New Coding Tools for Illumination and Focus Mismatch Compensation in Multiview Video Coding

Jae Hoon Kim, *Student Member, IEEE*, PoLin Lai, *Student Member, IEEE*, Joaquin Lopez, Antonio Ortega, *Fellow, IEEE*, Yeping Su, Peng Yin, and Cristina Gomila

(Invited Paper)

***Abstract*—We propose new tools for multiview video coding (MVC) that aim to compensate for mismatches between video frames corresponding to different views. Such mismatches could be caused by different shooting positions of the cameras and/or heterogeneous camera settings. In particular, we consider illumination and focus mismatches across views, i.e., such that different portions of a video frame can undergo different illumination and blurriness/sharpness changes with respect to the corresponding areas in frames from the other views. Models for illumination and focus mismatches are proposed and new coding tools are developed from the models. We propose a block-based illumination compensation (IC) technique and a depth-dependent adaptive reference filtering (ARF) approach for cross-view prediction in multiview video coding. In IC, disparity field and illumination changes are jointly computed as part of the disparity estimation search. IC can be adaptively applied by taking into account the rate-distortion characteristics of each block. For ARF, the disparity fields are used to estimate scene depth, such that video frames are first divided into regions with different scene-depth levels. A 2-D filter is then selected for each scene-depth level. These filters are chosen to minimize residual energy, with the goal of compensating for focus mismatches. The resulting filters are applied to the reference frames to generate better matches for cross-view prediction. Furthermore, we propose a coding system that combines IC and ARF. Adjustments are made so as to maximize the gains achieved by using both coding tools, while reducing the complexity of the final integrated system. We analyze the complexity of all proposed methods and present simulation results of IC, ARF and combined system for different multiview sequences based on the H.264/AVC reference codec. When applying the proposed tool to cross-view coding we observe gains of up 1.3 dB as compared to directly using an H.264/AVC codec to perform predictive coding across views.**

***Index Terms*—Adaptive filtering, cross-view prediction, H.264/AVC, illumination compensation (IC), multiview video coding (MVC).**

Manuscript received January 10, 2007; revised May 28, 2007. This paper was recommended by Guest Editor J. Ostermann.

J. H. Kim, P. Lai, and A. Ortega are with the Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: jaekim@sipi.usc.edu).

J. Lopez is with the Mavrix Technology, Newport Beach, CA 92660 USA.

Y. Su is with the Sharp Laboratories of America, Camas, WA 98607 USA.

P. Yin and C. Gomila are with the Thomson Corporate Research, Princeton, NJ 08540 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2007.909976

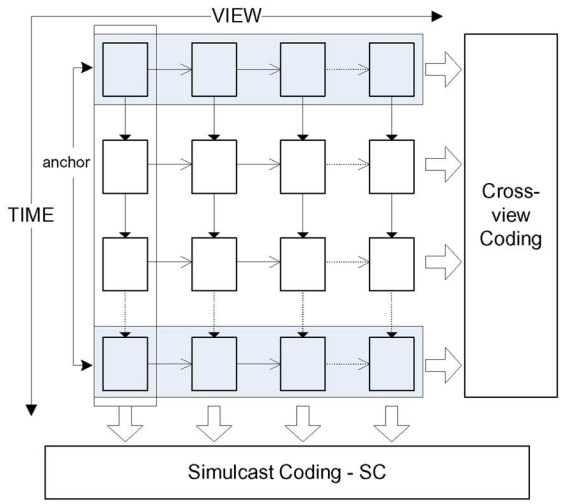


Fig. 1. Multiview video structure : simulcast and cross-view coding.

I. Introduction

MULTIVIEW video systems are used for simultaneously capturing scenes or objects with multiple cameras from different viewpoints. From the video sequences corresponding to each view it is possible to extract 3-D information, e.g., the scene depth can be estimated using the correspondence of objects from different views. Multiview video coding systems are being proposed for new multimedia services, e.g., 3-D cinema/TV, free viewpoint video and immersive virtual reality.

The amount of data generated by these systems increases in proportion to the number of views, and can be very large as compared to monoscopic video. Widespread use of multiview video thus requires the design of efficient compression techniques. Multiview video coding (MVC) has recently become an active research area [1], [2], focused on compression for efficient storage and transmission of multiview video data. A straightforward compression approach would be to employ standard video coding techniques and apply them to each of the views independently. This type of "simulcast" coding would allow temporal redundancy to be exploited using standard block-based motion compensation techniques. Since adjacent cameras in a multiview system capture overlapping areas in a scene, additional cross-view redundancy could also be exploited. A block matching procedure can be employed to find block correspondence from view to view, leading to a disparity estimation and compensation process, analogous to

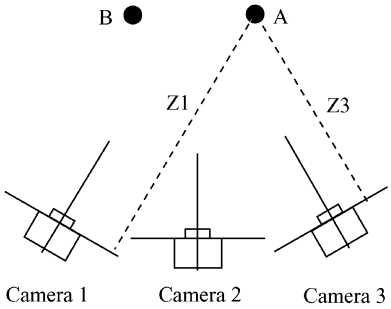


Fig. 2. Camera arrangement that causes local focus mismatches.

motion estimation and compensation as used in monoscopic video. Exploiting both temporal and cross-view redundancies achieves higher coding efficiency as compared to simulcast [3].

Fig. 1 depicts an example of a prediction structure that uses both motion and disparity compensation. To facilitate random access, video frames will be periodically encoded using only cross-view prediction, i.e., no temporal prediction will be used. We denote such frames "anchor frames," as illustrated in Fig. 1. Similar to I-frames in monoscopic video, these anchor frames serve as random access points for multiview video. While most of the experimental results presented in this paper evaluate the coding efficiency achieved in compressing anchor frames, our proposed techniques can also be applied to frames where both temporal and cross-view prediction are used (e.g., the non-anchor frames in Fig. 1) and to more general prediction structures.

In temporal predictive coding of monoscopic video, block matching techniques tend to be efficient in compensating relatively simple translational motion. Likewise, in cross-view predictive coding, block matching will be most efficient when objects appear in multiple views with only slight changes from view to view, e.g., with only a shift dependent on their depth within the scene. However, in the multiview case, simple displacements do not always provide a sufficiently good prediction, due to a series of other sources of mismatch.

Firstly, the multiview video capturing system might not be perfectly calibrated. Camera parameters may be inconsistent, so that the exposure and/or focus may be different for different views. These heterogeneous cameras can cause global (framewise) mismatches, e.g., frames in one view may appear brighter as compared to frames in another view; or localized mismatches, as objects may not always be in sharp focus across different views.

Secondly, even if all cameras are perfectly calibrated, camera positions and orientations lead to differences in how certain objects appear in different views. As an example, consider the camera arrangement in Fig. 2, where object A appears at a greater depth (z_1) in view 1 than in view 3 (z_3). Assume all cameras are set with the same focus at depth z_1, then object A may appear in focus in view 1 while it will likely be out of focus (blurred) in view 3. On the other hand, object B will appear sharper in view 3 as compared to view 1. This illustrates a scene-depth dependent focus mismatch across views. Furthermore, an object also possesses different projection directions, therefore different illumination effects will manifest themselves in views 1 and 3. Such illumination changes can be even more localized, in the sense that different parts within the same object can reflect light differently due to shape and texture. Under the camera setting shown in this example, different portions of a video frame can undergo different illumination and blurriness/sharpness changes with respect to the corresponding areas in frames from other views (localized illumination and focus mismatches). These factors lead to discrepancies among video sequences in different views.

The efficiency of cross-view prediction can deteriorate in the presence of mismatches such as those described above. For temporal prediction in monoscopic video coding, illumination and focus mismatches could also present. However, in general, they are far less significant as compared to what can be perceived in cross-view direction in MVC. They are typically caused by the displacement of certain objects, rather than by the change of shooting perspective of the whole scene as in multiview systems. Furthermore, temporal changes in illumination and focus tend to be relatively small between consecutive frames in monoscopic video, while more significant changes can be observed between corresponding frames in neighboring views.

Systems for efficient illumination and focus mismatch compensation in cross-view prediction should be designed with the following requirements in mind. First, *local compensation* is useful in addressing scene-depth dependent focus mismatch and localized illumination changes. Furthermore, as scenes change over time, cross-view mismatch characteristics will also change, so that the compensation process should be *adaptive*. The best predictive performance will in general be achieved when disparity estimation and mismatch parameter estimation are performed *jointly*, i.e., where the best match is identified as providing lowest residual energy after mismatch compensation. Finally, decisions on whether or not to use mismatch compensation should be based on *rate-distortion* (R-D) criteria in order to optimize overall coding efficiency.

Various approaches have been proposed for monoscopic video coding to address illumination and focus changes in temporal prediction (although, to the best of our knowledge, these two types of mismatches have not been treated jointly). In [4], illumination correction to improve motion estimation is proposed based on a block-wise additive term. Deciding whether illumination correction should be applied to a given block is based on two simple thresholds and does not take into consideration overall R-D cost. In [5], illumination is compensated in two steps. First, illumination mismatch is compensated globally using a decimated image (that contains the DC coefficients of all blocks). Then blockwise compensation is applied. In both steps, multiplicative and additive terms are used. This two step compensation is applied only to frames classified as having large illumination mismatches, which does not occur as frequently in monoscopic temporal prediction, as compared to cross-view prediction in MVC. Note also that local compensation is not fully integrated into the search step and that an efficient coding for mismatch parameters is not provided. In [6], an illumination component and a reflectance component are both compensated using scale factors that are quantized and Huffman coded. This illumination model is useful for contrast adjustment but cannot model severe mismatch in MVC properly. In [7], illumination mismatches are modeled by multiplicative and additive terms. These two parameters are used globally for whole frames. To reduce the impact of local brightness variation, a set of parameters is collected and

a pair is chosen based on the relative frequency of all parameter pairs. Illumination compensation is deactivated for those blocks for which the selected parameters are not efficient. These global approach cannot adapt to some large luminance variations in MVC, which are dependent on relative positions of camera and objects. In [8], both scale and offset parameters are proposed as an illumination model and jointly estimated along with the motion field using optical flow equations (OFE). These global illumination parameters can then be used to produce a pixelwise spatially varying model. The authors suggest that more localized illumination mismatches can be compensated for by applying their technique to "patches" within each frame, and introducing some connectivity constraints at the boundaries between patches. Motion and illumination parameter estimation via OFE is accomplished under the assumption of small and smoothly varying displacements in temporal prediction. In contrast, our proposed methods are designed for block-based cross-view prediction. When illumination compensation is to be performed on relatively small blocks we observe that a single parameter model is sufficient (i.e., using more parameters increases overhead without producing sufficient reductions in residual energy). Moreover, when small blocks are used, pixel-wise adjustments in illumination compensation (as enabled by [8]) become less attractive, since variations within such blocks tend to be relatively modest (thus, reductions in residual energy tend to be small as compared to using a single parameter for the whole block.) Finally, it is worth mentioning that motion vector information tends to exhibit more smoothness than disparity vector information (a smooth disparity would essentially mean that most objects in the scene are at a similar depth). Thus the smoothness assumptions that underpin the approach of [8] may not be a good fit to typical disparity fields, so that overall gains may be lower than for standard temporal prediction. Recently [9], a similar approach to our previous work [10] was proposed. Illumination mismatches are compensated using scale and offset parameters. Mismatch parameters are computed as part of the motion search and are differentially coded and selectively activated. However, this approach mainly targets the illumination compensation in video sequences where luminance changes progressively or due to abrupt changes in lighting, e.g., a flash.

Weighted prediction (WP) methods have also been proposed and adopted in H.264/AVC [11]. For global brightness changes that are uniform across an entire picture, such as fades, a single weighting factor and offset are sufficient to efficiently code all macroblocks that are predicted from the same reference picture. For nonuniform, locally-varying brightness variations, the standard allows more than one reference picture index to be associated with a particular reference picture store by using reference picture reordering or reference picture marking. Each reference can then be associated with a different set of weighting parameters, which enables different macroblocks in the same picture to use different weighting parameters even when predicted from the same reference picture. Although multiple weights provide expanded compensation capabilities, WP in H.264 is limited by the number of reference pictures and our proposed method can be R-D optimized locally since parameter selection and signalling decisions are made blockwise. In [12], local weights are calculated based on neighboring pixel values but such compensation is turned on and off at the slice level rather than at the block level, as in our system.

For focus changes and/or camera panning, Budagavi proposed blur compensation [13], where a fixed set of blurring (low-pass) 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. Second, this method relies on a predefined filter set. Sharpening filters (high-frequency enhancement), for example, which can be useful for focus mismatch in cross-view prediction, were not included. Instead, our work adaptively generates multiple filters based on the mismatches between the reference frame and the current frame. In the final disparity search, each block selects the filter that gives the lowest R-D cost. In [14]–[16], adaptive filtering methods have been proposed in generating subpixel references for motion compensation. Vatis *et al.* [16], calculate adaptive filters for different relative subpixel positions to interpolate subpel reference.[1] In the final motion compensation, the encoder chooses the best match by testing different subpixel positions on the same reference frame. This design approach, which we will refer to as adaptive interpolation filtering (AIF), addresses the aliasing problem and motion estimation error when generating subpel references. Instead, in our work we design filters using scene depth information in order to address the depth-dependent focus mismatches. On each of these filtered reference frames, subpel interpolation can also be applied leading to additional coding gains for disparity compensation.

In the paper, we propose novel coding tools for cross-view disparity compensation in MVC. Firstly, block-based illumination compensation (IC) techniques are presented in Section II. We start by defining an illumination model, and derive a coding scheme that efficiently compensates for illumination changes across views. To compensate local illumination mismatches efficiently, block-wise disparity and illumination parameters are jointly estimated. We integrate this approach with H.264/AVC coding tools. For efficient transmission of IC parameters, we propose differential coding for IC parameters using CABAC in H.264/AVC. Simulation results show that IC leads to up to 0.8 dB gains over standard H.264/AVC in cross-view prediction. Secondly, in Section III, a scene-depth dependent adaptive reference filtering method (ARF) is proposed. Extending our recent work [17], in this paper, we model cross-view prediction with focus mismatch using point spread functions and provide a derivation of how the proposed approach is designed. The main contribution is that we adaptively design multiple filters to compensate for depth-dependent focus mismatches across views. To provide better coding efficiency, we generate multiple filtered reference frames and allow each block to be predicted from the filtered reference providing lowest R-D cost. Simulation results show that when encoding across views with severe blur mismatches, ARF provides up to 0.8 dB gain over cross-view coding using standard H.264/AVC tools with a single reference. Most importantly, in Section IV, we extend the above work by proposing a new coding scheme that compensates for *both* focus and illumination mis-

[1]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 adaptive filter.

matches in cross-view prediction. It combines ARF and IC such that the focus changes are first treated with filtered references, and the remaining mismatches are compensated by IC. We introduce a new filter calculation based on covariance information, which generates ARF with higher AC compensation capability and is more efficient when integrated with IC techniques. The initial disparity search for ARF is replaced by a mean-removed search to maximize the joint benefits from ARF and IC (this also leads to reduced complexity with practically no impact on coding efficiency.) The combined coding system provides up to 1.3-dB gain over cross-view coding using H.264/AVC with a single reference. The complexity analyses of IC, ARF and combined system are given in Section V. Finally conclusions are drawn in Section VI.

II. Illumination Compensation (IC)

Blockwise disparity search aims to find the block in the reference frame that best matches a block in the current frame, leading to minimum residual error after prediction. Under severe illumination mismatch conditions, coding efficiency will suffer because: 1) residual energy for the best match candidate will generally be higher and 2) "true" disparity is less likely to be found, leading to a more irregular disparity field and likely increases to the rate needed for disparity field encoding.

As described previously, illumination mismatches can be local in nature. Thus, we adopt a local IC model to compensate both global and local luminance variation in a frame. The IC parameters are estimated as part of the disparity vector search and these parameters are differentially encoded for transmission to the decoder, in order to exploit the spatial correlation in illumination mismatch. Finally, a decision is made to activate IC on block per block basis using a rate distortion criterion.

A. Multiview One-Step Affine Illumination Compensation (MOSAIC)

When considering pixels corresponding to a given object but captured by different cameras, observed illumination mismatches need not be the same for all pixels, and will depend in general on the continuous plenoptic and radiance functions [18]. However, since our goal is to transmit explicit illumination mismatch information to the decoder, we adopt blockwise IC models, with the optimal block size decided based on R-D cost. As an initial step we evaluate a simple block-wise affine model, with an *additive offset* term C and a *multiplicative scale* factor, S, leading to a mismatch model $\Psi = \{S, C\}$ as proposed in [7].

The ith candidate reference block B_R^i for matching the current block can be decomposed into the sum of its mean μ_R^i and a zero mean signal, ω_R^i : $B_R^i(x,y) = \mu_R^i + \omega_R^i(x,y)$, where (x,y) is the pixel location within the block. Then the illumination compensated reference block signal $\tilde{B}_R^i(x,y)$ with IC model Ψ^i is

$$\tilde{B}_R^i(x,y) = \left[\mu_R^i + C^i\right] + S^i \cdot \omega_R^i(x,y). \tag{1}$$

This formulation allows us to separate the effect of each parameter, so that dc and ac mismatches are compensated, respectively. Furthermore, by applying a multiplicative compensation to the mean removed signal in (1) we avoid the propagation of quantization error from scale to offset [19].

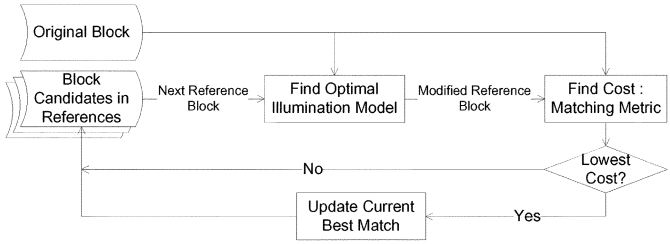


Fig. 3. Modified search loop for the current block.

As shown in Fig. 3, for the given current block, we look for the best matching block within the search range in the reference frame using a modified matching metric that incorporates an IC model between reference and current block. This new metric, sum of absolute differences after compensation (SADAC), essentially computes the SAD between the current and the reference block to which IC has been applied. Thus, for each candidate block, optimal IC parameters have to be computed. While SADAC is used in search with IC similarly to H.264/AVC, a quadratic metric, namely, sum of squared differences after compensation (SSDAC) is used to find IC parameters. For the current block B_C and illumination compensated ith reference block $\tilde{B}_R^i$, this is defined as

$$\mathrm{SSDAC}^i \equiv \Sigma_{\forall(x,y)} |B_C(x,y) - \tilde{B}_R^i(x,y)|^2. \tag{2}$$

Replacing $\tilde{B}_R^i$ using (1) and separating the mean from B_C, we have

$$\mathrm{SSDAC}^i = \Sigma_{\forall(x,y)} \left|\left|\left[\mu_C - \mu_R^i - C^i\right] + \left[\omega_C(x,y) - S^i \cdot \omega_R^i(x,y)\right]\right|\right|^2. \tag{3}$$

Then the optimal IC parameter $\Psi^i = \arg\min_{\{S^i, C^i\}}\{\mathrm{SSDAC}^i\}$ can be obtained by setting to zero the gradient of (3)

$$S^i = \frac{{\sigma_{CR}^i}^2}{{\sigma_{RR}^i}^2} \tag{4}$$

$$C^i = \mu_C - \mu_R^i \tag{5}$$

where

$${\sigma_{AB}}^2 = \frac{1}{N}\Sigma_{\forall(x,y)}[B_A(x,y) - \mu_A][B_B(x,y) - \mu_B] \tag{6}$$

with $A, B \in \{C, R\}$ and N is the number of pixels in the block.

This solution shows that the additive parameter directly removes the offset mismatch and the multiplicative parameter compensates zero-mean variations according to block statistics. If the mean removed current and reference block are not cross-correlated, this scale factor will be small and thus only additive offset compensation will affect the reference block.

Among all candidates within search range, the reference block $\tilde{B}_R$ minimizing SADAC with IC parameters is selected as the best match and the minimum SSDAC is given as follows:

$$\widetilde{\mathrm{SSDAC}} = N \cdot \left(\sigma_{CC}^2 - \frac{\sigma_{CR}^4}{\sigma_{RR}^2}\right) = N \cdot \sigma_{CC}^2 \cdot (1 - \rho^2) \tag{7}$$

TABLE I
UNARY BINARIZATION AND ASSIGNED PROBABILITY FOR INDEX OF QUANTIZED DIFFERENTIAL OFFSET

Absolute value (*val*)	Bin 1	Bin 2	Bin 3	Bin 4 ...
0	0			
1	1	0		
2	1	1	0	
3	1	1	1	0
...	...	...	...	...
Assigned probability.	P1	P2	P3	P4

where ρ is the correlation coefficient between B_R and B_C. As can be seen from (7), the proposed technique finds the best reference block in the sense of maximum correlation with the current block so that the patterns of the two blocks are well-matched, and adjusts parameters to minimize the residual energy.

B. Illumination Mismatch Parameter Coding

Using both scale and offset parameters leads to more flexibility in compensating for illumination mismatches but may not be efficient for coding, given the overhead required to represent both IC parameters. In our observation the scale parameter is also sensitive to quantization noise because it is multiplicative, so that even small quantization errors can lead to fairly large differences in the compensated reference block. Taking this into account, as well as the complexity involved in calculating this parameter within the disparity search step, in the rest of the paper we use only the offset parameter for IC.

To encode the offset parameter we exploit the correlations between the illumination mismatch in neighboring blocks. As a predictor of the IC parameter of a block, we use the IC parameter of the block to its left; this allows prediction to be performed in a causal manner. If the left block was not encoded using IC, the block above is used instead as a predictor. If IC is disabled for both of these blocks then no prediction is used to encode the IC parameter for the current block (equivalently, the predictor is set to zero).

The prediction residue is quantized and then encoded. We use a simple uniform quantizer, which offers good performance and low complexity. A more complex two-dimensional uniform vector quantizer design was proposed in [19]. This quantized differential offset is encoded using a binary arithmetic coder (BAC). We first separate the absolute value *(val)* and the sign of these quantized differential offsets. Then, the absolute values of quantized offsets are binarized by selecting a unary representation as in Table I. The differential offset parameters are prediction residues which tend to be small and exhibit a symmetric distribution around zero, with very limited spatial correlation. Different probability models are used for different binary symbol positions of val as shown in Table I. While under certain assumptions (e.g., Laplacian distribution) fewer separate models may be needed, we choose this configuration to allow for more flexibility in our modeling. The number of different probability models for binary symbols in val is chosen to be four and initialized experimentally. Bits corresponding to val greater than 3 use the same probability model. Binary symbols are binary arithmetic encoded, with adaptive probability models. Arithmetic coding is also used for the sign, with a probability model initialized with equal symbol probability. Note that online adaptation of the various probability models is applied along BAC.

Clearly, different blocks suffer from different levels of illumination mismatch, so that potential R-D benefits of using IC differ from block to block. Thus, we allow the encoder to decide whether or not the IC parameters are used on a block by block basis. This is achieved by computing the R-D values associated to coding each block with and without IC, and then letting the Lagrangian optimization tools in the H.264/AVC codec make an R-D optimal decision. There is an added overhead needed to indicate for each block whether IC is used but this is more efficient overall than sending IC parameters for all blocks. This IC activation bit is entropy-encoded using the context adaptive binary arithmetic coder (CABAC) [20], which consists of: 1) binarization; 2) context modeling; and 3) binary arithmetic coding. The context is defined based on the activation choices made for the left and upper blocks. If IC is enabled or disabled in *both* these blocks, it is highly probable that the same choice will be made for the current block. However if only one of these two neighboring blocks uses IC, the probability of the current block using IC should be close to 1/2. Based on this observation, three contexts are assigned and initialized for activation switch; this is similar to the context setup for the SKIP flag or the transform size in H.264/AVC.

C. Simulation Results

Three sequences, *Ballroom, Race1* and *Rena*, which have different characteristics are selected for simulation [1]. All test sequences are 640(w) $\times$ 480(h) with eight views as shown in Fig. 4. *Ballroom* has the most complicated background and fast moving objects. Objects are located at multiple depths and the distance from the camera to the front objects is small so the disparity of front objects is large. In *Race1*, a mounted and fixed camera array is used to follow racing carts so that there is global motion. Significant luminance and focus changes between views are observed due to imperfect camera calibration and illumination changes are also observed in time because of global motion by camera. In *Rena*, a gymnast moves fast in front of curtains. Distance between cameras is smaller than in the other sequences and luminance and focus changes between views are observed clearly.

Our proposed IC technique is combined with standard H.264/AVC [21] coding tools. IC is enabled only for 16×16, 16×8, 8×16, and 8×8 blocks. While the encoder could be given the option to select whether to use IC on smaller blocks, we observed that this choice was rarely made and thus, for complexity reasons, we choose 8×8 to be the smallest block size. Also, IC can be applied in skip/direct mode so that model parameters are predicted from neighboring blocks using spatial correlation.

Using the reference codec JM-10.2 [22] as a starting point, we encode frames in cross-view direction only, i.e., we take a

(a) (b) (c)

Fig. 4. MVC sequences : 1D/parallel. (a) Ballroom: 8 cameras with 20-cm spacing, (b) Race1: 8 cameras with 20-cm spacing, (c) Rena: 8 cameras with 5-cm spacing.

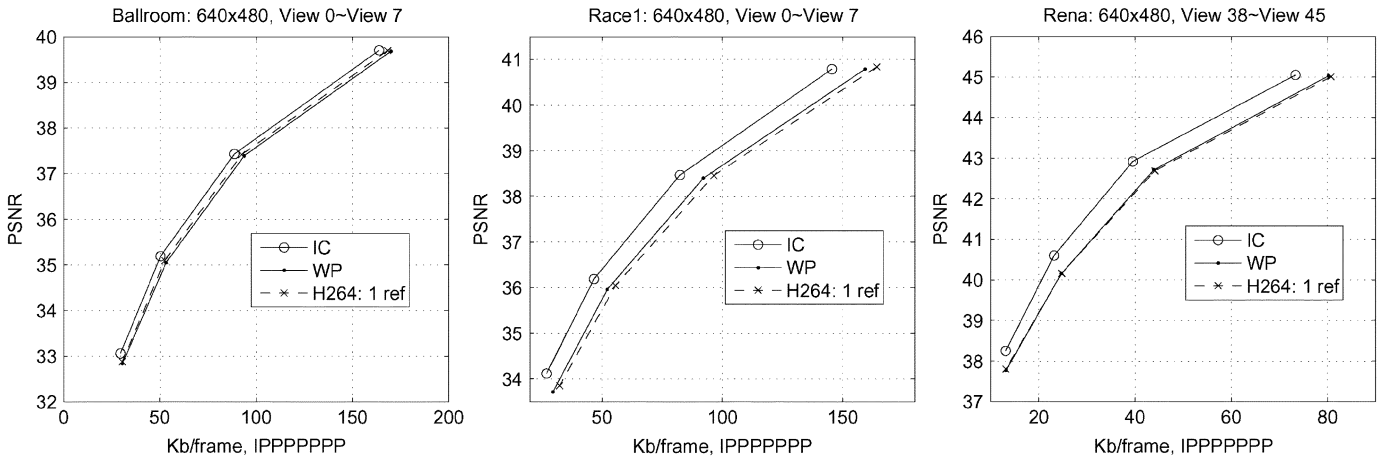


Fig. 5. Cross-view with IC, at time stamps 0. 10. 20. 30. 40.

sequence of frames captured at the same time from different cameras and feed this to the encoder as if it were a temporal sequence. Intra period is set equal to the number of views. Anchor frames at time stamps 0, 10, 20, 30, 40 are encoded with cross-view prediction. We performed simulations (H.264/AVC high profile) with full search, range equal to ± 64 pixels, quarter-pixel precision, 1 reference frame, and tested four different QP values (24, 28, 32, 36) to obtain different rate points in Fig. 5. It can be seen that for *Race1* and *Rena* there is significant improvement by using IC (0.8 dB) because of severe illumination mismatch across views. Instead, *Ballroom* showed small improvement (0.2 dB). *Ballroom* is the most difficult sequence to encode because of complicated background and irregular disparity field due to the large variance of object depth and illumination mismatch is not significant compared to the other sequences. Also, it can be seen that the weighted prediction (WP) does not provide significant coding gains because it cannot compensate severe local mismatches in cross-view prediction. From Table II, we can see that the number of blocks in Inter and Skip mode increases with IC, which means that the disparity search finds correct match after compensation. Also the reduction in residual energy provided by IC leads to the coding gains. Note that IC gains can be observed even at low bit rates because the selection of IC in each block is optimized based on R-D criteria.

Although our proposed IC techniques primarily aimed at compensating illumination mismatches in cross-view prediction, they can easily be used to compensate illumination mismatches in temporal prediction, which happens in moving objects and abrupt scene changes. In general MVC structures, references from different time stamps and views are available [23]. For example, if the current frame is at view 3 and time stamp 1 (v3t1), four references are available for current B-slice—(v3t0),(v3t2),(v2t1) and (v4t1). Fig. 6 demostrates coding results with IBPBPBPP for cross-view prediction and hierarchical B for temporal prediction [23]. For *Ballroom, Race1* and *Rena*, IC achieves 0.1–0.5-dB gains. Overall gains from using IC (as compared to using the same temporal/cross view prediction but no IC) are lower relative to the case where only cross-view prediction is used (Fig. 5) because illumination mismatches between frames in time are not as severe as across

TABLE II
PERCENTAGE OF NON-INTRA-SELECTION IN CROSS-VIEW PREDICTION (% IN H.264 → % IN H.264 + IC). NOTE THAT MORE SIGNIFICANT PSNR INCREASES CAN BE OBSERVED FOR THOSE SEQUENCES WHERE THE INCREASE IN NUMBER OF INTER CODE BLOCKS IS GREATER

Sequence	QP24	QP28	QP32	QP36
Ballroom	68.8 → 75.2	72.3 → 79.9	73.3 → 81.8	77.2 → 85.6
Race1	53.1 → 71.1	53.4 → 71.2	53.6 → 72.6	54.6 → 73.9
Rena	53.0 → 66.9	54.0 → 70.3	56.0 → 72.3	62.5 → 72.8

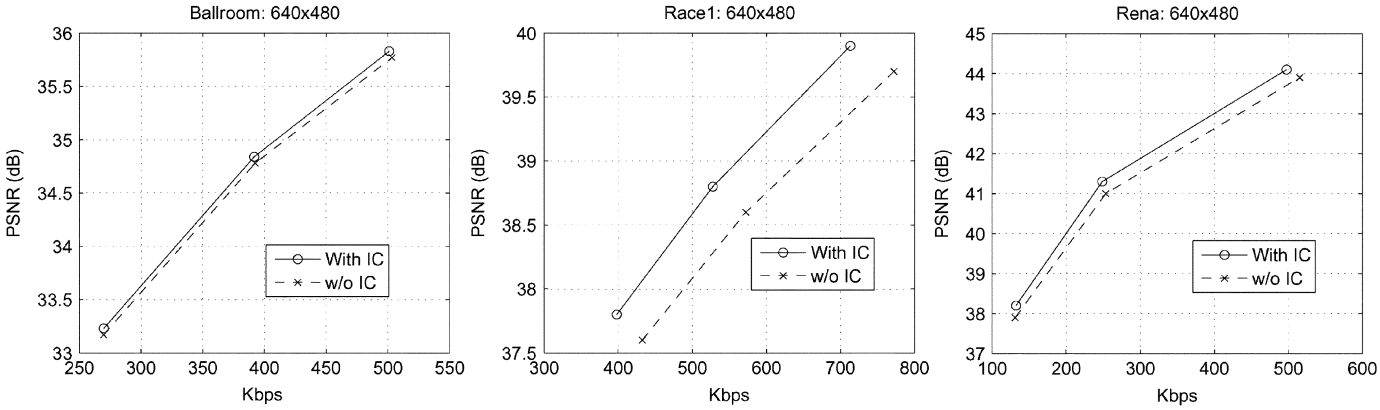


Fig. 6. MVC with IBPBPBPP cross-view, hierarchical B temporal [23].

views and most static background can be efficiently encoded by skip/direct mode in temporal prediction. Complete simulation results of our proposed IC in MVC for various multiview test sequences can be found in [24]–[27]. Comparisons with WP were provided in [28], where it was shown that IC achieves higher coding efficiency as compared to WP. In particular for *Race1*, IC achieves a 0.5-dB gain over WP.

III. ARF for Cross-View Prediction

In the previous section, we introduced block-wise offset and scale parameters to compensate for localized illumination mismatches. Now we consider more general filtering approaches to address depth-dependent focus change across different views in MVC. While the basic coding algorithm was presented in our previous work [17], in this paper we provide more rigorous derivation of the design approach and perform further analysis.

A. Examples of Cross-View Blurriness Discrepancies and Adaptive Filtering Model

Among the multiview video test sequences provided in MVC Call for Proposals document [1], the sequence *Race1* exhibits the most clearly perceivable blurriness discrepancy among different views. We denote its eight views 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. 7 shows portions of the frames from different views in *Race1*.

It can be seen from Fig. 7 that, besides displacement of the scene, frames from different views also exhibit blurriness mismatches. In the literature [29], [30], a blurred (smoothed) or sharpened image G produced from its original version F can be modeled as $G = H * F$, where $*$ denotes the 2-D convolution and H is the point spread function. We propose to model cross-view prediction with blurriness/sharpness mismatches as

$$\text{For each pixel}(x,y): S_{x,y} = \sum_{i,j} H_{i,j} R_{x+dv_x+i, y+dv_y+j} \quad (8)$$

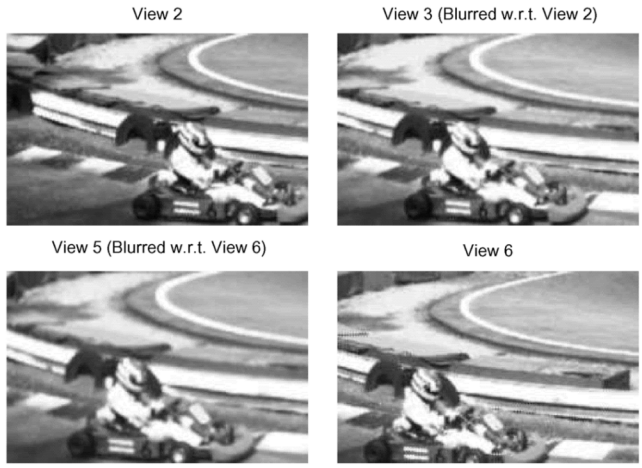


Fig. 7. Portions of frame 15 from different views.

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, and $(x + dv_x, y + dv_y)$ is its corresponding disparity-displaced pixel in the reference frame.

In our approach, the reference frame is first filtered by an estimator of the point spread function H chosen to minimize the error with respect to the current frame. Minimum mean-squared error (MMSE) estimation can be derived by optimizing the following:

$$\min_{\psi, dv_x, dv_y} \sum_{x,y} \left(S_{x,y} - \psi * R_{x+dv_x, y+dv_y}\right)^2. \quad (9)$$

The filter ψ will be an estimator of the point spread function H. To jointly estimate ψ and (dv_x, dv_y), the solution space of ψ has to be defined and at each position of disparity search, the whole ψ solution space has to be tested. Such an encoding system will require excessive computation. Instead, we adopted a procedure similar to that proposed in adaptive interpolation filtering [14]–[16], i.e., such that the disparity field is estimated first and then the filter coefficients of ψ are determined. In this paradigm, the filter will be designed based on the disparity-compensated difference between the reference frame and the current frame. To understand the effect of adaptive filtering in the presence of

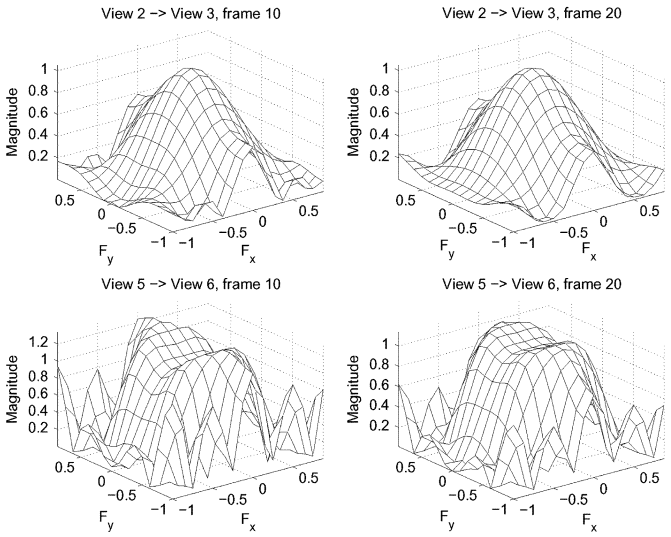


Fig. 8. Frequency responses of calculated MMSE filters.

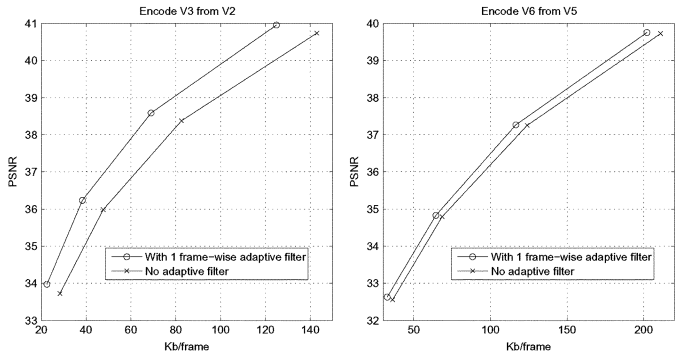


Fig. 9. Encoding results of the frame-wise filtering: Race1 sequence.

blurriness/sharpness focus mismatches, we performed a simple experiment that can be summarized as follows.[2]

1) Initial disparity estimation needed to obtain disparity field (dv_x, dv_y) of the current frame.
2) With the obtained (dv_x, dv_y), calculate MMSE filter ψ such that

$$\min_{\psi} \sum_{x,y} \left(S_{x,y} - \psi * R_{x+dv_x, y+dv_y}\right)^2. \tag{10}$$

3) Estimated adaptive filter ψ is applied to the reference frame to generate a better matched reference $\psi * R$. The final disparity compensation is performed with this filtered reference.

We use H.264/AVC to encode frames at time stamps 0, 10, 20, 30, 40, with cross-view prediction only. In this experiment, we define the solution space of ψ to be that of fixed-size 5×5 filters, symmetrical with respect to x- and y-axis. Fig. 8 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 framewise 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 (frames 10 and 20 as in Fig. 8), the filters for a given view have quite similar frequency responses. This result suggests that the blur effect was likely to be introduced by camera mismatches.

Fig. 9 provides the corresponding rate-distortion results for View 3 and View 6 at QP equal to 24, 28, 32, and 36. 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 available for disparity estimation [31]. As can be seen in our experiments, for QP around 28 and 32, the framewise filtering provides 0.7–0.8-dB gain for frames in View 3 and around 0.3 dB for frames in View 6.

[2]Note that for demonstration purposes, in this experiment we constrained the design to one adaptive filter per frame, as the blur effect appears to be global.

B. Proposed Adaptive Filtering Approach for Cross-View Disparity Compensation

To extend the adaptive filtering approach to situations in which different objects may suffer from different types of blurriness/sharpness changes, locally adaptive compensation has to be enabled by considering depth information of the scene. In this paper, disparity information is used to estimate scene depth. After an initial disparity search, blocks with similar disparity vectors are grouped into classes. Each class represents a scene-depth level D_k and will be associated with one adaptive filter ψ_k to be designed in the next step. We call this process "filter association." For each class (scene-depth level), a filter is optimized to minimize the residual energy for all blocks in the class, as described by (10). This approach will allow multiple filters to be estimated based on different portions of a video frame that undergo different blurriness/sharpness changes with respect to the corresponding areas in frames from the other views. These filters will be applied to the reference frame to provide better matches. Then the final disparity compensation is performed using both original and filtered frames as references. At this stage each block is allowed to select the reference that provides the lowest R-D cost, regardless of what the initial classification of the block was. In the following subsections, we describe each step in detail.

1) Filter Association: The first step is to identify different types of blurriness/sharpness changes in different parts of the current frame. An exhaustive approach could be to assign adaptively to each block a filter that minimizes the matching error. This approach is optimal in the sense that for every block the residue energy is minimized. However, it will significantly increase the bitrate since we have to transmit filter coefficients for every single block. In multiview systems, localized focus mismatches are expected to be associated with depth information. Thus, we consider procedures to identify image regions with different depths. When multiple cameras are employed, disparity information has been widely used as an estimation of scene depth [32]. Given that object depth and its disparity across two views are reciprocals, video frames can be partitioned into different depth levels by exploiting the disparity information [33]–[35].

We consider procedures to classify blocks into depth levels based on their corresponding disparity vectors. When cameras are arranged on the same horizontal line, classification can be

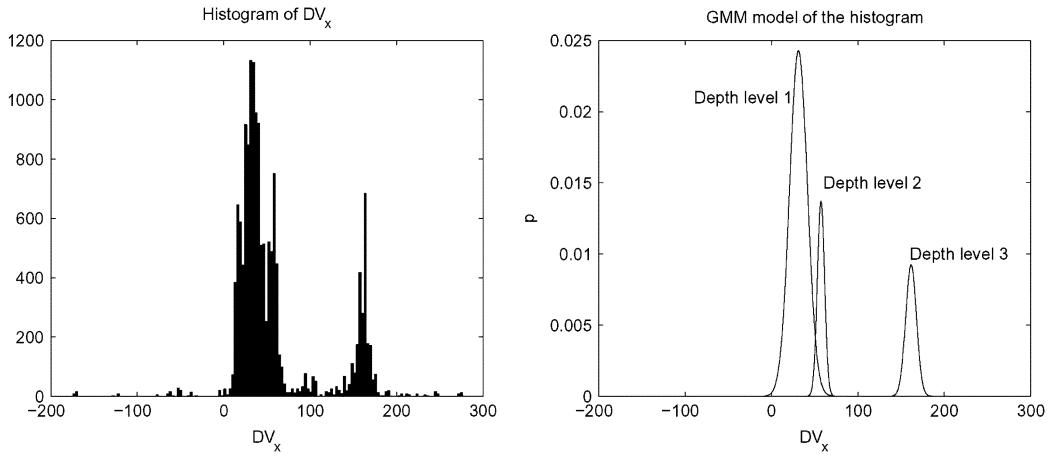


Fig. 10. Disparity vectors from View 6 to View 7 at the first frame in Ballroom: Histogram and GMM.

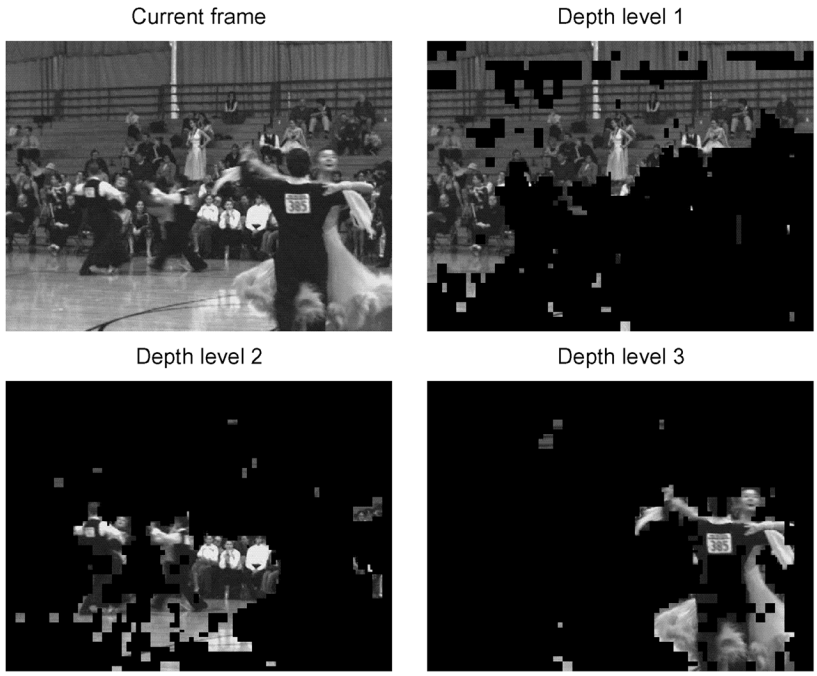


Fig. 11. Corresponding EM classification result of Fig. 10.

achieved by considering only the x component of the disparity vectors. For a 2-D 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 depth-level classes. We adopted expectation-maximization (EM) algorithm based on the GMM [36] to classify the disparity vectors and their corresponding blocks [37]–[39]. In this paper, an unsupervised EM classification tool developed by Bouman [40] is employed. 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. The only required parameter to be specified is the maximum number of Gaussian components (K) allowed in the GMM. It will test hypotheses with the number of Gaussian components from $1, 2, \ldots$ to K with MDL criteria. We refer to [40]–[42] for details about such techniques. Parameters of Gaussian components 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, 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. To demonstrate this filter association based on the classification of disparity vectors, we provide a segmentation result in Figs. 10 and 11.

In Fig. 10, the histogram of the x component of disparity vectors, obtained from the initial disparity estimation, is provided. A corresponding GMM is constructed with a number of components estimated to be 3 (K was set to 4). In Fig. 11, 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 midrange, along with their reflection on the floor; and class 3 includes the couple in the front. Note that intra-coded blocks in the initial disparity estimation are not involved in the filter association process. In this example, the classification tool successfully separates objects with different depths in the current frame.

2) Depth-Adaptive Filter Selection: We now discuss how to select a filter for all blocks belonging to a given depth level class D_k. We replace the convolution notation in (10) by explicitly expressing the filter operation as

$$\min_{\psi_k} \sum_{(x,y)\in D_k} \left(S_{x,y} - \sum_{j=-n}^{n} \sum_{i=-m}^{m} \psi_{i,j} R_{x+dv_x+i,y+dv_y+j} \right)^2. \tag{11}$$

The size and shape of 2-D filters can be specified by changing m and n. In adaptive interpolation filtering (AIF) approaches, even-length (6×6) filters are proposed in order to interpolate subpixels. In our proposed approach, 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.

The filter coefficients $\psi_{i,j}$ that satisfy (11) can be determined by taking derivative with respect to each coefficient, i.e., $\forall \psi_{I,J}$ where $-m \le I \le m, -n \le J \le n$

$$\begin{aligned}
&\frac{\partial}{\partial \psi_{I,J}} \sum_{(x,y)\in D_k} \left(S_{x,y} - \sum_{j=-n}^{n} \sum_{i=-m}^{m} \psi_{i,j} R_{\hat{x}+i,\hat{y}+j} \right)^2 = 0 \\
&\sum_{(x,y)\in D_k} \left(S_{x,y} - \sum_{j=-n}^{n} \sum_{i=-m}^{m} \psi_{i,j} R_{\hat{x}+i,\hat{y}+j} \right) R_{\hat{x}+I,\hat{x}+J} = 0 \\
&\sum_{j=-n}^{n} \sum_{i=-m}^{m} \left(\psi_{i,j} \sum_{(x,y)\in D_k} R_{\hat{x}+i,\hat{y}+j} R_{\hat{x}+I,\hat{y}+J} \right) \\
&\quad = \sum_{(x,y)\in D_k} S_{x,y} R_{\hat{x}+I,\hat{y}+J}.
\end{aligned} \tag{12}$$

These Wiener-Hopf equations will lead to optimal linear Wiener filters. The number of equations will be equal to the number coefficients of ψ. To reduce the number of unknowns in this adaptive filter estimation, constraints such as symmetry can be imposed. Filters with more unknowns can be more efficient to compensate for residue energy. However, this comes at the expense of having to 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, filters are designed to compensate for focus mismatches, which are generally assumed to be *isotropic* [43], [30], [29]. Thus we use as an *example* 5×5 filters ($m = n = 2$), with the coefficients selected as

$$\psi = \begin{pmatrix} a & b & c & b & a \\ d & e & f & e & d \\ g & h & j & h & g \\ d & e & f & e & d \\ a & b & c & b & a \end{pmatrix}. \tag{13}$$

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 (13) where we have chosen $h = f, g = c$, and so on. For each depth level, a filter in the above form will be obtained as a solution to (12).

3) Disparity Compensation With Local Adaptive Filtering: The obtained adaptive filters will be applied to the reference frame to provide better matches for cross-view prediction. In the reference picture list, the original unfiltered reference as well as multiple filtered references are stored. If subpixel disparity estimation is employed, all these references will be interpolated to generate subpixel values using interpolation filters specified by the codec (e.g., 6-tap interpolation filters in H.264/AVC). During the final encoding process, original and filtered references can be regarded as the input for predictive coding with multiple references, such as specified in H.264/AVC [21]. This provides two advantages: Firstly and most importantly, each block can select a block in any filtered or original reference frame, based on R-D optimization. This ensures highest coding efficiency. Secondly, the filter selection of each block can easily be handled by signaling the reference frame index in the bitstream.

To correctly decode the video sequence, the filter coefficients also have to be transmitted. In this paper, we directly extend the method proposed in [44] and [45], in which the filter coefficients are quantized and encoded as frame level overhead.

C. Comparisons With Adaptive Interpolation Filtering and H.264/AVC Multiple Reference Approaches

To justify the efficiency of the proposed ARF approach, we performed simulations based on JM 10.2 High Profile to encode multiview video in cross view direction. Again the search range is set to ±64 pixels and the compensation is performed with quarter-pixel precision. The proposed ARF is compared with AIF and current H.264/AVC video coding. Motion compensation with multiple references as specified in H.264/AVC [21] also aims to improve coding efficiency by providing better matches. Our proposed ARF utilizes multiple filtered versions from a *single* reference frame. In these simulations, the maximum number of classes allowed (K) was set to 4. Thus, we also compare our method to H.264/AVC with the number of reference frames set to 5. We encode the anchor frames at different time stamps to evaluate different cross-view prediction schemes. The rate-distortion results are provided in Fig. 12.

It can be seen that the proposed ARF method provides higher coding efficiency than other approaches we tested. Moreover, the coding gain is higher for sequences with stronger focus mismatches. For *Race1* we observe gains of 0.3, 0.6, and 0.8 dB over AIF, H.264/AVC with five references, and H.264/AVC with

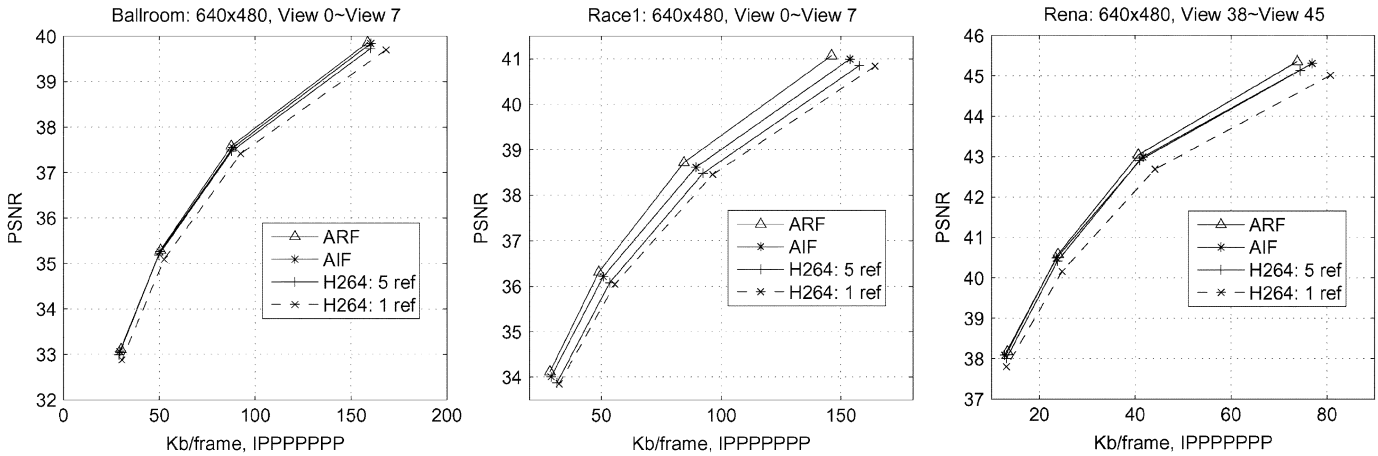


Fig. 12. Cross-view coding results at time stamps 0, 10, 20, 30, 40.

one reference, respectively. Simulations based on JMVM 3.0 for cross-view coding [46] also shows very similar results.

IV. Combination of IC and ARF Techniques

In previous sections, we demonstrated two separate compensation techniques for illumination and focus mismatches. In multiview systems, these two types of mismatch can occur simultaneously. To efficiently compensate for *both* illumination and focus changes across different views, we propose to apply the two techniques together for cross-view prediction.

We note that ARF generates new predictors by applying frame-level filtering. The encoder then selects blocks within these filtered frames. Instead, IC is applied to each block independently so that a block-specific illumination mismatch parameter is computed and no other blocks are needed to generate the new predictor. Note also that in effect IC searches matching mean-removed block patterns and separately conveys the difference in average values for corresponding blocks. Since ARF and IC operate on frames and blocks, respectively, a straightforward approach to integrate ARF and IC is to apply ARF first so that multiple reference frames can be created, then perform IC-based disparity compensation with all the references. The directly merged encoding process can be summarized as follows.

Algorithm 1:

i) Initial disparity search to obtain disparity field between current and reference frame. *[first search]*
ii) Using the disparity field obtained from Step i, blocks are classified into different depth-level classes. For each class, adaptive filter coefficients are calculated based on (10).
iii) Multiple filtered references are generated by applying adaptive filters obtained in Step *ii*.
iv) For original and filtered references, disparity compensation is performed with IC. *[second search]*

In this encoding system, all possible predictors provided by ARF and IC are enabled (original reference, filtered reference, original reference plus IC, filtered reference plus IC). However, this leads to some inefficiencies. First, note that in the design of ARF, the constraint of $\sum_{i,j} \psi_{i,j} = 1$ is not imposed. Thus the resulting filters could also have some DC gain [15]. As a result, DC compensation would be performed by both ARF and IC. This is inefficient considering that IC will perform a more accurate, blockwise DC compensation in the final disparity search. This also introduces a potential drawback that, when IC is applied to blocks that select different references generated by ARF, as the efficiency of differential coding of these IC offsets could be reduced.

Second, the complexity of the combined system is fairly high. In the final disparity compensation (Step *iv)*, multiple references have to be searched with IC-based coding on each of them. This leads to a second search having complexity that is the product of the complexity of ARF and IC, as compared to H.264/AVC search scheme with one reference and no IC. To maintain the best compensation capability provided by IC and ARF while reducing the coding complexity, we now propose modifications to *Algorithm 1*.

A. Mean-Removed Search for Initial Disparity Search

To address the potential duplication of DC compensation by ARF and IC, the first adjustment we propose is to modify the initial disparity search (*Step i*) to also utilize mean-removed search (MRS) as described in IC, such that the potential DC effect produced by ARF can be minimized. In determining filter coefficients for adaptive filters, MRS produces block correspondence with best matched patterns *after* mean removal. Therefore in the final disparity compensation, the coding efficiency of IC can still be well preserved even with different references generated by ARF. Moreover, in the presence of various cross-view mismatches, MRS provides higher accuracy for disparity estimation [10]. Thus, the classification result based on disparity vectors will also be improved. Due to these two factors, higher coding efficiency can be achieved by utilizing MRS in the initial disparity estimation.

Besides the above benefit, MRS in the first search also allows us to reduce overall complexity. Since the different filtered references created by ARF come from the same original reference frame, the disparity fields obtained from the first and second search (*Step i* and *Step iv*) should not be very different. In *Algorithm 1*, the two searching steps use different matching criteria:

TABLE III
MEAN OF ABSOLUTE DIFFERENCE BETWEEN DISPARITY VECTORS FROM *STEPS I* AND *IV*

Sequence	First search with normal block matching	First search using MRS
Ballroom	(6.59, 3.86)	(3.11, 2.42)
Race1	(8.07, 3.38)	(2.91, 1.09)
Rena	(6.68, 9.19)	(1.68, 3.20)

The former performs regular block matching while the latter applies IC based mean-removed block matching. With the modification of also using MRS in *Step i*, the two searches will be consistent. Complexity reduction can be achieved by taking the disparity field obtained from the first search as predictor for the second search with multiple filtered references and IC. To analyze the effect of different searching schemes, we conducted experiments with full search range set to ± 64 pixels, and compared the disparity fields from the first and second search. In Table III, two sets of mean absolute difference (MAD) between disparity vectors from the two searching steps are provided: For the initial search, one set is obtained based on normal block matching and the other utilizes MRS. It can be seen from the table that, by changing the first search to MRS, the difference between two disparity fields from the first and second search is reduced significantly. Based on these statistics, for the final disparity compensation in the combined coding system, a much reduced search range can be applied using disparity vectors obtained from the first search as predictors.

B. Depth-Dependent Mean-Removed Adaptive Filter Calculation

In ARF, filter coefficients for different depth-level class are optimized by solving the Wiener–Hopf equations, (12). Further analysis can be performed by writing them in terms of correlations, i.e., $\forall \psi_{I,J}$ where $-m \le I \le m, -n \le J \le n$:

$$\sum_{j=-n}^{n}\sum_{i=-m}^{m}\left(\psi_{i,j}\sum_{(x,y)\in D_k} R_{\hat{x}+i,\hat{y}+j}R_{\hat{x}+I,\hat{y}+J}\right) = \sum_{(x,y)\in D_k} S_{x,y}R_{\hat{x}+I,\hat{y}+J}$$

$$\sum_{j=-n}^{n}\sum_{i=-m}^{m}\psi_{i,j}E\left[R_{\hat{x}+i,\hat{y}+j}R_{\hat{x}+I,\hat{y}+J}\right] = E\left[S_{x,y}R_{\hat{x}+I,\hat{y}+J}\right]$$

$$\sum_{j=-n}^{n}\sum_{i=-m}^{m}\psi_{i,j}\mathrm{Cor}_{\hat{R}\hat{R}}(I-i,J-j) = \mathrm{Cor}_{S\hat{R}}(I,J) \qquad (14)$$

where $E[\cdot]$ is the expectation operator, Cor is correlation function, $(\hat{x},\hat{y}) = (x+dv_x, y+dv_y)$ and $\hat{R}$ denotes the disparity shifted reference frame $R_{x+dv_x,y+dv_y}$. Both $E[\cdot]$ and Cor operate over all the blocks that are classified into the depth-level D_k. It can be seen that the linear MMSE Wiener filter is optimized based on the autocorrelation of the disparity shifted reference frame $\hat{R}$; and the cross-correlation between the current frame and $\hat{R}$. Considering that IC has to compensate DC differences after ARF is applied to the reference frame, we modify

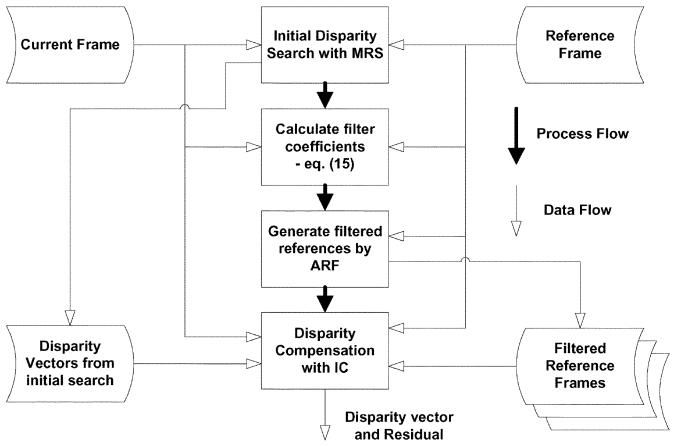


Fig. 13. Combined encoding process for cross-view prediction.

the calculation of filter coefficients such that, the mean of pixels in each class is subtracted. Let us define the mean of each depth-level class as

$$\mu_{S(D_k)} = mean_{(x,y)\in D_k} S_{x,y}$$
$$\mu_{R(D_k)} = mean_{(x,y)\in D_k} R_{x+dv_x,y+dv_y}.$$

From the Wiener–Hopf equations point of view, the mean-removed filter calculation becomes

$$\sum_{j=-n}^{n}\sum_{i=-m}^{m}\psi_{i,j}E\left[\left(R_{\hat{x}+i,\hat{y}+j}-\mu_{R(D_k)}\right)\left(R_{\hat{x}+I,\hat{y}+J}-\mu_{R(D_k)}\right)\right] = E\left[\left(S_{x,y}-\mu_{S(D_k)}\right)\left(R_{\hat{x}+I,\hat{y}+J}-\mu_{\hat{R}(D_k)}\right)\right]$$

$$\sum_{j=-n}^{n}\sum_{i=-m}^{m}\psi_{i,j}\mathrm{Cov}_{\hat{R}\hat{R}}(I-i,J-j) = \mathrm{Cov}_{S\hat{R}}(I,J). \qquad (15)$$

From (14) and (15), it can be seem that the new filter calculation substitutes *correlation information* with *covariance information*. Filters obtained in this manner will not spend their effort on compensating the depth-classwise DC differences. Within their solution space, e.g., $m = n = 2$, symmetric as in (13), filters will accommodate all compensation capability for focus mismatch, leaving the average DC unchanged. The remaining DC error can be further compensated efficiently by IC offset.

C. Fast Search With ARF References and IC

As described in Section IV-A, disparity vectors obtained from the initial mean-removed search can be used as predictors for the final search to reduce complexity. This can be achieved by creating extra memory to hold the initial disparity vectors. When searching over multiple references generated by ARF and combining with IC tools, each block will use its corresponding initial vector as search center to perform disparity compensation with much smaller searching range. To evaluate the performance of the reduced search, we perform simulations by changing the final search range from ± 64 to ± 4 with the initial vectors as predictors. The R-D degradation observed is negligible, while the total encoding time is reduced to about $1/4$. In the remaining of

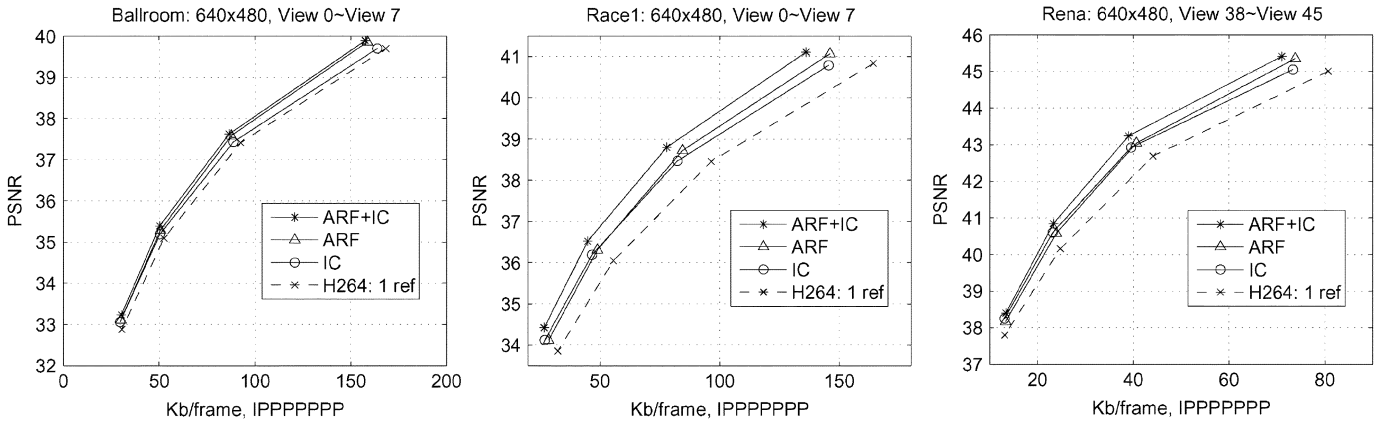


Fig. 14. Cross-view coding results at time stamps 0, 10, 20, 30, 40.

this paper, the combined coding system will be presented with reduced complexity.

The merged encoding procedure with proposed modifications is summarized as follows.

Algorithm 2:

i) *Mean removed Search (MRS)* is performed to obtain initial disparity field. *[first search]*
ii) Using disparity field from Step i, blocks are classified into different depth-level classes. For each class, adaptive filter coefficients are calculated with *mean-removed* class values as (15).
iii) Multiple filtered references are generated by adaptive filters obtained in Step *ii*.
iv) For original filtered references, disparity compensation is performed with IC *using reduced search range*. *[second search]*

Fig. 13 illustrates the block diagram of the new encoding system.

D. Simulation Result

We perform simulations for the combined coding system based on H.264/AVC (JM 10.2). Same setup as described in previous experiments is applied to encode frames across different views. We observe that for all sequences tested, *Algorithm 2* achieves higher coding efficiency as compared to *Algorithm 1*. Also we observe negligible RD reduction when applying reduced search to *Algorithm 2*. Thus in Fig. 14, we provide the RD results of *Algorithm 2* with reduced search range, and compare to ARF and IC by themselves. While ARF already outperforms other approaches such as AIF and multiple reference frame techniques, the combined method achieves even higher coding efficiency. For *Ballroom*, blockwise IC alone provides very limited gain. As a result, the combined system also barely outperforms the ARF coding. On the other hand, ARF and IC each achieves 0.5–0.8-dB gain for *Race1* and *Rena*. The combined system produces an additional 0.5-dB gain over them. The overall coding gain, as compared to using H.264/AVC with 1 reference for cross-view coding, is about 0.5 dB for *Ballroom*, about 1.3 dB for *Race1* and about 1 dB for *Rena*.

TABLE IV
NUMBER OF ADDITION/SUBTRACTION FOR SAD AND SADAC. N IS THE NUMBER OF PIXEL IN A MACROBLOCK AND S IS THE NUMBER OF SEARCH POINTS

SAD (Original)	SADAC (IC enabled)	
$\Sigma_N \lvert B_C - B_R^i \rvert \rightarrow 2N$	$\mu_C \rightarrow N$	$\mu_R^i \rightarrow N$
		$C^i = \mu_C - \mu_R^i \rightarrow 1$
		$\Sigma_N \lvert B_C - B_R^i - C^i \rvert \rightarrow 3N$
For S search points $\rightarrow 2NS$	N	For S search points $\rightarrow 4NS + S$
$2NS$	$N + S + 4NS \approx 4NS$	

V. COMPLEXITY ANALYSIS

In this section, we provide a complexity analysis of our proposed techniques. We consider first IC and ARF, and then the combined system.

The impact of IC on encoding complexity is mostly due to changes in the disparity estimation metric computation (other changes to the encoder such as encoding of IC parameter and R-D based IC activation, have a negligible effect on overall complexity). Thus, in what follows, additional complexity for IC is analyzed in terms of the number of "addition/subtraction" operations in the SAD calculation. As can be seen in Table IV, in each block mode, IC requires 4 NS calculations for SADAC, while 2 NS are required in SAD. For SAD, the differences of current and reference pixels (N) are calculated first. After absolute operation, N absolute differences are summed up to SAD, which requires total $2N$ operations. Similarly, for SADAC a total of 3 N operations are required after μ_C, μ_R and C^i calculations. For the mean calculation, we need to sum N pixels, which requires N additions. In Table IV, we assume that shift operation for mean calculation and absolute operation are not counted in the analysis. Assuming the center of search for different block modes does not deviate significantly, μ_R^i in small blocks can be reused in larger blocks avoiding redundant calculations. For example, the use of saved μ_R^i in 8×8 blocks reduces the complexity SADAC from 4 NS to 3 NS(*Fast IC mode*).

Considering different block modes supported in H.264/AVC, complexity for SAD calculation is summarized in Table V. For

TABLE V
COMPLEXITY FOR SAD AND SADAC CALCULATION IN DIFFERENT BLOCK MODES. N IS THE NUMBER OF PIXEL IN A MACROBLOCK AND S IS THE NUMBER OF SEARCH POINTS

Block Modes	Original (SAD)	IC (SADAC)	Fast IC (SADAC)
4×4	$2NS$	-	-
4×8	$2NS$	-	-
8×4	$2NS$	-	-
8×8	$2NS$	$4NS$	$4NS$
8×16	$2NS$	$4NS$	$3NS$
16×8	$2NS$	$4NS$	$3NS$
16×16	$2NS$	$4NS$	$3NS$
TOTAL	$14NS$	$16NS$	$13NS$

TABLE VI
COMPLEXITY WHEN SAD AND SADAC ARE CALCULATED AT THE SAME TIME. N IS THE NUMBER OF PIXEL IN A MACROBLOCK AND S IS THE NUMBER OF SEARCH POINTS

Block Modes	'SAD+SADAC'	'SAD+SADAC' in Fast IC mode
4×4	$2NS$	$2NS$
4×8	$2NS$	$2NS$
8×4	$2NS$	$2NS$
8×8	$5NS$	$5NS$
8×16	$5NS$	$4NS$
16×8	$5NS$	$4NS$
16×16	$5NS$	$4NS$
TOTAL	$26NS$	$23NS$

IC, both SAD and SADAC need to be calculated for IC activation thus, the total complexity for IC would be the sum of 14 NS + 16 NS (or 13 NS for fast IC mode). Therefore total complexity with IC is about 2.1 (or 1.9 for fast IC mode) times to H.264/AVC without IC. However, this complexity can be reduced further noting that the same search range is used for SAD and SADAC with IC. In the calculation of SADAC in Table IV, $B_C - B_R^i$ can be used to calculate SAD, so that SAD and SADAC are calculated at the same time for the same search point, which requires only N operations for SAD instead of $2N$. This leads to the total complexity with IC in fast mode would be about 1.64 times to H.264/AVC without IC, as can be seen in Table VI.

For ARF, the additional complexity can be decomposed into three parts: a) classification of the disparity vectors into different depth-level classes; b) calculation of the filter coefficients; and c) generations of filtered reference. With the maximum number of Gaussian components set to K, depth classification is based on performing the EM algorithm, for $k = K, K-1, \ldots 1$, to cluster the disparity vectors. The k which provides the lowest minimum description length (MDL), denoted k' will be selected to create the final model. The complexity of this unsupervised classification will be proportional to the *number of vectors* to be classified and the *maximum number* K. To speed up this process, one can consider performing classification with sub-sampled vectors and classifying the corresponding blocks. For the simulations provided in this paper, vectors are generated for each 4×4 block, as specified by H.264/AVC. Assume there are P pixels within a frame, the number of vectors to be classified is $P/16$. If we decimate the disparity field, e.g., select one vector for every four 4×4 blocks, the classification input size will be reduced by four as well. The possible degradation due to such sub-sampling will only be significant on detailed object boundaries, where smaller block size as 4×4 have to be used to differentiate disparity.

To analyze the complexity of filter calculation, let us denote C the number of distinct filter coefficients and P_{D_k} the number of pixels in depth class D_k. As shown by (12), constructing the Wiener–Hopf equation for one coefficient in one depth class requires $P_{D_k}C + C + P_{D_k}$ addition/multiplication operations to calculate the sum of products. Thus, for all filters the total number of operations will be $\sum_k C(P_{D_k}C + C + P_{D_k})$, which is upper bounded by $C(PC + C + P) \approx PC(C+1)$, as intra coded blocks will not be assigned to any depth class. Solving the linear system for each filter with Wiener-Hopf equations requires operations in the order of C^2. For all filters it will become $k'C^2$, which is relatively small compared to the previous term. As a result, the total number of operations in (b) can be approximated by $PC(C+1)$. Note that this will be similar to the complexity of applying different interpolation to generate frame data at sub-pel positions, as specified by H.264/AVC. Finally, to generate filtered references, the convolution operations will lead to $k'PC$ additions and multiplications. Again the complexity is similar to the calculation of subpel values with interpolation filters as in H.264/AVC.

We measure the execution time by performing ARF encoding in three steps: Initial disparity compensation, filter estimation, and final disparity compensation. On average, without any complexity reduction method, (a), (b), and (c) together lead to an increase about 25% in execution time as compared to a cross-view coding by H.264/AVC with one reference, ±64 search range. If incorporating the final disparity compensation which also use this same range on all the references, the total encoding time of ARF is about four times with respect to H.264/AVC with one reference. However, as proposed in Section IV-C, significant complexity reduction can be achieved by using a much smaller search range in the second search, utilizing the disparity field from the initial search as predictor.

In the combined approach in *Algorithm 2*, mean removed search in Step *i* has the same complexity as SADAC as in Table IV, i.e., 4 NS. Steps *ii–iii* have the same complexity analyzed as (a), (b), and (c). In Step *iv*, IC is applied to the filtered reference with reduced search range (e.g., from ±64 to ±4) thus, the number of reduced search points $S' = S/256$. Therefore, the search complexity in Step *iv* would be $(\mathrm{COMP_{IC}})/(S_1/S_2) \cdot (k' + 1)$, where $\mathrm{COMP_{IC}}$ is the complexity of 'SAD + SADAC' as in Table VI, S_1 and S_2 are

the number of search points in the first (step i) and the second (step *iv)* disparity search and $k'+1$ is the number of reference frames including the unfiltered one. If $S_1 = 64^2, S_2 = 4^2$ and $k' = 4, (\mathrm{COMP_{IC}})/(S_1/S_2) \cdot (k'+1) \approx 0.02 \cdot \mathrm{COMP_{IC}}$ which is only 2% of the complexity for IC (SAD+SADAC) and less than 4% of the complexity for MRS (SADAC). Therefore, in most cases, total complexity of combined system will be similar to the the sum of MRS (Step *i)* and ARF (Step *ii–iii)* complexity.

VI. Conclusions and Future Work

To compensate localized illumination and focus mismatches across different views in multiview systems, we proposed block-wise illumination compensation techniques and a depth-dependent adaptive filtering approach. Both coding tools are developed from the corresponding mismatch models and showed significant gains over standard H.264/AVC in cross-view prediction. To efficiently compensate both mismatches, we proposed a coding system that combines IC and ARF. Joint coding benefit and complexity of the combined system are discussed and an improved coding algorithm is presented. Simulation results show that, when performing predictive coding across different views in multiview systems, our proposed methods provide higher coding efficiency than other advanced coding tools. While most of the simulation results evaluated cross-view coding efficiency, our proposed techniques can also be applied to general MVC system where both temporal and cross-view prediction are used and to more general prediction structures.

Although our proposed IC technique is applied only to the luminance component, this can be easily extended to solve the mismatches in chrominance channel, i.e., color compensation (CC). Because the human eye is less sensitive to chrominance than luminance, the chrominance channel is compressed and in YUV420 format, subsampled at a factor of 2 both horizontally and vertically. Furthermore, the original signal is converted to YUV so that most information is condensed in luminance channel, therefore color mismatches may not be considered as critical as illumination mismatches. However, for high resolution content color mismatches could be a problem and may need to be compensated for both objective and subjective quality improvement. CC model parameters for chrominance can be found using (4)–(5) based on the same disparity vectors used for luminance.

In differential coding of IC parameter, disparity vector and filter coefficients, correlation from multiview video structure could be used. From multiview video structure in Fig. 1, anchor frames are only cross-view coded. The frames following anchor frames in time share the same background and only moving objects changes between frames in time. Thus, disparity vectors and IC models at an anchor frame can be used for the following frames. For example, disparity vectors and IC model parameters at the anchor frames of time-(t) can be saved for each view. For the disparity compensation with IC at time-$(t+1)$, disparity vectors and IC parameters for the block can be predicted from previous anchor frame coding of colocated block and improve differential coding results. As for filter coefficients generated by ARF, differential coding can also be applied to filters of a given view with respect to different time stamps. Considering scenes changes over time, an ordering mechanism should be imposed to the classification results such that the classes will be corresponding to depth levels with a consistent ascending or descending order. All the saved information from time-(t) can be updated if it changes at time-$(t+1)$ and this updated information provides more correct estimate for disparity vector and IC parameters of the frames at time-$(t+2)$.

It is possible to extend our ARF approach to B-frames. However, a region to be encoded within a B-frame may suffer from different types of focus mismatch with respect to the corresponding regions from the two views used for prediction. Whether to design two sets of filters for predictors from the two neighboring views, or just to design one set of filters for the average predictor, is a topic to be further investigated.

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. Gilge, "Motion estimation by scene adaptive block matching and illumination correction," in *Proc. Image Process. Algorithms and Techn.*, R. J. Moorhead and K. S. Pennington, Eds., Canada, June 1990, vol. 1244, pp. 355–366.

[5] S. H. Kim and R.-H. Park, "Fast local motion-compensation algorithm for video sequences with brightness variations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 4, Apr. 2003.

[6] W. Niehsen and S. Simon, "Block motion estimation using orthogonal projection," in *Proc. IEEE Int. Conf. on Image Process. (ICIP)*, Lausanne, Switzerland, Sep. 1996, pp. I.16–I.19.

[7] K. Kamikura, H. Watanabe, H. Jozawa, H. Kotera, and S. Ichinose, "Global brightness-variation compensation for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, Dec. 1998.

[8] Y. Altunbasak, R. Mersereau, and A. Patti, "A fast parametric motion estimation algorithm with illumination and lens distortion correction," *IEEE Trans. Image Process.*, vol. 12, no. 4, pp. 395–408, Apr. 2003.

[9] D. Liu, Y. He, S. Li, Q. Huang, and W. Gao, "Linear transform based motion compensated prediction for luminance intensity changes," in *Proc. of IEEE Int. Symp. on Circuits Syst. (ISCAS)*, Beijing, China, May 2005, pp. I.304–I.307.

[10] J. Lopez, J. Kim, A. Ortega, and G. Chen, "Block-based illumination compensation and search techniques for multiview video coding," in *Picture Coding Symp. (PCS) 2004*, Dec. 2004.

[11] J. M. Boyce, "Weighted prediction in the H.264/MPEG AVC video coding standard," in *Proc. of IEEE Int. Symp. on Circuits Syst. (ISCAS)*, Vancouver, Canada, May 2004, pp. III 789–792.

[12] P. Yin, A. M. Tourapis, and J. Boyce, "Localized weighted prediction for video coding," in *Proc. IEEE Int. Symp. on Circuits Syst. (ISCAS)*, May 2005, pp. 4365–4368.

[13] M. Budagavi, "Video compression using blur compensation," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, Genoa, Italy, Sept. 2005, pp. II.882–II.885.

[14] T. Wedi, "Adaptive interpolation filter for motion compensated prediction," in *Proc. IEEE ICIP*, Rochester, NY, Sep. 2002, pp. II.509–II.512.

[15] T. Wedi, "Adaptive interpolation filters and high-resolution displacements for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 484–491, Apr. 2006.

[16] 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 *Proc. IEEE ICIP*, Genoa, Italy, Sept. 2005, pp. II.894–II.897.

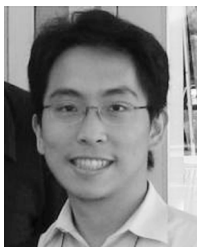
[17] P. Lai, Y. Su, P. Yin, C. Gomila, and A. Ortega, "Adaptive filtering for cross-view prediction in multi-view video coding," in *Proc. SPIE Visual Commun. Image Processing (VCIP)*, San Jose, CA, Jan./Feb. 30–1, 2006.

[18] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, M. Landy and J. A. Movshon, Eds., Jul. 1991.

[19] J. Lopez, "Block-based compression techniques for multiview video coding," Master's thesis, Univ. Politecnica de Catalunya, Barcelona, Spain, 2005.
[20] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.
[21] 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, July 2003.
[22] Software implementation of H.264: JM Version 10.2 The Image Communication Group at Heinrich Hertz Institute, July 2006 [Online]. Available: http:iphome.hhi.de/suehring/tml/index.htm
[23] "Description of core experiments in MVC," ISO/IEC JTC1/SC29/WG11 MPEG Document W8019, Jul. 2005.
[24] Y. Su, P. Yin, C. Gomila, J. Kim, P. Lai, and A. Ortega, "Thomson's response to MVC CfP," ISO/IEC-JTC1/SC29/WG11 MPEG Document M12969/2, Jan. 2006.
[25] J. Kim, P. Lai, A. Ortega, Y. Su, P. Yin, C. Gomila, and P. Pandit, "Results of CE2 on multi-view video coding," ISO/IEC JTC1/SC29/WG11 MPEG Document M13317, Apr. 2006.
[26] J. Kim, P. Lai, A. Ortega, Y. Su, P. Yin, and C. Gomila, "Results of CE2 on multi-view video coding," ISO/IEC JTC1/SC29/WG11 MPEG Document M13720, Jul. 2006.
[27] A. Vetro, Y. Su, H. Kimata, and A. Smolic, "Joint multiview video model (JMVM) 2.0," *ISO/IEC JTC1/SC29/WG11 MPEG Document N8459*, Oct. 2006.
[28] Y. Lee, J. Hur, Y. Lee, K. Han, S. Cho, N. Hur, J. Kim, J. Kim, P. Lai, A. Ortega, Y. Su, P. Yin, and C. Gomila, "CE11: Illumination compensation," *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG JVT-U052*, Oct. 2006.
[29] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Reading, MA: Addison-Wesley, 2002.
[30] W. K. Pratt, *Digital Image Processing*, 3rd ed. Hoboken, NJ: Wiley-Interscience, 2001.
[31] Y. Vatis and J. Ostermann, "Locally adaptive non-separable interpolation filter for H.264/AVC," in *Proc. IEEE ICIP*, Atlanta, GA, Oct. 2006.
[32] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.*. Upper Saddle River, NJ: Prentice Hall, 2003.
[33] T. Aach and A. Kaup, "Disparity-based segmentation of stereoscopic foreground/background image sequences," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 673–679, Feb.–Apr 1994.
[34] E. Francois and B. Chupeau, "Depth-based segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 237–239, Jun. 1997.
[35] E. Izquierdo, "Disparity/segmentation analysis: Matching with an adaptive window and depth-driven segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 589–607, Jun. 1999.
[36] E. Redner and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, Apr. 1984.
[37] Z. Wang, G. Liu, and L. Liu, "A fast and accurate video object detection and segmentation method in the compressed domain," in *Proc. IEEE Int. Conf. on Neural Networks and Signal Process.*, Nanjing, China, Dec. 2003, pp. II.1209–II.1212.
[38] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: A compressed domain approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 462–474, Apr. 2004.
[39] K. Y. Wong and M. E. Spetsakis, "Motion segmentation by EM clustering of good features," in *Proc. IEEE Comput. Vision and Pattern Recogn. Workshop (CVPR)*, Washington, DC, Jun. 2004, pp. 166–173.
[40] C. A. Bouman, "Cluster: An unsupervised algorithm for modeling Gaussian mixture," Jul. 2005 [Online]. Available: http:cobweb.ecn.purdue.edu/bouman/software/cluster
[41] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. 39, no. 1, pp. 1–38, 1977, Series B.
[42] J. Rissanen, "A universal prior for integers and estimation by Minimum Description Length," *J. Ann. Statist.*, vol. 11, no. 2, pp. 417–431, 1983.
[43] H.-C. Lee, "Review of image-blur models in a photographic system using the principles of optics," *SPIE Opt. Eng.*, vol. 20, no. 5, pp. 405–421, May 1990.
[44] 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 *Proc. SPIE Visual Commun. Image Process.*, San Jose, CA, Jul. 2005, pp. 623–631.
[45] Y. Vatis, Software Implementation of Adaptive Interpolation Filter Nov. 2005 [Online]. Available: http:iphome.hhi.de/suehring/tml/download/KTA/
[46] P. Lai, A. Ortega, P. Pandit, P. Yin, and C. Gomila, "Adaptive reference filtering for MVC," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG JVT-W065, Apr. 2007.

Jae Hoon Kim (S'04) received the B.S. degree from Seoul National University, Seoul, Korea, in 1999, and the M.S. degree from the University of Southern California, Los Angeles, in 2003, both in electrical engineering.

From 1999 to 2001, he was a research engineer at LG Electronics, Inc., Seoul, Korea. He is working toward his Ph.D. degree in the Department of Electrical Engineering at University of Southern California, Los Angeles. In the summer of 2005, he was an intern with Thomson Corporate Research, Princeton, New Jersey, USA. His research interests include image and video processing and multiview video coding.



PoLin Lai (S'03) received the B.S. degree from National Taiwan University in 2002, and the M.S. degree from the University of Southern California, Los Angeles, in 2004, both in electrical engineering. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering, University of Southern California, Los Angeles.

He was a summer intern in 2006 and 2007 at Thomson Corporate Research, Princeton, NJ. His research focuses on advance coding tools for multi-view video coding, which targeting on high coding efficiency and complexity reduction.

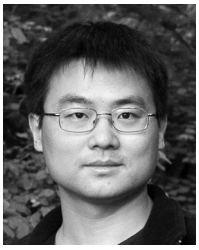
Joaquin Lopez received the B.S.-M.S. degree in electrical engineering from the Polytechnical University of Catalonia (UPC), Barcelona, Spain, in 2005. In 2004, he was with the Signal and Image Processing Institute, University of Southern California, Los Angeles, carrying out his Master Thesis on a visitor program.

In 2006, he joined Matrix Technology, Newport Beach, CA, as Video Coding Engineer. His current research interests include digital image and video compression, motion estimation and compensation, still image segmentation and biomedical applications.

Antonio Ortega (F'07) received the Telecommunications Engineering degree from the Universidad Politecnica de Madrid, Madrid, Spain, in 1989, and the Ph.D. in electrical engineering from Columbia University, New York, in 1994, where he was supported by a Fulbright scholarship.

In 1994 he joined the Electrical Engineering-Systems department at the University of Southern California, where he is currently a Professor and Associate Chair of the Department. His research interests are in the areas of multimedia compression and communications. His recent work is focusing on distributed compression, multiview coding, compression for recognition and classification applications, and wireless sensor networks.

Dr. Ortega is a member of the Association of Computing Machinery (ACM). He has been Chair of the Image and Multidimensional Signal Processing (IMDSP) technical committee and a member of the board of governors of the IEEE SPS (2002). He was the technical program co-chair of ICME 2002, and has served as Associate Editor for the IEEE Transactions on Image Processing and the IEEE Signal Processing Letters. He received the NSF CAREER award, the 1997 IEEE Communications Society Leonard G. Abraham Prize Paper Award, the IEEE Signal Processing Society 1999 Magazine Award and the 2006 EURASIP *Journal of Advances on Signal Processing* Best Paper Award.



Yeping Su was born in Rugao, JiangSu, China. He received the Ph.D. degree in electrical engineering at University of Washington at Seattle in 2005.

From 2005 till 2006, he was a Member of Technical Staff at Thomson Corporate Laboratory at Princeton, NJ. he is currently a senior researcher at Sharp Laboratories of America, Camas, WA. His research interests include digital video processing and multimedia communication.

Peng Yin received the B.E. degree in electrical engineering from University of Science and Technology of China in 1996 and Ph.D. degrees in electrical engineering from Princeton University in 2002.

She is currently a senior member of the technical staff at Corporate Research, Thomson Inc., Princeton, NJ. Her current research interest is mainly on image and video compression. Her previous research is on video transcoding, error concealment and data hiding. She is actively involved in JVT/MPEG standardization process.

Dr. Yin received the IEEE Circuits and Systems Society Best Paper Award for her aricle in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2003.

Cristina Gomila received the M.S. degree in telecommunication engineering from the Technical University of Catalonia, Spain, in 1997, and the Ph.D. degree from the Ecole des Mines de Paris, France, in 2001.

She then joined Thomson Inc., Corporate Research Princeton, Princeton, NJ. She was a core member in the development of Thomson's Film Grain Technology and actively contributed to several MPEG standardization efforts, including AVC and MVC. Since 2005, she manages the Compression Research Group at Thomson CR Princeton. Her current research interests focus on advanced video coding for professional applications.