An Interframe Prediction Technique Combining Template Matching Prediction and Block-Motion Compensation for High-Efficiency Video Coding

Wen-Hsiao Peng and Chun-Chi Chen

***Abstract*—This paper introduces an interframe prediction technique that combines two motion vectors (MVs) derived respectively from template and block matching for overlapped block motion compensation (OBMC). It has a salient feature of not having to signal the template MV, while achieving a prediction performance close to that of bi-prediction. We begin by studying template matching prediction (TMP) from a theoretical perspective. Based on two signal models, the template MV is shown to approximate the pixel true motion around the template centroid, through which we explain why TMP generally outperforms SKIP prediction but is inferior to block-based motion compensation in terms of prediction performance. We then approach the problem of finding another MV to best complement the template MV from both deterministic and statistical viewpoints, the latter leading to the search of its optimal sampling location in the motion field. The result is a search criterion with OBMC window functions forming a geometry-like motion partitioning when the template area is straddled on the top and to the left of a target block. Generalizations to adaptive template design, multihypothesis prediction and motion merging are made to explore the complexity and performance trade-offs. Extensive experiments based on the HM-6.0 software show that the best of them, in terms of compression performance, achieves 1.7–2.0% BD-rate reductions at a cost of 26% and 39% increases in encoding and decoding times, respectively.**

***Index Terms*—Adaptive OBMC window design, high efficiency video coding, motion field sampling, overlapped block motion compensation, template matching prediction.**

I. Introduction

A key issue in video coders with motion-compensated prediction is how to trade off effectively between the accuracy of the motion field representation and the required overhead. Often a rough representation of the motion field is sufficient to provide good temporal prediction in terms of rate-distortion (R-D) performance. Obvious evidences are the frequent occurrence of motion estimation and compensation with a large block size in typical H.264/AVC-coded sequences, and of SKIP mode, for which motion information for a target block is even inferred completely from that for nearby coded blocks without being communicated to the decoder.

With the realization that the motion field representation does not have to be accurate in order to be R-D effective, active works began on the investigation of decoder-side motion vector (MV) derivation techniques, hoping to leverage the ever-increasing processing capability of the decoder to save motion overhead. One prominent class of approaches borrows the notion of texture synthesis to perform motion estimation at the decoder [16]. In its initial form, the method, also known as template matching prediction (TMP), obtains the MV at a current pixel by finding, in the reference frames, the best match for a template region composed of its surrounding reconstructed pixels [16]. The recent research [11] further extended the basic unit for TMP from a pixel to a block, resulting in a scheme very similar to the conventional block-based motion compensation (BMC), except that the MV (referred hereafter to as the template MV) is estimated identically at both the encoder and decoder based on minimizing the prediction error accumulated over an inverse-L-shaped template region, which is straddled on the top and to the left of the target block. Despite the increased decoding complexity, this form of TMP has drawn much attention of the video standards community, due in large part to its promising performance and compatibility with the state-of-the-art coding architectures.

Over the years many improvements to TMP have been proposed. For instance, coding the target block at a lower spatial resolution followed by an interpolation was found more R-D efficient in flat areas of an image, where template matching does not always give a physically meaningful MV [18]. In fact, even in nonflat areas, the template MV, by its very nature, is merely a rough estimate of the target block's motion. Hence, the use of multihypothesis prediction to improve the motion compensation accuracy of TMP is very common [7], [10], [17]. Other alternatives include giving higher weight to pixels spatially closer to the target block when calculating the template matching error [9], and adapting the template shape and location to local signal characteristics at the expense of extra side information [9], [14]. Without referring to the content of the target block, these approaches usually cost more computational complexity in order to show a clear R-D benefit.

In view of this, another school of thought strives to form a better prediction at a similar motion cost to BMC. This

Manuscript received April 29, 2010; revised October 16, 2010; accepted December 23, 2012. Date of publication February 21, 2013; date of current version July 31, 2013. This work was supported in part by the National Science Council of Taiwan under Grants 101-2221-E-009-085-MY3 and 102-2218-E-009-003, and Ambarella, Inc. This paper was recommended by Associate Editor M. Hannuksela.

The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: wpeng@cs.nctu.edu.tw; cheerchen.cs98g@g2.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2013.2248214

is again accomplished by performing TMP in the context of multihypothesis prediction, but now requires one of the hypotheses to be derived through a coded MV. Apparently, how to determine and make the most of this MV is the key to its effectiveness. In [17] and [20], it is obtained by carrying out block matching at the encoder as for BMC and then utilized as an initial estimate to confine template matching search. This scheme, however, is not guaranteed to yield a minimal prediction residual, for it neglects to consider the combined effect of the resulting predictors. To overcome this problem, our prior works [4], [6], and [13] proceed in reverse order, starting with template matching (at the encoder) to obtain one predictor, followed by block matching, to estimate a MV, with a criterion that minimizes the difference between the combined prediction signal and the target block. It turns out that these schemes result in much less residual than TMP-only implementations (as well as BMC). Furthermore, this is often achieved with only two hypotheses, forming a particular bi-prediction scheme that features only one coded MV.

One critical step in the above bi-prediction method is the combination of predictors. A simple yet heuristic approach, as adopted in [21], is to compute their weighted average. One limitation of this approach, however, is that pixels in a predictor must be weighted equally, which is superfluous from the theoretical point of view. In order to seek the optimal solution, we turn to the more general weighting scheme of overlapped block motion compensation (OBMC) [15], where the weighting can be pixel adaptive. With this background, the problem is to determine the OBMC weights so that the resulting predictor would produce a minimal residual.

In the paper, we propose two approaches to solve the problem. The first one is the least-squares approach, which relies on an iterative algorithm to solve for the optimal weights. Although straightforward, its procedure is less instructive. By introducing statistical signal models, our least mean-square approach, on the other hand, provides many useful insights into the solution. For example, based on the underpinnings in [19] and [24], we first obtain that the template MV approximates the pixel true motion at the template centroid, which usually locates near the top-left corner of the target block. Because it can better compensate for the movement of the top-left region, it then follows that the optimal choice for the other MV is given by the true motion of a pixel in the bottom-right quadrant. These facts together lead naturally to unequal OBMC weights, forming a geometry-like motion partitioning [8]. The result not only refutes the simple weighted averaging to be optimal, but also justifies the use of OBMC.

Experiments based on the HM-6.0 software [2] have confirmed our theoretical predictions and the performance of this bi-prediction concept. Several variants of the algorithm, which implement adaptive template configuration with a varying number of hypotheses or extend the notion to motion merging [23], were studied to explore the performance and complexity trade-offs. The best of them, in terms of compression performance, achieves 1.7–2.0% BD-rate reductions (relative to the HM anchor [2]) at a cost of 26% and 39% increases in encoding and decoding times, respectively. Using motion merging to replace template matching for MV inference brings down the time increases to 21% and 2%, respectively, with rate reductions dropping to 0.9–1.5% as a result. While this is by no means an ideal operating point, our work demonstrates the potential of having the encoder and decoder work cooperatively to deliver better performance.

The rest of this paper is organized as follows. Section II analyzes TMP in a motion sampling framework and explains its superiority over SKIP prediction. Within the same framework, Section III formulates the combination of TMP and BMC based on OBMC as an optimization of the motion sampling structure. Optimal OBMC weights are solved using both a deterministic and a statistical approach, with the distinctions between the two solutions compared from various aspects. Section IV evaluates the performance and computational complexity of this bi-prediction method and its variants under common test conditions. Section V concludes this paper with a summary of observations and a list of future works.

II. Template Matching Prediction (TMP): A Theoretical Perspective

In this section, we study TMP from a theoretical viewpoint. Our goals are: 1) to reveal the factors that determine its prediction performance and 2) to understand its relationship to BMC and SKIP prediction. Some early results were published in our prior work [22], but a more thorough treatment of the topic is given in this paper. We adopt two signal models, [19] and [24], while doing the theoretical analysis. To proceed, we begin by reviewing these models.

A. Review of Signal Models

To analyze residuals of BMC, Tao *et al.* [19] modeled the *autocorrelation* functions of the intensity and motion fields by

$$E[I_k(\mathbf{s}_1)I_k(\mathbf{s}_2)]=\max(\sigma_I^2(1-\frac{||\mathbf{s}_1-\mathbf{s}_2||_2^2}{K}),0) \tag{1a}$$

$$E[v_x(\mathbf{s}_1)v_x(\mathbf{s}_2)]=E[v_y(\mathbf{s}_1)v_y(\mathbf{s}_2)]=\sigma_m^2\rho_m^{\|\mathbf{s}_1-\mathbf{s}_2\|_1} \tag{1b}$$

respectively, where $I_k(\mathbf{s})$ represents the intensity value of pixel $\mathbf{s}=(x(\mathbf{s}), y(\mathbf{s}))^T$ of frame k; $\mathbf{v}(\mathbf{s})=(v_x(\mathbf{s}), v_y(\mathbf{s}))^T$ denotes its true MV;[1] $\{\sigma_I^2, K\}$ and $\{\sigma_m^2, \rho_m\}$ are parameters related to their respective variances and correlation coefficients. Equations (1a) and (1b) suggest that the intensity and motion correlations between any two pixels decrease with the distance in between them.

Similarly, Zheng *et al.* [24] introduced a motion model assuming that the difference between the true MVs of any two pixels obeys the *normal* distribution

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

where α is a constant indicating the degree of motion variation in the horizontal or vertical direction, and $r(\mathbf{s}_1, \mathbf{s}_2)=||\mathbf{s}_1-\mathbf{s}_2||_2$ is the ℓ^2 distance between pixels $\mathbf{s}_1$ and $\mathbf{s}_2$. The "hat" in (2) indicates that its value will be clipped when exceeding

[1]Under the constant intensity assumption, the true motion (or, interchangeably, the true MV) $\mathbf{v}(\mathbf{s})$ of a pixel $\mathbf{s}$ in frame k satisfies $I_k(\mathbf{s}) = I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}))$, where I_{k-1} denotes the reference frame of frame k.

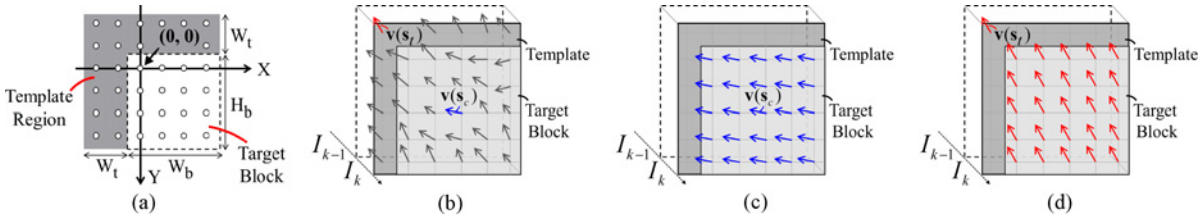

Fig. 1. (a) Coordinate system and definitions of symbols. (b) An assumed true motion field for frame k, and the reconstructed motion fields resulting from compensating a target block in frame k with (c) the least-squares-based block MV and (d) the template MV.

a maximum threshold, which, as has been shown in [5], is essential for the model to be proper. Equation (2) leads to the following *autocorrelation* function:

$$E[v_x(\mathbf{s}_1)v_x(\mathbf{s}_2)]=E[v_y(\mathbf{s}_1)v_y(\mathbf{s}_2)]=\sigma_m^2-\frac{\alpha}{2}\hat{r}^2(\mathbf{s}_1,\mathbf{s}_2) \quad (3)$$

assuming the motion field is (wide-sense) stationary and zero-mean.

With these models, a closed-form expression for the mean-sqaured prediction error, $E[d^2(\mathbf{s};\mathbf{v}(\mathbf{q}))]$, $d(\mathbf{s};\mathbf{v}(\mathbf{q})) \equiv I_k(\mathbf{s})\text{-}I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{q}))$, of pixel $\mathbf{s}$ based on the true MV of pixel $\mathbf{q}$ can be obtained. This result will be useful for analyzing various prediction schemes, as we shall see later. In [19], the derivation is done by a direct application of (1a) and (1b) in evaluating $E[(I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}))-I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{q})))^2]$, where under the constant intensity assumption, $I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}))$ has been substituted for $I_k(\mathbf{s})$. This gives

$$E[d^2(\mathbf{s};\mathbf{v}(\mathbf{q}))]=\frac{8\sigma_I^2\sigma_m^2}{K}\left(1-\rho_m^{||\mathbf{s}-\mathbf{q}||_1}\right). \quad (4)$$

Zheng *et al.* [24] take a different approach to find $E[d^2(\mathbf{s};\mathbf{v}(\mathbf{q}))]$ without requiring the use of an intensity model. They approximate the prediction error by Taylor expansion, $d(\mathbf{s};\mathbf{v}(\mathbf{q})) \approx I'^{(x)}_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{q}))(v_x(\mathbf{s})-v_x(\mathbf{q}))+I'^{(y)}_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{q}))\left(v_y(\mathbf{s})-v_y(\mathbf{q})\right)$, take expectation of the square of both sides, and assume the x, y components of $I'_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{q}))$ and $(\mathbf{v}(\mathbf{s})\text{-}\mathbf{v}(\mathbf{q}))$ are all independent of each other, to get

$$E[d^2(\mathbf{s};\mathbf{v}(\mathbf{q}))] \approx \epsilon\hat{r}^2(\mathbf{s},\mathbf{q})=\min(\epsilon||\mathbf{s}-\mathbf{q}||_2^2,\epsilon\tau^2) \quad (5)$$

where (2) has been put into use, $\epsilon=\alpha E[(I'^{(x)}_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{q})))^2+(I'^{(y)}_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{q})))^2]$ and $\tau=\sqrt{2\sigma_m^2/\alpha}$ is a clipping threshold [5].

B. Sampling the Motion Field

Based on (5), a block MV, $\mathbf{v}_b$, found from least-squares-based block matching was shown in [24] to approximate the true motion, $\mathbf{v}(\mathbf{s}_c)$, associated with the block center, $\mathbf{s}_c$, in the sense that the sum of prediction error variances over the target block is minimized when $\mathbf{v}_b$ is chosen to be $\mathbf{v}(\mathbf{s}_c)$:

$$\mathbf{s}_c=\arg\min_{\mathbf{q}}\sum_{\mathbf{s}\in\mathcal{B}}E[d^2(\mathbf{s};\mathbf{v}(\mathbf{q}))]=\left(\frac{\sum_{\mathbf{s}\in\mathcal{B}}x(\mathbf{s})}{|\mathcal{B}|},\frac{\sum_{\mathbf{s}\in\mathcal{B}}y(\mathbf{s})}{|\mathcal{B}|}\right)^T \quad (6)$$

where $\mathcal{B}$ is a set consisting of coordinates of every pixel in the block. This can be easily verified by substituting (5) in (6) and setting the derivatives with respect to the x, y components of $\mathbf{q}$ equal to zero.[2] Together these observations lead to an insightful interpretation of BMC: its operation may be viewed as a two-step process, in which block-based motion estimation acts as a motion sampler taking samples at block centers, while block-based motion compensation reconstructs the motion field by interpolating between motion samples using the nearest-neighbor rule (Fig. 1).

Following the same line of derivation with $\mathcal{B}$ replaced by $\mathcal{T}$, the pixel coordinates set for the template region, and using (5), it is straightforward to show that the template MV, $\mathbf{v}_t$, approximates the true motion, $\mathbf{v}(\mathbf{s}_t)$, associated with the template centroid, $\mathbf{s}_t=\left(\sum_{\mathbf{s}\in\mathcal{T}}x(\mathbf{s})/|\mathcal{T}|,\sum_{\mathbf{s}\in\mathcal{T}}y(\mathbf{s})/|\mathcal{T}|\right)^T$. This, in fact, holds more generally for any matching area of arbitrary shape. Repeating the same computation with (4) gives a somewhat different result, but the trend remains similar. As an illustration, Table II shows the locations of $\mathbf{s}_t$ predicted by the two models for various template configurations, with parameters W_b, H_b, and W_t defined in Fig. 2(a). From the table, both predict $\mathbf{s}_t$ to be at some point near the top-left corner of the target block. A closer look at the data reveals that it always falls in the template area, when computed based on (4), but may lie outside when (5) is in use. We also note that for fixed W_b, H_b, the resulting $\mathbf{s}_t$ departs further from the block center as the template width W_t increases; the phenomenon is common to both models.

Reviewing the above results suggests that: 1) the difference between $\mathbf{v}_t$ and $\mathbf{v}_b$ can be seen to be their sampling locations in the motion field (Fig. 1) and that 2) a change to the template configuration amounts to a variation of $\mathbf{v}_t$'s sampling location.

C. Prediction Error Surfaces of BMC and TMP

With the background developed so far, we now proceed to explore the distribution of prediction error variances over the target block $\mathcal{B}$, termed the *prediction error surface*, for BMC and TMP. To do so, the block and template MVs, $\mathbf{v}_b$ and $\mathbf{v}_t$, are modeled by $\mathbf{v}(\mathbf{s}_c)$ and $\mathbf{v}(\mathbf{s}_t)$, respectively, and substituted for $\mathbf{v}(\mathbf{q})$ in (4) or (5) to compute the error variance for every pixel $\mathbf{s}$ in $\mathcal{B}$. The results are visualized in Fig. 2 and compared with the empirical data generated by encoding 50 frames of the *BasketballDrill* sequence [2].

From Fig. 2, we see a close relationship between the shape of a prediction error surface and the sampling location of the

[2]A similar result also can be observed with (4). In the case, the optimal $\mathbf{q}$ is the one that minimizes $\sum_{\mathbf{s}\in\mathcal{B}}8\sigma_I^2\sigma_m^2(1-\rho_m^{\|\mathbf{s}-\mathbf{q}\|_1})/K$. Obviously, it cannot be found by differentiation because of the presence of the ℓ_1 norm. We thus find its coordinates by evaluating the expression $\sum_{\mathbf{s}\in\mathcal{B}}8\sigma_I^2\sigma_m^2(1-\rho_m^{\|\mathbf{s}-\mathbf{q}\|_1})/K$ for all permissible locations of $\mathbf{q}$ in quarter-pel precision.

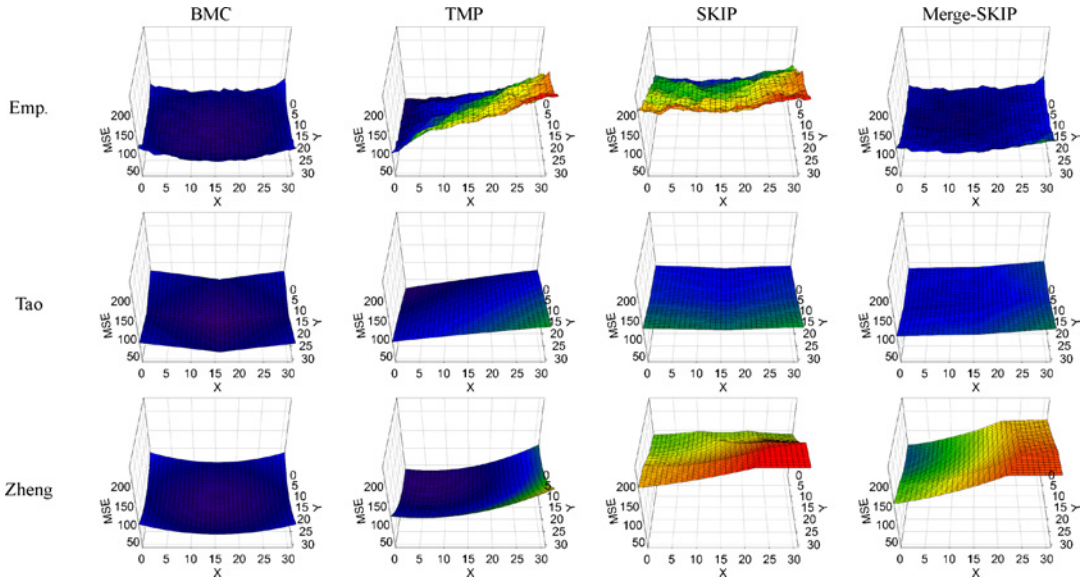

Fig. 2. Mean-squared prediction error surfaces of block $\mathcal{B}$ produced with BMC, TMP, SKIP, and Merge-SKIP. The second and third rows show the theoretical predictions made by Tao's ($\sigma_I^2\sigma_m^2/K = 29$, $\rho_m = 0.99$, $\Delta_{\text{Tao}} = 18$) and Zheng's ($\epsilon = 0.12$, $\tau = 22.6$, $\Delta_{\text{Zheng}} = 28$) models, respectively. The empirical data are based on encoding 50 frames of the *BasketballDrill* sequence [2] with QP=22 and $W_b = H_b = 32$, $W_t = 4$.

MV based on which it is computed. For instance, both models predict the error surface of BMC has a convex shape. The error variance tends to be larger at the block boundaries and smaller around the center, which is understandable if we recall that $\mathbf{v}_b$ approximates $\mathbf{v}(\mathbf{s}_c)$, the true motion associated with the block center. Following the same argument, it is intuitive to expect the residual of TMP has a large variance at the bottom-right quarter of the target block. This is because the template MV, when viewed as $\mathbf{v}(\mathbf{s}_t)$, generally has a weaker correlation to pixels' true motion there, thereby accounting for the poorer prediction. Comparing these results with their empirical counterparts confirms the accuracy of our theoretical predictions.

Further numerical evaluation indicates that TMP actually performs worse than BMC, in terms of the sum, $\sum_{\mathbf{s}\in\mathcal{B}} E[d^2(\mathbf{s};\mathbf{v}(\mathbf{q}))]$, of prediction error variances. This is also evident from (6), which says that the minimum of $\sum_{\mathbf{s}\in\mathcal{B}} E[d^2(\mathbf{s};\mathbf{v}(\mathbf{q}))]$ is reached when $\mathbf{q}$ is right at the block center. In this sense, the sampling structure of BMC is optimal. The performance gap, however, can be insignificant if the intensity and motion fields are less random or have a high spatial correlation—that is, in Tao's model, σ_I^2, σ_m^2 are small or ρ_m, K tend to be large, or in Zheng's model, α is small. Under these circumstances, the scaling factors $8\sigma_I^2\sigma_m^2/K$ and ϵ in (4) and (5) become very small, implying that the resulting sum of error variances is less sensitive to where the MV is sampled, i.e., the choice of $\mathbf{q}$. In this case, TMP is at an advantage because it need not additionally signal motion information.

D. Prediction Error Surface of SKIP Prediction

In this section, we study another decoder-side motion inference technique, which is widely known as SKIP. We start with the SKIP method in H.264/AVC [1] and establish a closed-form formula for estimating its prediction error variance over a target block. The result will then be extended to the more recently proposed Merge-SKIP in the High Efficiency Video Coding (HEVC) standard [3].

1) *SKIP in H.264/AVC:* In H.264/AVC [1], when an inter macroblock is coded in SKIP mode, its MV is inferred by computing the median of previously coded MVs in a causal neighborhood. For example, in Fig. 3, the inferred MV, $\widehat{\mathbf{v}}=(\widehat{v}_x, \widehat{v}_y)^T$, for block $\mathcal{B}$ is

$$\begin{aligned}\widehat{v}_x&=\text{Median}\{v_x(\mathbf{s}_1), v_x(\mathbf{s}_2), v_x(\mathbf{s}_3)\}\\ \widehat{v}_y&=\text{Median}\{v_y(\mathbf{s}_1), v_y(\mathbf{s}_2), v_y(\mathbf{s}_3)\}\end{aligned} \tag{7}$$

where $(v_x(\mathbf{s}_i), v_y(\mathbf{s}_i))^T$, $i = 1, 2, 3$ are MVs associated with blocks $\mathcal{B}_i$, which have been approximated by the true MVs of their respective centers $\mathbf{s}_i$. In this case, the prediction error variance at pixel $\mathbf{s}$, $\mathbf{s} \in \mathcal{B}$ is given by

$$\begin{aligned}E\left[d^2(\mathbf{s};\widehat{\mathbf{v}})\right]&=E\left[\left(I_k(\mathbf{s})-I_{k-1}(\mathbf{s}+\widehat{\mathbf{v}})\right)^2\right]\\ &=E\left[\left(I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}))-I_{k-1}(\mathbf{s}+\widehat{\mathbf{v}})\right)^2\right].\end{aligned} \tag{8}$$

Computing (8), which involves *order statistics,* is a difficult task. To circumvent the difficulties, we take a simpler approach by assuming that $(\widehat{v}_x, \widehat{v}_y)$ is equally likely to be any of the ordered pairs $(v_x(\mathbf{s}_i), v_y(\mathbf{s}_j))$, $i, j = 1, 2, 3$. Using the notation $\widehat{\mathbf{v}}_{ij} \equiv (v_x(\mathbf{s}_i), v_y(\mathbf{s}_j))$, we can then express (8) as

$$E\left[d^2(\mathbf{s};\widehat{\mathbf{v}})\right]=\frac{1}{9}\sum_{i=1}^{3}\sum_{j=1}^{3}E\left[\left(I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}))-I_{k-1}(\mathbf{s}+\widehat{\mathbf{v}}_{ij})\right)^2\right] \tag{9}$$

which can readily be evaluated by incorporating Tao's model [19]. The result is

$$E[d^2(\mathbf{s};\widehat{\mathbf{v}})]=\frac{1}{3}\sum_{i=1}^{3}E[d^2(\mathbf{s};\mathbf{v}(\mathbf{s}_i))]=\frac{8\sigma_I^2\sigma_m^2}{3K}\sum_{i=1}^{3}(1-\rho_m^{\|\mathbf{s}-\mathbf{s}_i\|_1}) \tag{10}$$

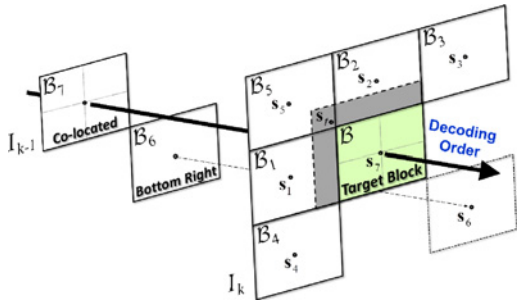

Fig. 3. Geometry of MVs used for SKIP, Merge-SKIP, TMP.

and the result for Zheng's model can be shown to be

$$E\left[d^2(\mathbf{s};\widehat{\mathbf{v}})\right] \approx \frac{1}{3}\sum_{i=1}^{3} E[d^2(\mathbf{s};\mathbf{v}(\mathbf{s}_i))] = \frac{\epsilon}{3}\sum_{i=1}^{3}\hat{r}^2(\mathbf{s},\mathbf{s}_i). \quad (11)$$

It is interesting to see that both (10) and (11) are merely a weighted sum of the mean-squared prediction errors when $\mathbf{v}(\mathbf{s}_i), i = 1, 2, 3$ are applied individually to the motion compensation of pixel $\mathbf{s}$, which is a direct consequence of assuming that all possible outcomes of $(\widehat{v}_x, \widehat{v}_y)$ are equally probable. The validity of this assumption is justified by the data in Fig. 2, where the error surfaces predicted by (10) and (11) resemble closely the empirical one. As expected, with the introduction of $\mathbf{v}(\mathbf{s}_3)$, the error variance becomes smaller in the upper half of the target block.

2) *Merge-SKIP in HEVC:* While taking the median of neighboring MVs helps to ensure the motion smoothness, the resulting MV is somehow artificial and may be unlike any of those found in the neighborhood. To avoid this problem, a Merge-SKIP method was recently proposed in the latest HEVC standard [3]. The idea is to reuse the MV(s) from one of the neighboring prediction blocks (prediction units) by sending extra bits to signal the choice. For example, in Fig. 3**,** the target block can choose any MV from blocks $\mathcal{B}_i$, $i = 1, 2, ..., 7$ for motion compensation. To save bits, those associated with $\mathcal{B}_i$, $i = 1, 2, 3, 4, 6$ are the first candidates for reuse, and only if any of the MVs from $\mathcal{B}_i$, $i = 1, 2, 3, 4$ (respectively, $\mathcal{B}_6$) is not available will that of $\mathcal{B}_5$ (respectively, $\mathcal{B}_7$) be considered.

Following the notions developed earlier, it is easy to show that in this case, the mean-squared prediction error for a pixel $\mathbf{s}$ in the target block $\mathcal{B}$ can be modeled as a weighted sum $\sum w_i E[d^2(\mathbf{s};\mathbf{v}(\mathbf{s}_i))]$ of the error squares produced by applying the true motion $\mathbf{v}(\mathbf{s}_i)$ of pixels $\mathbf{s}_i, i = 1, 2, ..., 7$ for motion compensation. In other words, it has exactly the same form as (10) and (11), except that the weighting factors w_i's now represent the probabilities of choosing the respective MV candidates, which are generally nonuniform. With some degree of approximation, we have regarded the two temporally co-located MVs from $\mathcal{B}_6$ and $\mathcal{B}_7$ as $\mathbf{v}(\mathbf{s}_6)$ and $\mathbf{v}(\mathbf{s}_7)$, respectively, both, as shown in the figure, are pixel true motion corresponding to the current frame rather than the reference frame. Clearly, this requires that their motion fields be very similar. Although it is not always the case, we shall assume so to proceed with the analysis.

With the above results, the right most column of Fig. 2 shows the error surfaces of the Merge-SKIP method. The theoretical ones have been generated by setting $(w_1, w_2, ..., w_7) = (0.52, 0.22, 0.08, 0.02, 0.05, 0.03, 0.09)$, which are the relative frequencies that different candidates are selected in the empirical experiment. In terms of waveforms, we see that the theoretical predictions match the empirical result; the higher weighting toward block $\mathcal{B}_1$ explains why the surfaces incline to the left.

From the relative frequencies, one fact that may seem paradoxical is the rare use of the temporally co-located MV from $\mathcal{B}_7$. Ideally, it should be the most selected one, provided that its approximation to the true motion of the target block's center is sufficiently accurate. This seeming paradox has to do with the priority-based candidate selection. As stated at the outset, the MV of $\mathcal{B}_7$ is only selectable when that of $\mathcal{B}_6$ is not available. In a side experiment where this restriction is removed, we do observe a dramatic increase in its use.

TABLE I
MEAN-SQUARED PREDICTION ERRORS

	BasketballDrill QP22				*Johnny* QP22			
(W_b=H_b=32)	Emp.	Tao	Zheng	Bits	Emp.	Tao	Zheng	Bits
BMC	49	49	49	7.5	5	5	5	6.3
SKIP	152	98	182	–	9	5	6	–
Merge-SKIP	68	90	163	1.8	5	5	6	2.1
TMP (W_t=4)	114	77	72	–	7	5	5	–
TMP (W_t=8)	117	80	80	–	6	5	5	–
TMP (W_t=16)	130	87	101	–	6	5	5	–

BasketballDrill—$(\sigma_I^2\sigma_m^2/K = 29,\ \rho_m = 0.99,\ \Delta_{\text{Tao}} = 18)$, $(\epsilon = 0.12,\ \tau = 22.6,\ \Delta_{\text{Zheng}} = 28)$, *Johnny*—$(\sigma_I^2\sigma_m^2/K = 0.2,\ \rho_m = 0.99,\ \Delta_{\text{Tao}} = 5)$, $(\epsilon = 0.01,\ \tau = 22.6,\ \Delta_{\text{Zheng}} = 5)$

E. Prediction Performance Comparison

To see how BMC, TMP, SKIP, and Merge-SKIP perform relative to each other, Table I compares their mean-squared prediction errors both empirically and theoretically. The empirical results are based on encoding 50 frames of two standard sequences, *BasketballDrill* and *Johnny* [2]. The former has complex motion, while the latter is of video-conferencing type and has less detail.

In comparing their empirical performance, we note that the computation of prediction error would vary with the choice of target blocks and their reference frames. To ensure that the same target blocks are used and the reference frames from which they are predicted are identical, the test sequences are first coded using the HM-6.0 software [2]. Then, from the reconstructed images, a predictor is created for every source frame by performing motion compensation on a block-by-block basis. In this way, every source block is a target block and different methods of forming its predictor are performed on exactly the same references.

Accordingly, to evaluate the theoretical models, we obtain the model parameters $(\sigma_I^2\sigma_m^2/K, \rho_m, \Delta_{\text{Tao}})$[3] and $(\epsilon, \tau, \Delta_{\text{Zheng}})$ for each sequence by fitting the theoretical error surface of BMC to the empirical one. That is, we choose these parameters so that the misfit, measured by the squared errors, between the theoretical and empirical surfaces is minimized. We then use

[3]For a reason that will become clear in §III-E1, all prediction error models must be amended by including a positive offset Δ. For example, (5) will become $\epsilon\hat{r}^2(\mathbf{s},\mathbf{q}) + \Delta$.

these same parameters in the models for the other prediction schemes. Admittedly, there are better and more practical ways for estimating parameters, but we will not pursue them further since obtaining accurate estimates is not our main focus here. All model parameters throughout the paper have been generated following the same procedure, unless otherwise stated.

With reference to the table, we see that: 1) in the *Johnny* sequence, all four schemes perform about the same, and 2) in the *BasketballDrill* sequence, their relationships, in terms of the magnitude of mean-squared error, are as follows.

1) (Theoretically) SKIP > Merge-SKIP > TMP (W_t=4) > BMC.
2) (Empirically) SKIP > TMP (W_t=4) > Merge-SKIP > BMC.

Most of these observations can have a simple explanation if we examine the motion sampling structure behind these prediction schemes. For instance, the first corroborates an earlier finding that the motion sampling structure will be less critical to prediction performance if the intensity and motion fields are less random or have a high spatial correlation. BMC performs the best because the MV used approximates the true motion of the block center. Similarly, TMP is superior to SKIP since, from Fig. 3, $\mathbf{s}_t$ is normally closer to the block center than any of the $\mathbf{s}_i$, $i = 1, 2, 3$. Both (4) and (5) suggest that the further the motion sampling point (the location of $\mathbf{q}$) is away from the block center, the larger the sum, $\sum_{\mathbf{s}\in\mathcal{B}} E[d^2(\mathbf{s}; \mathbf{v}(\mathbf{q}))]$, of error variances would be. The same argument also explains why increasing W_t (which causes $\mathbf{s}_t$ to deviate more from the block center) usually has a negative performance impact on TMP. Interestingly, these results together justify the TMP-SKIP [10] and the typically small-sized template width.

One exception, however, occurs when TMP is compared with Merge-SKIP. The latter performs much better in practice than it is expected theoretically. This may be attributed to the fact that the intensity and motion fields are likely nonstationary and under which case, the flexibility of Merge-SKIP in choosing MV candidate makes it possible to become easily adjusted to the varying statistics. This benefit, of course, comes at the expense of a higher rate cost. For comparison, provided in the column "Bits" of Table I is the average number of bits per prediction block (computed from the empirical experiment) that is required for different schemes to represent their motion parameters, e.g., MV in the case of BMC or merge index for Merge-SKIP. Here uni-prediction is assumed, and the cost for signaling the prediction mode is neglected as it depends highly on the syntax format.

III. Bi-prediction Combining TMP and BMC

This section introduces a novel bi-prediction scheme, in which the predictor is computed as a weighted average of two reference blocks, one pointed to by a template MV, $\mathbf{v}_t$—obtained through performing an identical template matching operation at both the encoder and decoder—and the other by a block MV, $\mathbf{v}_b$—signaled via regular MV coding. We have shown that the former can better compensate for the movement of the top-left area of a prediction block. The latter is thus aimed at reducing further the prediction residual in the remaining area. The spatially-varying contribution of these MVs to motion compensating a prediction block motivates the use of OBMC for combining their reference blocks. This leads to a pixel-adaptive bi-prediction method (see Fig. 4) with a motion cost just as that for uni-prediction.

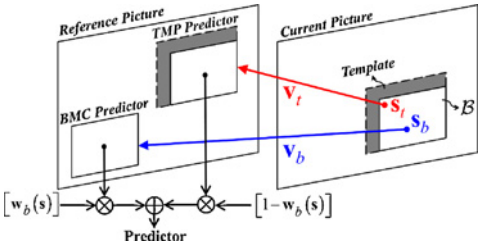

Fig. 4. Bi-prediction combining TMP and BMC: $\mathbf{v}_t$ is the template MV inferred implicitly, $\mathbf{v}_b$ is the block MV coded in the bit-stream, and $w_b(\widetilde{\mathbf{s}})$ and $1 - w_b(\widetilde{\mathbf{s}})$ specify the weighting coefficients (namely, the OBMC window functions) for prediction with $\mathbf{v}_b$ and $\mathbf{v}_t$, respectively.

A. Basics of OBMC

The notion of OBMC is to provide an estimate of a pixel's intensity value $I_k(\mathbf{s})$ based on linearly combining multiple motion-compensated signals $\sum_{i=1}^{L} w_i I_{k-1}(\mathbf{s}+\mathbf{v}_i)$, where $\{\mathbf{v}_i\}_{i=1}^{L}$ is a MV set composed normally of the MV of the prediction block where pixel $\mathbf{s}$ belongs and those of its neighboring blocks. As an example, if $\mathbf{s}$ was a pixel in the target block $\mathcal{B}$ shown in Fig. 3, then one possible composition of $\{\mathbf{v}_i\}$ may include $\mathbf{v}(\mathbf{s}_i), i = 1, ..., 7$. From an estimation-theoretic perspective, these MVs are probable hypotheses for its true motion $\mathbf{v}(\mathbf{s})$, with w_i's indicating their likelihood. Usually, the values of w_i's are estimated by minimizing the squared prediction error $(I_k(\mathbf{s}) - \sum_{i=1}^{L} w_i I_{k-1}(\mathbf{s}+\mathbf{v}_i))^2$ subject to $\sum_{i=1}^{L} w_i = 1$ in either a statistical or a deterministic sense, and under fairly mild conditions, they were shown to be a function of pixel position within a target block [15], [19], [24]. This suggests that the contribution of each MV to estimating pixel values across the target block is spatially varying and that prediction with OBMC is pixel adaptive. Hereafter, we will use OBMC weights and window functions interchangeably when referring to w_i's. The latter is used when the stress is on viewing each w_i as a separate function (of relative pixel position) characterizing the contribution of a MV.

B. Problem Formulation

The proposed bi-prediction method is a particular application of OBMC, which distinguishes from the conventional approach in using the template MV as a substitute for MVs from neighboring blocks. Since the template MV has to be produced identically at both the encoder and decoder, its values cannot be specified discretionarily. As a result, how to minimize the prediction residual by a suitable choice of the block MV and OBMC weights (refer to Fig. 4) is central to this application.

To define the problem more specifically, we assume that this bi-prediction method is just one of the options the encoder can use for predicting a target block and that, for the sake of illustrative convenience, all prediction blocks have the same

size of $W_b \times H_b$. With these in mind, our objective is to

$$\begin{aligned}\underset{\mathbf{v}_{b,i}, w_b(\widetilde{\mathbf{s}}), w_t(\widetilde{\mathbf{s}})}{\text{minimize}}\ \xi = &\sum_{i\in\mathcal{I}}\sum_{\mathbf{s}\in\mathcal{B}_i}(I_k(\mathbf{s}) - w_t\left(\widetilde{\mathbf{s}}\right) I_{k-1}\left(\mathbf{s}+\mathbf{v}_{t,i}\right)\\ &- w_b\left(\widetilde{\mathbf{s}}\right) I_{k-1}(\mathbf{s}+\mathbf{v}_{b,i}))^2\\ \text{subject to}\quad & w_t\left(\widetilde{\mathbf{s}}\right) + w_b\left(\widetilde{\mathbf{s}}\right) = 1\end{aligned} \tag{12}$$

where $\mathbf{s}=(x(\mathbf{s}), y(\mathbf{s}))^T$, as defined previously, refers to a pixel's absolute position relative to the picture origin, with

$$\widetilde{\mathbf{s}}=(x(\widetilde{\mathbf{s}}), y(\widetilde{\mathbf{s}}))^T=(x(\mathbf{s}) \text{ modulo } W_b,\ y(\mathbf{s}) \text{ modulo } H_b)^T \tag{13}$$

indicating its relative position within a prediction block; $\mathcal{B}_i, i \in \mathcal{I}$ are labels of prediction blocks (in a picture) that adopt this bi-prediction method, each further serving as a set collecting absolute coordinates for all pixels in one such block; and $w_t\left(\widetilde{\mathbf{s}}\right)$ and $w_b\left(\widetilde{\mathbf{s}}\right)$ are OBMC weights associated with the template and block MVs, respectively. Like the weighting factors for regular OBMC, both $w_t\left(\widetilde{\mathbf{s}}\right)$ and $w_b\left(\widetilde{\mathbf{s}}\right)$ are a function of (relative) pixel position within a target block. By substituting $1 - w_b\left(\widetilde{\mathbf{s}}\right)$ for $w_t\left(\widetilde{\mathbf{s}}\right)$ in the objective function, we further arrive at an unconstrained formulation:

$$\begin{aligned}\underset{\mathbf{v}_{b,i}, w_b(\widetilde{\mathbf{s}})}{\text{minimize}}\ \xi = &\sum_{i\in\mathcal{I}}\sum_{\mathbf{s}\in\mathcal{B}_i}(I_k(\mathbf{s}) - \left(1 - w_b\left(\widetilde{\mathbf{s}}\right)\right) I_{k-1}\left(\mathbf{s}+\mathbf{v}_{t,i}\right)\\ &- w_b\left(\widetilde{\mathbf{s}}\right) I_{k-1}(\mathbf{s}+\mathbf{v}_{b,i}))^2\end{aligned} \tag{14}$$

for which the unknowns to be found are the block MVs, $\mathbf{v}_{b,i}$, $i \in \mathcal{I}$, and the corresponding OBMC weights, $w_b(\widetilde{\mathbf{s}})$'s, for all the values taken by $\widetilde{\mathbf{s}}$—namely, all pairs of $(m, n)^T$, $m = 0, 1, ..., W_b - 1$ and $n = 0, 1, ..., H_b - 1$.

C. Iterative Least-Squares (LS) Solution

This section introduces an iterative algorithm for solving the problem in (14). Its procedure involves finding a block MV for all the prediction blocks $\mathcal{B}_i, i \in \mathcal{I}$ in raster-scan order using the current best estimates $w_b^{(k)}(\widetilde{\mathbf{s}})$'s of OBMC weights and the associated template MVs $\mathbf{v}_{t,i}^{(k)}$'s. Then, the resulting block MVs $\mathbf{v}_{b,i}^{(k)}$'s, along with the template MVs $\mathbf{v}_{t,i}^{(k)}$'s, will be utilized to improve the estimates $w_b^{(k)}(\widetilde{\mathbf{s}})$'s to $w_b^{(k+1)}(\widetilde{\mathbf{s}})$'s. These steps will be repeated until the change in the value of ξ between successive iterations is below a threshold. The following elaborates on each of these steps.

1) ***Estimating Block MVs:*** Assume that we are at the kth iteration and the current estimates of OBMC weights are $w_b^{(k)}(\widetilde{\mathbf{s}})$'s. Using these estimates in (14), we minimize ξ by finding, for each prediction block $\mathcal{B}_i, i \in \mathcal{I}$, a MV $\mathbf{v}_{b,i}^{(k)}$ that minimizes its OBMC prediction error

$$\begin{aligned}\mathbf{v}_{b,i}^{(k)} = \arg\min_{\mathbf{v}_{b,i}} &\sum_{\mathbf{s}\in\mathcal{B}_i}(I_k(\mathbf{s}) - (1 - w_b^{(k)}\left(\widetilde{\mathbf{s}}\right))I_{k-1}(\mathbf{s}+\mathbf{v}_{t,i}^{(k)}\\ &- w_b^{(k)}\left(\widetilde{\mathbf{s}}\right) I_{k-1}(\mathbf{s}+\mathbf{v}_{b,i}))^2\end{aligned} \tag{15}$$

where the template MV $\mathbf{v}_{t,i}^{(k)}$ is obtained, right before the search of the block MV $\mathbf{v}_{b,i}^{(k)}$, via template matching.

2) ***Adapting OBMC Weights:*** To obtain new estimates of OBMC weights, we substitute the resulting $\mathbf{v}_{b,i}^{(k)}$'s and $\mathbf{v}_{t,i}^{(k)}$'s for $\mathbf{v}_{b,i}$'s and $\mathbf{v}_{t,i}$'s in (14), respectively, and solve for $w_b(\widetilde{\mathbf{s}})$'s for all the values taken by $\widetilde{\mathbf{s}}$. In doing so, it is convenient to consider an alternative expression for ξ as follows:

$$\begin{aligned}\xi = \sum_{m=0}^{W_b-1}\sum_{n=0}^{H_b-1}\sum_{i\in\mathcal{I}}&(I_k(\mathbf{s}_{mn,i}) - \left(1 - w_b(\widetilde{\mathbf{s}_{mn,i}})\right) I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{t,i})\\ &- w_b(\widetilde{\mathbf{s}_{mn,i}}) I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{b,i}))^2\end{aligned}$$

which is obtained from (14) by first summing over the $\mathbf{s}$'s (which we denote as $\mathbf{s}_{mn,i}$'s) with the same relative pixel position $(m, n)^T$ in their respective blocks $\mathcal{B}_i$, $i \in \mathcal{I}$ and then adding up the results for all pairs of $(m, n)^T$, $m = 0, 1, ..., W_b - 1$ and $n = 0, 1, ..., H_b - 1$. In this form, ξ is a sum of $W_b \times H_b$ functions, the value of each represents the sum of OBMC prediction errors at a certain relative pixel position and is governed solely by the variable $w_b\left(\widetilde{\mathbf{s}_{mn,i}}\right)$. Note that in the present case, $\mathbf{v}_{t,i}$ and $\mathbf{v}_{b,i}$ assume the values of $\mathbf{v}_{t,i}^{(k)}$ and $\mathbf{v}_{b,i}^{(k)}$, respectively. Furthermore, according to (13), the coordinates of $\widetilde{\mathbf{s}_{mn,i}}$ will be fixed at $(m, n)^T$ regardless of what the value of i is, suggesting that $w_b(\widetilde{\mathbf{s}_{mn,i}})$ is a distinct variable for each pair of (m, n). It then follows that the minimization of ξ can be achieved through minimizing each of these functions separately;[4] that is,

$$\begin{aligned}\underset{w_b(\widetilde{\mathbf{s}_{mn,i}})}{\text{minimize}} \sum_{i\in\mathcal{I}}&(I_k(\mathbf{s}_{mn,i}) - (1 - w_b(\widetilde{\mathbf{s}_{mn,i}}))I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{t,i}^{(k)})\\ &- w_b(\widetilde{\mathbf{s}_{mn,i}}) I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{b,i}^{(k)}))^2\end{aligned} \tag{16}$$

for every pair of $(m, n)^T$. A little algebra gives (17), shown at the bottom of the page, which forms a new estimate $w_b^{(k+1)}(\widetilde{\mathbf{s}})$ of the OBMC weight $w_b(\widetilde{\mathbf{s}})$ at $\widetilde{\mathbf{s}}=(m, n)^T$.

In general, the ξ in (14) is a rather complicated function of $\mathbf{v}_{b,i}$'s, $\mathbf{v}_{t,i}$'s and $w_b(\widetilde{\mathbf{s}})$'s, in which case, there is hardly any guarantee that the above algorithm will always converge. In practice, however, its convergence to a possibly local minimum is found to be rapid (usually between 5 and 10 iterations). This might be explained by an observation that our initializing $w_b(\widetilde{\mathbf{s}})$'s to 1/2 often finds a set of block MVs, $\mathbf{v}_{b,i}$'s, that are close to their optimal values. We will explain the phenomenon from a theoretical viewpoint in the next section.

D. Least Mean-Square (LMS) Solution

The above iterative algorithm, although straightforward, is less instructive. We do not know what mechanisms cause the result, nor can we justify it. As an alternative, this section introduces a statistical approach for determining the block MVs, $\mathbf{v}_{b,i}, i \in \mathcal{I}$, and the OBMC weights, $w_b(\widetilde{\mathbf{s}})$'s. It has the advantages of more clearly revealing the essence of the proposed bi-prediction scheme and providing many useful insights into its design.

[4]The solution that minimizes a function $f(x_1, x_2)$ of the form $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$ can be found by minimizing $f_1(x_1)$ and $f_2(x_2)$ separately.

$$w_b\left(\widetilde{\mathbf{s}_{mn,i}}\right) = \frac{\sum_{i\in\mathcal{I}}\left(I_k\left(\mathbf{s}_{mn,i}\right) - I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{t,i}^{(k)})\right)\left(I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{b,i}^{(k)}) - I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{t,i}^{(k)})\right)}{\sum_{i\in\mathcal{I}}\left(I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{b,i}^{(k)}) - I_{k-1}(\mathbf{s}_{mn,i}+\mathbf{v}_{t,i}^{(k)})\right)^2} \tag{17}$$

In order to bring into use the signal models given in §II, we transform the problem of minimizing ξ into that of minimizing its expected value $E[\xi]$. This produces

$$\underset{w_b(\widetilde{\mathbf{s}}),\mathbf{s}_{b,i}}{\text{minimize}} \sum_{i\in\mathcal{I}}\sum_{\mathbf{s}\in\mathcal{B}_i} E[(I_k(\mathbf{s}) - \left(1-w_b\left(\widetilde{\mathbf{s}}\right)\right) I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_{t,i})) - w_b\left(\widetilde{\mathbf{s}}\right) I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_{b,i})))^2] \tag{18}$$

where, to allow for tractable calculations, we have tacitly substituted $\mathbf{v}(\mathbf{s}_{t,i})$ for $\mathbf{v}_{t,i}$ and $\mathbf{v}(\mathbf{s}_{b,i})$ for $\mathbf{v}_{b,i}$ [see (14)], with $\mathbf{v}(\mathbf{s}_{t,i})$ denoting the true MV at the template centroid $\mathbf{s}_{t,i}$ next to block $\mathcal{B}_i$, which has proved a good approximation to the template MV $\mathbf{v}_{t,i}$, and $\mathbf{v}(\mathbf{s}_{b,i})$ representing the true MV at an unknown position $\mathbf{s}_{b,i}$ in $\mathcal{B}_i$, which we shall use to model the block MV $\mathbf{v}_{b,i}$ to be estimated–that is, we regard $\mathbf{v}_{b,i}$ as $\mathbf{v}(\mathbf{s}_{b,i})$ and consider the determination of its values to be a problem of deciding its sampling location $\mathbf{s}_{b,i}$ in the motion field. With reference to Fig. 1, the problem in (18) can be understood as follows. Given that every block $\mathcal{B}_i, i \in \mathcal{I}$ is to be predicted using OBMC based on two MVs, one of which defaulting to the true MV $\mathbf{v}(\mathbf{s}_{t,i})$, we wish to find the other MV $\mathbf{v}(\mathbf{s}_{b,i})$ by sampling the motion field at some point $\mathbf{s}_{b,i}$ in block $\mathcal{B}_i$ and to determine a set of OBMC weights $w_b\left(\widetilde{\mathbf{s}}\right)$'s, so that the sum of mean-squared prediction errors evaluated over all $\mathcal{B}_i, i \in \mathcal{I}$ will be minimized. Here the locations of $\mathbf{s}_{t,i}, i \in \mathcal{I}$ (which can be obtained according to how the template region is defined for each $\mathcal{B}_i$) are assumed known.

To find the solution to (18), it is convenient to simplify the problem by assuming that both the intensity and motion fields are stationary and that a uniformly sized and shaped template region is defined for all $\mathcal{B}_i, i \in \mathcal{I}$. We thus only have to determine the $w_b\left(\widetilde{\mathbf{s}}\right)$'s and $\mathbf{s}_{b,i}$ for one specific block; the result will extend automatically to the other blocks by stationarity. In this case, we may just as well eliminate the block index i and the summation over $\mathcal{I}$ in (18), arriving at the simpler problem of minimizing the sum of prediction error variances for one single block:

$$\underset{w_b(\widetilde{\mathbf{s}}),\mathbf{s}_b}{\text{minimize}} \sum_{\mathbf{s}\in\mathcal{B}} E[(I_k(\mathbf{s}) - \left(1-w_b\left(\widetilde{\mathbf{s}}\right)\right) I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_t)) - w_b\left(\widetilde{\mathbf{s}}\right) I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_b)))^2]. \tag{19}$$

To proceed further, we break up the joint optimization of $w_b\left(\widetilde{\mathbf{s}}\right)$'s and $\mathbf{s}_b$ into two subproblems:

1) ***Fixing*** $\mathbf{s}_b$***, Determine the*** $w_b\left(\widetilde{\mathbf{s}}\right)$*:* Observe that every term $E[\cdot]$ in the sum above is a function of two variables, the location of $\mathbf{s}_b$ and one of the OBMC weights $w_b\left(\widetilde{\mathbf{s}}\right)$'s to be determined. The summation adds together a total of $W_b \times H_b$ such functions when $\mathbf{s}$ ranges over all pixels in $\mathcal{B}$. Since the indices $\mathbf{s}$'s are distinct (so are the $\widetilde{\mathbf{s}}$'s), there is no duplicate $w_b\left(\widetilde{\mathbf{s}}\right)$'s involved in these functions. Thus, fixing $\mathbf{s}_b$, we can find the $w_b\left(\widetilde{\mathbf{s}}\right)$'s that minimize (19), through minimizing each of them separately. Setting their derivatives with respect to the corresponding $w_b\left(\widetilde{\mathbf{s}}\right)$ to zero yields, for all $\mathbf{s} \in \mathcal{B}$ (and thus all the values of $\widetilde{\mathbf{s}}$),

$$w_b\left(\widetilde{\mathbf{s}}\right) = \frac{E[(I_k(\mathbf{s}) - I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_t)))(I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_b)) - I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_t)))]}{E[(I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_b)) - I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{s}_t)))^2]}$$

$$= \frac{E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))(d(\mathbf{s};\mathbf{v}(\mathbf{s}_t)) - d(\mathbf{s};\mathbf{v}(\mathbf{s}_b)))]}{E[(d(\mathbf{s};\mathbf{v}(\mathbf{s}_t)) - d(\mathbf{s};\mathbf{v}(\mathbf{s}_b)))^2]} \tag{20}$$

where for brevity we have made the substitution $I_k(\mathbf{s})$-$I_{k-1}(\mathbf{s}+\mathbf{v}(\mathbf{q}))=d(\mathbf{s};\mathbf{v}(\mathbf{q}))$, $\mathbf{q}=\mathbf{s}_t$ or $\mathbf{s}_b$. Equation (20) gives the best OBMC weights $w_b\left(\widetilde{\mathbf{s}}\right)$'s for a specific choice of $\mathbf{s}_b$.

2) ***Find the Optimal*** $\mathbf{s}_b$ ***that Yields the Global Minimum****:* Substituting (20) into (19) establishes (21) , shown at the bottom of the page which provides a means to find the optimal $\mathbf{s}_b$ (denoted hereafter by $\mathbf{s}_b^*$) that yields the global minimum.

Now we evaluate the various expectations, such as $E\left[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))^2\right]$, $E\left[d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))^2\right]$ and $E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))\ d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))]$, involved in (21). To this end, the signal models in §II-A are applied. The results of $E\left[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))^2\right]$ and $E\left[d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))^2\right]$ are immediate from (4) or (5), but it takes some work to compute $E\left[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))\right]$ as

$$\begin{aligned} &E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))] \\ &= \frac{4\sigma_I^2\sigma_m^2}{K}\left(1-\rho_m^{||\mathbf{s}-\mathbf{s}_t||_1}+1-\rho_m^{||\mathbf{s}-\mathbf{s}_b||_1}-1+\rho_m^{||\mathbf{s}_t-\mathbf{s}_b||_1}\right) \end{aligned} \tag{22}$$

with Tao's model, and as

$$E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))] = \frac{1}{2}\epsilon(\widehat{r}^2(\mathbf{s},\mathbf{s}_t)+\widehat{r}^2(\mathbf{s},\mathbf{s}_b)-\widehat{r}^2(\mathbf{s}_t,\mathbf{s}_b)) \tag{23}$$

with Zhang's model. Using these results in (21) yields an expression that allows us to find $\mathbf{s}_b^*$ numerically.

As an example, Fig. 5 plots the sum of prediction error variances over the target block $\mathcal{B}$ as a function of $\mathbf{s}_b$ (in quarter-pel precision), with the origin (0, 0) located directly at the top-left pixel of $\mathcal{B}$. We see that the sum becomes smaller when $\mathbf{s}_b$ sits in the bottom-right quarter. This is not surprising because, as was noted before, the template MV is less efficient for motion compensating pixels in the bottom-right area. It is natural to expect the block MV to be so sampled as to compensate for its inefficiency. The fact that $\mathbf{s}_b^*$ is not at the block center (see Table II for the numerical results of $\mathbf{s}_b^*$'s computed according to the $\mathbf{s}_t$'s given in the same Table) also proves the suboptimality of estimating $\{\mathbf{v}_{b,i}\}_{i\in\mathcal{I}}$ by simply minimizing the conventional block matching error.

Once we know the optimal location for $\mathbf{s}_b$, the corresponding window function $w_b(\widetilde{\mathbf{s}})$ (and hence $w_t(\widetilde{\mathbf{s}})=1 - w_b(\widetilde{\mathbf{s}})$) is immediately obvious by (20). Shown in Fig. 6(a) and (b) are the results computed using different models. Both suggest that the template MV should exert a greater influence on pixels in the top-left quarter, while the block MV should affect more heavily the others, particularly those in the bottom-right quarter. In some sense, this is equivalent to performing a geometry-like motion partitioning [12]. From Fig. 6(c), the same observation also holds for the LS solution, obtained

$$\underset{\mathbf{s}_b}{\text{minimize}} \sum_{\mathbf{s}\in\mathcal{B}} \left(E\left[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))^2\right] - \frac{(E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))(d(\mathbf{s};\mathbf{v}(\mathbf{s}_t)) - d(\mathbf{s};\mathbf{v}(\mathbf{s}_b)))])^2}{E[(d(\mathbf{s};\mathbf{v}(\mathbf{s}_t)) - d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))^2]} \right) \tag{21}$$

TABLE II
SAMPLING LOCATIONS OF $\mathbf{s}_t$ AND $\mathbf{s}_b^*$

		$W_t = 4$		$W_t = 8$	
W_b	H_b	Tao's $\mathbf{s}_t$	Zheng's $\mathbf{s}_t$	Tao's $\mathbf{s}_t$	Zheng's $\mathbf{s}_t$
16	16	(−1, −1)	(2, 2)	(−2, −2)	(0.25, 0.25)
32	32	(−1, −1)	(6, 6)	(−2, −2)	(4.25, 4.25)
16	32	(−2, 4)	(0.5, 8.5)	(−3, 0)	(−1, 7)
32	64	(−2, 9)	(3.25, 19.25)	(−2, −1)	(1.75, 17.75)
W_b	H_b	Tao's $\mathbf{s}_b^*$	Zheng's $\mathbf{s}_b^*$	Tao's $\mathbf{s}_b^*$	Zheng's $\mathbf{s}_b^*$
16	16	(10, 10)	(9.5, 9.5)	(10, 10)	(9, 9)
32	32	(19, 19)	(20, 20)	(19, 19)	(19.5, 19.5)
16	32	(9, 22)	(8.5, 22)	(9, 21)	(8.5, 21.5)
32	64	(18, 42)	(17.5, 45.5)	(17, 40)	(17.5, 45)

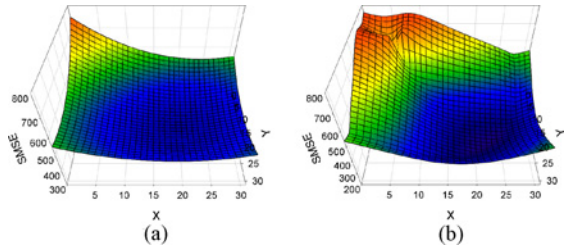

Fig. 5. Error surfaces showing how the sum of prediction error variances over the target block varies with $\mathbf{s}_b$. (a) Tao's model. (b) Zheng's model.

by carrying out the procedure in §III-C on the Class B source video sequences (see §IV) and applying uniformly the proposed scheme to every fixed-size prediction block.

In reality, knowing the location of $\mathbf{s}_b^*$ is of limited value since its true motion can be difficult to acquire. According to (20), there however exists a one-to-one correspondence between the location of $\mathbf{s}_b$ and the resulting window function $w_b(\widetilde{\mathbf{s}})$, assuming that $\mathbf{s}_t$ is given (as is the case currently). In other words, if we have the optimal window function (denoted by $w_b^*(\widetilde{\mathbf{s}})$) computed based on $\mathbf{s}_b^*$, we can find as follows a MV that approximates its true motion:

$$\mathbf{v}_{b,i}^* = \arg\min_{\mathbf{v}_{b,i}} \sum_{\mathbf{s}\in\mathcal{B}_i} (I_k(\mathbf{s}) - (1-w_b^*(\widetilde{\mathbf{s}}))I_{k-1}(\mathbf{s}+\mathbf{v}_{t,i}) - w_b^*(\widetilde{\mathbf{s}})I_{k-1}(\mathbf{s}+\mathbf{v}_{b,i}))^2. \quad (24)$$

$\forall i \in \mathcal{I}$. Recall that the expected value of the sum above is minimized when $\mathbf{v}_{t,i}$ is assumed implicitly to be equal to $\mathbf{v}(\mathbf{s}_t)$ and $\mathbf{v}_{b,i}$ is set equal to $\mathbf{v}(\mathbf{s}_b^*)$. Here we consider $\mathbf{v}_{b,i}^*$ only an approximation of $\mathbf{v}(\mathbf{s}_b^*)$ mainly because the expected value is now replaced by its instantaneous value.

We conclude this section with an interesting numerical accident obtained by fixing $w_b(\widetilde{\mathbf{s}})$'s in (19) at 1/2. Clearly, these OBMC weights are not optimal, but the resulting $\mathbf{s}_b$ is found in most cases either identical or very close to its theoretical values. This implies that in (24), if we let $w_b^*(\widetilde{\mathbf{s}})$=1/2, the resulting block MVs would approximate their optimal values, which corroborates an earlier finding that our iterative LS algorithm often shows very fast convergence when it is started with the initial value of $w_b(\widetilde{\mathbf{s}})$=1/2 [see (15)].

E. Analyses of the LS and LMS Solutions

In this section, we will examine in greater detail the window functions, $w_b(\widetilde{\mathbf{s}})$ and $w_t(\widetilde{\mathbf{s}})$, of the LS and LMS solutions.

1) *Window Function Comparison:* Fig. 7(a) displays the cross sections of $w_t(\widetilde{\mathbf{s}})$'s in Fig. 6 along the diagonal (running from the upper left to the lower right), in order to gain a better appreciation of their differences. As shown, the $w_t(\widetilde{\mathbf{s}})$ of the LS solution, although showing a similar trend, differs from those of the LMS solutions in several aspects. For instance, it is seen to be smaller in magnitude at low D_{idx} values (which correspond to the top-left area of the target block), while appearing to be larger elsewhere. This implies that, on one hand, the template MV is not as reliable for compensating pixels in the upper left area as predicted by the theoretical results, and on the other hand, its effect on distant pixels is not negligible. Another observation is that the $w_t(\widetilde{\mathbf{s}})$'s of both LMS solutions have a minimum equal to zero and occurring roughly at D_{idx}=19 and 22, respectively–i.e., where their respective $\mathbf{s}_b^*$'s are located–whereas the minimum of the LS scheme is nonzero and is achieved at a larger D_{idx} value. Recall that $w_t(\widetilde{\mathbf{s}})$ indicates the likelihood of $\mathbf{v}_t$ being the ture motion of a pixel at $\mathbf{s}$ relative to the other hypothesis $\mathbf{v}_b$. Intuitively, we would expect $w_t(\widetilde{\mathbf{s}})$ to drop to zero (or, equivalently, $w_b(\widetilde{\mathbf{s}})$ to increase to unity) at $\mathbf{s}_b^*$, provided that the approximation of $\mathbf{v}_b$ as $\mathbf{v}(\mathbf{s}_b^*)$ and $\mathbf{v}_t$ as $\mathbf{v}(\mathbf{s}_t)$ is exact, which has been the basis for our derivation of the LMS solutions. However, the obvious mismatch between the LS and LMS results has proved this only a mathematical idealization.

To alleviate the mismatch, the modeling of $\mathbf{v}_t$ and $\mathbf{v}_b$ needs to be amended. One way of doing so is to model them probabilistically as the true motion associated with pixels around $\mathbf{s}_t$ and $\mathbf{s}_b$, respectively. We let $\mathbf{v}_t = \mathbf{v}(\mathbf{s}_t + \widehat{\mathbf{n}}_t)$ and $\mathbf{v}_b = \mathbf{v}(\mathbf{s}_b + \widehat{\mathbf{n}}_b)$, $\widehat{\mathbf{n}}_t$ and $\widehat{\mathbf{n}}_b$ being two zero-mean random vectors utilized to reflect the uncertainty nature of their sampling locations.[5] Such approach was justified in our prior work [5], where we showed, using Zheng's model [24], that

$$\begin{aligned} E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t+\widehat{\mathbf{n}}_t))^2] &\approx E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))^2]+\epsilon\delta_t = \epsilon\widehat{r}^2(\mathbf{s},\mathbf{s}_t)+\epsilon\delta_t \\ E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_b+\widehat{\mathbf{n}}_b))^2] &\approx E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))^2]+\epsilon\delta_b = \epsilon\widehat{r}^2(\mathbf{s},\mathbf{s}_b)+\epsilon\delta_b. \\ E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t+\widehat{\mathbf{n}}_t))d(\mathbf{s};\mathbf{v}(\mathbf{s}_b+\widehat{\mathbf{n}}_b))] &= E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))]. \end{aligned} \quad (25)$$

The δ_t (respectively, δ_b) denotes the trace of the covariance matrix of $\widehat{\mathbf{n}}_t$ (respectively, $\widehat{\mathbf{n}}_b$), indicating the dispersion of $\mathbf{v}_t$'s (respectively, $\mathbf{v}_b$'s) sampling location around its mean $\mathbf{s}_t$ (respectively, $\mathbf{s}_b$). From (25), this uncertainty causes the mean-squared prediction error to increase [see (5)], but does not influence the cross term $E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t+\widehat{\mathbf{n}}_t))d(\mathbf{s};\mathbf{v}(\mathbf{s}_b+\widehat{\mathbf{n}}_b))]$. Substituting these results into (20) and (21) arrives at another set of $\mathbf{s}_b^*$, $w_b^*(\widetilde{\mathbf{s}})$ and $w_t^*(\widetilde{\mathbf{s}})$, which compares well with the LS result, as shown in Fig. 7(b). Repeating the same computation with Tao's model [19] results in a similar effect (see the black curve with dots in the same plot).[6]

[5]Here the locations of $\mathbf{s}_t$ and $\mathbf{s}_b$ are deterministic.

[6]The evaluation of $E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t+\widehat{\mathbf{n}}_t))^2]$ and $E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t+\widehat{\mathbf{n}}_b))^2]$ using Tao's model becomes cumbersome, since it requires assuming the distributions of $\widehat{\mathbf{n}}_t$ and $\widehat{\mathbf{n}}_b$. We thus settle for a computation expedient that uses the analogy between Tao's and Zheng's models to similarly add an offset term to each of $E\left[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t))^2\right]$ and $E\left[d(\mathbf{s};\mathbf{v}(\mathbf{s}_b))^2\right]$ as estimates for $E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_t+\widehat{\mathbf{n}}_t))^2]$ and $E[d(\mathbf{s};\mathbf{v}(\mathbf{s}_b+\widehat{\mathbf{n}}_b))^2]$, respectively.

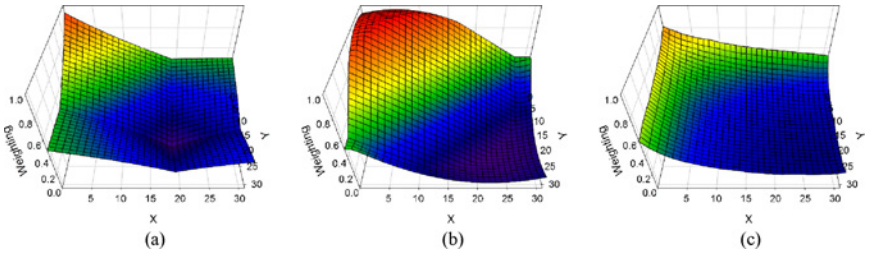

Fig. 6. Comparison of window functions $w_t(\widetilde{\mathbf{s}})$ computed based on (a) Tao's model, (b) Zheng's model, and (c) the LS solution. The target block is of size 32x32, along with an inverse-L-shaped template region of width 4. The LS solution is optimized for the Class B test sequences [2].

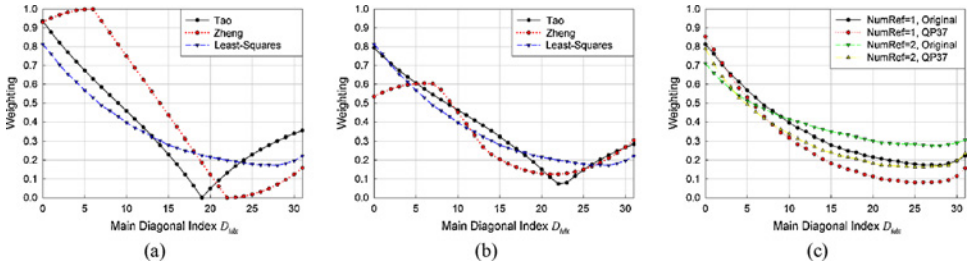

Fig. 7. Cross sections of $w_t(\widetilde{\mathbf{s}})$'s in Fig. 6 along the main diagonal running from the upper-left to the lower-right. Results in (a) and (b) correspond respectively to the cases *without* and *with* the amendment of $\mathbf{v}_t$ and $\mathbf{v}_b$'s modeling produced by Tao's ($\rho_m = 0.95, \delta_t = 0.18, \delta_b = 0.06$) and Zheng's ($\tau = 22.6, \delta_t = 405, \delta_b = 127$) models, respectively. Results in (c) effects of quantization and multiple reference frames. In the example, the reference frames are composed of one future frame and one past frame.

In order to apply the amended models above, we need to determine the parameters δ_t and δ_b. Obtaining their exact values, however, turns out to be difficult as it requires knowledge of the true motion field. For this, they are selected empirically by fitting the model predictions to the empirical results.

2) *Effects of Quantization and Multiple Reference Frames:* Up to here, our theoretical derivations (and experimental results) have been based on the assumptions that 1) both the predictors $I_{k-1}(\mathbf{s}+\mathbf{v}_t)$ and $I_{k-1}(\mathbf{s}+\mathbf{v}_b)$ come from the same *single* reference frame and that 2) motion estimation for $\mathbf{v}_t$ and $\mathbf{v}_b$ relies on *original* source frames. These assumptions are mainly for mathematical tractability. There, however, is no difficulty for our scheme to accommodate multiple reference frames and lossy compression. In the same way as before, the encoder first estimates $\mathbf{v}_t$ through template matching and then optimizes the choice of $\mathbf{v}_b$ according to $\mathbf{v}_t$'s values, except that the best match for a template region or a target block now has to be searched in multiple reference frames, which may have distortion due to residual quantization. The pixel-adaptive OBMC weighting still applies regardless of what reference frames $\mathbf{v}_t$ and $\mathbf{v}_b$ may refer to. A rigorous analysis for this more general case becomes difficult as it calls for the cross-correlation function between two motion fields and that between two intensity fields. We thus rely on simulations to verify whether our previous results can carry over to the present context.

Fig. 7(c) presents the $w_t(\widetilde{\mathbf{s}})$'s (of the LS solution) corresponding to a separate or a joint application of quantization and multiple reference frames. At first glance, they all have a similar waveform as before. A closer look at the figure indicates that using coded frames as references causes $w_t(\widetilde{\mathbf{s}})$ to taper more along the diagonal, while allowing multiple reference frames exerts an opposite effect. The former is attributed to the increased noise level in both the reference frame and the template region of a current frame, which makes the estimation of $\mathbf{v}_t$ less accurate. The latter arises because the predictor quality improves and the improvement is more prominent when the predictor is determined via template matching. Unlike the case with explicit MV coding, there is no need to trade off between predictor quality and motion overhead.

To conclude, this experiment shows that our theoretical predictions still remain valid (to a large extent) when there is more than one reference frame involved and when these reference frames undergo some distortion due to lossy compression.

F. Prediction Performance Comparison

This section compares the prediction performance of the LS and LMS solutions based on encoding the Class B test sequences [2]. Shown in Table III are their reductions (in percentage) in mean-squared prediction error relative to uni-predicted BMC, which has been chosen as the baseline to emphasize the gain due to the additional use of the template MV, $\mathbf{v}_t$. In particular, three heuristic variants of the proposed scheme (referred hereafter to as the TB-mode for convenience), demonstrating the effects when $\mathbf{v}_b$ is estimated independently or dependently of $\mathbf{v}_t$ and/or when a simple averaging of predictors is used in place of OBMC, are tested (see Section ♯2 of the table). Results are given for various combinations of block size (16, 32, 64) and QP (22, 37), with one or two reference frames. For a fair comparison of different algorithms, the procedure introduced in §II-E has been followed to ensure that the measurement of prediction error is carried out with respect to the same target blocks and that their reference frames are identical.

From Section ♯1 of the table, the superiority of our TB-mode over uni-predicted BMC is observed, when the LS or

TABLE III
PREDICTION PERFORMANCE OF TB-MODES AND BI-PREDICTION RELATIVE TO BMC

				Number of Reference Frames = 1						Number of Reference Frames = 2					
				QP22			QP37			QP22			QP37		
Sec	Mode	Weighting	ME Opt.	16	32	64	16	32	64	16	32	64	16	32	64
	TB	LS	o	10.4	11.0	12.1	4.5	4.6	5.9	17.0	22.3	27.5	4.1	6.5	11.6
	TB	LMS-Tao	o	9.7	10.0	9.0	3.8	3.6	3.1	15.4	21.1	25.9	3.4	5.7	10.1
♯1	TB	LMS-Zheng	o	6.0	6.3	7.4	1.6	1.1	2.0	10.4	13.7	17.6	0.9	2.1	5.9
	TB	LMS-Tao-Rev	o	10.1	10.7	10.4	4.3	4.2	4.6	16.8	22.0	26.8	4.0	6.3	11.1
	TB	LMS-Zheng-Rev	o	9.8	10.6	11.5	4.3	4.3	5.5	16.3	21.4	26.7	3.9	6.1	11.1
	TB	1/2		−14.1	−9.6	−5.6	−9.5	−8.3	−7.1	−10.1	−3.9	2.3	−9.3	−6.8	−3.5
♯2	TB	LS		4.1	5.8	6.9	1.3	2.2	2.8	7.1	10.8	14.0	1.3	3.1	5.9
	TB	1/2	o	2.6	2.7	3.5	0.0	−1.1	−0.9	10.6	15.7	21.3	0.1	1.6	6.5
♯3	Bi	1/2	o	15.2	12.8	11.6	7.9	6.6	6.7	25.1	30.2	35.4	8.4	11.0	17.8

o—$\mathbf{v}_b$ is optimized based on $\mathbf{v}_t$, or in the case of bi-prediction, the second MV is optimized according to the first MV found. In the case of single reference frame, both reference blocks of $\mathbf{v}_t$ and $\mathbf{v}_b$ come from the same reference frame. Negative values mean an increase in mean-squared prediction error.

LMS solutions are used. As might be predicted from Fig. 7(a), Tao's model [19] works better than Zheng's model [24], and those involving the amended modeling of $\mathbf{v}_t$ and $\mathbf{v}_b$ (indicated by the "-Rev" suffix) perform much closer to the LS solution.

Performance loss, however, may occur if we simply average the predictors derived from the independently found $\mathbf{v}_t$ and $\mathbf{v}_b$ to form a bi-prediction (see TB, 1/2, w/o ME Opt. in Section ♯2). In this case, further optimizing $\mathbf{v}_b$ based on $\mathbf{v}_t$ (TB, 1/2, w/ ME Opt.) or simply applying OBMC (TB, LS, w/o ME Opt.) helps to improve the performance, but is far from being ideal. Essentially, for the best prediction to be achieved, both the OBMC weights $w_b\left(\widetilde{\mathbf{s}}\right)$'s and $\mathbf{v}_b$ must be set right simultaneously. This is evident from the better performance of the LS and LMS solutions as compared to those heuristic variants.

In Table III, another comparison of interest is that between the TB-mode (of the LS version) and the traditional bi-prediction. The latter (Section ♯3 of the table) is found to outperform the former in almost every case, which is intuitively sound considering that two explicit MVs are used. There, however, are also situations where this increased overhead for MVs may not be justified. For instance, with one single reference frame, the TB-mode can perform very close to the bi-prediction. A similar phenomenon occurred in a side experiment that allows multiple decoded frames in the past to be referenced.

To summarize, the TB-mode can offer a superior prediction performance to uni-predicted BMC at almost the same motion cost, but its merit over the bi-prediction seems less obvious. Although, generally, it performs worse than the bi-prediction, the motion overhead incurred is also less. To see how this trade-off impacts the performance of a real codec, we will present simulation results based on the HM-6.0 software [2].

IV. EXPERIMENTAL RESULTS

This section reports experimental results on the performance of various TB-modes and their extensions to 1) adaptive template design, 2) multi-hypothesis prediction and 3) motion merging [23], when implemented with the HM-6.0 software [2]. To switch them on and off adaptively, one flag is sent for each non-skipped, $2N \times 2N$ Prediction Unit (PU) [3]. Further implementation details include: a) the search range for TMP is ± 4 pixels, with the central point given by the MV predictor and W_t=4; b) the OBMC weights for the LS solution are computed off-line based on several training sequences, while the model parameters of the LMS solutions are chosen empirically;[7] c) the OBMC weights are stored in tables at the encoder and decoder and thus need not be signaled; d) In the interest of space, the LMS results presented here are all *without* the amendment of MV modeling because the amended versions perform nearly the same as the LS solution.

Experiments were conducted following the common test conditions [2] defined for the development of HEVC [3]. This means, for each scheme tested, an encoding of 18 test sequences (which are grouped into five classes, Class A to E, with video resolution ranging from 416×240 to 2560×1600) at 4 QP values (22, 27, 32, and 37) for the 4 encoder configurations—Random Access High Efficiency (RA-HE10), Random Access Main (RA-Main), Low-Delay B High Efficiency (LB-HE10), and Low-Delay B Main (LB-Main). The coding gain over the HM-6.0 anchor was measured by the BD-rate saving. Note that negative values mean a rate reduction.

During the experiments, the encoding and decoding time increases relative to the anchor were also recorded to provide a rough indication of how the tested schemes may affect the codec's complexity. The decoding runtimes were measured sequentially on a single machine equipped with Intel Core i7-860 CPU, 16 GB RAM and Windows 7 64-bit, but to save time, the encoding runtimes were collected, while multiple encodings were executed simultaneously on a cluster of machines. Thus the encoding times may be unreliable. Anyhow, excessive interpretation of software runtimes should be avoided, as they can depend highly on the implementation quality.

A. Coding Performance

Table IV shows the coding performance of these schemes. Their results parallel the trend we observe from their prediction performance. As expected, those simple heuristics (Section ♯2 of the table) perform worse than the LS and LMS solutions (Section ♯1). In this circumstance, incorporating OBMC (TB, LS) seems more beneficial than optimizing the block MV (TB, 1/2, ME opt.). But, neither approach comes

[7]For Tao's model, ρ_m = 0.95; for Zheng's model, the clipping threshold τ is selected empirically to be $(2N)^2/2$ with $2N$ denoting the PU size.

TABLE IV
BD-RATE AND RUNTIME COMPARISONS OF TB-MODES AND TMP

Sec	Mode	Weighting	ME Opt.	RA-HE10	RA-Main	LB-HE10	LB-Main	All	Enc.	Dec.
	TB	LS	o	−0.5	−0.6	−0.6	−0.7	−0.6	107%	112%
♯1	TB	LMS-Tao	o	−0.5	−0.6	−0.6	−0.7	−0.6	107%	112%
	TB	LMS-Zheng	o	−0.4	−0.5	−0.8	−0.8	−0.6	107%	110%
	TB	1/2		−0.1	−0.1	0.0	0.0	0.0	103%	107%
♯2	TB	LS		−0.2	−0.3	−0.2	−0.3	−0.3	103%	108%
	TB	1/2	o	−0.2	−0.2	−0.1	−0.1	−0.2	106%	112%
♯3	TMP	1/2	o	0.0	0.0	−0.1	0.0	0.0	103%	112%

o—v_b is optimized based on v_t, or in the case of TMP, the second v_t is optimized according to the first v_t found.

TABLE V
BD-RATE AND RUNTIME COMPARISONS OF TB-MODES WITH FIXED OR VARIABLE TEMPLATE PATTERN

	(a) Fixed Template Pattern (Inverse-L)					(b) Variable Template Pattern				
	RA-HE10	**RA-Main**	**LB-HE10**	**LB-Main**	**All**	**RA-HE10**	**RA-Main**	**LB-HE10**	**LB-Main**	**All**
Class A	−0.3	−0.4	–	–		−0.7	−0.9	–	–	
Class B	−0.3	−0.4	−0.5	−0.5		−0.7	−0.9	−0.9	−1.0	
Class C	−0.6	−0.7	−0.8	−0.8		−1.2	−1.5	−1.5	−1.6	
Class D	−0.6	−0.7	−0.8	−0.8		−1.2	−1.4	−1.7	−1.7	
Class E	–	–	−0.5	−0.7		–	-	−1.1	−1.2	
Avg.	−0.5	−0.6	−0.6	−0.7	−0.6	−0.9	−1.1	−1.3	−1.4	−1.2
Enc.	106%	108%	107%	108%	107%	119%	121%	120%	123%	121%
Dec.	107%	112%	111%	117%	112%	108%	114%	114%	119%	113%

Results shown are with Tao's model and 2 hypotheses.

close to the LS and LMS schemes, which deliver, on average, an identical BD-rate saving of 0.6%. Without intra refresh, the gain is more noticeable in LB-Main, with the highest occurring in Class E (1.3%, on average) and achieved with Zheng's model. The latter arises because the associated window function happens to weight more heavily the template predictor [see Fig. 7(a)], while the high spatial and motion correlations inherent in the Class E sequences make the template MV more reliable for motion compensation. Surprisingly, performing TMP (TMP, 1/2, ME opt.) does not seem to provide any gain (see Section ♯3). This may be because of the poor prediction performance and the fact that its benefit in reducing motion cost diminishes as the MV coding becomes more efficient in the current HM. Another cause is the way it is implemented. For a fair comparison, TMP is currently made selectable only for nonskipped, $2N \times 2N$ PUs, just like the TB-mode, and the number of hypotheses used is limited to two. Both are expected to offer better performance when applied to the other PUs of different sizes.

From the right most columns, all the TB schemes cause about the same level of encoding and decoding time increases, and so does TMP. Essentially, the amount and type of computation conducted are similar. Their impact on encoding time (a 3–7% increase) is relatively modest, the reasons probably being that the template matching has a small search range and that the Enhanced Predictive Zonal Search algorithm has been implemented for speeding up the estimation of both the template and block MVs. But even so, the decoding time increase (7–12%) is still considerable. This does not come as a surprise, considering that template matching involves more data access and computation than motion compensation, which is one of the most computationally intensive parts in the decoding process.

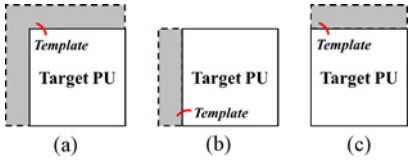

Fig. 8. Adaptive template switching.

B. Adaptive Template Switching

The coding performance of the TB-mode can improve if we are willing to pay extra computational cost and signaling overhead. For instance, to adapt to time-varying signal characteristics, the encoder can be provided with the flexibility to switch between different template designs (as shown in Fig. 8) at the $2N \times 2N$-PU level. Of course, the choice of the template needs to be coded in the bit-stream, and the block MV and OBMC weights must be optimized according to the procedure described in §III.

From Table V, this adaptive template switching further improves the rate saving by 0.4–0.7%, adding up to an average BD-rate saving of 1.2%, while elevating the encoding and decoding time ratios to 121% and 113%, respectively. The encoding time increases significantly due to the additional computation necessary for mode decision; the decoding time, on the other hand, has not changed much since the decoding process remains mostly the same as before. It, however, should be noted that extra tables may be needed at both the encoder and decoder for storing more OBMC weights.[8]

C. Multihypothesis Extension

So far, all the experimental results have been generated by limiting the predictor (hypothesis) number to two. We now

[8]With the LMS solutions, it is possible to compute OBMC weights on the fly, in which case tables are needless.

TABLE VI

BD-RATE AND RUNTIME COMPARISONS OF TB-MODES WITH MULTI-HYPOTHESIS PREDICTION

	(a) 2 Hypotheses (1 $\mathbf{v}_t$ + 1 $\mathbf{v}_b$)					(b) 3 Hypotheses (1 $\mathbf{v}_t$ + 2$\mathbf{v}_b$'s)				
	RA-HE10	**RA-Main**	**LB-HE10**	**LB-Main**	**All**	**RA-HE10**	**RA-Main**	**LB-HE10**	**LB-Main**	**All**
Class A	−0.7	−0.9	–	–		−1.1	−1.6	–	–	
Class B	−0.7	−0.9	−0.9	−1.0		−1.1	−1.4	−1.4	−1.6	
Class C	−1.2	−1.5	−1.5	−1.6		−1.5	−1.7	−1.8	−1.9	
Class D	−1.2	−1.4	−1.7	−1.7		−1.5	−1.6	−2.1	−2.0	
Class E	–	–	−1.1	−1.2		–	–	−1.4	−1.6	
Avg.	−0.9	−1.1	−1.3	−1.4	−1.2	−1.3	−1.5	−1.7	−1.8	−1.6
Enc.	119%	121%	120%	123%	121%	123%	126%	126%	129%	126%
Dec.	108%	114%	114%	119%	113%	112%	119%	119%	127%	119%
	(c) 3 Hypotheses (2 $\mathbf{v}_t$'s + 1 $\mathbf{v}_b$)					(d) 4 Hypotheses (2 $\mathbf{v}_t$'s + 2$\mathbf{v}_b$'s)				
Class A	−0.9	−1.3	–	–		−1.8	−2.5	–	–	
Class B	−1.0	−1.2	−1.1	−1.2		−1.4	−1.7	−1.7	−2.0	
Class C	−1.5	−1.7	−1.7	−1.8		−1.8	−1.9	−2.1	−2.1	
Class D	−1.4	−1.6	−1.9	−1.9		−1.8	−1.7	−2.5	−2.2	
Class E	–	–	−1.2	−1.3		–	–	−1.5	−1.7	
Avg.	−1.2	−1.4	−1.5	−1.6	−1.4	−1.7	−2.0	−2.0	−2.0	−1.9
Enc.	115%	117%	127%	124%	121%	123%	126%	126%	128%	126%
Dec.	122%	132%	133%	145%	133%	127%	140%	139%	153%	139%

Results shown are with adaptive template switching.

TABLE VII

BD-RATE AND RUNTIME COMPARISONS OF TB-MODES WITH MOTION MERGING

	(a) 2 Hypotheses (1 MRG $\mathbf{v}_t$ + 1$\mathbf{v}_b$)					(b) 3 Hypotheses (1 MRG$\mathbf{v}_t$ + 2 $\mathbf{v}_b$'s)				
	RA-HE10	**RA-Main**	**LB-HE10**	**LB-Main**	**All**	**RA-HE10**	**RA-Main**	**LB-HE10**	**LB-Main**	**All**
Class A	−0.3	−0.5	–	–		−0.5	−0.6	–	–	
Class B	−0.4	−1.5	−0.5	−0.7		−0.5	−0.7	−0.6	−0.9	
Class C	−0.7	−1.0	−0.9	−1.2		−0.9	−1.1	−1.1	−1.3	
Class D	−0.8	−1.0	−1.1	−1.4		−0.9	−1.1	−1.3	−1.5	
Class E	–	–	−1.0	−1.7		–	–	−1.1	−1.7	
Avg.	−0.5	−0.7	−0.8	−1.2	−0.8	−0.7	−0.9	−1.0	−1.3	−1.0
Enc.	117%	119%	118%	120%	118%	119%	121%	122%	124%	122%
Dec.	100%	101%	100%	101%	101%	99%	101%	100%	101%	100%
	(c) 3 Hypotheses (2 MRG $\mathbf{v}_t$'s + 1 $\mathbf{v}_b$)					(d) 4 Hypotheses (2 MRG$\mathbf{v}_t$'s + 2 $\mathbf{v}_b$'s)				
Class A	–0.4	−0.6	–	–		−0.7	−1.0	–	–	
Class B	−0.4	−0.6	−0.6	−0.8		−0.7	−0.9	−0.8	−1.1	
Class C	−0.9	−1.1	−1.1	−1.3		−1.0	−1.3	−1.3	−1.6	
Class D	−0.9	−1.1	−1.3	−1.5		−1.1	−1.3	−1.5	−1.8	
Class E	–	–	−1.1	−1.7		–	–	−1.3	−1.9	
Avg.	−0.6	−0.8	−1.0	−1.3	−0.9	−0.9	−1.1	−1.2	−1.5	−1.2
Enc.	116%	118%	117%	120%	118%	119%	121%	122%	124%	121%
Dec.	100%	102%	101%	102%	101%	101%	103%	102%	103%	102%

Results shown are with OBMC weight values rounded into power-of-two numbers.

relax this condition to explore the performance trade-offs. The extension to a hypothesis number greater than two is straightforward. When there is more than one template, or block, MV involved for motion compensation, their referred prediction blocks are simply averaged before the result is further weighted by OBMC. This avoids the need to create more window functions. For motion estimation, a greedy heuristic is implemented to estimate both the template and block MVs in a successive manner; the search criterion is to minimize the matching or prediction error when the MV in question is applied jointly with all the preceding MVs for template matching or temporal prediction. As an example, if two block MVs, $\mathbf{v}_{b1}$ and $\mathbf{v}_{b2}$, were to be estimated based on one template MV, $\mathbf{v}_t$, they would be searched sequentially using the following criteria:

$$\mathbf{v}_{b1}^{*}=\arg\min_{\mathbf{v}_{b1}}\sum_{\mathbf{s}\in\mathcal{B}}\left(I_k(\mathbf{s})-(1-w_b^{*}(\widetilde{\mathbf{s}}))I_{k-1}(\mathbf{s}+\mathbf{v}_t)-w_b^{*}(\widetilde{\mathbf{s}})I_{k-1}(\mathbf{s}+\mathbf{v}_{b1})\right)^2$$

$$\mathbf{v}_{b2}^{*}=\arg\min_{\mathbf{v}_{b1}}\textstyle\sum_{\mathbf{s}\in\mathcal{B}}(I_k(\mathbf{s})-\left(1-w_b^{*}(\widetilde{\mathbf{s}})\right)I_{k-1}(\mathbf{s}+\mathbf{v}_t)-\frac{1}{2}w_b^{*}(\widetilde{\mathbf{s}})\left(I_{k-1}\left(\mathbf{s}+\mathbf{v}_{b1}^{*}\right)+I_{k-1}(\mathbf{s}+\mathbf{v}_{b2})\right))^2.$$

Table VI compares the results of four experiments conducted with TB-modes that vary in hypothesis number. The legend to each experiment specifies the maximally allowed numbers of $\mathbf{v}_t$'s and $\mathbf{v}_b$'s for a $2N \times 2N$ PU coded in TB-mode. For instance, Experiment (b) ($1\mathbf{v}_t$+$2\mathbf{v}_b$'s) means that the encoder can choose adaptively the TB-mode with ($1\mathbf{v}_t$+$1\mathbf{v}_b$) or ($1\mathbf{v}_t$+$2\mathbf{v}_b$'s). To save bits, the signaling of $\mathbf{v}_b$ (or $\mathbf{v}_b$'s) reuses the syntax for representing the motion parameters of the ordinary inter prediction modes, while, in the present case, to indicate the number of $\mathbf{v}_t$'s used, one additional flag may be sent at the $2N \times 2N$ PU level.

From the table, increasing the maximum hypothesis number from 2 to 4 achieves an average BD-rate reduction of 1.9%, with a minimum of 1.4% and a maximum of 2.5%. The price paid, however, is a 39% increase in decoding time, which is about 20% higher than the two-hypothesis case, ($1\mathbf{v}_t$+$1\mathbf{v}_b$). Moreover, a doubling or more of memory access bandwidth solely for motion compensation can be expected in the worst case. As it stands, the setting ($1\mathbf{v}_t$+$2\mathbf{v}_b$'s) seems to offer

a better compromise between performance and the codec's runtime, with a comparable encoding/decoding time increase to ($1\mathbf{v}_t$+$1\mathbf{v}_b$), yet a moderate rate saving (1.6%, on average). The same observation does not apply to the other 3-hypothesis scheme, ($2\mathbf{v}_t$'s+$1\mathbf{v}_b$), which differs in using more $\mathbf{v}_t$'s. For this reason, its decoding time increase is as considerable as ($2\mathbf{v}_t$'s+$2\mathbf{v}_b$'s). Unlike the ordinary multihypothesis TMP [10], [18], which simply keeps the best N results in search of $\mathbf{v}_t$, ours estimates $\mathbf{v}_t$'s in a dependent manner; hence, the more the $\mathbf{v}_t$'s are used, the higher the codec's runtime, especially the decoder's.

D. Generalization to Motion Merging

In the TB-mode, there is no difficulty for the template MV $\mathbf{v}_t$ to be inferred by other techniques. In this section, we experiment with one particular generalization that borrows the notion of Motion Merging [23] to reuse MV(s) from a previously decoded neighboring PU as $\mathbf{v}_t$. When viewed from our framework, it is similar to deriving $\mathbf{v}_t$ from the motion sample taken at the center of the referred PU, in which case selecting adaptively from a range of candidate PUs is assimilated to switching between different template designs. Noting this analogy, we simply carry over the OBMC windows in §IV-B to the present case, rather than re-computing them according to the exact sampling location of $\mathbf{v}_t$, which is highly variable. As an example, when $\mathbf{v}_t$ is copied from the left PU, it is considered to be a MV as if it was produced by performing template matching with the configuration given in Fig. 8(b) and the corresponding OBMC weights are put into use. In doing so, the weight values are additionally rounded to power-of-two numbers for further simplification [4].

Comparing the results in Table VII with those in Table VI, we observe a 0.4–0.7% performance decline across different experiments, along with a significant decrease in decoding time. The former follows mainly from the fact that $\mathbf{v}_t$ now has a sampling location generally further away from the target block when compared with the case where it is obtained by template matching—a result that has also been noted in §II-E—and the latter arises as a consequence of removing template matching and rounding the weight values.

The above experiments demonstrate that the way how $\mathbf{v}_t$ is inferred in the TB-mode is capable of generalization. The high encoding and decoding times resulting from TMP can be resolved by using motion merging. Performance loss is inevitable; it, however, can be mitigated without significantly complicating the decoder [4].

V. Conclusion

In this paper, we proposed a bipredictive scheme that combines BMC and TMP predictors through OBMC. We first examined TMP in the context of motion field sampling and showed that the template MV may be viewed as the pixel true motion around the template centroid. It was thus concluded that in terms of prediction performance, TMP is inferior to BMC, but is, in general, superior to SKIP prediction. We then formulated, following a similar argument, the problem of finding another MV to best complement the template MV as the search of its sampling location in the motion field. This formulation has the advantage of allowing the problem to be solved analytically and leading to many useful insights into the solution, which would otherwise be difficult to see. We found that when sampled optimally, this MV, along with the template MV, forms a geometry-like motion partitioning. The notion of our scheme is capable of tremendous generalization. The template pattern need not be fixed, the number of hypotheses used can be more than two, and template matching can even be replaced with other decoder-side MV inference techniques, such as Motion Merging.

Our scheme had some drawbacks when it comes to hardware implementation. For TMP to work properly, pixels in the template region must be reconstructed prior to the motion estimation and compensation of a current PU. This data dependency complicated the pipeline design and hinders parallel processing. It was also the main reason why we restricted the use of TB-mode to $2N \times 2N$ PUs only. Although the problem can be alleviated further by using motion merging for MV inference, the sequential manner in which the template/merged and block MVs must be found is another obstacle. These open issues need further investigation, and we plan to address them in the future work.

Acknowledgment

The authors would like to thank all the anonymous reviewers and the Associate Editor M. Hannuksela for their invaluable comments that helped us to improve the quality of this paper.

References

[1] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264\ISO/IEC 14496-10 AVC)*, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-G050, 2003.

[2] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCVT-H1100, ITU-T/ISO/IEC JTC Joint Colllaborative Team on Video Calling, May 2012.

[3] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 6*, document JCVT-H1003, ITU-T/ ISO/IEC JTC1/SC29/WG11, Joint Colllaborative Team on Video Calling, Oct. 2012.

[4] C. C. Chen, Y. Y. Chen, C. L. Lee, W. H. Peng, and H. M. Hang, *CE2: Report of OBMC with Motion Merging*, document-F049, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Colllaborative Team on Video Calling, Jul. 2011.

[5] Y. W. Chen and W. H. Peng, "Parametric OBMC for pixel-adaptive temporal prediction on irregular motion sampling grids," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 1, pp. 113–127, Jan. 2012.

[6] Y. W. Chen, C. H. Wu, C. L. Lee, T. W. Wang, and W. H. Peng, *MB Mode with Joint Application of Template and Block Motion Compensations*, document JCTVC-B072 ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Joint Colllaborative Team on Video Calling, Jul. 2010.

[7] W. J. Chien, M. Karczewicz, and P. Chen, *TE1: Decoder-Side Motion Vector Derivation Report from Qualcomm*, document JCTVC-B097, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Colllaborative Team on Video Calling, Apr. 2011.

[8] D. Escoda, P. Yin, C. Dai, and X. Li, "Geometry-adaptive block partitioning for video coding," in *Proc. Int. Conf. Acoustics, Speech Signal Process.*, Apr. 2007, pp. 657–660.

[9] Y. W. Huang, C. Y. Chen, C. W. Hsu, J. L. Lin, Y. P. Tsai, J. An, and S. Lei, *TE1: Decoder-Side Motion Vector Derivation with Switchable Template Matching*, document JCTVC-B076, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Joint Colllaborative Team on Video Calling, Apr. 2011.

[10] S. Kamp, J. Balle, and M. Wien, "Multihypothesis prediction using decoder side motion vector derivation in inter frame video coding,"in *Proc. Vis. Commun. Image Process.*, Jan. 2009, pp. 725704.1-8.
[11] S. Kamp, M. Evertz, and M. Wien, "Decoder side motion vector derivation for inter frame video coding," in *Proc. Int. Conf. Image Processing*, Oct. 2008, pp. 1120–1123.
[12] M. Karczewicz, P. Chen, R. Joshi, X. Wang, W. J. Chien, and R. Panchal, *Video Coding Technology Proposal by Qualcomm Inc.*, document JCTVC-A121, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Jan. 2010.
[13] C. L. Lee, C. C. Chen, Y. W. Chen, M. H. Wu, C. H. Wu, and W. H. Peng, "Bi-prediction combining template and block motion compensations," in *Proc. Int Conf. Image Process.*, Sep. 2011, pp. 1221–1224.
[14] S. Lin, M. Yang, J. Zhou, J. Song, D. Wang, H. Yang, J. Fu, H. Yu, Y. Wang, L. Zhang, S. Ma, and W. Gao, *TE1: Huawei Report on DMVD Improvements*, document JCTVC-B037, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Jul. 2010.
[15] M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 693–699, Sep. 1994.
[16] K. Sugimoto., M. Kobayashi, Y. Suzuki, S. Kato, and C. S. Boon, "Inter frame coding with template matching spatio-temporal prediction," in *Proc. Int. Conf. Image Process.*, Oct. 2004, pp. 465–468.
[17] Y. Suzuki and C. S. Boon, "An improved low delay inter frame coding using template matching averaging," in *Proc. Picture Coding Symp.*, Dec. 2010, pp. 370–373.
[18] Y. Suzuki, C. S. Boon, and S. Kato, "Block-based reduced resolution inter frame coding with template matching prediction," in *Proc. Int. Conf. Image Process.*, 2006, pp. 1701–1704.
[19] B. Tao and M. T. Orchard, "A parametric solution for optimal overlapped block motion compensation," *IEEE Trans. Image Process.*, vol. 10, no. 3, pp. 341–350, Mar. 2001.
[20] M. Ueda and S. Fukushima, *TE1: Refinement Motion Compensation Using Decoder-Side Motion Estimation*, document JCTVC-B032, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Jul. 2010.
[21] R. Wang, L. Huo, S. Ma, and W. Gao, "Combining template matching and block motion compensation for video coding," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst.*, Dec. 2010, pp. 1–4.
[22] T. W. Wang, Y. W. Chen, and W. H. Peng, "Analysis of template matching prediction and its application to parametric overlapped block motion compensation," in *Proc. Int. Symp. Circuits Syst.*, May/Jun.2010, pp. 1563–1566.
[23] M. Winken, S. Bobe, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, D. Marpe, S. Oudin, M. Preib, H. Schwarz, M. Siekmann, K. Suhring, and T. Wiegand, *Description of Video Coding Technology Proposal by Fraunhofer HHI*, document JCTVC-A116, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Oct. 2010.
[24] W. Zheng, Y. Shishikui, M. Naemura, Y. Kanatsugu, and S. Itoh, "Analysis of space-dependent characteristics of motion-compensated frame differences based on a statistical motion distribution model," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 377–386, Apr. 2002.

Wen-Hsiao Peng received the B.S., M.S., and Ph.D. degrees in electronics engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1997, 1999, and 2005, respectively.

From 2000 to 2001, he was with the Intel Microprocessor Research Laboratory, Santa Clara, CA, USA, where he developed the first real-time MPEG-4 fine granularity scalability codec and demonstrated its application in 3-D, peer-to-peer video conferencing. In 2005, he joined the Department of Computer Science, NCTU, where he is currently an Associate Professor. Since 2003, he has actively participated in the International Organization for Standardization Moving Picture Expert Group (MPEG) digital video coding standardization process and contributed to the development of the MPEG-4 Part 10 AVC Amd.3 scalable video coding standard. He has published 45+ technical papers in the field of video and signal processing. His current research interests include high-efficiency video coding, scalable video coding, video codec optimization, and machine learning.

Dr. Peng is currently a Technical Committee Member for the *Visual Signal Processing and Communications* and *Multimedia Systems and Application Tracks* for the IEEE Circuits and Systems Society. He organized two special sessions on high-efficiency video coding in ICME 2010 and APSIPA ASC 2010, and was a Technical Program Co-Chair for VCIP 2011.

Chun-Chi Chen received the B.S. degree in computer science from National Central University, Nanjing, Taiwan, in 2007, and the M.S. degree in multimedia engineering from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 2009. He is currently pursuing the Ph.D. degree at the Institute of Computer Science and Engineering, NCTU.

Since 2011, he has been actively participating in the International Organization for Standardization (ISO) Moving Picture Expert Group (MPEG) digital video coding standardization process and contributed to the development of MPEG-H High Efficiency Video Coding standard. His current research interests include image/video compression and scalable video coding.