

# 國立交通大學

## 資訊科學與工程研究所

### 碩士論文

結合樣板及區塊動作補償之雙動作向量預  
測方法

Bi-prediction Combining Template and Block Motion  
Compensations

研究生：李宗霖

指導教授：彭文孝 教授

譚建民 教授

中華民國 一 百 年 十 月

結合樣板及區塊動作補償之雙動作向量預測方法  
Bi-prediction Combining Template and Block Motion Compensations

研究生：李宗霖

Student : Chung-Lin Lee

指導教授：彭文孝

Advisor : Wen-Hsiao Peng

譚建民

Advisor : Jimmy J.M. Tan

國立交通大學  
資訊科學與工程研究所  
碩士論文



Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

October 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年十月

## 結合樣板及區塊動作補償之雙動作向量預測方法

研 究 生：李宗霖

指 導 教 授：彭文孝

國立交通大學資訊科學與工程研究所 碩士班

### 摘 要

本篇論文介紹了一種基於結合樣板比對預測(TMP)和區塊動作補償(BMC)的雙動作向量預測方法。由於本技術採用交疊動作區塊動作補償(OBMC)方法所產生的像素可適性權重係數表來結合兩個分別來自樣板比對預測(TMP)和區塊動作補償(BMC)之預測區塊中的每個像素，因此如何設計這些權重係數值以達到最佳的預測效果，便成了本技術的關鍵部份。為了決定這些權重係數，本論文採用了一種參數化的理論模型來推導產生最佳的權重係數，同時也據此修正了原本傳統動作向量的預測方式，在動作向量預測階段即導入這些權重係數以達到更精確的預測效果。為了要驗證本論文所提出來的雙動作向量預測方法的可靠性及壓縮效能，本論文設計了一系列的實驗，企圖在壓縮效能及運算複雜度上面取得平衡。實驗結果顯示在最佳權衡之下，壓縮效能可以達到 1.1%到 4.1%的範圍內平均 2.2%的 BD-Rate 節省，複雜度方面在合理程度上編碼端上升了 46%的壓縮時間，解碼端則是上升了 33%。

# Bi-prediction Combining Template and Block Motion Compensations

Student : Chung-Lin Lee

Advisor : Wen-Hsiao Peng

Institute of Computer Science and Engineering

National Chiao Tung University

## ABSTRACT

This thesis introduces a bi-prediction scheme based on a joint application of template and block motion compensations. Since the template motion is decode-side inferable, this scheme needs only a motion overhead as that of uni-directional prediction. Two predictors derived from the template and block matchings are weighted in a pixel-adaptive manner using OBMC. From an analytical aspect, we provide an optimal design of window function in a parametric overlapped block motion compensation (OBMC) framework to further improve the efficiency of inter-frame prediction. In view of the tradeoff between the performance and the complexity, we discussed the impacts on the prediction efficiency of the proposed scheme when the number of template shapes and the number of motion hypotheses are changed. Also, a fast template search is provided, which greatly reduces the complexity at the decoder side. As compared with HM3.0, the proposed scheme achieves an average BD-rate saving of 2.2%, with a minimum of 1.1% and a maximum of 4.1%. The encoding time increases 46% while the decoding time increases moderately in 33%.

## 誌 謝

回顧兩年的研究所生涯，首先，我要感謝我的指導教授—彭文孝 博士，給予我於學問研究上的指導。彭老師實事求是的精神，與深入剖析問題的態度，其追根究柢與契而不捨的指導方式，已經成為我在學習與研究路上的典範與楷模。其次，我要感謝我的學長—陳漪紋 博士與陳俊吉 博士，還有我的摯友吳崇豪 博士，不辭辛勞的與我討論，給予許多珍貴的意見，並且能適時從旁給予建議修正我不慎偏差的研究方向，使我在這兩年的碩士生涯，不再舉步維艱。謹此對四人致上由衷的謝意。


有榮幸進入多媒體架構與處理實驗室，可以在這個優良的環境下不斷學習，又有熱心與親切的實驗室成員們的切磋與討論，是我在學士後時代最充實的時光。感謝我的學長姐們—陳漪紋 博士、陳俊吉 博士、詹家欣 博士、王澤瑋、吳思賢、蔡閏旭、與楊復堯，引領我進入研究生的階段；感謝我的好同學們吳崇豪 博士、曾于真、黃嘉彥與陳孟傑，不論是課業上或研究上，他們總是可以一針見血地提出問題的核心要點，給予最直接協助；感謝我的學弟陳彥宇、吳牧軒、吳昱興、朱弘正、與王信硯，在最後這一年內，給予許多無私的協助。

最後，我要感謝我的父母—李廷欽 先生與蔡素珠 女士的栽培，在爭取碩士學位的路上，給予百分之百的支持，讓我免去許多後顧之憂與煩擾。感謝我的胞弟—李文郁，給予我滿滿手足的關懷。感謝我的老師、家人、與朋友們，是你們的支持，使我有信心取得這個學位，謝謝你們。

---

# Contents

---

<b>Contents</b>		<b>i</b>
<b>List of Tables</b>		<b>iii</b>
<b>List of Figures</b>		<b>1</b>
<b>1 Introduction</b>		<b>2</b>
1.1 Research Overview . . . . .		2
1.2 Problem Statement . . . . .		4
1.3 Contribution . . . . .		4
1.4 Organization . . . . .		5
<b>2 Background</b>		<b>6</b>
2.1 Overview of High Efficiency Video Coding . . . . .		6
2.1.1 Coding, Prediction and Transform Units . . . . .		7
2.1.2 Inter Prediction . . . . .		8
2.2 Template Matching Prediction . . . . .		9
2.3 Motion Sampling and Reconstruction . . . . .		10
2.4 Overlapped Block Motion Compensation . . . . .		11
2.5 Parametric OBMC . . . . .		12

---

<b>3</b>	<b>Combining Template and Block Motion Compensations</b>	<b>15</b>
3.1	Concept of Operation . . . . .	15
3.2	Optimized Block Matching Criterion . . . . .	15
3.3	Window Functions . . . . .	18
<b>4</b>	<b>Experiments</b>	<b>20</b>
4.1	Experimental Conditions . . . . .	20
4.1.1	Common Test Conditions . . . . .	20
4.1.2	TB-mode . . . . .	21
4.2	Heuristic Window Functions . . . . .	22
4.3	Compression Performance of TB-mode . . . . .	24
4.3.1	Coding Efficiency versus Number of Templates . . . . .	24
4.3.2	Theoretical versus Heuristic Window Functions . . . . .	25
4.4	Multiple Hypotheses . . . . .	25
4.5	Fast Algorithm . . . . .	27
4.5.1	Enhancements for Encoder Only . . . . .	27
4.5.2	Enhancements for Encoder and Decoder . . . . .	27
4.5.3	Summary . . . . .	28
<b>5</b>	<b>Conclusion</b>	<b>29</b>
	<b>Bibliography</b>	<b>31</b>

---

# List of Tables

---

2.1	Comparisons of tool features between H.264/AVC and HEVC. . . . .	7
4.1	Common test conditions. . . . .	21
4.2	Experimental settings of TB-mode. . . . .	22
4.3	The start position <b>a</b> and <b>b</b> for various $2N \times 2N$ PU sizes. . . . .	22
4.4	BD-rate savings and processing time ratios of TB-mode with 3- and 5-shape-adaptive configurations. . . . .	24
4.5	BD-rate savings and processing time ratios of TB-mode with theoretical and heuristic window functions. . . . .	25
4.6	BD-rate savings and processing time ratios of enabling multiple-hypotheses.	26
4.7	BD-rate savings and processing time ratios of TB-mode after applying fast algorithms. . . . .	28



---

# List of Figures

---

1.1	Joint application of TMP and BMC. . . . .	3
2.1	Basic concept of MRG. . . . .	8
2.2	Basic concept of TMP. . . . .	9
2.3	Motion sampling and prediction error of BMC. . . . .	10
2.4	Motion sampling and prediction error of TMP. . . . .	11
3.1	Joint application of TMP and BMC. . . . .	16
3.2	(a) geometry relationship between TMP centroid, and $\mathbf{s}_b$ . (b) SMSE surface as a function of the location of pixel $\mathbf{b}$ . . . . .	18
3.3	Window functions for typical template designs. . . . .	19
4.1	Designs of heuristic window functions. . . . .	23

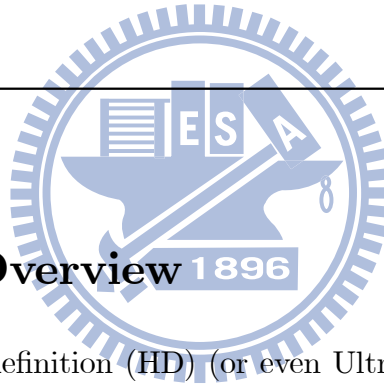
# CHAPTER 1

---

## Introduction

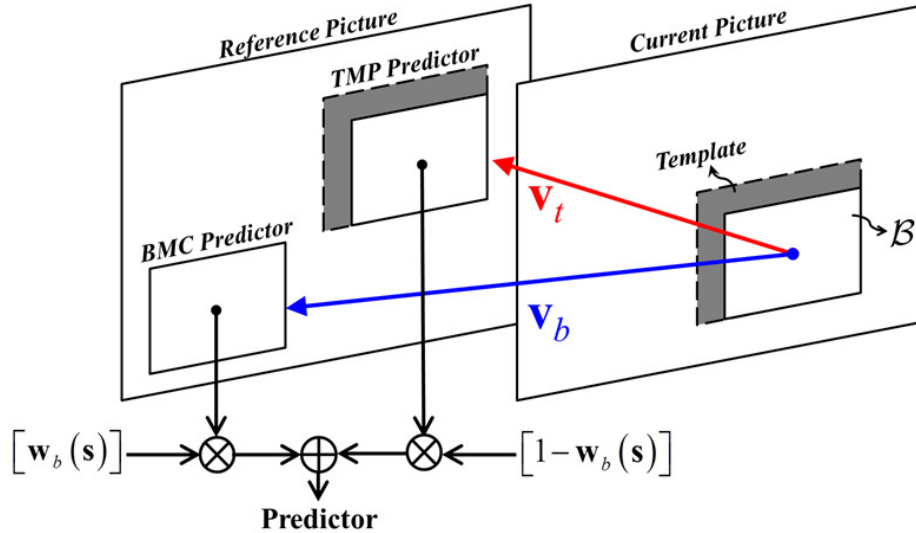
---

### 1.1 Research Overview



With the growth of high definition (HD) (or even Ultra HD) acquisition and display technologies, anticipation of a need for higher coding efficiency has led to the development of a new video coding standard, named high efficiency video coding (HEVC). This standard, which is currently under the joint development of the Moving Pictures Expert Group of the International Organization for Standardization and International Electrotechnical Commission (ISO/IEC MPEG) and the Video Coding Experts Group of the International Telecommunication Union Telecommunication Standardization Sector (ITU-T VCEG), aims to provide higher compression efficiency ( $\sim 50\%$  bit-rate reduction) than the state-of-the-art H.264/AVC standard while being capable of operating in low-complexity or high-efficiency modes. Its applications range from mobile HD video to Ultra HD video, covering a frame resolution from WVGA ( $800 \times 480$ ) to  $4K \times 2K$  and beyond, such as  $8K \times 4K$ .

Using the data accessible to the decoder for motion inference has recently emerged as a promising technique for the next generation video coding standard. Template



**Figure 1.1:** Joint application of TMP and BMC.

matching prediction (TMP) [1][2][3] is a typical example that estimates the motion vector (MV) for a target block on the decoder side by minimizing the matching error over the reconstructed pixels in its immediate inverse-L-shaped neighborhood (usually termed the *template*). Overlapped block motion compensation (OBMC) [4][5], first proposed more than one and a half decades ago, also follows the same rationale. It utilizes the received motion data as a source of information about the motion field and forms a better prediction of a pixel's intensity based on its own and nearby block MVs.

Motivated by the preceding investigations, we were led to develop a bi-prediction scheme, which requires the same motion cost as that of uni-directional prediction. The idea is to combine predictors resulting from the template and block matchings using OBMC. Of particular interest in this combination is that the MV derived from TMP is inferred at the decoder side. Conceptually, the bi-prediction scheme signals only one block motion while retaining almost the same performance as that of the conventional bi-prediction. Moreover, a modified block matching criterion is also proposed to optimize the motion parameters to be signaled based on the contribution from the MVs inferred by TMP.

## 1.2 Problem Statement

Fig. 1.1 depicts the basic concept of our bi-prediction scheme. Based on the concept of the conventional bi-prediction, our scheme also predicts a target prediction unit (PU<sup>1</sup>) by the two MVs,  $\mathbf{v}_b$  and  $\mathbf{v}_t$ , where  $\mathbf{v}_b$  is explicitly signaled and  $\mathbf{v}_t$  is derived by TMP. Since  $\mathbf{v}_t$  cannot be specified discretionarily, it is important to find  $\mathbf{v}_b$  that, when applied jointly with  $\mathbf{v}_t$ , minimizes the mean square prediction error:

$$\mathbf{v}_b^* = \arg \min_{\{\mathbf{v}_b\}, \{w_b(\tilde{\mathbf{s}})\}} \sum_{i=1}^N \sum_{\mathbf{s} \in \mathcal{B}_i} (I_k(\mathbf{s}) - w_t(\tilde{\mathbf{s}}) I_{k-1}(\mathbf{s} + \mathbf{v}_{t,i}) - w_b(\tilde{\mathbf{s}}) I_{k-1}(\mathbf{s} + \mathbf{v}_{b,i}))^2, \quad (1.1)$$

where  $\mathbf{v}_{b,i}$  and  $\mathbf{v}_{t,i}$  is the template and block motions of a specific target block  $\mathcal{B}_i$ ;  $I_k$  and  $I_{k-1}$  are the current frame and a previously coded reference frame, respectively. The symbol  $\tilde{\mathbf{s}}$  is the pixel index relative to the absolute position in a target PU.  $w_b(\tilde{\mathbf{s}})$  and  $w_t(\tilde{\mathbf{s}})$  denotes the window functions associated with every  $\mathbf{v}_{b,i}$  and  $\mathbf{v}_{t,i}$ , where  $w_b(\tilde{\mathbf{s}}) = 1 - w_t(\tilde{\mathbf{s}})$ . Since  $w_b(\tilde{\mathbf{s}})$  has a decisive effect on the prediction performance, this problem is thus turned into jointly optimizing  $w_b(\tilde{\mathbf{s}})$  and  $\mathbf{v}_b$ .

## 1.3 Contribution

Specifically, our main contributions are included as follows:

- A bi-prediction scheme that requires a motion cost as for uni-directional prediction.
- An analytical interpretation of block-based motion-compensated prediction.
- A model-based approach that determines an optimal OBMC window function.
- A modified motion search criterion that makes the best of both MV's abilities in reducing prediction errors.

Experimental results indicate that our best scheme makes an average BD-rate saving [6] of 2.2%, with a minimum of 1.1% and a maximum of 4.1%. Its encoding time increases by 46% while the decoding time increases by 33%. Regardless of complexity, the highest coding gain is achieved with an average of 2.9%, while the minimum of

---

<sup>1</sup>The coding unit (CU), the basic compression unit as the macroblock (MB) in AVC, has various sizes but is restricted to a square shape. The PUs are the various partitions having a square or rectangular shape with several sizes.

1.3% and the maximum is 5.2%.

## 1.4 Organization

The rest of this thesis is organized as follows: Chapter 2 briefly introduces the current status of HEVC, then reviews the motion sampling and reconstruction issues, and parametric OBMC framework. Chapter 3 details the proposed technique. Chapter 4 provides experimental results. Finally, this thesis is concluded with a summary of our work.



# CHAPTER 2

---

## Background

---

### 2.1 Overview of High Efficiency Video Coding

High Efficiency Video Coding (HEVC) is a draft video compression standard, a successor to H.264/MPEG-4 AVC (Advanced Video Coding), currently under joint development by the ISO/IEC MPEG and ITU-T VCEG. MPEG and VCEG have established a Joint Collaborative Team on Video Coding (JCT-VC) to develop the HEVC standard. In February 2010, the JCT-VC issued a call-for proposal (CfP). 27 proposals submitted to the ITU/ISO joint committee competing for the next generation video standard. The proposal evaluation results in the April 2010 JCT-VC meeting indicated that a better coding scheme is possible and thus the HEVC work item was launched. So far, the development of the HEVC is still in progress, and the new-generation video standard is expected to be defined in 2012.

After the CfP competition in the 1<sup>st</sup> JCT-VC meeting, Test Model Under Consideration (TMuC) is constructed mainly from the best performer's code-base and the other top-performing HEVC proposals. It serves as a good starting point at the very beginning of the collaborative phase, and aims at creating a minimum set of well-tested

**Table 2.1:** Comparisons of tool features between H.264/AVC and HEVC.

Feature	H.264/AVC	HEVC
<b>Coding, Prediction and Transform Unit</b>		
Coding Unit	16 × 16 Macroblock	Variable; Large size (8 × 8 to 64 × 64)
Prediction Unit	Quadtree-based structure	Irregular partitioning; Large size
Transform Unit	4 × 4 and 8 × 8	Rectangular; Large size
<b>Inter Prediction</b>		
MVp Derivation	Median	Advanced motion vector prediction (AMVP)
Motion Inference	DIRECT, SKIP	Motion merging mode (MRG)
Interpolation Filter	6-tap FIR; Bilinear filter	8-tap DCT-based interpolation filter (DCT-IF)
<b>Intra Prediction</b>		
Directional Prediction	At most 8 directions	Angular intra prediction (34 directions)
Chroma Prediction	Independent prediction	Refer to reconstructed luma samples
<b>Transform, Quantization, In-loop Filter, Entropy Coding</b>		
Transform	Integer DCT	Residual quad-tree transform (RQT)
Quantization Matrix	Fixed	Context-adaptive selection
Adaptive Loop Filter	No	Yes
Entropy Coding	VLC; CAVLC; CABAC	Modified CAVLC; CABAC
<b>Internal Bit Depth Increase</b>		
Bit Depth	8 bits	8/10 bits

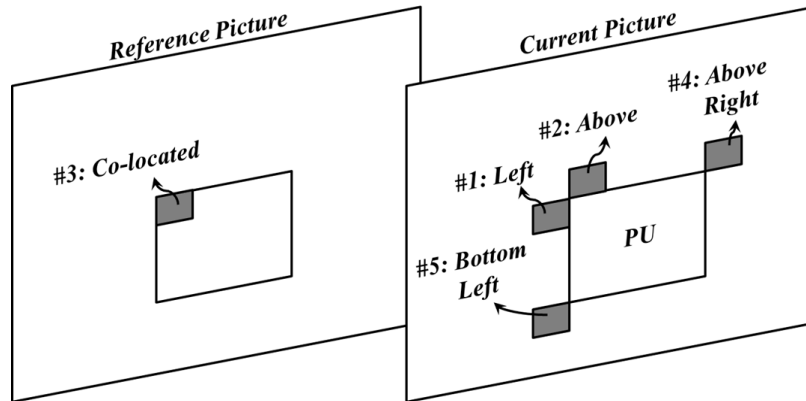
tools to establish the HEVC Test Model (HM)<sup>1</sup>. In this thesis, our proposed scheme is implemented based on the HM version 3.0 (HM-3.0) software.

To see the current HEVC base structure and coding tools in comparison with H.264/AVC, we summarize the tool features in Tab. 2.1. Important features that are relative to our research in this thesis are further described in the followings subsections:

### 2.1.1 Coding, Prediction and Transform Units

The basic unit for HEVC compression, referred as coding units (CU), is usually a  $N \times N$  square region of a frame from  $8 \times 8$  to  $64 \times 64$  luma and  $4 \times 4$  to  $32 \times 32$  chroma samples. It may contain several prediction units (PU) and transform units (TU) for inter/intra prediction and transform/residual coding. In comparison with H.264/AVC, a CU is a Macroblock (MB), which covers  $16 \times 16$  luma and  $8 \times 8$  chroma samples. Its PUs are the variable MB partitions having a square or rectangular shape with several sizes. In HEVC, the size and shape of CU, PU and TU become more flexible.

<sup>1</sup>HEVC Test Model (HM) is a common test platform such as the JM software for AVC.



**Figure 2.1:** Basic concept of MRG.

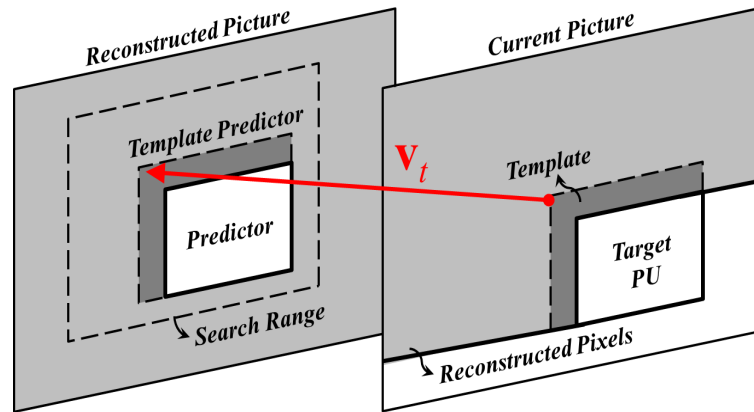
### 2.1.2 Inter Prediction

Inter prediction is crucial to the performance of video compression. Variants of inter prediction concept associated with MV could potentially contribute to many new designs in HEVC. In H.264/AVC, MV is predictive-coded by a motion vector predictor (MVP). The method of forming MVP depends on the median of the available nearby MVs. In HEVC, advanced motion vector prediction (AMVP) [7] introduces an adaptive motion vector prediction techniques, which sufficiently exploits spatial and temporal correlation of motion vector with neighboring PUs. It constructs MVP candidate set by inferring the available MV from left, top, and co-located PUs with the same reference list and reference frame. The encoder can select the best MVP from the candidate set and explicitly transmits the corresponding index indicating the selected MVP.

The SKIP and Direct modes in H.264/AVC are extended to the motion merging mode (MRG) [8] in HEVC. MRG derives its motion information from neighboring PUs. According to current design, the inferred motion parameters come from left, above, co-located, above-right, and bottom-left PUs as depicted in Fig. 2.1. Those neighboring PUs, which have available motion parameters, termed merge candidates, are grouped into merge candidate set. The index of the chosen candidate will be transmitted to the decoder as an indicator for decoding motion inference. Furthermore, MRG can be coded either w/ or w/o residual.

The existing sub-pel interpolation method in H.264/AVC has been improved by redesigning the filter coefficients. In HEVC, a 8-tap DCT-based interpolation filter (DCT-IF) provides fractional pel accuracy interpolation by replacing the combination



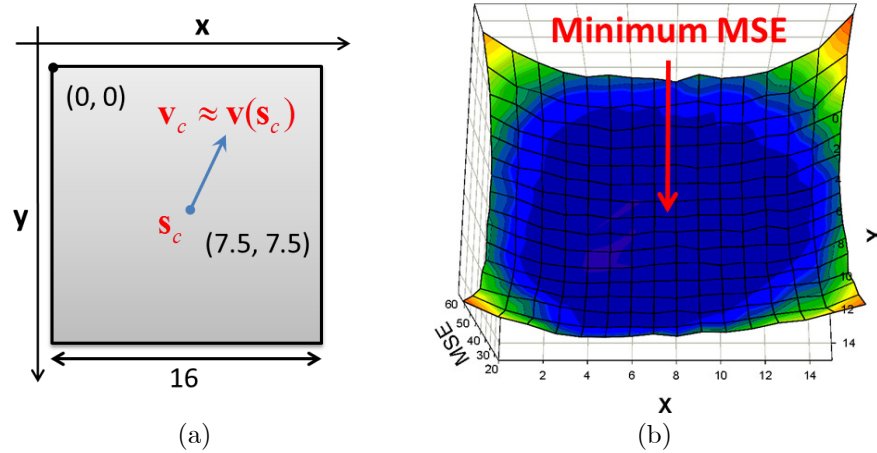


**Figure 2.2:** Basic concept of TMP.

of Wiener and bilinear filters with a set of interpolation filters at the desired fractional accuracy. More specifically, instead of a combination of 6-tap and bilinear filtering procedures in H.264/AVC, only one filtering procedure is needed to provide the interpolation pixel to any pixel accuracy. Thus, the motion compensation process can be simplified in the implementation point of view and the complexity can also be reduced for quarter-pel accuracy.

## 2.2 Template Matching Prediction

Template Matching Prediction (TMP) [1][2][3] is one of the realization methods of decoder-side motion vector derivation. As can be seen in Fig. 2.2, TMP exploits the correlation between reconstructed pixels of the causal neighborhood (usually an inverse-L-shaped region termed *template*) and those in the reconstructed frames. The motion search process of TMP is quite similar to that of inter prediction except the inputs to the process. For TMP, since the original pixels of the target PU are not available at the decoder side, the template is used as it were the original pixels for motion search. The MV derived from TMP is thus determined by minimizing the prediction error of pixel intensity between the current template and each template predictor lying in the search range. Then, the MV is directly used for predicting the intensities of the target PU.



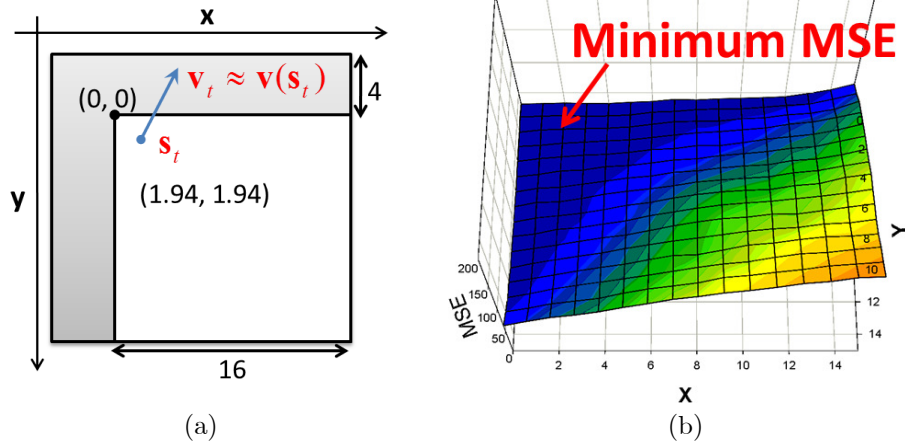
**Figure 2.3:** Motion sampling and prediction error of BMC.

## 2.3 Motion Sampling and Reconstruction

Motion-compensated prediction (MCP) has been a crucial technique in the state-of-the-art video compression design such as HEVC for reducing temporal redundancy. An insightful perspective in MCP is to formalize its notion into a two stage process comprising of sparse motion sampling followed by the reconstruction of temporal predictors. In motion sampling phase, motion estimation acts as a motion sampler taking samples at a true motion field, whereas in reconstruction phase, the prediction pixel values are then acquired by the corresponding area in previous coded frames based on the sampled motions.

Block motion compensation (BMC) is an essential component of MCP. In respect of motion sampling, BMC uses one single motion vector as a linear estimator of the true motion field for a block of pixels. As depicted in Fig. 2.3 (a), the video models of motion field and the field of block motion estimation proposed in [9] have shown that the best MV estimation of BMC is assumed to be the true motion of the block center  $\mathbf{s}_c$ . In Fig. 2.3 (b), empirical result also indicates that the best motion sampling position with minimum mean square error (MSE) is located on the block center. Furthermore, it has been observed that the MSE at block boundaries tend to be larger than those at block centers. This phenomenon is called motion uncertainty, which produces the blocking artifacts.

Likewise, the motion sampling of TMP also follows the same rationale. As depicted in Fig. 2.4 (a) and (b), statistically, the best MV estimation of TMP is obtained by



**Figure 2.4:** Motion sampling and prediction error of TMP.

approximating a MV to the true motion of the left-top template centroid  $\mathbf{s}_t$ . It is said that the behavior of MV derived from TMP can also be interpreted by the concept of motion sampling.

On the other hand, overlapped block motion compensation (OBMC) uses more sophisticated algorithms to reconstruct the motion field without additional samples. It directly gives a LMMSE estimate for every pixel's intensity based on motion compensated signals derived from motion vectors sampled at nearby block centers. The design of OBMC window functions are optimized for recognizing and exploiting the non-stationary structure of motion uncertainty. It is noted that OBMC produces smoother motion fields while giving similar or lower mean square prediction error.

## 2.4 Overlapped Block Motion Compensation

This section briefly reviews the basics of OBMC, to aid the understanding of POBMC. In words, OBMC is to find a LMMSE estimate of a pixel's intensity value  $I_k(\mathbf{s})$  based on motion-compensated signals  $\{I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_i))\}_{i=1}^L$  derived from its nearby block MVs  $\{\mathbf{v}(\mathbf{s}_i)\}_{i=1}^L$ . From an estimation-theoretic perspective, these MVs are plausible hypotheses for its true motion, and to maximize coding efficiency, their weights  $\mathbf{w} = [w_1, w_2, \dots, w_L]^T$  are chosen to minimize the mean squared prediction error subject

to the unit-gain constraint [5]:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \xi(\mathbf{w}) \text{ s.t. } \sum_{i=1}^L w_i = 1, \quad (2.1)$$

where

$$\xi(\mathbf{w}) = E \left\{ \left( I_k(\mathbf{s}) - \sum_{i=1}^L w_i I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_i)) \right)^2 \right\}. \quad (2.2)$$

Applying the Lagrangian method to (2.1) then gives

$$\mathbf{w}^* = \mathbf{R}^{-1} \left[ \mathbf{P} - \mathbf{U} \left( \frac{\mathbf{U}^T \mathbf{R}^{-1} \mathbf{P} - 1}{\mathbf{U}^T \mathbf{R}^{-1} \mathbf{U}} \right) \right], \quad (2.3)$$

where  $[\mathbf{R}]_{ij} = E[I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_i))I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_j))]$  and  $[\mathbf{P}]_j = E[I_k(\mathbf{s})I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_j))]$  respectively stand for auto- and cross-correlation matrices while  $\mathbf{U}$  is a column vector with all elements equal to one [5]. Given that the underlying intensity and motion fields are stationary and that motion samples are taken on a square lattice (such is the case when an image is divided into a group of square blocks for motion search), the optimal weights  $\mathbf{w}^*$  for pixel  $\mathbf{s}$  depend solely on its relative position within a block. They are often obtained using the least-squares approach, due to a lack of probabilistic models of real data.

The concept of OBMC can generalize to the case of irregular motion sampling structure. Since that both auto- and cross-correlation functions are spatially varying, the challenge lies within the weighting coefficient optimization of each pixel associated with nearby MVs. The least-squares solution, although feasible in theory, is impractical because the storage of weighting coefficients optimized for different contexts is spatially-demanding. A parametric solution which have proposed in [10] is an alternative to tackle this problem.

## 2.5 Parametric OBMC

POBMC, which is proposed in [10], gives a closed-form formula for the optimal weights. To do so, they need to assume signal models for the intensity and motion fields, which

gives a direct estimate of the optimal weights  $\mathbf{w}^*$ . This is accomplished by using the motion model proposed in [11], which assumes that the difference between the true motion of any two pixels, e.g.,  $\mathbf{s}_1$  and  $\mathbf{s}_2$ , has a normal distribution of the form

$$v_x(\mathbf{s}_1) - v_x(\mathbf{s}_2) \text{ or } v_y(\mathbf{s}_1) - v_y(\mathbf{s}_2) \sim \mathcal{N}(0, \alpha r^2(\mathbf{s}_1, \mathbf{s}_2)), \quad (2.4)$$

where  $\alpha$  is a positive number indicating the degree of motion randomness in horizontal or vertical direction<sup>2</sup>, and  $r(\mathbf{s}_1, \mathbf{s}_2)$  is the  $\ell^2$  distance (measured in the unit of pixel) between  $\mathbf{s}_1$  and  $\mathbf{s}_2$ .

With the signal model in (2.4), they next proceed to determine the optimal weights  $\mathbf{w}^*$  using *calculus*. To begin with, they rewrite, by noting that  $\sum_{i=1}^L w_i = 1$ , the mean squared prediction error  $\zeta(\mathbf{w})$  in Eq. (2.1) as

$$\xi(\mathbf{w}) = E \left\{ \left( \sum_{i=1}^L w_i d(\mathbf{s}; \mathbf{v}(\mathbf{s}_i)) \right)^2 \right\}, \quad (2.5)$$

where  $d(\mathbf{s}; \mathbf{v}(\mathbf{s}_i)) = I_k(\mathbf{s}) - I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_i))$  denotes the residual signal when  $I_k(\mathbf{s})$  is predicted from the motion-compensated signal  $I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_i))$  using the MV for block  $i$ ,  $\mathbf{v}(\mathbf{s}_i)$ .

To continue, they borrow a result in [11], which shows that if (2.4) is valid, then  $E\{d^2(\mathbf{s}; \mathbf{v}(\mathbf{s}_i))\}$  has a closed-form formula given by

$$E\{d^2(\mathbf{s}; \mathbf{v}(\mathbf{s}_i))\} = E\{(I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s})) - I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_i)))^2\} = \epsilon r^2(\mathbf{s}, \mathbf{s}_i), \quad (2.6)$$

where  $\epsilon$  is a constant indicating the joint randomness of the motion and intensity fields;  $I_k(\mathbf{s}) = I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}))$  with  $\mathbf{v}(\mathbf{s})$  denoting the true motion of pixel  $\mathbf{s}$ ; and the block MV  $\mathbf{v}(\mathbf{s}_i)$  is approximated as the motion associated with the block center  $\mathbf{s}_i$ . What remains to be determined is those non-diagonal terms, i.e.,  $E\{d(\mathbf{s}; \mathbf{v}(\mathbf{s}_i))d(\mathbf{s}; \mathbf{v}(\mathbf{s}_j))\}$ ,  $i \neq j$ . Under some mild conditions, they assume that the prediction errors  $\{d(\mathbf{s}; \mathbf{v}(\mathbf{s}_i))\}_{i=1}^L$  are uncorrelated to each other, i.e.,  $E\{d(\mathbf{s}; \mathbf{v}(\mathbf{s}_i))d(\mathbf{s}; \mathbf{v}(\mathbf{s}_j))\} = 0, \forall i \neq j$ . Thus, they rewrite (2.5) to have the following form:

$$\xi(\mathbf{w}) = \sum_{i=1}^L \epsilon w_i^2 r^2(\mathbf{s}; \mathbf{s}_i), \quad (2.7)$$

<sup>2</sup>The smaller  $\alpha$  value suggests the motion field has higher spatial correlation.

Upon setting the gradient of  $\xi(\mathbf{w})$  with respect to  $\mathbf{w}$  to 0, the optimal weights  $\mathbf{w}^*$  becomes

$$\mathbf{w}^* = \left( \sum_{i=1}^L \frac{1}{r^2(\mathbf{s}, \mathbf{s}_i)} \right)^{-1} \left[ \frac{1}{r^2(\mathbf{s}, \mathbf{s}_1)}, \frac{1}{r^2(\mathbf{s}, \mathbf{s}_2)}, \dots, \frac{1}{r^2(\mathbf{s}, \mathbf{s}_L)} \right]^T. \quad (2.8)$$

The significance of this result is that it requires only the geometry relations of pixel  $\mathbf{s}$  and its nearby block centers  $\{\mathbf{s}_i\}_{i=1}^L$  to obtain  $\{w_i^*\}_{i=1}^L$ . This remarkable property allows MVs associated with any PUs to be incorporated for OBMC.



# CHAPTER 3

---

## Combining Template and Block Motion Compensations

---

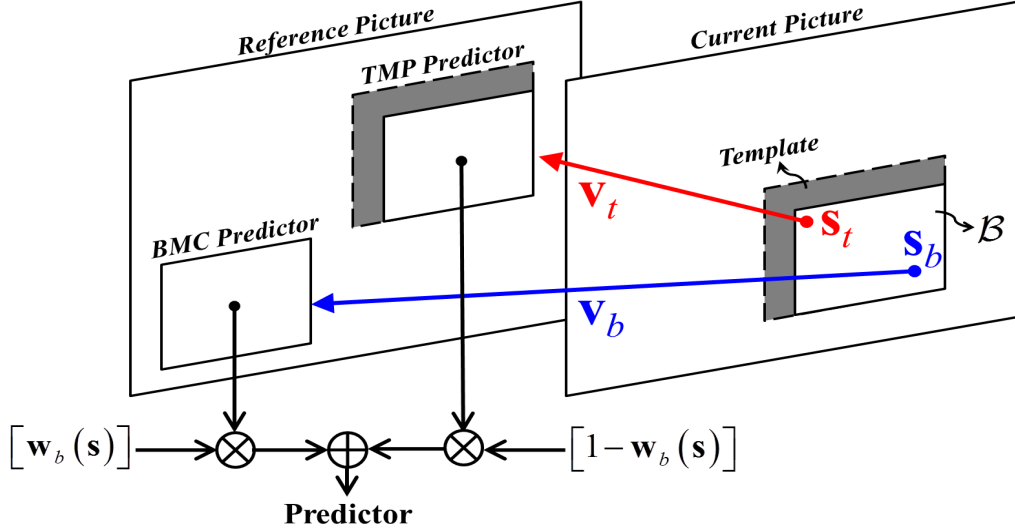


### 3.1 Concept of Operation

Figure 3.1 depicts the basic concept of the proposed scheme. Like the conventional bi-prediction, it predicts a target PU based on two predictors. These predictors however are weighted in a pixel-adaptive manner using POBMC [10], with one of them derived from a MV  $\mathbf{v}_t$  found by TMP [1][2][3] and the other from the usual motion compensation. Since  $\mathbf{v}_t$  can be inferred on the decoder side, this scheme has to signal motion parameters for only one block MV (denoted as  $\mathbf{v}_b$ ). Additionally, we restricted  $\mathbf{v}_b$  to be uni-directional prediction here in order to reduce the motion cost needed for bi-prediction.

### 3.2 Optimized Block Matching Criterion

In our scheme,  $\mathbf{v}_t$  cannot be specified discretionarily. It is thus important to find a  $\mathbf{v}_b$  that, when applied jointly with  $\mathbf{v}_t$ , minimizes the mean square prediction error over



**Figure 3.1:** Joint application of TMP and BMC.

the target block  $\mathcal{B}$  by POBMC framework. It should be noted that we restrict different PUs of the same size share the identical window functions. As a result, the definition of  $\mathbf{v}_b$  is described as (1.1).

One of the solution of minimizing (1.1) is the least-squares method, which is an under-determined problem since a distinct solution has to be sought for each possible context. Although the least-squares method is feasible and optimal in theory, it's still impractical since the training process is too much time consuming.

Instead of least-squares method, we resort to the parametric framework in Section 2.5. To proceed, we start with an exploration of its average behavior. According to the motion sampling positions in a target PU  $\mathcal{B}$ ,  $\mathbf{v}_t$  is approximate to the true motion  $\mathbf{v}(\mathbf{s}_t)$  of pixel  $\mathbf{s}_t$  at the template centroid [12]. However, we avoid making the same approximation for  $\mathbf{v}_b$  because the search criterion is no longer to minimize the sum of squared prediction errors<sup>1</sup> (cf. (1.1)).  $\mathbf{v}_b$  is approximatd as the true motion of some unknown pixel  $\mathbf{b}$  in  $\mathcal{B}$ . Now the problem of determining  $w_b(\tilde{\mathbf{s}})$  as the search for an optimal sampling position  $\mathbf{s}_b$ ,  $\mathbf{s}_b \in \mathcal{B}$  that minimizes the sum of mean squared prediction errors (SMSE) over  $\mathcal{B}$ :

$$E \left[ \sum_{\mathbf{s} \in \mathcal{B}} (I_k(\mathbf{s}) - w_t(\tilde{\mathbf{s}}) I_{k-1}(\mathbf{s} + \mathbf{v}_t) - w_b(\tilde{\mathbf{s}}) I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{b})))^2 \right]. \quad (3.1)$$

<sup>1</sup>A block MV approximates the pixel true motion at the block center only if its search criterion is to minimize the sum of squared prediction errors [11][9].



To compute the expectation in (3.1), we replace  $I_k(\mathbf{s}) - I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_i))$  with  $d(s; \mathbf{v}(\mathbf{s}_i))$ , which denotes the residual signal when  $I_k(\mathbf{s})$  is predicted from the motion-compensated signal  $I_{k-1}(\mathbf{s} + \mathbf{v}(\mathbf{s}_i))$ , and then rewrite (3.1) as

$$\sum_{\mathbf{s} \in \mathcal{B}} E [(w_t(\tilde{\mathbf{s}}) d(s; \mathbf{v}(\mathbf{s}_t)) + w_b(\tilde{\mathbf{s}}) d(s; \mathbf{v}(\mathbf{b})))^2]. \quad (3.2)$$

We assume the prediction errors in the target PU  $\mathcal{B}$  are uncorrelated with each other, i.e.,  $E[d(s; \mathbf{v}(\mathbf{s}_b))d(s; \mathbf{v}(\mathbf{s}_t))] = 0$ , then (3.2) is approximate to

$$\sum_{\mathbf{s} \in \mathcal{B}} w_t(\tilde{\mathbf{s}})^2 E[d^2(s; \mathbf{v}(\mathbf{s}_t))] + w_b(\tilde{\mathbf{s}})^2 E[d^2(s; \mathbf{v}(\mathbf{b}))]. \quad (3.3)$$

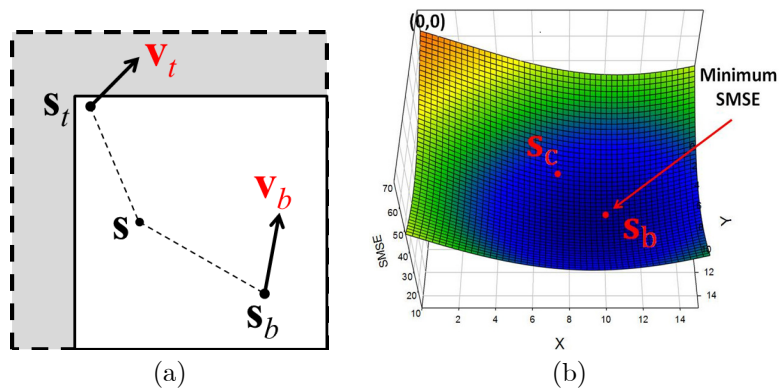
Here we borrow the result in (2.6), that is,  $E[d^2(s; \mathbf{v}(\mathbf{s}_b))]$  and  $E[d^2(s; \mathbf{v}(\mathbf{s}_t))]$  have closed-form formulas given by  $\epsilon r^2(\mathbf{s}; \mathbf{s}_b)$  and  $\epsilon r^2(\mathbf{s}; \mathbf{s}_t)$ . Moreover, according to (2.8), we have  $w_b(\tilde{\mathbf{s}}) = w_b^*(\tilde{\mathbf{s}}) = r^2(\mathbf{s}; \mathbf{s}_t) / (r^2(\mathbf{s}; \mathbf{s}_t) + r^2(\mathbf{s}; \mathbf{b}))$ . Hence, (3.3) becomes

$$\sum_{\mathbf{s} \in \mathcal{B}} \epsilon (w_t^*(\mathbf{s}))^2 r^2(\mathbf{s}; \mathbf{s}_t) + (w_b^*(\mathbf{s}))^2 r^2(\mathbf{s}; \mathbf{b}). \quad (3.4)$$

Due to the non-linear nature of (3.4),  $\mathbf{s}_b$  must be found by numerical method, that is, to compute SMSE for every admissible location of  $\mathbf{b}$ . Once it is solved, the  $w_b^*(\mathbf{s})$  and  $w_t^*(\mathbf{s})$  are thus obtained immediately by (2.8). Then, (1.1) is reformulated as

$$\mathbf{v}_b^* = \arg \min_{\mathbf{v}_b} \sum_{\mathbf{s} \in \mathcal{B}} (I_k(\mathbf{s}) - w_t^*(\mathbf{s}) I_{k-1}(\mathbf{s} + \mathbf{v}_t) - w_b^*(\mathbf{s}) I_{k-1}(\mathbf{s} + \mathbf{v}_b))^2. \quad (3.5)$$

To verify where  $\mathbf{s}_b$  should be located, we take an example as illustrated in Fig. 3.2 (a). In such case, Fig. 3.2 (b) plots the SMSE surface as a function of  $\mathbf{b}$  according to (3.4). As can be seen, SMSE value decreases when  $\mathbf{b}$  approaches to the the bottom right quarter. A more precise calculation shows that the optimal location of  $\mathbf{b}$  (thus  $\mathbf{s}_b$ ) occurs at point (9.5, 9.5) for a  $16 \times 16$  target block. The result is no surprising because  $\mathbf{v}_t$ , located at the template centroid (1.9, 1.9), has a higher correlation with the motion field lying to the upper left quarter and thus contributes more to minimizing the prediction errors of the upper-left part. Intuitively,  $\mathbf{s}_b$  should be placed in the



**Figure 3.2:** (a) geometry relationship between TMP centroid, and  $\mathbf{s}_b$ . (b) SMSE surface as a function of the location of pixel  $\mathbf{b}$ .

bottom-right quarter to minimize the prediction errors in the remaining part of  $\mathcal{B}$ .

### 3.3 Window Functions

In this thesis, five different template designs are evaluated in  $2N \times 2N$  PUs as described in the first column of Fig. 3.3, while the second and third columns plot the corresponding window functions of  $w_t^*(\mathbf{s})$  and  $w_b^*(\mathbf{s})$ , for  $\mathbf{v}_t$  and  $\mathbf{v}_b$ . The waveforms of template shapes, e.g. AL, suggest a special type of geometry motion partitionings [13] with two MVs located on the diagonal running from above-left to bottom-right corners within a PU. Also, AR and BL follow the same rationale. Following the same line of derivation, we can obtain the window functions for those rectangular template designs. In particular, asymmetric-like motion partitionings [13][7] result when the template region locates directly above or to the left of a target PU (cf. Fig. 3.3). Two conceptual differences however are to be noted. First, unlike explicit geometry or asymmetric partitions, these implicit “soft” partitions incur less motion cost (only one MV is to be signaled). Second, there is a strong interdependency between the transmitted and inferred MVs because of OBMC (cf. (1.1)).

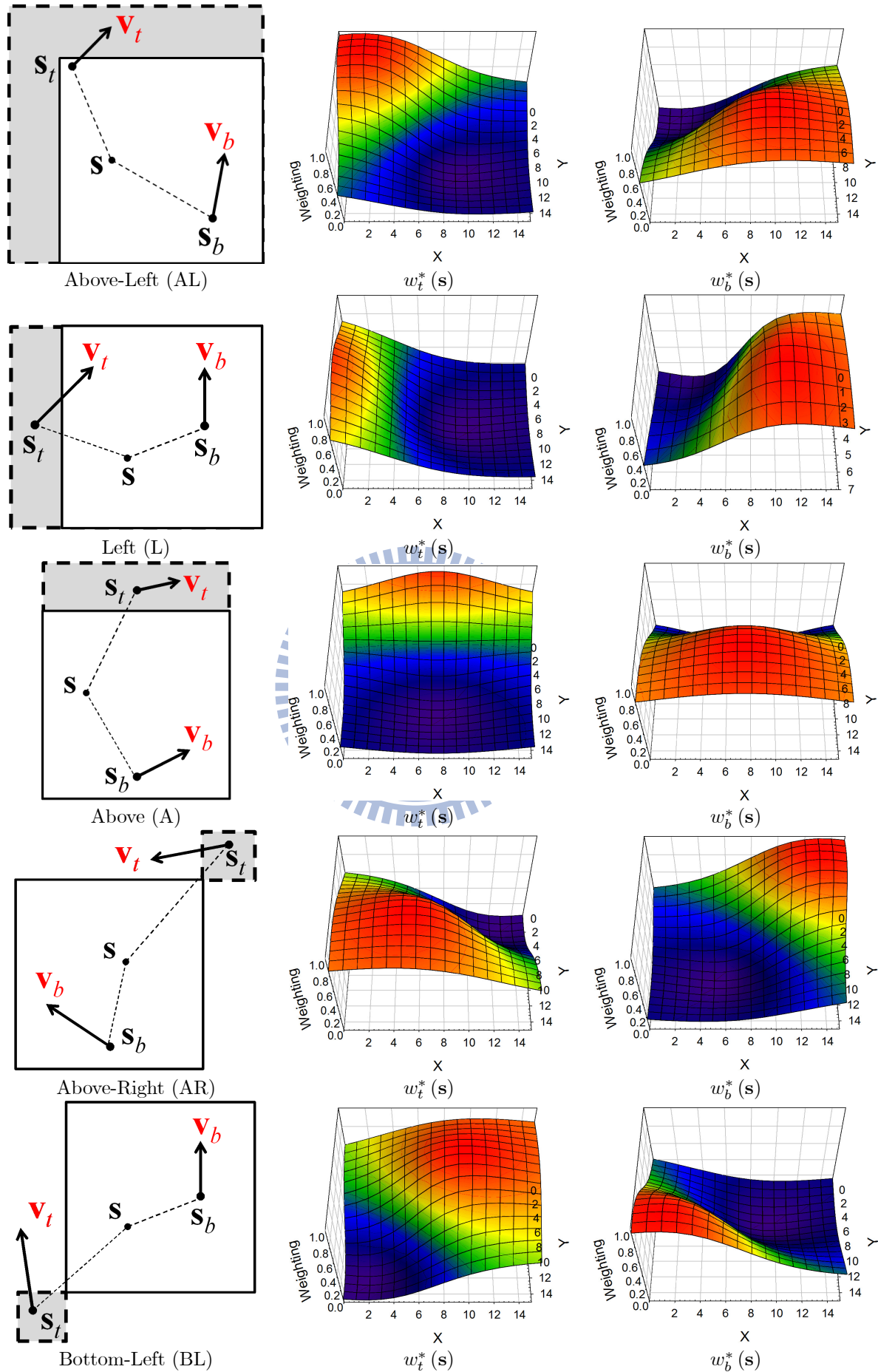


Figure 3.3: Window functions for typical template designs.

# CHAPTER 4

---

## Experiments

---

### 4.1 Experimental Conditions

#### 4.1.1 Common Test Conditions

In this chapter, the experiments are conducted based on the HEVC reference software HM-3.0 and the HEVC common test conditions (JCTVC-E700 [14]). The HEVC common test conditions are desirable to configure experiments in a well-defined environment and ease the comparison of the outcome of experiments. JCTVC-E700 defines eight different test conditions, but only four of them are related to bi-directional inter-frame coding:

- Random access, high efficiency (RAHE).
- Random access, low complexity (RALC).
- Low delay, high efficiency (LDHE).
- Low delay, low complexity (LDLC).

Each test condition has a specific configuration with the ON/OFF of coding tools which are summarized in Tab. 4.1. Our proposed scheme are tested based on those test conditions in order to compare their BD-rate savings [6] with HM-3.0 anchor.

Encoder Configurations	RAHE	RALC	LDHE	LDLC
GOP Size	8	8	1	1
NumOfReference	L0:2, L1:2		L0:4	
Entropy Coder	CABAC	CAVLC	CABAC	CAVLC
Adaptive Loop Filter (ALF)	Y	N	Y	N
Internal Bit Depth (IBDI)	10	8	10	8
QP	22, 27, 32, 37			
Sequences	1080p, 832 × 480, 416 × 240, 720p			
CU Sizes	8 × 8 ~ 64 × 64			
Search Range	±64			
Bi-Prediction Search Range	±4			
Interpolation Filter	8-tap DCT-IF			

**Table 4.1:** Common test conditions.

Rough estimations of complexity are performed by showing the encoding time ratio and decoding time ratio relative to HM-3.0 anchor.

### 4.1.2 TB-mode

For the configuration of our proposed template-based bi-prediction scheme (referred hereafter as *TB-mode*), we applied it only to  $2N \times 2N$  PUs. Three (AL, L, and A) or five (AL, L, A, AR, and BL) template shapes are fetched from the reconstructed frame with template width 4. For each  $2N \times 2N$  PU, one flag is set to switch adaptively between TB-mode and the usual inter mode. When the former is chosen, it codes at most two (three templates) or three (five templates) extra bits to specify the template shapes.

Moreover, in this chapter, we have two types of window functions to be evaluated on TB-mode. One is formed by the theoretical window functions that have been mentioned in Section 3.3 and the other is formed by a heuristic design of window functions. For theoretical window functions, the weighting coefficients are rounded offline into 16-bit integers. On the other hand, the weighting coefficients of the heuristic window functions are represented in 3-bit integers. To verify the performance of TB-mode, several experiments featuring different performance and complexity trade-offs are summarized in Tab. 4.2 and will be discussed in the following sections.

**Table 4.2:** Experimental settings of TB-mode.

Algo.	Template	Window Functions	TMP SrchRng	TMP-Bi SrchRng	Hypothesis	Fast Algo.
T3-C-UU	3 shapes	Theoretical	$\pm 4$	N/A	2	N
T5-C-UU	5 shapes	Theoretical	$\pm 4$	N/A	2	N
T5-S-UU	5 shapes	Heuristic	$\pm 4$	N/A	2	N
T5-S-UB	5 shapes	Heuristic	$\pm 4$	N/A	3	N
T5-S-BU	5 shapes	Heuristic	$\pm 4$	$\pm 1$	3	N
T5-S-BB	5 shapes	Heuristic	$\pm 4$	$\pm 1$	4	N
T3-S-F	3 shapes	Heuristic	$\pm 4$	$\pm 1$	4	Y

PU Size	Pos	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$w_{1,4}$	$w_{1,5}$	PU Size	Pos	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$	$w_{2,5}$
<b>64x64</b>	a	47	15	15	15	15	<b>64x64</b>	a	62	31	31	31	31
	b	2	2	2	2	2		b	2	2	2	2	2
<b>32x32</b>	a	23	7	7	7	7	<b>32x32</b>	a	30	15	15	15	15
	b	2	2	2	2	2		b	2	2	2	2	2
<b>16x16</b>	a	11	3	3	3	3	<b>16x16</b>	a	14	7	7	7	7
	b	2	2	2	2	2		b	2	2	2	2	2
<b>8x8</b>	a	5	1	1	1	1	<b>8x8</b>	a	6	3	3	3	3
	b	2	2	2	2	2		b	2	2	2	2	2

**Table 4.3:** The start position **a** and **b** for various  $2N \times 2N$  PU sizes.

## 4.2 Heuristic Window Functions

Each PU, when coded in the proposed scheme, has multiple window functions as denoted by  $w_{n,m}$  with  $n = 1, 2$  and  $m = 1, \dots, 5$ . The parameter  $n$  is explicitly signaled in one extra flag, and the value of  $m$  is inferred according to the choice of the template shapes. The coefficient value of each  $w_{n,m}$  takes values from either the set  $\{0, 1, 4, 7\}$  or the set  $\{0, 1, 4, 6\}$ , and thus the multiplication by a floating-point number can be easily replaced by an integer arithmetic. Their waveforms illustrated in Fig. 4.1 form a partitioning of a PU into four non-overlapping regions, and each region corresponds to a specific coefficient. It should be noted that the zero numbers cover over half or three-fourth region of a window function. Pixels in that region are not compensated by OBMC, which can effectively halve the expense of memory bandwidth.

To resize a window function according to the size of the considered PU, the start point  $a$  and the width  $b$  are recorded. Tab. 4.3 lists the values of  $a$  and  $b$  for every possible  $2N \times 2N$  PU size. In view of this resizing criterion, the storage requirements for weighting coefficients are thus conspicuously reduced.

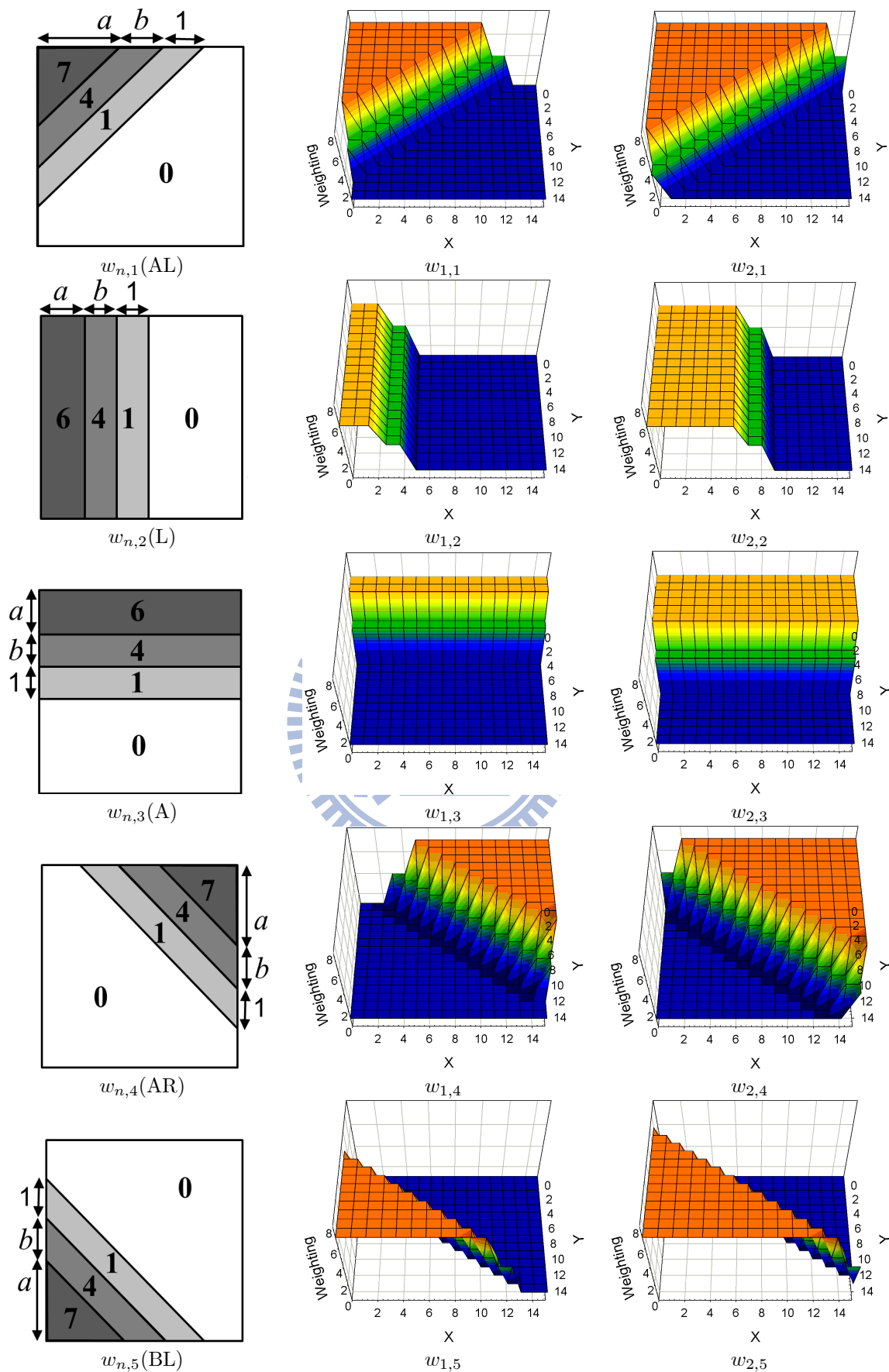


Figure 4.1: Designs of heuristic window functions.

**Table 4.4:** BD-rate savings and processing time ratios of TB-mode with 3- and 5-shape-adaptive configurations.

<i>Random Access</i>	<b>RAHE</b>		<b>RALC</b>	
Algo.	T3-C-UU	T5-C-UU	T3-C-UU	T5-C-UU
<b>S03/S05/S06</b>	-1.2	-1.3	-1.4	-1.5
<b>Class C</b>	-2.0	-2.1	-1.7	-1.8
<b>Class D</b>	-1.9	-2.0	-1.6	-1.7
<b>All</b>	-1.7	-1.9	-1.6	-1.7
<b>Enc. Time [%]</b>	176	184	175	182
<b>Dec. Time [%]</b>	172	175	194	197
<i>Low Delay</i>	<b>LDHE</b>		<b>LDLC</b>	
Algo.	T3-C-UU	T5-C-UU	T3-C-UU	T5-C-UU
<b>S03/S05/S06</b>	-1.9	-2.0	-2.4	-2.6
<b>Class C</b>	-2.3	-2.4	-2.4	-2.5
<b>Class D</b>	-2.1	-2.2	-2.2	-2.4
<b>Class E</b>	-3.3	-3.6	-3.4	-3.6
<b>All</b>	-2.4	-2.5	-2.6	-2.7
<b>Enc. Time [%]</b>	157	166	155	163
<b>Dec. Time [%]</b>	220	225	269	273

### 4.3 Compression Performance of TB-mode

This section illustrates the compression performance and complexity of TB-mode by restricting the two MVs of TMP and the target PU to uni-directional predictions. Both theoretical and heuristic window functions as well as 3- or 5-shape-adaptive implementations are evaluated.

#### 4.3.1 Coding Efficiency versus Number of Templates

We first focus on the coding efficiency between 3- and 5-shape-adaptive implementations. Tab. 4.4 presents the average BD-rate savings of T3-C-UU and T5-C-UU. The former experiment is the typical TB-mode with 3-shape-adaptive theoretical window functions, while the latter shows the result of 5-shape-adaptive TB-mode with theoretical window functions. Clearly, T5-C-UU evaluates two additional templates in each  $2N \times 2N$  PU at the encoder, these additional RD-comparisons constantly delivers about 0.1% coding gains at the cost of increasing encoding complexity. Since the sizes of the two additional templates are small, the time ratio increment at the encoder side is only 14.3%. Moreover, 4% increase of decoding time is observed due to the two additional template searches performed at the decoder side.



**Table 4.5:** BD-rate savings and processing time ratios of TB-mode with theoretical and heuristic window functions.

<i>Random Access</i>	<b>RAHE</b>		<b>RALC</b>	
Algo.	T5-C-UU	T5-S-UU	T5-C-UU	T5-S-UU
<b>S03/S05/S06</b>	-1.3	-1.3	-1.5	-1.4
<b>Class C</b>	-2.1	-2.2	-1.8	-2.0
<b>Class D</b>	-2.0	-2.3	-1.7	-2.0
<b>All</b>	-1.9	-2.0	-1.7	-1.8
<b>Enc. Time [%]</b>	184	207	182	210
<b>Dec. Time [%]</b>	175	166	197	186
<i>Low Delay</i>	<b>LDHE</b>		<b>LDLC</b>	
Algo.	T5-C-UU	T5-S-UU	T5-C-UU	T5-S-UU
<b>S03/S05/S06</b>	-2.0	-2.0	-2.6	-2.1
<b>Class C</b>	-2.4	-2.6	-2.5	-2.7
<b>Class D</b>	-2.2	-2.6	-2.4	-2.7
<b>Class E</b>	-3.6	-3.5	-3.6	-3.2
<b>All</b>	-2.5	-2.6	-2.7	-2.6
<b>Enc. Time [%]</b>	166	186	163	186
<b>Dec. Time [%]</b>	225	216	273	262

### 4.3.2 Theoretical versus Heuristic Window Functions

Here we focus on the comparison between the design of theoretical and heuristic window functions, which are denoted by T5-C-UU and T5-S-UU in Tab. 4.5. Experimental results of T5-C-UU and T5-S-UU reveals that the additional set of heuristic window functions not only compensates the coding loss after the simplification of weighting coefficients, but also slightly increases 0.1% coding gains on average. For the encoding time increment, although each set of heuristic window functions reduces computation overhead more than a theoretical one, the extra set of RD-comparisons still brings about 24% increments on encoding time ratio. With regard to decoding complexity, since zero weighting coefficients reduces the computations for performing OBMC, the decoding time drops at about 10%.

## 4.4 Multiple Hypotheses

In this section, we discuss the effect of coding efficiency when multiple hypotheses are enabled for template and block motions. Experiments of multiple hypotheses are tested based on 5-shape-adaptive TB-mode and heuristic window functions.

In Tab. 4.6, we compare T5-S-UU with the three methods which are specified as follows:

**Table 4.6:** BD-rate savings and processing time ratios of enabling multiple-hypotheses.

<i>Random Access</i>	<b>RAHE</b>				<b>RALC</b>			
Algo.	T5-S-UU	T5-S-UB	T5-S-BU	T5-S-BB	T5-S-UU	T5-S-UB	T5-S-BU	T5-S-BB
<b>S03/S05/S06</b>	-1.3	-1.4	-1.4	-1.9	-1.4	-1.7	-1.6	-2.2
<b>Class C</b>	-2.2	-2.4	-2.5	-2.8	-2.0	-2.3	-2.2	-2.6
<b>Class D</b>	-2.3	-2.4	-2.5	-2.8	-2.0	-2.3	-2.2	-2.6
<b>All</b>	-2.0	-2.1	-2.2	-2.5	-1.8	-2.1	-2.0	-2.5
<b>Enc. Time [%]</b>	207	285	250	301	210	297	259	302
<b>Dec. Time [%]</b>	166	169	239	262	186	197	281	327
<i>Low Delay</i>	<b>LDHE</b>				<b>LDLC</b>			
Algo.	T5-S-UU	T5-S-UB	T5-S-BU	T5-S-BB	T5-S-UU	T5-S-UB	T5-S-BU	T5-S-BB
<b>S03/S05/S06</b>	-2.0	-2.0	-2.2	-2.3	-2.1	-2.3	-2.4	-2.6
<b>Class C</b>	-2.6	-2.7	-2.8	-3.0	-2.7	-2.8	-2.9	-3.1
<b>Class D</b>	-2.6	-2.9	-2.9	-3.2	-2.7	-3.0	-2.9	-3.2
<b>Class E</b>	-3.5	-3.4	-3.7	-3.8	-3.2	-3.5	-3.6	-4.0
<b>All</b>	-2.6	-2.8	-2.9	-3.1	-2.6	-2.9	-2.9	-3.2
<b>Enc. Time [%]</b>	186	269	250	253	186	280	257	263
<b>Dec. Time [%]</b>	216	216	368	375	262	269	464	485

- T5-S-UB: Experiment of enabling bi-prediction to block motions.
- T5-S-BU: Experiment of enabling bi-prediction to template motions.
- T5-S-BB: Experiment of enabling bi-prediction to both template and block motions.

Averagely, T5-S-UB outperforms 0.2% in terms of BD-rate saving with 5% average decoding time increment. Enabling bi-prediction to the target PU for finding block motions almost has no effect on decoding time complexity. T5-S-BU reflects similar benefits to T5-S-UB with 0.3% BD-rate saving. Nevertheless, since the bi-prediction of TMP performs at the decoder side, the decoding time dramatically increases about 136% on average. On the other hand, it is interesting that the encoding time increment of T5-S-UB is 30% higher than T5-S-BU. The reason is that the size of templates and the range of template matching are generally smaller than the size of target PUs and the range of block motion search.

T5-S-BB enables bi-directional to both template and block motion search, which reaches an average BD-rate saving of 2.9% and a maximum BD-rate saving up to 5.2%. Although the performance of T5-S-BB are very impressive between all the configurations for TB-mode, the significantly increased encoding and decoding times make this scheme less practical. As a result, a further reduction in TB-mode complexity is necessary.

## 4.5 Fast Algorithm

As concluded in previous section, several enhancements of TB-mode are conducted to achieve the decreased time complexity and the moderate coding gains. To tackle the complexity issue of TB-mode, we start with the modification from T5-S-BB, which has promising coding gains and copious runtimes over all the TB-mode experiments. As summarized below, four major enhancements will be applied for speedup the runtimes of TB-mode:

- Reduce the number of template shapes moderately.
- Fast mode decision by skip TB-mode when SKIP mode has lowest RD cost among all the other modes.
- Limit the number of reference frames to be searched.
- Use bilinear filter for sub-pel interpolation during the TMP process.

### 4.5.1 Enhancements for Encoder Only

The major contribution of encoding time is the extra mode decision process of TB-mode R-D comparisons. As an additional prediction mode, decreasing the number of TB-mode evaluations is one way to reduce the encoding time complexity. According to our observation, the area size of enabling SKIP mode adjusts slightly before and after TB-mode is applied. This observation implies that the encoder is unlikely to choose TB-mode when SKIP mode is the best candidate. As a result, if the best mode is SKIP, we bypass the TB-mode evaluations. Moreover, since the 3-shape-adaptive TB-mode drops neglectable coding gains than the 5-shape adaptive one, we reduce the number of template shapes to be tested for further reducing encoding time complexity.

### 4.5.2 Enhancements for Encoder and Decoder

TMP performs its motion search on both encoder and decoder sides, which has a great impact of TB-mode complexity. To diminish the motion cost caused by TMP, two approaches have been taken: The first is to reduce the number of reference frames tested in TMP motion search, and the second is to simplify the interpolation filter applied during the TMP motion search of fractional-pel MV refinements.

To resolve the former issue, we truncate the number of reference frames tested in

**Table 4.7:** BD-rate savings and processing time ratios of TB-mode after applying fast algorithms.

<i>Random Access</i>	<b>RAHE</b>		<b>RALC</b>	
Algo.	T3-S-F	T5-S-BB	T3-S-F	T5-S-BB
<b>S03/S05/S06</b>	-1.6	-1.9	-1.8	-2.2
<b>Class C</b>	-2.3	-2.8	-2.1	-2.6
<b>Class D</b>	-2.1	-2.8	-2.0	-2.6
<b>All</b>	-2.0	-2.5	-2.0	-2.5
<b>Enc. Time [%]</b>	149	301	153	302
<b>Dec. Time [%]</b>	124	262	136	327
<i>Low Delay</i>	<b>LDHE</b>		<b>LDLC</b>	
Algo.	T3-S-F	T5-S-BB	T3-S-F	T5-S-BB
<b>S03/S05/S06</b>	-1.7	-2.3	-1.9	-2.6
<b>Class C</b>	-2.3	-3.0	-2.3	-3.1
<b>Class D</b>	-2.5	-3.2	-2.5	-3.2
<b>Class E</b>	-2.9	-3.8	-3.1	-4.0
<b>All</b>	-2.5	-3.1	-2.5	-3.2
<b>Enc. Time [%]</b>	142	253	144	263
<b>Dec. Time [%]</b>	128	375	144	485

TB-mode. These reference frames are derived by referring to the reference indices used in the  $2N \times 2N$  MRG mode during the mode decision process. If there is only one available reference index or there are duplicated reference indices, an additional reference frame with the lowest QP in GOP structure is considered as another candidate to be evaluated in TB-mode.

In the latter issue, we revise the interpolation filter of TMP fractional-pel motion search by interpolating the reference PU with a bilinear filter. The bilinear filter brings a conspicuous complexity reduction however, it also generates poor template motions resulting in a coding loss. Fortunately, this inefficiency can be partially compensated by other MVs (thus the block motions) in TB-mode.

### 4.5.3 Summary

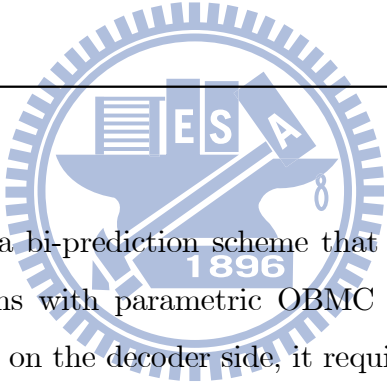
Experiment T3-S-F describes the performance of TB-mode after applying those enhancements introduced in this section. As in Tab. 4.7, T3-S-F has a moderate to significant average BD-rate saving of 2.2%, with a minimum of 1.1% and a maximum of 4.1% over all test cases. Although T3-S-F has an average coding loss of 0.7% compared with T5-S-BB, 131% for encoding and 237% for decoding time consumption are still impressive in reducing time complexity.

# CHAPTER 5

---

## Conclusion

---



In this thesis, we propose a bi-prediction scheme that combines predictors found by template and block motions with parametric OBMC window functions. Since the template motion is inferred on the decoder side, it requires only a motion cost as that of uni-directional prediction. For optimizing the motion parameters to be signaled, the motion search criterion is modified to reflect the interdependency between  $\mathbf{v}_b$  and  $\mathbf{v}_t$ . The choice of window function is based on the inferred MV constellation, which brings a better adaptation and prediction efficiency. Refer to the experimental results, a promising coding gain (2.9%) brings a cost of significant increase in both the encoding and decoding times. As a result, several modifications are made to strike a better balance between performance and complexity. After applying those modifications, the best scheme shows moderate-to-significant coding gains (2.2%) with reasonable complexity increments (46% and 33%). This result shows that it is possible to keep the gain while reducing the runtime to a great extent.

Our works remains in its early stage. We believe that there is still plenty of room for improvement. A possible direction is to study the low-complexity and TMP-free implementations. This can be accomplished by replacing the  $\mathbf{v}_t$  with one of the other

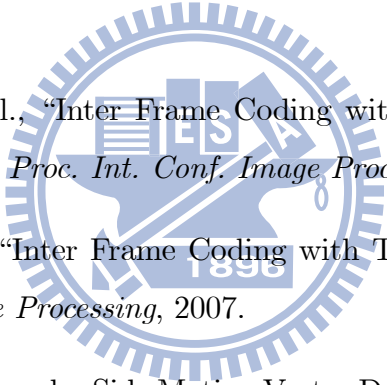
decoded MVs from neighboring PUs. In this manner, the need to perform TMP is waived at the cost of extra bits. We shall continue these investigations in our future work.



---

## Bibliography

---

- 
- [1] K. Sugimoto and et al., “Inter Frame Coding with Template Matching Spatio-Temporal Prediction,” *Proc. Int. Conf. Image Processing*, 2004.
- [2] Y. Suzuki and et al., “Inter Frame Coding with Template Matchin Averaging,” *Proc. Int. Conf. Image Processing*, 2007.
- [3] S. Kamp and et al., “Decoder Side Motion Vector Derivation for Inter Frame Video Coding,” *Proc. Int. Conf. Image Processing*, 2008.
- [4] S. Nogaki and M. Ohta, “An overlapped block motion compensation for high quality motion picture coding,” *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 184–187, May 1992.
- [5] M. T. Orchard and G. J. Sullivan, “Overlapped Block Motioin Compensation: An Estimation-Theoretic Approach,” *IEEE Trans. on Image Processing*, vol. 3, pp. 693–699, May 1994.
- [6] G. Bjontegaard, “Improvements of the BD-PSNR Model,” *ITU-T SG16 Q.6 Document, VCEG-A111*, Jul. 2008.
- [7] K. McCann and et al., “Samsung’s Response to the Call for Proposals on Video Compression Technology,” *JCTVC-A124*, Apr. 2010.

- [8] M. Winken and et al., “Description of Video Coding Technology Proposal by Fraunhofer HHI,” *JCTVC-A116*, Apr. 2010.
- [9] B. Tao and M. Orchard, “A Parametric Solution for Optimal Overlapped Block Motion Compensation,” *IEEE Trans. on Image Processing*, vol. 10, pp. 341–350, Mar. 2001.
- [10] Y. W. Chen and W. H. Peng, “Parametric OBMC for Pixel-Adaptive Temporal Prediction on Irregular Motion Sampling Grids,” *IEEE CSVT*, 2011.
- [11] W. Zheng and et al., “Analysis of Space-dependent Characteristics of Motion-compensated Frame Differences based on a Statistical Motion Distribution Model,” *IEEE Trans. on Image Processing*, vol. 11, pp. 377–386, Mar. 2002.
- [12] T.-W. Wang and et al., “Analysis of Template Matching Prediction and its Application to Parametric Overlapped Block Motion Compensation,” *IEEE ISCAS*, 2010.
- [13] M. Karczewicz and et al., “Video Coding Technology Proposal by Qualcomm Inc.,” *JCTVC-A121*, Apr. 2010.
- [14] F. Bossen, “Common Test Conditions and Software Reference Configurations,” *JCTVC-E700*, Mar. 2011.