Adaptive filtering for cross-view prediction in multi-view video coding

PoLin Lai[a,b], Yeping Su[a], Peng Yin[a], Cristina Gomila[a] and Antonio Ortega[b]

[a]Thomson Corporate Research, 2 Independence Way, Princeton, NJ 08540
[b]Signal and Image Processing Institute, Univ. of Southern California, Los Angeles, CA 90089

ABSTRACT

We consider the problem of coding multi-view video that exhibits mismatches in frames from different views. Such mismatches could be caused by heterogeneous cameras and/or different shooting positions of the cameras. In particular, we consider focus mismatches across views, i.e., such that different portions of a video frame can undergo different blurriness/sharpness changes with respect to the corresponding areas in frames from the other views. We propose an adaptive filtering approach for cross-view prediction in multi-view video coding. The disparity fields are exploited as an estimation of scene depth. An Expectation-maximization (EM) algorithm is applied to classify the disparity vectors into groups. Based on the classification result, a video frame is partitioned into regions with different scene-depth levels. Finally, for each scene-depth level, a two-dimensional filter is designed to minimize the average residual energy of cross-view prediction for all blocks in the class. The resulting filters are applied to the reference frames to generate better matches for cross-view prediction. Simulation results show that, when encoding across views, the proposed method achieves up to $0.8dB$ gain over current H.264 video coding.

Keywords: Multi-view video coding, adaptive filtering, disparity compensation, disparity segmentation

1. INTRODUCTION

In multi-view video systems, scenes are captured simultaneously from multiple cameras. These cameras are set to shoot the scenes from different view points. They provide digital video data that could be useful in several applications, such as surveillance systems, on-demand telecommunications, entertainment, and immersive virtual reality. Multi-view video contains very large amounts of data as compared to monoscopic video, with the amount of added data increasing with the number of views. Multi-view video coding (MVC) has recently become an active research area[1,2] focused on compression for efficient storage and transmission of multi-view video data.

Simulcast is a straightforward coding scheme for multi-view video in which each view sequence is encoded independently. This allows temporal redundancy to be exploited, using standard block-based motion compensation techniques. In a multi-view video scenario, there exists an additional source of redundancy, namely, cross-view redundancy. Similar to motion compensation, the block matching procedure can be employed to find block correspondence from view to view, through disparity compensation. Exploiting both temporal and cross-view redundancies achieves higher coding efficiency as compared to simulcast.[3] Fig. 1 depicts a prediction structure with both motion and disparity compensation. To facilitate random access, with a certain time interval, video frames are encoded with only cross-view prediction, i.e. no temporal references are needed. We denote such frames "anchor frames", as illustrated in Fig. 1. Similar to I frames in monoscopic video, these anchor frames serve as temporal access points for multi-view video.

Block matching techniques are very efficient when the video scene undergoes only displacement with no other transformation. For cross-view prediction in MVC, the displacement is the disparity that results from different shooting positions of the cameras. However, video frames from different views are prone to suffer from mismatches other than simple displacement. Firstly, multi-view video systems could be built with heterogeneous

Further author information:
PoLin Lai: polinlai@usc.edu, Antonio Ortega: ortega@sipi.usc.edu
Yeping Su, Peng Yin, Cristina Gomila: {yeping.su, peng.yin, cristina.gomila}@thomson.net

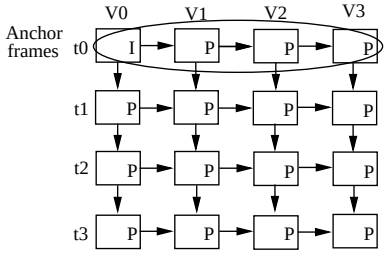


Figure 1. MVC with both motion and disparity compensations

cameras, or cameras that have not been perfectly calibrated. This can cause frame-wise (global) mismatches among different views. For example, frames in one view may appear blear as compare to frames from the other view, due to mis-calibration. Secondly, objects may appear differently due to shooting positions that are used. Consider the camera arrangement in Fig. 2, object A will possess a larger scene-depth (z_1) in view 1 than in view 3 (z_3). Even if all cameras are calibrated with the same focus at scene depth z_1, object A appears in focus in view 1 while it is de-focused (blurred) in view 3. On the other hand, object B will become sharpened in view 3 as compared to in view 1. For this focus mismatch example, different portions of a video frame can undergo different blurriness/sharpness changes with respect to the corresponding areas in frames from the other views (localized focus mismatches). These two factors lead to discrepancies among video sequences in different views. The efficiency of cross-view disparity compensation could deteriorate in the presence of these mismatches.

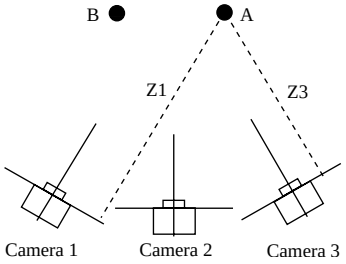


Figure 2. Camera arrangement that causes local focus mismatches

For monoscopic video coding, reference frame filtering approaches have been proposed to improve motion compensation performance.[4–7] For camera panning and/or focus changes, Budagavi proposed blur compensation,[4] where a fixed set of blurring (lowpass) filters are used to generate blurred reference frames. This technique has two shortcomings for the scenarios we consider. First, the filter selection is made only at the frame-level, i.e., applying different filters to different parts of a frame was not considered. For the focus mismatch example described above, adaptive local compensation should be exploited. Second, this method relies on a predefined filter set, which does not include certain filters, e.g., sharpening filters (high frequency enhancement), which may be useful in some scenarios. Instead, our approach generates filters based on the mismatches between the reference frame and the current frame; thus allowing blurring and sharpening filters to be used, with further extensions possible, e.g., using directional filters. Adaptive filtering methods have been proposed to address aliasing in generating subpixel references for motion compensation.[5–7] Vatis et al. 7, after an initial motion search using the 6-tap interpolation filters defined in H.264,[8] divide blocks in the current frame into groups exclusively based on the *subpixel positions** of their motion vectors. For each group, an adaptive filter is estimated. The subpixels of the reference frame will then be interpolated by these adaptive filters. In the final motion compensation, the encoder chooses the best match by testing different subpixel positions on the same reference frame. This

*For example, $(1\frac{3}{4},\ 23\frac{1}{2})$ and $(45\frac{3}{4},\ 6\frac{1}{2})$ will be assigned to the same group. Such a design particularly targets the aliasing problem and motion estimation error when generating subpel reference.

approach, which we will refer to as adaptive interpolation filtering (AIF), is designed for subpixel interpolation and does not directly address the cross-view mismatches that are our target in this paper.

In the paper, we propose a novel adaptive filtering approach for cross-view disparity compensation in MVC. Contrary to AIF, video frames are first divided into regions with different scene-depth levels. The disparity fields are exploited as an estimation of scene depth. For each scene-depth level, a 2D filter is calculated to compensate for the cross-view mismatch by minimizing the residue energy. To provide better coding efficiency, multiple versions of the filtered reference are generated after applying the adaptive filters. For each block, the encoder selects the filter that gives the lowest matching cost. In Section 2, we first demonstrate an example of view-wise mismatch in multi-view video and justify that adaptive filtering is applicable in such a scenario. The proposed method will then be described in detail in Section 3. Simulation results based on H.264 are summarized in Section 4. Finally, conclusions are provided in Section 5.

2. CROSS-VIEW DISCREPANCY AND ADAPTIVE FILTERING

Among the multi-view video test sequences provided in the initial MVC Call for Proposals document,[1] the sequence "Race1" exhibits the most clearly perceivable discrepancy among different views. It consists of 8 parallel views with a 20cm spacing between each, which we will denote as View 0∼View 7. The frames in View 3 are blurred as compared to the frames in View 2; similarly, the frames in View 5 are blurred as compared to the frames in View 6. Fig. 3 shows portions of the frames from different views in Race1.

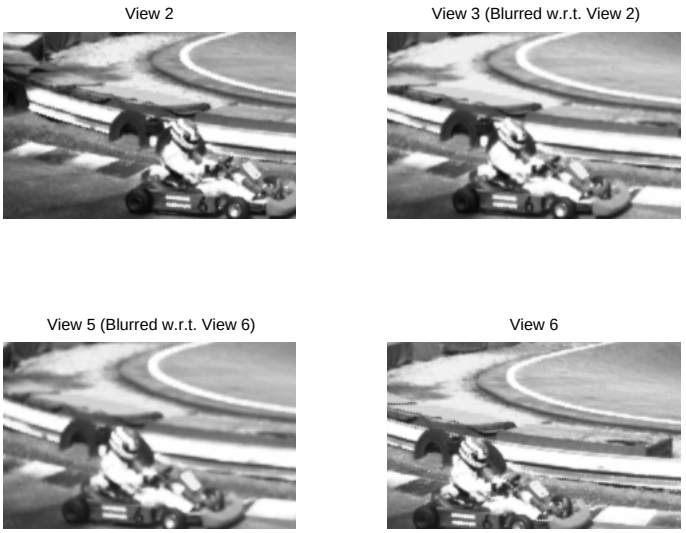


Figure 3. Portions of frame 15 from different views

It can be seen from Fig. 3 that, besides displacement of the scene, frames from different views also possess blurriness mismatches. To compensate for these effects, 2D spatial filters such as smooth and sharpening filters can be considered to create a better match. We performed an experiment to design adaptive filters based on the difference between the reference frame and the current frame. The procedure can be summarized as follows:[†]

1. Initial disparity estimation to obtain disparity field (dv_x, dv_y) of the current frame.
2. Calculate MMSE filter ψ such that:

$$\min_{\psi} \sum_{x,y} \left(S_{x,y} - \psi * R_{x+dv_x, y+dv_y} \right)^2, \tag{1}$$

[†]Note that the formulation of filter estimation is similar to Refs. 5–7. For demonstration purposes, in this experiment we constrained the design to one adaptive filter per frame, as the blur effect appears to be global.

where S is the current frame to be encoded, R is the reference frame, the subscript denotes (x, y) the pixel position within a frame, $(x + dv_x, y + dv_y)$ is its corresponding disparity-displaced pixel in the reference frame, and $*$ denotes the two-dimensional convolution.

3. The estimated adaptive filter is applied to the reference frame to generate a better match. The final disparity compensation is performed with the filtered reference.

In this experiment, ψ is a 5×5 filter, symmetrical with respect to x- and y-axis. Fig. 4 provides the frequency responses of the calculated MMSE filters when we perform disparity compensation from View 2 to View 3 and from View 5 to View 6. For the former case, in which the current frame is blurred, it can be seen that the frame-wise filters have a low-pass characteristic. On the other hand, when the reference frame is a blurred version of the current frame (View 5 to View 6), the filters emphasize higher frequency ranges so that the reference can be sharpened to create a better match. Another feature worth noting is that, for different time stamps (frame 10 and 20 as in Fig. 4), the filters for the same view have quite similar frequency responses. This result suggests that the blur effect was likely introduced by camera mismatches.

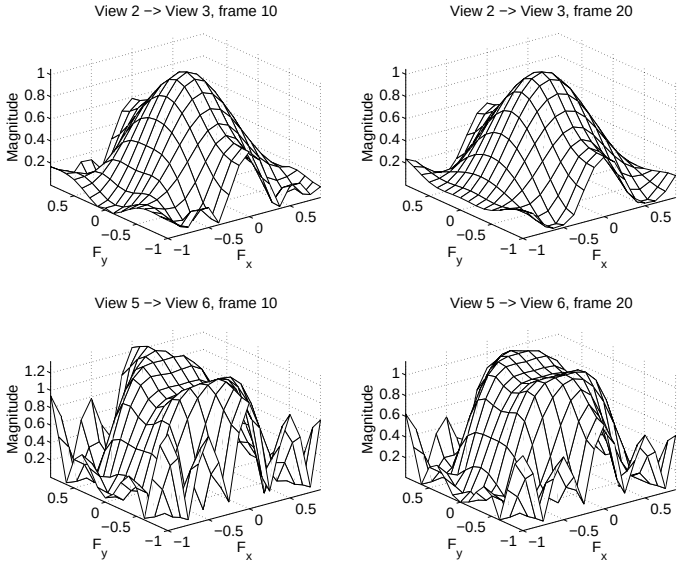


Figure 4. Frequency responses of calculated MMSE filters

We performed simulations with the adaptive filtering approach described in this section. The simulations use H.264 to encode frames in cross-view direction only, i.e., we take a sequence of frames captured at the same time from different cameras and feed this to the encoder as if it were a temporal sequence. Anchor frames at time stamps 0, 10, 20, 30, 40 are encoded for View 3 and View 6. The RD results are provided in Fig. 5.

Note that in this experiment, to focus specifically on the effect of the filtering, only the filtered reference frame will remain in the reference buffer; the original reference frame is discarded. Higher coding efficiency can be achieved if both filtered and original reference frames are stored.[9] From Fig. 5, at about 80 to 100Kb/frame, the frame-wise filtering provides $0.7 \sim 0.8dB$ gain for frames in View 3 and around $0.3dB$ for frames in View 6. Our interpretation to this non-symmetrical performance is as follows: A blurred frame is a frame with some mid-, high-frequency components being lost or suppressed from the original version. When cross-view prediction is performed from a normal view to a blurred view (V2 to V3 in Fig. 5), the effect of this "loss" can be approximated by applying lowpass filters to the reference frame. On the other hand, to encode a normal view from a blurred view, adaptive filters try to compensate for the difference by enhancing mid-, high-frequency ranges. However, with the fixed-shaped 5×5 symmetrical filter presented in this section, it is not sufficient to fully "recover the loss".

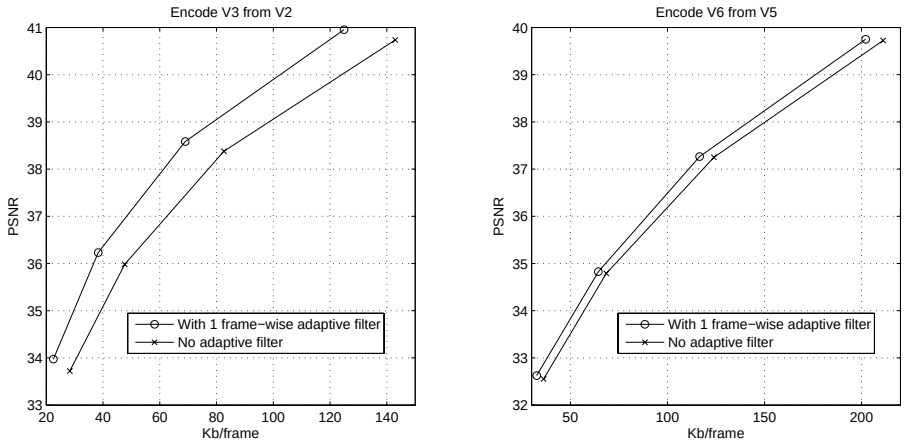


Figure 5. Encoding results of the frame-wise filtering: Race1 sequence

3. THE PROPOSED ADAPTIVE FILTERING APPROACH FOR CROSS-VIEW DISPARITY COMPENSATION

We have illustrated in Fig. 2 a camera arrangement that could cause local focus mismatches for cross-view prediction. Another scenario that can result in a similar local effect is when cameras were not perfectly calibrated with the same in-focus scene-depth. To extend the adaptive filtering approach to these situations, locally adaptive compensation has to be enabled by considering scene depth.

For cross-view prediction, we first perform disparity estimation to obtain the disparity field from the reference frame to the current frame. The disparity vectors are then considered to provide estimates scene depth. Blocks with similar disparity vectors are grouped into classes. Each class represents a scene-depth level and will be associated with one adaptive filter to be designed in the next step. We call this process "filter association". For each class (scene-depth level), we then select a filter that is optimized to minimize the residual error for all blocks in the class. This depth-dependent adaptive filter design is chosen to minimize:

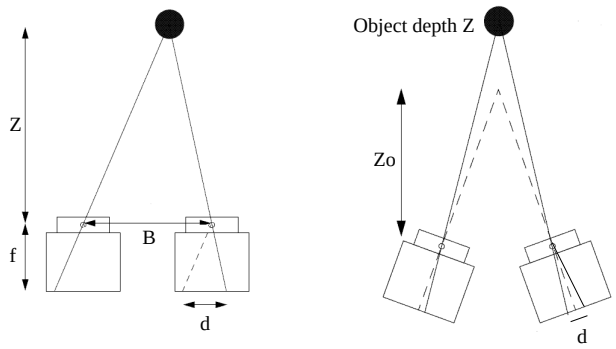
$$\min_{\psi_k} \sum_{k} \sum_{(x,y)\in D_k} \left(S_{x,y} - \psi_k * R_{x+mv_x,y+mv_y}\right)^2 \tag{2}$$

As compared to Equation (1), D_k and ψ_k here are, respectively, the set of pixels and the filter corresponding to depth-level class k. Filters that satisfy (2) are Wiener filters that minimize the mean squared error (MMSE filters). Once these filters have been computed for each depth-level class, they are all applied to the reference frame to provide better matches. Then the disparity estimation and compensation is performed using both original and filtered frames as references, and each block is allowed to select a reference that provides the lowest rate-distortion (RD) cost. In the following subsections, we describe each step in detail.

3.1. Filter association

The first step is to identify image portions with different depths, which therefore are likely to have different types of mismatch. In situations where multiple cameras are employed, disparity information has been widely used as an estimation of scene depth.[10] The underlying principle is illustrated by Fig. 6, in which both parallel and convergent camera arrangements are depicted.

For both cases in Fig. 6, f is the focal length of the camera lens and B is the spacing between the cameras. On the right-hand side, camera convergence occurs at depth Z_0. These three parameters will remain constant once the multiple camera system has been set up. Thus, it can be derived from Equation (3) that the depth Z and disparity d are reciprocals: objects closer to the cameras (smaller depth) will have a larger disparity; while objects far away (larger depth) will possess a smaller disparity. Based on these relationships, numerous approaches have



$$Z = \frac{fB}{d} \text{ if parallel cameras, } Z \approx \frac{fB}{d + \frac{fB}{Z_0}} \text{ if convergent} \tag{3}$$

Figure 6. Relationship between depth and disparity

been proposed to estimate scene depth by exploiting the disparity.[11–13] While in some existing approaches the goal is to find an accurate/smooth disparity map at the pixel-level, here we simply aim at separating objects with different scene depths by modeling their disparities. This is similar to video object segmentation methods in which the motion field is used to identify moving objects.[14] To reduce complexity, compressed domain fast segmentation methods have been proposed by taking the block-wise motion vectors, obtained with video coding tools, as input features to classify image blocks.[15–17] Similarly, we consider procedures to classify blocks into depth levels based on their corresponding disparity vectors. For multi-view systems with cameras arranged on the same horizontal line, such as in Fig. 6, classification can be achieved by considering only the x component of the disparity vectors. For a 2D camera arrangement as can be found in a camera array, the classification could be extended by taking both x and y components as input features.

We propose to use classification algorithms based on the Gaussian Mixture Model (GMM) to separate blocks into classes (depth levels). As presented in Refs. 16–18, we adopted EM clustering based on the GMM[19] to classify the disparity vectors and their corresponding blocks. In this paper, an unsupervised EM classification tool developed by Bouman[20] is employed. Based on the distribution of the disparity vectors, it first constructs a GMM for these vectors. To automatically estimate the number of Gaussian components in the mixture (thus making the approach unsupervised), the software tool performs an order estimation based on minimum description length (MDL) criteria. We refer to Refs. 20–22 for details about such techniques. Parameters (mean, covariance matrix, prior) of each Gaussian are estimated using an iterative EM algorithm. Each Gaussian component is used to construct a Gaussian probability density function (pdf) that models one class for classification. Likelihood functions can be calculated based on these Gaussian pdfs. Disparity vectors are classified into different groups by comparing their corresponding likelihood value in each Gaussian component. Blocks are classified accordingly based on the class label of their corresponding disparity vectors. Refining processes can also be considered in the classification based on GMM, such as eliminating a class to which a very small number of blocks has been assigned. In the classification result, each class represents a depth level within the current frame, and blocks classified into a certain level will be associated with one adaptive filter. An example of such filter association results using the proposed method is provided in Figs. 7 and 8.

In Fig. 7, a GMM is constructed with a number of components estimated to be 3. In Fig. 8, the corresponding blocks within each class are shown. It can be observed that after EM classification, depth class 1 corresponds to the far background; class 2 captures two dancing couples and some audience in the mid-range, along with their reflection on the floor; and class 3 includes the couple in the front. Note that intra-coded blocks are not involved in the filter association process. In this example, the classification tool successfully separates objects with different depths in the current frame.

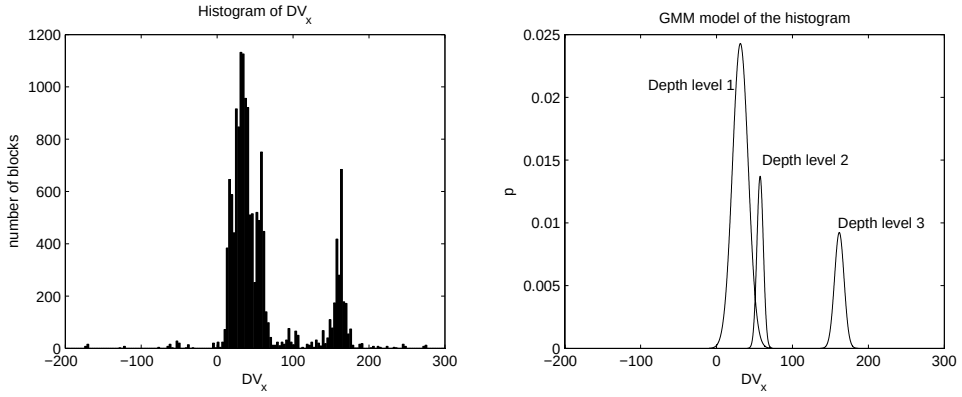


Figure 7. Disparity vectors from view 6 to view 7 at the 1st frame in Ballroom: Histogram and GMM

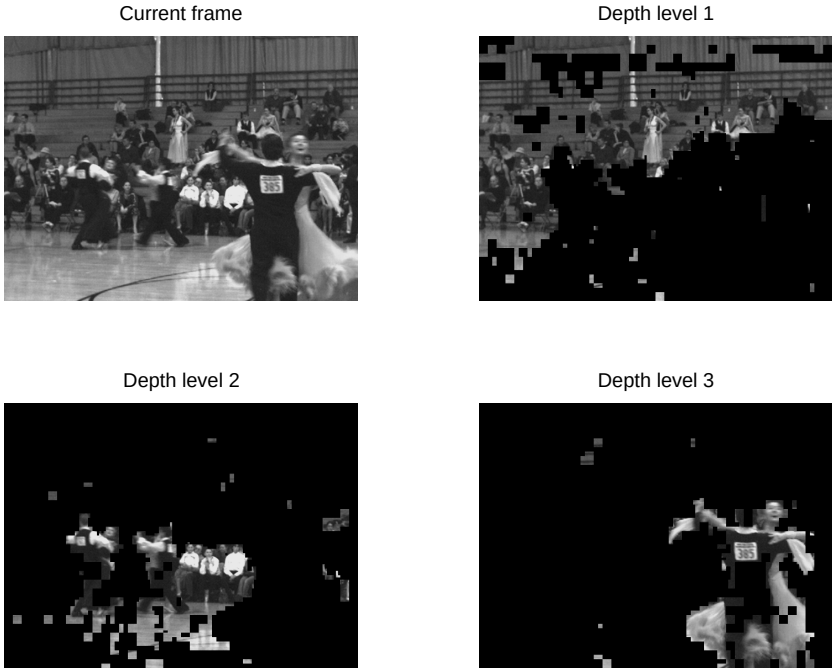


Figure 8. The corresponding EM classification result of Fig. 7

3.2. Depth-class-adaptive filter selection

We now discuss how to select a filter for all blocks belonging to a given depth level class. We rewrite Equation (2) so that we are looking at the ψ for *each* class k, that is, $\forall(x, y) \in D_k$:

$$\min_{\psi} \sum_{(x,y)} \left(S_{x,y} - \sum_{j=-n}^{n} \sum_{i=-m}^{m} \psi_{ij} R_{x+mv_x+i, y+mv_y+j} \right)^2 \tag{4}$$

Here we replace the convolution notation by explicitly expressing the filter operations. The size and shape of 2D filters can be specified by changing m and n. In AIF, even-length (6×6) filters are proposed in order to interpolate subpixels. Here we apply adaptive filters directly to the reference frame to generate better matches. Odd-length filters (e.g., 5×5) centered at the pixel to be filtered are employed in this paper. Constraints such

as symmetry can be imposed to reduce the number of unknowns in the adaptive filter estimation. Filters with more unknowns can be more efficient to compensate for residue energy. However, this comes at the expense of having to estimate and transmit more filter coefficients. (For example, a circular symmetric 3×3 filter contains only 3 coefficients, while a full 3×3 matrix has 9 coefficients) In this paper, we use as an *example* 5×5 filters ($m = n = 2$), with the coefficients selected as:

$$\psi = \begin{pmatrix} a & b & c & b & a \\ d & e & f & e & d \\ g & h & j & h & g \\ d & e & f & e & d \\ a & b & c & b & a \end{pmatrix} \tag{5}$$

This can be viewed as a compromise between a full matrix and a circular symmetric one. Note that the circular symmetric filter is a degenerated case of Equation (5) where we have chosen $h = f$, $g = c$, and so on. For each group, a filter in the above form will be obtained as a solution to Equation (4).

3.3. Disparity compensation with local adaptive filtering

The obtained adaptive filters will be applied to the reference frame to generate better matches for cross-view prediction. In the reference picture list, the original unfiltered reference as well as multiple filtered references are stored. During the encoding process, each block in the current frame can select a block in any filtered or original reference frame, i.e., the one that provides the lowest matching cost, independently of whether the block was classified in a different class during the filter association process.[‡]

To correctly decode the video sequence, the filter coefficients and the filter chosen for each block have to be transmitted to the decoder. Using the reference picture list as described above, the filter selection can easily be handled by signaling the reference frame index.[8] To encode the filter coefficients, in this paper, we directly extend the method proposed in Refs. 23,24, in which the filter coefficients are quantized and differentially coded with respect to the filter in the previous frame.

4. SIMULATION RESULTS

We performed simulations using H.264 to encode multi-view video across views. The EM classification tool[20] based on GMM is combined with our encoder to classify the disparity vectors. We encode the anchor frames at different time stamps to evaluate different cross-view prediction schemes.

The proposed adaptive reference filtering approach (ARF) is compared with AIF and current H.264. Motion compensation with multiple references is a coding option in H.264[8] that also aims to improve coding efficiency by providing better matches. Our proposed ARF utilizes multiple filtered versions from a *single* reference frame. If the EM classification generates K classes, each with a corresponding filter, there will be $N = K + 1$ references in the reference list, including the original unfiltered one. In our simulations, the maximum K allowed is 4. Thus, we also compare our method to H.264 with the number of reference frames set to 5. The rate-distortion results are provided in Figure 9. Simulations were performed with QP = 24, 28, 32, and 36 to obtain four rate points.

It can be seen that, for the multi-view sequences we tested, the proposed ARF coding scheme provides higher coding efficiency than H.264 coding method. Moreover, the performance of different methods varies significantly among different test sequences:

In the Ballroom sequence, almost no cross-view mismatch can be observed. Furthermore, frames from different views have been rectified (properly registered) by applying homography matrices.[1] In this situation, the three enhanced predictive coding schemes: multiple reference frame, AIF, and our ARF all provide very similar coding efficiency, i.e. with about a $0.3 \sim 0.4dB$ gain over H.264 using 1 reference.

[‡]Note that after this stage, the newly estimated disparity vectors could be regarded as the new input for EM clustering for "filter association" D_k, and filters ψ_k could be estimated again based on blocks in different classes. Thus, the estimation of D_k and ψ_k can be carried iteratively until a stopping criterion is met. The complexity involved in such process will be fairly high. In this paper we limited ourselves to a single pass algorithm as described in Section 3.

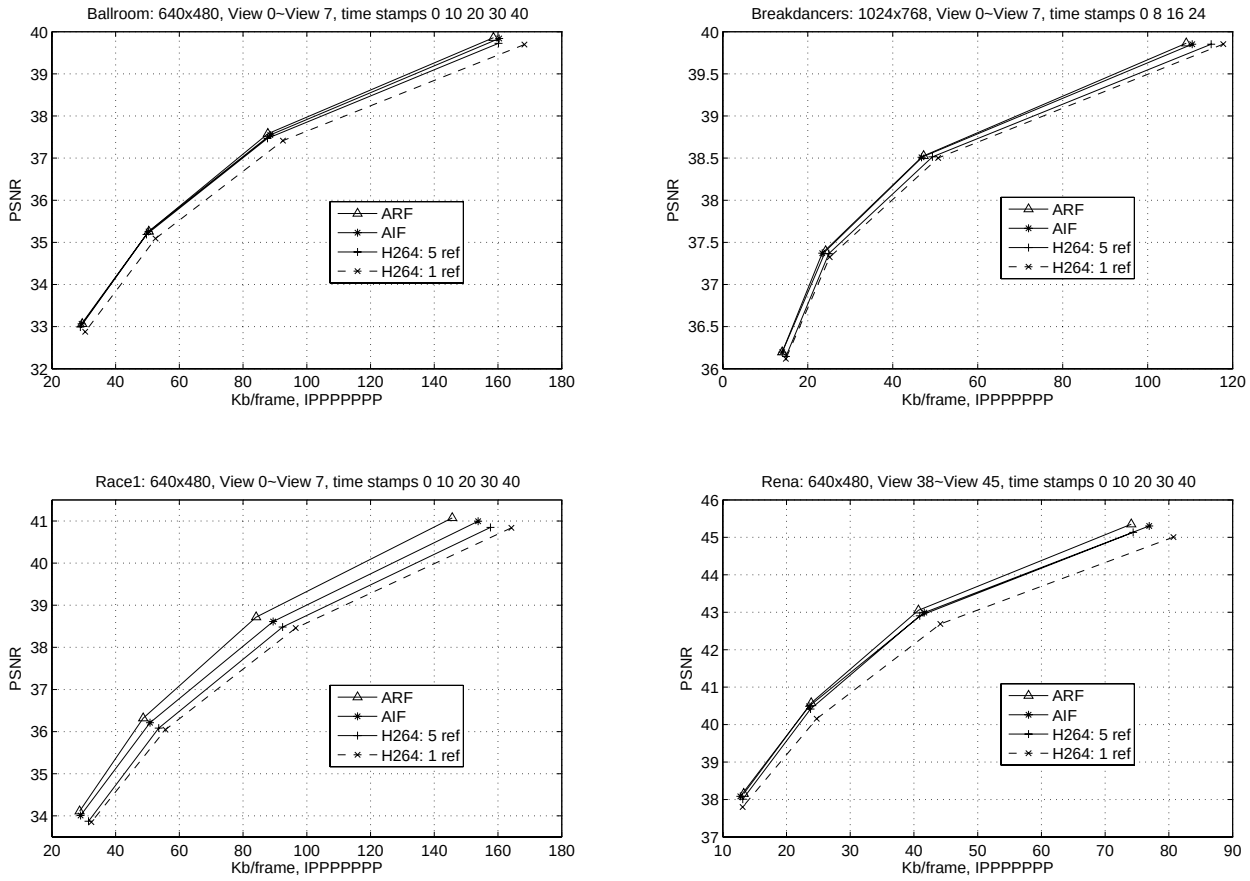


Figure 9. Rate-Distortion comparison of different coding schemes

Cameras for the Breakdancers sequence are arranged on an arc. View 4 in this sequence is somewhat blurred while all the other views have very a similar subjective visual quality. From the simulation results, we see that adaptive filtering can be used to provide better reference for predictive coding. Both the proposed method and AIF achieve higher coding efficiency than the multiple references method in H.264. Our ARF method provides a marginal gain over AIF with filter association based on scene depth. However, it is worth noting that the achievable gain with these enhanced predictive codings is relatively limited as compared to that achievable with other multi-view sequences in Fig. 9. It is because the frames in the Breakdancers sequence have large homogeneous areas for which intra coding was selected. Fig. 10 in the Appendix section provides an example of the disparity-based classification and intra-coded areas during the initial disparity estimation. Due to the relative high amount of intra-coded blocks, our proposed method provides a $0.1dB$ gain at 100Kb/frame and a 0.2dB at 200Kb/frame, over H.264.

Race1 is the sequence that suffers from severe cross-view mismatches. Besides the two pairs of blurriness mismatches described in Section 2, its View 4 is particularly blurred. As a result, the benefit of adaptive filtering is more prominent: AIF provides a $0.3dB$ gain over multiple reference H.264, and the proposed ARF achieves an additional $0.2\sim0.3dB$ gain over AIF ($0.7\sim0.8dB$ over H.264 with one reference). The cross-view mismatches are more efficiently compensated with our depth dependent adaptive filtering. Similarly for the Rena sequence, in the presence of some degree of cross-view discrepancy, our ARF method again provides a $0.15dB$ gain over AIF ($0.4dB$ gain with respect to H.264 with one reference). In this sequence, due to a much closer spacing of the cameras as compared to other test sequences (5cm versus 20cm), the standard multiple reference method achieves coding efficiency similar to that of AIF.

Based on the simulation results, it can be concluded that our proposed method is especially helpful for disparity compensation with cross-view mismatches. It effectively estimates filters to compensate for the possible discrepancies associate with scene depth. For sequences with stronger cross-view mismatches, it provides greater coding gains over single reference AIF and the multiple reference method without adaptive filtering.

As for encoding complexity, we measured the encoding time and disparity estimation (DE) time with a profiling tool in JM 10.2 software.[25] For fair comparison, the full search disparity estimation is applied for all coding schemes. There are two factors in our proposed ARF approach that will increase the complexity. One is the initial disparity estimation and EM classification for filter association. The other is the DE loop over multiple filtered references. Instead, filters in AIF are imposed on the subpixel positions at a single reference frame. As a result, the DE time in our method is similar to that in H.264 with 5 references, and is about 2.5 times as long as in AIF. However, unlike multiple reference method, the references in our system are simply different filtered versions of the same frame. Taking this into account, significant complexity reduction could be achieved by reusing disparity information: As we proceed from the unfiltered reference to the filtered ones, a much refined search range based on previously computed disparity could be applied.

At the decoder side, our method requires only one reference frame to be decoded. Filtered references will be generated upon receiving the filter coefficients. This provides an advantage over H.264 multiple reference method. For cross-view prediction, multiple references will consist of frames from different views. To decode frames in a particular view, several neighboring views have to be decoded in advance. Such *cross-view dependency* significantly affects the decoding complexity.[26] For scenarios such as view switching, low delay methods with less cross-view dependency will be preferred.

5. CONCLUSIONS

For multi-view video coding, performing disparity compensation provides additional coding efficiency as compared to simulcast. However, frames from different views can exhibit mismatches caused by heterogeneous cameras and shooting positions, such as focus mismatch among different views. Furthermore, these discrepancies could be localized such that different portions within a frame may suffer from different types of mismatches.

We propose an adaptive filtering approach for cross-view prediction to compensate for such discrepancies. Our approach first performs a disparity estimation to obtain the disparity field. The disparity vectors are exploited as an estimation of scene depth. Blocks with similar disparity vectors are grouped into classes (scene-depth levels) and associated with adaptive filters. EM classification with GMM basis is applied and the number of class is automatically decided with MDL criterion. Based on the filter association result, an adaptive filter is constructed for each class. This depth-dependent filter design is adaptive to the changes between the current frame and the reference frame.

To provide better matches, filtered references are generated by applying these adaptive filters. For the sequences we tested, for cross-view prediction, the proposed method provides higher coding efficiency as compared to the current H.264 with multiple reference frames and other adaptive filtering such as AIF. Larger coding gain is achieved especially for sequences with stronger cross-view mismatches.

APPENDIX A. DISPARITY VECTORS CLASSIFICATION EXAMPLES

In this section we provide some results of the proposed classification method based on disparity vectors.

REFERENCES

1. "Call for proposals on multi-view video coding," *ISO/IEC-JTC1/SC29/WG11 MPEG Document N7327* , Jul. 2005.
2. "MPEG press release," *ISO/IEC-JTC1/SC29/WG11 MPEG Document N8195* , Jul. 2006.
3. "Submissions received in CfP on multi-view video coding," *ISO/IEC-JTC1/SC29/WG11 MPEG Document M12969* , Jan. 2006.
4. M. Budagavi, "Video compression using blur compensation," in *IEEE Proc. of International Conference on Image Processing (ICIP) 2005*, **vol.2**, pp. 882–885, Sep. 2005.

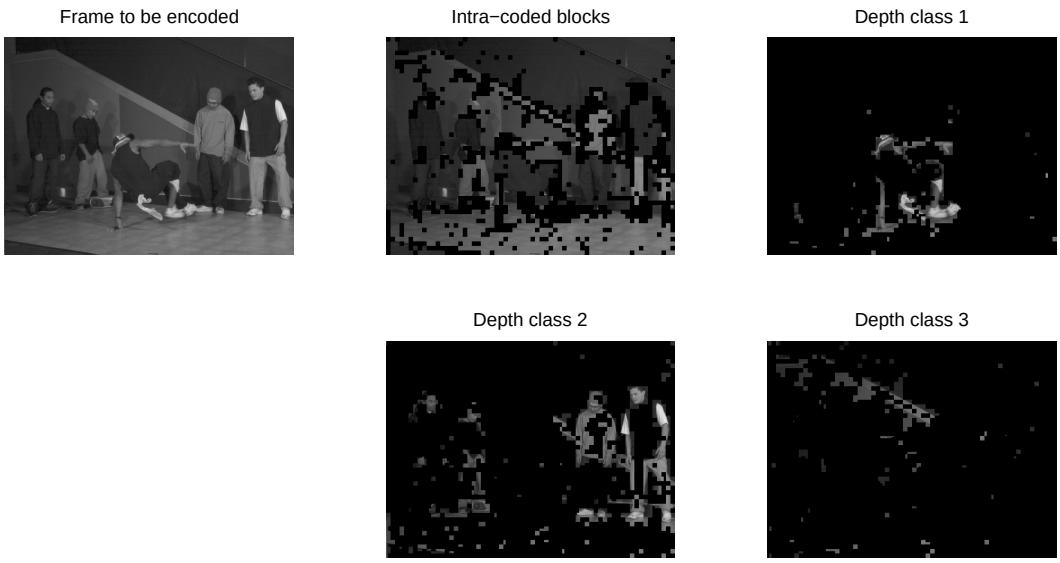


Figure 10. Breakdancer: View 1, frame 0

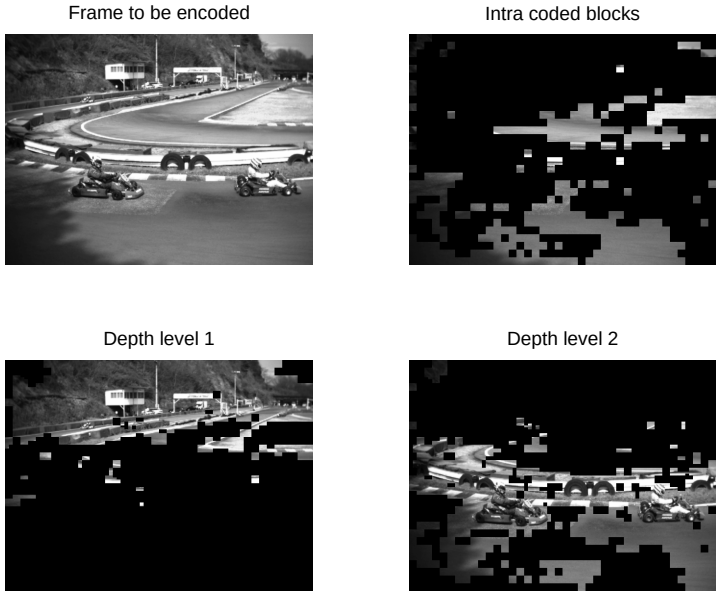


Figure 11. Race1: View 2, frame 30

5. T. Wedi, "Adaptive interpolation filter for motion compensated prediction," in *IEEE Proc. of ICIP 2002*, **vol.2**, pp. 509–512, Sep. 2002.
6. T. Wedi, "Adaptive interpolation filters and high-resolution displacements for video coding," *IEEE Trans. Circuits Systems and Video Technologies* **vol.16, no.4**, pp. 484–491, Apr. 2006.
7. Y. Vatis, B. Edler, D. T. Nguyen, and J. Ostermann, "Motion-and aliasing-compensated prediction using a two-dimensional non-separable adaptive wiener interpolation filter," in *IEEE Proc. of ICIP 2005*, **vol.2**, pp. 894–897, Sep. 2005.
8. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Systems and Video Technologies* **vol.13, no.7**, pp. 560–576, Jul. 2003.

9. Y. Vatis and J. Ostermann, "Locally adaptive non-separable interpolation filter for H.264/AVC," in *IEEE Proc. of ICIP 2006*, Oct. 2006.
10. D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2003.
11. T. Aach and A. Kaup, "Disparity-based segment ation of stereoscopic foreground/background image sequences," *IEEE Trans. Communications* **vol.42, issue.2/3/4, Part.1**, pp. 673–679, Feb-Apr. 1994.
12. E. Francois and B. Chupeau, "Depth-based segmentation," *IEEE Trans. Circuits and Systems for Video Technology* **vol.7**, pp. 237–239, Jun. 1997.
13. E. Izquierdo, "Disparity/segmentation analysis: Matching with an adaptive window and depth-driven segmentation," *IEEE Trans. Circuits and Systems for Video Technology* **vol.9, no.4**, pp. 589–607, Jun. 1999.
14. D. Zhang and G. Lu, "Segmentation of moving objects in image sequence: A review," *Springer Journal of Circuits, Systems and Signal Processing* **vol.20, no.2**, pp. 143–183, Mar. 2001.
15. M. L. Jamrozik and M. H. Hayes, "A compressed domain video object segmentation system," in *IEEE Proc. of ICIP 2002*, **vol.1**, pp. 113–116, Sep. 2002.
16. Z. Wang, G. Liu, and L. Liu, "A fast and accurate video object detection and segmentation method in the compressed domain," in *IEEE Proc. of International Conference on Neural Networks and Signal Processing*, **vol.2**, pp. 1209–1212, Dec. 2003.
17. R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: A compressed domain approach," *IEEE Trans. Circuits Systems and Video Technologies* **vol.14, no.4**, pp. 462–474, Apr. 2004.
18. K. Y. Wong and M. E. Spetsakis, "Motion segmentation by EM clustering of good features," in *IEEE Proc. of Computer Vision and Pattern Recognition Workshop 2004*, pp. 166–173, Jun. 2004.
19. E. Redner and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review* **vol.26, no.2**, Apr. 1984.
20. C. A. Bouman, "Cluster: An unsupervised algorithm for modeling gaussian mixtures," *http://cobweb.ecn.purdue.edu/bouman/software/cluster/* , this version was released in Jul. 2005.
21. A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society* **Series B, vol.39, no.1**, pp. 1–38, 1977.
22. J. Rissanen, "A universal prior for integers and estimation by Minimum Description Length," *Institute of Mathematical Statistics Journal: Annals of Statistics* **vol.11, no.2**, pp. 417–431, 1983.
23. Y. Vatis, B. Edler, I. Wassermann, D. T. Nguyen, and J. Ostermann, "Coding of coefficients of two-dimensional non-separable adaptive Wiener interpolation filter," in *SPIE Proc. of Visual Communication and Image Processing 2005*, **vol.5960**, pp. 623–631, Jul. 2005.
24. Y. Vatis, "Software implementation of adaptive interpolation filter," *http://iphome.hhi.de/suehring/tml/download/KTA/* , this software was released in Nov. 2005.
25. "Software implementation of H.264: JM Version 10.2," *The Image Communication Group at Heinrich Hertz Institute Germany, http://iphome.hhi.de/suehring/tml/index.htm* , this version was released in Jul. 2006.
26. P. Lai, J. H. Kim, A. Ortega, Y. Su, P. Purvin, and C. Gomila, "New rate-distortion metrics for MVC considering cross-view dependency," *ISO/IEC-JTC1/SC29/WG11 MPEG Document M13318* , Apr. 2006.