Adaptive Patch Matching for Motion Compensated Prediction

Tianmi Chen[1*], Xiaoyan Sun[2], Feng Wu[2],Guangming Shi[1]

1. Key Lab. of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, China
2. Microsoft Research Asia, Beijing, China

ctm020421@163.com, xysun@microsoft.com, fengwu@microsoft.com, gmshi@xidian.edu.cn

***Abstract*—Motion compensated prediction (MCP) plays an important role in video coding due to its great capability of reducing temporal redundancy. In this paper, we propose a new MCP scheme by adaptive patch matching with the full use of the reconstructed pixels surrounding the current block (referred to as the template inside the patch) aiming at achieving a more accurate prediction than conventional MCP. The proposed scheme not only takes advantage of the temporal correlation but also efficiently exploits the spatial correlation between the current block and its template inside the patch. An adaptive linear combination of the current block and its template in motion estimation is designed to generate an optimal prediction while maintaining the local variation of the current block. Accordingly, a modification of the rate-distortion criterion is introduced to select the combined prediction. Experimental results show that our proposed APM achieves improved coding performance compared with H.264/AVC.**

I. INTRODUCTION

Video coding exploits the temporal and spatial correlations and makes use of the redundancies in video sequences to achieve high coding efficiency. The latest video coding standard H.264/AVC, also known as one of the state-of-the-art coding schemes, is becoming more and more popular due to its remarkable improvement on coding efficiency. This hybrid coding scheme introduces some new techniques which enable H.264/AVC to achieve significant coding gain in video coding. Among them, advanced motion-compensated prediction (MCP) is one of the main contributors to the improvement in H.264/AVC [1].

MCP plays a key role in removing the temporal redundancy between video frames. Recently, many efforts have been made in developing the potential of MCP for further coding efficiency improvement. A flexible block partitioning scheme is presented in [2] to compensate the motion by considering the geometric characteristics at the cost of extra partition parameters. Larger partitions and transform sizes, such as 32x32 and 64x64 block sizes, are enabled in [3] to reduce the overhead bits. Moreover, by combing an elastic motion model with larger block partitioning for inter prediction, [4] achieves significant improvement while the optimal basis functions for the elastic motion model remain to be further determined. Different from traditional block-based coding structure, [5] presents a line-by-line macroblock (MB) partitioning and prediction scheme, in which motion prediction is performed row by row or column by column and resulting a 1D motion vector. Besides, motion-hypothesis is proposed in [6] for inter prediction. The motion-hypothesis prediction is generated using the motion information of its left and above neighbors with the same weight values.

In the past few years, template matching (TM), widely used in texture synthesis, has drawn an increasing attention in inter frame coding due to its advantage of allowing prediction of a target block by using its surrounding reconstructed pixels (defined as template) without transmission of motion vectors. The effectiveness of integrating TM with conventional motion compensation has been demonstrated in [7]. Further improvement is achieved in [8] by using decoder side motion vector derivation (DMVD) for inter frame coding. Later, [9] shows its possibility to apply the DMVD scheme to B frames. Enhanced performance can be achieved by using multi-hypothesis prediction with DMVD in both B and P frames. Recently, a refined motion-compensated prediction based on DMVD is presented in [10]. It enables multiple MCPs with only one motion vector transmitted so as to smooth the distortion and compensate some missing textures. In addition, two DMVD modes are presented in [11], in which the adaptive template shape, boundary weighting, and refinement search are designed to enhance the efficiency of TM prediction.

However, it has been found that the coding efficiency of TM highly depends on the correlation between the target block and its template. It is difficult for TM to handle the patches with weak correlation. Trying to solve this problem, a predictive patch matching is designed in [12] by considering both the reconstructed template and the predicted pixels generated by MCP in matching to achieve an enhanced prediction. In this scheme, the template and the predicted block in the patch matching make the same contribution to the final combined prediction.

In this paper, we propose a novel motion compensated prediction by adaptive patch matching (APM) in which the local texture characteristics of patches is considered in generating the compensated prediction. There are two contributions in our proposed scheme. First, it introduces the surrounding reconstructed pixels in motion estimation (ME).

[*] This work has been done during Tianmi Chen's internship in Microsoft Research Asia

978-1-4244-9472-9/11/$26.00 ©2011 IEEE

Second, a weighted prediction mechanism is proposed in APM. It generates a prediction by a linear combination between the predictions formed by both TM and modified MCP. The key idea in APM is to adaptively combine the target block with its template in the weighted patch matching for motion-compensated prediction.

The remaining of this paper is organized as follows. First, the conventional MCP in H.264/AVC is briefly reviewed in part A of Section II, and part B discusses our proposed APM in detail. Then experimental results are reported in Section III followed by conclusions in Section IV.

II. WEIGHTED PATCH MATCHING

A. MCP in H.264/AVC

H.264/AVC supports a tree-structured MCP with variable block sizes ranging from 16x16 down to 4x4. Each inter coded block (or partition) has one motion vector (MV) generated by MCP. The motion vector difference (MVD) is coded into bitstream together with its corresponding reference frame index and residual.

In H.264/AVC, ME is performed on MB level to check all the partition types and loop over all the reference frames, followed by a mode selection. During ME, a motion vector predictor (MVP) is first derived from neighboring available coded motion vectors. Centered on the position pointed by MVP, a full search is performed on integer-pel positions. The resulting MV is then refined by sub-pel motion search. The motion vector for each partition is selected by minimizing the cost

$$J_{motion}(MV, REF|\lambda_{motion}) = SAD_{target} + \lambda_{motion} \cdot \big(R(MVD) + R(REF)\big), \quad (1)$$

where MVD is calculated by subtracting MVP from MV, REF represents the index of reference frame, λ_{motion} denotes the Lagrangian parameter for motion, $R(MVD)$ and $R(REF)$ specify the cost for coding the MVD and REF, respectively. SAD_{target} measures the similarity between the original pixels and the predicted pixels pointed by MV referred to REF in the target block.

After ME, a mode decision is employed to compare the costs of all the modes (including intra mode) to select the best mode by minimizing

$$J_{mode}(Mode|\lambda_{mode}) = SSD_{target} + \lambda_{mode} \cdot (R(RESIDUE) + R(MVD) + R(REF) + R(MODE)) \ ,(2)$$

where SSD_{target} measures the distortion between the original pixels and reconstructed pixels in the target block, λ_{mode} denotes the Lagrangian parameter for mode decision, $R(RESIDUE)$ and $R(MODE)$ specify the coding bits for residual and mode, respectively.

B. Adaptive Patch Matching (APM)

As mentioned before, predictive patch matching is able to achieve enhanced performance by using both the surrounding reconstructed pixels and the predicted pixels generated by MCP to synthesize a prediction [12]. It assumes that the template and the predicted block have the same importance in the patch matching. However, we believe that a weighted combination of the template and the predicted block considering the spatial correlation inside the patch is necessary during ME. Therefore, an adaptive patch matching for motion compensated prediction is presented in this section for better adaption to local characteristic. As shown in Fig. 1, benefiting from the spatial continuity between the target block *b* and its template *t*, the proposed scheme can estimate a prediction b_1 through reconstructed pixels by TM without any side information. Then taking advantage of the temporal continuity, the traditional MCP is adopted to refine the prediction b_1. With the assist of template, the improved MCP is able to estimate a more accurate prediction b_2 than traditional MCP by considering the local statistical properties.

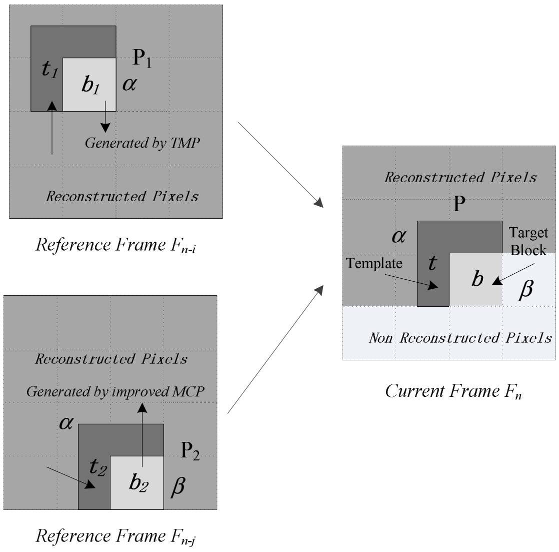


Figure.1 Illustration of adaptive patch matching in ME. The light gray and white regions denote the reconstructed and unknown regions, respectively. In this figure, the current frame is F_n. The target block is denoted by *b* and its template *t* is shown by dark gray regions. F_{n-i} and F_{n-j} present the i-th and j-th reference frames, respectively. P_1 and P_2 are candidate patches in F_{n-i} and F_{n-j}. b_1 denotes the prediction formed by TM by minimizing the distance between *t* and t_1, while b_2 denotes the prediction estimated by improved MCP with template *t* added.

In our proposed APM scheme, we define a patch *P* consisting of a target block *b* and its template *t*, and fill them with original pixels and its surrounding reconstructed pixels, respectively. During the ME, the whole patch *P* instead of the target block *b* is utilized to search for the best-matched prediction b_2 by minimizing the joint cost function (3). Correspondingly, the residual cost between the templates (*t* and t_2) as well as the target blocks (*b* and b_2) is calculated by SAD function. The motion cost is as same as that in (1). Finally, the cost function for determining the motion vector of patch P_2 is modified as

$$J_{motion}(MV, REF|\lambda_{motion}) = SAD_{patch} + \lambda_{motion} \cdot \big(R(MVD) + R(REF)\big), \quad (3)$$

Here the residual cost function is redefined as

$$SAD_{patch} = \alpha \cdot SAD_{template} + \beta \cdot SAD_{target}, \quad (\alpha + \beta = 1), \quad (4)$$

where SAD_{patch} and $SAD_{template}$ denote the residual cost of the whole patch and the residual cost at template regions respectively, α and β are weighting factors representing the importance of the reconstructed pixels and original pixels for patch matching in ME, respectively. It can be observed that whether a TM is performed or not depends on the weighting factor α. The main purpose of incorporating TM process in ME is to make full use of the reconstructed pixels to complement local variations and introduce diversity, resulting in a robust prediction. When α doesn't equal to 0, an additional prediction b_1 is derived by employing TM process. Then the final prediction of target block b comes from a weighted prediction between b_1 and b_2 which is directly influenced by α and β,

$$b = \alpha \cdot b_1 + \beta \cdot b_{2,} \tag{5}$$

The parameters α and β can be adaptively determined by the correlation between template regions. However, in our current solution, we define a weighting factor set F={ α_i } and code it as side information to the decoder side. For simplicity and limitation of overhead bits, we set α vary from 0 to 1 increasing by 0.2. The best $\alpha_i, i = 0,1,\dots 5$, for each partition will be selected according to (2) and the motion vector is determined by (3). To further reduce the overhead bits, we made a statistical analysis of the usage of the weighting factors. From our observation, among all the six weighting factors, 0, 0.2 and 0.4 take a large proportion at high bitrates and 1 occupies an increased proportion at low bitrates, as shown in Table 2. Accordingly, we choose to remove the factors 0.6 and 0.8 from factor set F. Thus, in our current solution, there are four factors {0, 0.2, 0.4, 1} utilized in AMP.

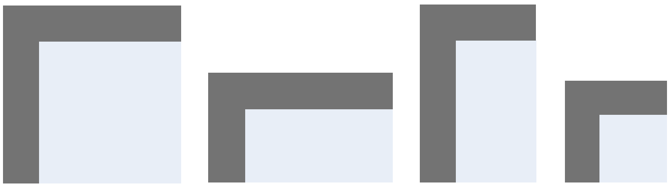
Since the weighting factors are variable and adaptive to the local characteristic, we call this patch matching adaptive patch matching and accordingly the final prediction generated by APM is an adaptive linear combination depending on the weighting factors. Typically, for α = 0 and β = 1, the APM performs as the same as conventional MCP; and for α = 1 and β = 0, the APM performs as the same as conventional TM.

We incorporate the APM into H.264 inter frame coding as a new motion prediction mode. Note our APM is only enabled for integer-pel ME. For sub-pel refinement, the conventional MCP is performed. Moreover, to limit the overhead bits as well as complexity, the proposed APM mode is only performed at MB level and has not extended to sub-block level so far. In this case, one mode flag in addition to a weighting factor is coded and transmitted to the decoder side for each MB. At the decoder side, the mode flag as well as the weighting factor α are decoded for each MB. With the motion vector and reference frame index decompressed, the predictions b_1 and b_2 are derived by MCP and TM, respectively. Then the final prediction is obtained by combining the two predictions by (5), where $\beta = 1 - \alpha$.

III. EXPEIMENTAL RESULTS

To evaluate the performance of our proposal, the proposed APM is incorporated into the H.264/AVC reference software [13]. Eleven sequences with resolutions from QCIF to 1080p are utilized in the tests. The test sequences are coded in IPPP prediction structure, with search range of 32x32, 4 reference frames and CABAC entropy coding. The RDO is enabled and four QPs of 22, 27, 32 and 37 are tested.

The patch size in our adaptive patch matching varies with the partition size of a macroblock. For each partition type, only the neighboring left four columns and upper-left four rows of the target block are used as the template, as shown in Fig. 2. Taking an 8x8 partition type as an example, the patch size is set to 12x12. To reduce the overhead bits, CABAC is used for coding the weighting factors.



(a) 16x16 (b) 16x8 (c) 8x16 (d) 8x8

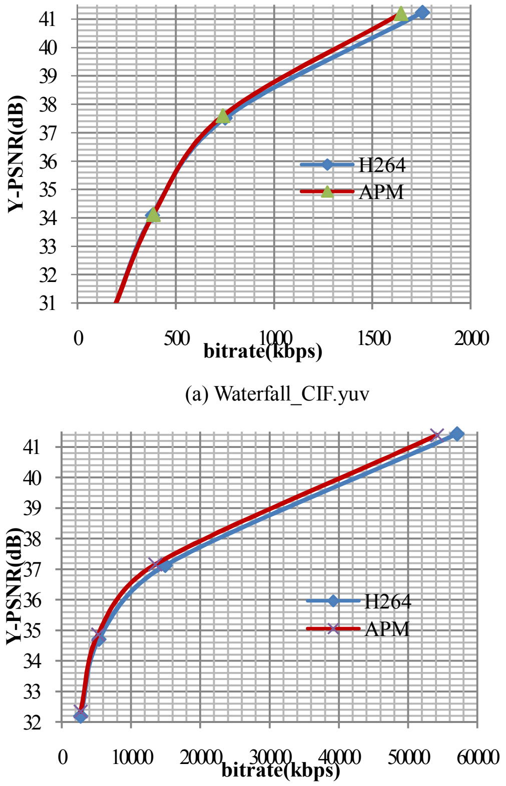
Figure.2 Patch sizes with different partitions. The gray and white regions denote the template and target block, respectively.

Table 1 shows the bitrate reduction of our proposed scheme calculated according to [14]. It can be observed that due to the use of our APM mode, the new coding scheme reduces 2.85% bitrate on average and achieves up to 8.02% bits saving in comparison with H.264/AVC.

Table 1. BDRATE reduction of the proposed scheme

Video Sequences		BDRATE (%)
176x144	Foreman	-1.39
352x288	Waterfall	-1.86
	Mobile	-3.29
416x240	BQSquare	-3.82
	BlowingBubble	-1.48
832x480	BQMall	-0.33
	PartyScene	-1.67
1280x720	City	-3.60
	BigShips	-1.75
	Spincalendar	-8.02
1920x1080	BQTerrace	-4.18
Average		-2.85

Fig. 3 exhibits two examples of the RD performance of our proposed scheme in comparison with H.264/AVC. In this figure, results of our scheme and H.264/AVC are marked by 'APM' and 'H264', respectively. It can be observed that the improvement of 0.3~0.4 dB in terms of PSNR can be achieved by our scheme at the high bitrates over H.264/AVC. For Waterfall_CIF.yuv, our scheme shows superior performance to H.264/AVC at high bitrates. While at low bitrate almost all of the MBs select 0 as the best weighting factor since it's hard for TM to find a comparable prediction to H.264/AVC due to the weak correlation within the low-quality reconstructed images for the texture regions at low bitrates. For Spincalendar_720p.yuv, constant coding gain can be achieved from high to low bitrates because of its regular edges and repeated patterns.



(a) Waterfall_CIF.yuv

(b) Spincalendar_720p.yuv

Figure. 3 RD performance comparison

The average percentage of weighting factors usage of the test sequences at different QPs is shown in Table 3. It can be seen that 0.2 and 0.4 take a large proportion at high bitrates, which means our APM works better than conventional MCP by using the spatial correlation inside the patch. In contrast, an increased percentage of 1 is obtained at low bitrates due to the saving of motion vector by TM at low bitrate.

Table 2. Percentage of weighting factors usage in F at different QPs

	Weighting Factors Usage (%)					
QP	0	0.2	0.4	0.6	0.8	1
22	37	30	22	6	1	3
27	41	25	17	8	2	7
32	48	19	13	8	3	10
37	53	14	9	6	3	15

Table 3. Percentage of weighting factors usage at different QPs

	Weighting Factors Usage (%)			
QP	0	0.2	0.4	1
22	44.5	33.4	19.5	2.6
27	49.8	32.0	14.1	4.0
32	60.2	24.5	9.4	5.9
37	76.7	13.0	4.7	5.6

IV. CONCLUSION

This paper proposes an adaptive patch matching for motion compensated prediction for inter frame coding. This new prediction mode enables us to adaptively make use of the reconstructed pixels surrounding the current block according to the spatial correlation inside the patch. A weighted prediction is also presented by linear combination of the template matching prediction and motion compensated prediction. Experimental results demonstrate the effectiveness of our proposed APM mode in inter frame coding.

ACKNOWLEDGMENT

This work is supported by NSFC（60736043，61033004，61070138，61072104）and by a grant from the Ph.D. Programs Foundation of Ministry of Education of China (No. 200807010004， 20090203110003).

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” IEEE Trans on CSVT, vol. 13, no. 7, pp. 560-576, 2003.

[2] O. D. Escoda, P. Yin, C. Dai and X. Li, “Geometry-adaptive block partitioning for video coding,” ICASSP 2007, pp. 657-660.

[3] P.Chenn, Y. Ye and M. Karczewicz, “Video coding using extended block sizes,” ITU-T VCEG contribution C123, Geneve, Switzerland, Jan. 2009.

[4] Muhit, A.A, Pickering, M.R, Frater, M.R, and Arnold, J. F, “Video coding using elastic motion model and larger blocks,” IEEE on CSVT, vol.20, no. 5, pp. 661-672, 2010.

[5] J-M. Thiesse, J.Jung, and M.Antonini, “Hybrid-1D macroblock prediction for video compressing,” EUSIPCO, Glasgow, Scotland, August 24-28, 2009.

[6] J Lim, S Park, and B Jeon, “Extended merging scheme using motion-hypothesis inter prediction,” JCTVC-B023, Geneva, CH, 21-28 July, 2010.

[7] Kazuo Sugimoto, Mitsuru Kobayashi, Yoshinori Suzuki, Sadaatsu Kato, and Choong Seng Boon, “Inter frame coding with template matching spatio-temporal prediction,” in Proc. IEEE Int. Conf. on Image Processing ICIP’04, Singapore, Oct.2004, pp.465-468.

[8] Steffen Kamp, Michael Evertz, and Mathias Wien, “Decoder side motion vector derivation for inter frame video coding,” in Pro. IEEE Int. Conf on Image Processing ICIP’ 08, San Diego, Oct.2008, pp.1120-1123.

[9] S. Kamp and M. Wien, “Decoder-side motion vector derivation for hybrid video inter coding,” in Proc. of IEEE International Conference on Multimedia and Expo '10, (Singapore), IEEE, Piscataway, July 2010.

[10] Motoharu Ueda, Shigeeru Fukushima, “Refinement motion compensation using decoder-side motion estimation,” JCTVC-B032, Geneva, CH, 21-28 July, 2010.

[11] Yu-Wen Humang, Ching-Yeh Chen, Chih-Wei Hsu, Jian-Liang Lin, Yu-Pao Tsai, Jicheng An, and Shawmin Lei, “Decoder-side motion vector derivation with switchable template matching,” JCTVC-B076, Geneva, CH, 21-28 July, 2010.

[12] Tianmi Chen, Xiaoyan Sun, and Feng Wu, “Predictive patch matching for inter frame coding,” Proc. SPIE, Vol.7744, 774412 (2010).

[13] http://iphone.hhi.de/suehring/tml/download.

[14] G.Bjontegaard, “Calculation of average psnr differences between rd-curves,” Tech. Rep., VCEG-M33, ITU-T Q.6/SG16, Apr 2001.