ADAPTIVE REFERENCE FILTERING FOR BIDIRECTIONAL DISPARITY COMPENSATION WITH FOCUS MISMATCHES

PoLin Lai[*], *Antonio Ortega*

Ming Hsieh Dept. of Electrical Engineering
Signal and Image Processing Institute
University of Southern California
Los Angeles, CA 90089

Purvin Pandit, Peng Yin, Cristina Gomila

Thomson Corporate Research
2 Independence Way
Princeton, NJ 08540

ABSTRACT

In this paper, we consider compensation of focus mismatches for frames that are encoded with inter-view bi-prediction (B-frames) in multiview coding (MVC). We start with an analysis of a multiview system with focus mismatches, to demonstrate that a B-frame may suffer from different types of mismatches with respect to the frames from different views used as references. As compared to our previous work for inter-view P-frames, filter estimation for B-frames has to consider not only the depth-dependency of focus mismatches, but also i) the possibility that the two predictors, from different directions, exhibit different types of prediction mismatches, and ii) the effect of bi-predictive search on the generation of filtered references. We show that, designing filters only for the averaged bi-predictor could lead to a suboptimal solution when combined with conventional bi-predictive search schemes. Instead, we propose a filter design approach that independently estimates depth-related filters for the two references used for prediction. Simulation results shows that for views coded with inter-view bi-prediction, the proposed method provides up to 0.7 dB gain over current H.264/AVC in the sequences we tested.

Index Terms— multiview video coding, bi-prediction, focus mismatches, adaptive filtering, disparity compensation

1. INTRODUCTION

In multiview video systems, multiple cameras are utilized to simultaneously capture scenes from different viewpoints. Due to differences in camera settings and/or shooting positions, frames from different views are prone to suffer from mismatches other than simple displacement. When encoding across-views (inter-view coding), the efficiency of block-based disparity compensated prediction can suffer the presence of these non-translational mismatches.

Previously, we proposed a depth-related adaptive reference filtering (ARF) approach [1, 2] to compensate for focus mismatch in multiview systems, which results in blurriness/sharpness discrepancy among different views. In the proposed coding scheme, after an initial disparity search, a frame S is partitioned into regions S^1, S^2, ... S^k corresponding to different depth levels (where classification is based on block-wise disparity vectors (DVs)). For each region (depth level) S^i, a parametric 2D spatial filter ψ^i is estimated by minimizing the mean-squared prediction error between S^i and the corresponding block-wise predictors found in the initial search. The resulting filters are applied to the reference frame to create filtered references. Finally in the encoding stage, each block in S selects the predictor (filtered or unfiltered) that provides the lowest rate-distortion cost (RD-cost), thus ensuring highest coding efficiency. This method was developed for inter-view P-frames, for which a single reference frame is used, taken from one of the neighboring views (IPPP for coding V0∼V3 for example).

In this paper, we extend compensation of focus mismatches to B-frames, where predictive coding is performed by using reference frames from two reference lists (e.g., frames from the left and right views in List 0 and List 1, respectively). A straightforward extension of ARF to B-frames can be achieved by designing depth-dependent filters ψ^i_{Bi} that minimize the prediction error between current blocks and the chosen bi-predictors, which will be obtained by averaging two reference blocks, one from each reference list. Note that such an extension would be analogous to that selected for bi-prediction in adaptive interpolation filtering (AIF) [3], in which for a given interpolation position, only one filter is designed and is applied to generate interpolated pixel values for references in both List 0 and 1.

After implementing this straightforward ARF approach, we observed experimentally that coding performance showed no significant improvements; in particular, as compared to previous ARF for P-frames, filtered frames were not chosen as often in bi-prediction scenarios. In this paper, we analyze the causes of the differences in performance between P-frames (for which ARF provides significant gains) and B-frames. We propose alternative filter design techniques that allow us to obtain substantial gains for the bi-predictive case as well. The key observation is that with the above described approach, *joint* filter design is followed by conventional *independent* search for predictors in each list. Because of this mismatch between filter design and search, the gain with respect to un-filtered bi-prediction is minimal. As an alternative, we propose a simple independent filter design that leads to increased gains of up to 0.3 dB as compared to the straightforward filter design for the averaged predictors. As a result, we achieve coding gains up to 0.7 dB gain over current H.264/AVC in the sequences we tested.

The remainder of this paper is organized as follows: In Section 2, we provide an analysis of focus mismatch in inter-view bi-prediction scenario. The proposed filter estimation method is presented in Section 3, along with discussion of the interaction between filter design and bi-predictive search. Simulation results are presented in Section 4. Finally, we conclude this work in Section 5.

[*] Further author information: Send correspondence to polinlai@usc.edu

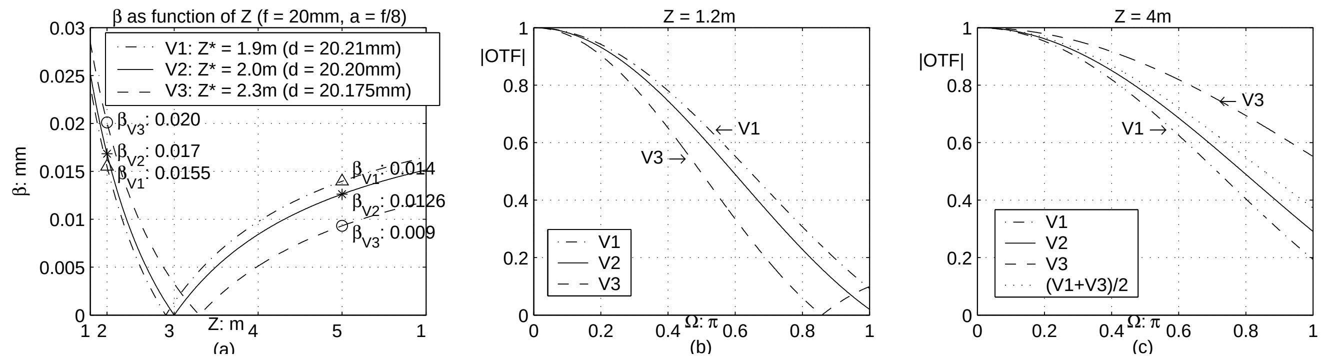

Fig. 1. An example of focus mismatches in multiview bi-prediction, with $Z^*_{V1} = 1.9$m, $Z^*_{V2} = 2.0$m, and $Z^*_{V3} = 2.3$m. We consider image sensor type 1/2" (H×W = 6.4mm×4.8mm) with a resolution of 640×480 pixels, i.e. the spacing between pixels is 0.01mm (Nyquist rate 100/2 = 50 cycles/mm). In polar system, $q = \sqrt{50^2 + 50^2} \approx 70.71$, which corresponds to $\Omega = \pi$ in (b) and (c).

2. INTER-VIEW BI-PREDICTION WITH FOCUS MISMATCHES

A digital camera is typically modeled as an imaging system consisting of a lens with focal length f, an aperture with diameter a, and a "film" made up with an array of image sensors. The plane containing the film is referred as the "image plane". The distance between the image plane and the lens is called the "image place distance", which we denote as d. According to geometrical optics, a visible point will produce a point projection (perfectly focused) on the image plane only if it is at a particular depth Z^* that satisfies:

$$\frac{1}{Z^*} + \frac{1}{d} = \frac{1}{f} \;\Rightarrow\; Z^* = \frac{d \cdot f}{d - f} \tag{1}$$

With a fixed zoom set by f, we can focus on a specified distance Z^* by fine tuning d ($d \geq f$). Operating in a very narrow range, a slight change in d can cause relatively large variation in Z^*. This can be achieved by using autofocus (AF), or by manually adjusting the focus ring. For points at other distances, the corresponding projections on the image plane will be uniform circles with diameter β, which can be derived as [4]:

$$\beta = \frac{af\left(|Z - Z^*|\right)}{Z\left(Z^* - f\right)} \tag{2}$$

It can be seen from (2), that the characteristics of the camera will be affected by parameters a, d, f, and the object depth Z. Now let us consider an example with three cameras V1, V2, and V3, in a multiview system: Assume they have the same focal length setting f (same zoom), and their aperture settings are also identical: $a = f/8$. However, the fine tuning of their Z^* was not done perfectly ($Z^*_{V1} \neq Z^*_{V2} \neq Z^*_{V3}$), resulting in differences of their β values as functions of Z. Fig.1 shows such an example with heterogeneous settings. To illustrate the effect of the differences in β, we plot the optical transfer function (OTF), which is the frequency transform of the point spread function (PSF) specified by β. That is, in the polar coordinates system [5]:

$$\text{PSF}_{(r)} = \begin{cases} 4/(\pi\beta^2), & \text{if } r^2 \leq (\beta/2)^2 \\ 0, & \text{otherwise} \end{cases} \;\rightarrow\; \text{OTF}_{(q)} = \frac{2J_1(\pi\beta q)}{\pi\beta q} \tag{3}$$

In (3), J_1 is the Bessel function of the first kind of order 1. Fig.1(b) and (c) show the differences in the corresponding OTF. If we encode V2 with bi-prediction by putting V1 in List 0 and V3 in List 1 as references, for image portions correspond to visible regions at Z = 1.2m, we need to perform lowpass on V1 and enhancement on V3 in order to match V2. On the other hand, for visible regions at Z = 4m, the corresponding image portions in V1 need to be slightly "sharpened" while V3 has to undergo a significant amount of lowpass filtering. As for the averaged predictor $\frac{1}{2}(V1 + V3)$ (dotted line) in Fig.1(c), a lowpass filter is required to bring down the curve to that of V2.

If V1 V2 V3 are arranged on a 1-D horizontal line from left to right with equal spacing b between each other, with their image plane distance d being very similar, it can be derived [6] that an object at depth Z will result in a disparity $\delta_Z = \frac{d}{Z}(-b)$ from V1 to V2 and also from V2 to V3. Without direct measurement of depth, we can exploit disparity vectors as estimation of scene depth to identify image portions corresponding to different depth levels and to achieve depth-dependent filter design [1, 2]. In Section 3, we will discuss adaptive filtering methods using the three-view we just discussed.

3. ADAPTIVE REFERENCE FILTERING AND BI-PREDICTIVE DISPARITY SEARCH

From the analytical results, to compensate for focus mismatches, an adaptive filtering approach can be developed by partitioning images into regions at different depth levels and designing filters to minimize the prediction error for each level. We again propose to utilize a two-pass coding scheme with an initial search (the first coding pass) to obtain the block-wise disparity vectors (DVs) and predictors, for disparity-based frame partition and for designing filters. In what follows, we will discuss different filter estimation methods, especially emphasizing on their interaction with bi-predictive search when filtered references are generated.

3.1. Filter design for averaged bi-predictor

In bi-prediction, the predictor for a given block is actually the average of two reference blocks, one from the reference frame in List 0 (R^{L0}) and one from the reference frame in List 1 (R^{L1}). A straightforward filter design approach, which minimizes the prediction error between current blocks and the averaged predictors, can be summarized as:

For pixels within a given depth level i,

$$\min_{\psi_{Bi}^i} \sum_{x,y} \left(S_{x,y} - \psi_{Bi}^i * \frac{1}{2}(R^{L0}_{x+dx0,y+dy0} + R^{L1}_{x+dx1,y+dy1}) \right)^2 \tag{4}$$

In (4), (x, y) is the pixel position within a frame, $(dx0, dy0)$ and $(dx1, dy1)$ are the disparity vectors for R^{L0} and R^{L1} respectively, and $*$ denotes convolution. The frame-partition can be achieved by classifying the DVs in either direction, or by taking both directions as two input features for classification. Since for each depth-level the filter is designed for the averaged predictors, it should be applied to both List 0 and 1, thus filtered references $\psi_{Bi} * R^{L0}$ and $\psi_{Bi} * R^{L1}$ can be generated.

The limitation of the approach in (4) is that there is no guarantee that searching for the best matching blocks in $\psi_{Bi} * R^{L0}$ and $\psi_{Bi} * R^{L1}$ will lead to an optimal solution to the problem of finding the two blocks in List 0 and List 1 that provide the best prediction after averaging *and* filtering. Clearly, this is also the case even if no filtering is used [7]. However, our experiments indicate that the suboptimality of independently searching is exacerbated when filtering is used.

Consider first the case of independent search, where for each block, the encoder *independently* searches for one best predictor from references in List 0 and one from references in List 1. The bi-predictor is formed by simply averaging the two without performing any additional search. As for the example in Fig.1(c), during the search within List 0, due to the effect of the lowpass filter ψ_{Bi}, the reference $\psi_{Bi}*$V1 is not preferred over V1, i.e. it is less likely to be selected. Consequently, the improved predictor $\frac{1}{2}\psi_{Bi}*$(V1+V3) may not even be tested by the encoder.

As an alternative, in the iterative search [7], the search is conducted by, iteratively, fixing the obtained predictor from one side ($R^{L0/L1}$) to estimate the best predictor from the the other side ($R^{L1/L0}$). This can help alleviate the disadvantage in independent search, as some joint estimation is made possible. However the iterative process could still be trapped in local minimum. For example in Fig.1(c), if the initial selected predictor from List 0 is V1 instead of $\psi_{Bi}*$V1, the resulting predictor after iterations may not converge to the optimal predictor $\frac{1}{2}\psi_{Bi}*$(V1+V3).

One possible approach to resolve such problem is to modify bi-predictive search as follows: For the search within each list, instead of picking only a single "best" predictor, record the best matched predictors from each reference ($R^{L0}, \psi_{Bi}^i * R^{L0} \ldots ;\ R^{L1}, \psi_{Bi}^i * R^{L1} \ldots$). With different combinations of one predictor from each side, multiple averaged predictors can then be evaluated. While complexity is increased, for a given depth-level k, still only $\frac{1}{2}\psi_{Bi}^k * (R^{L0} + R^{L1})$ corresponds to the focus compensated predictor.

In addition to the problems related to the search algorithm, if the filters are designed jointly for averaged blocks there is no guarantee that after applying them to individual frames they will provide good approximations to the original frame (which explains why filtered frames are rarely selected when (4) is used.) As an example, consider Fig.1(c), after applying the lowpass filter ψ_{Bi} designed for $\frac{1}{2}$(V1+V3), the new reference $\psi_{Bi}*$V1 will actually has stronger mismatch to V2 as its frequency response is further brought down from that of V1.

3.2. Filter design for predictors from each reference list

To overcome the drawbacks (limited coding choices, integration with bi-predictive search) of the method in (4), we consider an alternative filter design approach that independently estimates depth-related filters for each reference list. After the first coding pass, we partition the current frame S into $S^{L0,1}$, $S^{L0,2} \cdots\ S^{L0,M}$ based on classification of $(dx0, dy0)$. We also partition it into $S^{L1,1}$, $S^{L1,2} \cdots\ S^{L1,N}$ based on classification of $(dx1, dy1)$. By recording *separately* the pixel values of the reference blocks from List 0 and List 1 (instead of minimizing error with respect to the averaged predictor), two sets of filters can be estimated as follows:

$$\Psi_{L0} = \left\{ \psi_{L0}^i \,\middle|\, \min_{\psi_{L0}^i} \sum_{x,y} \left(S_{x,y}^{L0,i} - \psi_{L0}^i * R^{L0}_{x+dx0,y+dy0} \right)^2 \right\}$$

$$\Psi_{L1} = \left\{ \psi_{L1}^j \,\middle|\, \min_{\psi_{L1}^j} \sum_{x,y} \left(S_{x,y}^{L1,j} - \psi_{L1}^j * R^{L1}_{x+dx1,y+dy1} \right)^2 \right\} \tag{5}$$

This filter design method directly addresses the potentially different types of depth-dependent mismatches exhibited in reference frames from List 0 and 1, such as the example depicted in Fig.1. In (5), set Ψ_{L0} will contain M filters and Ψ_{L1} will have N filters. They will be applied to List 0 and List 1 respectively to generate filtered references. Note that in this approach, a given block in S will participate in both filter estimations to minimize prediction errors with respect to references in List 0 and 1. As compared to the method in Section 3.1, the two sets filter design has the following advantages:

1. Better integration with conventional bi-predictive search schemes: Since in both lists the focus compensated references are generated, the search within each list is likely to obtain better matched predictor. As a results, the averaged bi-predictor would also be an improved one.
2. More coding options: For B-frame, a block can simply be encoded using predictor from only one of the lists, if the rate-distortion (RD) cost of doing so is smaller than using the averaged bi-prediction. Based on (5), the filtered references in each list provide better matched predictors that can be used by themselves, leading to more options for encoder to perform RD optimization.
3. Potential speed up for pi-predictive search: Consider the example as Fig.1 in which two filters are designed (for depth 1.2m and 4m) in each reference list ($\psi_{L0}^{1.2m}$, ψ_{L0}^{4m}, and $\psi_{L1}^{1.2m}$, ψ_{L1}^{4m}). If we observe that a given block selects $\psi_{L0}^{4m} * R^{L0}$ after the search within List 0, it is reasonable to constrain the search in List 1 to the reference $\psi_{L1}^{4m} * R^{L1}$. From the analytical results, the degradation in coding efficient should be small as this is likely to be the best matched reference.

Thus, without modifying the bi-predictive search schemes and increasing complexity, this method is preferred as compared to the joint estimation in Section 3.1.

3.3. Hybrid filter design

Finally, we can consider applying both the methods as in Section 3.1 and 3.2, resulting in three sets of filters: Ψ_{L0}, Ψ_{L1} and Ψ_{Bi}. The first two will be applied to List 0 and 1 respectively. On the other hand, Ψ_{Bi} should be applied to both list.

While the references filtered by sets Ψ_{L0} and Ψ_{L1} can be readily used, as discussed in Section 3.1, references generated by applying Ψ_{Bi} have to be treated with special consideration. In order to fully exploit the advantage of Ψ_{Bi}, the bi-prediction search scheme has to be modified such that predictors $\frac{1}{2}\psi_{Bi}^i * (R^{L0} + R^{L1})$ can still be tested even if $\psi_{Bi}^i * R^{L0/L1}$ alone might not provide higher coding efficiency. As a results, a properly implemented hybrid filter design will have the highest complexity among the three methods discussed in this Section 3, especially with more filtered references to search over and the additional step to evaluate more combinations for bi-predictors.

4. SIMULATION RESULTS

The proposed approaches are integrated with the JMVM 5.0, which is a software implementation dedicated for multiview video coding based H.264/AVC. The classification of DV for frame-partition is performed using a tool [8] based on Gaussian Mixture Model. We partition a frame into up to three depth-level and estimate the corresponding filters. According to the analysis in Section 2, 5×5 filters with circular symmetric constraint are used. We encode frames only at given timestamps using inter-view coding with IBPBP structure. The interval between two timestamps is 0.5 sec. (e.g. Inter-view coding at every 12th frame for frame rate 25fps.)

Without making any modification to the bi-predictive search schemes, we currently performed simulations based on methods in Section 3.1 and 3.2 using iterative search. (Initial search range ±64, plus 4 iterations with refinement search range ±8.) For the sequences tested, the independent filter design (Section 3.2) achieves coding efficiency which is up to 0.3 dB higher than the method in Section 3.1. Thus in Fig.2, we provide the corresponding rate-distortion results of independent filter design. The four rate points correspond to QP 22, 27, 32, and 37.

It can be seen that, **for views encoded with bi-prediction**, the sequence Race1 achieves 0.5∼0.7 dB gain when applying proposed independently designed ARF; while the improvement is about 0.3∼0.4 dB for Rena. The higher efficiency comes with a penalty with increased complexity introduced by the 2-pass ARF coding scheme. However, we have demonstrated that [9], by evaluating the RD performance across views and comparing the depth-composition across time, complexity reduction techniques can be developed, without sacrificing coding efficiency, such that ARF is applied only to views with substantial coding gain and the filters are only estimated when scene depth changes (instead of at every timestamps). We expect the same results, i.e. negligible degradation in coding efficiency, can also be achieved when applying those techniques to ARF for bi-prediction.

5. CONCLUSIONS

This work considers compensating focus mismatches for frames that are encoded with inter-view bi-prediction in multiview coding. We analyze a multiview system with focus mismatches to demonstrate different types of mismatches as compared to the reference frames from different views. We show that the filter design approach for the averaged bi-predictor leads to a suboptimal solution when combined with conventional bi-predictive search schemes. Taking into account the interaction between filter design and the bi-predictive search with filtered references, we proposed filter estimation method which independently design depth-related filers for each reference list. Simulation results shows that for views coded with inter-view bi-prediction, the proposed method provides up to 0.7 dB gain over current H.264/AVC in the sequences we tested.

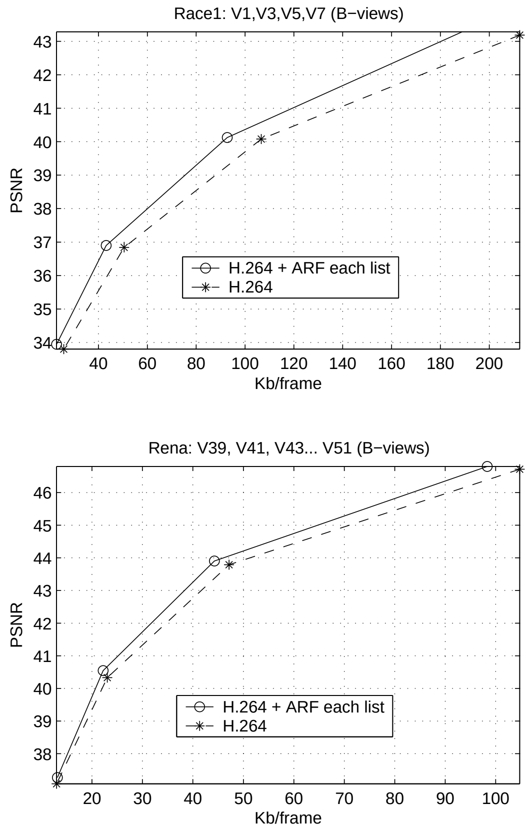

Fig. 2. Rate-Distortion performance of the proposed ARF

6. REFERENCES

[1] J.-H. Kim, P. Lai, J. Lopez, A. Ortega, Y. Su, P. Yin, and C. Gomila, “New coding tools for illumination and focus mismatch compensation in multi-view video coding,” *IEEE Trans. Circuits Systems and Video Technologies (CSVT)*, vol. 17, no. 11, pp. 1519–1535, Nov 2007.

[2] P. Lai, Y. Su, P. Yin, C. Gomila, and A. Ortega, “Adaptive filtering for cross-view prediction in multi-view video coding,” in *Proc. SPIE 2007 Visual Communications and Image Processing (VCIP)*, Jan 2007.

[3] Y. Vatis and J. Ostermann, “Prediction of P- and B-frames using a two-dimensional non-separable adaptive Weiner interpolation filter for H.264/AVC,” *ISO/IEC-JTC1/SC29/WG11 MPEG Document M13313*, Apr 2006.

[4] H.-C. Lee, “Review of image-blur models in a photographic system using the principles of optics,” *SPIE Optical engineering*, vol. 20, issue. 5, pp. 405–421, May 1990.

[5] R. N. Bracewell, *The Fourier Transform and Its Applications*, McGRAW-HILL, 3rd edition, 2000.

[6] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2003.

[7] M. Flierl, T. Wiegand, and B. Girod, “A locally optimal design algorithm for block-based multi-hypothesis motion-compensated prediction,” in *Proc. IEEE Data Compression Conference 1998 (DCC)*, Mar 1998, pp. 239–248.

[8] C. A. Bouman, “Cluster: An unsupervised algorithm for modeling Gaussian mixtures,” *http://cobweb.ecn.purdue.edu/ bouman/software/cluster/*, the version we used was released in Jul. 2005.

[9] P. Lai, A. Ortega, P. Pandit, P. Yin, and C. Gomila, “Focus mismatches in multiview systems and efficient adaptive reference filtering for multiview video coding,” in *Proc. SPIE 2008 Visual Communications and Image Processing (VCIP)*, Jan 2008.