

PICTURE-LEVEL ADAPTIVE FILTER FOR ASYMMETRIC STEREOSCOPIC VIDEO

Ying Chen¹, Ye-Kui Wang², Miska M. Hannuksela², and Moncef Gabbouj¹

¹Institute of Signal Processing, Tampere University of Technology

²Nokia Research Center

ABSTRACT

In asymmetric stereoscopic video coding, one view is coded in a quarter of the resolution of the other and the low-resolution view is predicted from the high-resolution view. This way, stereoscopic video effect could be achieved with only moderately increased bandwidth and complexity. Inter-view prediction tools for generating the predictor of a macroblock (MB) or MB partition in the low-resolution view from the high-resolution view play a vital role for coding efficiency in asymmetric video coding. In this paper, we propose a method that applies an adaptive filter to generate picture-level adaptive inter-view predictors for MBs or MB partitions. At the encoder, a low complexity preprocessing module is built to find out the filters. Simulation results show that the proposed method provides a bit-rate saving of 26% at maximum and 5% on average.

Index Terms—Multiview video coding, stereoscopic video, asymmetric coding, inter-view prediction, motion compensation, adaptive filter

1. INTRODUCTION

The amount of interest in multiview video technologies has been increasing recently. As views are correlated, view redundancy has been reduced in the multiview video coding (MVC) standard [1], which will become an extension to H.264/AVC. Many display arrangements for multiview video are based on rendering of different images to viewer's left and right eyes. For example, when data glasses or autostereoscopic displays are used, only two views are observed by a viewer at a time in typical multiview applications, such as 3D TV [2], although the scene can often be viewed from different positions or angles. Based on the concept of asymmetric coding, one view in a stereoscopic pair can be coded with lower fidelity, while the perceptual quality degradation for the stereoscopic display can be negligible by human eyes [3]. Stereoscopic video applications therefore seem feasible with moderately increase of complexity and bandwidth on top of mono-view applications, even in the mobile application domain [4].

Asymmetric MVC or asymmetric stereoscopic video (ASV) codec can be realized by an H.264/AVC-compliant base view (view 0) and a lower resolution second view

(view 1) compressed using inter prediction as specified in H.264/AVC as well as inter-view prediction. Approaches have been proposed to downsample base view pictures for inter-view prediction [4][5].

It is favorable to design the coding of low-resolution view in a manner with high efficiency as well as low complexity. A low-complexity motion compensation (MC) scheme has been proposed for asymmetric MVC to reduce the complexity of asymmetric MVC without compression efficiency loss [6]. In [6], direct MC from high-resolution inter-view reference picture to the low-resolution picture was proposed.

2D non-separable adaptive filter has been proposed for interpolating non-integer sample values for inter prediction in H.264/AVC [7]. The difference between inter-view prediction in ASV and inter prediction in H.264/AVC is that, in the former the reference pictures are of larger resolution than the picture to be predicted, and thus contain more information that can be potentially beneficial for inter-view prediction. To efficiently exploit this, we propose an algorithm to motion compensate the low-resolution picture from the high-resolution picture based on picture-level adaptive filters in this paper. A preprocessing module with a minor complexity increase to the encoder is designed to generate the optimal filters. Simulation results show that compared to the direct MC proposed in [6], about 5% bit-rate saving on average and up to 26% bit-rate saving can be achieved for ASV.

2. ASYMMETRIC STEREOSCOPIC VIDEO

A typical prediction structure of stereoscopic video is shown in Fig. 1. Pictures in a view form a hierarchical B temporal prediction structure. Each picture is associated with a temporal level. The base view (view 0, denoted as S0) is independently coded and the other view (view 1, denoted as S1) is dependent on view 0. Note that the MVC Joint Draft (JD) can deal with more views predicted from each other in the view dimension in a hierarchical manner [1].

Typically, in ASV, view 0 is in the original resolution (e.g., VGA) and the view 1 is in a lower (quarter) resolution (e.g., QVGA). The ASV approaches are motivated by the suppression theory of binocular vision [3], which indicates that the perceived sharpness and depth effect of a

stereoscopic pair with different fidelities is dominated by the higher-quality view, which can correspond to the right-eye, for example [4]. A 2D mobile system based on H.264/AVC then can be enhanced to stereoscopic mobile system with acceptable transmission bandwidth and decoder complexity increase, which is around 25%.

To support inter-view prediction between two views with different spatial resolutions, two solutions have been proposed. The first one downsamples the high-resolution views when they are used for inter-view prediction [4, 5] and the second applies a direct MC from the high-resolution pictures [6].

In [6], if the motion vector points to integer or half-sample positions in the virtually downsampled picture as in [4, 5], it will point to even or odd integer sample positions in the high-resolution (view 0) picture. If it points to quarter-sample positions, it will point to half-sample positions in the view 0 picture. As shown in Fig. 2, the samples in a 4x4 block can be predicted from an 8x8 block in the view 0 picture consisting of integer samples. The integer samples with the same parity then form a 4x4 block (each sample is predicted from the sample marked with the same number in the figure). When the motion vector points to half-sample positions in the view 0 picture, two neighboring integer sample values are averaged to get a predicted sample value.

In [6], the process of downsampling of the inter-view picture to the same resolution, as well as the interpolation of values at half-sample positions can be avoided for MC, and there is no extra buffer required to store the downsampled pictures. Storage of the high-resolution picture also preserves more information that can be utilized by potentially advanced inter-view prediction algorithms, e.g. the algorithm proposed in this paper.

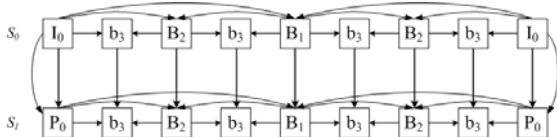


Fig. 1: Typical prediction structure for stereoscopic video.

3. ADAPTIVE FILTER FOR ASYMMETRIC STEREOSCOPIC VIDEO

For multiview content, different camera parameters may lead to differences that can not easily be compensated with conventional MC. Moreover, higher resolution in view 0 can potentially benefit the coding of view 1. Therefore, adaptive filters are designed to get a better predicted signal for the integer samples. A simple average from the filtered integer sample values is still used in the proposed algorithm for non-integer samples.

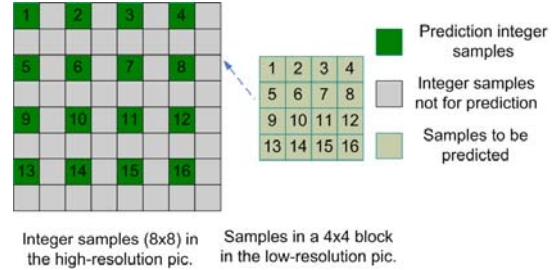


Fig. 2: MC when a motion vector points to integer samples in the inter-view picture (with high resolution).

3.1. Adaptive filter generation

Assume $\mathbf{S} = \{s_p \mid p=1 \dots M\}$ is the set of the samples in a picture of view 1 and the intensity (luma value) of s_p is b_p . For each sample s_p , there is a group of pixels that are located by motion estimation, e.g. using a block matching algorithm, at picture in view 0 in the same time instance. Let the pixel group corresponding to s_p be $R_p = [r_{p1}, r_{p2} \dots r_{pm}]$ and their intensity values be $U_p = [u_{p1} \ u_{p2} \ \dots \ u_{pm}]$. Let the filter applied in inter-view prediction be $H = [h_1, h_2 \dots h_N]^T$, wherein N is the length of the filter and also the number of the pixels in a corresponding pixel group. To get the best prediction of those samples in \mathbf{S} , the following optimization problem is to be solved.

$$H^* = \arg \min_H (e^2) = \arg \min_H \left(\sum_{s_p \in \mathbf{S}} (b_p - U_p \cdot H)^2 \right)$$

The problem can be solved by Least Mean Square (LMS) algorithm as follows: $H^* = \mathbf{U}^+ \cdot \mathbf{b}$, wherein \mathbf{U}^+ is the pseudoinverse of the matrix \mathbf{U} , which is of $N \times M$ and has all the $U_p, p=1 \dots M$ as row vectors and $\mathbf{b} = [b_1 \ b_2 \ \dots \ b_M]$.

There are two remaining issues: how to define the corresponding pixel group and how to construct set \mathbf{S} that includes the pixels considered for optimization.

3.2. Corresponding pixel group localization

As there are disparities between views, the corresponding block in the picture in view 0 is found by block matching. Running the entire encoding process including motion estimation is an accurate way to get the disparity motion vectors. However, it requires multi-pass encoding and thus has shortcomings, e.g., high complexity, since the motion estimation and mode decision invoked in the H.264/AVC as well as MVC encoding includes searches for each reference pictures (several inter prediction reference pictures and one inter-view picture) and different modes of macroblock (MB) partitions and sub-macroblock partitions. So, in this paper, we utilize simple 16x16 block matching algorithm which greatly reduces the complexity. The block matching is

applied for each MB of the low-resolution picture. Only the integer samples in view 0 are searched.

Given a disparity motion vector, the best match sample in the inter-view picture (named center sample in this paper) for a pixel can be located by adding the disparity to the sample position. Two types of non-separable filters are proposed. Type-I filter requires a corresponding pixel group that has only the center sample and the nearest samples with the same parity (odd or even) as the center sample, as shown in Fig. 3 with upper case letters. Type-II filter requires a corresponding pixel group that includes also the samples with different parity, i.e., the center sample as well as all the nearby integer samples, with both upper case letters and the lower case letters in Fig. 3.

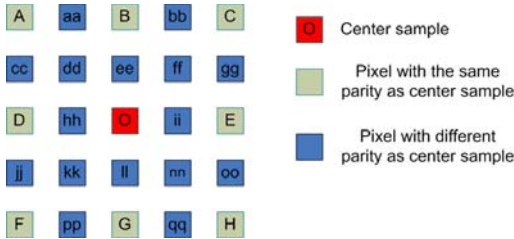


Fig. 3: Corresponding pixel groups for 9 and 25 samples.

3.3. Relevant regions selection for adaptive filter

The optimization problem targets on the least square error for a specific sample set, which tends to use inter-view prediction. As a matter of fact, view 1 is coded in a hybrid way, which enables not only inter-view prediction (that utilizes adaptive filters) but also conventional H.264/AVC (intra-view) modes: inter prediction and intra prediction. The MBs or MB partitions for which intra-view modes are selected cannot benefit from the adaptive filter and thus allowing them be considered for adaptive filter generation can lead to a less optimal filter, which is less sensitive to the prediction errors of those samples finally predicted by inter-view prediction. So, relevant regions need to be well defined before the optimization equation is built. To guarantee that the adaptive filter is generated only by the prediction errors of the MBs for which inter-view prediction mode is preferred, the following function is proposed to select the MBs for the filter optimization.

$$f(MB_t) = \begin{cases} 1 & \text{Distortion}(MB_t) \leq T \\ 0 & \text{else} \end{cases}$$

wherein $f(\cdot)$ equal to 1 indicates that the t -th MB is selected for generation the adaptive filter and $Distortion(\cdot)$ returns the distortion of an MB. So we have

$$S = \{s^i | s^i \in MB_t, f(MB_t) = 1\}$$

For simplicity, MB_t denotes the t -th MB in the picture in view 1. The thresholded T is content dependent and can vary picture by picture.

In this paper, T is set as follows to satisfy $Rate = |D| / NumMB, D = \{MB_t | f(MB_t) = 1\}$, wherein $||$ returns the number of elements in a set. When the *Rate* (percentage) of MBs that are used for inter-view prediction is decided, the threshold T can be decided by ordering the distortion values of all the MBs in a picture. Details of how to set different *Rate* values will be discussed in Section 4.

4. IMPLEMENTATION AND SIMULATION

The adaptive filter generation is applied by the following steps: 1. disparity estimation, 2. relevant MB selection, 3. construction and solving of the LMS equation. As step 1 and step 3 can follow canonical processes, in this section, we only describe more on the step 2.

As can be observed, inter-view prediction benefits pictures in different extents. Many factors influence this, but one of the most obvious factors is the temporal level (TL) of the pictures. Pictures with lower temporal levels tend to have more MBs predicted from the other view, because the temporal distance to intra-view reference for lower temporal levels is larger than for higher temporal levels. Thus, a simple method to get the *rate* values and therefore the threshold T is proposed by adopting the following temporal level to *rate* table (Table 1) for each sequence.

Table 1. Temporal level to *rate* table

TL	0	1	2	3	4 or higher
<i>Rate</i>	0.9	0.8	0.6	0.4	0.2

The proposed algorithm was implemented into the MVC reference software, JMVM (Joint Multiview Video Model) version 5 [8]. The tested sequences were *Exit*, *Ballroom*, *Rena*, *Race1*, *Akyo&Kayo*, *Breakdancers* and *Flamenco2*. For each video set, the first two views are selected to be coded as view 0 and view 1 in our simulation. Other parameters, e.g., the temporal prediction structure, the search range and the number of reference frames follow the common test condition of MVC [9].

The low-resolution input views were generated by utilizing the MPEG-4 downsampling filter, which is $[2, 0, -4, -3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64$. After decoding, the low-resolution video was upsampled by the H.264/AVC interpolation filter $([1, -5, 20, 20, -5, 1]/32)$ for peak signal-to-noise (PSNR) calculation.

The rate distortion (RD) performance comparison for the type-I filter (ASV-I), type-II filter (ASV-II) and the original ASV (ASV-O) [6] are compared and the results are listed in Table 2. Note that, in the table, a bit-rate saving or Δ PSNR value greater than zero indicates that the algorithm of the left is better than the one on the right. Results were generated using the Bjontegaard measurement [10] based on the bit-rate and average PSNR values of the four test points corresponding to different QP values.

For the *ballroom* sequence, the RD curves of view 1 are shown in Fig. 4. The performance increases resulting from the use of type-I and type-II filters are noticeable.

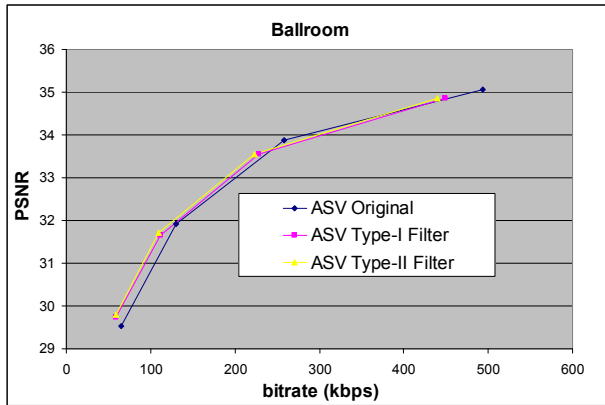


Fig. 4: RD curves for “Ballroom”.

Table 2. Comparison of ASV-II filter to ASV-II and ASV-O (view 1 only)

Sequence	ASV-I vs ASV-O		ASV-II vs ASV-O	
	Bit-rate saving	Δ PSNR	Bit-rate saving	Δ PSNR
<i>Akyo & Kayo</i>	-4.85%	-0.172	-3.32%	-0.120
<i>Ballroom</i>	5.82%	0.134	9.11%	0.221
<i>Exit</i>	-1.80%	-0.041	0.28%	0.003
<i>Race1</i>	-10.39%	-0.216	-6.07%	0.118
<i>Rena</i>	21.76%	0.622	26.67%	0.755
<i>Breakdancers</i>	6.42%	0.144	5.94%	0.137
<i>Flamenco2</i>	2.16%	0.063	1.07%	0.042
Average	2.73%	0.076	4.81%	0.165

Although the filters have many taps, they are applied only once to get the prediction value for each sample. So, if the related samples are in a small region (e.g., 5x5), the complexity of MC can be comparable to the H.264/AVC half-sample positions MC, which requires interpolation that may be related to a region as large as 6x6 for a sample.

5. DISCUSSION AND FUTURE WORK

As shown in Table 2, the proposed method can give an average bit-rate saving of 4.81%. However, it also introduces loss to some sequences. The proposed method relies on the distribution of the disparity motion vectors. If the picture contains different depth-levels, e.g., as shown in Fig. 5 for *Race1*, the adaptive filter for the global picture is not optimal and will even blur the signal and provide worse performance. Another typical distribution of disparity motion vectors for *Rena* indicates a highly converged single depth level and high efficiency is observed for this sequence. To further improve coding efficiency for those

sequences with multiple depth levels, multiple adaptive filters can be utilized for different regions.

6. CONCLUSION

Picture-level adaptive filter was proposed to generate a better predicted signal in inter-view prediction for asymmetric stereoscopic video coding, wherein the second view is coded in a quarter resolution compared to that of the H.264/AVC-compliant base view. In the method, relevant macroblocks for filter generation are selected based on their distortions as well as the associated temporal levels. Two types of filters were proposed, the first one only uses samples with the same parity (odd or even) with 2.7% bit-rate saving and the second one fully utilizes odd and even integer samples, with 2.1% additional bit-rate saving on average on top of the first type.

7. REFERENCES

- [1] “Joint Draft 5.0 on Multiview Video Coding,” *JVT-Y209*, Shenzhen, China, Oct. 2007
- [2] A. Vetro, W. Matusik, H. Pfister, J. Xin, “Coding Approaches for End-to-End 3D TV Systems,” *Picture Coding Symposium*, 2004
- [3] Julesz B., *Foundations of Cyclopean Perception*, University of Chicago Press, Chicago, IL, USA, 1971
- [4] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, J. Kim, “Asymmetric coding of stereoscopic video for transmission over T-DMB,” *Proc. 3DTV-CON 2007*, Kos Island, Greece, May 2007
- [5] H. Kimata, S. Shimizu, K. Kamikura, Y. Yashima, “Inter-view prediction with downsampled reference pictures,” *JVT-W079*, San Jose, CA, USA, Apr. 2007
- [6] Y. Chen, S. Liu, Y.-K. Wang, M. M. Hannuksela, H. Li and M. Gabbouj, “Low complexity asymmetric multiview video coding,” *IEEE Proc. ICME*, 2008.
- [7] Y. Vatis, B. Edler, D. T. Nguyen, J. Ostermann, “Motion and Aliasing-Compensated Prediction Using a Two-dimensional Non-separable Adaptive Wiener Interpolation Filter,” *ICIP 2005*
- [8] “JMVM 5 software,” *JVT-X208*, Geneva, Switzerland, Jun.-Jul. 2007
- [9] “Common Test Conditions for Multiview Video Coding,” *JVT-T207*, Klagenfurt, Austria, Jul. 2006
- [10] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” *VCEG-M33*, Mar., 2001

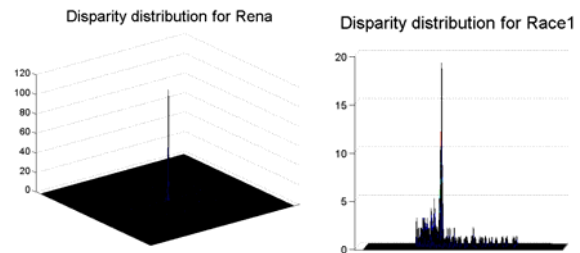


Fig. 5: Histograms of the disparities for “Rena” and “Race1”.