

國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

頻寬-位元率-失真最佳化之移動估測

Bandwidth-Rate-Distortion Optimized Motion Estimation

研究生：戴瑋呈

指導教授：張添烜

中華民國 九十七年 九月

頻寬-位元率-失真之移動估測

Bandwidth-Rate-Distortion Optimized Motion Estimation

研究生：戴瑋呈

Student : Wei-Cheng Tai

指導教授：張添烜 博士

Advisor: Tian-Sheuan Chang



A Thesis

Submitted to Department of Electronics Engineering & Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of Requirements

for the Degree of Master

In

Electronics Engineering

September 2008

Hsinchu, Taiwan, Republic of China

中華民國 九十七年 九月

頻寬-位元率-失真最佳化之移動估測

研究生：戴瑋呈

指導教授：張添烜 博士

國立交通大學電子研究所碩士班

摘 要

移動估測在 H.264 視訊編碼的過程中，具有龐大的運算量和記憶體需求，然而傳統的移動估測只考慮位元率和失真，未將記憶體頻寬納入考量，因此，在頻寬受限的情形下，位元率和失真並未做到最佳化。為了解決以上因素，在此本論文提出一個頻寬-位元率-失真最佳化的移動估測演算法，首先，我們提出一個頻寬-位元率-失真最佳化的模型，藉由設立一個合理的搜尋範圍，在允許的頻寬下提升位元率和失真的最大效能。其次，我們提出兩個方法來進一步提高前者模型的效能，一種方法為隨內容感測的跳躍預測演算法，藉由跳躍預測所節省的頻寬來提升其餘複雜畫面的編碼品質；另一方法為搜尋範圍邊界預測演算法，藉由設立一個適宜的搜尋範圍邊界而對搜尋範圍作進一步的修正。相較於之前的研究，對於靜態畫面影像，我們的設計在相同位元率與失真，以及平均搜尋範圍為 16 的情形下，頻寬可節省 70%，若加上跳躍預測演算法，則頻寬可節省 84%；對於動態畫面影像，我們的設計在低頻寬的環境下，位元率可降低 13%，同時峰值信噪比也可提升 0.1dB。總結我們的設計對於不同的頻寬環境變化下，不僅維持了效能，甚至更進一步提升了效能，這顯現出我們的設計可適用於改進移動估測的處理。

Bandwidth-Rate-Distortion Optimized Motion Estimation

Student: Wei-Cheng Tai

Advisor: Tian-Sheuan Chang

Institute of Electronics
National Chiao Tung University

Abstract

Motion estimation (ME) processing is the most computational and memory intensive component in H.264 encoder. However, traditional ME algorithms focus on rate and distortion performance and thus do not take memory bandwidth into consideration. Therefore, the rate and distortion performance are not optimized under bandwidth constraint. In this thesis, we propose bandwidth-rate-distortion (B-R-D) optimized ME algorithm to solve the issue mentioned above. First, we mainly propose a B-R-D optimized modeling method to determine an appropriate search range (SR) for maximizing rate distortion efficiency while can dynamically meet the available bandwidth. Then, we propose two methods, skip mode detection with content-aware scheme and SR boundary prediction method, to enhance the performance of B-R-D optimized modeling method. The skip mode detection with content-aware scheme is presented to save the most memory bandwidth and thus gives other complex MBs more bandwidth for better quality, and the SR boundary prediction method is presented to determine a feasible SR boundary for SR refinement. Compared with reference software [3], when coding in low motion sequence, the simulation result shows the proposed BRD design could improve the bandwidth saving up to 70% with almost the same performance at bit rate and PSNR under average search range size 16, and up to 84% with negligible PSNR degradation with skip design added; while coding in high motion sequence, the simulation result shows our design could save average bit rate up to 13% and at the same time increase average PSNR up to 0.1dB under low bandwidth constraint. In summary, our design could achieve the same and sometimes even better performance under various bandwidth constraints and thus it is suitable for improving ME process.

誌謝

在交大的兩年時光裡，經歷了許多研究的困難，能夠順利地得到這個學位，得感謝許多人的幫助。首先，要感謝我的指導教授—張添烜博士，這兩年來給我的支持與鼓勵，無論在研究或生活上，每當遇到問題總是能給予我建議與協助，亦師亦友般地訓練自己獨立思考的能力，讓我克服難關，順利地完成學業。另外，我也要感謝我的口試委員們，交大資工彭文孝教授和清華電機陳永昌教授，感謝你們百忙中抽空指導我，你們的寶貴意見使我獲益良多，讓我的論文更加完備。

感謝 VSP 實驗室的好夥伴們，特別感謝引我入門的林佑昆學長，帶領我一步一步的做好研究，不厭其煩的給予指導，使我順利的完成研究。謝謝張彥中、李國龍、李得璋、郭子筠、林嘉俊、吳私璟、廖英澤學長，教導我 IC 設計與 H.264 編碼的觀念與技巧。再來要感謝曾宇晟、蔡宗憲、詹景竹同學，和你們一同討論研究、嬉鬧彼此的過程，是一段很難忘的回憶。還有感謝張瑋城同學，從大學至今的相互砥礪，一起在研究室或寢室熬夜趕研究，是一段很珍貴的日子。另外感謝實驗室的學弟妹們：黃筱珊、許博淵、沈孟維、蔡政君、陳之悠、廖元歆，活潑的你們，使我的研究生涯充滿歡樂。還要謝謝呂進德、李韋磬，和你們一同聊天運動，是我減輕壓力的最好方式。

謝謝我的女友，無論任何時刻，總是在第一時間傾聽我、包容我，一同分享我的喜怒哀樂，有時假日更陪著我在實驗室裡奮鬥，滿滿的感動溢於言表，沒有你的支持與鼓勵，就不會有成為碩士生的我。

最後要感謝支持我的家人們，我的爸媽和兩個弟弟們，在電話的另一端給予我愛的鼓勵，你們的溫暖是我努力的最大支柱。

在此，僅將本論文獻給所有愛我與所有我愛的人。

Content

1.	Introduction.....	1
1.1.	Background.....	1
1.2.	Motivation and contribution	2
1.3.	Thesis organization	2
2.	Overview of environment-aware motion estimation algorithms	3
2.1.	Overview of variable block-based motion estimation	3
2.2.	Review of adaptive search range motion estimation	5
2.3.	Review of power-aware motion estimation	6
2.4.	Review of computation-aware motion estimation	7
2.5.	Review of skip mode detection algorithm	8
2.5.1.	Lagrangian cost motion estimation	9
2.5.2.	All zero DCT blocks detection	9
3.	Proposed B-R-D optimized motion estimation algorithm	11
3.1.	Introduction.....	12
3.2.	Proposed skip mode detection with content-aware scheme.....	13
3.2.1.	Review of SAD-4x4-block threshold.....	13
3.2.2.	Refinement of SAD-4x4-block threshold	15
3.3.	Proposed B-R-D optimized modeling method.....	16
3.4.	SR boundary prediction method	24
3.5.	Summary	25
4.	Simulation and Analysis	27
4.1.	BW pattern setting	27
4.2.	Experimental result	29
4.2.1.	Performance comparison	30

4.2.2.	The distribution of MB for skip mode analysis	46
4.2.3.	Timing comparison with skip detection.....	48
4.2.4.	Completion time comparison of BW random patterns	49
4.3.	Summary	51
5.	Hardware implementation.....	53
5.1.	Hardware design	53
5.2.	Implementation result	54
6.	Conclusion and future work.....	55
6.1.	Conclusion	55
6.2.	Future work.....	56
7.	Reference	57



List of Figure

Fig. 2-1 (a)The mode hierarchy and (b) its block size for H.264	4
Fig. 2-2 Different modes for H.264 motion estimation	4
Fig. 2-3 Search range prediction using neighboring vectors	5
Fig. 2-4 Power aware multimedia systems [8].....	7
Fig. 3-1 The total BRD optimized motion estimation algorithm flow	12
Fig. 3-2 Skip mode detection flow.....	13
Fig. 3-3 The trend between boundary and SAD (Normalized to Akiyo).....	15
Fig. 3-4 (a) SAD and SAD-4x4-block threshold under different QP Threshold estimation under (b) QP20 (c) QP24 (d) QP28 (e) QP32 (f) QP36 ..	16
Fig. 3-5 B-R-D optimized modeling method flow.....	17
Fig. 3-6 Illustration of BW budget.....	17
Fig. 3-7 Illustration of BW prediction	20
Fig. 3-8 Illustration of bandwidth boundary determination.....	20
Fig. 3-9 Illustration of SR decision.....	22
Fig. 3-10 Illustration of SR modification.....	23
Fig. 3-11(a) Search range boundary predicted method flow	24
Fig. 4-1 6 kind of BW patterns: (a) SR constant 8 (b) SR constant 16 (c) SR constant 24 (d) SR random 8 (e) SR random 16 (f) SR random 24	28
Fig. 4-2 The example of dynamically adjust the SR (a) The performance example of dynamiclly adjust the SR: (b) PSNR (c) Bit-rate .	29
Fig. 4-3 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time.....	34
Fig. 4-4 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time.....	34
Fig. 4-5 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time.....	35
Fig. 4-6 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time.....	35
Fig. 4-7 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time.....	36
Fig. 4-8 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time.....	36
Fig. 4-9 RD curve comparison under SR constant 8 for “Akiyo” sequence.....	40
Fig. 4-10 RD curve comparison under SR constant 8 for “Foreman” sequence	40

Fig. 4-11 RD curve comparison under SR constant 8 for “Stefan” sequence	40
Fig. 4-12 RD curve comparison under SR constant 16 for “Akiyo” sequence.....	41
Fig. 4-13 RD curve comparison under SR constant 16 for “Foreman” sequence	41
Fig. 4-14 RD curve comparison under SR constant 16 for “Stefan” sequence	41
Fig. 4-15 RD curve comparison under SR constant 24 for “Akiyo” sequence.....	42
Fig. 4-16 RD curve comparison under SR constant 24 for “Foreman” sequence	42
Fig. 4-17 RD curve comparison under SR constant 24 for “Stefan” sequence	42
Fig. 4-18 RD curve comparison under SR random 8 for “Akiyo” sequence.....	43
Fig. 4-19 RD curve comparison under SR random 8 for “Foreman” sequence	43
Fig. 4-20 RD curve comparison under SR random 8 for “Akiyo” sequence.....	43
Fig. 4-21 RD curve comparison under SR random 16 for “Akiyo” sequence.....	44
Fig. 4-22 RD curve comparison under SR random 16 for “Foreman” sequence	44
Fig. 4-23 RD curve comparison under SR random 16 for “Stefan” sequence	44
Fig. 4-24 RD curve comparison under SR random 24 for “Akiyo” sequence.....	45
Fig. 4-25 RD curve comparison under SR random 24 for “Foreman” sequence	45
Fig. 4-26 RD curve comparison under SR random 24 for “Stefan” sequence	45
Fig. 4-27 The distribution of MB for “Akiyo” sequence.....	47
Fig. 4-28 The distribution of MB for “foreman” sequence	47
Fig. 4-29 The distribution of MB for “Stefan” sequence.....	47
Fig. 4-30 Coding time curve with skip detection of CIF sequences under:	
(a) SR constant 8 (b) SR constant 16 (c) SR constant 24 (d) SR random 8	
(e) SR random 16 (f) SR random 24 patterns.....	49
Fig. 4-31 Completion time comparison under SR random 8 pattern.....	50
Fig. 4-32 Completion time comparison under SR random 16 pattern.....	50
Fig. 4-33 Completion time comparison under SR random 24 pattern.....	51
Fig. 4-34 Illustration of completion time comparison under different SR random pattern	
.....	51
Fig. 5-1 BRD optimized motion estimation algorithm hardware architecture	53
Fig. 5-2 Hardware verification for BRD optimized motion estimation algorithm	54

List of Table

TABLE 3-1 Boundary determination of QP 28 (mean, variance, boundary and maxima for the 4x4-block SAD distribution which higher than T0).....	14
TABLE 3-2 Spike threshold under different QP.....	14
TABLE 3-3 Boundary determination under different QP.....	14
TABLE 3-4 The boundary and SAD value under QP28 of different sequences	15
TABLE 3-5 The boundary and SAD value under QP 28 of different sequences (Normalized to Akiyo)	15
TABLE 4-1 Bandwidth usage of one MB.....	29
TABLE 4-2 Performance of BRD and BRD + Skip model under SR const pattern for “Akiyo” sequence	33
TABLE 4-3 Performance of BRD and BRD + Skip model under SR const pattern for “Foreman” sequence	33
TABLE 4-4 Performance of BRD and BRD + Skip model under SR const pattern for “Stefan” sequence	33
TABLE 4-5 Performance of BRD and BRD + Skip model under SR random pattern for “Akiyo” sequence	33
TABLE 4-6 Performance of BRD and BRD + Skip model under SR random pattern for “Foreman” sequence	33
TABLE 4-7 Performance of BRD and BRD + Skip model under SR random pattern for “Stafen” sequence	33
TABLE 4-8 RD comparison of BRD and BRD + Skip model under SR constant 8 for “Akiyo” sequence	37
TABLE 4-9 RD comparison of BRD and BRD + Skip model under SR constant 8 for “Foreman” sequence	37
TABLE 4-10 RD comparison of BRD and BRD + Skip model under SR constant 8 for “Stefan” sequence	37
TABLE 4-11 RD comparison of BRD and BRD + Skip model under SR constant 16 for “Akiyo” sequence	37
TABLE 4-12 RD comparison of BRD and BRD + Skip model under SR constant 16 for “Foreman” sequence	37

TABLE 4-13 RD comparison of BRD and BRD + Skip model under SR constant 16 for “Stefan” sequence	37
TABLE 4-14 RD comparison of BRD and BRD + Skip model under SR constant 24 for “Akiyo” sequence	38
TABLE 4-15 RD comparison of BRD and BRD + Skip model under SR constant 24 for “Foreman” sequence	38
TABLE 4-16 RD comparison of BRD and BRD + Skip model under SR constant 24 for “Stefan” sequence	38
TABLE 4-17 RD comparison of BRD and BRD + Skip model under SR random 8 for “Akiyo” sequence	38
TABLE 4-18 RD comparison of BRD and BRD + Skip model under SR random 8 for “Foreman” sequence	38
TABLE 4-19 RD comparison of BRD and BRD + Skip model under SR random 8 for “Stefan” sequence	38
TABLE 4-20 RD comparison of BRD and BRD + Skip model under SR random 16 for “Akiyo” sequence	39
TABLE 4-21 RD comparison of BRD and BRD + Skip model under SR random 16 for “Foreman” sequence	39
TABLE 4-22 RD comparison of BRD and BRD + Skip model under SR random 16 for “Stefan” sequence	39
TABLE 4-23 RD comparison of BRD and BRD + Skip model under SR random 24 for “Akiyo” sequence	39
TABLE 4-24 RD comparison of BRD and BRD + Skip model under SR random 24 for “Foreman” sequence	39
TABLE 4-25 RD comparison of BRD and BRD + Skip model under SR random 24 for “Stefan” sequence	39
TABLE 4-26 Coding time with skip detection of CIF sequences under: (a) SR constant 8 (b) SR constant 16 (c) SR constant 24 (d) SR random 8 (e) SR random 16 (f) SR random 24 patterns	48

1. Introduction

1.1. Background

The emerging popular multimedia technology, such as digital television, mobile phone and DVD player bring us convenience in daily life. However, the data amount of video is too large to transmit or record without compression techniques. Therefore, several compression techniques have been proposed to reduce the data and bandwidth efficiently. The H.264/AVC standard [1] has been adopted recently as a popular compression technique from its high compression rate. In which, motion estimation (ME) part is the most computational and memory intensive component in H.264 encoder. To support these high computation and high bandwidth on ME, several algorithms have been proposed. However, traditional ME algorithms focus on rate and distortion performance, and thus do not take memory bandwidth into consideration. While coding under bandwidth constraint, it will lead to a significant quality loss or the coding time will be delayed. Therefore, the rate and distortion performance are not optimized under bandwidth constraint.

1.2. Motivation and contribution

The issue mentioned above motivates us to develop rate distortion optimized motion estimation under the available memory bandwidth constraint. The bandwidth-rate-distortion optimized concept has a lot of similarities with the power-aware design [8][9][10][11] and computation-aware design [12][13][14][15][16], and both these designs develop as a basis of rate-control-like procedure. Therefore, we propose a rate-control-like procedure for macroblock (MB)-level bandwidth allocation, which not only meets the bandwidth constraint, but also maximizes the coding efficiency.

The contribution of the thesis is described as follows:

We proposed a bandwidth-rate-distortion (B-R-D) optimized motion estimation algorithm. The concept has three phases including

- 1) We propose a simple skip mode detection with content-aware scheme to find if that is a skipped MB for saving the most memory bandwidth.
- 2) We propose a bandwidth-rate-distortion (B-R-D) optimize modeling method to decide a feasible search range (SR) while can dynamically meet the available bandwidth and maximize the coding efficiency.
- 3) We propose a SR boundary prediction method to determine a feasible SR boundary for SR refinement.

1.3. Thesis organization

In chapter 2, we give an overview of the environment-aware motion estimation algorithms. In chapter 3, we propose a B-R-D optimized ME algorithm to maximize rate distortion efficiency while can dynamically meet the available bandwidth. In chapter 4, we show the simulation result and analysis. In chapter 6, we implement the hardware of the B-R-D optimized ME algorithm. Conclusion and future work are given in chapter 7.

2. Overview of environment-aware motion estimation algorithms

Motion estimation (ME) part is the most important component in H.264 encoder. In which, the variable block size integer-pel motion estimation (IME) not only contributes a lot for coding efficiency but also dominate the computation, power, and bandwidth loading of the whole encoding process. To support high performance under limited computation, power, and bandwidth, various environment-aware motion estimations have been proposed. The environment-aware motion estimation means that it has several modes of motion estimation process, and could dynamically adapts its operating configurations based on the awareness of environmental conditions, such as computation-constrained, power-constrained, bandwidth-constrained or user preferences.

In this chapter, we first introduce variable block-based motion estimation as a basis of the following sections. And then, we review the environment-aware motion estimation algorithms as follows:

- 1) Adaptive search range motion estimation
- 2) Power-aware motion estimation
- 3) Computation-aware motion estimation
- 4) Skip mode detection algorithm

2.1. Overview of variable block-based motion estimation

The block-based motion estimation is the most widely used motion estimation method for video coding, since most of the pictures are normally rectangular in shape and block-division can be easily done. In H.264 [2], the standard adopts hierarchical variable block size motion estimation technique to improve the accuracy. Fig. 2-1(a) shows the

mode hierarchy and Fig. 2-1(b) shows the mode type and its block size. In one frame, it consists of several macroblocks (MB), which are “16 by 16” pixels square. In one macroblock, it can be divided into four “8 by 8” pixels 8x8 blocks. And within one 8x8 block, it can be further divided into four “4 by 4” pixels 4x4 blocks. Fig. 2-2 illustrates the shape of various block size as listed in Fig. 2-1(b). For the video with complex textures, the smaller blocks will provide better coding efficiency but with more motion vectors. In contrast, as for the video with smooth textures, the larger blocks will provide better coding efficiency with fewer motion vectors.

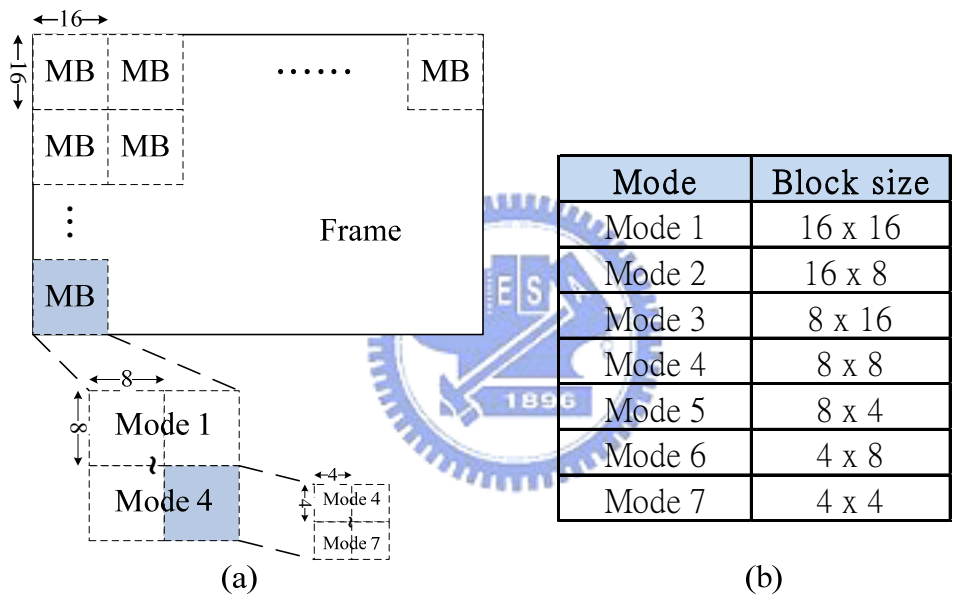


Fig. 2-1 (a)The mode hierarchy and (b) its block size for H.264

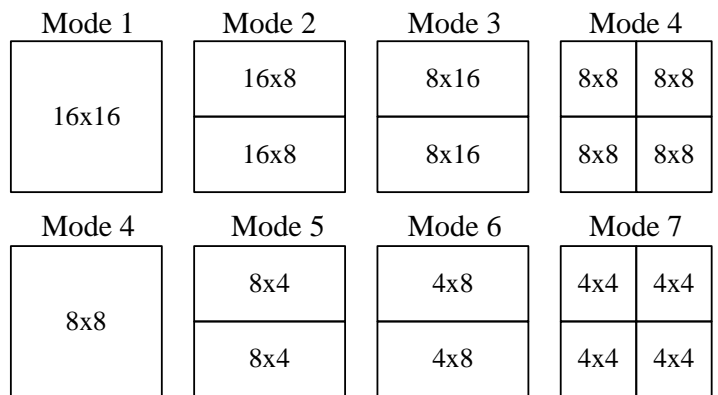


Fig. 2-2 Different modes for H.264 motion estimation

2.2. Review of adaptive search range motion estimation

Because of the motion estimation in H.264 induces a high computational complexity and leads high power consumption. Several motion estimation algorithms for low complexity and low power have been proposed. However, most of the traditional fast motion estimation algorithms reduce the complexity or power with more or less image quality sacrifice that compared with the full search motion estimation. For this reason, there is an approach to reduce the complexity or power by adjusting the search range size to suit the motion level of a video sequence. In Tian's algorithm [4] and In Toru's algorithm [5], an appropriate search range is determined on the basis of neighboring motion vectors (MV) (i.e. as shown in Fig. 2-3) and prediction errors due to spatial correlation between neighboring blocks and current block. And in Shih's algorithm [6], it adjusts the horizontal and vertical search ranges independently since there have no relationship between horizontal motion and vertical motion. In addition, to serve different resolution video content, Wang's algorithm [7] particularly adjusts the search range on the basis of the quantization parameter and the input size. In above algorithms, narrow search ranges are chosen for slow motion to reduce the complexity and power without quality degradation while wide search ranges are chosen for high motion to maintain the quality.

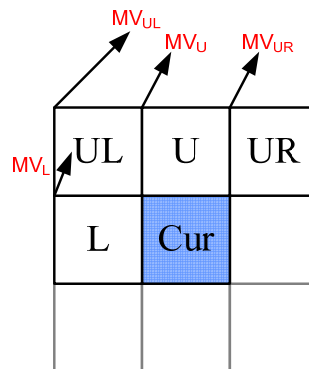


Fig. 2-3 Search range prediction using neighboring vectors

2.3. Review of power-aware motion estimation

Power-aware design concept has been introduced recently due to supporting high computation and high bandwidth on mobile devices. A power-aware design is not only a low power design, but also aware the environment to execute the functions with limited power. Traditional ME design is considered for worst case, and therefore always uses full energy no matter whether the execution is easy or difficult. However, it leads on unnecessary power consumption and shortens the lifetime of devices. Thus to fully utilize the available power in an efficient way, several power-aware designs have been proposed. In [8], it focuses on the introduction of power-aware concepts and considerations to the architecture design of a video coder as shown in Fig. 2-4, including the discussions of power-aware motion estimation and discrete cosine transform. And in motion estimation, it adopts several fast algorithms, and several skills like bit truncation scheme and sub-sampling for multiple power modes support. In [9] and [10], they propose a dedicated hardware with reconfigurable macroblock pipelining architecture for adopting its motion estimation pre-skip algorithm. Through the pre-skip algorithm, the power can be efficiently utilized, thus the power scalability can be improved for more power management. And in [11], it develops a power-rate-distortion (P-R-D) model for optimizing the rate-distortion (R-D) behavior under the power constraints. By using the P-R-D model, given a power supply level and a bit rate, the power-scalable video is able to find the best configuration of complexity control to maximize the video quality.

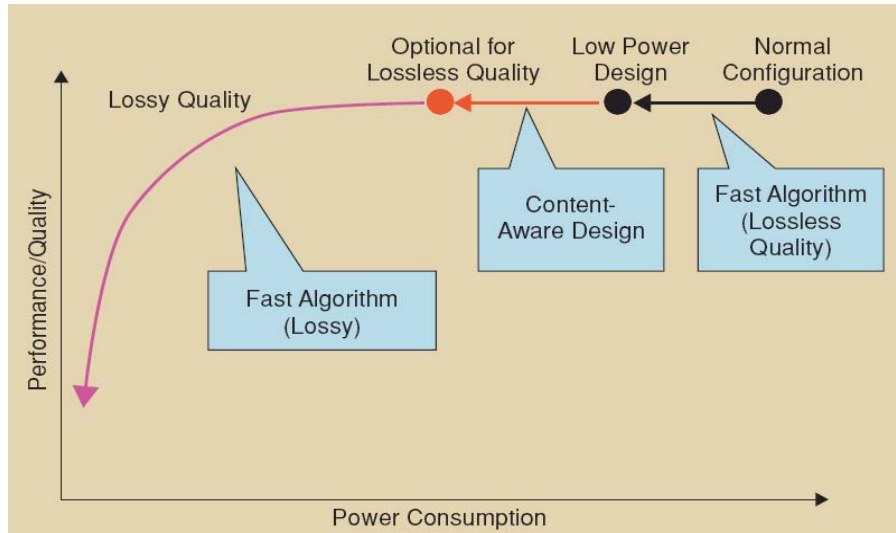


Fig. 2-4 Power aware multimedia systems [8]

2.4. Review of computation-aware motion estimation

Many fast algorithms reduce the computation complexity of motion estimation to meet the computation constraints, and thus lead to significant quality loss. Therefore, several computation-aware ME algorithms have been proposed while can dynamically adjust the target function under limited computation resource. In [12], its proposed computation-aware scheme can dynamically determine the target computation which is allocated to each frame, and then to each block in a computation-distortion-optimized manner. The mean-square-error difference obtained from initial motion vector and best motion vector is regarded as a distortion gain measure under computation constraints, and thus can achieve better coding efficiency by adopting its computation-aware scheme. In [13], it develops a complexity-rate-distortion framework, which extends the traditional R-D analysis by including another dimension, the computation complexity. This framework determines for each MB which partitions are likely to be optimal and motion vector search is only carried out for only the selected partitions, thus reducing the complexity of the ME step. In [14], Through investigating various issues in H.264, such as

complexity prediction methods, MB complexity scaling and time scheduling algorithms, it proposes a method based on dynamic control of the encoding parameters to meet real-time constraints while minimizing coding efficiency loss. In [15], it uses the sum of absolute components of predict motion vectors to help allocating the available computation to a frame, and then the computation to a frame is distributed to MBs. And in [16], it presents a complexity aware motion estimation for H.264 based on pixel representation of different bit-depth and a simple scene change detection module.

2.5. Review of skip mode detection algorithm

In MPEG-4 AVC/H.264 video coding, integer-pel motion estimation (IME) and fraction-pel motion estimation (FME) contribute a lot for coding efficiency due to new techniques, such as variable block size and six-tap interpolation filter. However, these new complex techniques make ME dominate the computational loading and power of the whole encoding process up to 96% [15]. The most efficient way to lower the complexity and power of ME is to directly skip the MB encoding and simply denotes it with skip mode if the encoding situation is allowed. Therefore, as long as we can predict the skip mode before ME, we can skip the whole coding stage and save encoding power of this skipped MB. And in H.264/AVC, the MB will be skipped without encoding the motion vectors and residues and is denoted as skip mode if the following conditions are matched:

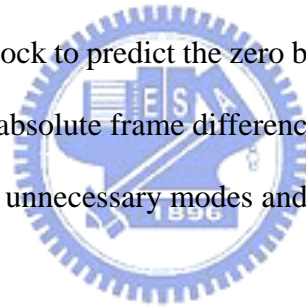
- 1) The chosen block type is 16x16 (Mode 1).
- 2) The best motion vector equals the predicted motion vector (MVP).
- 3) The chosen frame is the previous frame.
- 4) All coefficients are zero after transformation and quantization.

2.5.1. Lagrangian cost motion estimation

In [18], [19] and [20], they propose a skip prediction through Lagrangian cost function. The paper [18] uses a Lagrangian rate-distortion cost function which incorporates an adaptive model for the Lagrangian multiplier parameter base on local sequence statistics. The paper [19] predicts the Lagrangian multiplier parameter from the statistical dependency of previous co-located block. And the paper [20], the skip decision is based on a partially computed SAD metric combined with utilization of the Lagrangian cost function from the previous frame.

2.5.2. All zero DCT blocks detection

In [21], [22] and [23], they perform a comprehensive analysis of the dynamic properties of the DCT and quantization in H.264. They use several partial sum of absolute differences (SADs) in a 4x4 block to predict the zero blocks in various conditions. And in [24], a classifier based on the absolute frame differences has been employed to detect the zero blocks to effectively skip unnecessary modes and reference frames.



3. Proposed B-R-D optimized motion estimation algorithm

Motion estimation (ME) part is the most computational and memory intensive component in H.264 encoder. Traditional ME design focuses on its rate distortion performance, and thus assumes a worst case memory bandwidth requirement to the whole system. However, such assumption ignores the realistic facts of diverging contents and varying available memory bandwidth in a whole system. Diverging contents imply worst case requirement to be an overdesign or waste. Varying bandwidth could limit the available data and thus degrade the video quality or fail the real time constraints. Thus, in this thesis, we propose a rate-distortion optimized motion estimation design while can dynamically meet the available bandwidth, which is called bandwidth-rate-distortion (B-R-D) optimized motion estimation.

The rest of chapter is organized as follows. We will first introduce the whole B-R-D optimized ME algorithms. Then we will discuss each part in details in the rest of the chapter.

3.1. Introduction

The overall B-R-D optimized ME is shown in Fig. 3-1. This algorithm is developed with the following concepts. First, the target problem is to develop rate distortion optimized motion estimation under the available memory bandwidth constraint. To make the maximum use of the bandwidth, we first adopt a simple skip mode detection to find if that is a skipped MB. A skipped MB implies the lowest memory bandwidth ME (zero search range) and thus gives other complex MB more bandwidth for better quality. Thus, for other non-skipped MBs, they go through two steps for optimization: bandwidth prediction and bandwidth evaluation. Note that bandwidth is determined by the search range. Thus, bandwidth prediction is first determined by initial search range boundary prediction and refined by the current available bandwidth with the proposed B-R-D model. Then the B-R-D optimized search range is used for the current MB calculation and the resulted B-R-D data is used for bandwidth evaluation for further refinement in the next MB.

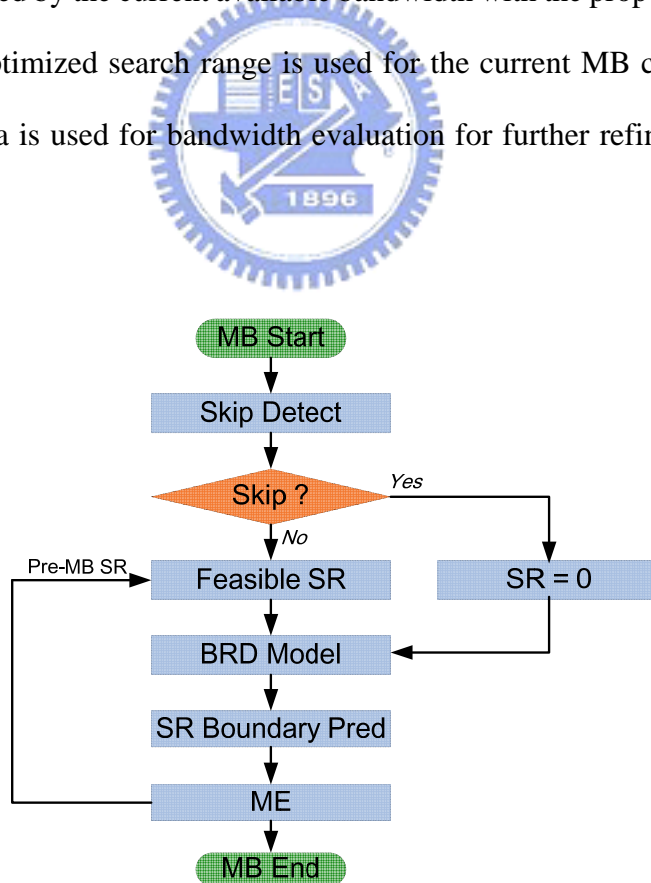


Fig. 3-1 The total BRD optimized motion estimation algorithm flow

3.2. Proposed skip mode detection with content-aware scheme

This algorithm is based on our previous work [23] with some refinement. The object of the algorithm is to detect a zero MB with content-aware scheme, and save the most memory bandwidth for other complex MB coding. The whole algorithm is illustrated in Fig. 3-2. First, we detect whether a 4x4-block is zero or not by a refined SAD-4x4-block threshold, and count the number of zero-4x4-blocks in a MB. If the number is larger than MB-zero-block threshold, we refer to this current MB as a zero MB and skip this coding. In addition, to avoid above SAD threshold affected by local large variations, we adopt a spike threshold to remove such cases for more accurate detection. More details are described in the following.

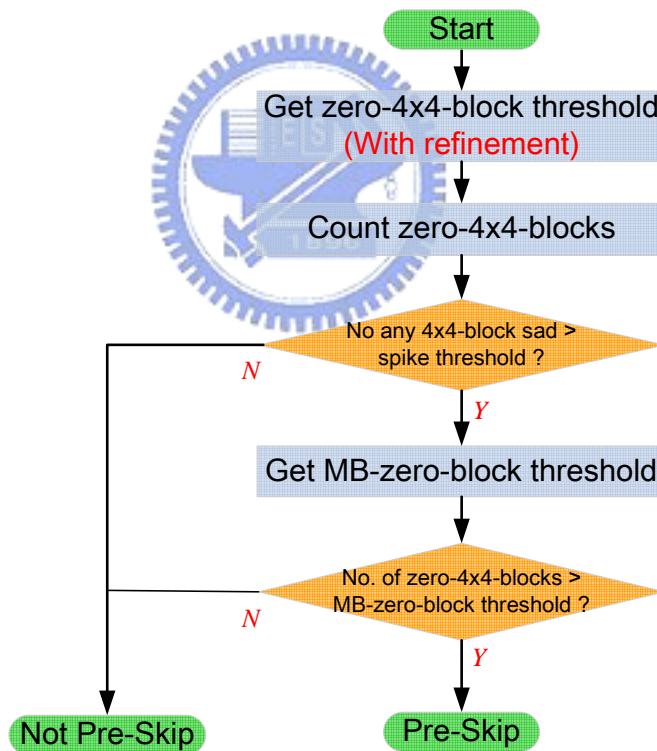


Fig. 3-2 Skip mode detection flow

3.2.1. Review of SAD-4x4-block threshold

From our previous work [23], the SAD-4x4-block threshold is used to decide if a 4x4-block is zero. We determine the threshold by analyzing the distribution of the

4x4-block SADs higher than the must-be-zero-block threshold [22](Denote it by T0), but also quantized to zero block in skipped MB. We analyze five 300-frame CIF size test sequences to determine this threshold as shown in TABLE 3-1. In which, the “mean”, variance, “maxima” stand for the average, standard variation and maxima values of 4x4-blocks whose SADs are higher than T0. The boundary that we refer to as the SAD-4x4-block threshold is the summation of mean and variance.

From TABLE 3-1, we can find that almost 85.9% in average of the 4x4-block SADs in one skip MB is less than the boundary. When the SAD of the 4x4-block is less than the boundary, we refer to the 4x4-block as a zero block. Therefore, we choose the SAD-4x4-block threshold to prevent from large prediction error. And the SAD-4x4-block threshold under different QPs is shown in TABLE 3-3.

TABLE 3-1 Boundary determination of QP 28 (mean, variance, boundary and maxima for the 4x4-block SAD distribution which higher than T0)

QP 28	Mean	Variance	Boundary	Maxima
Akiyo	44.3	10.6	55	111
Mother	45.1	9.9	55	97
Foreman	45.4	10.3	56	103
Football	50.8	11.4	62	103
Silence	55.9	12.8	69	109

TABLE 3-2 Spike threshold under different QP

	QP20	QP24	QP28	QP32	QP36
Spike threshold	37	66	97	160	230

TABLE 3-3 Boundary determination under different QP

	QP20	QP24	QP28	QP32	QP36
Akiyo	22	34	55	91	139
Mother	23	37	55	86	131
Foreman	23	38	56	91	142
Football	29	45	62	94	134
Silence	30	47	69	101	144

3.2.2. Refinement of SAD-4x4-block threshold

In this section, we analyze the relationship between prediction error and SAD-4x4-block threshold to help with refining the SAD-4x4-block threshold that presented from section 3.2.1. From TABLE 3-4 and TABLE 3-5, we found that both SAD-4x4-block threshold and SAD have high correlation as shown in Fig. 3-3. According to above relationship, we make a list including SAD and boundary information as shown in Fig. 3-4(a), and use this list to dynamically adjust SAD-4x4-block threshold from the prediction error under different QP as shown in Fig. 3-4(b)-(f). We can see that the SAD-4x4-block threshold is much proportion to SAD in lower QP cases. Thus such cases will make a better approximation with SAD-4x4-block threshold refinement.

TABLE 3-4 The boundary and SAD value under QP28 of different sequences

Original	Akiyo	Mother	Foreman	Football	Silence
Boundary	55	55	56	62	69
SAD	757	776	779	822	1337

TABLE 3-5 The boundary and SAD value under QP 28 of different sequences (Normalized to Akiyo)

Normalized	Akiyo	Mother	Foreman	Football	Silence
Boundary	1	1	1.018	1.127	1.254
SAD	1	1.025	1.029	1.086	1.768

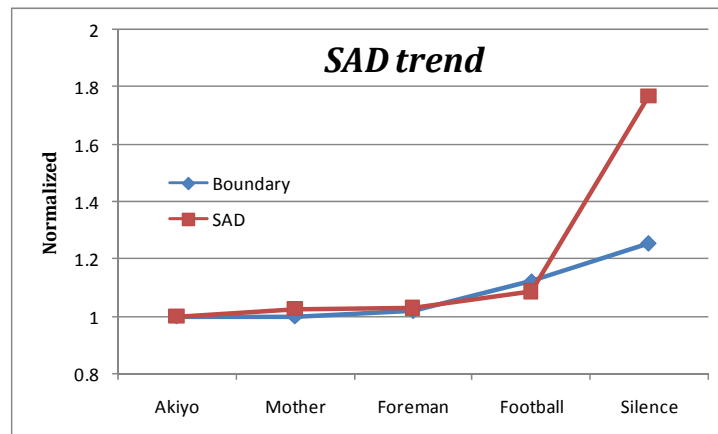
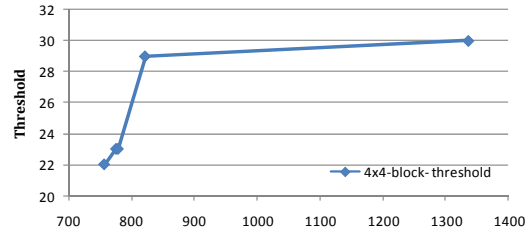


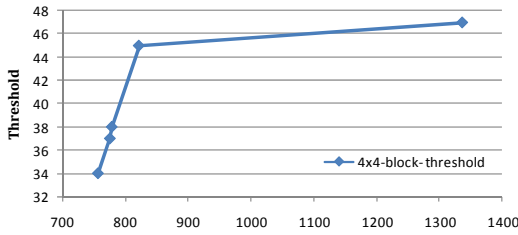
Fig. 3-3 The trend between boundary and SAD (Normalized to Akiyo)

	Akiyo	Mother	Foreman	Football	Silence
SAD	757	776	779	822	1337
QP20 thre	22	23	23	29	30
QP24 thre	34	37	38	45	47
QP28 thre	55	55	56	62	69
QP32 thre	91	86	91	94	101
QP36 thre	139	131	142	134	144

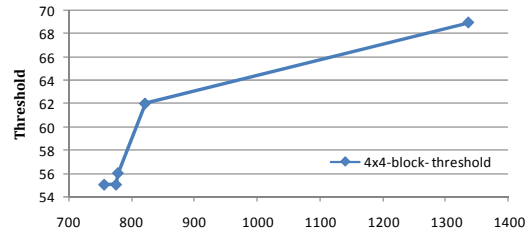
(a)

Threshold estimation - QP 20

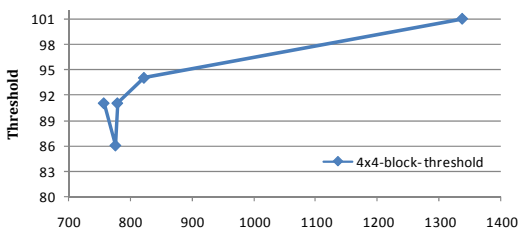
(b) SAD

Threshold estimation - QP 24

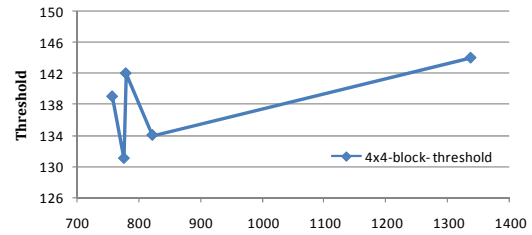
(c) SAD

Threshold estimation - QP 28

(d) SAD

Threshold estimation - QP32

(e) SAD

Threshold estimation - QP36

(f) SAD

Fig. 3-4 (a) SAD and SAD-4x4-block threshold under different QP

Threshold estimation under (b) QP20 (c) QP24 (d) QP28 (e) QP32 (f) QP36

3.3. Proposed B-R-D optimized modeling method

In this section, a bandwidth-rate-distortion (B-R-D) optimized modeling method is proposed as shown in Fig. 3-5. The method is developed with the following concepts. First, to make maximum use of the bandwidth from bus system, we transform this bandwidth into an available system search range for bandwidth budget estimation. According to the bandwidth budget, we make an appropriate bandwidth allocation for further MB coding process. Then, to justify the coding efficiency under a given bandwidth, we define a bandwidth efficiency G_{ave} up to i -th MB. And we adopt G_{ave} , rate-distortion cost, and usable bandwidth budget to make a bandwidth prediction for

keeping quality smoothness. Afterward, to make maximum use of the bandwidth budget, we make a bandwidth boundary prediction by considering the bandwidth prediction condition to determine a feasible bandwidth interval. Finally, we employ this interval and certain rate distortion data to make a search range decision and set an appropriate search range for further ME use. More details are as follows:

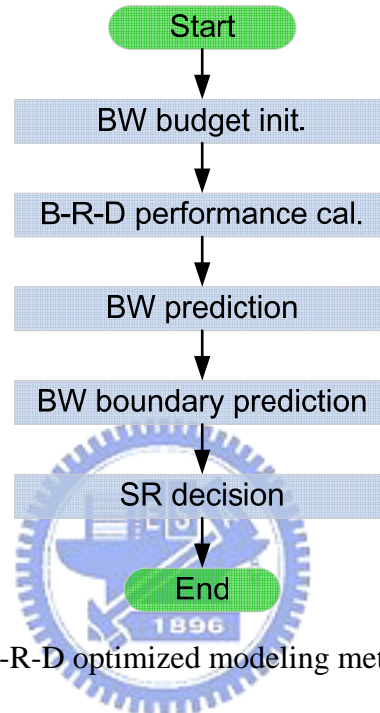
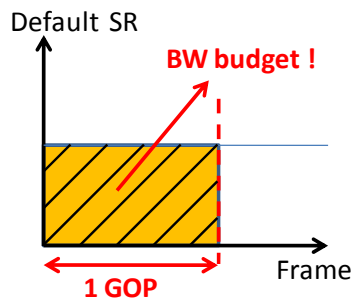


Fig. 3-5 B-R-D optimized modeling method flow

Step 1: Bandwidth (BW) budget initialization

First, according to bus status and user's preferences, we calculate the system search range for bandwidth budget estimation. Then, we initialize the bandwidth budget for bandwidth allocation in later coding process as shown in Fig. 3-6. Both system search range and bandwidth budget equation are defined as follows:



$$Default_SR = \left\lfloor \frac{\sqrt{\frac{BW_{Bus}}{Frame_rate * MBs_in_one_frame}} - 16}{2} \right\rfloor$$

$$BW_{budget} = (2 * Default_SR + 16) * (2 * Default_SR + 16) * MBs_in_one_frame * GOP_size$$

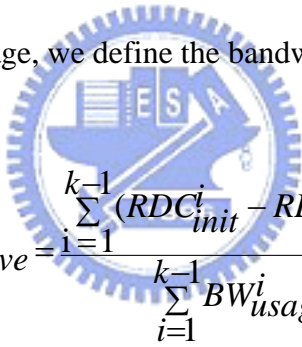
Fig. 3-6 Illustration of BW budget

$$MBs_in_one_frame * GOP_size$$

In which, the word BW_{Bus} denotes the bus data transmission rate (MBps), $Frame_rate$ denotes coded frame numbers per second, $MBs_in_one_frame$ denotes MB numbers per frame, $Default_SR$ denotes default search range in a group of picture (GOP), and GOP_size denotes frame numbers in a GOP. While coding at the beginning of GOP, our design receive data transmission rate supplied from the bus system. To make maximum use of the bandwidth from bus system, we transform this rate into a default search range, and use this default search range to estimate a bandwidth budget. Base on this bandwidth budget, we allocate appropriate bandwidth within a GOP for better quality maintain. For ease of decision, we set the bandwidth budget for a GOP with 16 frames.

Step 2: B-R-D performance calculation

To justify the bandwidth usage, we define the bandwidth efficiency G_{ave} up to i -th MB as follows.



$$G_{ave} = \frac{\sum_{i=1}^{k-1} (RDC_{init}^i - RDC_{BMA}^i)}{\sum_{i=1}^{k-1} BW_{usage}^i}$$

In which, let RDC_{init}^i denotes the rate-distortion cost that obtained using the initial MV (i.e. MVP), RDC_{BMA}^i denotes the rate-distortion cost that obtained after a motion search from block-matching algorithm (BMA) (i.e. Full search algorithm), and BW_{usage}^i denotes actual BW usage that performed in previous $k-1$ MB. G_{ave} means the average rate-distortion gain of a given bandwidth. The more G_{ave} we gain, the better coding efficiency we will perform. In the following step, we will use G_{ave} for bandwidth prediction.

Step 3: Bandwidth prediction

In this step, the objective of bandwidth prediction is to predict usable bandwidth for next MB. First, to keep the quality smoothness between the current and the previous MBs, we adopt certain data from previous MBs for further prediction as shown in Fig. 3-7. The following equation should hold:

$$RDC_{init}^k - G_{ave} BW_{BP}^k = \frac{\sum_{i=1}^{k-1} RDC_{BMA}^i}{k-1}$$

Let BW_{BP}^k be the backward bandwidth prediction. Where the left-hand side of the equation stands for the target rate distortion cost (RDC) of the current MB, the right-hand side of the equation is the averaged RDC value of the previous MB. While we obtained larger G_{ave} from the former step, it means the less bandwidth (i.e. BW_{BP}^k) we need for maintaining the rate distortion gain (RDG) of the previous MB. Therefore, the backward prediction for the current MB k can be derived as

$$BW_{BP}^k = \frac{RDC_{init}^k - \frac{\sum_{i=1}^{k-1} RDC_{BMA}^i}{k-1}}{G_{ave}}$$

In contrast to BW_{BP}^k , we define the forward prediction BW_{FP}^k for further prediction to keep the quality smoothness between the current and the future MB by adopting certain bandwidth information as shown in Fig. 3-7. The equation is as follows:

$$BW_{FP}^k = \frac{BW_{budget} - \sum_{i=1}^{k-1} BW_{usage}^i}{n - (k-1)}$$

Because we have no knowledge of the future RDG performance, and therefore the forward prediction BW_{FP}^k is equal to the remaining bandwidth budget divided by the remaining MBs in a GOP that are not coded yet.

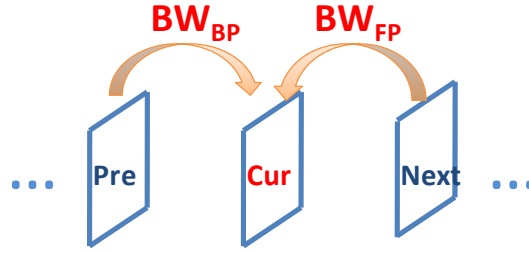


Fig. 3-7 Illustration of BW prediction

Step 4: Bandwidth boundary prediction

In this step, to make maximum use of the bandwidth budget, we make a bandwidth boundary prediction by considering the bandwidth prediction condition as mentioned previously to determine a feasible bandwidth interval as follows:

```

if ( $BW_{FP} > BW_{BP}$ ) (condition 1)
{
 $BW_{lower} = BW_{BP} + 0.5 * (BW_{FP} - BW_{BP})$ ;
 $BW_{upper} = BW_{FP} + 0.25 * (BW_{FP} - BW_{BP})$ ;
}
else (condition 2)
{
 $BW_{lower} = BW_{FP} - 0.5 * (BW_{BP} - BW_{FP})$ ;
 $BW_{upper} = BW_{FP}$ ;
}

```

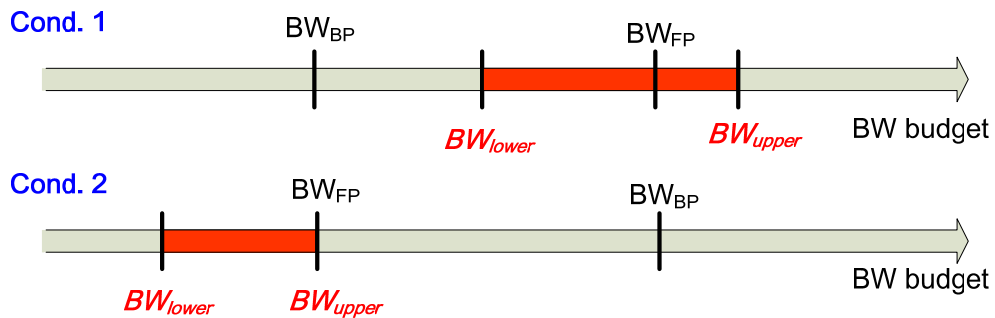
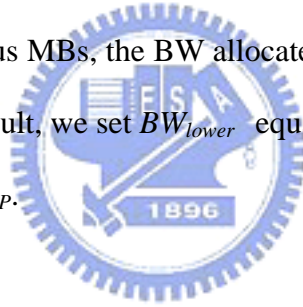


Fig. 3-8 Illustration of bandwidth boundary determination

In which, BW_{lower} and BW_{upper} denotes lower and upper bound of bandwidth usage per MB, respectively. We allow the bandwidth vary within an interval that bounded by

BW_{lower} and BW_{upper} . To consider the condition 1 as depicted above in Fig. 3-8, BW_{BP} smaller than BW_{FP} implies that insufficient BW had been allocated to the previous MBs, and thus more bandwidth could be allocated to the next MB. As a result, we set BW_{lower} equal to $BW_{BP} + 0.5 * (BW_{FP} - BW_{BP})$, and set BW_{upper} equal to $BW_{FP} + 0.25*(BW_{FP} - BW_{BP})$. To improve the coding efficiency under feasible bandwidth supply, above equations imply a reasonable allocation. In contrast, to consider the condition 2 as depicted above in Fig. 3-8, BW_{FP} smaller than or equal to BW_{BP} implies that too much bandwidth had been allocated to the previous MBs, and hence less bandwidth could be allocated to the next MB. In other words, to keep the smooth constraint under feasible BW supply, we should save bandwidth for further use. Note that although adopt BW_{BP} in bandwidth allocation for the further coding process will guarantee the average B-R-D performance as for the previous MBs, the BW allocated to the next MB is excessive that compared with BW_{FP} . As a result, we set BW_{lower} equal to $BW_{FP} - 0.5 * (BW_{BP} - BW_{FP})$, and set BW_{upper} equal to BW_{FP} .



Step 5: SR decision

Finally, we employ this interval and certain rate distortion data to make a search range decision and set an appropriate search range for further ME use. In the final step, we make a search range decision by considering bandwidth boundary interval and some rate distortion data. The search decision mainly divides into two phases as follows:

- 1) Decision for bandwidth concern
- 2) Decision for quality concern

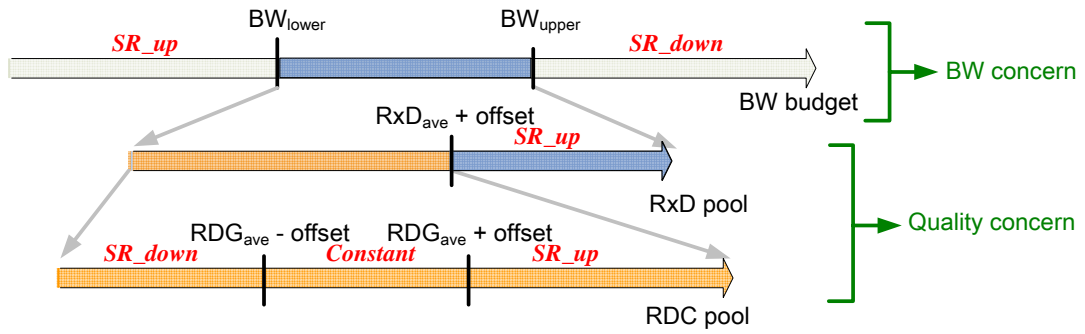


Fig. 3-9 Illustration of SR decision

The whole decision is illustrated in Fig. 3-9. In phase 1, making a search range decision under feasible BW supply is considered. If average bandwidth usage for previous MBs is more than BW_{upper} , the search range should be decreased for next MB. While if average bandwidth usage for previous MBs is less than BW_{lower} , the search should be increased for the next MB.

Phase 2 is on the other hand. If the average bandwidth usage for previous MBs is in the interval that bounded by BW_{lower} and BW_{upper} , then making a search decision under quality maintain is next to consider. If the RDG (i.e. $RDC_{init} - RDC_{BMA}$) in current MB is less than the average RDG subtracted with an adaptive offset (i.e. $RDC_{BMA}/20000$) in the previous MBs, the search range should be decreased for next MB. Because in spite of the coding is under feasible bandwidth supply, the rate distortion performance could not be maintained with previous MBs. In contrast, if the RDG in current MB is more than the average RDG added with an adaptive offset in previous MBs, the search range should be increased for next MB. Otherwise, if the RDG in current MB is in the interval that bounded by $RDG_{ave} - offset$ and $RDG_{ave} + offset$, the search range should be hold.

Meanwhile, a special case must be considered. To avoid the terrible quality loss, the search range no longer be decreased as mentioned above. Instead, the search range could be increased by checking rate multiplied distortion (RxD). If RxD in current MB is more than average $4 \times RxD$, the search range should be increased. After the search has decided,

the search window will be updated that corresponds to the new search range as shown in

Fig. 3-10.

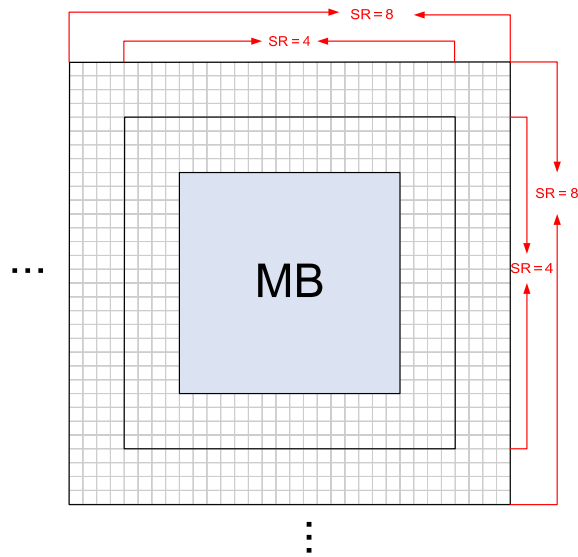


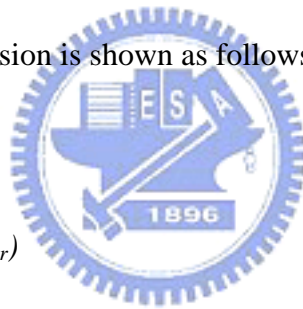
Fig. 3-10 Illustration of SR modification

The total search range decision is shown as follows:

```

if ( $BW_{ave} > BW_{upper}$ )
{
     $SR_{down} = 8$ ;
}
else if ( $BW_{ave} < BW_{lower}$ )
{
     $SR_{up} = 8$ ;
}
else
{
    if ( $RxD_{cur} > RxD_{ave} \times 4$ )
    {
         $SR_{up} = 16$ ;
    }
    else
    {
        if ( $RDC_{gain_{cur}} < RDC_{gain_{ave}} - offset$ )
        {
             $SR_{up} = 4$ ;
        }
        else if ( $RDC_{gain_{cur}} > RDC_{gain_{ave}} + offset$ )
        {
             $SR_{down} = 4$ ;
        }
    }
}

```



3.4. SR boundary prediction method

The objective of search range boundary prediction method is presented to refine the SR that had decided from section 3.3 by determining a feasible search range boundary and it could avoid unnecessary bandwidth waste for further ME use. The search range boundary prediction method is illustrated in Fig. 3-11(a). First, we get the adjacent motion vectors (MV) from neighboring blocks and current block (co-located block of previous frame), such as MV_{UL} , MV_U , MV_{UR} , MV_L , MV_{Cur} . These blocks are local maximum MV within their own blocks, and the relationship between neighboring blocks and current block are shown as Fig. 3-11(b). Second, we compare with these five local maximum MV that mentioned above, and choose a global maximum MV. Finally, we set the available search range by referring to global maximum MV for next block coding as shown in Fig. 3-11(c).

3-11(c).

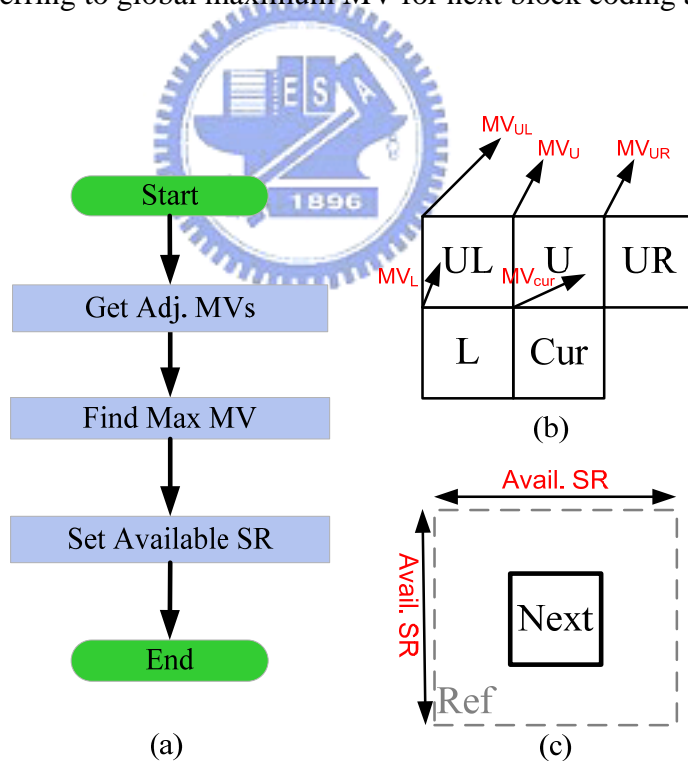


Fig. 3-11(a) Search range boundary predicted method flow

(b) The relationship between neighboring blocks and current block

(c) Example of available SR for next block coding

The search range boundary corresponding to MVs is shown as follows:

```
if (max_mv<=2) max_avail_SR = 4;  
else if(max_mv<=4) max_avail_SR = 8;  
else if(max_mv<=8) max_avail_SR = 12;  
else if(max_mv<=12) max_avail_SR = 16;  
else if(max_mv<=16) max_avail_SR = 20;  
else if(max_mv<=20) max_avail_SR = 24;  
else if(max_mv<=24) max_avail_SR = 28;  
else max_avail_SR = 32;
```

3.5. Summary

In this chapter, we propose a B-R-D optimized ME algorithm in H.264 video coding. To summarize, we first detect a MB whether it skip or not by skip mode detection with content-aware scheme. Then according to the current SR, BW status and data characteristics, we make a SR decision for next MB coding from B-R-D optimized modeling method. Finally, the SR decided before will be refined by SR boundary prediction method for further ME use. In addition, the simulation result is described in chapter 4.

4. Simulation and Analysis

In this chapter, we simulate the algorithms that proposed in chapter 3. First, we will introduce the bandwidth (BW) pattern which is used to stand for various bus systems. Second, we will show the experimental results as four phases:

- 1) Performance comparison
- 2) The distribution of MB for skip mode analysis
- 3) Timing comparison with skip detection
- 4) Completion time comparison of SR random patterns

Compared with the reference software [3], we can not only achieve better efficiency than JM 12.2, but also allow the motion estimation (ME) algorithm to be realized by external bus system. This is attributed to that our algorithm could save unnecessary bandwidth (BW) by detecting bus status, and then utilizing remaining BW to search more in the search window for finding better solution.

4.1. BW pattern setting

In this section, we introduce six different BW patterns to stand for various bus systems as shown in Fig. 4-1(a)-(f). These BW patterns are as follows, SR constant 8, SR constant 16, SR constant 24, SR random 8, SR random 16 and SR random 24. The numbers '8', '16' and '24' mentioned above stand for average SR usage which are 8, 16 and 24 respectively. The word 'constant' represents BW supply in bus system is constant; In contrast, the word 'random' represents BW supply in bus system is random. The SR 8, 16 and 24 patterns are used to fit the low, medium, and high BW design respectively. Then, the SR constant pattern is used to fit the BW design with stable bus status; In contrast, the SR random pattern is used to fit the BW design with unstable bus status.

The bandwidth usage of different SR in one MB is listed in TABLE 4-1. By adopting

these BW patterns as mentioned above, our algorithm dynamically adjust the SR (i.e. the ‘proposed’ in Fig. 4-1(a)-(f)) to meet the BW supply in bus system. While adjusting the SR, the PSNR and bit-rate also vary per frame, thus further improve the coding efficiency. Here is an example below as follows. When we coding under the system bandwidth with blue line as shown in Fig. 4-2(a), our algorithm dynamically adjust the search range with red line as shown in Fig. 4-2(a) for better quality maintain. Fig. 4-2(b) and Fig. 4-2(c) shows the PSNR and bit-rate variation under our feasible search range in 300 frames compared with JM 12.2.

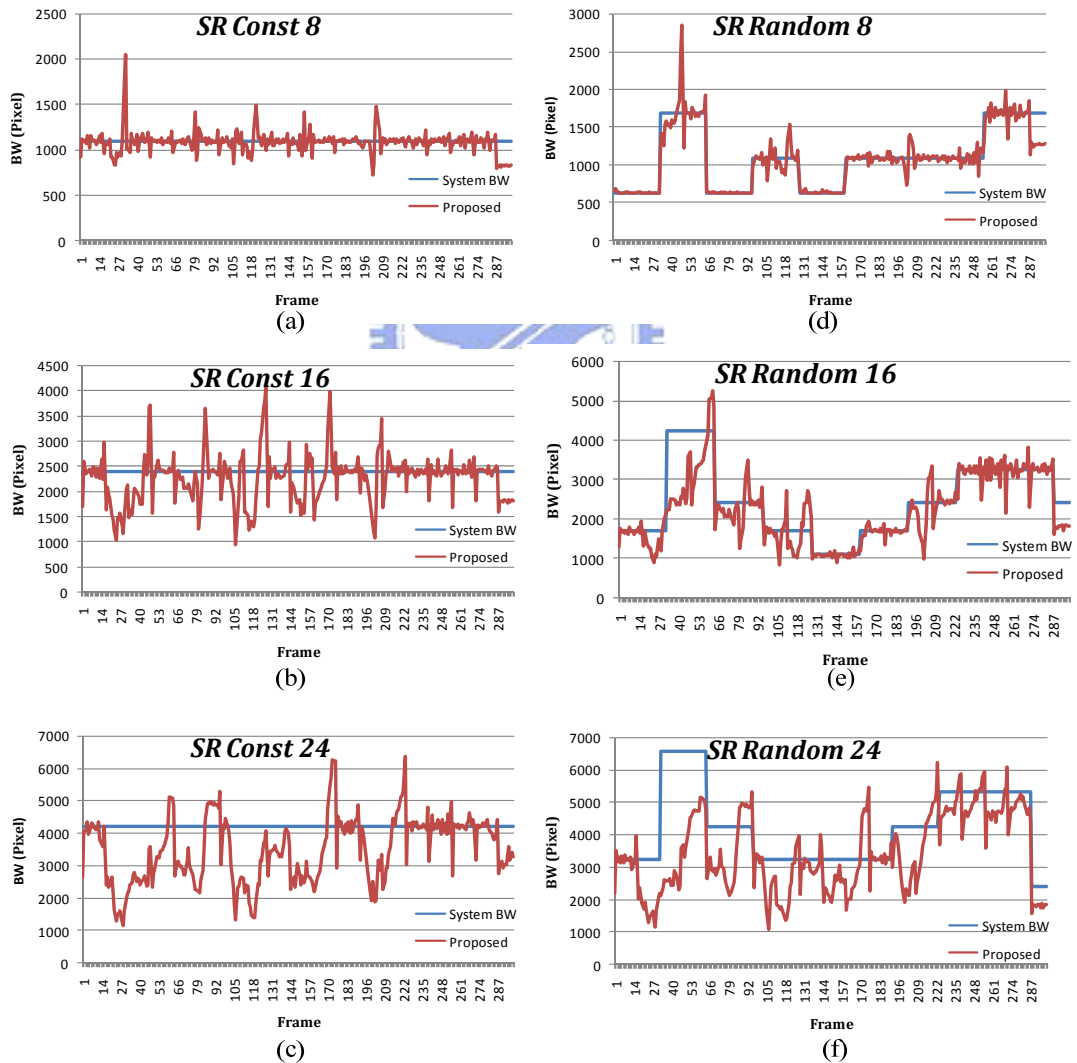


Fig. 4-1 6 kind of BW patterns: (a) SR constant 8 (b) SR constant 16 (c) SR constant 24 (d) SR random 8 (e) SR random 16 (f) SR random 24

TABLE 4-1 Bandwidth usage of one MB

	SR 4	SR 8	SR 12	SR 16	SR 20	SR 24	SR 28	SR 32	Skip
BW	625	1089	1681	2401	3249	4225	5329	6561	256

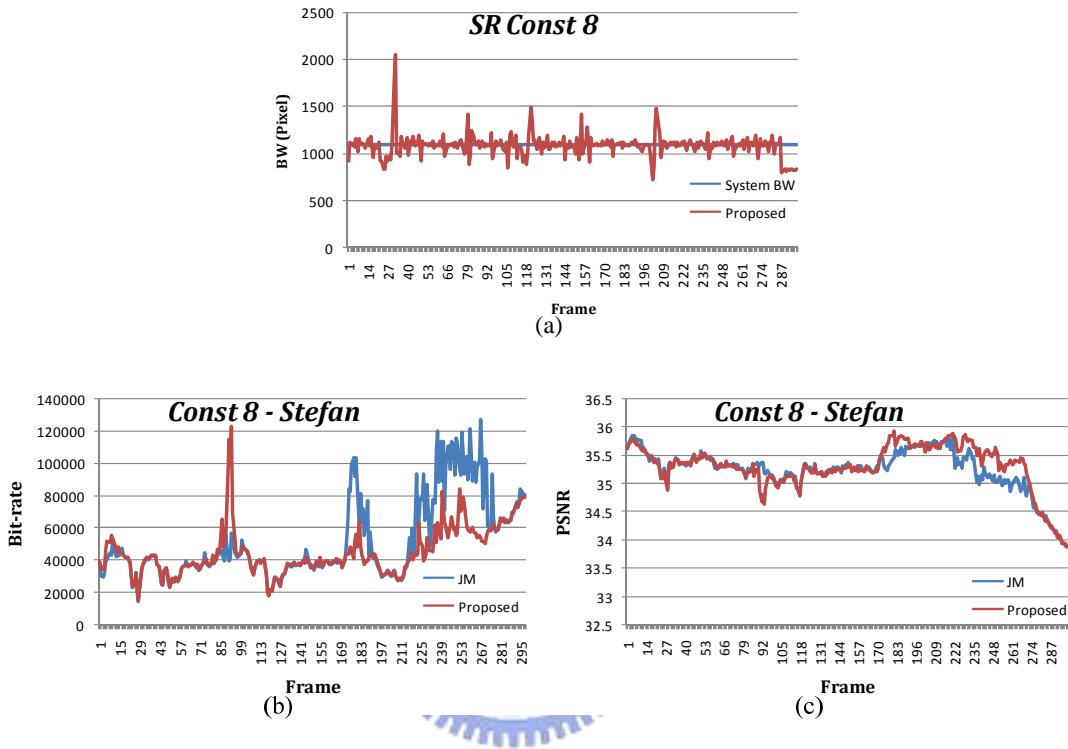


Fig. 4-2 The example of dynamically adjust the SR (a)

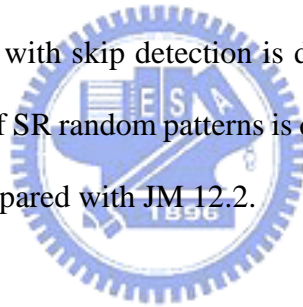
The performance example of dynamically adjust the SR: (b) PSNR (c) Bit-rate

4.2. Experimental result

The simulation environment is described below. At First, we partition the simulation into three versions, JM version, BRD version, and BRD + Skip version. The JM version stands for JM 12.2. The BRD version stands for our B-R-D optimized ME algorithm without skip detection. And the BRD + Skip version stands for our B-R-D optimized ME algorithm with skip detection.

In each of above version, then, we will show the performance under various small size test sequences. We analyze three 300-frame CIF size test sequence, including low motion, medium motion and high motion sequences for more accurate evaluation. The low motion sequence is ‘Akiyo’, and the medium motion sequence is ‘Foreman’. The high motion sequence is ‘Stefan’. In addition, we use the BW patterns mentioned from section 4.1 to stand for simulation of various bus systems.

The test environments are: baseline profile, no rate-distortion optimization, one reference frame, full-search algorithm in ME, image sequences are IPPP... (i.e. only one I-frame) and no B frames were used. The mainly performance of our design is described in section 4.2.1. The distribution of MB for skip mode analysis is described in section 4.2.2. The timing comparison with skip detection is described in section 4.2.3. And the completion time comparison of SR random patterns is described in section 4.2.4. All these following simulations are compared with JM 12.2.



4.2.1. Performance comparison

The average comprehensive performances in QP 28 of JM version, BRD version, and BRD + Skip version are listed in TABLE 4-2 – TABLE 4-7. These performances include 4 parts: BW saving, PSNR drop, bit-rate saving, and encoding time comparison. With all Tables, these reveal that the BW saving is roughly proportion to BW pattern. With high motion sequence, our design will save more bit-rate and even produce rising PSNR with low BW supply. In BRD + Skip design, the time saving with 79%, 23% and 18% is shown in “TABLE 4-2 - TABLE 4-4” or “TABLE 4-5 - TABLE 4-7”, which is in ‘Akiyo’, ‘Foreman’, ‘Stefan’ respectively.

Fig. 4-3 to Fig. 4-5 shows the BW usage, PSNR, bit-rate, and coding time of low motion, medium motion and high motion sequences respectively in SR constant patterns in QP 28. The less motion sequence is encoded, the more BW is saved in our design that compared with JM 12.2. It because we used SR boundary predicted method to analyze adjacent MV, and it could set a proper SR for saving unnecessary BW without quality loss. Both the PSNR and bit-rate curves of our BRD version design are almost the same as the original curve, and particularly is better than the original curve under high motion sequence. Since that the unnecessary memory access can be saved by fully utilize the BW while applying our B-R-D optimized process, and unexhausted BW can be used to enlarge the SR for finding better solution in the posterior MBs or frames. Thus JM12.2 is no longer the optimal algorithm if our B-R-D optimized concept is considered. In addition, plenty of coding time can be saved with negligible PSNR drop while applying our detection. Since the pre-skipped MBs are nearly to be skipped, hence these MBs do not degrade performance significantly. On the other hand, Fig. 4-6 to Fig. 4-8 shows the BW usage, PSNR, bit-rate, and coding time of low motion, medium motion and high motion sequences respectively in SR random patterns. The performance is nearly the same that compared with the SR constant patterns mentioned above.

The PSNR and bit-rate comparison in QPs of JM version, BRD version, and BRD + Skip version are listed in TABLE 4-8 – TABLE 4-25. In BRD version, most PSNR and bit-rate are hold compared with JM 12.2. And in BRD + Skip version, the PSNR slightly descend with high QP because the rising skip ratio. Fig. 4-9, Fig. 4-10, Fig. 4-12, Fig. 4-13, Fig. 4-15, Fig. 4-16, Fig. 4-18, Fig. 4-19, Fig. 4-21, Fig. 4-22, Fig. 4-24 and Fig. 4-25 shows the rate-distortion curves of low and medium motion sequences in JM version, BRD version, and BRD + Skip version. Our design has almost the same curve as original

JM curve. Fig. 4-11, Fig. 4-14, Fig. 4-17, Fig. 4-20, Fig. 4-23 and Fig. 4-26 shows the rate-distortion curves of high motion sequences in JM version, BRD version, and BRD + Skip version. Our design has better than the original JM curve, particularly in low BW cases (i.e. SR Const 8, SR Random 8). Because the SR in high motion sequence would have large variation, and the more BW limited, the more BW control needed. While coding under the BW limited situation, our design could manage the BW usage more accurately compared with JM12.2.

In summary, our BRD design could save average bit rate up to 13% and increase average PSNR up to 0.1 dB (i.e. SR Const 8 pattern of TABLE 4-4) in high motion sequence under low bandwidth constraint. While coding in low motion sequence, our BRD design could save bandwidth from 35% to 83% with almost the same performance at bit rate and PSNR (i.e. TABLE 4-2 and TABLE 4-5) under different bandwidth supply. In addition, our BRD-Skip design could almost save 80% coding time with negligible PSNR degradation (i.e. TABLE 4-2 and TABLE 4-5) under different bandwidth supply.

TABLE 4-2 Performance of BRD and BRD + Skip model under SR const pattern for “Akiyo” sequence

Akiyo	BRD				BRD + Skip			
	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$
Const 8	-35.17	-0.02	0.24	0.45	-63.73	-0.14	-1.73	-78.96
Const 16	-69.80	-0.01	-0.35	-0.57	-83.55	-0.14	-1.70	-79.20
Const 24	-82.82	-0.01	-0.45	-1.94	-90.65	-0.15	-1.80	-79.54

TABLE 4-3 Performance of BRD and BRD + Skip model under SR const pattern for “Foreman” sequence

Foreman	BRD				BRD + Skip			
	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$
Const 8	-4.78	-0.02	1.79	0.06	-4.32	-0.04	0.78	-22.89
Const 16	-22.07	-0.02	2.10	-0.32	-24.91	-0.04	1.98	-23.00
Const 24	-43.74	-0.02	1.99	-0.69	-50.65	-0.05	1.41	-23.78

TABLE 4-4 Performance of BRD and BRD + Skip model under SR const pattern for “Stefan” sequence

Stefan	BRD				BRD + Skip			
	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$
Const 8	-1.01	0.10	-13.42	0.19	-1.10	0.04	-7.92	-18.05
Const 16	-6.04	0.01	-2.45	-0.06	-6.00	0.00	-2.51	-18.24
Const 24	-17.59	0.01	-1.21	-0.38	-19.69	0.00	-1.76	-18.48

TABLE 4-5 Performance of BRD and BRD + Skip model under SR random pattern for “Akiyo” sequence

Akiyo	BRD				BRD + Skip			
	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$
Random 8	-37.83	-0.01	0.10	0.38	-63.73	-0.14	-1.73	-79.03
Random 16	-69.89	-0.02	0.14	-0.76	-83.55	-0.14	-1.70	-79.14
Random 24	-82.82	-0.01	-0.45	-2.13	-90.65	-0.15	-1.80	-79.47

TABLE 4-6 Performance of BRD and BRD + Skip model under SR random pattern for “Foreman” sequence

Foreman	BRD				BRD + Skip			
	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$
Random 8	-12.30	-0.02	2.32	0.25	-12.58	-0.04	0.93	-23.08
Random 16	-31.03	-0.03	3.23	-0.13	-34.19	-0.05	2.04	-23.44
Random 24	-45.56	-0.02	1.69	-0.88	-51.91	-0.05	2.28	-23.59

TABLE 4-7 Performance of BRD and BRD + Skip model under SR random pattern for “Stafan” sequence

Stefan	BRD				BRD + Skip			
	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$	$\Delta BW(\%)$	$\Delta PSNR(dB)$	$\Delta Bit-rate(\%)$	$\Delta time(\%)$
Random 8	-1.38	0.06	-9.28	0.26	-1.74	0.04	-7.78	-17.54
Random 16	-7.29	0.01	-2.99	0.19	-8.45	0.00	-2.99	-18.11
Random 24	-19.10	0.00	-1.69	-0.75	-20.43	0.00	-1.79	-18.61

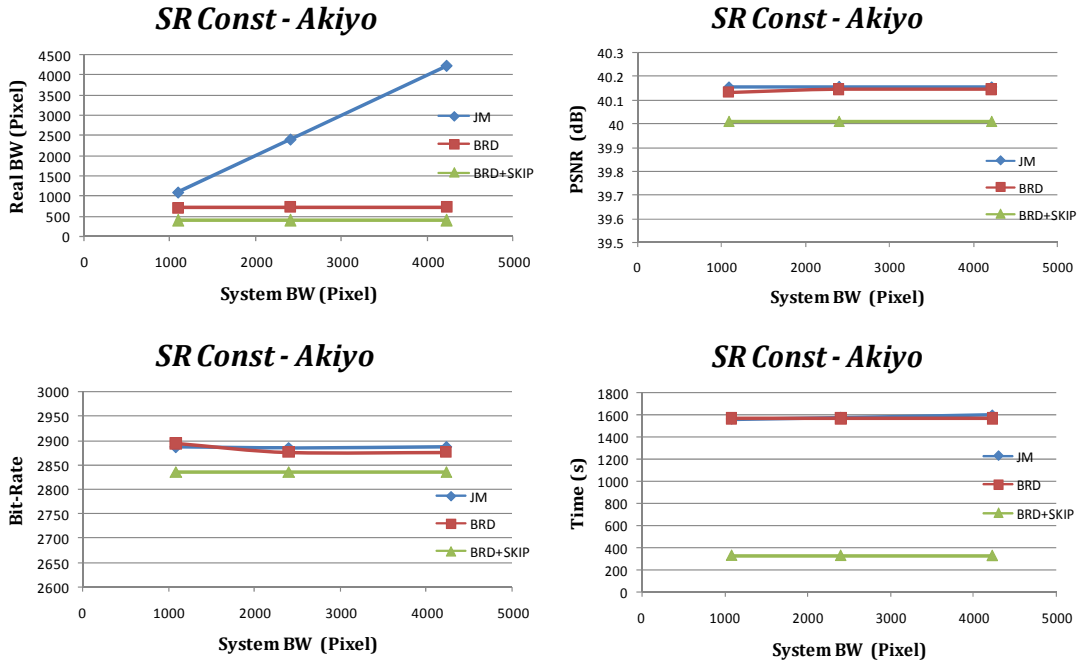


Fig. 4-3 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time under SR const pattern for “Akiyo” sequence

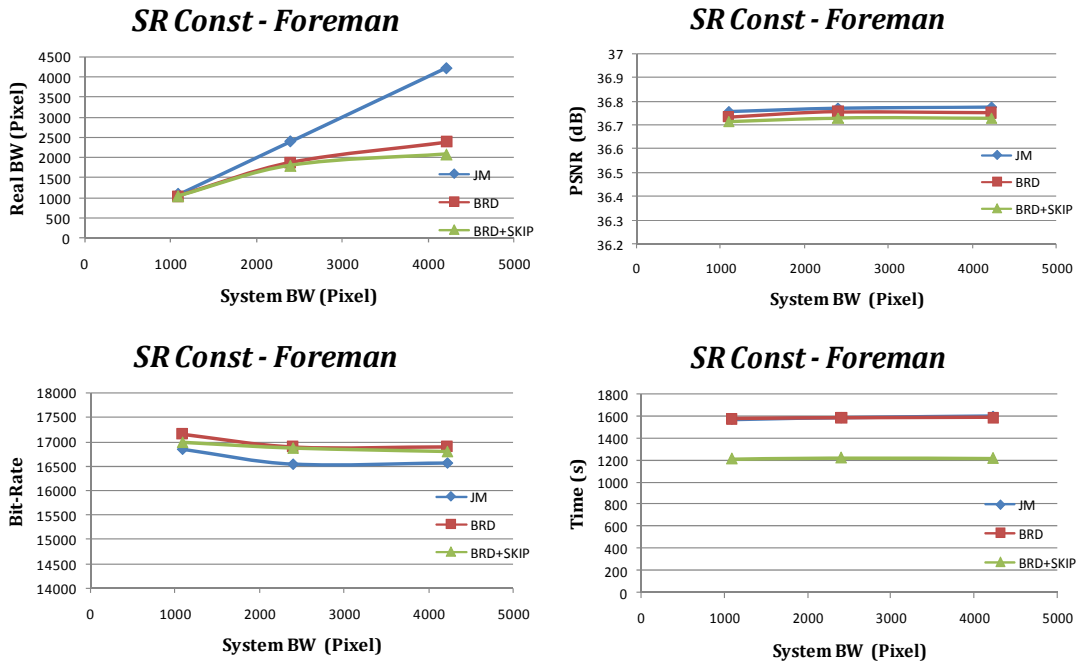


Fig. 4-4 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time under SR const pattern for “Foreman” sequence

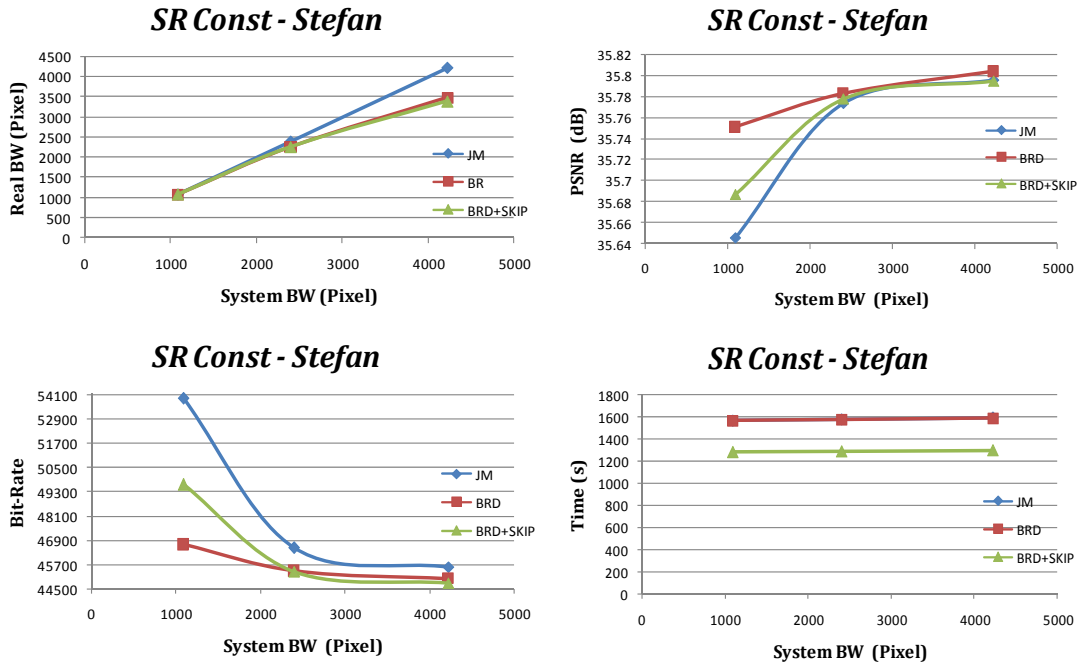


Fig. 4-5 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time under SR const pattern for “Stefan” sequence

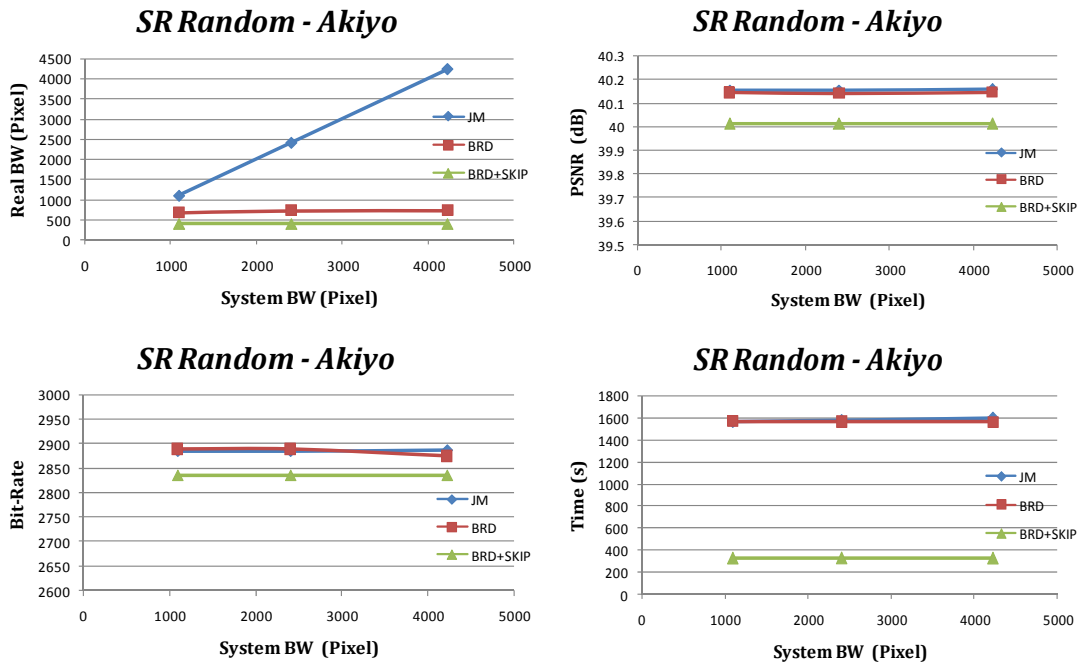


Fig. 4-6 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time under SR random pattern for “Akiyo” sequence

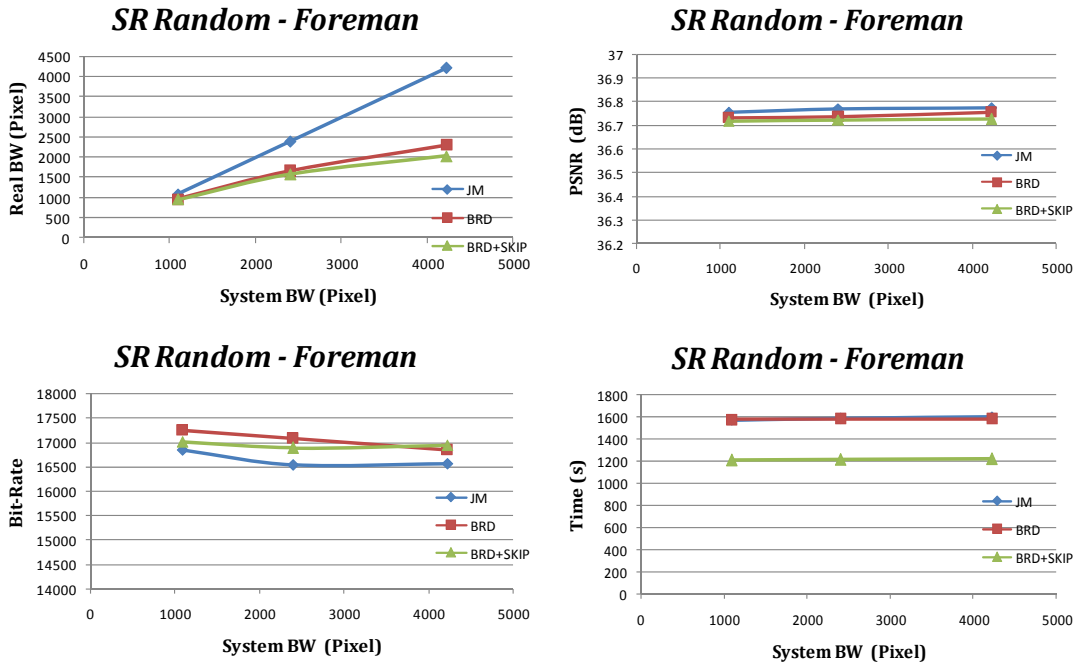


Fig. 4-7 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time under SR random pattern for “Foreman” sequence

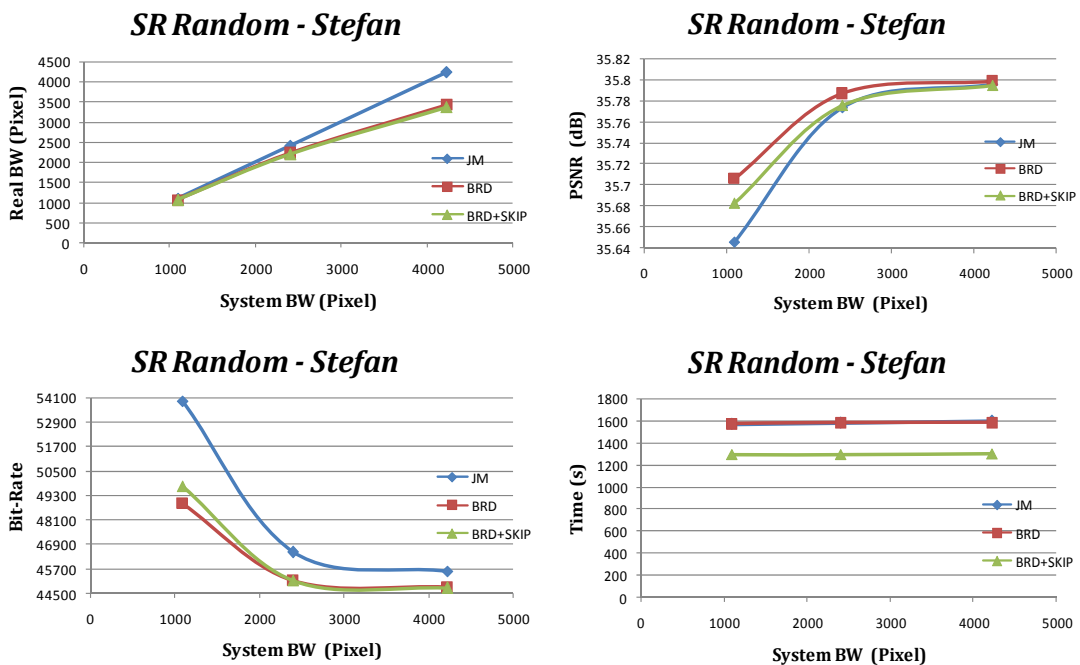


Fig. 4-8 Performance comparison in (a) BW (b) PSNR (c)Bit-rate (d) Time under SR random pattern for “Stefan” sequence

TABLE 4-8 RD comparison of BRD and BRD + Skip model under SR constant 8 for “Akiyo” sequence

Const 8 Akiyo	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.00	-0.11	-0.02	-0.17
QP 24	-0.02	0.20	-0.09	-0.85
QP 28	-0.02	0.24	-0.14	-1.73
QP 32	-0.02	-1.04	-0.24	-4.16
QP 36	0.01	0.12	-0.22	-4.71

TABLE 4-9 RD comparison of BRD and BRD + Skip model under SR constant 8 for “Foreman” sequence

Const 8 Foreman	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	-0.01	0.56	0.02	0.76
QP 24	-0.02	1.87	0.00	1.24
QP 28	-0.02	1.79	-0.04	0.78
QP 32	-0.01	1.59	-0.11	-0.25
QP 36	-0.02	2.31	-0.17	-1.24

TABLE 4-10 RD comparison of BRD and BRD + Skip model under SR constant 8 for “Stefan” sequence

Const 8 Stefan	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.04	-7.42	0.02	-3.47
QP 24	0.05	-7.23	0.04	-5.49
QP 28	0.10	-13.42	0.04	-7.92
QP 32	0.06	-8.80	0.04	-11.77
QP 36	0.06	-12.62	0.00	-13.51

TABLE 4-11 RD comparison of BRD and BRD + Skip model under SR constant 16 for “Akiyo” sequence

Const 16 Akiyo	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.00	0.05	-0.01	-0.07
QP 24	-0.01	0.00	-0.09	-0.94
QP 28	-0.01	-0.35	-0.14	-1.70
QP 32	0.01	-0.97	-0.22	-4.35
QP 36	0.02	-0.12	-0.23	-4.59

TABLE 4-12 RD comparison of BRD and BRD + Skip model under SR constant 16 for “Foreman” sequence

Const 16 Foreman	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	-0.01	1.35	0.02	1.16
QP 24	-0.01	1.82	0.00	1.61
QP 28	-0.02	2.10	-0.04	1.98
QP 32	-0.02	2.94	-0.13	1.40
QP 36	-0.05	4.20	-0.20	1.92

TABLE 4-13 RD comparison of BRD and BRD + Skip model under SR constant 16 for “Stefan” sequence

Const 16 Stefan	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.01	-2.70	0.02	-1.87
QP 24	0.01	-3.32	0.02	-2.78
QP 28	0.01	-2.45	0.00	-2.51
QP 32	0.00	-1.52	-0.04	-1.88
QP 36	-0.02	-0.01	-0.06	-1.95

TABLE 4-14 RD comparison of BRD and BRD + Skip model under SR constant 24 for “Akiyo” sequence

Const 24 Akiyo	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.00	-0.02	-0.01	-0.21
QP 24	-0.01	-0.13	-0.08	-1.07
QP 28	-0.01	-0.45	-0.15	-1.80
QP 32	-0.02	-1.17	-0.24	-4.54
QP 36	0.00	-0.12	-0.25	-4.59

TABLE 4-15 RD comparison of BRD and BRD + Skip model under SR constant 24 for “Foreman” sequence

Const 24 Foreman	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.00	0.55	0.03	1.02
QP 24	-0.01	0.67	0.00	1.31
QP 28	-0.02	1.99	-0.05	1.41
QP 32	-0.02	2.03	-0.13	0.89
QP 36	-0.03	4.48	-0.19	1.94

TABLE 4-16 RD comparison of BRD and BRD + Skip model under SR constant 24 for “Stefan” sequence

Const 24 Stefan	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	-0.01	-0.20	0.01	-0.08
QP 24	0.00	-1.13	0.01	-1.32
QP 28	0.01	-1.21	0.00	-1.76
QP 32	-0.01	0.36	-0.05	-0.08
QP 36	-0.03	2.19	-0.09	1.18

TABLE 4-17 RD comparison of BRD and BRD + Skip model under SR random 8 for “Akiyo” sequence

Rand 8 Akiyo	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.00	0.06	-0.02	-0.17
QP 24	-0.02	0.11	-0.09	-0.85
QP 28	-0.04	0.10	-0.14	-1.73
QP 32	-0.01	-1.04	-0.24	-4.16
QP 36	0.01	0.24	-0.22	-4.71

TABLE 4-18 RD comparison of BRD and BRD + Skip model under SR random 8 for “Foreman” sequence

Rand 8 Foreman	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	-0.01	0.38	0.03	0.85
QP 24	-0.02	1.59	0.00	1.36
QP 28	-0.02	2.32	-0.04	0.93
QP 32	-0.02	2.49	-0.12	0.25
QP 36	-0.04	2.82	-0.19	-1.10

TABLE 4-19 RD comparison of BRD and BRD + Skip model under SR random 8 for “Stefan” sequence

Rand 8 Stefan	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.03	-6.43	0.03	-3.75
QP 24	0.05	-7.22	0.03	-4.74
QP 28	0.06	-9.28	0.04	-7.78
QP 32	0.07	-11.02	0.05	-12.88
QP 36	0.06	-11.06	-0.01	-13.25

TABLE 4-20 RD comparison of BRD and BRD + Skip model under SR random 16 for “Akiyo” sequence

Rand 16 Akiyo	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.00	-0.02	-0.01	-0.07
QP 24	-0.01	0.04	-0.09	-0.94
QP 28	-0.02	0.14	-0.14	-1.70
QP 32	0.01	-0.97	-0.22	-4.35
QP 36	0.02	0.47	-0.23	-4.59

TABLE 4-21 RD comparison of BRD and BRD + Skip model under SR random 16 for “Foreman” sequence

Rand 16 Foreman	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	-0.01	0.71	0.02	1.25
QP 24	-0.01	1.76	-0.01	2.20
QP 28	-0.03	3.23	-0.05	2.04
QP 32	-0.02	3.84	-0.13	1.58
QP 36	-0.06	4.53	-0.19	1.83

TABLE 4-22 RD comparison of BRD and BRD + Skip model under SR random 16 for “Stefan” sequence

Rand 16 Stefan	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.01	-2.68	0.02	-2.15
QP 24	0.02	-3.56	0.02	-2.70
QP 28	0.01	-2.99	0.00	-2.99
QP 32	0.00	-1.79	-0.03	-2.04
QP 36	-0.02	-0.35	-0.08	-1.05

TABLE 4-23 RD comparison of BRD and BRD + Skip model under SR random 24 for “Akiyo” sequence

Rand 24 Akiyo	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	0.00	-0.07	-0.01	-0.21
QP 24	-0.01	-0.04	-0.08	-1.07
QP 28	-0.01	-0.45	-0.15	-1.80
QP 32	-0.02	-1.17	-0.24	-4.54
QP 36	0.00	-0.12	-0.25	-4.59

TABLE 4-24 RD comparison of BRD and BRD + Skip model under SR random 24 for “Foreman” sequence

Rand 24 Foreman	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	-0.01	0.55	0.02	1.02
QP 24	-0.01	1.04	0.00	1.37
QP 28	-0.02	1.69	-0.05	2.28
QP 32	-0.02	1.83	-0.13	0.92
QP 36	-0.02	3.28	-0.19	1.94

TABLE 4-25 RD comparison of BRD and BRD + Skip model under SR random 24 for “Stefan” sequence

Rand 24 Stefan	BRD		BRD + Skip	
	Δ PSNR (dB)	Δ Bit-rate (%)	Δ PSNR (dB)	Δ Bit-rate (%)
QP 20	-0.01	-0.30	0.01	-0.32
QP 24	0.00	-1.77	0.01	-1.80
QP 28	0.00	-1.69	0.00	-1.79
QP 32	-0.02	0.46	-0.05	0.27
QP 36	-0.03	2.12	-0.08	1.47

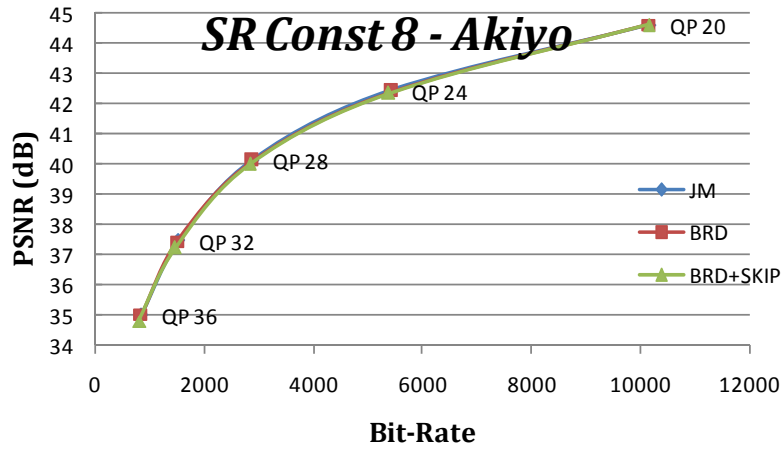


Fig. 4-9 RD curve comparison under SR constant 8 for “Akiyo” sequence

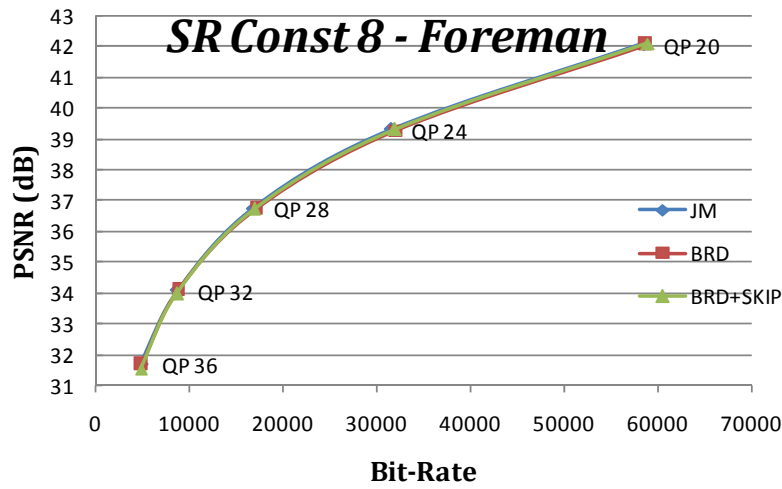


Fig. 4-10 RD curve comparison under SR constant 8 for “Foreman” sequence

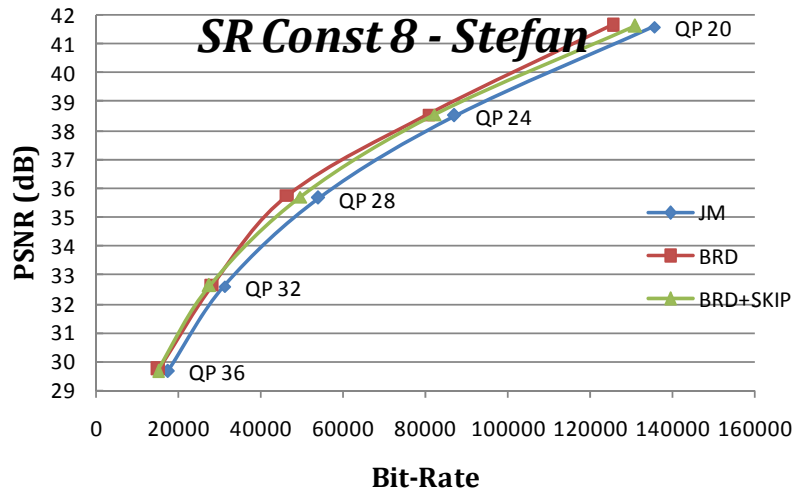


Fig. 4-11 RD curve comparison under SR constant 8 for “Stefan” sequence

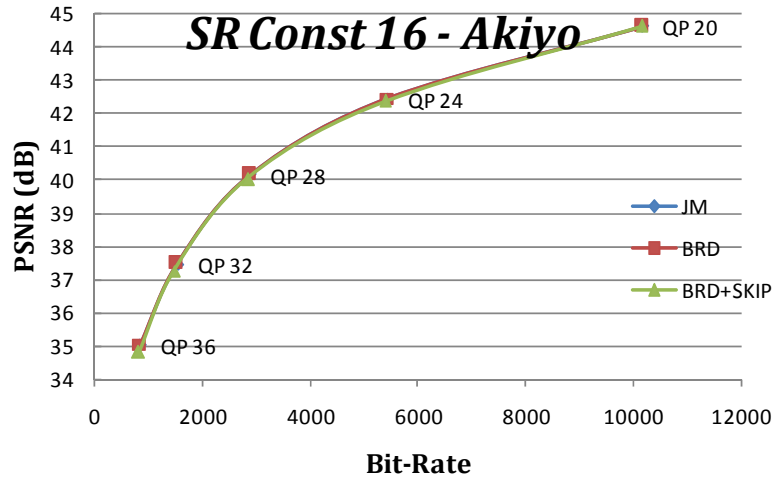


Fig. 4-12 RD curve comparison under SR constant 16 for “Akiyo” sequence

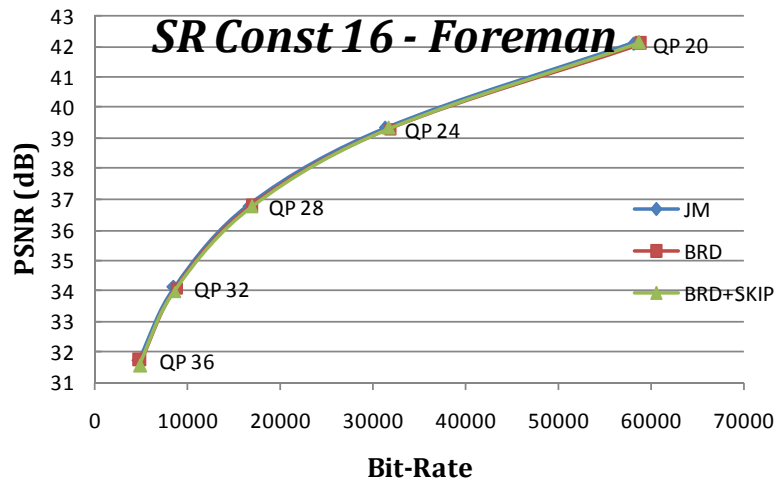


Fig. 4-13 RD curve comparison under SR constant 16 for “Foreman” sequence

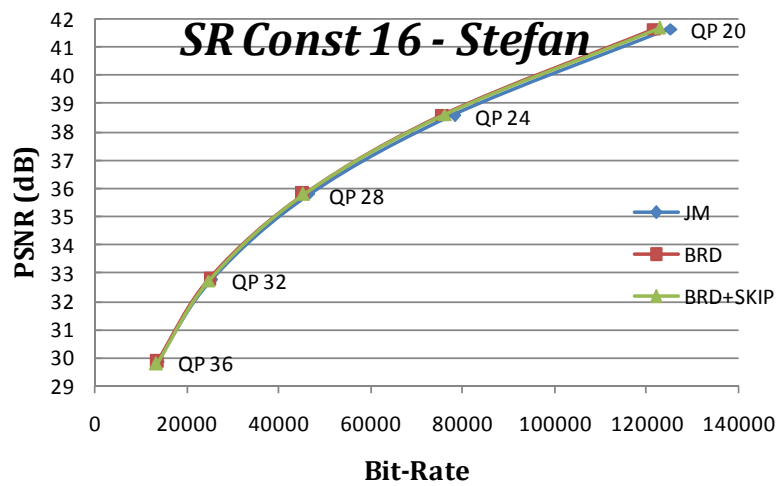


Fig. 4-14 RD curve comparison under SR constant 16 for “Stefan” sequence

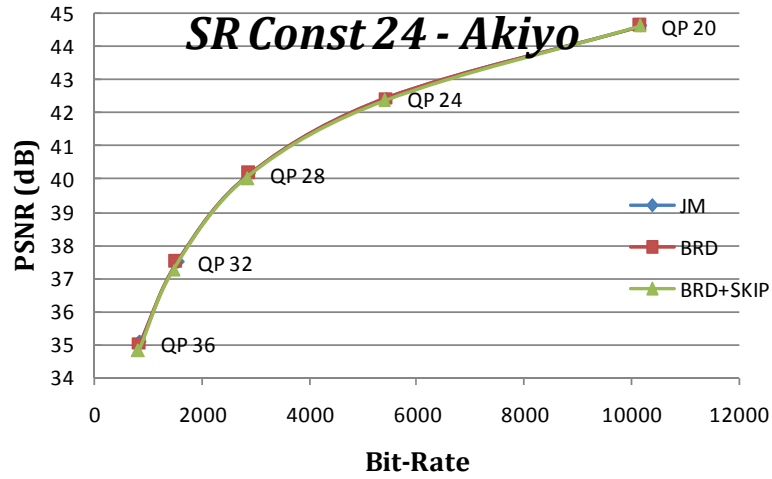


Fig. 4-15 RD curve comparison under SR constant 24 for “Akiyo” sequence

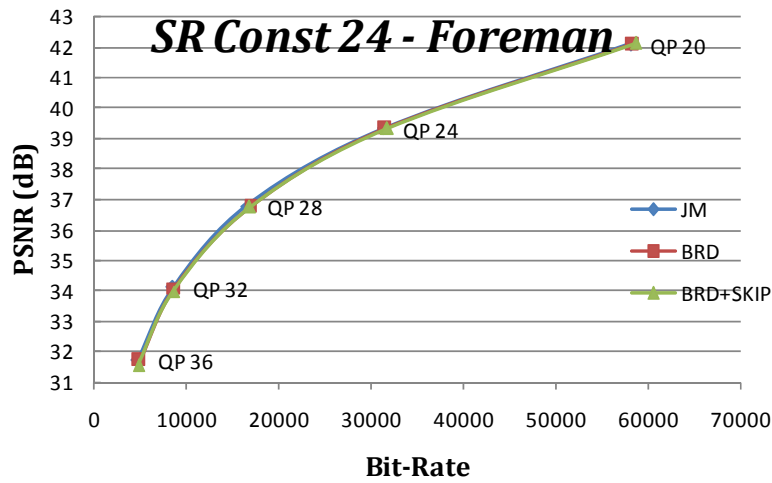


Fig. 4-16 RD curve comparison under SR constant 24 for “Foreman” sequence

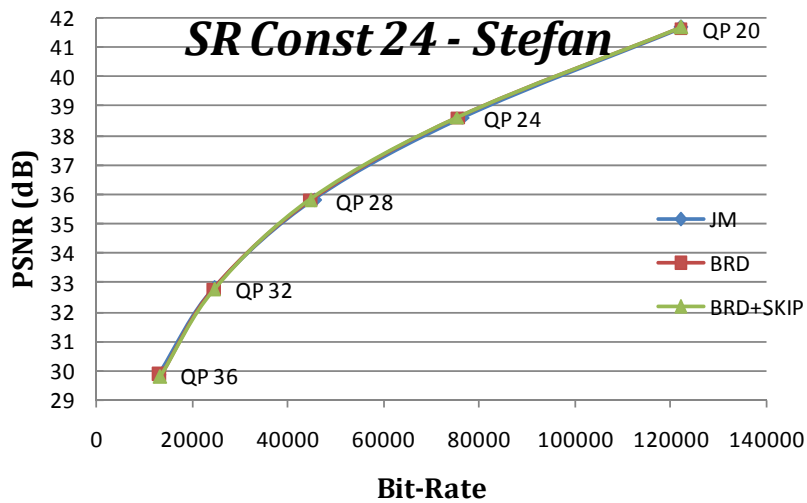


Fig. 4-17 RD curve comparison under SR constant 24 for “Stefan” sequence

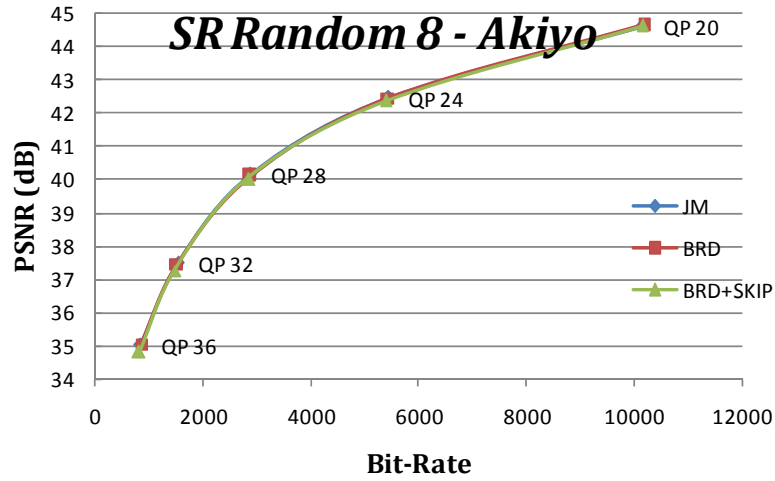


Fig. 4-18 RD curve comparison under SR random 8 for “Akiyo” sequence

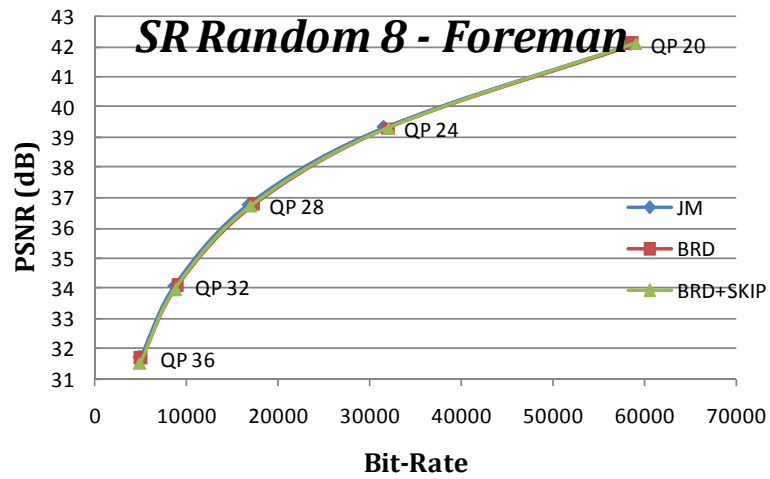


Fig. 4-19 RD curve comparison under SR random 8 for “Foreman” sequence

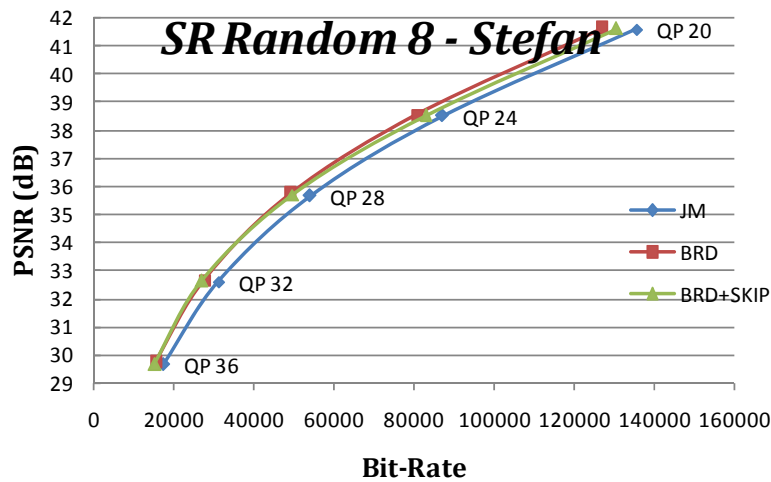


Fig. 4-20 RD curve comparison under SR random 8 for “Akiyo” sequence

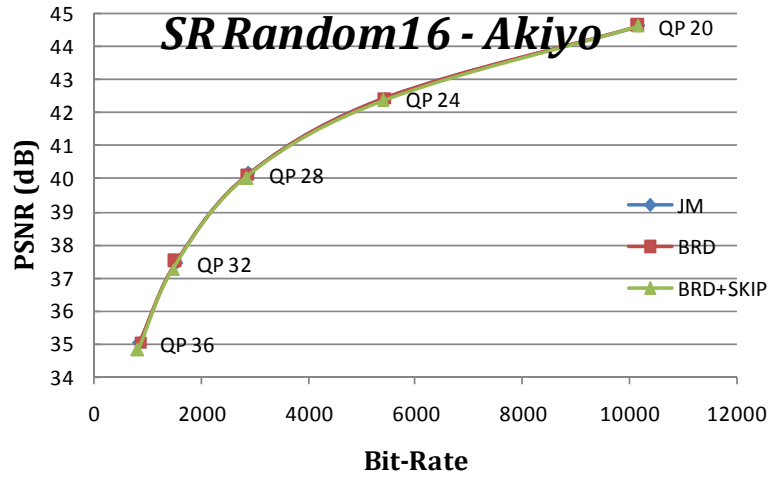


Fig. 4-21 RD curve comparison under SR random 16 for “Akiyo” sequence

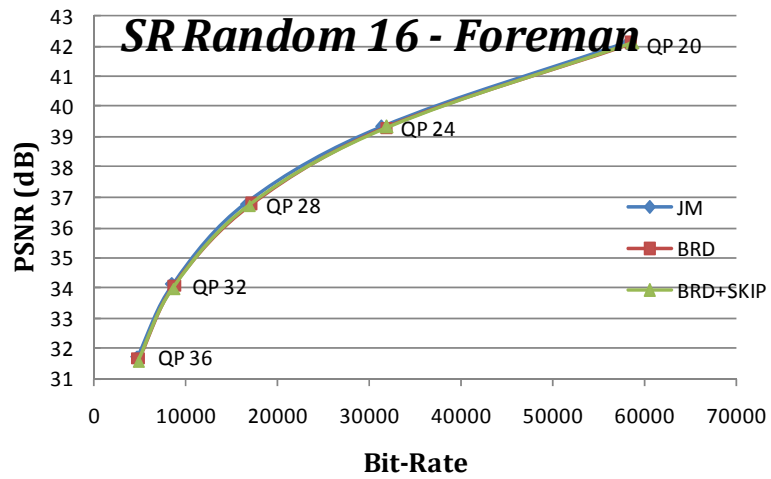


Fig. 4-22 RD curve comparison under SR random 16 for “Foreman” sequence

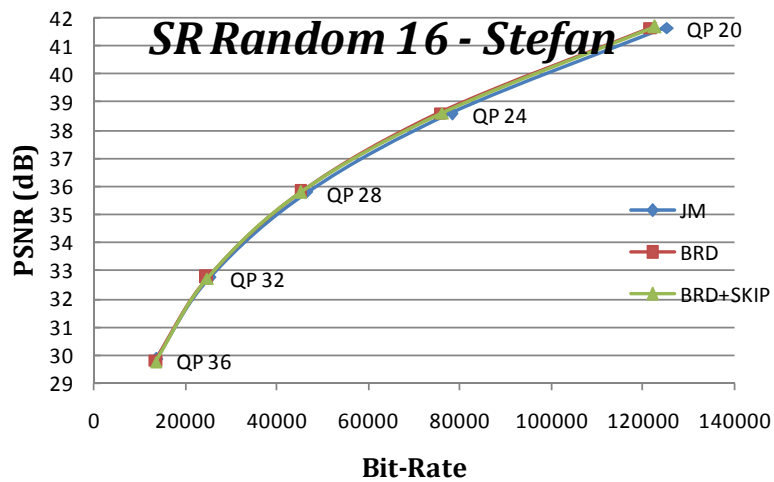


Fig. 4-23 RD curve comparison under SR random 16 for “Stefan” sequence

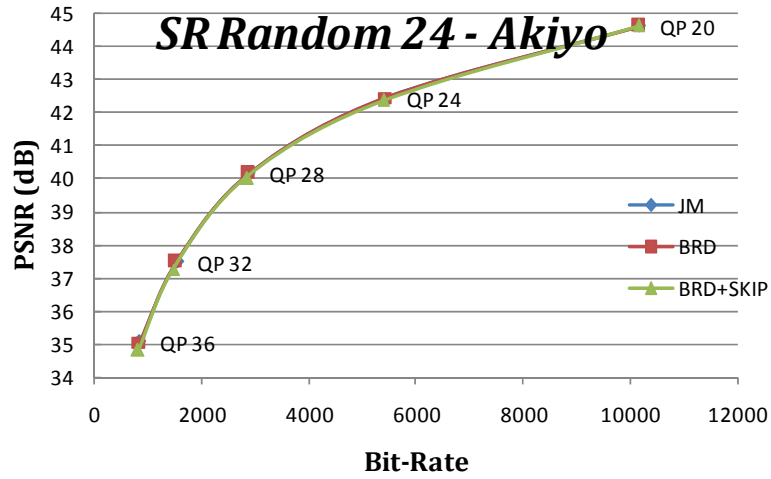


Fig. 4-24 RD curve comparison under SR random 24 for "Akiyo" sequence

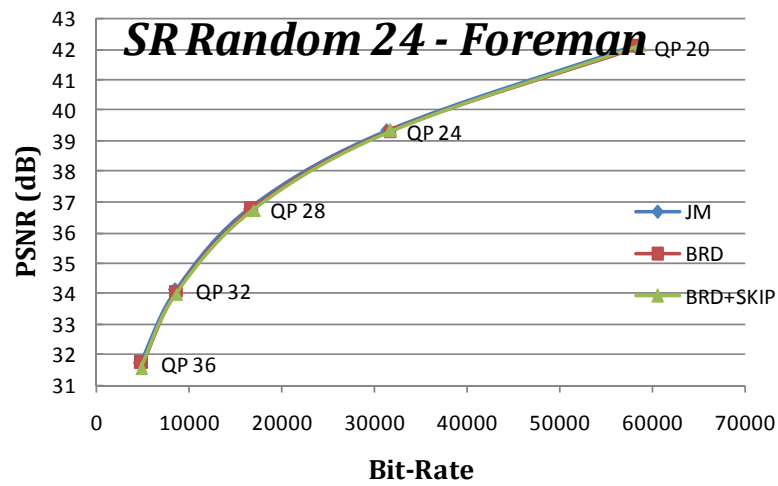


Fig. 4-25 RD curve comparison under SR random 24 for "Foreman" sequence

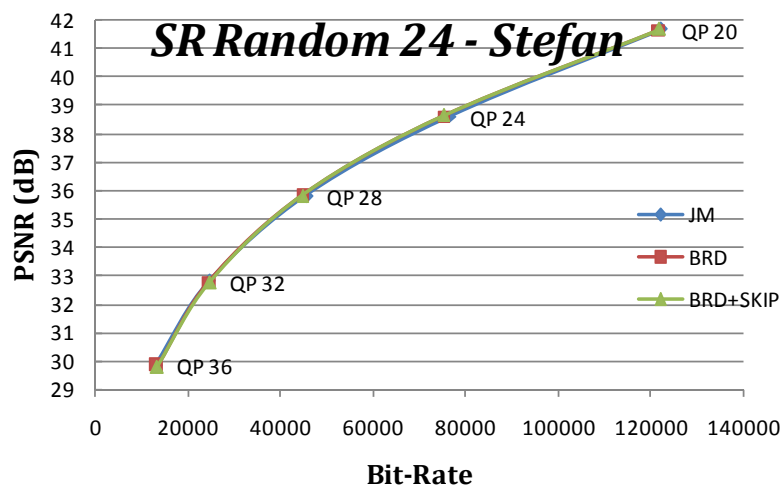


Fig. 4-26 RD curve comparison under SR random 24 for "Stefan" sequence

4.2.2. The distribution of MB for skip mode analysis

Fig. 4-27 - Fig. 4-29 are the average distribution of not skipped and skipped MB for low, medium, and high motion CIF size sequences. The portion of skipped MB consists of three types: miss skip predict, error skip predict, and correct skip predict. “miss skip” means that the MB should be pre-skipped but it is not detected in the pre-skip stage by our skip detection design. “error skip” means that the MB is pre-skipped but it should not be skipped. “correct skip” means that we can accurately pre-skip the MB. The results represent that the high QP case will skip more MBs, up to 89.15% for low motion, 46.68% for medium motion, and 31.96% for high motion sequence. On the other hand, we find that the “error skip predict” does not degrade the performance a lot because these error-skipped MBs are nearly to be skipped.



Akiyo

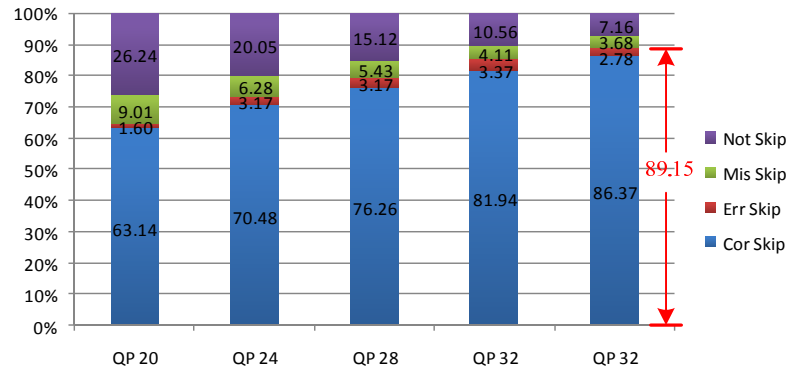


Fig. 4-27 The distribution of MB for “Akiyo” sequence

Foreman

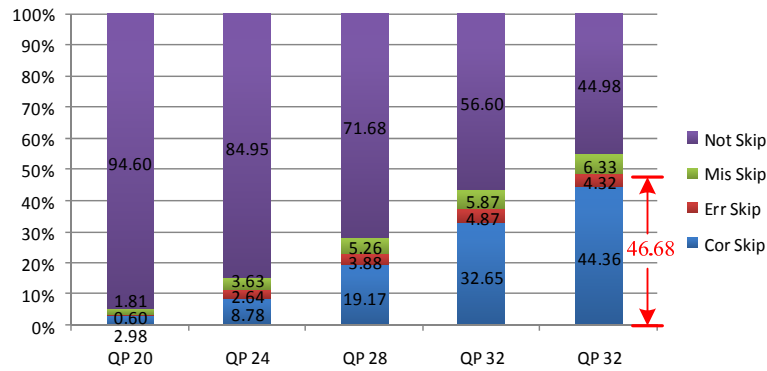


Fig. 4-28 The distribution of MB for “foreman” sequence

Stefan

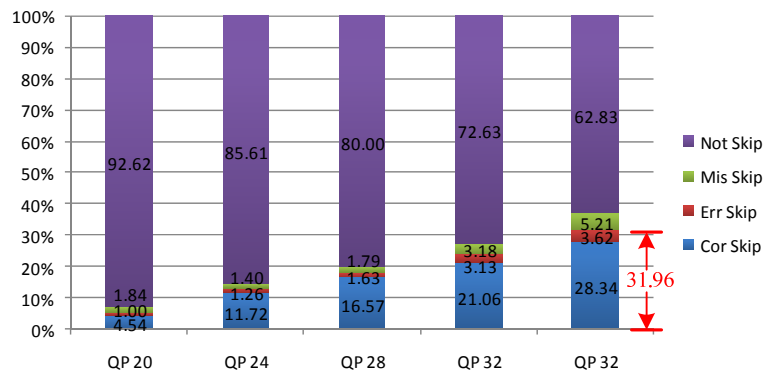


Fig. 4-29 The distribution of MB for “Stefan” sequence

4.2.3. Timing comparison with skip detection

TABLE 4-26 (a)-(f) are the coding time of CIF sequences in 6 BW patterns compared with JM12.2. For low motion sequence, our design only needs 11% to 35% coding time from QP 20 to QP 36. Nevertheless, for high motion sequence, the performance is limited. Fig. 4-30 (a)-(f) are the coding time of CIF sequences in 6 BW patterns. Each of them reveals that the coding time is roughly inverse proportion to QP. It is because that the MB coding with low QP can be hardly skipped.

SR Const 8						SR Random 8					
Time (%)	QP20	QP24	QP28	QP32	QP36	Time (%)	QP20	QP24	QP28	QP32	QP36
Akiyo	35.52	26.77	21.04	15.14	11.33	Akiyo	35.46	26.77	20.97	15.21	11.33
Foreman	96.44	88.92	77.11	63.10	51.85	Foreman	96.69	88.73	76.92	62.84	51.91
Stefan	94.59	87.52	81.95	76.34	68.67	Stefan	94.72	87.14	82.46	75.89	68.92

(a) (d)

SR Const 16						SR Random 16					
Time (%)	QP20	QP24	QP28	QP32	QP36	Time (%)	QP20	QP24	QP28	QP32	QP36
Akiyo	35.07	26.52	20.80	15.06	11.35	Akiyo	35.39	26.52	20.86	15.06	11.22
Foreman	96.21	88.38	77.00	62.48	51.70	Foreman	95.96	88.19	76.56	62.42	51.89
Stefan	94.44	86.98	81.76	75.61	68.12	Stefan	94.56	86.98	81.89	75.87	68.06

(b) (e)

SR Const 24						SR Random 24					
Time (%)	QP20	QP24	QP28	QP32	QP36	Time (%)	QP20	QP24	QP28	QP32	QP36
Akiyo	35.03	26.38	20.46	14.87	11.15	Akiyo	35.10	26.32	20.53	14.87	11.03
Foreman	95.86	87.91	76.22	61.59	51.03	Foreman	95.61	87.98	76.41	61.53	51.15
Stefan	94.60	86.98	81.52	75.61	67.67	Stefan	94.47	86.79	81.39	75.74	67.67

(c) (f)

TABLE 4-26 Coding time with skip detection of CIF sequences under: (a) SR constant 8 (b) SR constant 16 (c) SR constant 24 (d) SR random 8 (e) SR random 16 (f) SR random 24 patterns

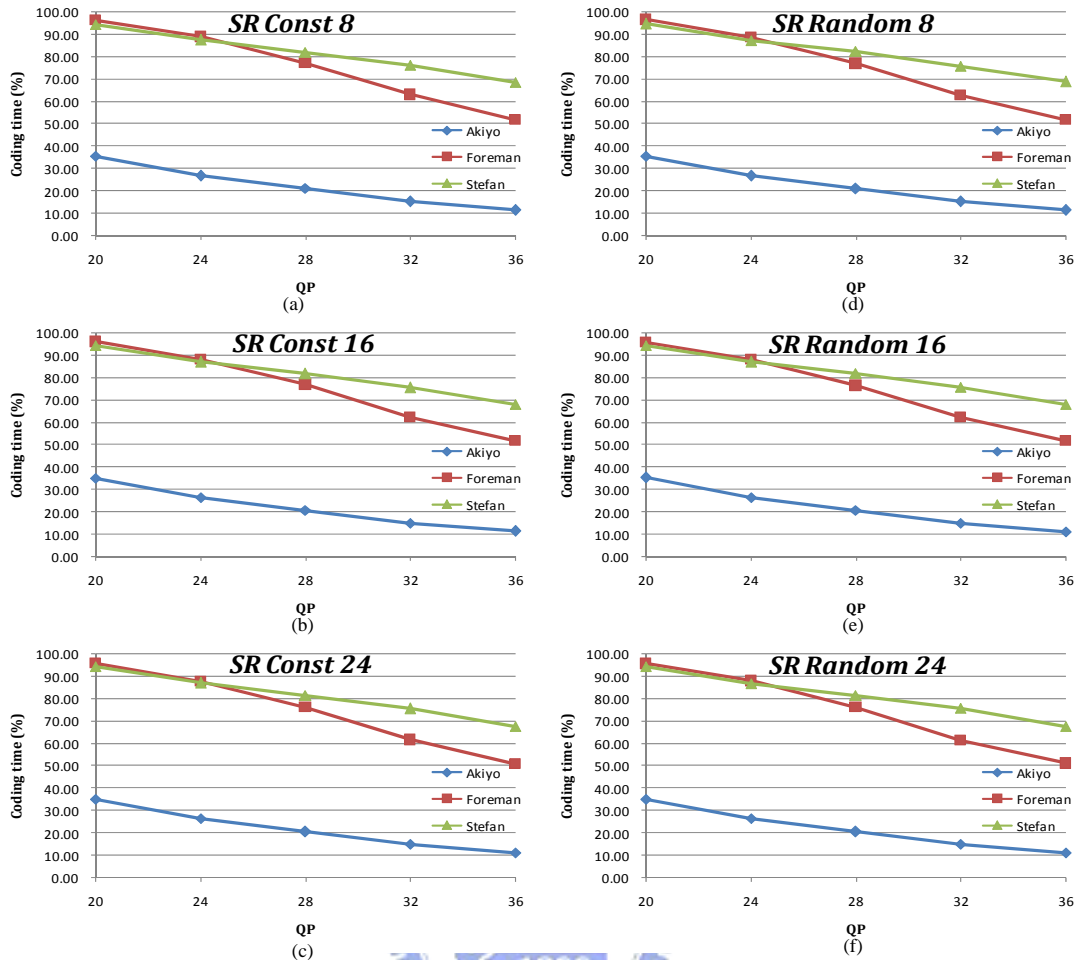


Fig. 4-30 Coding time curve with skip detection of CIF sequences under: (a) SR constant 8 (b) SR constant 16 (c) SR constant 24 (d) SR random 8 (e) SR random 16 (f) SR random 24 patterns

4.2.4. Completion time comparison of BW random patterns

The object of BW random patterns is used to represent a various bus systems, whereas it has various bandwidth supplies. In this section, we use BW random patterns to analyze the completion time between JM12.2 and our proposed design.

For example as shown in Fig. 4-31, the SR random 8 of BW patterns represents low BW status. For JM12.2, the data was only accessed with the constant SR, and it means the coding time would be limited by low BW status. If the BW is insufficient, the process must be waited and delayed to obtain the data. The area with slashed line in Fig. 4-31 shows the insufficient BW, and it causes the time overhead under original JM algorithm.

In contrast with our design, the data access with adaptable SR to approximate the curve of SR random 8 of BW patterns, and it means the coding would not be limited by low BW status. The other two examples are as shown in Fig. 4-32 and Fig. 4-33, and Fig. 4-34 shows the completion time comparison with SR random 8, 16, and 24 of BW patterns. Our design can complete the coding process on time, while original JM12.2 has to complete the coding process with 10% to 13% timing overhead.

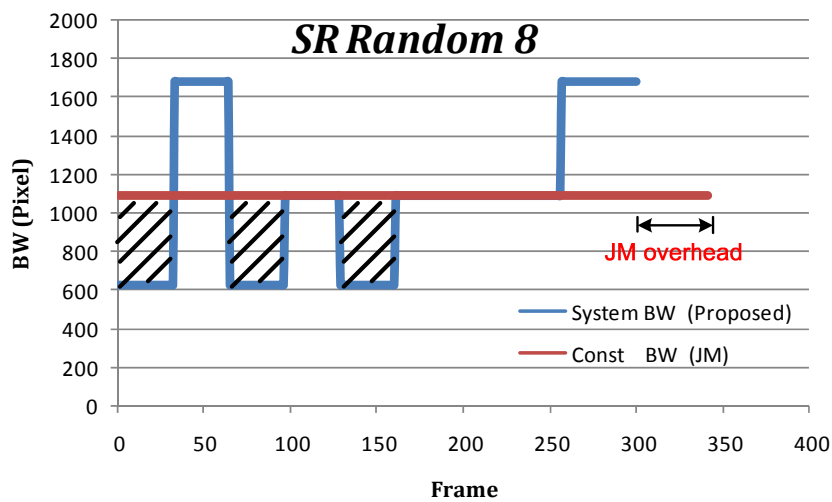


Fig. 4-31 Completion time comparison under SR random 8 pattern

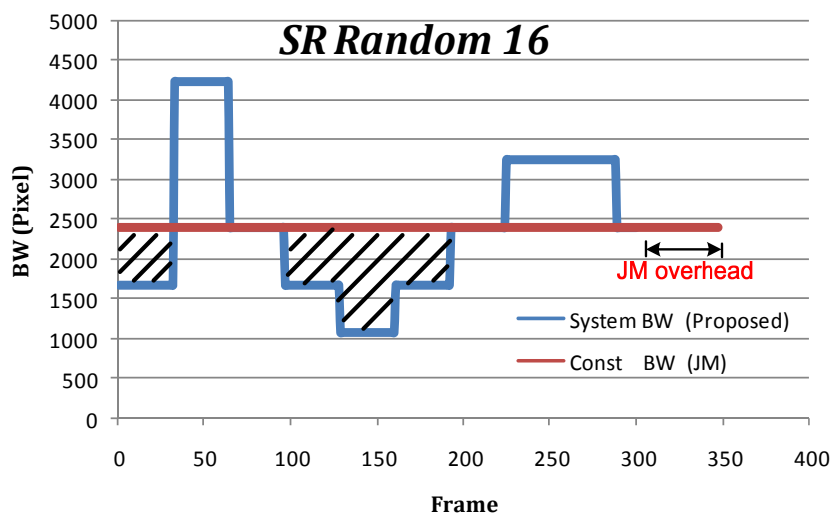


Fig. 4-32 Completion time comparison under SR random 16 pattern

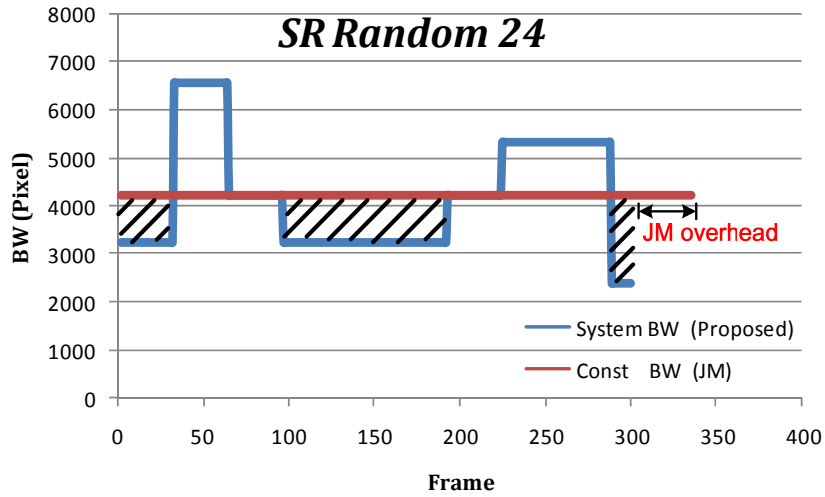


Fig. 4-33 Completion time comparison under SR random 24 pattern

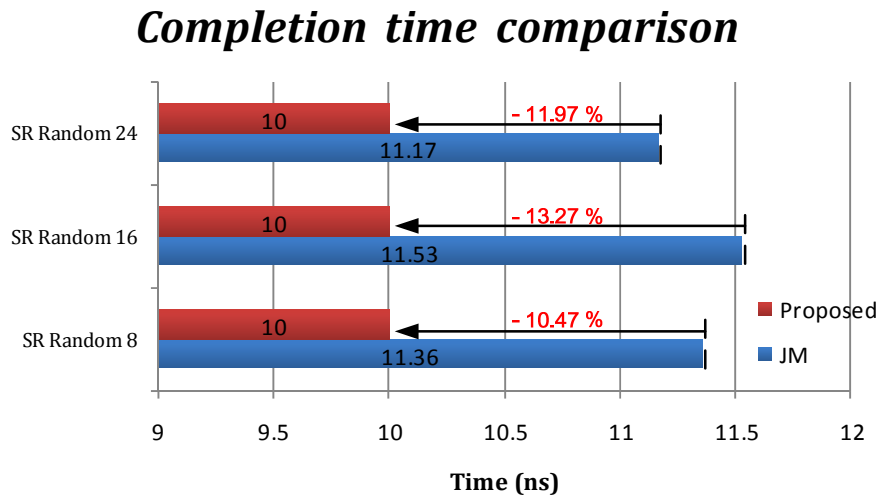


Fig. 4-34 Illustration of completion time comparison under different SR random pattern

4.3. Summary

In this chapter, we simulate the algorithms that proposed in chapter 3. We have analyzed the relationship between BW, PSNR, bit-rate, coding time, and skip rate. The experimental result shows that our design could achieve the same and sometimes even better performance under various BW bus systems. And thus it could be applied to the BW constrained devices.

5. Hardware implementation

5.1. Hardware design

Fig. 5-1 shows the proposed hardware architecture. The “ RDC_{init} ” stand for rate-distortion cost that obtained by using the predicted motion vector, in contrast, the “ RDC_{BMA} ” stand for rate-distortion cost that obtained by block-matching algorithm (i.e. full-search algorithm). The “Adj. MVs” stand for the adjacent motion vector from neighboring blocks and current block. Note that we use the “Skip_flag” to choose the “ SAD_{pre} ” or “ SAD_{cur} ” into Skip detection circuit. While the current MB have been skipped, we have no SAD value (i.e. SAD_{cur}) for skip detection in the next MB. In addition, the skipped MB has spatial correlation between the previous MB and the current MB. Therefore we use “ SAD_{pre} ” to substitute for “ SAD_{pre} ” for more accurate skip detection.

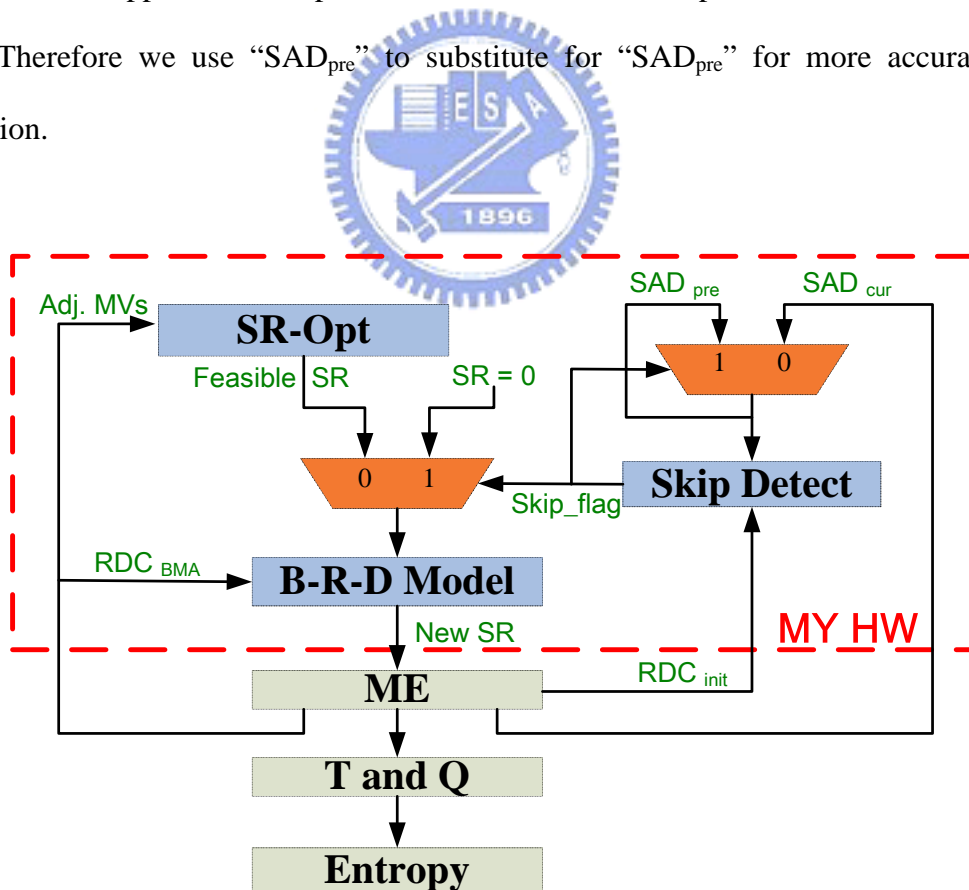


Fig. 5-1 BRD optimized motion estimation algorithm hardware architecture

5.2. Implementation result

The proposed design has been designed by using Verilog HDL and synthesized by TSMC 0.13 CMOS technology. Fig. 5-2 shows the test system of our design. Our design can provide various search ranges from $[-4, 3]$ to $[-32, 31]$ and just needs 1.3K gate count. In addition, our design could be achieved under bandwidth constrained system with only 100 MHz operating frequency. Thus it is suitable for portable devices with bandwidth constrained.

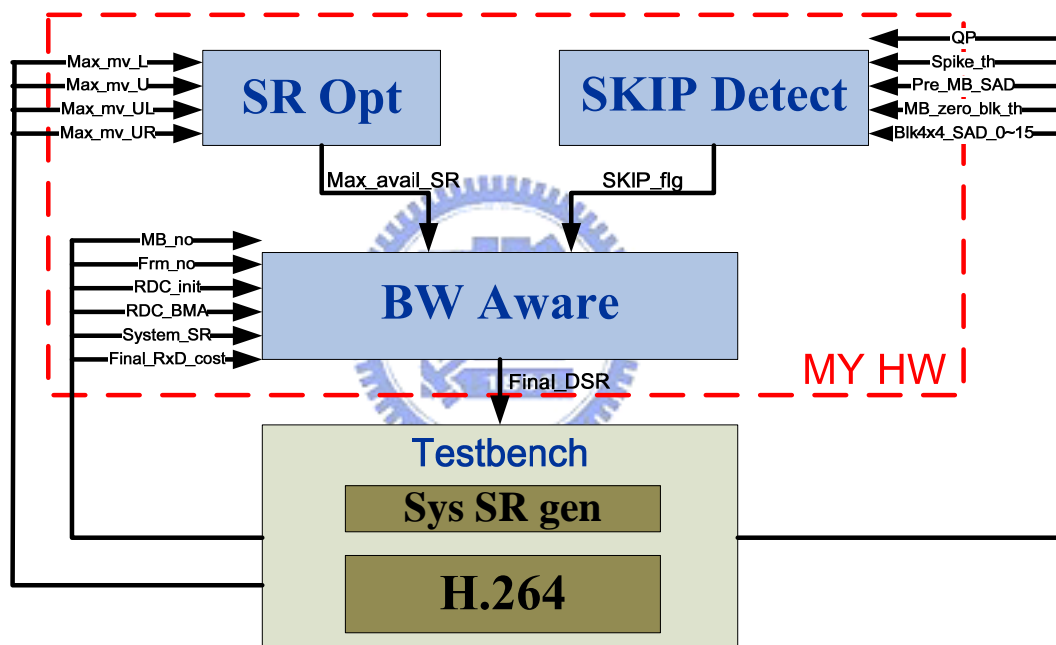


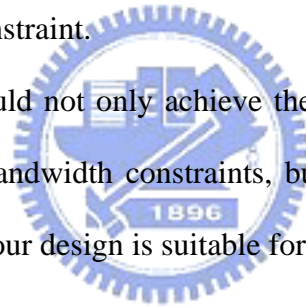
Fig. 5-2 Hardware verification for BRD optimized motion estimation algorithm

6. Conclusion and future work

6.1. Conclusion

In this thesis, we proposed a bandwidth-rate-distortion (B-R-D) optimized motion estimation algorithm to maximize rate distortion efficiency while can dynamically meet the available bandwidth. Compared with JM12.2, our B-R-D design could improve the bandwidth saving up to 70% under average search range size 16 due to appropriate MB-level bandwidth allocation by our B-R-D model calculation and bandwidth budget estimation; while the bandwidth saving up to 84% with further skip design added. In addition, while coding in high motion sequence, the simulation result shows our design could save average bit rate up to 13% and increase average PSNR up to 0.1dB at the same time under low bandwidth constraint.

In summary, our design could not only achieve the same and sometimes even better performance under various bandwidth constraints, but also fully utilize bandwidth for better quality maintain. Thus our design is suitable for video application nowadays.



6.2. Future work

We have proposed rate distortion optimized motion estimation under the available memory bandwidth constraint, while there are several issues should be analyzed to further improve the performance of the B-R-D design. We find that the bandwidth can be determined in another ways. For example, we can adopt other fast algorithms, such as three-step search [25], four-step search [26], or diamond search [27] to reduce the checking points. Another example is like “pixel truncation scheme” [28], which truncates lower bits of a coefficient during ME search, and “sub-sampling scheme” [29], which sub-sample the image to a small one for ME search. All these skills mentioned above make a trade-off between the performances and the bandwidth requirement to support more flexible design in an efficient way. This is another challenge need to study in the future.



7. Reference

- [1] T. Wiegand, G. J. Sullivan, G. Bjontegaad, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, pp. 560-575, July 2003
- [2] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ ISO/ IEC14496-10 AVC), Mar. 2003)
- [3] Joint Video Team Reference Software JM12.2, ITU-T
- [4] Tian Song and et al, "Adaptive Search Range Motion Estimation Algorithm for H.264/AVC," *IEEE International Symposium on Circuits and Systems*, pp. 3956-3959, May 2007
- [5] Toru YAMADA and et al, "Fast and Accurate Motion Estimation Algorithm by Adaptive Search Range and Shape Selection," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 897-900, March 2005
- [6] Y. C. Shih, "On sub-mW R-D Optimized Motion Estimation for Portable H.264/AVC," Master Thesis, Department of electronic Engineering, National Chiao-Tung University, Hsinchu, Taiwan, July 2007
- [7] Y. J. Wang, "Fast Algorithms and Architecture Designs for H.264/MPEG-4 AVC Motion Estimation," Master Thesis, Department of Electronic Engineering, National Chiao-Tung University, Hsinchu, Taiwan, June 2006
- [8] C. J. Lian and et al, "Power-Aware Multimedia: Concepts and Design Perspectives," *IEEE Circuits and Systems Magazine*, vol. 7, issue 2, pp. 26-34, 2007
- [9] Y. H. Chen and et al, "Power-Scalable Algorithm and Reconfigurable Macro-Block Pipelining Architecture of H.264 Encoder for Mobile Application," *IEEE International Conference on Multimedia and Expo*, pp. 281-284, July 2006
- [10] T. C. Chen and et al, "2.8 to 67.2mW Low-Power and Power-Aware H.264 Encoder

- for Mobile Applications,” *IEEE Symposium on VLSI Circuits*, pp. 222-223, June 2007
- [11] Z. HE and et al, “Power-Rate-Distortion Analysis for Wireless Video Communication Under Energy Constraints,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, issue 5, pp. 645-658, May 2005
- [12] P. L. Tai and et al, “Computation-Aware Scheme for Software-Based Block Motion Estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, issue 9, pp. 901-913, Sept 2003
- [13] H. F. Ates and Y. Altunbasak, “Rate-Distortion and Complexity Optimized Motion Estimation for H.264 Video Coding,” *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 18, issue 2, pp. 159-171, Feb 2008
- [14] Y. V. Ivanov and C. J. Bleakley, “Dynamic Complexity Scaling for Real-Time H.264/AVC Video Encoding,” *Proceeding of the 15th International Conference on Multimedia*, pp. 962-970, 2007
- [15] C. Y. Chang and et al, “A New Computation-Aware Scheme for Motion Estimation in H.264,” *IEEE International Conference on Computer and Information Technology*, pp. 561-565, July 2008
- [16] S. H. Wang and et al, “A Complexity Aware Variable-Bit-Depth Motion Estimation,” *International Conference on Consumer Electronics*, pp. 233-234, Jan 2005
- [17] T. C. Chen and et al, “Analysis and Architecture Design of an HDTV720p 30 Frames/s H.264/AVC Encoder,” *IEEE transaction on Circuits and Systems for Video technology*, vol. 16, no. 6, pp. 673-688, June 2006
- [18] C. Sam. Kann. and et al, ”Low-Complexity Skip Prediction for H.264 Through Lagrangian Cost Estimation,” *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 16, issue 2, pp. 202-208, Feb 2006

- [19] Y. C. Lin and et al, "Fast Mode Decision for H.264 Based on Rate-Distortion Cost Estimation," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 1137-1140, April 2007
- [20] Y. V. Ivanov and et al, "Skip Prediction and Early Termination for Fast Mode Decision in H.264/AVC," *International Conference on Digital Telecommunications*, pp. 7-7, 2006
- [21] Hanli Wang and et al, "Efficient Prediction Algorithm of Integer DCT Coefficients for H.264/AVC Optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 547-552, 2006
- [22] Gyu Yeong Kim and et al, "An Early Detection of All-Zero DCT Blocks in H.264," *International Conference on Image Processing*, vol. 1, pp. 453-456, Oct 2004
- [23] C. C. Lin and et al, "Hardware Efficient Skip Mode Detection for H.264/AVC," *International Conference on Consumer Electronics*, pp. 1-2, Jan 2008
- [24] X. J. Zhu and et al, "Fast Mode Decision and Reduction of Reference Frames for H.264 Encoder," *International Conference on Control and Automation*, vol. 2, pp. 1040-1043, June 2005
- [25] Reoxiang Li and et al, "A New Three-Step Search Algorithm for Block motion estimation," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 4, pp. 438-442, Aug 1994
- [26] L. M. Po and et al, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, issue 3, pp. 313-317, June 1996
- [27] S. Zhu and et al, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," *International Conference on Information, Communications and Signal Processing*, vol. 1, pp. 292-296, Sept 1997

- [28] Z. L. He and et al, "Low-Power VLSI Design for Motion Estimation Using Adaptive Pixel Truncation," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 10, issue 5, pp. 669-678, Aug 2000
- [29] H. Y. Chin and et al, "A Bandwidth Efficient Subsampling-Based Block Matching Architecture for Motion Estimation," *Asia and South Pacific Design Automation Conference*, pp. 18-21, 2005



