

## Research Article

# Improved Intra-coding Methods for H.264/AVC

Li Song,<sup>1</sup> Yi Xu,<sup>1</sup> Cong Xiong,<sup>1</sup> and Leonardo Traversoni<sup>2</sup>

<sup>1</sup>The Institute of Image Communication and Information Processing, Shanghai Jiaotong University, Shanghai 200240, China

<sup>2</sup>División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana-Iztapalapa, 09340 México, DF, Mexico

Correspondence should be addressed to Li Song, song.li@sjtu.edu.cn

Received 2 June 2008; Revised 3 September 2008; Accepted 1 February 2009

Recommended by Liang-Gee Chen

The H.264/AVC design adopts a multidirectional spatial prediction model to reduce spatial redundancy, where neighboring pixels are used as a prediction for the samples in a data block to be encoded. In this paper, a recursive prediction scheme and an enhanced (block-matching algorithm BMA) prediction scheme are designed and integrated into the state-of-the-art H.264/AVC framework to provide a new intra coding model. Extensive experiments demonstrate that the coding efficiency can be on average increased by 0.27 dB with comparison to the performance of the conventional H.264 coding model.

Copyright © 2009 Li Song et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

H.264/AVC [1] is the newest international video coding standard of ITU-T (as Recommendation H.264) and ISO/IEC (as International Standard 14496-10 akin MPEG-4 part 10) advanced video coding (AVC). It considerably reduces the bit rate by approximately 30 to 70 percent when compared with previous video coding standards such as MPEG-4 Part 2, H.263, H.262/MPEG-2 Part 2 and to name a few, while providing the same or better image quality.

The intracoding algorithm of H.264 exploits the spatial and spectral correlation present in an image. Intraprediction removes spatial redundancy between adjacent blocks by predicting one block from its spatially adjacent causal neighbors. A choice of coarse and fine intraprediction is allowed on a block-by-block basis. There are two types of prediction modes for the luminance samples, that is, the so-called Intra  $4 \times 4$  mode which predicts each  $4 \times 4$  block independently within a macroblock and the Intra  $16 \times 16$  mode which predicts a  $16 \times 16$  macroblock as a whole unit. As for Intra  $4 \times 4$  mode, nine prediction modes are available for the encoding procedure, among which one represents a plain DC prediction and the remaining ones operate as directional predictors distributed along eight different angles, as shown in Figure 1. Intra  $16 \times 16$  mode is suitable for smooth image areas, where four directional prediction modes are provided as well as the separate intraprediction mode for the chrominance samples of a macroblock.

H.264 achieves excellent compression performance and complexity characteristics in the intramode even when compared against the standard image codecs (JPEG and JPEG2000) [2, 3]. In recent years, extended works have been developed to further improve the performance of intraprediction. Gang et al. proposed an intraprediction method based on subblock, altering the encoding order of the predictive subblocks so as to make the intraprediction adaptive to various textures [4]. However, this method needs to add new syntax elements and as well incurs nonnegligible complexity. Some authors introduced intramotion compensated prediction of macroblocks [5]. Block size and accuracy adaptation can be brought into the intra block-matching scheme to further improve the prediction results. In such a manner, the position of reference block should be coded into the bit stream. Thus a lot of extra side information would affect the performance significantly. To reduce this overhead information, special processing techniques are developed and result in a big change of intracoding structure in the H.264/AVC standard [6]. In [7], block-matching algorithm (BMA) is utilized to substitute for H.264 DC intraprediction mode with no need to code side information. However, prediction performance would be degraded if directly using previously reconstructed pixels for the matching procedure. Also, improved lossless intracoding methods are proposed to substitute for horizontal, vertical, diagonal-down-left (mode 3), diagonal-down-right (mode 4) of H.264/AVC [8, 9]. They employ samplewise differential pulse code modulation

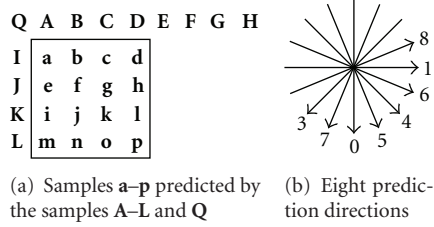


FIGURE 1: Intra  $4 \times 4$  coding mode.

(DPCM) method to conduct prediction of pixels in a target block. Yet this kind of methods can only be used in lossless mode.

From the above-mentioned analysis, current-enhanced intracoding methods still have problems remained, namely, either changing the coding structures a lot (e.g., [5, 6]) or having limited usage (e.g., [9, 10]) or alternatively less gain (e.g., [4]). In this paper, we focus on how to improve the performance of intracoding without incurring high cost of complexity and major changes for the design structure of H.264/AVC. Two prediction schemes are advanced to improve current intracoding performance. In the first scheme, more neighboring pixels contribute to recursively predict current pixel inside one block in a samplewise manner. Consequently, this scheme would match texture characteristics of the input source with high adaptation and minor extra complexity as well. The other prediction scheme is motivated by the fact that loop filter can significantly enhance the performance of the inter prediction. We propose to extend the classical BMA method [7] by imposing loop filtering on previously reconstructed macroblocks before BMA operation. Specifically, we change the orders of standard deblocking loop filter of H.264/AVC to achieve extra gains without incurring extra complexity. Extensive experiments show that intracoding of H.264 can be further improved in the proposed work for both lossy and lossless case.

The remaining parts of this paper are structured as follows. Section 2 describes the proposed recursive prediction scheme and the enhanced BMA prediction scheme. Codec-related issues are discussed in Section 3. Comparison experiments of the proposed intracoding model and the standard one in H.264/AVC are shown in Section 4. Finally, Section 5 concludes the paper.

## 2. Two Prediction Schemes for Intracoding of H.264/AVC

In this section, we will explain the improvement mechanism behind the recursive prediction scheme and the enhanced BMA prediction scheme. Both schemes join in the prediction modes of H.264/AVC with good compatibility and complementary merits. The resultant intracoding model can well improve the overall performance of H.264/AVC.

*2.1. Mechanism of Recursive Prediction Scheme.* It is generally accepted that Gaussian-like distribution could approximate the local intensity variations in smooth image regions. The

correlation between neighboring pixels would be attenuated while the distance is increasing and negligible when pixels are far enough apart. Furthermore, the assumption of the Gaussian distribution would become weak around the irregular texture areas and edge structures. The current prediction methods of H.264/AVC take an assumption that the intensity is uniform within the block to be predicted. Thus over-smoothness would be induced to the target block after prediction. As a result, the original intensity distribution is more or less destroyed. Especially for those natural images with abundant textures, the perception distortions are distinct. In all cases, high correlation can be expected among the nearest neighbors spaced one pixel apart except those within the image structures thinner than one pixel.

Given a  $4 \times 4$  luma block to be coded as shown in Figure 2(a), namely, the sequence of pixels from **a-p**, the mechanism of standard prediction mode and recursive prediction mode can then be, respectively, illustrated in Figures 2(b) and 2(c). Here we use gray color to mark the reference pixels, that is, the pixel set  $S = \{A, B, C, D, Q, I, J, K, L\}$ . Then pixels **a-p** will be predicted from these reference pixels. Now we explain the prediction procedure of pixels **a, f, k, p** referring to Figure 2. In standard prediction mode, these four pixels would take the same value which is deduced from reference pixels **A, Q**, and **I**. Residuals might have large values if the assumption of uniform intensity is violated. Alternatively, we select different reference pixels to recursively predict the value of **a, f, k**, and **p**. Only the left, the top, and the left-top pixels are actively involved in computing the center pixel value. Therefore the contribution of neighboring pixels is gradually decayed with distance increasing during the recursive prediction. The textures within the block would be retained, which results in smaller residual deviations.

In block-based H.264/AVC, we cannot obtain reconstruction of pixels inside current coding block except lossless case, where the reconstructed frame is identical to the original frame. In fact, only predicted value of neighboring pixels obtained in previous step is used to predict current pixel in our method. That is, it recursively predict each pixel inside block in the raster scan order.

Furthermore, we emphasize two facets in the implementation of the proposed recursive prediction method. On one hand, no modification should be imposed on the other parts of the design structure of H.264/AVC, besides part of intraprediction module. Specifically, we only change the five modes of H.264/AVC intraprediction module, among which are DDR mode (mode 4), HD mode (mode 5), VR mode (mode 6) for  $4 \times 4$  luma blocks, plane mode (mode 3) for  $16 \times 16$  luma blocks, and plane mode (mode 3) for chroma block. These five modes can easily support prediction neighborhood of our method. On the other hand, we would expect to find the tradeoff between the complexity and efficiency of the whole intracoding procedure.

For convenience in representation, we denote current pixel value as  $p$ , where  $(x, y)$  is the spatial position within the block, for example,  $(0, 0)$  indicates the left-top pixel. As

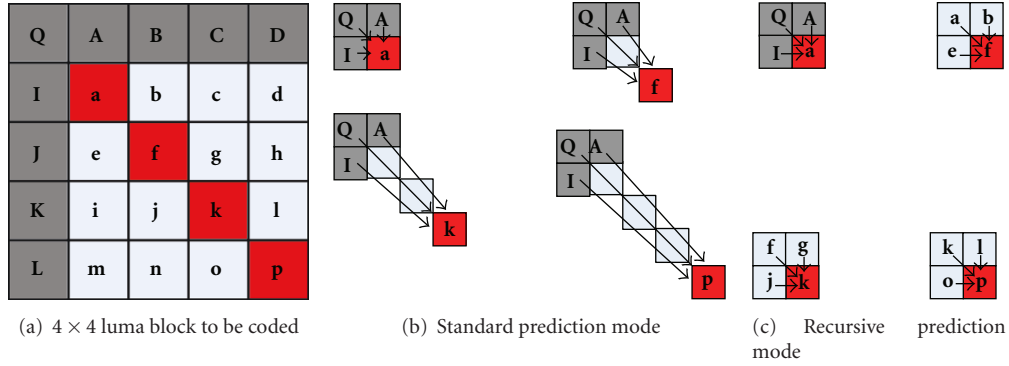


FIGURE 2: Comparison between recursive prediction mode and standard prediction mode.

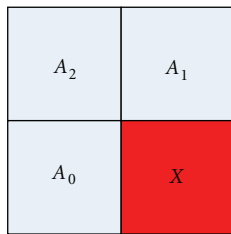


FIGURE 3: The tap filter of recursive prediction.

TABLE 1: The tap filters corresponding to the five-modified prediction modes.

New mode	Tap filter coefficients			(x, y)
	A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	
4 × 4 DDR	1	2	1	0~3, 0~3
4 × 4 VR	3	-1	2	0~3, 0~3
4 × 4 HD	2	-1	3	0~15, 0~15
16 × 16 PLANE	5	-2	5	0~7, 0~7
Choma PLANE	5	-2	5	0~3, 0~3

shown in Figure 3, the value of the predicted pixel can be computed from

$$p(x, y) = \text{Clip}(\text{Round}(A_0 \times p(x-1, y) + A_1 \times p(x, y-1) + A_2 \times p(x-1, y-1))), \quad (1)$$

where Round( $\cdot$ ) is the numerical operation that returns the closest integer to “ $\cdot$ ,” Clip( $\cdot$ ) is another numerical operation which clamps the predicted value to the range of [0, 255]. The tap filter coefficients corresponding to the five-modified prediction modes, which are gotten from experiments, are listed in Table 1.

**2.2. The Mechanism of Enhanced BMA Intraprediction Scheme.** Block matching is originally used in image restoration task to recover missing blocks [11]. The main assumption behind this application is that one block always has similar counterparts in the same frame. Yang et al. [7] integrated block-matching algorithm into DC mode of

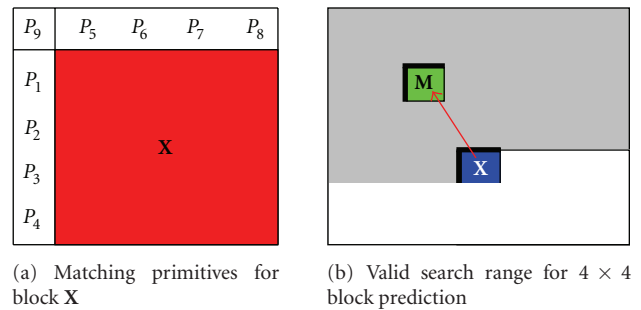


FIGURE 4: BMA prediction mode.

H.264/AVC standard prediction methods and generated an outcome of BMA mode for intraprediction. As coding is a sequential execution, one only can use the upper side, the left side, and the left up side of the boundary to perform block-matching, that is, the pixel set consisting of  $p_1$ – $p_9$  around block “X,” as depicted in Figure 4(a). The green block “M” in Figure 4(b) is the candidate block while the blue block “X” is the block to be predicted. The black pixels along the boundary are selected as the matching primitives. The valid search range is marked as the gray region. The matching process is formulated as the minimization of the following cost function:

$$\text{MSE} = \sum_{i=1}^9 (p_i - p'_i)^2, \quad (2)$$

where  $p_i$  and  $p'_i$ , respectively, represent the pixel values within block “X” and block “M.”

It is noted that original DC mode should be still used when the upper or left side is not available for the block to be predicted. Similar to the encoder, the decoder also needs to do block-matching.

The BMA prediction method has been proved as a good means to achieve gains in some video sequences [7], whereas there are still two open problems. BMA is accurate in high bitrates encoding case but not much good in low bitrate. The main reason is that the candidate macroblocks have not yet been passed to the loop filter, thus the best matches and the residuals would be greatly affected by the conspicuous

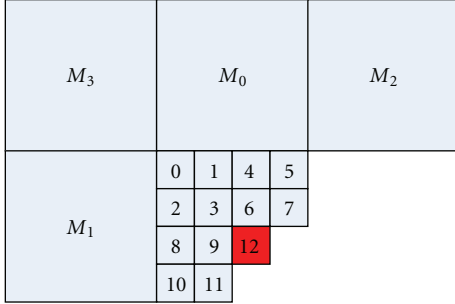


FIGURE 5: Candidate matches for a  $4 \times 4$  luma block in the enhanced BMA intraprediction scheme.

blocking artifacts. Especially when the best match spans two or more encoding macroblocks, it might be considered as a false match or the prediction residuals would increase sharply due to the blocking artifact. In addition, only the upper side, left side, and left up side pixels along the boundary of the block contribute to the block-matching results. The limited number of primitives would result in high ambiguities in the matching process. It is important to rationally reduce the solution space to a more restricted one.

To alleviate the ill-effects incurred by blocking artifacts, we put the loop filtering at the rear of BMA intracoding step for each macroblock rather than perform it after the whole slice has already been coded. Thus all the previously coded macroblocks are well deblocked and provide more correct details for the subsequent blocks to find a good match. The prediction error propagation through all the macroblocks is then well controlled. The good compatibility with standard H.264/AVC can be expected since we only change the order of loop filtering step in the whole functional structure but not change the loop filtering itself. Also no extra complexity is induced by this improvement.

To further reduce the ambiguities involved in matching, we constrain the search space to a more restricted one than that in the original BMA method, as shown in Figure 5, only the left macroblock “ $M_1$ ,” the left-top macroblock “ $M_3$ ,” the top macroblock “ $M_0$ ,” the right-top macroblock “ $M_2$ ,” and those blocks numbered from 0–11 predicted ahead are considered as the candidate match of the current  $4 \times 4$  luma block 12. Our extensive experiments proved that in most cases the globally optimal match can be captured by neighboring candidates  $M_0$ – $M_3$ . It should be noted that macroblocks  $M_0$ – $M_3$  have been loop filtered but the luma blocks 0–11 are not involved in deblocking before the current macroblock has been wholly predicted. Considering the compatibility with standard H.264/AVC, we restrict the search space to  $M_0$ – $M_3$ .

### 3. Codec-Related Issues

We hybridize the two proposed schemes into an H.264/AVC functional structure as the new modes for intraprediction. For purpose of easy implementation and bit savings, we substitute mode, 4, 5, 6 of  $4 \times 4$  luma prediction, mode 3 of  $16 \times 16$  luma prediction, and  $8 \times 8$  chroma prediction

with corresponding recursive prediction mode. In addition, we replace mode 2 (which is DC mode in intraprediction for  $4 \times 4$  blocks) with the enhanced BMA prediction mode without concern over those blocks on the upper or left frame boundary. Such a combination depends on the complementary properties of the two proposed schemes, which would be discussed in Section 4.

The encoder uses the new modes along with the other preserved modes to perform prediction for  $4 \times 4$ ,  $16 \times 16$ , and  $8 \times 8$  blocks. Among these prediction modes, the mode with the lowest rate-distortion cost would be selected as the optimal mode for prediction. Since there is no extra mode introduced, the syntax of the original standard of H.264/AVC remains unchanged. Only semantic or decoding processing needs to be modified correspondingly.

On the decoder part, we can directly perform the operations similar to those at the encoder for recursive prediction. As for mode 2, we first check whether the block is located at the upper or the left boundary of the frame. If so, we decode it using normal DC mode. Otherwise, we decode it using enhanced BMA intraprediction mode. Before decoding one block in enhanced BMA mode, loop filter is imposed on the nearest neighboring macroblocks to alleviate blocking effects, as shown in Figure 5. Afterward, the decoder runs a block search in the current frame. The best match would be utilized for prediction.

## 4. Experimental Results

To characterize the performance of two proposed prediction schemes, we select a variety of video sequences to execute the intracoding tests. Here we provide comparison experiments to evaluate the performance of five intracoding prediction schemes. Besides the proposed recursive intraprediction scheme (R scheme) and enhanced BMA intraprediction scheme (E-BMA scheme), the standard intraprediction scheme in H.264/AVC (S scheme) [1], the original BMA intracoding scheme (BMA scheme) [7], and the hybrid intraprediction scheme (H scheme) combined the two proposed methods are testified in terms of computational complexity, lossless compression, and variable bitrate. The baseline work is referred to the open H.264/AVC codec rev602 [12].

At first, we provide the common configuration parameters in the tests. Frame rate is set at 30 Hz. The total number of the encoded frames is 100 for each test sequence. Hadamard transform is enforced on these video frames.  $8 \times 8$  transform is not chosen. As for the entropy coding, the CAVLC (context-based adaptive variable length coding) is used for the experiment, RDO is enabled and all I frames of video are encoded as intraframe with different QP (QP = 0 for lossless). As for other typical settings such as CABAC entropy coding, RDO disabled, rate control enabled, experiments consistently show similar gains of our proposed scheme. In the following experiments, we regard S scheme as the anchor and analyze the relative performance of the other four counterparts.

TABLE 2: Computational complexity analysis of R scheme.

Mode	Operation	$4 \times 4$ DDR	$4 \times 4$ VR	$4 \times 4$ HD	$16 \times 16$ plane	Chrome plane
S scheme	ADD/SUB	45	58	58	1229	347
	MUL/DIV	30	36	36	789	203
R scheme	ADD/SUB	64	64	64	1024	256
	MUL/DIV	16	32	32	768	192

TABLE 3: The bitrate saved by E-BMA scheme, R scheme, and H scheme.

	Resolution	Sequence $\Delta B(\%)$		
		E-BMA scheme	R scheme	H scheme
QCIF	Bus	0.40	0.18	0.73
	Carphone	1.30	0.09	1.49
	City	0.21	0.65	0.97
	Crew	0.05	0.37	0.55
	Football	0.10	0.37	0.64
	Foreman	1.92	-0.83	1.92
	Harbor	0.17	0.47	0.78
	Ice	0.44	0.00	0.56
	Mobile	0.18	-0.07	0.21
	Soccer	0.11	0.24	0.49
CIFb	Bridge	0.19	0.08	0.40
	Bus	0.61	0.28	1.04
	City	1.65	0.66	2.35
	Coastguard	0.26	0.52	0.89
	Crew	0.02	0.26	0.37
	Flower	0.08	-0.15	0.09
	Football	0.38	0.54	1.13
	Foreman	1.01	-0.51	1.09
	Harbor	0.37	0.64	1.14
	Highway	0.62	-0.22	0.62
	Ice	0.52	0.09	0.70
	Mobile	0.53	-0.31	0.42
	News	0.77	-0.03	0.86
	Paris	0.70	-0.20	0.75
	Soccer	0.22	0.39	0.76
	Stefan	0.44	0.26	0.84
Tempete	0.25	0.32	0.70	
Waterfall	0.02	0.40	0.65	
4CIF	City	2.08	0.84	2.94
	Crew	0.06	0.44	0.64
	Harbor	0.69	0.96	1.74
	Ice	0.37	0.25	0.74
	Soccer	0.42	0.72	1.28
HD	City	1.26	1.27	2.31
	Harbor	0.70	1.18	1.97
<i>Average <math>\Delta B</math></i>		<i>0.55</i>	<i>0.29</i>	<i>1.00</i>

TABLE 4: Video quality assessment of BMA scheme, E-BMA scheme, R scheme, and H scheme.

Resolution	Sequence $\Delta$ PSNR(dB)				
	BMA scheme	E-BMA scheme	R scheme	H scheme	
QCIF	Bus	0.01	0.05	0.08	0.14
	Car-phone	0.17	0.47	0.01	0.47
	City	-0.01	0.07	0.11	0.18
	Crew	-0.02	0.20	0.08	0.27
	Football	-0.01	0.07	0.06	0.13
	Foreman	0.40	0.60	-0.19	0.49
	Harbor	-0.02	0.08	0.14	0.21
	Ice	0.06	0.27	0.02	0.29
	Mobile	0.01	0.05	0.03	0.09
	Soccer	-0.02	0.05	0.05	0.09
CIF	Bridge	-0.01	0.05	0.05	0.09
	Bus	0.05	0.09	0.14	0.22
	City	0.20	0.30	0.13	0.41
	Coastguard	-0.01	0.00	0.12	0.11
	Crew	-0.01	0.22	0.05	0.26
	Flower	0.00	0.03	0.07	0.12
	Football	0.01	0.16	0.12	0.27
	Foreman	0.22	0.45	-0.17	0.35
	Harbor	0.01	0.14	0.16	0.29
	Highway	0.22	0.46	-0.08	0.41
	Ice	0.10	0.29	0.00	0.30
	Mobile	0.07	0.13	0.02	0.16
	News	0.13	0.35	0.05	0.39
	Paris	0.09	0.30	0.02	0.35
	Soccer	0.00	0.08	0.07	0.15
	Stefan	0.04	0.16	0.17	0.34
Tempete	0.00	0.14	0.13	0.27	
Waterfall	-0.03	0.01	0.13	0.13	
4CIF	City	0.21	0.33	0.21	0.51
	Crew	-0.01	0.18	0.06	0.23
	Harbor	0.03	0.25	0.24	0.47
	Ice	0.11	0.30	0.00	0.32
	Soccer	0.01	0.11	0.15	0.24
HD	City	0.02	0.15	0.17	0.28
	Harbor	-0.04	0.23	0.22	0.33
<i>Averaged <math>\Delta</math>PSNR</i>		0.06	0.19	0.07	0.27

4.1. *Experiment I: Performance Evaluation with Respect to Computational Complexity.* The computational complexity of R scheme can be easily calculated. The anchor computation of standard H.264/AVC corresponding modes can be referred to [13]. In the case of DDR mode (diagonal down right) of Intra  $4 \times 4$  prediction, the pixels from **a-p** in Figure 1 are predicted from the uniform formulation  $(I + 2Q + A + 2)/4$  as referred to formula (1) and the tap filters designated in Table 1. It needs 3 addition operations, 1 multiplication (bitwise left shift) operation and 1 division (bitwise right shift) operation to calculate the prediction sample. However, we can replace some multiplication operations with addition,

for example, using  $Q + Q$  instead of  $2 \times Q$ . So we only need four times additions and one division operations for one pixel. Besides DDR mode, the other modes can be computed in the similar way.

Table 2 presents the computational complexity analysis of recursive prediction relative to the normal mode in H.264/AVC (S scheme), which is obtained by counting addition/subtraction and multiplication/division for corresponding  $4 \times 4$ ,  $16 \times 16$ , or chrome block. The difference of computational complexity between BMA scheme and E-BMA scheme mainly depends on the search range selected in both schemes since two computational structures are

equal except the loop filtering order. Thus E-BMA can be expected with lower computational complexity than BMA scheme because of a narrower search range. Compared with S scheme, the increased complexity is high because of the additional block-matching step. The computational complexity in the encoder is similar to that of motion estimation, using a 9-pixel template inside the search region. Therefore the order of complexity in the encoder is similar to that of P-slice. In Figure 5, we need 748 times computation (formula (2)) and comparison for every  $4 \times 4$  block with full-pixel block-matching in our current implementation, which makes both encoder and decoder 5~8 times slower than standard H.264/AVC intraprediction.

It seems such high computational complexity will offset benefits of E-BMA scheme. However, fast search techniques similar to fast motion estimation in inter prediction and parallel algorithms can be employed in block-matching to greatly speed up our current full-pixel procedure. Such accelerated methods are out of the scope of this paper, but we conjecture complexity of H scheme which integrate R and EBMA scheme should be something between intra-(I slice) and inter prediction(P slice). Therefore, the complexity issue of proposed hybrid intramode is not so serious when used with inter prediction (P or B slice).

In the decoder, the increase of computational complexity depends on the number of the blocks that use this mode. In sequences where the E-BMA mode really helps in the coding efficiency, this mode is selected in the order of 15% ~35% of the blocks. In sequences where the E-BMA mode is not selected, the additional computational complexity is negligible.

*4.2. Experiment II: Performance Evaluation with Respect to Lossless Compression.* As analyzed in experiment I, the difference between BMA scheme and E-BMA scheme depends on two facets, namely, the order of loop filtering step in the whole functional structure and the search range of the best match to output the prediction residuals. Since the search range is a kind of parameter setting problem, the main difference related to the fundamental mechanism exists in the loop filtering order. However, there is no loop filtering adopted in the H.264/AVC video coding standard under the lossless compression case. Therefore the BMA scheme and the E-BMA scheme would be expected with similar performance evaluation with regard to lossless compression.

According to (3), we list in Table 3 the bitrate saved by E-BMA scheme, R scheme, and H scheme when compared to S scheme in a varied corpus of YUV video sequences recorded at QCIF, CIF, 4CIF, and HD resolutions,

$$\Delta B = \frac{B_s - B_x}{B_s} \times 100\%. \quad (3)$$

In the above formulation,  $B_x$  denotes the bitrate required in the given scheme while  $B_s$  represents the anchor one required in S scheme.

From the above analysis in Table 3, it is shown that the bitrate is positively reduced in E-BMA scheme for lossless compression of all the test video sequences. As for R scheme,

the bitrate is somewhat oscillatory with negative reduction in a few sequences which have more local directional smooth structures (e.g., background of “foreman”). The pixelwise recursive prediction is not effective in these areas. As a hybrid combination of E-BMA scheme and R scheme, H scheme achieves the highest bitrate savings with the average reduction of 1%. In general, the test sequences are coded at a slightly lower bitrate in E-BMA scheme, R scheme, and H scheme as compared to S scheme for achieving lossless quality.

*4.3. Experiment III: Performance Evaluation with Respect to Variable Bitrate.* To cover a wide range of bitrates, we choose the QP values among 16, 20, 24, 28, 32, and 36. Thus the performance of the prediction schemes could be evaluated from high bitrate to low bitrate. Here PSNR tool is used to measure video quality under varied prediction schemes. Given PSNR measurement of S scheme, we define the PSNR gain of the other schemes as

$$\Delta\text{PSNR} = \text{PSNR}_x - \text{PSNR}_s, \quad (4)$$

where  $\text{PSNR}_x$  denotes the peak signal-to-noise ratio acquired in the given scheme while  $\text{PSNR}_s$  represents the reference one acquired in S scheme. Similar to the calculation in [14], the outputs of  $\Delta\text{PSNR}$  are averaged for all the QP options (16~36) and listed in Table 4.

BMA scheme shows its advantages in a few video sequences, such as “Foreman,” “City,” and “Highway.” However, no distinct improvements can be observed in the major part of the test sequences. Even degradation is introduced to some sequences as the blocking artifact increases the cost of (2). In contrast, the proposed E-BMA scheme improves the video quality by 0.2 dB on average. As for the proposed R scheme, half of the sequences are improved by over 0.1 dB in quality. A few sequences are somehow degraded while using R scheme, such as “Foreman” and “Highway” (the possible reason has been explained in Section 4.2). As the hybrid scheme between E-BMA and R, H scheme presents its promising performance in all the cases. We even can get 0.35 dB improvement in some sequences, for example, “Carphone,” “Foreman,” “City,” and “Harbor.” The main reason of such a positive evolution can be found in the complementary properties of E-BMA scheme and R scheme. In our experiments, it is shown that E-BMA has better performance under low bitrates in that block-matching is well known for its good performance in smooth regions. On the contrary, R scheme is motivated to preserve the textures within the block, which shows more promising performance under high bitrates. Also the video contents would affect the performance of these two schemes due to the distribution of smooth regions and nonsmooth regions. For example, R scheme achieves the higher prediction accuracy as in those sequences like “Carphone,” “Crew,” “Ice,” “Foreman,” and “Paris.” But it runs in the opposite way as in the sequences like “Bus,” “Coastguard,” and “Waterfall.”

Furthermore, we use three rate distortion (RD) curves to demonstrate the improvement induced by the hybrid combination of E-BMA scheme and R scheme, respectively,

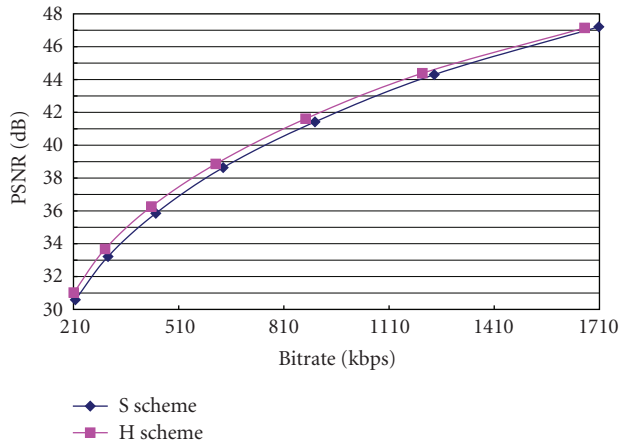


FIGURE 6: RD curves of encoding Carphone (QCIF) sequence with S scheme and H scheme.

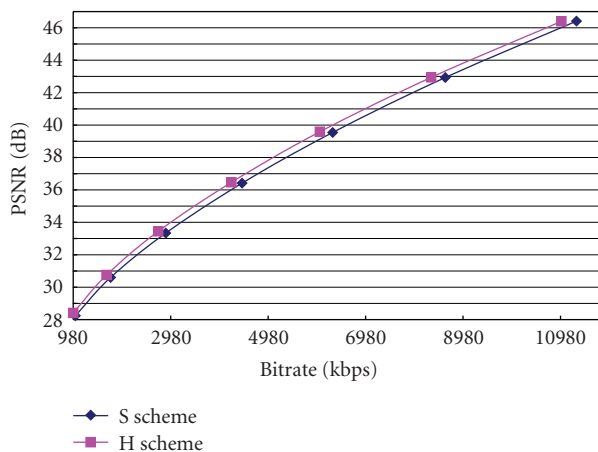


FIGURE 7: RD curves of encoding City (CIF) sequence with S scheme and H scheme.

for three sequences recorded at different resolutions, that is, “Carphone,” “City,” and “Harbor,” as illustrated in Figures 6, 7, and 8.

## 5. Conclusion

In this paper, we propose two schemes to further improve the performance of intraprediction in H.264/AVC. The new modes developed by these schemes replace the classical direction prediction modes of H.264. The experimental results demonstrate that our schemes could improve the overall performance of compressed I frame by 0.1~0.47 dB as compared to the H.264/AVC standard. In addition, our schemes have high compatibility with many existing prediction methods. However, for video sequences with directional structures, recursive prediction degrades its performance a little. In our future research, we will explore more complex context to improve its performance of prediction. As for E-BMA, further gains can also be expected if we introduce adaptive template and extend our block-matching to the subpixel accuracy case.

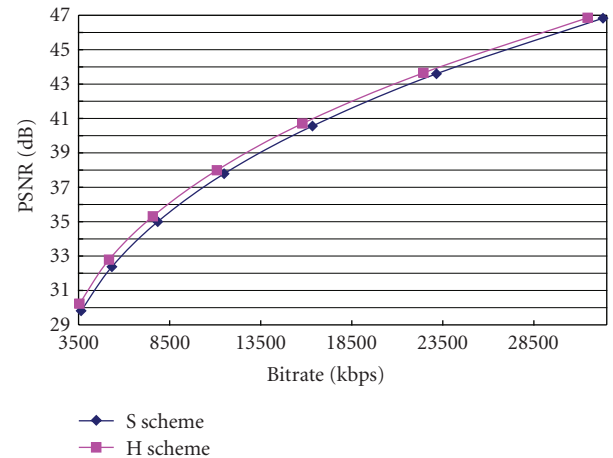


FIGURE 8: RD curves of encoding Harbor (4CIF) sequence with S scheme and H scheme.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (60702044 and 60632040) and Research Fund for the Doctoral Program of Higher Education of China (200802481006).

## References

- [1] ITU-T Recommendation H.264 and ISO/IEC 14496-10, “Advanced video coding for generic audiovisual services,” May 2003.
- [2] D. Marpe, V. George, H. L. Cycon, and K. U. Barthel, “Performance evaluation of Motion-JPEG2000 in comparison with H.264/AVC operated in pure intra coding mode,” in *Wavelet Applications in Industrial Processing*, vol. 5266 of *Proceedings of SPIE*, pp. 129–137, Providence, RI, USA, October 2003.
- [3] A. Al, B. P. Rao, S. S. Kudva, S. Babu, D. Sumam, and A. V. Rao, “Quality and complexity comparison of H.264 intra mode with JPEG2000 and JPEG,” in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 1, pp. 525–528, Singapore, October 2004.
- [4] Z. Gang, G. Li, and Y. He, “The intra prediction based on sub block,” in *Proceedings of the 7th International Conference on Signal Processing Proceedings (ICSP '04)*, vol. 1, pp. 467–469, Beijing, China, August–September 2004.
- [5] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, “New intra prediction using intra-macroblock motion compensation,” in *Proceedings of the 3rd JVT Meeting*, Fairfax, Va, USA, May 2002, JVT-C151.
- [6] K. L. Tang and K. N. Ngan, “Enhancement techniques for intra block matching,” in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 420–423, Beijing, China, July 2007.
- [7] J. Yang, B. Yin, and N. Zhang, “A block-matching based intra frame prediction for H.264/AVC,” in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 705–708, Toronto, Canada, July 2006.
- [8] Y.-L. Lee, K.-H. Han, and G. J. Sullivan, “Improved lossless intra coding for H.264/MPEG-4 AVC,” *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2610–2615, 2006.



- [9] S. Takamura and Y. Yashima, "H.264-based lossless video coding using adaptive transforms," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 2, pp. 301–304, Philadelphia, Pa, USA, March 2005.
- [10] A. Robert, I. Amonou, and B. Pesquet-Popescu, "Improving intra mode coding in H.264/AVC through block oriented transforms," in *Proceedings of the 8th IEEE Workshop on Multimedia Signal Processing (MMSP '06)*, pp. 382–386, Victoria, Canada, October 2006.
- [11] Z. Wang, Y. Yu, and D. Zhang, "Best neighborhood matching: an information loss restoration technique for block-based image coding systems," *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1056–1061, 1998.
- [12] "x264—a free H.264/AVC encoder," <http://www.videolan.org/developers/x264.html>.
- [13] Y.-L. Lee and K.-H. Han, "Complexity of the proposed lossless intra for 4:4:4," JVT-Q035, October 2005.
- [14] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *Proceedings of the ITU-T VCEG 13th Meeting*, Austin, Tex, USA, April 2001, VCEG-M33.