stages. The cache-based prediction core introduced in Chapter 6 is adopted as the inter-prediction part. The two prefetch stages for IMDE and FMDE not only reduce the burden of pipeline-cycle budget but also enhance the hardware utilization of IMDE and FMDE. Besides, the propose of NOP stage is introduced later. In the sixth MB pipeline stage, IP and MDC is performed and followed by the REC of an MB in the next pipeline stage. EC and DB are processed simultaneously in the eighth ME pipeline stage. To provide high symbol rate for detailed texture image, EC cores are doubled.

Directly increasing the number of MB pipeline stages causes conflict of data dependency and difficulties of resource sharing between computation cores. There are two critical issues, as shown in Fig. 8.5.

- Before beginning the prefetch stage, the initial guess of motion vectors (MVs) and disparity vectors (DVs) should be derived in advance. If the conventional MB pipelining is applied, IMDE for MB1 and IMDE prefetch for MB2 are performed simultaneously. Conflict of data dependency occurs because MB2

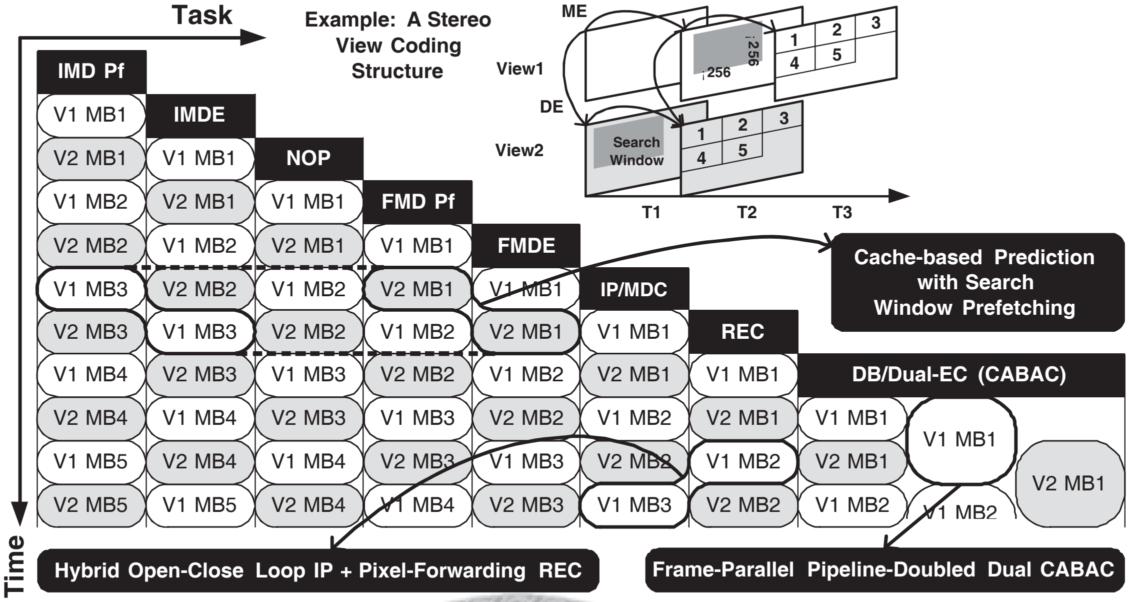

Task
Time
Example: A Stereo View Coding Structure
ME
DE
View1
View2
256
256
Search Window
1
2
3
4
5
T1
T2
T3
IMD Pf
IMDE
NOP
FMD Pf
FMDE
IP/MDC
REC
DB/Dual-EC (CABAC)
V1 MB1
V2 MB1
V1 MB2
V2 MB2
V1 MB3
V2 MB3
V1 MB4
V2 MB4
V1 MB5
V2 MB5
Cache-based Prediction with Search Window Prefetching
Hybrid Open-Close Loop IP + Pixel-Forwarding REC
Frame-Parallel Pipeline-Doubled Dual CABAC

Figure 8.6: Features of view-parallel MB-interleaved scheduling.

requires the MV predictors provided by MB1.

- Another data hazard occurs between the IP and REC pipeline stage. In H.264/AVC standard, if an MB is intra-coded, it is predicted by the reconstructed boundary pixels around each sub-block. Conflict of data dependency occurs when IP and REC are split into two pipeline stage.

Therefore, view-parallel MB-interleaved (VPMBI) scheduling is proposed to overcome the above issues. With the VPMBI scheduling, the proposed system can process nine MBs simultaneously without the above problems, so the throughput is enhanced to support 4096×2160p videos.

Figure 8.6 shows the operation and features of VPMBI scheduling. A stereo view video coding structure is taken for an example. In this case, two views are processed in parallel, and MBs are processed in an interleaving manner. Each capsule unit represents the cycle budget for an MB pipeline. VPMBI is characterized as follows: 1) Cache-based prediction with SW prefetching, which is composed of five pipeline stages, is proposed. SW prediction and prefetching are to lower cache miss rate. The purpose of inserting a bubble pipeline stage is to prevent IMDE and FMDE from fighting for the same cache reading port. 2) Hybrid open-close loop IP and pixel-forwarding REC are decomposed into two pipeline stages without any conflict of data hazard. Reconstructed pixels in neighboring MB boundaries are forwarded to IP and adopted as intra predictors, while intra predictors inside the current MB use original pixels instead of reconstructed pixels. DCT-based rate-distortion optimization (RDO) is also adopted to avoid quality degradation. The throughput of the proposed architecture is 1.8× and 2.7× better than previous works with similar silicon area [78][93]. 3) To achieve the symbol rate of 4096×2160p resolution, EC cores are doubled to perform frame-parallel pipeline-doubled dual (FPPDD) (CABAC). Each EC core encodes two symbols per cycle. The cycle budget of this pipeline stage is doubled, and two EC cores operate in a ping-pong manner to connect with REC stage. FPPDD CABAC provides 3.88 times of symbol rates over direct implementation so that it can meet the real-time requirement for encoding 4096×2160p resolution.

In summary, the proposed VPMBI scheduling with eight-stage MB pipelining provides the benefits listed as follows.

- In IMDE and FMDE stages, MB-interleaved processing effectively provides time budget for prefetching search window. The hardware utilization is also enhanced.

- VPMBI scheduling solves the data hazard in IMDE prefetch/IMDE and IP/REC stages.

- In EC stage, available processing cycles per MB is doubled, so the symbol processing capability is doubled. In addition, other stages remain single computation core and the original throughput for area-efficient consideration.

- Proposed system scheduling is suitable for MVC scheduling. The view scalability is thus achieved.

- Nine MBs are simultaneously processed. In addition, proxessing cycles of the eight stages are balanced to achieve high utilization.

The architecture of each MB pipeline stage is shown in the remainder.

## 8.3 Cache-Based Temporal/Inter-view Prediction Stage

### 8.3.1 Cache Controller Architecture

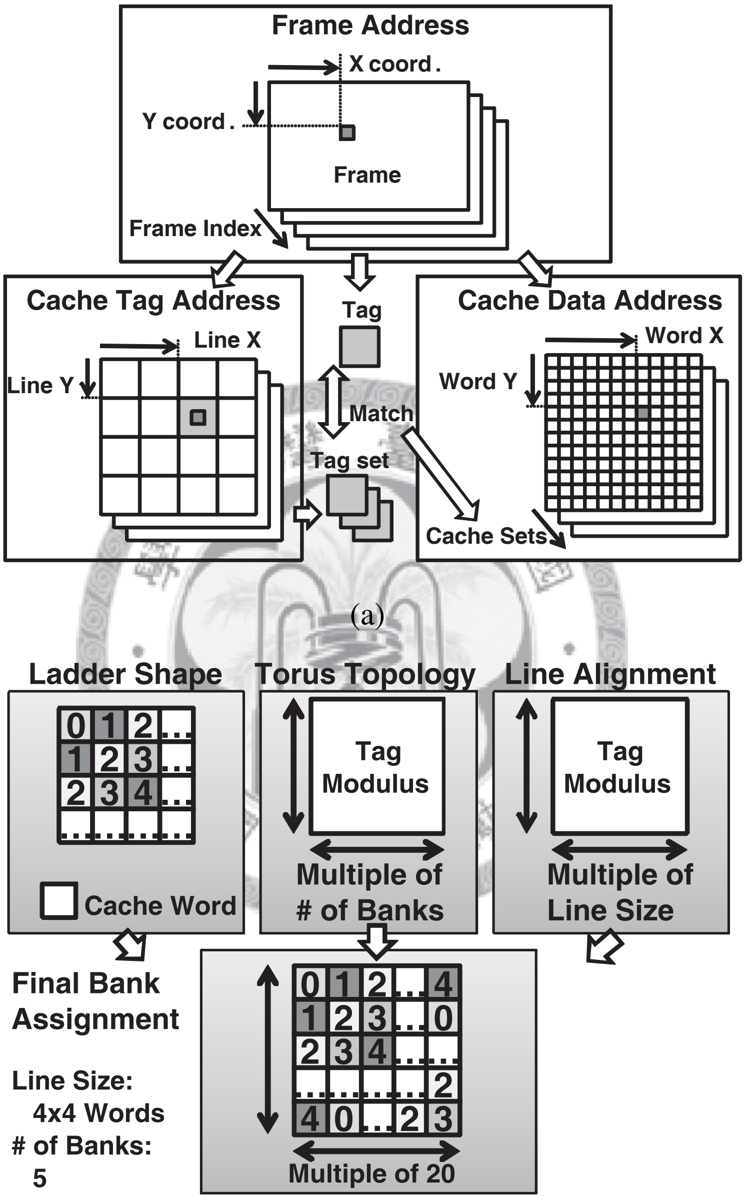
The cache architecture for reference frames replaces traditional SW buffer. For better locality, the internal addressing in the cache keeps the intrinsic 2D nature of frames. The address-resolving flow and bank assignment is shown in Fig. 8.7, which is introduced in Chapter 6 and reviewed here. The 3-tuple vector (x, y, frame-index) is translated to the tag address and the tag. A tag-set is located by the tag address, and the tag is compared to that set. Upon cache-hit, the word address locates the word in a 5-banked on-chip SRAM. The bank assignment is determined by the 3 constraints shown in the figure. In this 4-way non-blocking architecture, the control logic supports reading after up to 6 misses and concurrent reading and prefetching/locking.

Figure 8.8 shows the hardware architecture of cache controller. To meet the throughput of the prediction core, the proposed architecture supports sustained rate of matching 4 cache lines, reading 5 words, and refilling 4 words per cycle without cache line split penalty. With IMDE and FMDE prefetching, the penalty of cache miss is reduced by 93% shown in Fig. 8.9 . Therefore, the length of pipeline stage is shorter than 350 cycles.

### 8.3.2 Predictor-Centered Algorithm and Architecture for IMDE Stage

In IMDE stage, the predictor-centered fast ME/DE algorithm is used. Figure 8.10 explains the proposed algorithm. First, several predictors are classified into intra-frame and inter-frame predictor, including the $16 \times 16$ MVs of the left, top-left, top, and top-right MBs. They are from highly correlated sources of MVs like neighboring and best matching MBs. These MV predictors are set as the refining centers and evaluated by sum of absolute difference (SAD) cost. Then a $\pm 16 \times \pm 16$ searching range is used around the best predictor. The computation of the proposed algorithm is three orders lower than that of full search and 95% less than that of hierarchical search, as shown in Fig. 8.11

The Architecture of IMDE is shown in Fig. 8.12. The most different part from



(a)

(b)

Figure 8.7: (a) Address-resolving flow (b) Bank assignment.

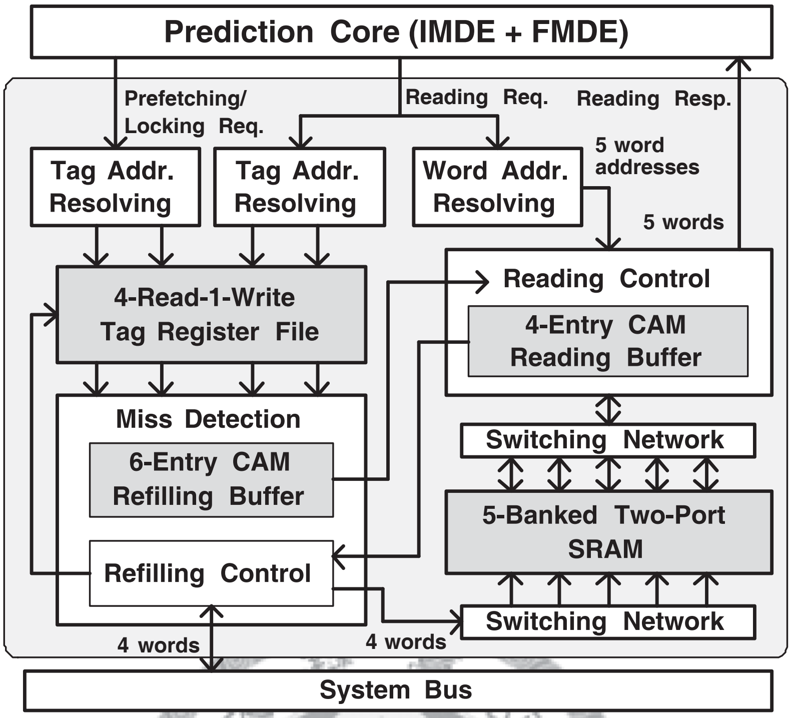


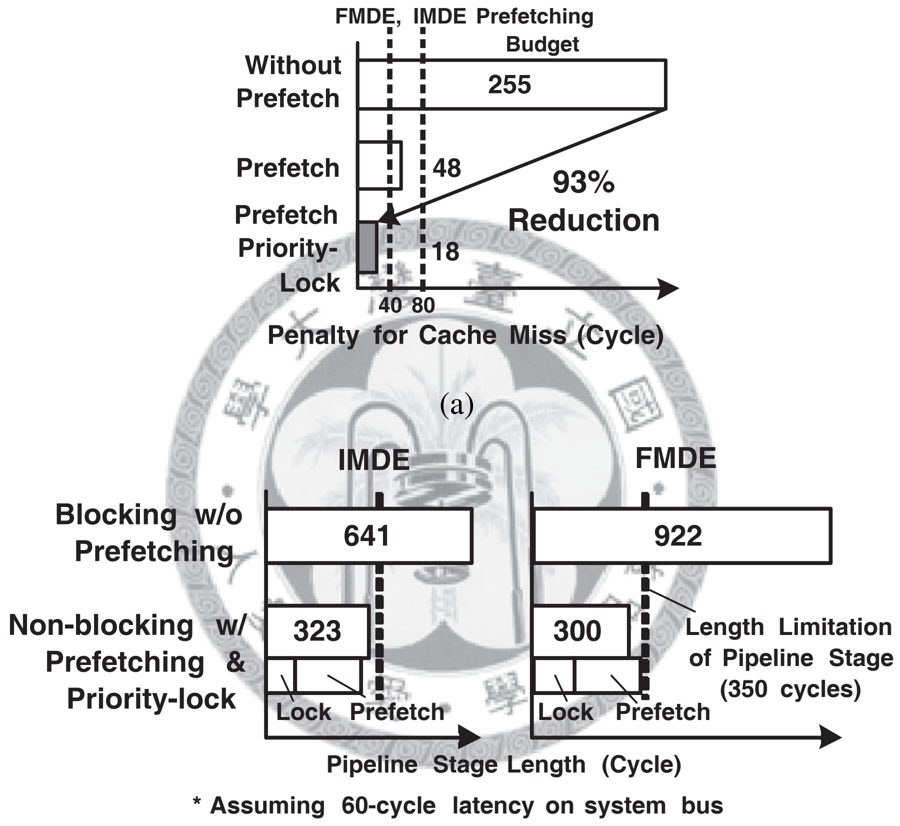
Figure 8.8: Hardware architecture of the cache controller.

previous ME architecture is “refining range speculation” and “data prefetching and locking.” These two parts support predictor-centered fast ME/DE algorithm and effectively reduce on-chip memory requirement. In addition, to support variable-block-size (VBS) ME/DE, reconfigurable processing element (PE) array are applied to provide various computing throughput in the multiple-hints stage and the refining stage. Sixteen reconfigurable 256-PE array compute sixteen search candidates per cycle. Therefore, the overlapped pixels between 4 × 4 search candidates are fully reused, the on-chip memory access is thus minimized.

### 8.3.3 Algorithm and Architecture Optimization for FMDE Stage

#### Bandwidth Reduction of FMDE Algorithm

In H.264/AVC, the precision of MV/DV can be down to a quarter pixel. The algorithm of sub-pixel interpolation for motion compensation is defined in the coding standard. However, the interpolation scheme is a encoder issue which depends on the designer. In H.264/AVC, the half-pixel interpolation is done by a 6-tap filter with coefficients $\frac{1,\ -5,\ 20,\ 20,\ -5,\ 1}{32}$. With wider FIR filter, the supporting region grows. If



(a)

(b)

Figure 8.9: (a) Cycle reduction of cache miss penalty and (b) Reduction of pipeline cycles after applying proposed prefetching scheme.

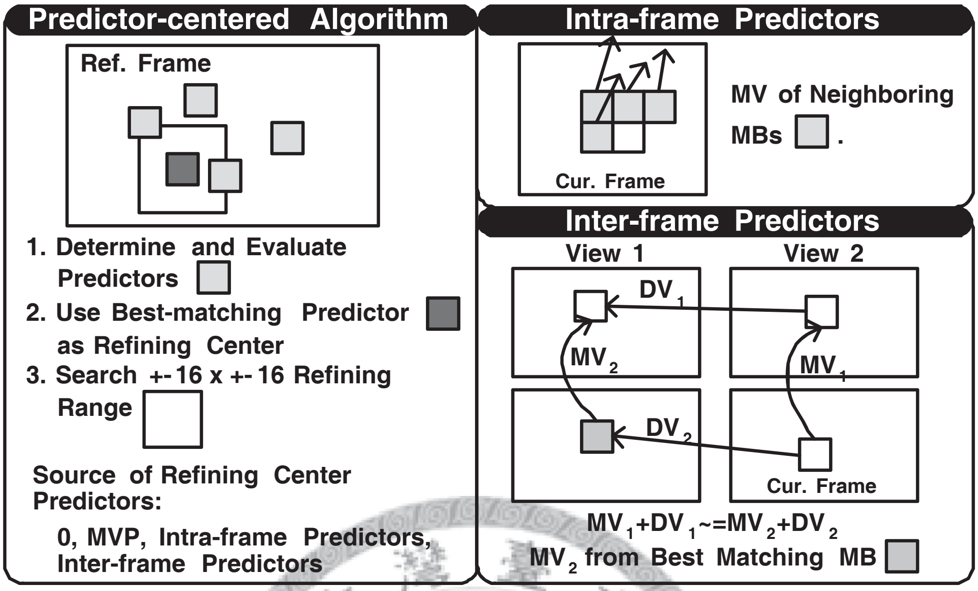

Predictor-centered Algorithm
Ref. Frame
1. Determine and Evaluate Predictors
2. Use Best-matching Predictor as Refining Center
3. Search +-16 x +-16 Refining Range
Source of Refining Center Predictors:
0, MVP, Intra-frame Predictors, Inter-frame Predictors
Intra-frame Predictors
Cur. Frame
MV of Neighboring MBs .
Inter-frame Predictors
View 1
View 2
DV1
MV2
MV1
DV2
Cur. Frame
MV1+DV1~=MV2+DV2
MV2 from Best Matching MB

Figure 8.10: Predictor-centered algorithm and illustrations of intra-frame and inter-frame predictors, respectively.

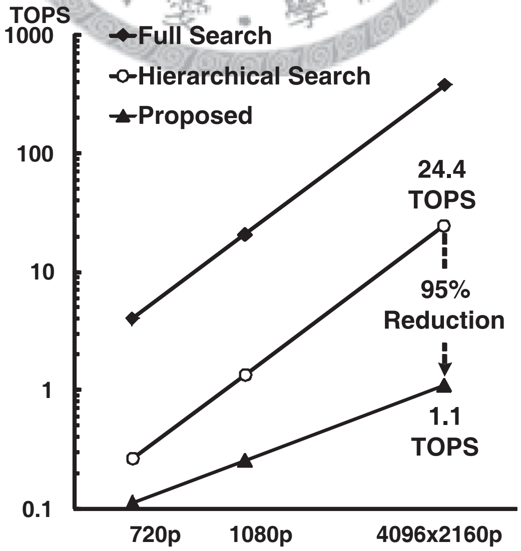


Figure 8.11: Computation reduction of the proposed predictor-centered algorithm.

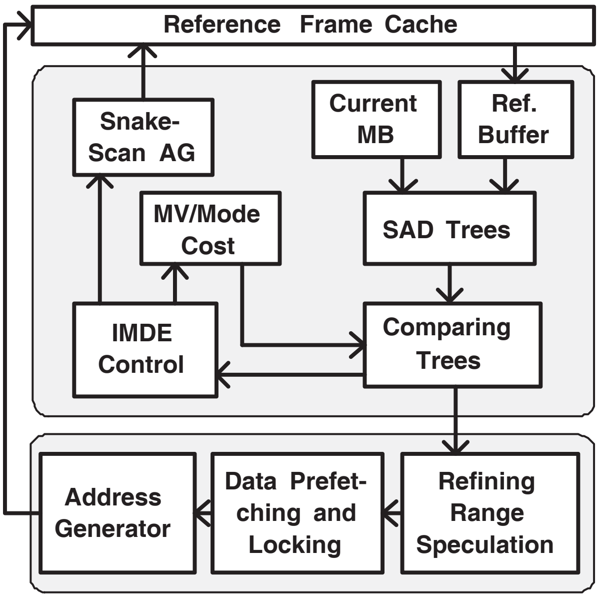


Figure 8.12: Architecture of IMDE computation core.

we replace this 6-tap filter by a 2-tap filter with coefficients $\frac{1, 1}{2}$, then the supporting region for an 8×8 block reduces from 14×14 to 10×10 as shown in in Fig. 8.13. With this simplification, the area of supporting region decreases for 31% for a 8×8 block, thus reduces the memory bandwidth to the reference frame buffer and external memory.

The experimental result is shown from Fig. 8.14 to Fig. 8.16, where Fig. 8.14 and Fig. 8.15 are the PSNR drop comparison of all the sequences, and Fig. 8.16 is the rate-distortion curve of a single sequence. Four different FME algorithms are compared in the experiments. Label ”6-Tap” is the one with original interpolation algorithm, and this is used in the reference software. ”2-Tap” is the proposed algorithm with simplified interpolation. These two are with $\left[-\frac{3}{4}, \frac{3}{4}\right] \times \left[-\frac{3}{4}, \frac{3}{4}\right]$ search range. Label ”ISSCC 08” is the FME algorithm used in [87], and ”No FME” just disables FME and uses the results from IME directly. Compared with the algorithm used in the reference software, the average PSNR degradation of the other three algorithms is shown in Table 8.2. From the result, the quality drop caused by the proposed algorithm is far less than ”ISSCC 08”.

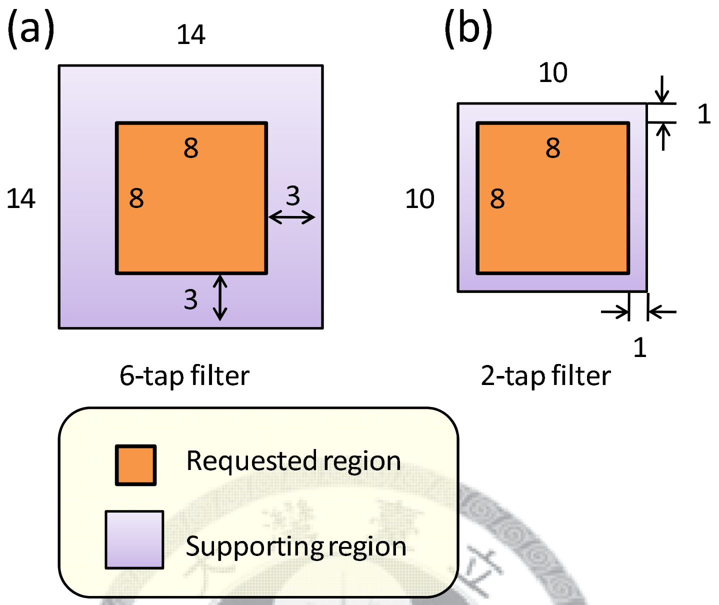


Figure 8.13: Comparison of required data due to different half-pixel interpolation algorithms

Table 8.2: Average PSNR Drop (dB) of Different FME Algorithms Comparied With 6-Tap Filter

| Coding Parameters | Proposed (Bilinear) | Lin (*ISSCC*08) | Without FME |
|---|---|---|---|
| 720p, Qp=20 | 0.029 | 0.12 | 0.70 |
| 720p, Qp=30 | 0.034 | 0.28 | 1.25 |
| 1080p, Qp=20 | 0.026 | 0.10 | 0.45 |
| 1080p, Qp=30 | 0.024 | 0.27 | 0.81 |

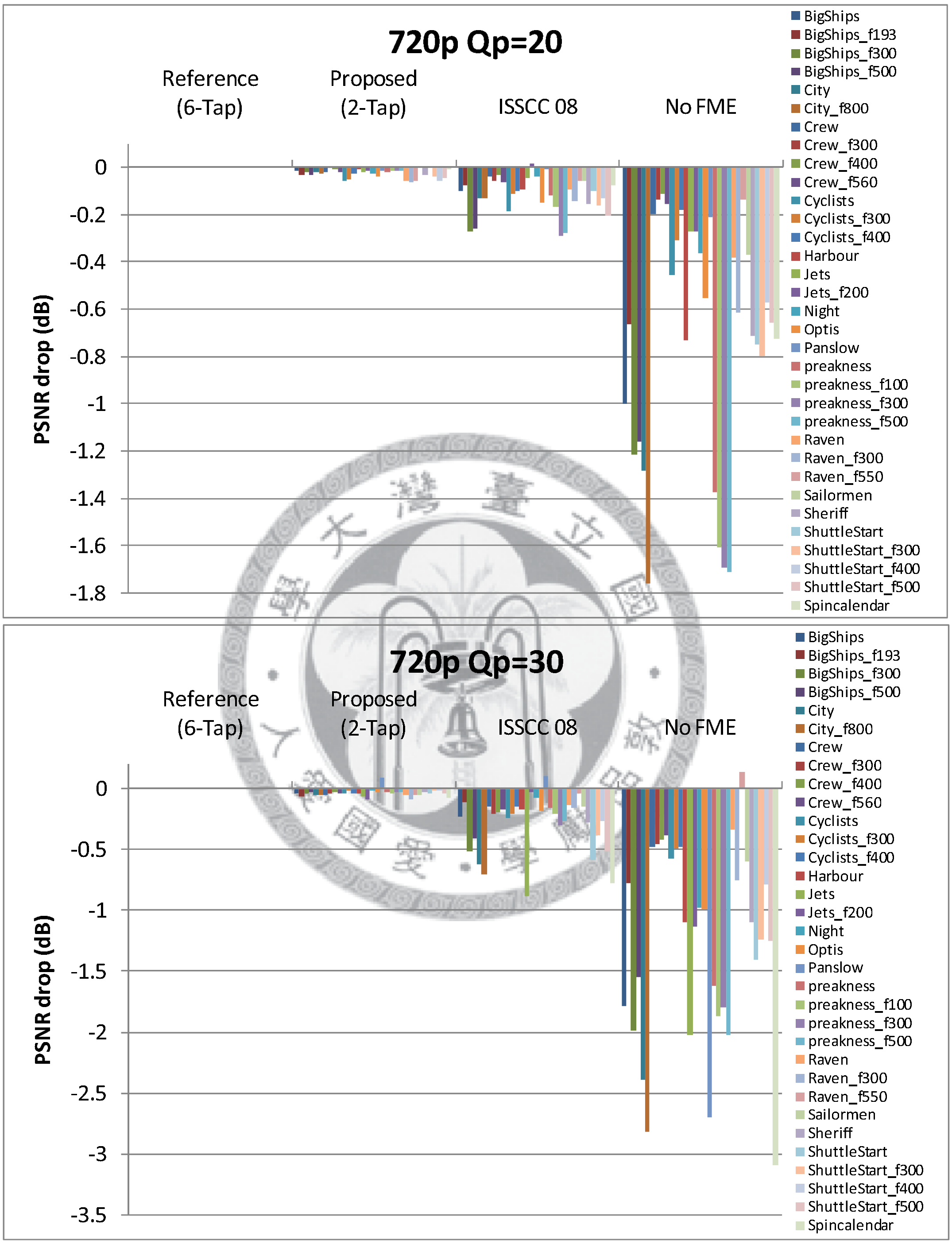


Figure 8.14: PSNR drop comparison of different FME algorithms (all 720p sequences with Qp=20 and 30)

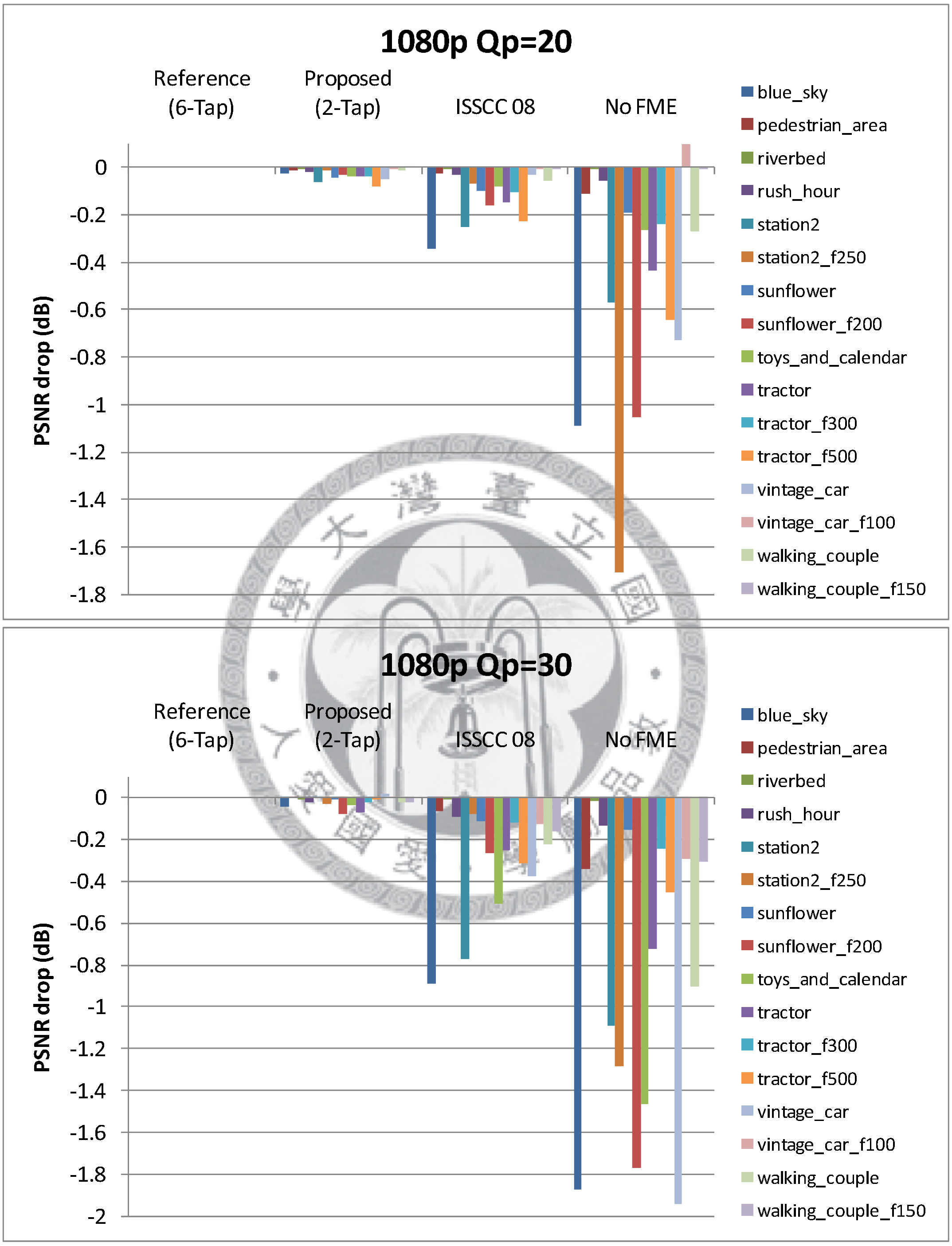


Figure 8.15: PSNR drop comparison of different FME algorithms (all 1080p sequences with Qp=20 and 30)

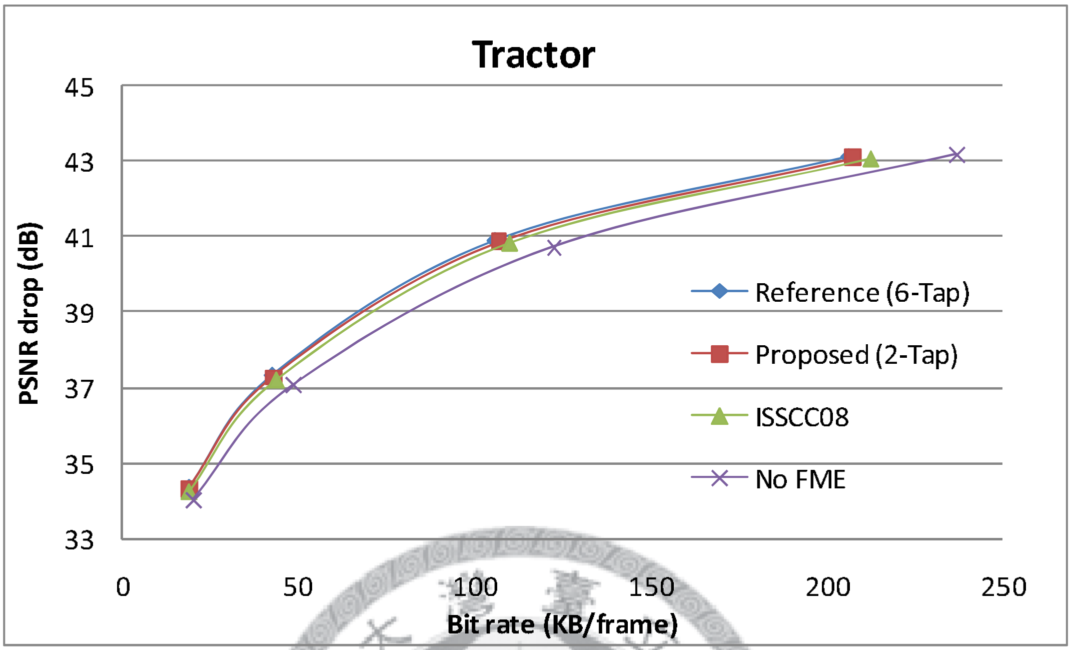


Figure 8.16: Rate-distortion comparison of different FME algorithms (Tractor)

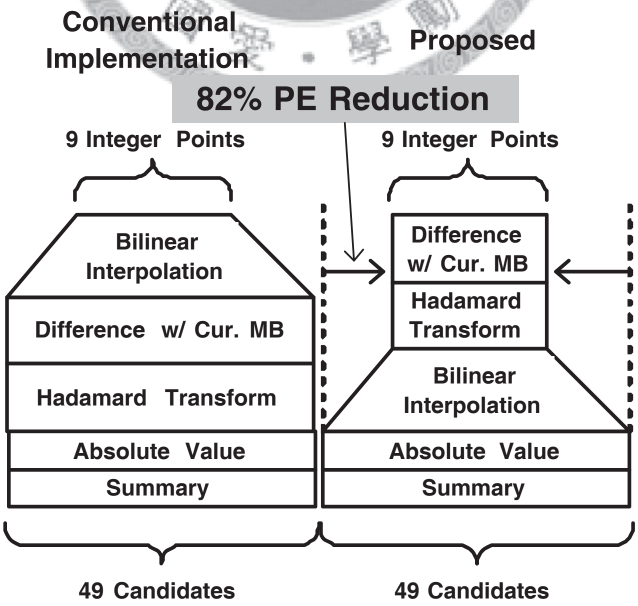


Figure 8.17: Datapath optimization of FMDE.

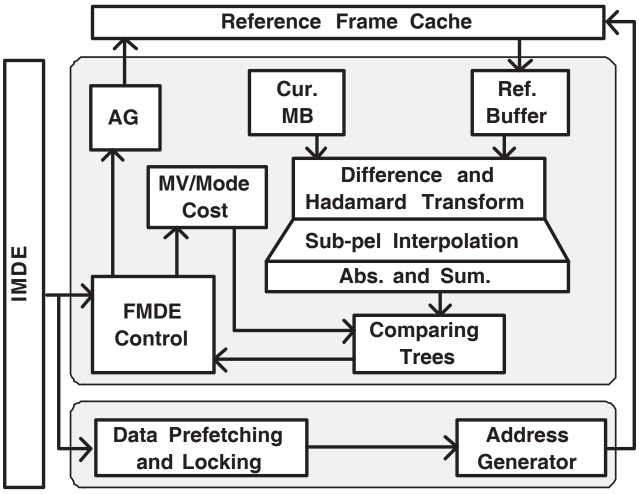


Figure 8.18: Architecture of FMDE computation core.

### Architecture Optimization of FMDE Core

In FMDE stage, the bilinear filter rather than the 6-tap filter with 49× parallelism is used for half-pixel interpolation, which reduces the bandwidth by 51%. In addition, since all the non-integer locations are bilinear-interpolated, this linearity helps architectural simplification. The simplification in interpolation can further lead to simpler hardware architecture. In FMDE, the common cost metric for residue is sum of absolute transformed difference (SATD), and the transformation is Hadamard transform. Since the Hadamard transform is linear, it can be factored out and save some hardware resource. In the original algorithm, half pixels are 6-tap filtered, and the quarter pixels are linearly interpolated, so the Hadamard factorization can only apply to the quarter pixels. When the half pixels are 2-tap filtered, we can use bilinear interpolation to get all the non-integer pixels, so that the Hadamard factorization can be applied to all the non-integer pixels. Figure 8.17 shows the original and optimized datapath of FMDE. With data-flow rescheduling, 82% area of the transformation and difference circuit are saved. The corresponding FMDE architecture is illustrated in Fig. 8.18.

## 8.4 Hybrid Open-Closed Loop Intra Prediction and Pixel-Forwarding Reconstruction Stage

### 8.4.1 Design Challenges

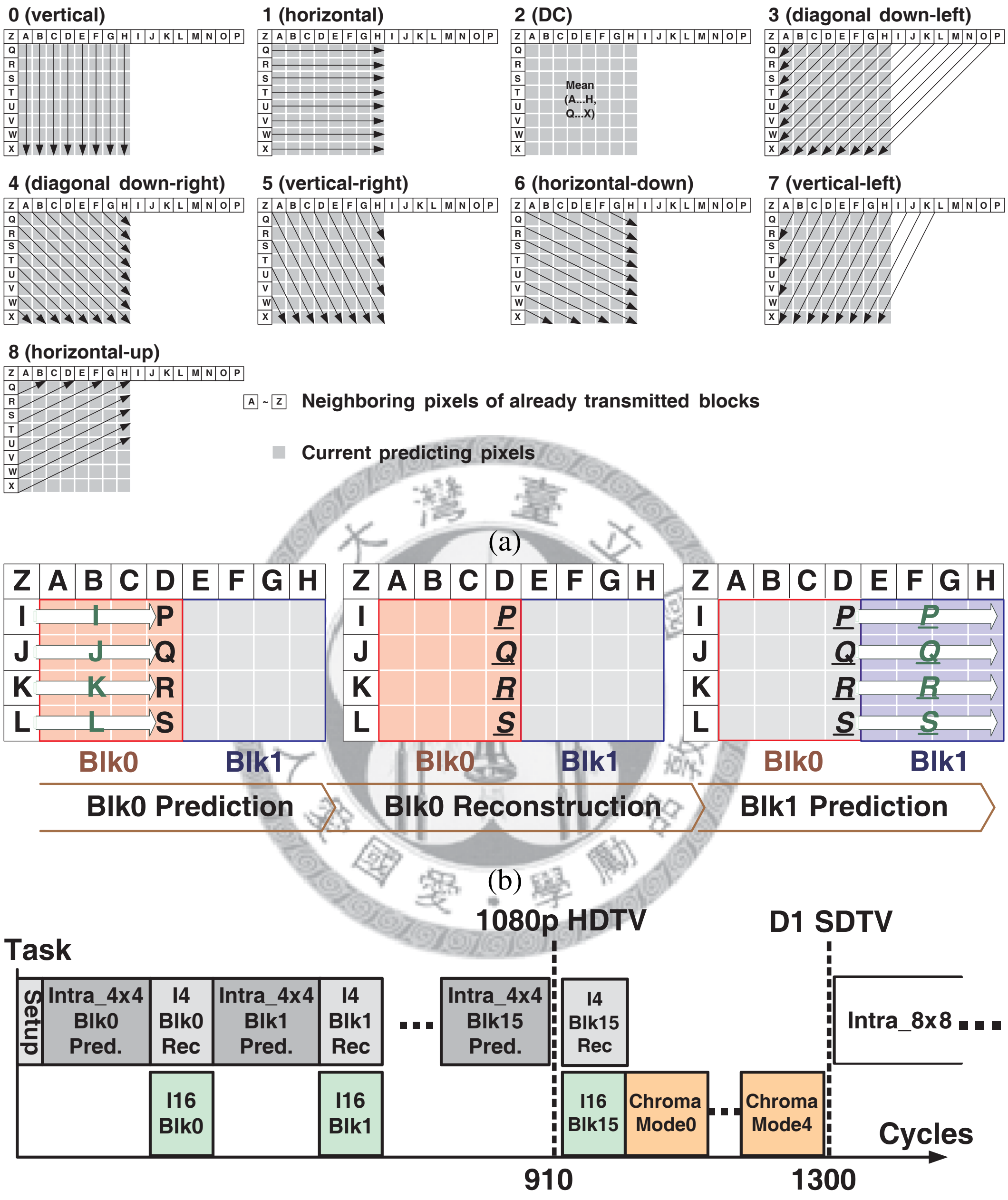
In previous H.264/AVC designs [95, 96, 97, 98], intra prediction for baseline and main profile are well-developed for D1 and HD 720p specification. However, there are two main design challenges which lower the efficiency of above designs after Intra_8×8 of high profile being taken into consideration.

The first issues comes from the data dependency of intra prediction between each sub-block. Based on H.264/AVC standard definition for Intra_4×4 and Intra_8×8 modes, each sub-block should be processed by the zig-zag scan order, and the 13 or 25 reconstructed pixels are required for prediction as shown in Fig. 8.19 (a). Since the reconstructed pixels can be only available until the neighboring blocks are predicted and reconstructed, each sub-block should be processed sequentially, as shown in Fig. 8.19 (b). Figure 8.19 (c) illustrates the corresponding hardware processing scheduling of Fig. 8.19 (b). In Suh's [97] and Ku's [96] design take about 1000 cycles to process one MB's intra prediction to meet HD 720p specification without Intra_8×8. However, the above designs will require much higher operating frequency when Intra_8×8 of high profile is taken into consideration.

The other issue is the throughput and hardware utilization. For Intra_4×4 and Intra_16×16 mode, four pixel parallelism is usually adopted [95, 96, 98]. But for Intra_8×8 mode, eight pixel parallelism should be applied due to 8 × 8 transform. The mismatch throughput of different intra modes should be unified for intra predictor generator, transform, and reconstruction to improve the hardware utilization and processing capability.

### 8.4.2 Hardware-Oriented Hybrid Open-Closed Loop Intra Prediction

In order to improve the processing parallelism limited by data dependency in Sec. 8.4.2, proposed hybrid open-closed loop intra prediction scheme use original pixels

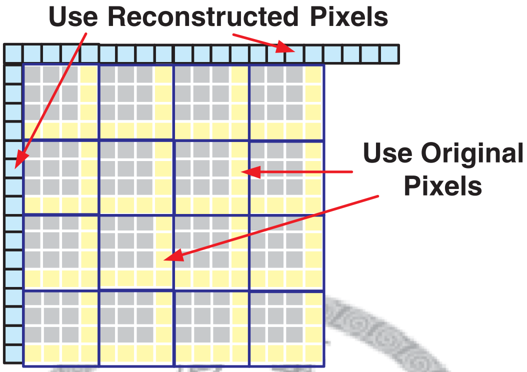


(a)

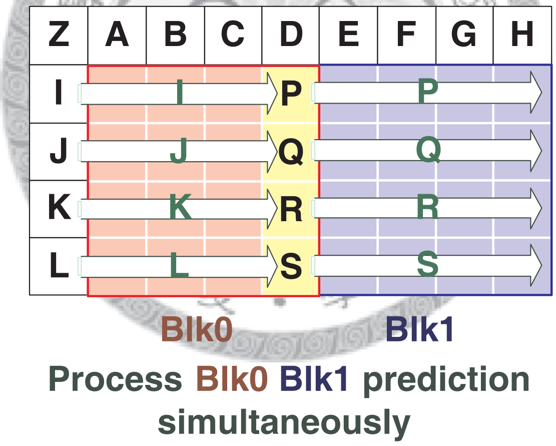
(b)

(c)

Figure 8.19: Illustration of design challenges of IP. (a) Nine 8×8 luma prediction modes. (b) Data dependency of IP and REC between neighboring sub-blocks. (c) Conventional hardware processing schedule of IP and REC.



(a)



(b)

Figure 8.20: Illustration of hybrid open-closed loop intra prediction. (a) Hybrid open-closed loop prediction of boundary pixels. (b) Proposed corresponding processing scheduling of the hybrid open-closed loop intra prediction.

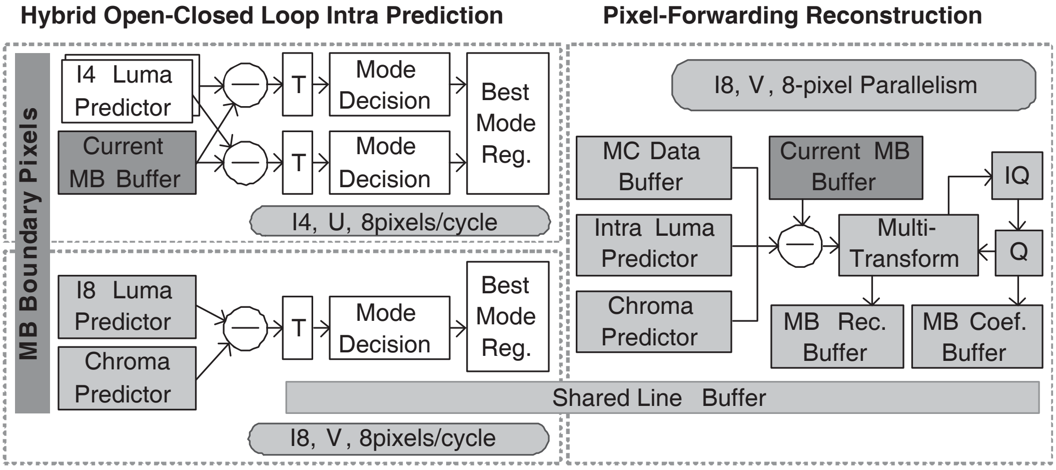


Figure 8.21: Architecture of IP and REC computation cores.

instead of reconstructed pixels as boundary pixels for intra predictors, as shown in Fig. 8.20 (a). This is because that original pixels are close to reconstructed pixels in our target high definition application which PSNR is always greater than 35dB. Thanks to VPMBI scheduling, the reconstructed MB boundary pixels can be derived and adopted as intra predictors from the neighboring MBs before IP of the current MB starts. Take Intra_4x4 as an example, when block 1 in Fig. 8.20 (b) is processing, it uses the four original pixels (yellow pixels) as its left boundary pixels and the nine reconstructed pixels from upper row as upper boundary pixels. The proposed hybrid open-closed loop scheme has very slight quality degradation comparing to closed-loop DCT-base intra prediction and is still better than JM 9.5. By proposed open-loop scheme, the intra prediction of each sub block can be predicted in parallel without waiting neighboring blocks' reconstruction loop.

### 8.4.3 Architecture of Hybrid Open-Closed Loop Intra Prediction and Pixel-Forwarding Reconstruction

The architecture of hybrid Open-closed loop intra prediction and pixel-forwarding reconstruction is shown in Fig. 8.21. In order to be consistent with the throughput of 8×8 DCT in Intra_8×8 prediction, the parallelism of our architecture is set to be 8-pixel parallel. Since the Intra_8×8 prediction mode is similar to Intra_4×4 prediction, a reconfigurable intra luma predictor generator is proposed that can generate

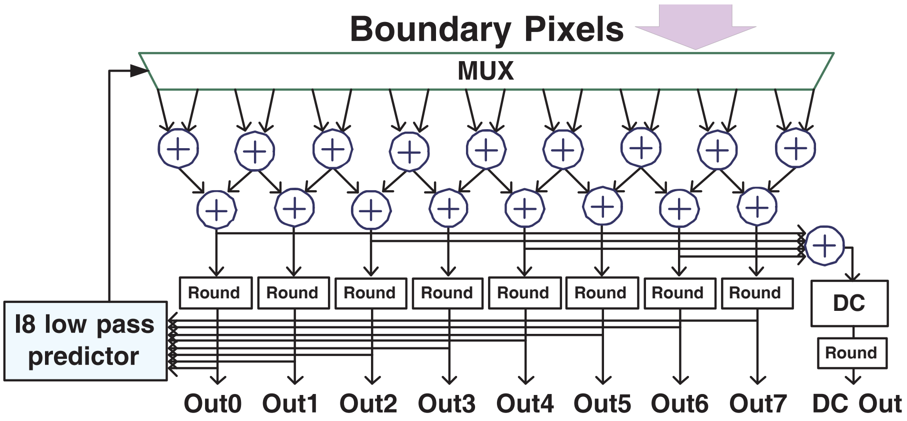


Figure 8.22: Architecture of intra luma predictor.

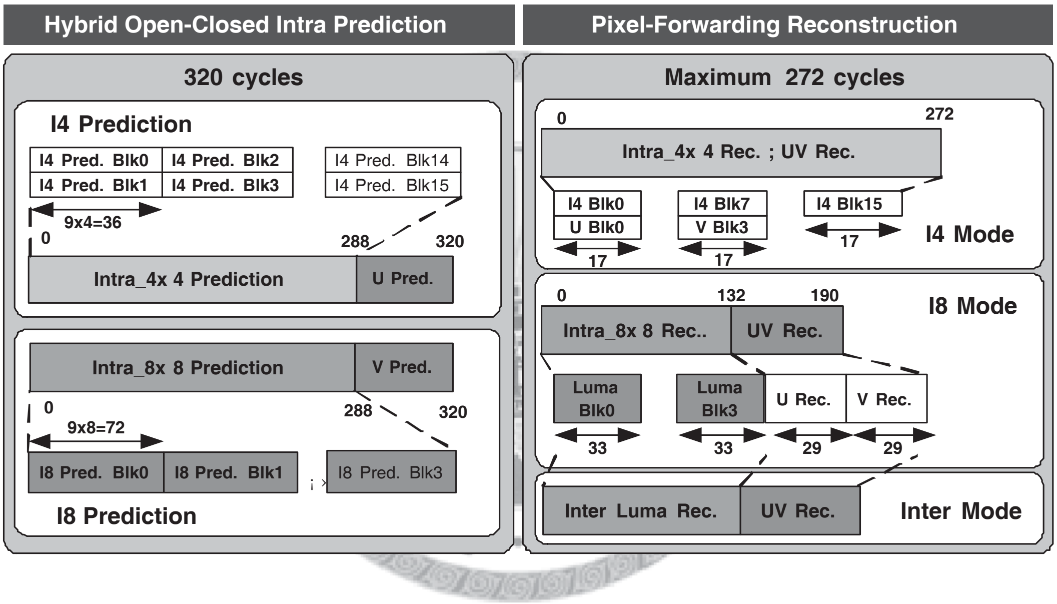


Figure 8.23: Schedule of IP and REC computation cores.

eight predictors for Intra_8×8 mode, or eight predictors for two 4×4 sub blocks for Intra_4×4 mode as shown in Fig. 8.22 Besides, the multi-transform can be configured as two 4×4 Hadamard/DCT /IDCT or one 8×8 DCT/IDCT transform for cost estimation and reconstruction. The proposed hardware architecture can unified throughput and improve processing capability with excellent area efficiency by using these reconfigurable 8-pixel parallel PEs.

Unlike previous prediction-reconstruction interleaved scheme [95, 96, 97], the schedule of proposed architecture can be divided into two MB pipeline stages, open-loop prediction and closed-loop reconstruction as shown in Fig. 8.23. In prediction stage, the proposed architecture can process two 4×4 sub blocks in parallel in In-

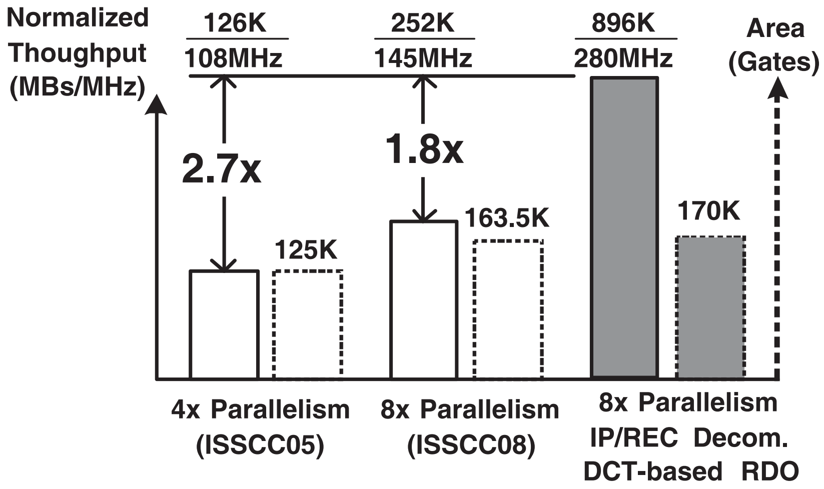


Figure 8.24: Comparison of throughput and silicon area with the previous works.

tra_4×4 mode and can process next two sub blocks immediately without reconstruction because of open-loop prediction. In this stage, only the best mode for each sub block and total MB mode cost are stored. In reconstruction stage, only one mode is selected for reconstruction. If Intra_4×4 mode is selected, luma 4×4 block and chroma 4×4 block will be reconstructed in parallel for higher hardware utilization as shown in Fig. 8.23. It is because in H.264 decoding process, each 4×4 luma sub block should be reconstructed in the zig-zag scan order. Once Intra_8×8 mode or inter mode is chosen, it will process only one 8×8 luma sub block at a time. The chroma reconstruction will be executed after four 8×8 luma blocks are done. This architecture can make 8-pixel-parallelism PEs to achieve almost 100% hardware utilization and save operating cycles from useless reconstruction. It only takes less than 272 cycles to process one MB.

Figure 8.24 shows the comparison of throughput and silicon gate count with previous IP and REC architectures [78][93]. In [78], IP and REC are placed in one pipeline stage, and they process MBs in an interleaved manner, shown as Fig. 8.19 (c). In [93], REC engine is placed between the IP stage and EC stage. The scheduling is well-organized so that the reconstruction of intra predictors and generation of MB residues can generated without the conflict and stalling cycles. The proposed architecture can process 896K MBs at 280MHz, the highest operating frequency. The normalized throughput is also 1.8× and 2.7× better than the previous works.

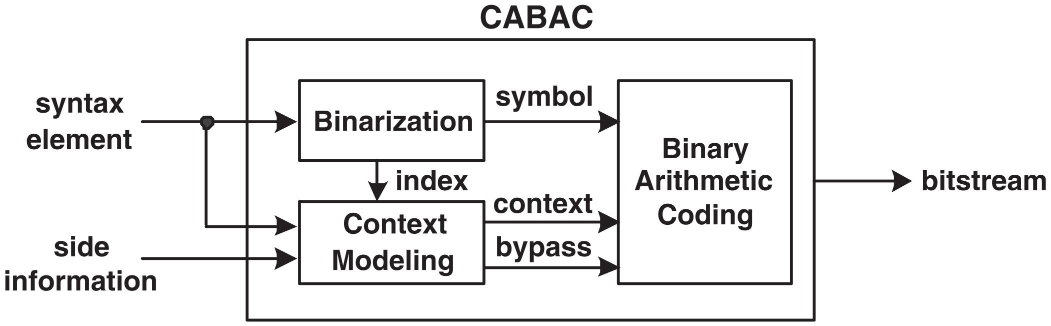


Figure 8.25: Block diagram of CABAC.

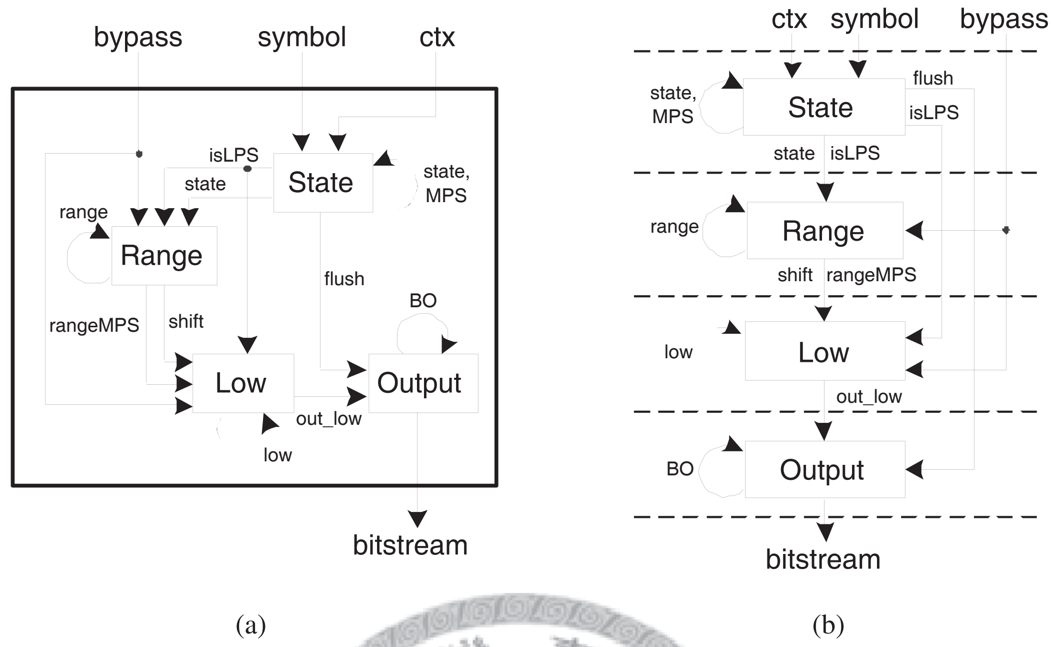
The main reason of this achievement is the separation of IP and REC into two MB pipeline stages makes the system processing throughput higher. In addition, DCT-based rate-distortion optimization (RDO) rather than SAD-based RDO maintains the video quality. Near 0dB quality loss of PSNR is achieved. The logic gate count of the proposed architecture is similar to [93] because of the benefit of reconfigurable architecture of intra predictor and reconstruction engine.

## 8.5 Frame-Parallel Pipeline-Doubled Dual Context-Based Adaptive Binary Arithmetic Coding Stage

### 8.5.1 Introduction to Context-Based Adaptive Binary Arithmetic Coding (CABAC): Algorithm and Architecture

Entropy coding is to compress data based on their probability distribution. It plays an important role in video coding. In baseline profile, H.264/AVC adopts Context-Based Adaptive Variable Length Coding (CAVLC) as entropy coding. In main or more advanced profile, Context-Based Adaptive Binary Arithmetic Coding (CABAC) is adopted. CABAC achieves 9% to 14% bit-rate savings over CAVLC [99], but its computation is much more complicated. Furthermore, due to the sequential nature of arithmetic coding, the hardware design is extremely difficult to exploit pipelining or parallel techniques.

Figure 8.25 shows the block diagram of H.264 CABAC. The inputs of CABAC are syntax elements (SE) and side information. Syntax elements are the essential



(a) (b)

Figure 8.26: Data flow of one-symbol arithmetic coder. (a) The functional block of one-symbol arithmetic encoding. (b) The Optimized four-stage pipelined one-symbol arithmetic coder.

data to be coded, such as MB type, prediction mode, residues, etc. Side information, usually the information of neighboring coded blocks, helps estimate the probability of symbol. These SEs must be transformed into binary symbols before entering binary arithmetic encoder. The adaptive effect is achieved through the context (ctx) assigned to symbol. These ctxs are modeled according to SE type, side information and the binary index. Symbols with the same ctx have similar statistic property and use the same adaptive probability state for estimation. Besides normal arithmetic coding, bypass mode is introduced to speed up the encoding process. Then, symbol along with its associated ctx and bypass flag enter binary arithmetic coder. Finally, arithmetic coder generates output bitstream.

The data flow of one-symbol arithmetic coder is shown in Fig. 8.26. one-symbol arithmetic encoder, which can encode a symbol per cycle, is optimized for short critical path. Among these functional blocks, arithmetic coder is the most critical part. Due to the data dependency of continuous symbols, the critical path of the arithmetic encoder is increased. By applying four-stage pipelining, processes without data de-

Table 8.3: Statistics of symbol count per MB of sequences with 3840×2160 resolution.

| Sequence | Maximum Count<br>(symbols/MB) | Average Count<br>(symbols/MB in 1 slice) |
|---|---|---|
| Blue_sky | 1051 | ≤300 |
| Pedestrain_area | 1140 | ≤200 |
| Riverbed | 923 | ≤300 |
| Rush_hour | 857 | ≤200 |
| Station2 | 1191 | ≤250 |
| Toys_and_calender | 1465 | ≤400 |
| Tractor | 1003 | ≤300 |
| Vintage_car | 1357 | ≤500 |
| Walking_couple | 1769 | ≤400 |

pendency can be parallel computed. Table-dividing and data pre-computing schemes make module critical paths more balanced. Then, multi-symbol architecture is developed based on one-symbol design. By solving the problem of multiple contexts, our design can be extended to encode arbitrary number of symbols in one cycle.

### 8.5.2 Analysis of CABAC Symbol Rate

Due to limited cycles of an MB pipeline, an EC engine with one-symbol arithmetic encoder can only process about 300 symbols in MB pipeline stage. Before defining the target throughput of CABAC, the analysis of CABAC symbol rate is analyzed. The group-of-picture (GoP) structure is $\{$I, P, $B_1$, $B_2$, $B_1$, P, $B_1$, $B_2$, $B_1\}$. $B_2$ is the hierarchical B frame which is predicted by $B_1$. Symbol count per MB of sequences with 3840×2160 resolution is analyzed and shown in Table 8.3. The maximum symbol count of an MB are distributed from 857 to 1769 symbols. The average symbol count in all of the test cases are fewer than 500 symbols. Figure 8.27 shows the statistics of symbol count from 1st frame to 20th frame. In the case of sequence “Walking_couple,” the distribution highly depends on the slice type. The symbol

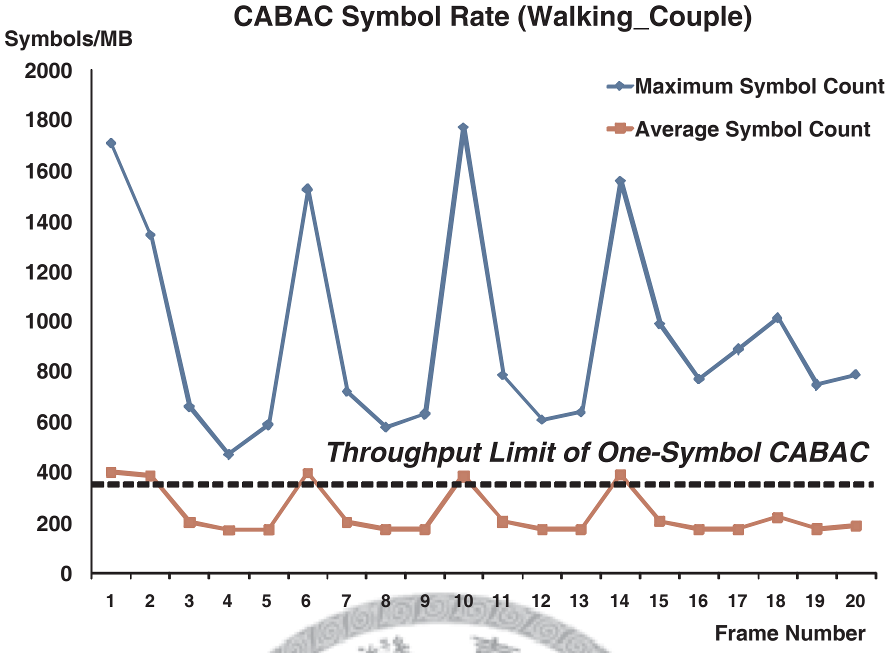


Figure 8.27: CABAC symbol rate of sequence "Walking_couple."

count of I-frame and P-frame are much higher than that of B-frame. The conventional one-symbol architecture is not feasible because it cannot meet the throughput requirement of maximum symbol rate. If the EC engine cannot deal with all the symbols in the limited cycles, the computation cores in other pipeline stages are stalled and the utilization degrades. Therefore, multi-symbol CABAC architecture is necessary.

### 8.5.3 Frame-Parallel Pipeline-Doubled Dual CABAC Architecture

According to the analysis in the Section 8.5.2, the architecture of one-symbol arithmetic encoder to two-symbol architecture, as shown in Fig. 8.28 (a). For *Range Stage*, *Low Stage* and *Output Stage*, two one-symbol PEs are directly cascaded, with *range*, *low* and *BO* as interconnections, respectively. However, we cannot simply cascade two one-symbol *State Stage*s because they are possibly the same. For example, if ctx1 and ctx2 are the same, *state* and *MPS* of ctx2 should be replaced by the updated ones of ctx1. Besides, only the updated values of ctx2 should be written back to Ctx State registers. Fig. 8.28 (b) shows the architecture of two-symbol *State*

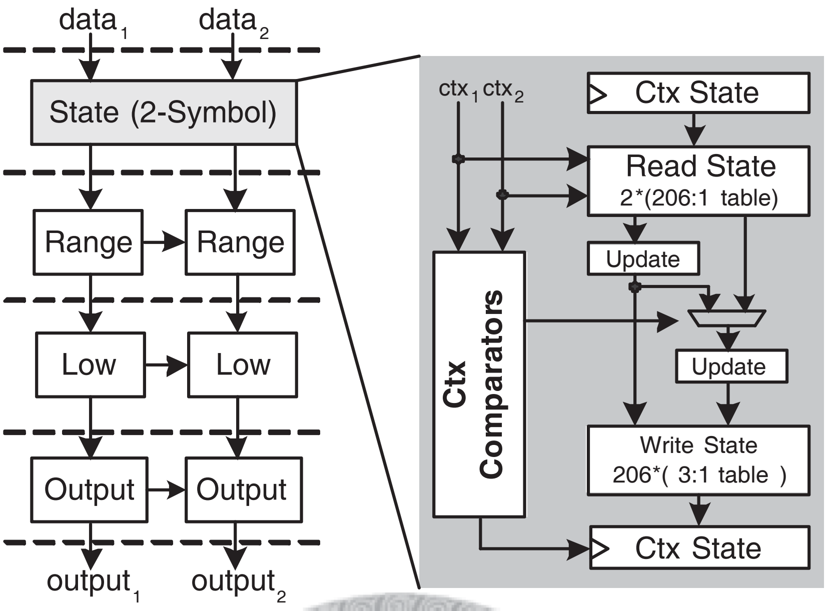


Figure 8.28: Architecture of the two-symbol arithmetic coder. The data dependency between all two symbols are solved by data forwarding.

*Stage*. In one-symbol case, *State Stage* is obviously the slowest stage. The situation becomes better in two-symbol case. Note that, the 206:1 table lookups of two ctxs are parallel-processed, with the same time as in one-symbol case.

Applying two-symbol CABAC architecture can enhance the throughput to doubled. However, for some textured MBs, two-symbol CABAC architecture still does not meet the throughput requirement. Therefore, frame-parallel pipeline-doubled dual (FPPDD) CABAC is proposed. Figure 8.29 shows the MB pipeline scheduling

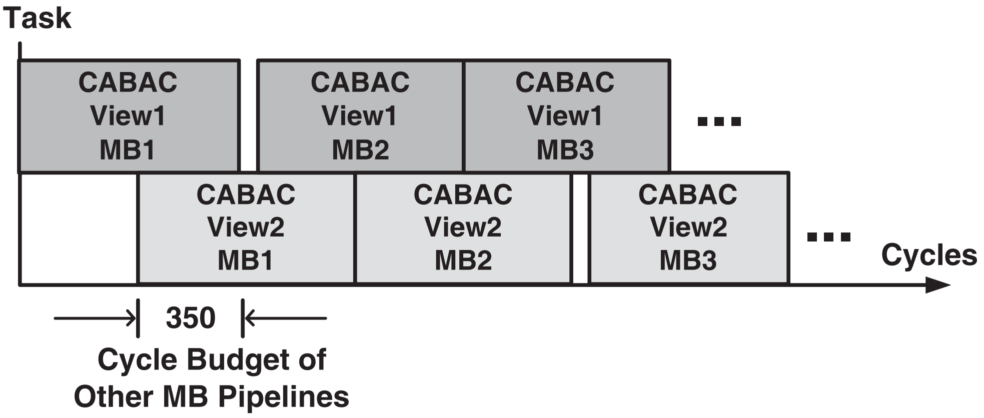


Figure 8.29: MB pipeline scheduling of FPPDD CABAC.

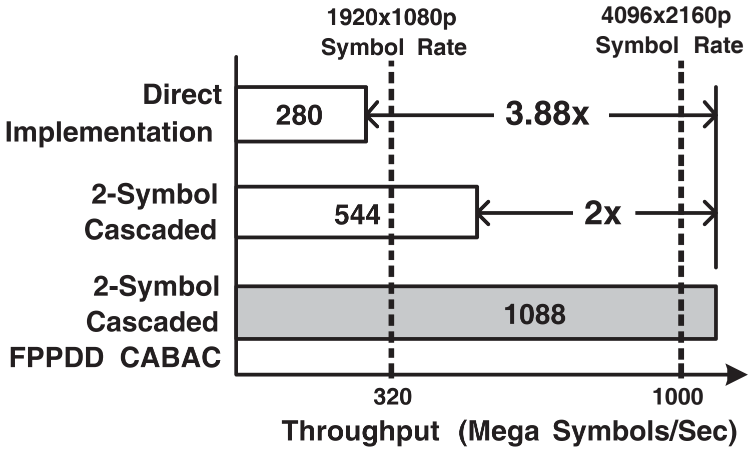


Figure 8.30: Throughput comparison with direct implementation and single CABAC arhcitectures.

of FPPDD CABAC. Dual CABAC computation cores are adopted, and each CABAC core has doubled pipeline cycle budget of 700 cycles. Dual CABAC computation cores process in an interleaved manner to be compatible with the proposed VPMBI scheduling, so the MB scheduling is preformed smoothly without being stalled by the pipeline-doubled CABAC stage. The throughput of the proposed architecture is shown in Fig. 8.30. The throughput (mega symbol per second) of the FPPDD CABAC architecture is 3.88× and 2× better than direct implementation and two-symbol cascaded architectures, respectively. Only FPPDD CABAC can meet the throughput requirement of 4096×2160p symbol rates.

## 8.6 Implementation Results

### 8.6.1 Chip Implementation

The detailed chip features and specifications are shown in Table 8.4. The core size of the chip is 11.46mm$^2$ (3.95mm×2.90mm), which contains 1732K gates using 90nm CMOS technology. This chip supports both H.264/AVC Multivew Extension and High Profile at Level 5.1. In addition, view scalability, which depends on the frame resolution, is supported for one to seven views. This chip supports maximum throughput of 212M pixels/sec and 830k MB/s at 280MHz for 4096×2160p videos.

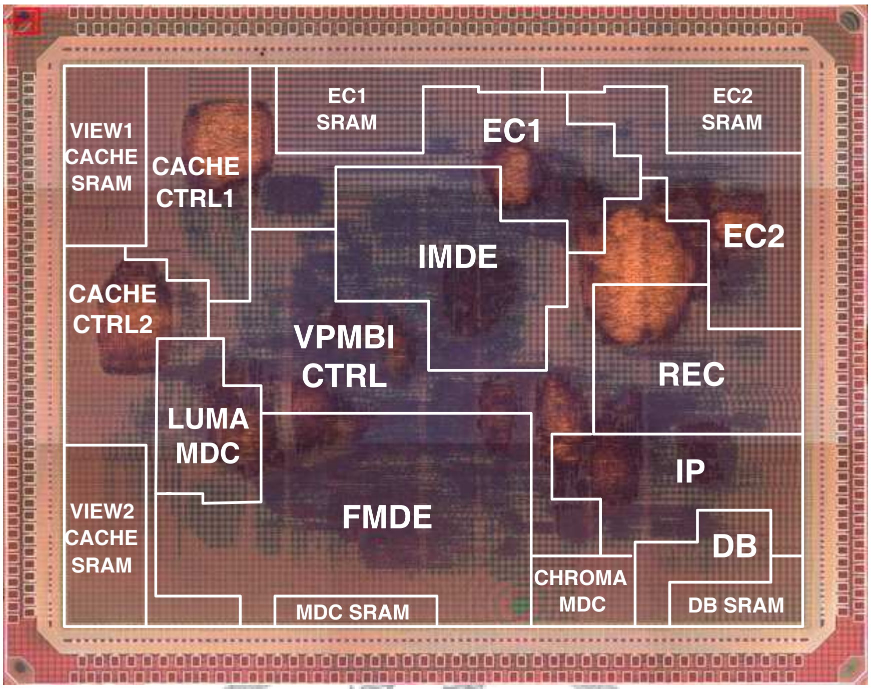


Figure 8.31: Chip micrograph of the MVC encoder.

Figure 8.31 shows the chip micrograph and the distribution of main modules.

### 8.6.2 Chip Comparison

Table 8.5 summarizes the the performance evaluation of the MVC encoder chip with the state-of-the-art encoder chips [78][94][93]. With the VPMBI scheduling and the 8-stage MB pipelining, our work provides 3.4 to 7.7 times throughput better than the previous works and supports the maximum frame resolution. The search range of ME/DE is 4 to 64 times larger than the previous works while only 20.1KB on-chip SRAM is used with the penalty of only 0.1dB quality degradation. The comparison of power efficiency is shown in Fig. 8.32. The power efficiency is defined as mega pixels per Watt. Note that the technology is scaled from 0.18$\mu$m and 0.13$\mu$m process to 90nm process. The MVC encoder chip provides the power efficiency 10% to 153% better than the previous works.

Figure 8.33 shows the evaluation of external memory bandwidth and on-chip SRAM size among these works. The external memory bandwidth and on-chip SRAM requirement for full search and hierarchical search algorithm are also illustrated. In the three kinds of HD resolution, the MVC chip requires the least external memory

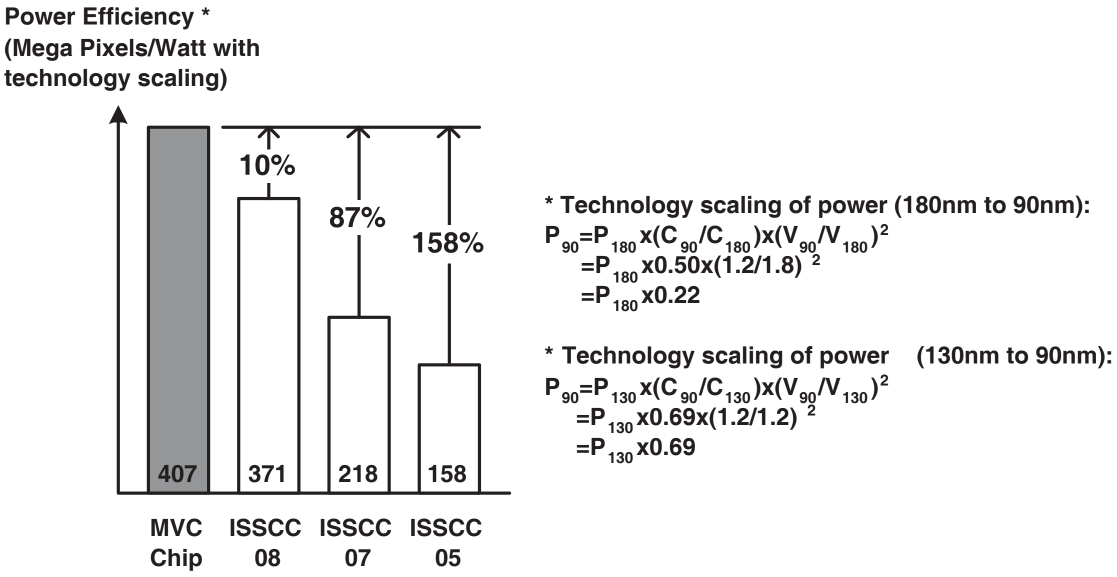
Table 8.4: Chip Specifications

| | |
|---|---|
| Technology | TSMC 90nm 1P9M CMOS |
| Supply Voltage | Core 1.2V, I/O 3.3V |
| Core Area | 3.95mm×2.90mm |
| Logic Gate Count | 1732K (2-input NAND gate) |
| On-chip SRAM | 20.1KB |
| Encoding Features | H.264/AVC Multiview Extension/High Profile@Level5.1 |
| View Scalability | 4096×2160p for 1 view |
| | 1920×1080p for up to 3 views |
| | 1280×720p for up to 7 views |
| Maximum Throughput | 212Mpixels/sec, 830k MB/sec@280MHz |
| ME/DE Search Range | [-256,+255]/[-256,+255] (horizontal/vertical) |
| Operating Frequency | 522mW@280MHz for 4096×2160p/24fps/Single view |
| &Power Consumption | 366mW@166MHz for 1920×1080p/30fps/Stereo views |
| | 317mW@144MHz for 1280×720p/30fps/Quad views |
| | 148mW@181MHz for 1920×1080p/30fps/Single view |
| | 58mW@136MHz for 1280×720p/30fps/Single view |

Table 8.5: Comparison with the state-of-the-art encoder chips

| | This Work | ISSCC’05 [78] | ISSCC’07 [94] | ISSCC’08 [93] |
|---|---|---|---|---|
| Maximum Resolution | 4096×2160@24fps | 1280×720@30fps | 1280×720@30fps | 1920×1080@30fps |
| Maximum Throughput | 212Mpixels/s | 27.6Mpixels/s | 27.6Mpixels/s | 62.2Mpixels/s |
| H.264 Profile | Multiview/High@Level5.1 | Baseline | Baseline | High@Level4 |
| Search Range H/V | [-256,+255]/[-256,+255] | [-64,+63]/[-32,+31] | [-32,+31]/[-32,+31] | [-128,+127]/[-128,+127] |
| Quality Loss† | 0.03 to 0.08dB | 0dB | ¡0.6dB | 0.1dB |
| Technology | TSMC 90nm | UMC 0.18$\mu$m | TSMC 0.13$\mu$m | UMC 0.13$\mu$m |
| Core Size | 3.95×2.90mm$^2$ | 7.68×4.13mm$^2$ | 4.30×4.30mm$^2$ | 3.17×3.17mm$^2$ |
| Gate Count | 1732K | 923K | 470K | 593K |
| On-chip SRAM | 20.1KB | 34.7KB | 13.3KB | 22.0KB |
| Power Consumption | 522mW@280MHz | 785mW@108MHz | 183mW@108MHz | 242mW@145MHz |

† Compared with full search block matching algorithm.



* Technology scaling of power (180nm to 90nm):

$$P_{90}=P_{180}\text{x}(C_{90}/C_{180})\text{x}(V_{90}/V_{180})^2$$
$$=P_{180}\text{x}0.50\text{x}(1.2/1.8)^2$$
$$=P_{180}\text{x}0.22$$

* Technology scaling of power (130nm to 90nm):

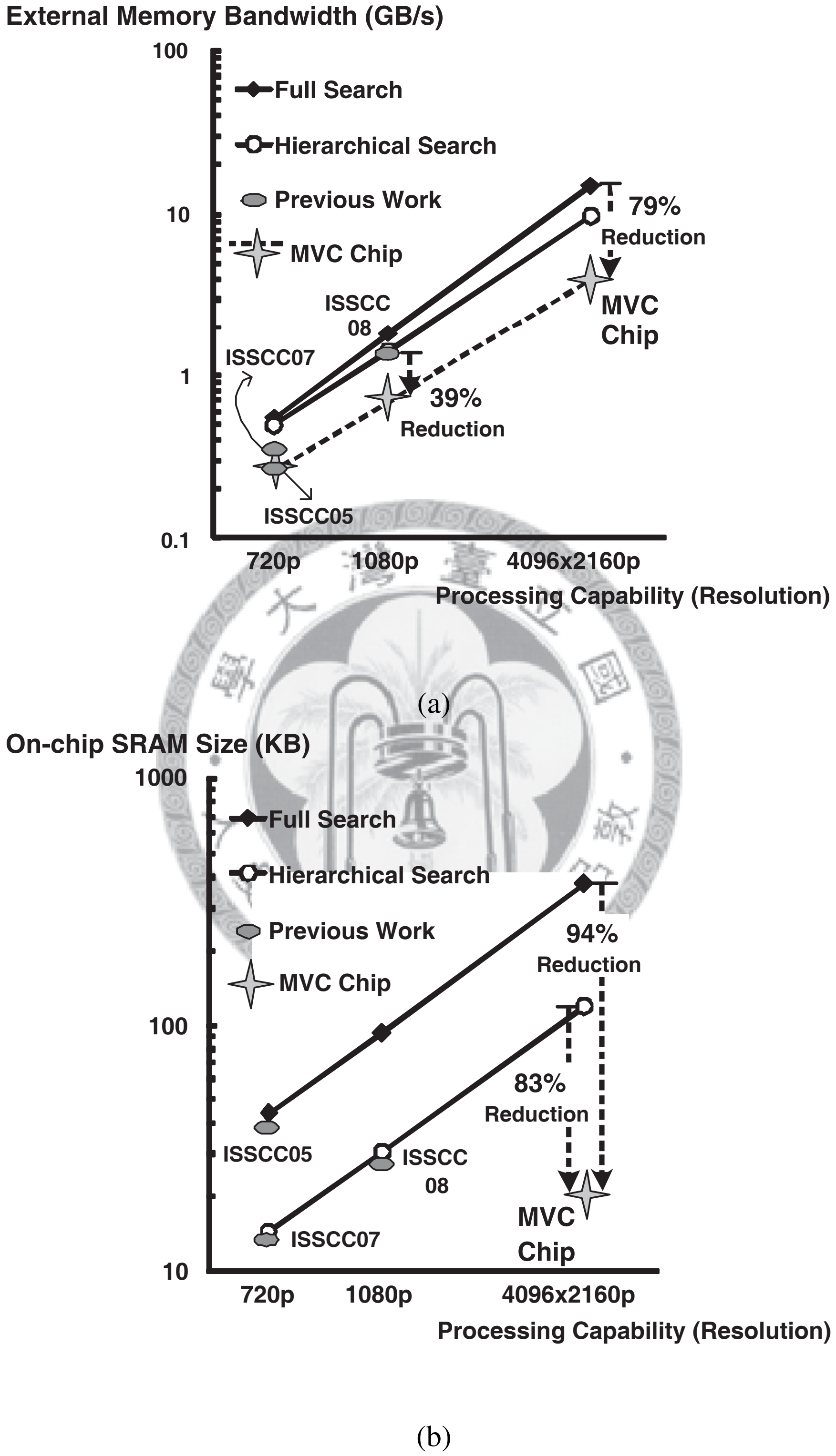
$$P_{90}=P_{130}\text{x}(C_{90}/C_{130})\text{x}(V_{90}/V_{130})^2$$
$$=P_{130}\text{x}0.69\text{x}(1.2/1.2)^2$$
$$=P_{130}\text{x}0.69$$

Figure 8.32: Comparison of power efficiency with the state-of-the-art encoder chips.

bandwidth and on-chip size. The proposed predictor-centered ME/DE algorithm is most suitable for the hardware implementation. Compared with [93], the proposed cache-based prediction core along with SW prefetching scheme reduces 39% external memory bandwidth. Also, 83% to 94% on-chip SRAM size is saved compared with the previous works scaled up to 4096×2160p resolution.

## 8.7 Summary

This chapter presents the design and implementation of an MVC encoder chip. The VPMBI scheduling is proposed to overcome the design challenges of high processing capability required for MVC and dealing with various MVC structures. The cache-based prediction core with a search window (SW) prefetching scheme and a predictor-centered ME/DE algorithm is proposed to save large on-chip SRAM and external memory bandwidth. In addition, the architecture and scheduling of each MB pipeline stage are analyzed and designed. The cache-based temporal/inter-view prediction stage saves 95% computation with quality loss of less than 0.1 dB in PSNR. The hybrid open-closed loop IP and pixel-forwarding REC stage overcomes the design challenge of data dependency and enhance the throughput of 1.8 to 2.7 times better than the conventional architectures. The FFPPDD CABAC co-operated with



(a)

(b)

Figure 8.33: Comparison of (a) off-chip memory bandwidth and (b) on-chip memory size with the state-of-the-art encoder chips.

the VPMBI scheduling provides 2 to 3.88 times throughput, and the encoding symbol rate required for 4096×2160p resolution is achieved.

The proposed MVC encoder is the first reported MVC single-chip encoder. This chip supports view scalability for encoding 1-view 4096×2160p, 3-view 1080p, and 7-view 720p videos for 3DTV and quad HDTV applications. The 212M pixels/s throughput is 3.4 to 7.7 times higher than the state-of-the-art encoder chips. In addition, the highest power efficiency of 407M pixels/Watt is achieved with the proposed VPMBI system scheduling, highly parallelism to reduce memory access, and module-wise clock gating. 79% system bandwidth and 94% on-chip SRAM are saved with cache-based prediction core. The search range is 4 to 64 times larger than the previous works to maintain HD video quality.

# Chapter 9

# Conclusion

## 9.1 Principal Contributions

In this dissertation, multiview video coding (MVC) technology, which is the key component in general three-dimensional television systems, is discussed from three different levels: algorithm level, VLSI architecture level, and system design level. At algorithm level, we develop an algorithm for color correlation, two fast prediction algorithms for stereo and MVC, and a predictor-centered algorithm for quad full high definition (QFHD) videos. These algorithms successfully reduce most computational complexity while maintain the video quality. At VLSI architecture level, two prototype chips are implemented for the prediction cores for stereo and MVC, respectively. According to the proposed architectures combined with the proposed algorithms, the prototype chips overcome the design challenges of large on-chip memory size and high off-chip memory bandwidth. At system design level, a bandwidth analysis method for MVC VLSI systems is first proposed. In addition, design and implementation of the worldwide first reported MVC single-chip encoder are presented. The detailed principal contributions are discussed in the following subsections.

### 9.1.1 Efficient Algorithm Development for Multiview Video Coding

In this part, the computational efficiency of MVC systems is discussed. To overcome the design challenge of huge computational complexity, the most computation-intensive prediction part, which is composed of motion estimation (ME) and disparity estimation (DE), is analyzed. Therefore, the content-aware prediction algorithm (CAPA) with inter-view mode decision is proposed. We utilize the characteristic of high inter-view correlation which results from the close distance between cameras. The mode distribution of neighboring view channels are analyzed first, and then the CAPA is designed according to the analysis. View channels are classified into primary and secondary channels, and a secondary channel is predicted by a primary channel with DE. High-quality ME, such as full search blocking matching algorithm (FSMBA), is performed in primary channels only to ensure the coding quality. In secondary channels, not only the motion vectors (MVs) but also macroblock (MB) types can be derived from the neighboring primary channels. Only small refinement ranges are required. In this way, 98.4–99.1% ME computational complexity can be saved with tiny quality loss of 0.03–0.06dB. It indicates that the computational redundancy indeed exists and is successfully explored and removed. The proposed CAPA can be easily combined with other fast prediction algorithms or coding tools to further enhance the coding efficiency.

In addition to the prediction part which removes the temporal and inter-view redundancy in MVC systems, the pre-processing part also play an important role. To overcome the design challenge of color mismatch between view channels, we develop an illuminance and chrominance correlation algorithm. The proposed algorithm is combined with ME flow so that the motion information can be reused. The linear regression is adopted to the motion estimated pixel pairs to derive color correlation parameters. Compared with the conventional histogram matching model, the proposed algorithm has better coding efficiency up to 0.4dB. Besides, the data flow of the proposed algorithm is suitable for VLSI implementation because it is proposed in MB level rather than frame level.

### 9.1.2 Algorithm and Architecture Co-Design of Prediction Core for 3D/Quad High Definition Video Coding

In the second part, the algorithm and architecture co-design of prediction cores for stereo and multiview/QFHD video coding are discussed, respectively. The stereo video coding is implemented on the MPEG-4 platform. A new prediction algorithm, joint prediction algorithm (JPA), is proposed. We develop a new coding tool called joint block compensation in JPA. The joint block compensation, which utilizes weighted sum of motion and disparity compensated block, effectively derive the better coding efficiency than conventional methods. In addition, the corresponding JPA architecture is also designed and implemented. The proposed architecture adopts hierarchical search block matching algorithm (HSBMA) to save 89% oh-chip memory. Different from previous ME architectures, the JPA architecture includes joint block generator and bandwidth saving circuits. Therefore, both near-FSBMA quality and bandwidth reduction is achieved. A prototype chip is fabricated with TSMC 0.18$\mu m$ technology to encode a D1 stereo video sequence in real-time with only 137K gates. Compared with the previous HSBMA architecture, the proposed architecture supports more powerful processing capability (two DE and one ME operations in 0.033 second) with similar silicon area.

Next, we extend the design space of prediction architectures from stereo video coding to multiview/QFHD video coding. Taking the implementation of multiview/ QFHD video processing architecture, FSBMA is no longer feasible due to its huge requirement of both on-chip memory size, off-chip memory bandwidth, and processing elements (PEs). Therefore, a predictor-centered algorithm is proposed to overcome the design challenges which results from FSBMA. The proposed preserves more motion information as predictors to enhance the MV accuracy. The design challenge of data dependency is also overcome by adopting the same coarse-grained MB predictors for all MB types. 96% computational complexity for integer ME is saved accompanied with the penalty of quality degradation of only 0.045dB. Because of irregular data access of the predictor-centered algorithm, we develop a cache-based prediction core architecture. A rapid prefetching algorithm is proposed to overcome the design challenge of cache miss penalty. 93% reduction of cache miss rate greatly enhance the hardware utilization. With well-organized prediction pattern and scheduling, the proposed architecture saves 39% off-chip memory bandwidth compared with previous works. A prototype chip is implemented with TSMC 90*nm* process with only 230K gates and 8KB SRAM. This chip supports ME for up to 4096×2160 resolution and 24 frames per second (fps) with [−256,+255]/[−256,+255] horizontal/vertical search ranges. Our design easily supports various HDTV specifications from 720p to 2160p resolution.

### 9.1.3 System Analysis and Architecture Design of 4096×2160p Multivew Video Single-Chip Encoder

In the last part, the system analysis and architecture design of 4096×2160p multivew video single-chip encoder is presented. A new system bandwidth method is proposed first to deal with the design challenges of limited bandwidth issues in a VLSI system. We adopt the concept of the precedence constraint in graph theory for deriving the processing order of frames in a complex MVC structure. Two frame-level data reuse schemes, current block centric scheduling (CBCS) and search window centric scheduling (SWCS) are proposed. The reuse schemes are classified according to the access times of the data, CB or SW. Combined with level-C+ data reuse scheme and the proposed inter-view data reuse scheme, hardware resource allocation can be systematical defined with the trade-off between system memory bandwidth and on-chip memory. Moreover, other possible frame-level data reuse schemes can be easily analyzed by our methodology.

Finally, the system architecture design and implementation of a 4096×2160p multivew video encoder chip which integrates the algorithms and architectures in Part I and Part II is presented. The proposed encoder provides the greatest processing capability which supports view scalability from one-view 4096×2160p to seven-view 1280×720p resolution. Proposed view-parallel macroblock interleaved (VPMBI) scheduling with eight-stage MB pipelined architecture successfully overcomes the design challenges of processing requirement of limited pipeline cycles and various complex MVC structures. The cache-based prediction core supports [−256, +255]/ [−256, +255] horizontal/vetical search rages while saves 95% computational complexity and overcomes the design challenges of large on-chip SRAM size and external memory bandwidth. The hybrid open-closed loop intra prediction and pixel-forwarding reconstruction circuits solves the data hazard and achieves 1.8 to 2.7 times throughput over conventional architectures. In addition, proposed frame-parallel pipeline-doubled (FPPDD) CABAC realizes 1088M symbol rate to meet the processing requirement of 4096×2160p videos. The prototype chip is fabricated on a 11.46 mm$^2$ die with TSMC 90*nm* technology. The proposed architecture is an area-efficient design with 1732K logic gate count and 20.1KB on-chip SRAM. The 212M pixels/s throughput is up to 7.7 times higher than the previous H.264/AVC encoder architectures. The proposed MVC encoder chip is the worldwide fastest published work in the world (*ISSCC*2009).

To sum up, the techniques related to multiview video coding proposed in this dissertation can be the essentials of future three-dimensional television systems.

## 9.2 Future Directions

We believe 3D video processing will be the mainstream in the future. Based on the contributions of MVC systems discussed in this dissertation, the research can be further extended at algorithm, architecture, and system levels as follows.

### 9.2.1 Algorithm Level

**Development of Efficient Camera Calibration Algorithm for Pre-Processing of Multiview Video Coding**

In this dissertation, the input multivew video frames are assumed to be calibrated. To extend the research space to be more practical, the un-calibrated multiple camera with different view angles contains more information of a scene structure. If MVC is applied to such a camera system, camera calibration is definitely an important issue because it directly influences the coding efficiency of MVC. It is considered as a pre-processing issue in a 3DTV system. Moreover, quality enhancement and combination with color correlation are also attractive research topics.

**Multivew Video Plus Depth (MVD): Depth Encoding and View Synthesis Prediction Algorithms**

Multiview video plus depth (MVD), which regards the depth maps as the source input, is the extension of MVC. It is considered as the next generation of coding standard after MVC. Depth information can provide many functionalities in an MVD system. If camera parameters are known, view synthesis prediction can help the corresponding view to predict other views. In addition, the coding method of these depth maps are also worthy of being discussed because of the different characteristics from color videos. They are important and challenging research topics.

**Development of Hardware-Oriented Virtual View Synthesis Algorithms**

Free view-point and smooth view transition are critical functionalities for 3DTV applications. However, the view channels provided by a camera array are not always sufficient. Therefore, the technology of virtual view synthesis is critical for connecting the MVC decoder and 3DTV display side. The rendering quality of a synthesized view is an important research issue because it directly effects users' subjective perception. It is considered as a post-processing issue in a 3DTV system. In addition, the real-time interaction for user' view selection is also a design challenge. For this reason, how to design a hardware-oriented algorithm for virtual view synthesis is a new and open research problems.

### 9.2.2 Architecture Level

**Compression-Assistant Pre-Processing Architecture**

Pre-processing is connected closely with MVC encoder, so the requirement of processing capability is critical as well. Neither camera calibration which contains matrix computation or color correlation which contains frame level computation can be afforded by general-purpose processors. Therefore, the VLSI architecture for preprocessing is worth designing. In addition, the on-chip and off-chip memory allocation for frame-level and MB-level processing, combination with MVC encoder to become compression-assistant hybrid architecture, are innovative research issues.

**High-Throughput Virtual View Rendering Architecture**

Virtual view rendering will become the most significant feature of future 3DTV and free-view point TV (FTV). Take human behavior into consideration, immediate feedback to view selection and high computational complexity for virtual view rendering make hardware acceleration become necessary. Virtual view rendering is a frame-level processing, how to optimize the data flow for task partitioning and pipelining is an important issue. The architecture can be co-designed with hardware-oriented virtual view synthesis algorithms.

### 9.2.3 System Level

**3D/Ultra High Definition Multivew Video Decoder with View Scalability**

Multiview video decoder is considered as the most critical part for 3DTV applications. Combined with high-throughput virtual view rendering architecture, MVC decoder with view scalability is the main component in a future 3DTV set-up box. The system architecture of the MVC decoder contains many research issues. First, high resolution is the trend for developing TV, and ultra high definition TV with 8192×4096 resolution is regarded as a extreme target. To decode such videos in real-time, external memory bandwidth and on-chip memory of the MVC decoder to support the motion and disparity compensation are challenging design issues. Second, MVC decoding is an MB-level processing while virtual view synthesis is a frame processing. Data flow arrangement and scheduling are required to be well-organized. Third, the component such as CABAC decoder, reconstruction, and transform circuits, may no be afforded by conventional architectures under much higher specifications and are necessary to be re-designed. This research will be very useful for 3D video processing in home entertainment applications.

In summary, 3D video processing is a difficult research direction and contains a lot of research topics. It will be urgently required in the future and lead a new evolution of digital content industry.

國立臺灣大學
敦品勵學・愛國愛人

# Bibliography

[1] O. Schreer, P. Kauff, and T. Sikora, *3D video communication: algorithms, concepts and real-time systems in human centred communication*, 2005.

[2] B. S. Wilburn, M. Smulski, H.-H. K. Lee, and M. A. Horowitz, "Light field video camera," in *Proceedings of Media Processors, SPIE ElectronicImaging*, 2002, vol. 4674, pp. 29–36.

[3] C. Zhang and T. Chen, "A self-reconfigurable camera array," in *Eurographics symposium on Rendering*, 2004.

[4] http://en.wikipedia.org/wiki/High-definition_television, "High-definition television," in *Wikipedia*.

[5] *Video Codec for Audiovisual Services at p × 64 Kbit/s*, ITU-T Recommendation H.261, Mar. 1993.

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

[7] *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2 and ITU-T Recommendation H.262, 1996.

[8] *Video Coding for Low Bit Rate Communication*, ITU-T Recommendation H.263, May 1996.

[9] *Video Coding for Low Bit Rate Communication*, ITU-T Recommendation H.263 version 2, Sept. 1997.

[10] *Video Coding for Low Bit Rate Communication*, ITU-T Recommendation H.263 version 3, Feb. 1998.

[11] *Information Technology - Coding of Audio-Visual Objects - Part 2: Visual*, ISO/IEC 14496-2, 1999.

[12] Joint Video Team, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, 2003.

[13] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. J. Sullivan, "Performance comparison of video coding standards using lagragian coder control," in *Proc. of IEEE International Conference on Image Processing*, 2002.

[14] G. Sullivan, P. Topiwala, and A. Luthra, "The H.264 advanced video coding standard : Overview and introduction to the fidelity range extensions," in *SPIE Conference on Application of Digital Image Processing XXVII*, Aug. 2004.

[15] D. Marpe and et al., "H.264/MPEG4-AVC fidelity range extensions : Tools, profiles, performance, and application areas," in *Proc. IEEE ICIP*, Sept. 2005, vol. 1, pp. 593–596.

[16] J. Reichel, H. Schwarz, and M. Wien, "Working Draft 4 of ISO/IEC 14496-10:2005/AMD3 Scalable Video Coding," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. N7555, Jan. 2005.

[17] ISO/IEC JTC 1/SC 29/WG11 N1088, *Proposed draft amendament No. 3 to 13818-2 (multi-view profile)*, MPEG-2, 1995.

[18] S.-Y. Chien, S.-H. Yu, L.-F. Ding, Y.-N. Huang, and L.-G. Chen, "Efficient stereo video coding system for immersive teleconference with two-stage hybrid disparity estimation algorithm," in *Proceedings of 2003 IEEE International on Image Processing*, 2003.

[19] J. M. Martinez, "Mpeg 3dav ahg activities report," Munich University of Technology, July 2003.

[20] Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, “Joint multiview video model (jmvm) 8.0,” ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Apr. 2008.

[21] F. Isgrò, E. Trucco, P. Kauff, and O. Schreer, “Three-dimensional image processing in the future of immersive media,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 388–303, Mar. 2003.

[22] A. Smolic and P. Kauff, “Interactive 3-D video representation and coding technologies,” *Proc. IEEE*, vol. 93, no. 1, pp. 98–110, Jan. 2005.

[23] M. Tanimoto, “Free viewpoint television - FTV,” in *Proceedings of 2004 Picture Coding Symposium*, Dec. 2004.

[24] T. Fujii and M. Tanimoto, “Free-viewpoint TV system based on ray-space representation,” *Proceedings of SPIE*, vol. 4864, pp. 175–189, Mar. 2002.

[25] A. Smolic, K. Mueller, P. Merkle, T. Rein, M. Kautmer, P. Eisert, and T. Wiegand, “Free viewpoint video extraction, representation, coding, and rendering,” in *Proceedings of 2003 IEEE International on Image Processing*, Oct. 2004, vol. 5, pp. 3287–3290.

[26] ISO/IEC JTC1/SC29/WG11 N6501, *Requirements on multi-view video coding*, 2004.

[27] ISO/IEC JTC1/SC29/WG11 N6501 W8019, *Description of Core Experiments in MVC*, Apr. 2006.

[28] G. Li and Y. He, “A novel multi-view video coding scheme based on H.264,” in *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing*, Dec. 2003, vol. 4, pp. 218–222.

[29] Y.-W. Huang, T.-C. Chen, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, C.-S. Chen, C.-F. Shen, S.-Y. Ma, T.-C. Wang, B.-Y. Hsieh, H.-C. Fang, and L.-G. Chen, “A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications,” 2005.

[30] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, no. 2, pp. 139–154, Mar. 1985.

[31] N. Grammalidis and M. G. Strintzis, "Disparity and occlusion estimation in multiocular systems and their coding for the communication of multiview image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 328–344, June 1998.

[32] J.-R. Ohm and K. Müller, "Incomplete 3-D multiview representation of video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 2, pp. 389–400, Mar. 1999.

[33] R.-S. Wang and Y. Wang, "Multiview video sequence analysis, compression, and virtual viewpoint synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 397–410, Apr. 2000.

[34] G. Heising, "Efficient and robust motion estimation in grid-based hybrid video coding schemes," in *Proceedings of International Conference on Image Processing*, 2002, pp. 687–700.

[35] Y. Luo, Z. Zhang, and P. An, "Stereo video coding based on frame estimation and interpolation," *IEEE Trans. Broadcast.*, vol. 49, no. 1, pp. 14–21, Jan. 2003.

[36] X. Guo, Y. Lu, and W. Gao, "Inter-view direct mode for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1527–1532, Dec. 2006.

[37] L.-F. Ding, S.-Y. Chien, and L.-G. Chen, "Joint prediction algorithm and architecture for stereo video hybrid coding systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 11, pp. 1324–1337, Nov. 2006.

[38] P.-L. Lai and A. Ortega, "Predictive fast motion/disparity search for multiview video coding," in *Proceedings of SPIE Visual Communications and Image Processing 2006*, 2006.

[39] ISO/IEC JTC1/SC29/WG11 N6501 m13544, *Results on CE1 for multi-view video coding*, ISO/IEC, July 2006.

[40] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, June 2004.

[41] ISO/IEC JTC1/SC29/WG11 N6501 N7829, *AHG on Multiview Video Coding*, ISO/IEC, Jan. 2006.

[42] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video processing and communication*, Prentice Hall, 2001.

[43] T. Koga, K. Linuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motioncompensated interframe coding for video conferencing," *in Proc. NTC*, pp. C9.6.1–9.6.5, Nov. 1981.

[44] L.M. Po and W.C. Ma, "A new center-biased search algorithm for block motion estimation," *IEEE Trans. Image Processing*, pp. 23–26, Oct. 1995.

[45] S. Zhu and K.K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *Information, Communications and Signal Processing*, pp. 9–12, Sept. 1997.

[46] Boyce and J. M, "Weighted prediction in the H.264/MPEG AVC video coding standard," May 2004, vol. 3, pp. 789–792.

[47] Y. Chen and C. Cai J. Chen, "Ni luminance and chrominance correction for multi-view video using simplified color error model," in *Proceedings of 2006 Picture Coding Symposium*, Dec. 2006.

[48] K. Kamikura, H. Watanabe, H. Jozawa, and S. Ichinose, "Global brightness-variation compensation for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 11, pp. 988–1000, Dec. 1998.

[49] ISO/IEC JTC1, "JSVM 3.5 software," ISO/IEC JTC1/WG11 JVT-P064.

[50] S.-H. Kim and R.-H. Park, "Fast local motion-compensation algorithm for video sequences with brightness variations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 4, pp. 289–299, Apr. 2003.

[51] ISO/IEC JTC1, "Luminance and chrominance compensation for multi-view sequences using histogram matching," ISO/IEC JTC1/SC29/WG11, 2005.

[52] U. Fecker, M. Barkowsky, and A. Kaup, "Improving the prediction efficiency for multi-view video coding using histogram matching," in *Proceedings of 2006 Picture Coding Symposium*, Apr. 2006.

[53] H.-C. Chang, L.-G. Chen, M.-Y. Hsu, and Y.-C. Chang, "Performance analysis and architecture evaluation of MPEG-4 video codec system," 2000.

[54] S.-Y. Chien, S.-H. Yu, L.-F. Ding, Y.-N. Huang, and L.-G. Chen, "Fast disparity estimation algorithm for mesh-based stereo image/video compression with two-stage hybrid approach," in *Proceedings of SPIE Visual Communications and Image Processing 2003*, 2003.

[55] S.-Y. Chien, *Video segmentation: algorithms, hardware architectures, and applications*, Ph.D. thesis, Nation Taiwain University, Taipei, May 2003.

[56] MPEG-4 Video Group, *Generic Coding of Audio-Visual Objects: Part 2-Visual 14496-2*, ISO/IEC JTC1/SC29/WG11 N2502a, FDIS, Atlantic City, 1998.

[57] W. Yang, K.-N. Ngan, and J. Cai, "MPEG-4 based stereoscopic and multiview video coding," in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2004, pp. 61–64.

[58] Joint Vieo Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, *Advanced Video Coding for Generic Audiovisual Services*, Joint Vieo Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, May 2003.

[59] J. N. Ellinas and M. S. Sangriotis, "Stereo video coding based on interpolated motion and disparity estimation," in *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, 2003, pp. 301–306.

[60] Y. Luo, Z. Zhang, and P. An, "Stereo video coding based on frame estimation and interpolation," *IEEE Trans. Broadcast.*, vol. 49, no. 1, pp. 14–21, Mar. 2003.

[61] H.-Z. Jia, W. Gao, and Y. Lu, "Stereoscopic video coding based on global displacement compensated prediction," in *International Conference on Information and Communications Security*, 2003, pp. 61–65.

[62] MPEG-4 Group, *The MPEG-4 Video Standard Verification Model version 18.0*, ISO/IEC JTC 1/SC 29/WG11 N3908, 2001.

[63] S. Cho, K. Yun, B. Bae, Y. Hahm, C. Ahn, Y. Kim, K. Sohn, and Y. h. Kim, *Report for EE3 in MPEG 3DAV*, ISO/IEC JTC1/SC29/WG11 M9186, December 2002.

[64] C. Zhu, X. Lin, and L.P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Transaction on Circuit and System for Video Technology*, pp. 349–355, May 2002.

[65] B.-C. Song and K.-W. Chun, "Multi-resolution block matching algorithm and its vlsi architecture for fast motion estimation in an mpeg-2 video encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 1119–1137, 2004.

[66] W.-M. Chao, C.-W. Hsu, Y.-C. Chang, and L.-G. Chen, "A novel hybrid motion estimator supporting diamond search and fast full ssearch," in *IEEE International Symposium on Circuits and Systems*, 2002.

[67] J. C. Tuan, T. S. Chang, and C. W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching vlsi architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61–72, Jan. 2002.

[68] L.-F. Ding, S.-Y. Chien, and L.-G. Chen, "Algorithm and architecture of prediction core in stereo video hybrid coding system," in *Proceedings of 2005 IEEE Workshop on Signal Processing Systems*, 2005.

[69] Tung-Chien Chen, Yu-Wen Huang, and Liang-Gee Chen, "Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture," *Circuits and Sys-*

*tems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 2, pp. II–273–6 Vol.2, 23-26 May 2004.

[70] Alexis Michael Tourapis, Oscar C. Au, and Ming Lei Liou, "Predictive motion vector field adaptive search technique (PMVFAST) - enhancing block based motion estimation," in *Proceedings of Visual Communications and Image Processing 2001 (VCIP'01)*, 2001.

[71] Alexis M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," 2002, vol. 4671, pp. 1069–1079, SPIE.

[72] D. Tzovaras, M. G. Strintzis, and H. Sahinolou, "Evaluation of multiresolution block matching techniques for motion and disparity estimation," *Signal Processing: Image Commun.*, vol. 6, pp. 56–67, 1994.

[73] C. J. Duanmu, M. O. Ahmad, and M. N. S. Swamy, "8-bit partial sum of 16 luminance values for fast block motion estimation," in *Proc. of IEEE Int. Conf. Multimedia Expo (ICME'03)*, 2003, pp. 689–692.

[74] V. Iverson, J. McVeigh, and B. Reese, "Real-time H.264/AVC codec on Intel architectures," in *IEEE International Conference on Image Processing. Singapore*, 2004, pp. 757–760.

[75] Jen-Chieh Tuan, Tian-Sheuan Chang, and Chein-Wei Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 1, pp. 61–72, Jan 2002.

[76] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, and L.-G. Chen, "Level C+ data reuse scheme for motion estimation with corresponding coding orders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 553–558, Apr. 2006.

[77] Chuan-Yuan Tsai, Chen-Han Chung, Yu-Han Chen, Tung-Chien Chen, and Liang-Gee Chen, "Low power cache algorithm and architecture design for fast

motion estimation in H.264/AVC encoder system," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2, pp. II–97–II–100, 15-20 April 2007.

[78] et al. Y.-W. Huang, "A 1.3TOPS H.264/AVC single-chip encoder for hdtv applications," Feb. 2005, pp. 128–129.

[79] Hansoo Kim and In-Cheol Park, "High-performance and low-power memory-interface architecture for video processing applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 11, pp. 1160–1170, Nov. 2001.

[80] Tetsuro Takizawa and Masao Hirasawa, "An efficient memory arbitration algorithm for a single chip MPEG2 AV decoder," *IEEE Transactions on Consumer Electronics*, vol. 47, no. 3, pp. 660–665, Aug. 2001.

[81] A.J. Smith, "Cache Memories," *ACM Computing Surveys (CSUR)*, vol. 14, no. 3, pp. 473–530, 1982.

[82] T.F. Chen and J.L. Baer, "A performance study of software and hardware data prefetching schemes," *Computer Architecture, 1994. Proceedings the 21st Annual International Symposium on*, pp. 223–232, 1994.

[83] J.L. Baer and T.F. Chen, "An effective on-chip preloading scheme to reduce data access penalty," *Proceedings of the 1991 ACM/IEEE conference on Supercomputing*, pp. 176–186, 1991.

[84] JWC Fu, JH Patel, and BL Janssens, "Stride Directed Prefetching In Scalar Processors," *Microarchitecture, 1992. MICRO 25., Proceedings of the 25th Annual International Symposium on*, pp. 102–110, 1992.

[85] Yu-Wen Huang, Tung-Chien Chen, Chen-Han Tsai, Ching-Yeh Chen, To-Wei Chen, Chi-Shi Chen, Chun-Fu Shen, Shyh-Yih Ma, Tu-Chih Wang, Bing-Yu Hsieh, Hung-Chi Fang, and Liang-Gee Chen, "A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications," *Solid-State Circuits Conference, 2005.*

*Digest of Technical Papers. ISSCC. 2005 IEEE International*, pp. 128–588 Vol. 1, 10-10 Feb. 2005.

[86] Zhenyu Liu, Yang Song, Ming Shao, Shen Li, Lingfeng Li, S. Ishiwata, M. Nakagawa, S. Goto, and T. Ikenaga, “A 1.41W H.264/AVC real-time encoder SOC for HDTV1080p,” *VLSI Circuits, 2007 IEEE Symposium on*, pp. 12–13, 14-16 June 2007.

[87] Y. K. Lin, D. W. Li, C. C. Lin, T. Y. Kuo, S. J. Wu, W. C. Tai, W. C. Chang, and T. S. Chang, “A 242mW 10mm$^2$ 1080p H.264/AVC high-profile encoder chip,” in *IEEE C ISSCC*, 2008.

[88] T.-C. Chen, Y.-W. Huang, C.-Y. Tsai, C.-T. Huang, and L.-G. Chen, “Single reference frame multiple current macroblocks scheme for multi-frame motion estimation in H.264/AVC,” May 2005, pp. 1790–1793.

[89] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, “On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61–72, Jan. 2002.

[90] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, and L.-G. Chen, “Level C+ data reuse scheme for motion estimation with corresponding coding orders,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 553–558, Apr. 2006.

[91] Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, “Joint draft 7.0 on multiview video coding,” ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Apr. 2008.

[92] P. G. J. Barten, “Evaluation of subjective image quality with the square-root integral method,” *Journal of the Optical Society of America A*, vol. 7, pp. 2024–2031, Oct. 1990.

[93] et al. Y.-K. Lin, “A 242mW 10mm2 1080p H.264/AVC High-Profile encoder chip,” Feb. 2008, pp. 314–315.

[94] et al. H.-C. Chang, “A 7mW to 183mW dynamic quality-scalable H.264 video encoder chip,” Feb. 2007, pp. 128–129.

[95] Y.-W. Huang and et al., “Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder,” *IEEE Trans. on CSVT*, vol. 15, no. 3, pp. 378–401, Mar. 2005.

[96] C.-W. Ku and et al., “A high-definition H.264/AVC intra-frame codec IP for digital video and still camera applications,” *IEEE Trans. on CSVT*, vol. 16, no. 8, pp. 917–928, Aug. 2006.

[97] K. Suh, S. Park, and H.Cho, “An efficient hardware architecture of intra prediction and TQ/IQIT module for H.264 encoder,” *ETRI Journal*, vol. 27, 2005.

[98] Y.-W. Huang and et al., “A 1.3tops H.264/AVC single-chip encoder for HDTV applications,” in *Proc. of IEEE ISSCC*, 2005, pp. 128–588.

[99] D. Marpe, H. Schwarz, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–644, July 2003.