Block-Wise Adaptive Motion Accuracy Based B-Picture Coding With Low-Complexity Motion Compensation

Xiangyang Ji, Debin Zhao, and Wen Gao

***Abstract*—This paper presents a novel B-picture coding based on block-wise adaptive motion accuracy (BAMA) with low-complexity motion compensation (MC). BAMA is able to adaptively select the motion accuracy for each inter-block in a B-picture depending on whether it is of bidirectional prediction, in which compared to the unidirectionally predicted block, lower motion accuracy is applied to the bidirectionally predicted block to reduce MC complexity. To further reduce MC complexity for the bidirectionally predicted luminance block with subpixel motion vectors in both directions, the forward and backward MC interpolations are merged according to the linear operation property of MC interpolation. In addition, a simplified MC interpolation method is also introduced for chrominance components for the bidirectionally predicted block. When integrating the proposed methods into H.264/AVC reference software, the experimental results demonstrate that they can significantly reduce MC complexity for a B-picture while yielding the comparable rate distortion performance in comparison with H.264/AVC although they no longer generate H.264/AVC compliant bitstream.**

***Index Terms*—Block-wise adaptive motion accuracy (BAMA), B-picture, MC interpolation, video coding.**

I. Introduction

In many video applications, compression efficiency is always constrained because higher compression efficiency comes with stronger decoding capability requirement. In general, different profiles and levels are defined to support different application requirements in video coding standards like MPEG-x and H.26x. Baseline profile usually supports applications with low complexity and low latency requirements such as video conference, video telephony and wireless transmission [1] and hence usually only contains I-and P-pictures. Compared with P-picture coding only with forward prediction, B-picture coding can use both forward and backward pictures as references and thus provide higher compression performance. Furthermore, B-pictures can also be dropped in some applications and do not propagate errors if they are not used as references. However, B-picture decoding usually requires higher decoding complexity and also results in latency. Therefore, it is only included in the higher profiles that allow high decoding complexity and latency.

The bidirectionally predicted block usually requires more motion compensated (MC) interpolation complexity than the unidirectionally predicted one in B-picture decoding. In particular, both forward and backward MC interpolations are required if motion vectors in both directions are of subpixel accuracy. This always increases computational complexity and memory access frequency. As stated in [2], one inserted B-picture between adjacent I/P-pictures requires an extra 50% cost for the very low bit rate video decoding, 20%–35% for medium and high bit rate video decoding. More inserted B-pictures will further increase the decoding time.

On the other hand, motion vector accuracy is usually fixed during encoding the whole video sequence in video coding standards. For each inter-macroblock, adaptive motion accuracy (AMA) proposed in [3] tried to use Lagrangian criterion to select the best motion vector accuracy in terms of rate distortion optimization and a flag is needed to be signaled to tell the decoder what motion vector accuracy is used. At the early stage of H.264/AVC evolution, AMA was adopted in TML-2 with $1/3$-pixel accurate motion-compensated prediction using a 4-tap filter in both horizontal and vertical directions. When MC interpolation was modified (in TML-4) to $1/4$-pixel accurate prediction, AMA was dropped due to the lack of coding efficiency improvement [1].

This paper presents a block-wise adaptive motion accuracy (BAMA) in B-picture coding. In BAMA, the bidirectionally predicted block employs lower motion vector accuracy than the unidirectionally predicted one to reduce MC interpolation complexity while keeping strong temporal decorrelation capability. To further reduce the MC interpolation complexity for bidirectionally predicted luminance block with subpixel accuracy in both directions, the forward and backward MC interpolations are merged according to the linear operation property of MC interpolation. For chrominance components, a simplified MC interpolation method is also introduced. By changing the corresponding motion vectors representation and MC interpolation operation at both encoder and decoder, the proposed methods can be easily integrated into H.264/AVC, in which the prediction signals for the bipredictive block are achieved from an arbitrary set of reference pictures in the forward and/or backward predictions, though the generated bitstream is no longer compliant with H.264/AVC.

The remaining sections of this paper are organized as follows. Section II describes the BAMA algorithm in the B-picture coding. In Section III, low-complexity MC interpolation according to the linear operation property of interpolation is presented firstly for bidirectionally predicted luminance block and then another simplified MC interpolation is introduced for the chrominance components. Section IV provides the experimental results in terms of rate distortion performance and MC interpolation complexity comparisons. Finally, the conclusions are given in the last section.

Manuscript received April 28, 2006; revised March 22, 2007. This work was supported in part by the National Science Foundation of China under Grants 60672088 and 60333020. This paper was recommended by Associate Editor J. Boyce.

X. Ji and W. Gao are with the Institute of Computing Technology and the Graduate School, Chinese Academy of Sciences, Beijing 100085 China (e-mail: xyji@jdl.ac.cn; wgao@jdl.ac.cn).

D. Zhao is with the Department of Computer Science, Harbin Institute of Technology, Harbin 15001, China (e-mail: dbzhao@vilab.hit.edu.cn).

Digital Object Identifier 10.1109/TCSVT.2007.903315

II. Block-Wise Adaptive Motion Accuracy

In comparison with unidirectionally predicted blocks, on many occasions, the motion compensated prediction signals with lower motion accuracy may be accurate enough for most bidirectionally predicted blocks. Based on this observation, BAMA is proposed for inter-block coding to reduce MC interpolation complexity. Different from AMA, BAMA is able to adaptively select motion accuracy for each inter-block in a B-picture depending on whether it is of bidirectional prediction. For simplicity of our following description, motion vector allowed up to half-pixel and quarter-pixel accuracies are used for the bidirectionally and unidirectionally predicted blocks, respectively. All motion vectors are still represented in quarter-pixel unit but they need to be scaled down for motion vectors coding in the bidirectionally predicted blocks.

Let $F(\vec{n})$ denote the image with space-discrete coordinate $\vec{n}=(n_x,n_y)$ and its corresponding reconstructed one is $\tilde{F}(\vec{n})$. In the hybrid video coding standards, the whole image$F(\vec{n})$ is usually divided into a number of macroblocks S and each S consists of one 16×16 luminance and two associated chrominance components. An inter-macroblock can be further segmented into smaller inter-blocks with variable block sizes, in which each block has its own motion vector $\vec{v}=(v_x,v_y)$ with the horizontal component v_x and vertical component v_y, respectively. For the unidirectional motion estimation of an inter-block, the rate-constrained motion estimation for a given block S_i in a macroblock S is able to be performed by minimizing the Lagrangian cost function [4]

$$J(\lambda_{\mathrm{ME}},\vec{v},r)=D_{\mathrm{SAD}}(S_i,\tilde{F}_r,\vec{v})+\lambda_{\mathrm{ME}}\times R(r,\vec{v}-\vec{p}) \quad (1)$$

where λ_{ME} is the Lagrange multiplier, r and $\vec{p}$ are the reference index and predicted motion vector, respectively. $R(r,\vec{v}-\vec{p})$is the number of bits coding motion vector and reference index. The distortion term D_{SAD} is measured as the sum of absolute differences (SAD) between current block S_i and the reconstructed reference picture $\tilde{F}_r$ using

$$D_{\mathrm{SAD}}(S_i,\tilde{F}_r,\vec{v}) = \sum_{(n_x,n_y)\in S_i} |F(n_x,n_y)-\tilde{F}_r(n_x+v_x,n_y+v_y)|. \quad (2)$$

When BAMA is used, the number of bits to transmit motion vector in each prediction direction for a bidirectionally predicted block should be calculated from$(\vec{v}-\vec{p}+1)\gg 1$. The final cost for a bidirectionally predicted block is achieved by

$$J(\lambda_{\mathrm{ME}},\vec{v}_f,\vec{v}_b,r_f,r_b) = D_{\mathrm{SAD}}(S_i,\tilde{F}_f,\tilde{F}_b,\vec{v}_f,\vec{v}_b)+\lambda_{\mathrm{ME}}\times R(r_f,r_b,(\vec{v}_f-\vec{p}_f+1)\gg 1,(\vec{v}_b-\vec{p}_b+1)\gg 1) \quad (3)$$

where the subscripts f and b denote the forward and backward prediction directions, respectively. The distortion term D_{SAD} is equal to the SAD between current block S_i and its prediction signals achieved by the linear combination of prediction blocks from the forward reference $\tilde{F}_f$ and backward reference $\tilde{F}_b$. Finally, one type of prediction direction with the minimum distortion among forward, backward and bidirectional prediction directions will be selected for the current inter-block. It should be noted that if the effect in the rate calculation term between$(\vec{v}-\vec{p}+1)\gg 1$ and $(\vec{v}-\vec{p})$ is negligible, the forward and backward motion vectors of the the bidirectionally predicted block may be taken directly from independently estimated ones to reduce motion estimation complexity. If joint bidirectional motion estimation in [5] is applied, the criteria described in (3) can be used during the motion estimation process for the bidirectionally predicted block.

Regarding to motion vector coding of the bidirectionally predicted block, the differential motion vector between the estimated and predicted motion vector needs to be scaled down by a factor of 2 correspondingly. At decoder, to achieve the motion vectors of the bidirectionally predicted block in quarter-pixel unit, the differential motion vector of bidirectionally predicted block needs to be magnified with a factor of 2 and then added to the corresponding predicted motion vector, which needs to be rounded to the nearest half-pixel position in quarter-pixel unit towards minus infinity.

III. Low-Complexity MC Interpolation in BAMA

A. Low-Complexity MC Interpolation Process

In a B-picture, the prediction signals for a bidirectionally predicted block are achieved by a linear combination of forward and backward prediction signals as follows:

$$\begin{aligned}P_{\mathrm{bi}}(\vec{n}) &= w_f\times P_f(\vec{n})+w_b\times P_b(\vec{n})\\ &= w_f\times\tilde{F}_f(\vec{n}+\vec{v}_f)+w_b\times\tilde{F}_b(\vec{n}+\vec{v}_b)\end{aligned} \quad (4)$$

where w_f and $w_b$are the relative weights of the forward prediction block P_f and the backward prediction one P_b, respectively. Usually, $w_f+w_b=1$ and $w_f=w_b=1/2$. More generally, w_f and w_b may be any real number between 0 and 1 [6]. If the forward and/or backward motion vectors are of subpixel accuracy, the MC interpolations are applied to the corresponding references to yield the motion-compensated prediction signals accordingly. When a finite impulse response (FIR) filter h with $2M$-tap is used for the 2-D separate MC interpolation, the prediction signals with motion vectors of horizontally, vertically and diagonally half-pixel accuracy in each prediction direction can be yielded by

$$P(\vec{n})=\sum_{u=-M+1}^{M} h(u)\tilde{F}_r(n_x+\lfloor v_x\rfloor+u,n_y+\lfloor v_y\rfloor) \quad (5)$$

$$P(\vec{n})=\sum_{u=-M+1}^{M} h(u)\tilde{F}_r(n_x+\lfloor v_x\rfloor,n_y+\lfloor v_y\rfloor+u) \quad (6)$$

and

$$P(\vec{n})=\sum_{u_1=-M+1}^{M} h(u_1)\left(\sum_{u_0=-M+1}^{M} h(u_0)\times\tilde{F}_r(n_x+\lfloor v_x\rfloor + u_0,n_y+\lfloor v_y\rfloor+u_1)\right) \quad (7)$$

where $\lfloor\rfloor$ represents the operation rounded to the nearest full-pixel position toward minus infinity and $h(u)$ represents the tap

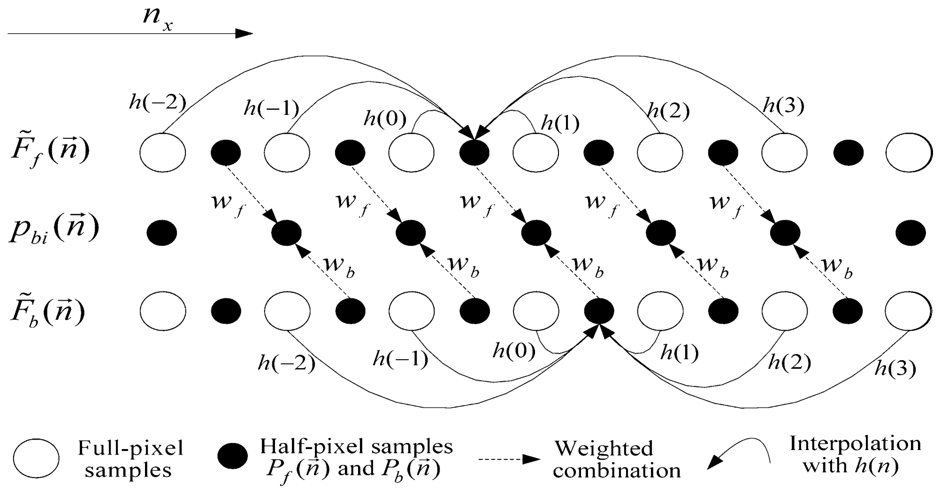

Fig. 1. The normal MC interpolations for bidirectionally predicted block when $\vec{v}_f = (-(1)/(2), 0)$ and $\vec{v}_b = ((1)/(2), 0)$.

coefficient. The prediction values at the horizontally and vertically half-pixel positions are obtained by applying a one-dimensional $2M$-tap FIR filter horizontally and vertically using (5) and (6), respectively. For the diagonally half-pixel position, one-dimensional 2M-tap FIR filter needs to be performed horizontally firstly and then vertically using (7). Fig. 1 illustrates the normal prediction block generation process with MC interpolations for the bidirectionally predicted block with horizontally half-pixel accuracy in both directions when a FIR filter h with 6-tap is used. Firstly, the corresponding interpolations are used to generate the forward and backward prediction blocks $P_f(\vec{n})$ and $P_b(\vec{n})$ using (5), respectively. And then, the prediction block can be yielded by (4).

For BAMA, MC interpolation for the bidirectionally predicted block with half-pixel accuracy in both directions can be simplified according to the linear operation property of interpolation. Firstly, the new full-pixel position samples $S^{/}(\vec{n})$are achieved by the linear combination of full pixels in the forward and backward references, which have the corresponding motion vectors rounded to full-pixel position towards minus infinity. The corresponding equation calculating $S^{/}(\vec{n})$ can be described as follows:

$$S'(\vec{n}) = w_f \times \vec{F}_f(n_x + \lfloor v_{x,f} \rfloor, n_y + \lfloor v_{y,f} \rfloor) + w_b \times \vec{F}_b(n_x + \lfloor v_{x,b} \rfloor, n_y + \lfloor v_{y,b} \rfloor). \quad (8)$$

And then, MC will be performed on the new full-pixel position sample $S'(\vec{n})$ to yield the final prediction signals for the bidirectionally predicted block with half-pixel accuracy in both directions. Fig. 2 illustrates the generation process for $S'(\vec{n})$ when the motion vectors for the bidirectionally predicted block have horizontally half-pixel accuracy in both directions. Compared with the normal MC interpolation as shown in Fig. 1, the forward and backward interpolations can be merged. It can be observed that the final prediction signals yielded from the normal MC interpolation and low-complexity MC interpolation are the same regardless of the round operation during the MC interpolation calculation. According to this observation, MC interpolation operation for the bidirectionally predicted block with BAMA can be categorized into the following five cases.

1) When the forward motion vector is of vertically half-pixel accuracy and the backward motion vector is of horizontally half-pixel accuracy or *vice versa* for the bidirectionally predicted block, the normal prediction block generation process can be used.
2) When both forward and backward motion vectors are of only horizontally half-pixel accuracy for bidirectionally predicted block, MC interpolation can be calculated by

$$\begin{aligned} p_{\mathrm{bi}}(\vec{n}) = & \sum_{u=-M+1}^{M} h(u) \\ & \times [w_f \times \vec{F}_f(n_x + \lfloor v_{x,f} \rfloor + u, n_y + \lfloor v_{y,f} \rfloor) \\ & + w_b \times \vec{F}_b(n_x + \lfloor v_{x,b} \rfloor + u, n_y + \lfloor v_{y,b} \rfloor)] \\ = & \sum_{u=-M+1}^{M} h(u) \times S'(n_x + u, n_y). \end{aligned} \quad (9)$$

3) When both forward and backward motion vectors are of only vertically half-pixel accuracy for bidirectionally predicted block, MC interpolation can be calculated by

$$p_{\mathrm{bi}}(\vec{n}) = \sum_{u=-M+1}^{M} h(u) \times S'(n_x, n_y + u). \quad (10)$$

4) When both forward and backward motion vectors are of diagonally half-pixel accuracy for bidirectionally predicted block, MC interpolation can be calculated by

$$p_{\mathrm{bi}}(\vec{n}) = \sum_{u_1=-M+1}^{M} h(u_1) \times \left(\sum_{u_0=-M+1}^{M} h(u_0) \times S'(n_x + u_0, n_y + u_1) \right). \quad (11)$$

5) When the forward or backward motion vector is of diagonally half-pixel accuracy and motion vector in the opposite direction is of only horizontally or vertically half-pixel accuracy, the horizontally or vertically half-pixel position sample with the diagonally half-pixel motion vector is first generated, and then (9) or (10) can

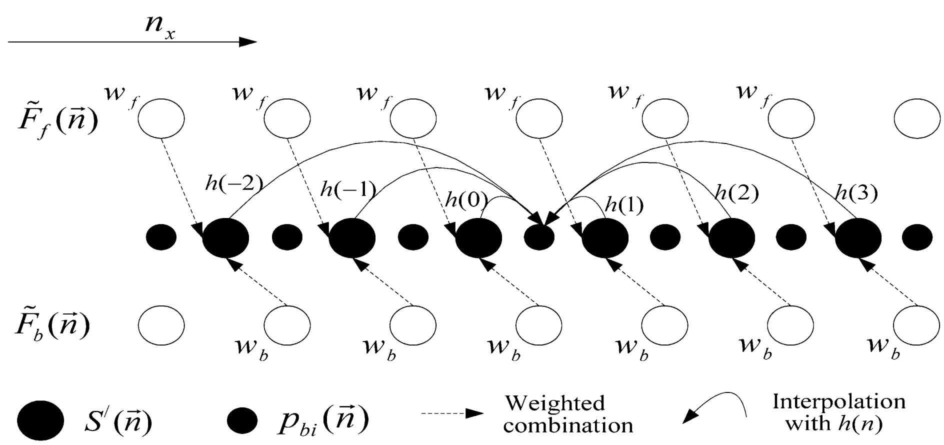

Fig. 2. Merging forward and backward MC interpolations for bidirectionally predicted block when $\vec{v}_f = (-(1)/(2), 0)$ and $\vec{v}_b = ((1)/(2), 0)$.

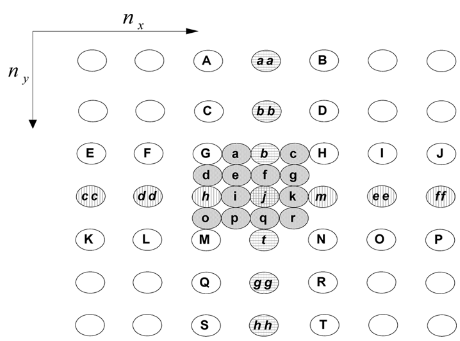

Fig. 3. Sample grid with quarter-pixel MC interpolation accuracy for signals (upper-case letters indicate samples on the full-pixel positions, lower-case italic letters indicate samples on the half-pixel positions and the remaining low-case letters indicate samples on the quarter-pixel positions).

be used for further MC interpolation. For example, if the forward motion vector is of diagonally half-pixel accuracy at position "j" in Fig. 3 and the backward motion vector is of horizontally half-pixel accuracy at position "b." Firstly, the vertically half-pixel position samples including "cc," "dd," "h," "m," "ee," and "ff" will be interpolated, and then, together with full-position samples "E," "F," "G," "H," "I," and "J," (9) will be used to yield the final prediction signals.

B. Simplified MC Interpolation for Chrominance

For video data in *YCbCr* color space with 4:2:0 sampling, the chrominance components (Cb and Cr) have one fourth of the number of luminance samples (Y). If the quarter-pixel accuracy is allowed for luminance, motion accuracy allowed up to one-eighth pixel should be used for chrominance. Since the 2-D separate MC interpolation allowed up to one-eighth pixel accuracy is hard to calculate the subpixel position sample due to taking many conditional branches into account, a simple bilinear interpolation is used in H.264/AVC for chrominance [1]. However, since coefficients of the filter-tap vary with the different interpolation positions and need to be calculated on-line rather than using the fixed-tap coefficients like the luminance component, the bilinear interpolation possibly results in nearly the same computational time per sample for chrominance as for luminance [7]. To further reduce MC interpolation for chrominance, a simplified interpolation method was proposed in [8], in which motion vectors for chrominance are represented in quarter-pixel units by motion vectors downscaling. When BAMA is used, the motion vector for chrominance can be further downscaled to half-pixel accuracy for the bidirectionally predicted block and thus, is able to further reduce the interpolation complexity of chrominance components for the bidirectionally predicted block while maintaining the comparable objective quality.

IV. Experimental Results and Analysis

In order to evaluate the rate distortion performance and computational complexity, the proposed methods were integrated into H.264/AVC reference software JM10.1 [9]. It should be pointed out that after the modifications of the H.264/AVC reference software the generated bitstream is no longer compatible with H.264/AVC. The test sequences include *bus, news, foreman and tempete* in QCIF@30 Hz and CIF@30 Hz. The first picture of sequence is intra-coded and the remaining pictures are inter-coded. The search range of ± 32, five reference pictures and CABAC entropy coder are used. The RD optimization based mode decision is applied and quantization values include 24, 28, 32, and 36.

A. Rate Distortion Performance

To give the overall rate distortion performance comparisons of the proposed methods (BAMA_LC) versus H.264/AVC (ORG), we employed Bjontegaard delta PSNR (BDPSNR) as described in [10] to provide the average PSNR difference between the RD curves derived from BAMA_LC and ORG, respectively. The detailed results are depicted in Table I. For IBPBP... and IBBBPBBBP... structures, different testing parameters are used, in which *One-ref and Multi-refs* indicate that one and five *list0* references is used for B- and P-picture coding and one *list1* reference is used for B-picture coding. Compared with anchor H.264/AVC, BAMA_LC is able to yield the comparable overall compression efficiency. And meanwhile, no perceptual loss is observed. Furthermore, there is also no compression efficiency loss even if the hierarchical B-picture coding is used.

TABLE I
AVERAGE PSNR DIFFERENCE OF BAMA *LC* VERSUS ORG IN TERMS OF BDPSNR

		Luminance				Chrominance_U				Chrominance_V			
Video sequences		news	foreman	tempete	bus	news	foreman	tempete	bus	news	foreman	tempete	bus
QCIF	One-ref IBPBP...	0.023	-0.007	0.149	0.074	-0.039	-0.004	0.119	0.004	0.032	-0.014	0.078	0.036
	One-ref IBBPBBP...	0.040	-0.020	-0.100	-0.028	-0.065	0.032	-0.006	-0.011	0.036	0.017	-0.038	0.012
	Multi-refs IBPBP...	0.037	-0.008	0.086	0.051	-0.049	-0.002	0.077	-0.068	0.046	-0.006	0.047	0.015
	Multi-refs IBBPBBP...	0.035	-0.023	-0.076	-0.032	-0.078	0.023	0.014	-0.035	0.035	-0.009	-0.024	0.053
CIF	One-ref IBPBP...	0.024	-0.017	0.017	0.005	0.051	0.035	0.028	-0.037	0.035	0.041	0.018	0.038
	One-ref IBBPBBP...	0.019	-0.021	-0.066	-0.050	0.005	0.010	-0.005	-0.045	0.013	-0.004	-0.008	-0.007
	Multi-refs IBPBP...	0.035	-0.020	-0.027	-0.004	0.056	0.021	0.016	-0.042	0.029	0.031	-0.002	0.066
	Multi-refs IBBPBBP...	0.020	-0.014	-0.097	-0.053	0.007	0.017	-0.011	-0.052	0.018	-0.006	-0.019	-0.018

TABLE II
NUMBER OF THE BASIC SUBPIXEL MC INTERPOLATION OPERATIONS INCLUDING MULTIPLICATION, ADDITION, AND SHIFT

	Multi	Add	Shift
'b','h'	2	6	1
'j'	6.5	17.25	1
'a','c','d','o'	2	8	2
'i','f','k','q'	6.5	20.25	3
'e','g','p','r'	4	14	3

B. MC Interpolation Complexity Comparisons and Analysis

In general, the computational complexity for chrominance interpolation is lower than that of luminance and thus plays a less important role on the overall computational complexity for 4:2:0 video decoding. In this subsection, we focus on MC interpolation complexity analysis for the luminance component. In H.264/AVC, a 6-tap FIR filter (1, −5, 20, 20, −5, 1)/32 is used for half-pixel luminance sample interpolation. During evaluating MC interpolation complexity for luminance, a block-based implementation with the 2-D separate filtering in [11] is used for the diagonally half-pixel interpolation. Table II gives the number of the basic MC interpolation operations including multiplication, addition and shift at different subpixel positions. Compared with ORG, the extra average operation for pixels outside the block is needed in (8) when motion vectors in both directions are of half-pixel accuracy for the bidirectionally predicted block. For example, to generate the half-pixel position sample "b" for Case 2 in Section III-A, the normal interpolation operations to generate samples b_f and b_b in both forward and backward directions can be implemented as follows:

$$b_f = [(G_f + H_f) \times 20 - (F_f + I_f) \times 5 + (E_f + J_f) + 16] \gg 5 \tag{12}$$

and

$$b_b = [(G_b + H_b) \times 20 - (F_b + I_b) \times 5 + (E_b + J_b) + 16] \gg 5. \tag{13}$$

When w_f and w_b are both equal to $1/2$, the final prediction sample is calculated according to

$$b_{\rm bi} = (b_f + b_b + 1) \gg 1. \tag{14}$$

When low-complexity interpolation method in Subsection III.A is used, the half-pixel position sample "b" for Case 2 can be generated as follows:

$$b_{\rm bi} = [(G_{\rm bi} + H_{\rm bi}) \times 20 - (F_{\rm bi} + I_{\rm bi}) \times 5 + (E_{\rm bi} + J_{\rm bi}) + 32] \gg 6 \tag{15}$$

where the samples $G_{\rm bi}, H_{\rm bi}, F_{\rm bi}, I_{\rm bi}, E_{\rm bi}, J_{\rm bi}$ are calculated by the intermediate result $S'(x,y)$ in (8). Here, it should be noted that the relative weight 1/2 of the forward and backward prediction blocks have been moved to the later MC interpolation part to further reduce computational complexity. To calculate the intermediate result $S'(x,y)$, $(B\times(B+5))/(B\times B)$times addition operations for Cases 2, 3 and 5 and $((B+5)\times(B+5))/(B\times B)$ times addition operations for Case 4 are required for each pixel on average when MC interpolation is implemented based on the $B \times B$ block size. As MC block size increasing, the number of addition operations for each pixel on average to calculate the intermediate result $S'(x,y)$ is decreased. For simplicity, MC interpolations are all implemented based on the 4×4 block. And thus, the basic MC interpolation operation numbers for different cases are listed in Table III when the low complexity interpolation technique in Section III-A is used or not. It should be noted that w_f and w_b are both set to 1/2 in above discussion. If w_f and w_b are not both equal to 1/2, the extra multiplication operations are also required.

For comparisons, Org_B and Org_P are used to indicate the number of basic MC interpolation operations for luminance per B- and P-picture on average in H.264/AVC. BAMA_LC_B and BAMA_B are used to indicate the corresponding ones when BAMA_LC and only BAMA described in Section II are used. Table IV gives the percentage for the basic MC interpolation operations when compared to ORG_B. For all test sequences, it can be observed that BAMA_B is able to significantly reduce the number of the basic MC interpolation operations per B-picture on average compared with ORG_B. Compared with BAMA_B, BAMA_LC_B is able to further reduce the number of the basic MC interpolation operations. It may result in the comparable complexity with ORG_P in most cases.

V. CONCLUSION

This paper firstly presents the BAMA B-picture coding, in which the bidirectionally predicted block employs lower motion vector accuracy than the unidirectionally predicted one. For the

TABLE III
NUMBER OF THE BASIC MC INTERPOLATION OPERATIONS FOR THE BIDIRECTIONALLY PREDICTED SAMPLE WITH HALF-PIXEL ACCURACY IN BOTH PREDICTION DIRECTIONS

	BAMA_LC			ORG		
	Multi	Add	Shift	Multi	Add	Shift
case 1	4	14	3	4	14	3
case 2	2	8.25	1	4	14	3
case 3	2	8.25	1	4	14	3
case 4	6.5	22.3125	1	13	36.5	3
case 5	6.5	19.5	1	8.5	25.25	3

TABLE IV
PERCENTAGE OF THE BASIC MC INTERPOLATION OPERATIONS WHEN COMPARED TO ORG_B FOR DIFFERENT CODING PARAMETERS

				Bus			News			Foreman			Tempete		
				ORG_P	BAMA_B	BAMA_LC_B	ORG_P	BAMA_B	BAMA_LC_B	ORG_P	BAMA_B	BAMA_LC_B	ORG_P	BAMA_B	BAMA_LC_B
QCIF	One-ref IBPBP...	Multi	Avg	63%	68%	48%	113%	53%	41%	68%	65%	50%	102%	42%	31%
		Add	Avg	59%	65%	50%	64%	70%	65%	61%	61%	51%	82%	45%	37%
		Shift	Avg	53%	58%	39%	39%	79%	73%	51%	55%	42%	61%	48%	38%
	One-ref IBBPBBP...	Multi	Avg	67%	65%	59%	98%	55%	47%	69%	68%	61%	83%	54%	50%
		Add	Avg	62%	62%	57%	62%	68%	64%	63%	64%	59%	73%	54%	51%
		Shift	Avg	55%	56%	50%	40%	75%	71%	53%	57%	51%	60%	53%	49%
	Multi-refs IBPBP...	Multi	Avg	60%	65%	47%	103%	47%	38%	65%	61%	48%	79%	44%	38%
		Add	Avg	56%	62%	49%	60%	66%	62%	58%	58%	49%	67%	44%	40%
		Shift	Avg	51%	55%	39%	37%	76%	72%	49%	52%	40%	53%	44%	39%
	Multi-refs IBBPBBP...	Multi	Avg	64%	65%	57%	89%	48%	43%	68%	65%	59%	69%	49%	44%
		Add	Avg	59%	61%	55%	57%	63%	60%	61%	61%	57%	61%	48%	45%
		Shift	Avg	52%	54%	47%	38%	71%	69%	51%	55%	49%	50%	45%	41%
CIF	One-ref IBPBP...	Multi	Avg	65%	70%	52%	98%	54%	42%	58%	73%	58%	61%	69%	50%
		Add	Avg	60%	66%	52%	60%	68%	62%	55%	69%	58%	57%	66%	52%
		Shift	Avg	52%	57%	40%	38%	75%	70%	50%	61%	46%	52%	59%	41%
	One-Ref IBBPBBP...	Multi	Avg	67%	70%	62%	88%	52%	46%	61%	76%	68%	66%	67%	61%
		Add	Avg	61%	66%	60%	58%	63%	60%	57%	71%	66%	61%	64%	58%
		Shift	Avg	53%	57%	50%	38%	71%	67%	51%	62%	56%	55%	58%	49%
	Multi-refs IBPBP...	Multi	Avg	63%	69%	51%	87%	48%	38%	57%	71%	55%	55%	60%	46%
		Add	Avg	58%	64%	51%	55%	63%	58%	54%	67%	55%	51%	57%	47%
		Shift	Avg	51%	56%	40%	35%	72%	67%	48%	59%	43%	46%	52%	38%
	Multi-refs IBBPBBP...	Multi	Avg	66%	70%	61%	79%	48%	43%	60%	72%	65%	57%	61%	53%
		Add	Avg	60%	65%	58%	53%	60%	57%	56%	68%	62%	53%	57%	52%
		Shift	Avg	52%	56%	48%	35%	67%	64%	49%	59%	52%	47%	51%	43%

bidirectionally predicted block, MC complexity for luminance is further simplified by merging the forward and backward MC interpolations according to the linear operation property and a simplified MC interpolation is also introduced for chrominance. As a result, the proposed methods can significantly reduce the decoding complexity for B-picture albeit H.264/AVC compliant bitstream can not be generated when they are integrated into H.264/AVC. It is also able to yield the comparable rate distortion performance in comparison with H.264/AVC.

REFERENCES

[1] 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.

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

[3] J. Shen and J. Ribas-Corbera, "Benefits of adaptive motion accuracy in H.26L video coding," in *Proc. Int. Conf. Image Process.*, 2000, vol. 1, pp. 1012–1015.

[4] G. Sullivan and T. Wiegand, "Rate distortion optimization for video compression," *IEEE Signal Process. Mag.*, pp. 74–11–90, Nov. 1998.

[5] S. Wu. and A. Gersho, "Joint estimation of forward and backward motion vectors for interpolative prediction of video," *IEEE Trans. Image Process.*, vol. 3, no. 9, pp. 684–687, Sep. 1994.

[6] A. Pun *et al.*, "Video coding with motion compensated interpolation for CD-ROM applications," *Image Commun.*, vol. 2, no. 2, pp. 127–144, Aug. 1990.

[7] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264AVC baseline profile decoder complexity analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 04–716, Jul. 2003.

[8] G. Bjontegaard and A. Fuldseth, *Simplified Chroma Interpolation Filter* Busan, Korea, Apr. 2005, ITU-T SG 16 Doc. VCEG-Z08.

[9] H.264/AVC Reference Software JM10.1 [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[10] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves,* Mar. 2001, ITU-T SG16 Doc. VCEG-M33.

[11] T. Chu *et al.*, *Discussion on Efficient 16-Bit Implementation of the Interpolation Routine,* 2001, ITU-T SG16 Doc. VCEG-N19.