Subsampled Block-Matching for Zoom Motion Compensated Prediction

Lai-Man Po, *Senior Member, IEEE,* Ka-Man Wong, *Student Member, IEEE,* Kwok-Wai Cheung, *Member, IEEE,* and Ka-Ho Ng, *Student Member, IEEE*

***Abstract*—Motion compensated prediction plays a vital role in achieving enormous video compression efficiency in advanced video coding standards. Most practical motion compensated prediction techniques implicitly assume pure translational motions in the video contents for effective operation. Some attempts aiming at more general motion models are usually too complex requiring parameter estimation in practical implementation. In this paper, zoom motion compensation is investigated to extend the assumed model to support both zoom and translation motions. To accomplish practical complexity, a novel and efficient subsampled block-matching zoom motion estimation technique is proposed which makes use of the interpolated reference frames for subpixel motion estimation in a conventional hybrid video coding structure. Specially designed subsampling patterns in block matching are used to realize the translation and zoom motion estimation and compensation. No zoom parameters estimation and additional frame buffers are required in the encoder implementation. The complexity of the decoder is similar to the conventional hybrid video codec that supports subpixel motion compensation. The overall increase in memory requirement and computational complexity is moderate. Experimental results show that the new technique can achieve up to 9.89% bitrate reduction using KTA2.2r1 reference software implementation.**

***Index Terms*—Motion compensated prediction, motion estimation translation and zoom motion, video coding.**

I. Introduction

ADVANCED video coding standards like H.26X [1], [2] and MPEG [3], [4] series are all based on hybrid video coding schemes using transform coding and motion compensated prediction (MCP) to achieve high compression efficiency. The compression efficiency is mainly contributed by the motion estimation and compensation that removes temporal redundancy among video frames. Numerous approaches have been developed [5], [6] for efficient estimation and compensation but block-matching with simple translational motion vectors has been shown to be the most effective approach/technique in terms of simplicity and ease of implementation [7], [8].

In H.264/AVC [9], which is the state-of-the-art video coding standard, many advanced block-matching-based MCP techniques are adopted for achieving superior compression efficiency [10], [11]. However, it still uses the classical translation motion model and the improvements are partially obtained by performing block matching on interpolated reference frames with finer motion vector accuracy [12] of up to 1/4-pixel. Multiple reference frames (MRFs) [13] are also adopted to provide additional candidates for prediction over longer period of time. The most significant new technique adopted in H.264/AVC is variable block size (VBS) [14] motion compensation in tree-structured fashion from 16 × 16 down to 4 × 4 block sizes, which substantially improves the rate-distortion performance in complex motion areas of a frame. In addition, the best partition mode is selected based on rate distortion optimization (RDO) [15] using Lagrange multiplier for optimizing the tradeoff between the prediction gain and the side bits introduced by these techniques. These advanced techniques allow H.264/AVC to achieve 50% compression reduction compared with its predecessor H.263 with the same image quality. H.264/AVC is a very efficient coding scheme but its motion compensation is still based on the pure translation motion model. For the development of the next generation of video coding standards such as H.265 [16] of VCEG and high-performance video coding (HVC) [17] of MPEG, further improvements may be realized by addressing the limitations of this simple motion model.

The pure translation motion model used in conventional video coding schemes is shown in Fig. 1(a), in which 2-D rigid objects with translation motion are assumed. In actual 3-D environment, however, the scenes have depth and objects can move near to or away from the projection plane. The objects with different depths will be projected with different sizes due to the perspective effect as shown in Fig. 1(b). Also, objects may not be always rigid and without rotation. In recent years, several techniques have been proposed to improve the performance by addressing the limitations of the pure translation motion model. The most popular approach is to apply higher order motion models such as affine [18]–[20], perspective [21], polynomial [22], or elastic [23] ones. These higher order motion models include translation, rotation, and zoom motion to increase the prediction accuracy at the expense of additional motion vectors or parameters. In [18],

Manuscript received December 11, 2009; revised April 26, 2010 and June 21, 2010; accepted August 15, 2010. Date of publication October 14, 2010; date of current version November 5, 2010. This work was substantially supported by a GRF grant, under Project 9041501 (CityU 119909), from the City University of Hong Kong. This paper was recommended by Associate Editor J. F. Arnold.

L.-M. Po, K.-M. Wong, and K.-H. Ng are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: eelmpo@cityu.edu.hk; kmwong@ee.cityu.edu.hk; kahomike@gmail.com).

K.-W. Cheung is with the Department of Computer Science, Chu Hai College of Higher Education, Tsuen Wan, Hong Kong (e-mail: kwcheung@chuhai.edu.hk).

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.2010.2087474

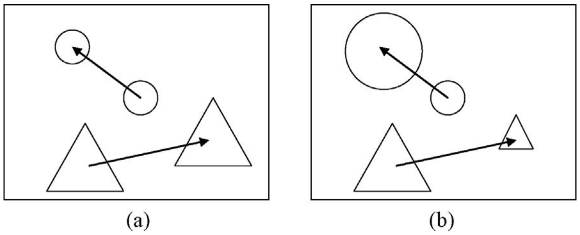


Fig. 1. 2-D frame based motion model. (a) Translation motion model. (b) Translation and zoom motion model.

two-stage motion compensation is carried out to determine local and global motion vectors involving affine parameter estimation at both stages. In [19], affine parameter sets are estimated and multiple "wrapped" frames are generated based on the parameter sets as references. The affine parameters are transmitted if the block in a wrapped frame is selected as the best prediction candidate. In [20], motion vectors of other blocks are used to make an affine transformed block as a searching candidate. It focuses on local complex general motion. All these methods require affine parameter estimation and their complexity limits the deployment of these motion models in practical applications.

In this paper, MCP based on a translation and zoom motion model as shown in Fig. 1(b) will be investigated for enhancing block-matching-based motion compensation performance. By combining translation and zoom motion components, the resulting motion compensation can effectively handle more real-life motion. Moreover, with subsampled block-matching-based implementation utilizing the interpolated reference frame for conventional subpixel motion estimation, the more general motion model can be easily and efficiently deployed in the existing video coding framework. It should be noted that the major difference between this paper and the methods in [18]–[20] is that the proposed method does not require affine parameter estimation and extra memory for storing additional reference frames. Zoom parameters are predefined using subpixel motion estimation interpolation scales. Thus the side information is minimal. The rest of this paper is organized as follows. In Section II, block-matching methods for conventional motion compensation will be first reviewed. The proposed model utilizing the zooming property will be introduced in Section III. A new subsampled block-matching technique for zoom motion compensated prediction implementation using the interpolated reference frame of the conventional subpixel motion estimation will also be described. Experimental results given in Section IV show the achievable gains of the proposed technique. Finally, conclusions will be given in Section V.

II. Block-Matching-Based Translation Motion Compensated Prediction

This section reviews the block-matching-based pure translation MCP techniques used in conventional hybrid video coding standards. Basically, this section forms a foundation for describing the details of the proposed translation and zoom motion MCP techniques.

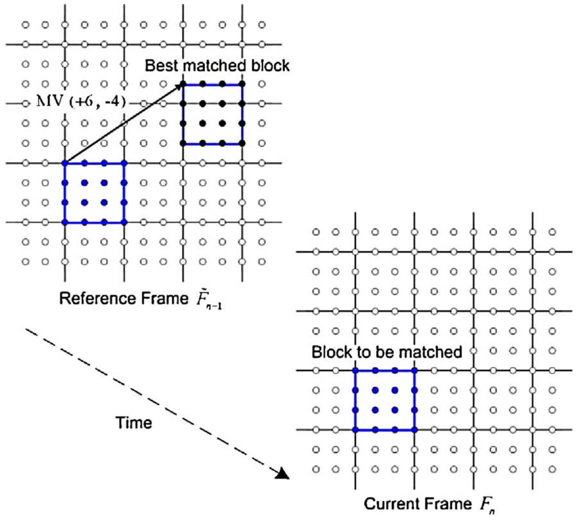


Fig. 2. Block-matching motion estimation with integer-pixel motion vector accuracy.

A. Conventional Block-Matching-Based MCP

Consider the nth image frame $\mathbf{F}_n$ of size $W \times H$ of a video sequence with pixel value $F_n(\mathbf{s})$ for each pixel location $\mathbf{s} = (x, y)$, in which x and y are integers such that $0 \leq x < W$ and $0 \leq y < H$. The pixel value of the reconstructed frame of a hybrid video coding system is denoted as $\tilde{F}_n(\mathbf{s})$. In block-matching-based translation MCP, frame is segmented into K non-overlapping blocks, $\left\{B_{i,n}\right\}_{i=1}^{K}$. In single reference frame MCP, each block is predicted from previous reconstructed frame $\tilde{\mathbf{F}}_{n-1}$ by block-matching motion estimation as shown in Fig. 2. The motion vector (MV), $\mathbf{v}_{i,n} = (u, v)_{i,n}$, specifying a spatial displacement for motion compensation of ith block in $\mathbf{F}_n$, is determined by block-matching process as

$$\mathbf{v}_{i,n} = \arg\min_{d} BDM_{B_{i,n}}\left(F_n(\mathbf{s}), \tilde{F}_{n-1}(\mathbf{s} - \mathbf{d})\right) \tag{1}$$

where block distortion metric (BDM) measures the difference between two blocks and the 2-D displacement vector $\mathbf{d}$ is limited to have finite integer vector component within a search area. A commonly used BDM is sum of absolute difference (SAD), which is defined as

$$SAD_B\left(F(\mathbf{s}), G(\mathbf{s})\right) = \sum_{s \in B} \left|F(\mathbf{s}) - G(\mathbf{s})\right|. \tag{2}$$

Then, a motion estimated frame's pixel value $\hat{F}_n(\mathbf{s})$ is formed for each $\mathbf{s} \in B_{i,n}$ as

$$\hat{F}_n(\mathbf{s}) = \tilde{F}_{n-1}(\mathbf{s} - \mathbf{v}_{i,n}), \mathbf{s} \in B_{i,n} \tag{3}$$

where $\tilde{F}_{n-1}(\mathbf{s})$ represents the reconstructed pixel value of the reference frame. The residue error $E_n(\mathbf{s})$ between the original frame pixel values $F_n(\mathbf{s})$ and its motion estimated $\hat{F}_n(\mathbf{s})$ is approximated as $\tilde{E}_n(\mathbf{s})$. The reconstructed frame's pixel values are obtained by

$$\tilde{F}_n(\mathbf{s}) = \hat{F}_n(\mathbf{s}) + \tilde{E}_n(\mathbf{s}), \mathbf{s} \in B_{i,n}. \tag{4}$$

In the first generation of video coding standards such as MPEG-1, the motion vector $\mathbf{v}_{i,n} = (u, v)_{i,n}$ representation is limited to integer-pixel accuracy as shown in Fig. 2.

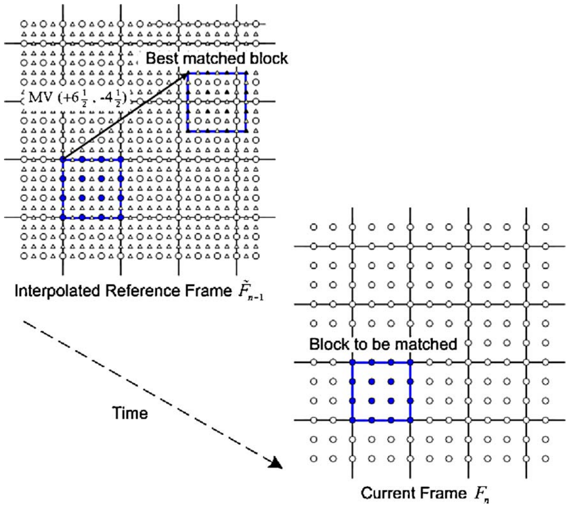


Fig. 3. Block-matching motion estimation with ½-pixel motion vector accuracy.

To enhance the conventional MCP performance, subpixel (fractional-pixel) motion vector precision was introduced into the second-generation video coding standards such as MPEG-2 by performing block-matching on the interpolated restructured reference frame. Let $f_t(\mathbf{s}_c)$, $\mathbf{s}_c \in \Re^2$, be the original continuous in time and space dynamic scene that is being imaged and the video sequence $f_n(\mathbf{s})$ be the scene after sampling according to the Nyquist criterion in time and space. That means $F_n(\mathbf{s}) = f_n(\mathbf{s})$ for integer grid position, $\mathbf{s} \in Z^2$. In subpixel block-matching MCP, the motion vector is determined as

$$\mathbf{v}_{i,n} = \arg\min_{v} BDM_{B_{i,n}}\left(F_n(\mathbf{s}), \tilde{f}_{n-1}(\mathbf{s}-\mathbf{v})\right) \tag{5}$$

where $\mathbf{v} \in \Re^2$ is not limited to integer vector components. In practice, the pixel values at subpixel accuracy location $\tilde{f}_{n-1}(\mathbf{s}-\mathbf{v})$ can be obtained from interpolation of $\tilde{F}_{n-1}(\mathbf{s})$ to achieve higher prediction performance of MCP.

Fig. 3 shows an example of the half-pixel (½-pixel) motion vector accuracy block-matching motion estimation. The previous reconstructed frame is interpolated by a factor of 2 in both horizontal and vertical spatial directions to generate a higher resolution reference frame. The circular dots represent the original pixels of the reference frame pixel $\tilde{F}_{n-1}(\mathbf{s})$ and the triangular dots represent the interpolated pixels generated by interpolation filter based on the original pixels (circular dots). With these interpolated pixels, block-matching can be performed between interpolated-pixels-blocks and the image block of the current frame for motion estimation with subpixel motion vector accuracy. For example, the best-matched block in Fig. 3 is a block composed of interpolated pixels (solid triangular dots) with motion vector (6½, −4½).

B. Multiple Reference Frames MCP

The weakness of single reference frame in cases of temporary occlusion and periodic deformation may be resolved by selecting another reference frames. That is the idea of MRF motion compensation to provide additional candidates for prediction over longer period of time. Fig. 4 shows an example of MRF motion estimation in case of temporary occlusion, where two good matching blocks cannot be found in the previous reference frame. Since MRF uses frames with different time delays, the reference frame for generating the prediction is not limited to the previous frame. Thus, additional information identifying the reference frame is required and the displacement expression becomes

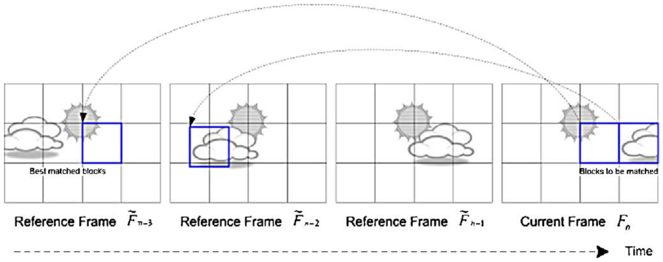

Best matched blocks
Blocks to be matched
Reference Frame $\tilde{F}_{n-3}$
Reference Frame $\tilde{F}_{n-2}$
Reference Frame $\tilde{F}_{n-1}$
Current Frame F_n
Time

Fig. 4. Example of MRF in case of temporary occlusion.

$$\left(m, \mathbf{v}_{i,n}\right) = \arg\min_{m,\mathbf{v}} BDM_{B_{i,n}}\left(F_n(\mathbf{s}), \tilde{f}_m(\mathbf{s}-\mathbf{v})\right) \tag{6}$$

where m is the reference frame index. MRF motion compensation is also a new feature introduced in H.264/AVC to improve video coding performance.

III. Block-Matching Translation and Zoom Motion Compensated Prediction (BTZMCP)

The MCP techniques described in Section II and the variable block size techniques adopted in H.264/AVC intrinsically assume a pure translation motion model as shown in Fig. 1(a). To achieve a more realistic motion model as shown in Fig. 1(b), the prediction can be generalized to include zoom reference frames $\tilde{f}_m(\mathbf{s}/a)$, $a \in \Re$. The zoom factor a is determined as an additional parameter in the motion estimation process as

$$\left(a, m, \mathbf{v}_{i,n}\right) = \arg\min_{a,m,v} BDM_{B_{i,n}}\left(F_n(\mathbf{s}), \tilde{f}_m(\mathbf{s}/a-\mathbf{v})\right). \tag{7}$$

For $a > 1$, $\tilde{f}_m(\mathbf{s}/a)$ is a zoom-in reference frame. For $a < 1$, $\tilde{f}_m(\mathbf{s}/a)$ is a zoom-out reference frame. In block-matching MCP, since each block $B_{i,n}$ may has its own zoom factor a, a single frame may be composed of both zoom-in and zoom-out blocks of different zoom factors. Thus, this block-matching translation and zoom MCP (BTZMCP) as described by (7) can better model the real-life situation in which the projection of different regions or objects of a scene onto the imaging plane may exhibit zoom effects of various degrees such as the example shown in Fig. 1(b).

A. Multiple Zoom Reference Frames

The most straightforward way to implement the proposed BTZMCP is to extend the set of reference frames for prediction that include zoom reference frames. This approach may be considered as a generalization of MRF with a set of zoom reference frames associated with each temporal reference frame in the original MRF method. The idea is illustrated in Fig. 5 in which the time axis represents the original temporal reference frames in MRF and the additional zoom axis shows

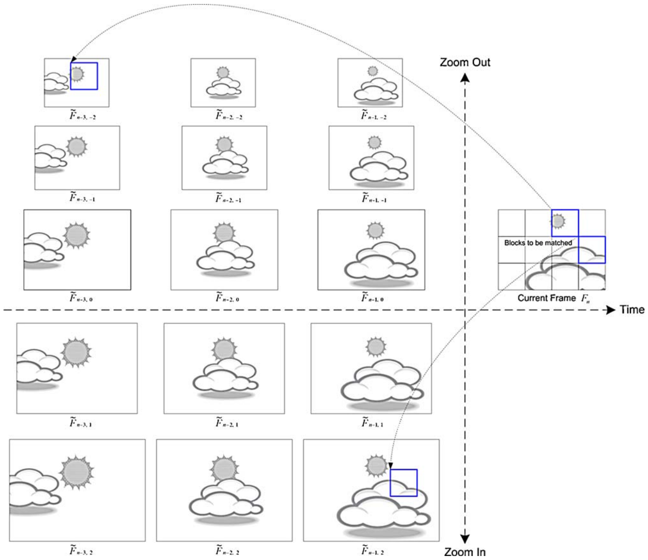


Fig. 5. Example of block-matching MCP with multiple zoom reference frames. The cloud is moving to the right with zoom-in effect along the time axis. The sun is stationary with zoom-out effect along the time axis.

TABLE I
LIST OF ZOOM FACTORS OF CORRESPONDING ZOOM STEP USED IN THE ANALYSIS

Step size	Zoom factors a_i of zoom reference frames
1/16	{5/16, …, 15/16, 16/16, 17/16, …, 26/16}
1/32	{21/32, …, 31/32, 32/32, 33/32, …, 42/32}
1/64	{53/64, …, 63/64, 64/64, 65/64, …, 74/64}
1/128	{117/128, …, 127/128, 128/128, 129/128, …, 138/128}

the set of zoom reference frames for each temporal reference frame. Along the zoom axis, the zoom reference frames are frames with a set of predefined zoom factor a_i of discrete values. The time axis shows the motions of two objects. The cloud is moving to the right and toward the camera (zoom–in effect) while the sun is relatively stationary with zooming out effect along the time axis. In this case, MRF cannot provide good predictions for both the cloud and the sun objects but two of the zoom reference frames can provide much better motion prediction as shown in Fig. 5.

To investigate the prediction gain, BTZMCP using multiple zoom reference frames is applied to sequences *Akiyo*, *Foreman*, *Mobile*, and *Stefan* with CIF resolution and 300 frames. The zoom reference frames are generated based on the previous frame by bilinear interpolation. The macroblock size is fixed at 16 × 16. The search windows size is set at ±16 pixels and exhaustive search is used within the search window. Sum of squared error is used for the block distortion measure. With zoom steps from 1/16 to 1/128 and ±11 zoom level as shown in Table I, a total of 23 zoom reference frames are available for each motion search.

PSNR improvement of BTZMCP relative to the ordinary block motion compensated prediction without zoom reference frames is shown in Fig. 6. It shows that BTZMCP significantly improves the estimation accuracy. Also, better improvement can be achieved with smaller zooming steps. For the sequences under investigation, improvement up to 1.69–2.34 dB can be obtained. The prediction gains obtained from different number of available zoom levels ranging from 3 to 23 are shown in Fig. 7. As expected, more zoom levels available will improve the prediction gain but the improvement is less significant when the number of zoom levels increases from 11 to 23. As the zoom reference frames are generated from a single temporal reference frame, these results demonstrate that BTZMCP is a promising approach for further enhancing the prediction efficiency in video coding.

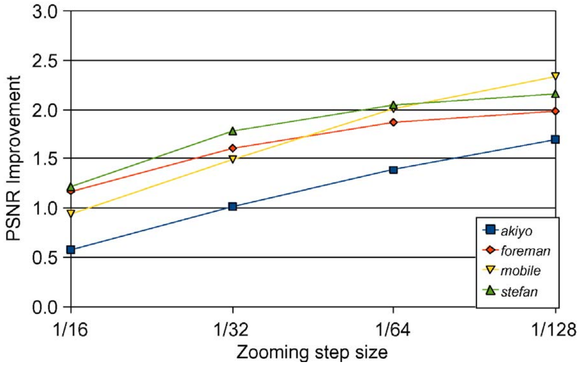


Fig. 6. PSNR improvement of BTZMCP with various zooming step sizes with bi-linear interpolation.

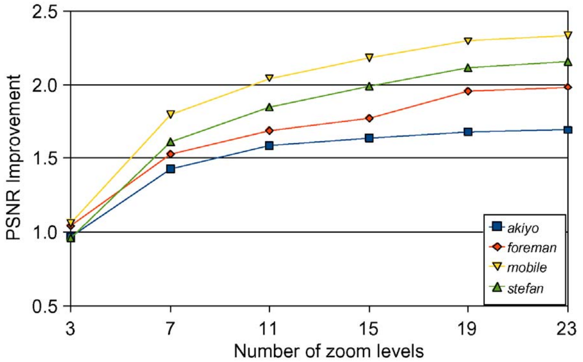


Fig. 7. PSNR improvement of BTZMCP with various numbers of zooming levels.

Unfortunately, there are several problems in practical implementation of this multiple zoom reference frames approach. Firstly, the computational and memory requirements are both significantly increased. The extra computation, as compared with MRF, is composed of the estimation of the zoom factors a_i and generation of the zoom reference frames based on these estimated zoom factors. The block-matching process in these zoom reference frames is similar to the MRF approach. In addition, the memory required to store these zoom reference frames is huge especially for the zoon-in reference frames. Besides these complexity problems, the most challenging problem is to develop good zoom factors estimation techniques for generating appropriate zoom reference frames.

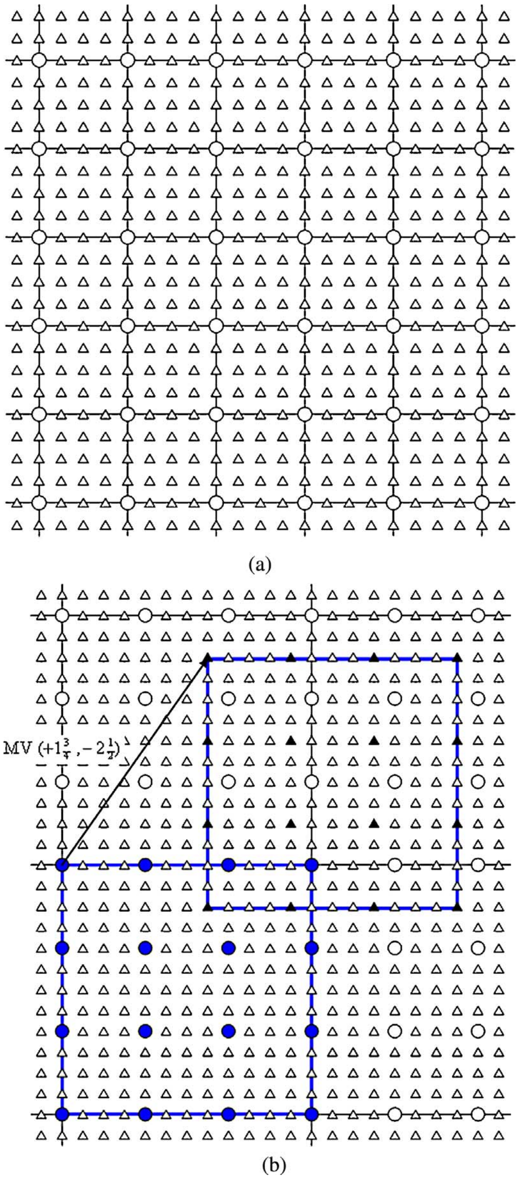
B. Subsampled Block-Matching on the Subpixel Motion Estimation Interpolated Frame

To tackle the problems of generating zoom reference frames in BTZMCP implementation, a new subsampled block-matching technique utilizing the interpolated reference frame in subpixel motion estimation is proposed in this section. The main idea is to make use of the interpolated reference frame for subpixel motion estimation to create the zoom reference frames by several pre-defined subsampling factors. In practice, the block-matching process on the zoom reference frame can be performed similar to the subpixel motion estimation but using specially designed sampling pattern for realizing block-matching on the zoom reference frames. As subpixel motion estimation is commonly used in advanced video coding standards such as MPEG-4 and H.264/AVC, interpolated zoom-in reference frames are already available. Thus, no additional memory is required to store the zoom reference frames and this approach can be easily integrated into the conventional video coding standards that support subpixel motion estimation.

In the proposed method, zoom factors are limited by the maximum resolution of the interpolated reference frame and the choice of the subsampling factors. For example, the reference frame of ¼-pixel accuracy motion estimation is interpolated with a factor 4 in both horizontal and vertical axes as shown in Fig. 8(a). In this figure, the circular dots represent original pixels of the reference frame $\tilde{\mathbf{F}}_{n-1}$ and the triangular dots are the interpolated pixels generated from these circular dot pixels. Subpixel motion estimation is implemented by block-matching with subsampling this frame with 4-time resolution in both horizontal and vertical axes with shift of the subpixel locations. An example of a 4 × 4 matched-block (black dots) with motion vector displacement (1¾, −2½) in ¼-pixel subpixel motion estimation is shown in Fig. 8(b). The block-matching is performed in the same resolution but with subpixel block displacement.

Similarly, the block-matching with zoom-in and zoom-out reference frames in BTZMCP can be implemented by subsampling the same interpolated reference frame with specially selected sampling rate. For example, if the sampling rate is 1 in both horizontal and vertical directions as shown in Fig. 9, which corresponds to a block (black dots) in zoom-in reference frame with zoom factor $a = 4$. The best matched-block in the example of Fig. 9 is represented by a motion vector with displacement of (5, −9) in the 4 times zoom-in reference frame. To perform block-matching on a zoom-in reference frame with zoom factor of 4/3, we can subsample the interpolated frame in both horizontal and vertical directions by a factor of 3 as shown in Fig. 10. On the other hand, we can also perform the block-matching on zoom-out reference frame if the subsampling rate is higher than 4. For example, a 4 × 4 block of the zoom-out frame with zoom factor of 4/5 is shown in Fig. 11 by subsampling the interpolated frame by 5 in both horizontal and vertical directions.

In general, a temporal reference frame can be interpolated by a factor S that may not be the same as the subpixel motion estimation accuracy. In modern hybrid video coding system, the half-sample positions of the interpolated frame can be obtained by applying a 1-D 6-tap FIR filter horizontally and vertically. Pixel values at quarter-sample, 1/8-sample, and so on are generated by bilinear interpolation of the integer-sample and half-sample positions. The interpolated frame has the


(a)

(b)

Fig. 8. (a) Pixel locations of a ¼-pixel interpolated reference frame. (b) Example of ¼-pixel accuracy subpixel motion estimation with MV (1¾, −2½).

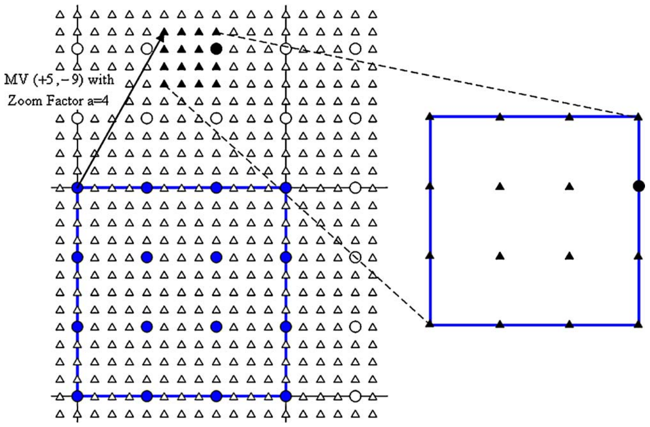


Fig. 9. Example of MV (5, −9) with zoom factor a = 4.

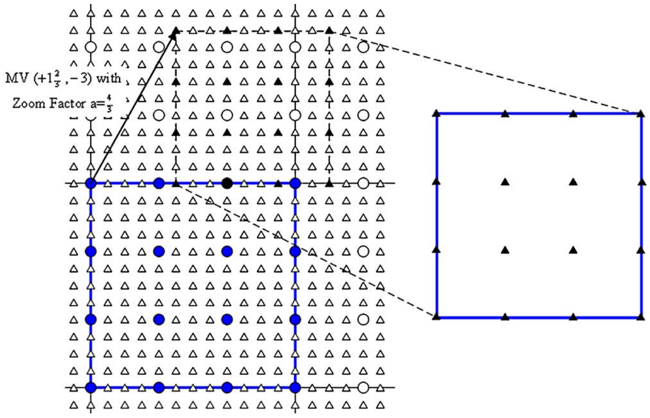


Fig. 10. Block-matching on a reference frame of zoom factor $a = 4/3$.

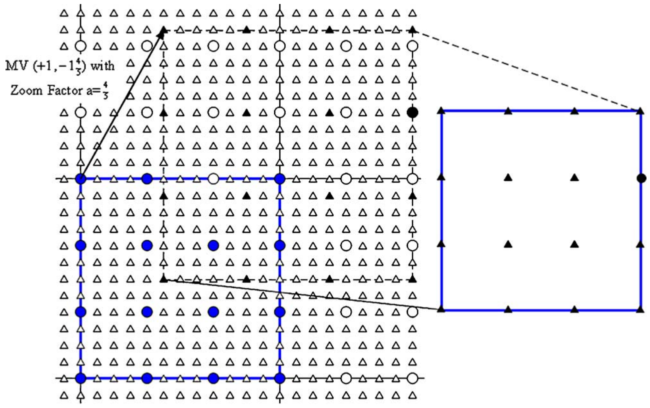


Fig. 11. Block-matching on a reference frame of zoom factor $a = 4/5$.

maximum resolution S times larger than the original temporal reference frame in the horizontal and vertical dimensions. This interpolated frame can be viewed as a zoom-in frame with zoom factor S. Then, a set of lower resolution frames can be obtained from this maximum resolution frame by different subsampling rates. For zooming application, the same subsampling rate is applied to both the horizontal and vertical dimensions. Thus, the corresponding horizontal or vertical resolution of the resulting set of zoom reference frames is ranging from S to S/M and the possible zoom factors are given by

$$\{S/1, S/2, S/3, \ldots, S/j, \ldots, S/M\} \tag{8}$$

where j is the subsampling rate and M is maximum subsampling rate. For $j < S$, the zoom reference frames are zoom-in frames. For $j > S$, the zoom reference frames are zoom-out frames. In practice, it is not necessary to use all zoom factors provided by (8) as it will generate a heavy loading on motion estimation. Based on the results in Fig. 6, the zoom factors should be smaller for achieving better prediction performance. It is because within the short time interval covered by most MCP reference frame, the zooming effect, i.e., the magnitude of the zoom factors, are usually small for video sequences with continuous motions. For a given number of N_1 zoom-in factors and N_2 zoom-out factors, the set of zoom factors

$$\left\{ \begin{array}{l} \dfrac{S}{S-N_1}, \dfrac{S}{S-N_1+1}, \ \ldots\ , \dfrac{S}{S-1}, \dfrac{S}{S}, \dfrac{S}{S+1}, \ \ldots\ , \\ \dfrac{S}{S+N_2-1}, \dfrac{S}{S+N_2} \end{array} \right\} \tag{9}$$

should be selected. If the subpixel motion estimation accuracy is $1/s$-pixel and we set the $S = s$, then the maximum resolution zoom-in frame will be with s time resolution in both horizontal and vertical directions. For example in ¼-pixel motion estimation, we can select $S = s = 4$. If both the number of zoom-in and zoom-out factors are chosen to be 2 ($N_1 = N_2 = 2$), based on (9) the recommended set of zoom factors is

$$\{2, 4/3, 1, 4/5, 2/3\} \tag{10}$$

in which the subsampling rates are selected around the non-zooming subsampling rate with additional two zoom-in and two zoom-out levels. Moreover, the subpixel motion estimation accuracy is different in each zoom reference frame. For example, we cannot perform subpixel motion estimation for zoom factor with $a = S$ as shown in the Fig. 9, due to no higher resolution pixels are available. For the case with zoom factor of 4/3 in Fig. 10, the subpixel accuracy is only up to 1/3-pixel. In general, subpixel motion estimation accuracy in each zoom factor is equal to $1/j$-pixel, where j is the subsample factor of (8).

The disadvantage of this method is that it does not provide flexible zoom factors compared to generate the zoom reference frames directly. However, this approach provides a fast and simple implementation of BTZMCP in modern video codec implementation. Moreover, no additional memory is required in both encoder and decoder implementation as compared with subpixel motion estimation. The complexity of the decoder is also very similar to conventional hybrid video coding methods that support MRF and subpixel motion estimation.

C. Implementation of BTZMCP

To integrate the proposed BTZMCP into modern video codec, the cost of motion vector in determining the prediction should also be considered. The motion vectors constituting the spatial displacement, frame reference and zoom factor are required to be coded and then transmitted to the decoder. In H.264/AVC, rate distortion optimized motion search is implemented by minimization of the Lagrangian cost function. Thus, (7) is modified as

$$\begin{aligned} &\left(a, m, \mathbf{v}_{i,n}\right) \\ &= \arg\min_{a,m,\mathbf{v}} \begin{pmatrix} BDM_{B_{i,n}}\left(F_n(\mathbf{s}), \tilde{f}_m(\mathbf{s}/a - \mathbf{v})\right) \\ +\lambda_{motion} R(a, m, \mathbf{v}) \end{pmatrix} \end{aligned} \tag{11}$$

where λ_{motion} is the Lagrangian multiplier for motion search used in H.264/AVC and $R(a, m, \mathbf{v})$ is the bitrate associated with a particular choice of motion vector. The motion vector $(a, m, \mathbf{v})$ for each block mode (inter modes of various subblock sizes, intra mode, and so on) is the one that gives minimum Lagrangian cost.

TABLE II
DEFINITION OF ZOOM PARAMETER AND ASSOCIATED VALUES IN THE IMPLEMENTATION

Zoom parameter	0	1	2	3	4
Subsampling rate (8× interpolated frame with S = 8)	8	9	7	10	6
Zoom factor	1	8/9	8/7	4/5	4/3

Although the nature and generation of zoom reference frames in BTZMCP are very different from that of the temporal reference frames, their usages in MCP are the same, i.e., to provide candidates for motion prediction. Thus, by considering the addition of zoom reference frame as increasing the time window of the temporal reference frame, the zoom reference frame index can be easily embedded with the frame number parameter of H.264/AVC bit-stream provided that the sum of both types of reference frames do not exceed the maximum number of reference frames supported. For example, the zoom factor of the reference frame can be embedded to the frame number by offsetting the original frame number by {0, 16, 32, 48, 64} to indicate the reference frame's zoom parameter {0, 1, 2, 3, 4}, respectively. It can also be represented by the following equation:

$$n' = 16z + n \tag{12}$$

where n' is the modified frame number, n is the original temporal reference frame number, and z is the zoom parameter. For example, to indicate the zoomed frame with temporal reference frame 3 and zoom parameter 2, the modified frame number should be 16 × 2 + 3 = 35. This design assumes the support of up to 16 temporal reference frames. In addition, this approach can add new parameter without changing the syntax of the bit stream. For example, if the subsampled block-matching BTZMCP is applied on 1/8-subpixel motion estimation accuracy ($S = s = 8$) with two zoom-in frames and two zoom-out frames ($N_1 = N_2 = 2$), then, based on (9), the following set of zoom factors should be used:

$$\{4/3, 8/7, 1, 8/9, 4/5\}. \tag{13}$$

The representation of zoom factors using zoom parameters is defined in Table II. The overhead of the zoom information is therefore added onto representation of zoom vector's reference frame. This method is not optimal for rate-distortion (RD) performance but it is a simple way to implement the coding of the zoom information in KTA reference software.

Theoretically, for subsampling process with factor larger than the interpolation factor used in generating the new resolution frame, the overall process is effectively a decimation operation on the original frame and the aliasing effect may exist in the subsampled blocks. Common practice is to apply lowpass filtering before subsampling in order to reduce the aliasing effect. However, for the sake of lower computation cost, no filtering is employed before subsampling. This decision is feasible in BTZMCP as the main purpose is to generate more prediction candidates and the prediction error including aliasing will be absorbed in residue coding stage.

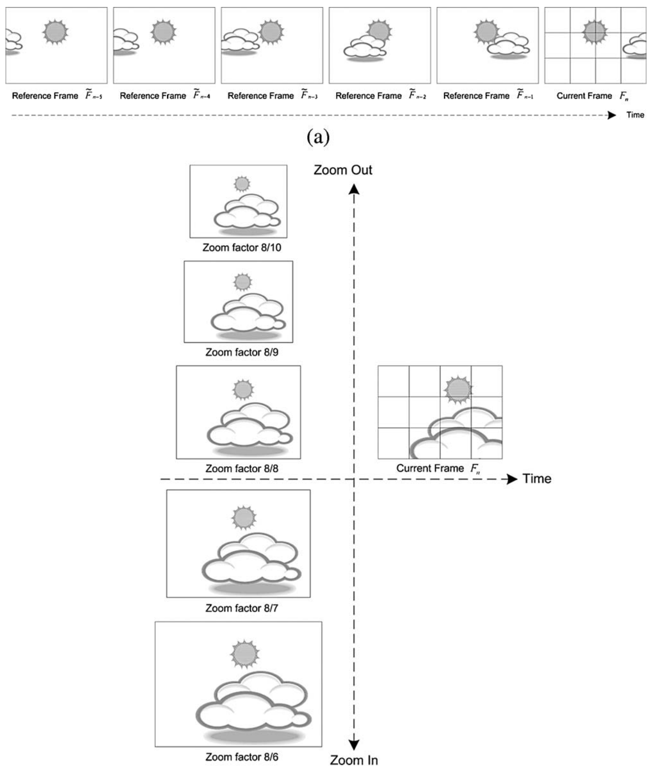
Conceptually, a set of zoom reference frame is generated for one or more temporal reference frame(s) and the storage requirement is proportional to the number of zoom levels used. Practically, additional storage for zoom reference frame is not required as, in subsampled block-matching, prediction candidates for block-matching are obtained by subsampling from the interpolated frame for subpixel motion estimation. As that frame is already available, no additional frame is required. Additional storage compared with the existing codec may be involved if the resolution of the interpolated frame is higher than that of the ¼ pixel frame used in the existing H.264/AVC codec or 1/8 pixel frame used in KTA reference software. In that case, additional storage for an interpolated frame for each temporal reference frame is required.

The main cost of subsampled block-matching is the larger pool for prediction searching. Since the cost of searching a prediction candidate for a temporal reference frame is the same as that for a zoom reference frame. Practically, the searching operation is very similar in searching temporal and zoom reference frames for prediction candidate. The main difference is the retrieval of the block pixel values. In conventional block matching, a spatially continuous block of pixel values is read from the frame buffer while, in subsampled block-matching, the prediction block values are obtained by subsampling a spatial area in the frame buffer. The main computation costs are still the block distortion measure and comparison in both temporal and zoom reference frame cases. Thus, a simplified way to compare the computation complexity is to compare the number of reference frame (temporal and zoom ones). For example, BTZMCP in Fig. 5 has three temporal reference frames and five zoom levels. Thus, the total number of reference frames is 15. Therefore, the computational complexity is about 15 times higher than that of the single reference MCP algorithm. Moreover, three times more memory is required. Fig. 12 shows an example of MRF with five temporal reference frames and an example BTZMCP with five zoom reference frames. As both configurations have five reference frames, they have similar computational complexity but the memory requirement of BTZMCP is five times lower than MRF if the interpolated reference of subpixel motion estimation is used to implement BTZMCP.

Applying BTZMCP to all the temporal reference frames in MRF will significantly increase the computational complexity of the encoder without greatly improving the coding efficiency. A trade-off is to apply BTZMCP to only the previous reference frame as shown in Fig. 13 in which a T-shaped time and zoom reference frames model is formed. The computational requirement of this configuration is about two times compared with using only MRF. The rate-distortion performance of this configuration is evaluated in experimental results section.

IV. Experimental Results

The RD performance evaluation is investigated by integrating the proposed BTZMCP with subsampled block-matching technique in KTA2.2r1 [24], [25], which is a reference software for future video coding standard development. Experimental results are compared to the results generated


Reference Frame $\widetilde{F}_{n-5}$
Reference Frame $\widetilde{F}_{n-4}$
Reference Frame $\widetilde{F}_{n-3}$
Reference Frame $\widetilde{F}_{n-2}$
Reference Frame $\widetilde{F}_{n-1}$
Current Frame F_n
Time
Zoom Out
Zoom factor 8/10
Zoom factor 8/9
Zoom factor 8/8
Current Frame F_n
Time
Zoom factor 8/7
Zoom factor 8/6
Zoom In

(a)

(b)

Fig. 12. (a) MRF with five temporal reference frames. (b) BTZMCP with five zoom reference frames.

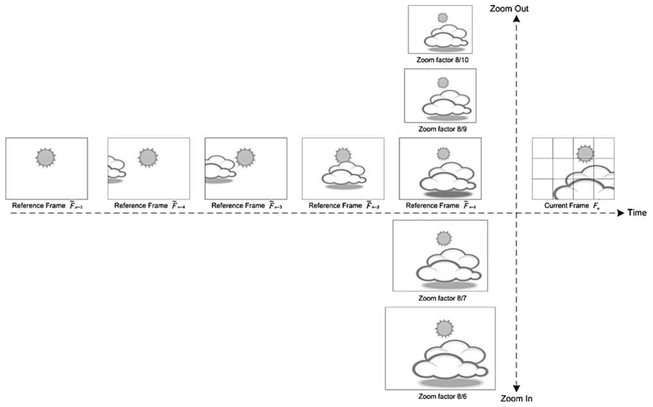
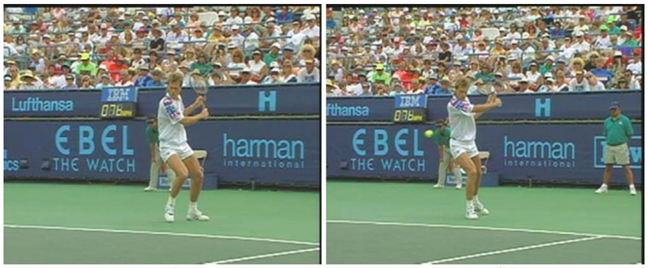

Zoom Out
Zoom factor 8/10
Zoom factor 8/9
Reference Frame $\widetilde{F}_{n-5}$
Reference Frame $\widetilde{F}_{n-4}$
Reference Frame $\widetilde{F}_{n-3}$
Reference Frame $\widetilde{F}_{n-2}$
Reference Frame $\widetilde{F}_{n-1}$
Current Frame F_n
Time
Zoom factor 8/7
Zoom factor 8/6
Zoom In

Fig. 13. T-shaped BTZMCP model.



(a)

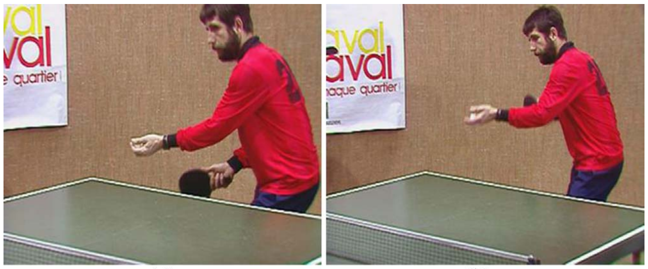
(b)

Fig. 14. Two frames in *Stefan* sequence with a short fast zoom out scene. (a) Frame 9. (b) Frame 17.

TABLE III

BITRATE REDUCTION OF BTZMCP COMPARED TO THE ORIGINAL KTA WITH THE SAME NUMBER OF B FRAMES WITHOUT BTZMCP

	BTZMCP IPPP	BTZMCP IBPBP	BTZMCP HB3	BTZMCP HB7	BTZMCP HB15
Foreman	−3.55%	−3.30%	−2.76%	−2.56%	−2.32%
Stefan	−2.96%	−2.39%	−7.74%	−9.89%	−8.76%
Mobile	−2.04%	−1.94%	−1.63%	−1.87%	−2.94%
Table	−4.39%	−4.19%	−5.22%	−5.41%	−3.37%
Average	−3.24%	−2.96%	−4.34%	−4.93%	−4.35%



(a)

(b)

Fig. 15. Two frames in *Table* sequence with a long fast zoom out scene. (a) Frame 57. (b) Frame 65.

by the original KTA reference software. The average bitrate reduction and average PSNR improvement are calculated using Bjøntegaard's method [26]. The results are verified by decoder implementation. The initial 100 frames from four CIF sequences (*Foreman, Stefan, Table, Mobile*) are used for initial analysis. Hierarchical GOP structures [27] with 0 to 15 B frames are used. Hierarchical B frames are used as it provides higher coding efficiency than non-hierarchical structure. Hierarchical structure can also provide temporal scalability. The number of B frames is related to the number of frames between I and P frames. In the experiments, functions like MRF, VBS and RDO are all turned on. Motion estimation's search window size is set to ±32 pixels, exhaustive search is used and quantization parameters (QPs) of 40, 36, 32, and 28 are used. Nine temporal reference frames are used in motion prediction of MRF unless specified. The proposed BTZMCP is applied to P slices only and implemented by the proposed subsampled block-matching on the subpixel motion estimation's interpolated reference frames with 1/8-subpixel accuracy ($S = s = 8$). The number of the zoom-in and zoom-out factors are both set to 2 ($N_1 = N_2 = 2$) and the set of zoom factors as shown in (13) is used.

To show the performance improvement of BTZMCP, the experimental results are divided into four parts. In the first part, the RD performance of BTZMCP is compared with that of the original KTA with the same number of B frames in hierarchical GOP structures. It reveals the situations in which BTZMCP gives the largest improvement. The second part shows the best possible RD performances with and without BTZMCP for analyzing the overall coding efficiency. The third part shows the impact on the coding efficiency of BTZMCP with the T-shaped motion model that only uses BTZMCP in the previous reference frame. The forth part is to evaluate the performance of BTZMCP for next generation video coding

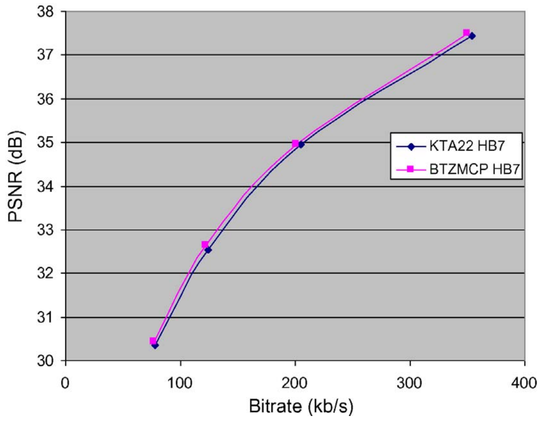


Fig. 16. RD curve comparison of *Foreman* sequence.

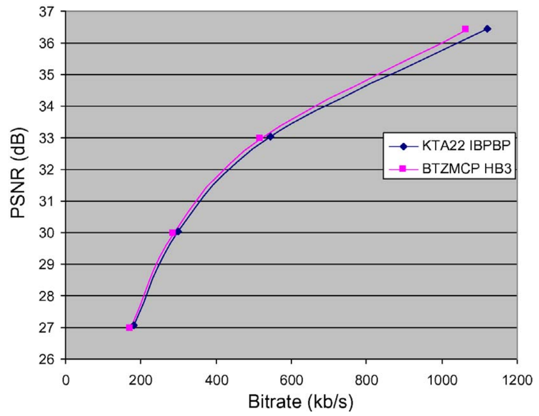


Fig. 17. RD curve comparison of *Stefan* sequence.

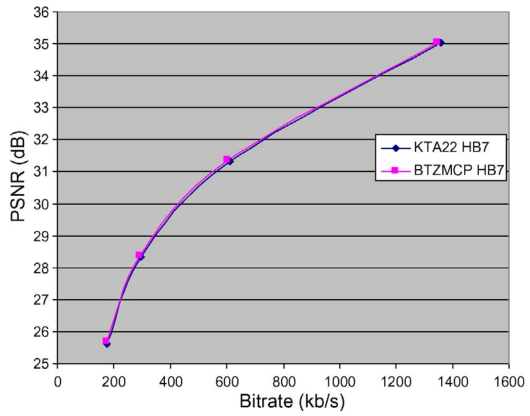


Fig. 18. RD curve comparison of *Mobile* sequence.

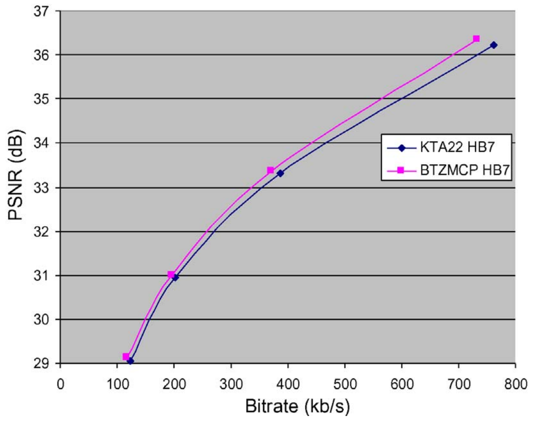


Fig. 19. RD curve comparison of *Table* sequence.

standard applications using MPEG HVC call for proposal (CfP) [28] testing conditions and high-resolution sequences.

A. RD Performance with Different GOP Structures

Table III shows the average bitrate reduction achieved by BTZMCP with different GOP structures—IPPP (no B frames), B1 (one B frame), HB3 (hierarchical GOP with 3 B frames), HB7 (hierarchical GOP with 7 B frames), and HB15 (hierarchical GOP with 15 B frames). Improvements are observed in all these cases—different sequences and different number of B frames. That means the benefit of providing more prediction candidates (zoom prediction) outweighs the overhead on representing the zoom parameters introduced. The best average improvement is 4.93% for 7 hierarchical B frames case. The average improvement is relatively higher for more B frames cases. In particular, the performance improvements on *Stefan* and *Table* sequences are significant as the average bitrate reductions are up to 9.89% and 5.41%, respectively. For the other two sequences, the bitrate reductions are up to 2.94% and 3.55%, respectively. These variations are due to the difference in motion contents. There is a short fast zooming in the *Stefan* sequence and a long fast zooming in the *Table* sequence. The zooming effects for the sequences are shown in Figs. 14 and 15.

However, the improvement as shown in Table III is lower than the initial analysis results as shown in Figs. 6 and 7. There are two main reasons.

1) The overhead for representing the zoom motion information are not considered in the initial analysis.
2) The zoom step size in the practical implementation of subsampled block matching is limited to 1/8 which is larger than step sizes (1/16, 1/32, 1/64, and 1/128) used in these initial analysis.

However, even with this relatively large zoom step size of 1/8, BTZMCP is still possible to achieve good RD improvement for sequences with zoom motion content.

B. RD Performances with and without BTZMCP

The numbers of B frames in hierarchical GOP structures with the best coding efficiency in original KTA implementation may not be the same as that of BTZMCP implementation. Table IV uses original KTA with IPPP GOP for comparing various number of B frames settings in hierarchical GOP structures for both KTA and BTZMCP. The shaded boxes in Table IV show the best configurations of original KTA and BTZMCP for each sequence. According to Table IV, *Foreman* and *Mobile* sequences have the best RD performance with 7 hierarchical B frames for both KTA and BTZMCP. *Stefan* and *Table* sequences have the best RD performance with one B frame for KTA and with three hierarchal B frames for BTZMCP, respectively. From Table IV, it is found that for sequences with higher motion contents, like *Stefan* and *Table*,

TABLE IV

Bitrate Reduction Percentages as Compared to Original KTA Software with GOP of IPPP

	KTA IPPP	KTA IBPBP	KTA HB3	KTA HB7	KTA HB15
Foreman	0.00%	−12.08%	−14.93%	−20.15%	−20.12%
Stefan	0.00%	−6.80%	−2.42%	1.89%	3.62%
Mobile	0.00%	−13.23%	−18.84%	−19.75%	−18.13%
Table	0.00%	−3.60%	−3.37%	−6.42%	−4.74%
Average	0.00%	−8.93%	−9.89%	−11.11%	−9.84%

	BTZMCP IPPP	BTZMCP IBPBP	BTZMCP HB3	BTZMCP HB7	BTZMCP HB15
Foreman	−3.55%	−15.00%	−17.26%	−22.17%	−21.93%
Stefan	−2.96%	−9.03%	−9.99%	−8.20%	−5.51%
Mobile	−2.04%	−14.91%	−20.16%	−21.25%	−20.56%
Table	−4.39%	−7.68%	−8.42%	−11.54%	−8.01%
Average	−3.24%	−11.65%	−13.96%	−15.79%	−14.00%

TABLE V

RD Comparisons Between Best Configurations of Different Number of B Frames

Foreman	KTA22 HB7		BTZMP HB7	
QP	Bitrate	PSNR	Bitrate	PSNR
28	354.16	37.44	349.91	37.48
32	204.94	34.94	201.28	34.96
36	123.58	32.55	122.50	32.65
40	77.84	30.35	76.78	30.44
Average bitrate reduction (%)				−2.56
Average PSNR improvement (dB)				0.12

Stefan	KTA22 IBPBP		BTZMP HB3	
QP	Bitrate	PSNR	Bitrate	PSNR
28	1121.28	36.45	1065.53	36.42
32	544.16	33.03	518.38	32.98
36	298.94	30.02	287.49	29.97
40	180.79	27.08	173.49	26.98
Average bitrate reduction (%)				−3.42
Average PSNR improvement (dB)				0.18

Mobile	KTA22 HB7		BTZMP HB7	
QP	Bitrate	PSNR	Bitrate	PSNR
28	1360.58	35.04	1346.05	35.03
32	611.38	31.33	603.24	31.35
36	296.34	28.32	293.81	28.38
40	176.56	25.63	174.07	25.70
Average bitrate reduction (%)				−1.87
Average PSNR improvement (dB)				0.09

Table	KTA22 HB7		BTZMP HB7	
QP	Bitrate	PSNR	Bitrate	PSNR
28	761.86	36.23	732.77	36.34
32	387.05	33.31	371.68	33.36
36	202.71	30.94	195.58	30.99
40	122.94	29.04	116.82	29.12
Average bitrate reduction (%)				−5.41
Average PSNR improvement (dB)				0.22

the performance becomes worse when using more than one B frames in KTA case. However, the improvement remains similar up to seven hierarchical B frames in BTZMCP case. This property is quite attractive that BTZMCP can achieve high coding efficiency while using more temporal layers for temporal scalability.

Table V lists the rate-distortion data for the best configuration for each sequence given in Table IV. The corresponding RD plots are given in Figs. 16–19. Table V shows that BTZMCP outperforms the best configuration of KTA with bitrate reduction of 1.87% to 5.41%. These RD performance improvements make the coding efficiency of BTZMCP the best among the existing video coding standards.

C. Performance on Using T-Shaped Motion Model

The T-shaped motion model that only applies BTZMCP on previous reference frame for reducing the complexity is also investigated. Table VI shows the average bitrate reduction achieved by BTZMCP with T-shaped motion model. Improvement is very similar to that of the original BTZMCP and the differences are less than 1% in most cases. That means most of the benefit from zooming can be provided by just using the previous reference frame. The best average improvement is 4.57% for 7 hierarchical B frames case. It is only 0.36% degraded from the original BTZMCP but it only requires searching 11% of the original zoomed frames in 9 reference frames configuration. For practical implementation, it is therefore suggested to use BTZMCP with the T-shaped motion model as it can greatly reduce the computational complexity. The computational requirement of BTZMCP can be further reduced by using advanced fast motion estimation algorithms. This will make the proposed techniques more practical.

D. Performance with High-Resolution Sequences

To further evaluate the performance of BTZMCP for next generation video coding standard application, MPEG HVC CfP [28] testing conditions and high-resolution sequences are used. Fifteen sequences with resolutions of 2560 × 1600, 1080p, WVGA and WQVGA are used to evaluate the proposed BTZMCP. The major differences of testing parameters are: 1) the number of hierarchical B frames is fixed to 7; 2) motion estimation's search window size is set to ±128 pixels; 3) the QPs are 22, 27, 32, 37; 4) four reference frames are used; and 5) fast motion estimation, weighted prediction and RDO quantization are turned on. The results are shown in Table VII with BTZMCP using T-shaped motion model. Similarly, RD performance improvement can be obtained in all these high-resolution sequences. The average improve-

TABLE VI

AVERAGE BITRATE REDUCTION OF BTZMCP WITH T-SHAPED MOTION MODEL COMPARED TO THE ORIGINAL KTA WITH THE SAME NUMBER OF B FRAMES

	BTZMCP (T) IPPP	BTZMCP (T) IBPBP	BTZMCP (T) HB3	BTZMCP (T) HB7	BTZMCP (T) HB15
Foreman	−3.08%	−2.87%	−2.52%	−2.61%	−2.14%
Stefan	−3.00%	−2.35%	−7.34%	−9.39%	−7.46%
Mobile	−2.15%	−1.70%	−1.50%	−1.73%	−1.42%
Table	−4.30%	−3.80%	−3.90%	−4.53%	−2.94%
Average	−3.13%	−2.68%	−3.81%	−4.57%	−3.49%

TABLE VII

RD PERFORMANCE COMPARISONS BETWEEN BTZMCP AND ORIGINAL KTA SOFTWARE WITH DISTRIBUTION OF ZOOM MOTION

Sequences		Average Bitrate Reduction (%)	Average PSNR Improvement (dB)	% of Macroblock Selected in BTZMCP	
				Non-Zoom	Zoom
Class A 2560 × 1600	*Traffic*	−4.63	0.18	96.07%	3.93%
	People	−5.92	0.29	94.99%	5.01%
	Average	−5.28	0.24		
Class B 1080p	*Kimono1*	−8.13	0.31	93.84%	6.16%
	ParkScene	−4.60	0.17	96.18%	3.82%
	Cactus	−4.72	0.12	97.60%	2.40%
	BasketballDrive	−9.34	0.26	96.69%	3.31%
	BQTerrace	−3.03	0.06	97.98%	2.02%
	Average	−5.97	0.19		
Class C WVGA 832 × 480	*BasketballDrill*	−4.56	0.20	99.26%	0.74%
	BQMall	−5.24	0.23	97.77%	2.23%
	PartyScene	−2.75	0.12	98.90%	1.10%
	RaceHorses	−7.37	0.32	95.71%	4.29%
	Average	−4.98	0.22		
Class D QWVGA 416 × 240	*BasketballPass*	−5.31	0.28	97.28%	2.72%
	BQSquare	−1.48	0.06	98.82%	1.18%
	BlowingBubbles	−2.72	0.12	97.49%	2.51%
	RaceHorses	−6.72	0.35	93.19%	6.81%
	Average	−4.06	0.20		

ment is around 4.06–5.97% for different resolutions and the improvement is relatively higher in HD and 2560 × 1600 sequences. In particular, the bitrate reduction is up to 9.34% in the HD sequence *BasketballDrive*. The percentages of the macroblocks that selected zoomed blocks are also shown in Table VII. Even the percentages are below 5%, these selected zoomed blocks are possible to provide high prediction accuracy improvement. For example, only 3.31% of zoomed blocks are used in *BasketballDrive* sequence but these zoomed blocks can provide 9.34% bitrate reduction. In addition, the RD performances as shown in Table VII are not optimal yet because motion vector prediction associated with BTZMCP is not yet investigated and that will affect the selections in the RDO process. The coding of zoom motion vector by multiple reference frame method is also not optimized for RD performance. Even with such premature implementation, BTZMCP can still achieve robust and promising results for large range of video resolutions and contents. Thus, BTZMCP and subsampled block-matching technique have high potential for next generation video coding standard applications.

E. Comparison with MCP Using More General Motion Models

To understand the potential of BTZMCP, indirect performance comparisons with affine multi-picture motion compensated prediction (AMMCP) [19] and elastic motion model [23] are also performed. From [19], AMMCP can achieve average bitrate reduction of 15% as compared to H.263 while 20 initial clusters are used. As reported in [23], elastic motion model can provide 0.5–1 dB PSNR improvements. Both of these two RD improvements are very significant. Based on these results, BTZMCP appears to be inferior to these two methods. However, it should be noted that BTZMCP only extend the motion model to include zoom motions. Both AMMCP and elastic motion model provide more general representation than just zooming to achieve better RD performance. Considering the computational complexity and memory requirement associated with these two methods, the subsampled block-matching of zoom content prediction used in BTZMCP is more practical and much easier to integrate into the current video coding standards. For example, both affine parameters estimation and wrapped frames storage are required in AMMCP and therefore significantly increase the complexity and memory requirement in the codec implementation. For elastic motion model, the elastic parameters are usually obtained by iteration. The partition in reference frame is wrapped and compared to the current image. It iterates until the error is smaller than a threshold or until the maximum number of iterations is reached. In BTZMCP, no parameter estimation is required and the implementation is very similar to multiple reference frames technique by using subsampled block-matching. It is also very easy to integrate BTZMCP into hybrid coding structure for next generation video coding standard applications.

On the other hand, the proposed subsampled block-matching technique can be further extended to achieve affine and elastic motions by using other subsampling grid patterns. For example, stretching and compression can be achieved by changing the subsampling rate only on one axis. With an appropriate set of affine or elastic parameters, subsampled block-matching will be a practical way to implement these motion models without requiring the calculation of the wrapped frames or partitions. Basically, BTZMCP is just the first attempt to demonstrate the subsampled block-matching technique on zoom motion compensated prediction application. It could be used to solve the practical implementation problems of conventional motion compensation prediction algorithms with higher motion order.

V. Conclusion

Conventional motion compensated prediction utilizes prior frames as references for generating prediction. It performs well when the object motion in the scene of video sequence is limited to translational motion on the imaging plane. However, such assumption is not always valid in reality. To better handle more realistic motions such as zooming effect due to object moving close to or away from the camera, a new block-matching-based translation and zoom motion compensated prediction (BTZMCP) technique was proposed. This zoom motion compensation technique used temporal reference frames to generate additional reference frames for zoom motion prediction. A simple and efficient subsampled block-matching implementation was also proposed. It performed zoom motion estimation on the interpolated temporal reference frames of the subpixel motion estimation. Thus, no significant extra storage was required in both encoder and decoder implementation. In addition, the new zoom motion compensation technique could be easily integrated into the modern video codec to further improve the compression efficiency. The rate-distortion performance of BTZMCP was evaluated by implementation in KTA reference software. Different hierarchical B frame GOP structure was used for evaluation. Experimental results showed that bitrate reduction up to 9.89% can be achieved. It was also found that BTZMCP performs especially well in video sequences with fast zooming like sports contents. For those sequences without obvious zoom motion contents, BTZMCP can still provide certain improvement. Thus, zoom motion based MCP should be a promising direction in video coding. In addition, the proposed subsampled block-matching on interpolated reference frames technique is also feasible for realizing other motion models such as rotation, shear, elastic, and so on, using specially designed subsampling patterns in block-matching.

References

[1] *Video Codec for Audiovisual Services at p × 64 kbit/s*, ITU-T Rec. H.261 Version 2, ITU-T_SG15, Mar. 1993.

[2] *Video Coding for Low Bitrate Communication*, ITU-T Rec. H.263 Version 2, ITU-T_SG16, Feb. 1998.

[3] *Information Technology: Coding of Moving Pictures and Associated Audio for Digital Storage Media up to About 1.5 Mbit/s, Part 2: Video*, ISO/IEC 11172-2 (MPEG-1 Video), JTC1/SC29/WG11, 1993.

[4] *Generic Coding of Moving Pictures and Associated Audio Information, Part 2: Video*, ITU-T Rec. H.262, ISO/IEC 13818-2 (MPEG-2), ITU-T_and_ISO/IEC_JTC-1, 1995.

[5] C. Cafforio and F. Rocca, "Methods for measuring small displacements of television images," *IEEE Trans. Inform. Theory*, vol. 22, no. 5, pp. 573–579, Sep. 1976.

[6] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, no. 12, pp. 1799–1808, Dec. 1981.

[7] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, Nov.–Dec. 1981, pp. 1–5.

[8] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. 33, no. 8, pp. 888–896, Aug. 1985.

[9] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[10] Z. B. Chen, P. Zhou, and Y. He, *Fast Integer Pel and Fractional Pel Motion Estimation for JVT*, Document JVT-F017, Joint Video Team Meeting, Awaji Island, Japan, Dec. 5–13, 2002.

[11] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Proc. SPIE Vis. Commun. Image Process.*, 2002, pp. 1069–1079.

[12] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 604–612, Apr. 1993.

[13] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, Feb. 1999.

[14] G. J. Sullivan and R. L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in *Proc. Global Telecommun. Conf.*, 1991, pp. 85–90.

[15] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[16] *Draft Meeting Report VCEG-AE01 for Marrakech Meeting*, ITU-T_Q.6/16_VCEG, 31st Meeting, Marrakech, Morroco, 2007.

[17] *Vision and Requirements for High-Performance Video Coding (HVC) Codec*, MPEG Output document N10361, ISO/IEC_JTC1, Lausanne, Switzerland, Feb. 2009.

[18] H. Jozawa, K. Kamikura, A. Sagata, H. Kotera, and H. Watanabe, "Two-stage motion compensation using adaptive global MC and local affine MC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 75–85, Feb. 1997.

[19] T. Wiegand, E. Steinbach, and B. Girod, "Affine multipicture motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 197–209, Feb. 2005.

[20] R. C. Kordasiewicz, M. D. Gallant, and S. Shirani, "Affine motion prediction based on translational motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1388–1394, Oct. 2007.

[21] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 3, pp. 339–356, 366–367, Jun. 1994.

[22] M. Karczewicz, J. Nieweglowski, and P. Haavisto, "Video coding using motion compensation with polynomial motion vector fields," *Signal Process.: Image Commun.*, vol. 10, nos. 1–3, pp. 63–91, Jul. 1997.

[23] M. R. Pickering, M. R. Frater, and J. F. Arnold, "Enhanced motion compensation using elastic image registration," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 1061–1064.

[24] *Key Technical Area (KTA) Reference Model 2.2r1* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/KTA

[25] *Joint Video Team (JVT) Reference Software Version 11.0* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/old_jm

[26] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, Document VCEG-M33, ITU-T Video Coding Expert Group Meeting, Austin, TX, Apr. 2001.

[27] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 1929–1932.

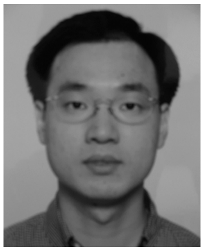
[28] *Call for Proposals on High-Performance Video Coding (HVC)*, MPEG document N10926, ISO/IEC JTC1/SC29/WG11, Oct. 2009.

Lai-Man Po (M'92–SM'09) received the B.S. and Ph.D. degrees in electronic engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 1988 and 1991, respectively.

Since 1991, he has been with the Department of Electronic Engineering, City University of Hong Kong, and is currently an Associate Professor. His current research interests include vector quantization, motion estimation for video compression, and H.264/AVC fast mode decision algorithms.

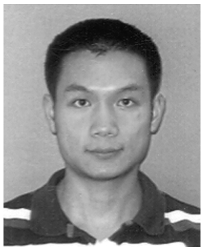
Ka-Man Wong (S'08) received the B.E. degree in computer engineering and the M.Phil. degree in electronic engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2001 and 2004, respectively. He is currently pursuing the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong.

His current research interests include motion estimation for video compression and content-based image retrieval.



Kwok-Wai Cheung (M'10) received the B.E., M.S., and Ph.D. degrees, all in electronic engineering, from the City University of Hong Kong, Kowloon, Hong Kong, in 1990, 1994, and 2001, respectively.

He was with Hong Kong Telecom, Hong Kong, as a Trainee Engineer/Project Engineer from 1990 to 1995. He was a Research Student/Research Assistant with the Department of Electronic Engineering, City University of Hong Kong, from 1996 to 2002. He joined the Chu Hai College of Higher Education, Tsuen Wan, Hong Kong, in 2002. Currently, he is an Associate Professor with the Department of Computer Science, Chu Hai College of Higher Education. His current research interests include the areas of image/video coding and multimedia database.



Ka-Ho Ng (S'10) received the B.E. and M.Phil. degrees, both in electronic engineering, from the City University of Hong Kong, Kowloon, Hong Kong, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong.

His current research interests include video coding and motion estimation.