# Focus Mismatches in Multiview Systems and Efficient Adaptive Reference Filtering for Multiview Video Coding

PoLin Lai[*,†], Antonio Ortega[*], Purvin Pandit[†], Peng Yin[†], and Cristina Gomila[†]

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

## ABSTRACT

In this paper, we analyze focus mismatches among cameras utilized in a multiview system, and propose techniques to efficiently apply our previously proposed adaptive reference filtering (ARF) scheme to inter-view prediction in multiview video coding (MVC). We show that, with heterogeneous focus setting, the differences exhibit in images captured by different cameras can be represented in terms of the focus setting mismatches (view-dependency) and the depths of objects (depth-dependency). We then analyze the performance of the previously proposed ARF in MVC inter-view prediction. The gains in coding efficiency show a strong view-wise variation. Furthermore, the estimated filter coefficients demonstrate strong correlation when the depths of objects in the scene remain similar. By exploiting the properties derived from the theoretical and performance analysis, we propose two techniques to achieve efficient ARF coding scheme: i) *view-wise ARF adaptation* based on RD-cost prediction, which determines whether ARF is beneficial for a given view, and ii) *filter updating based on depth-composition change*, in which the same set of filters will be used (i.e., no new filters will be designed) until there is significant change in the depth-composition within the scene. Simulation results show that significant complexity savings are possible (e.g., the complete ARF encoding process needs to be applied to only 20% ∼35% of the frames) with negligible quality degradation (e.g., around 0.05 $dB$ loss).

**Keywords:** multiview video coding, focus mismatches, adaptive filtering, disparity compensation

## 1. INTRODUCTION

Multiview video systems utilize multiple cameras to simultaneously capture scenes from different viewpoints. As compared to conventional monoscopic video, frames from different views are prone to suffer from mismatches other than simple displacement, due to heterogeneous camera settings and/or shooting positions. One example is the focus mismatch which results in blurriness/sharpness discrepancy among different views. To compensate for such mismatch, we previously proposed a depth-related adaptive reference filtering (ARF) approach[1,2] for encoding frames across different views in multiview video coding (MVC). In the proposed coding scheme, to encode a video frame $S$, an initial disparity estimation is performed first. It obtains block-wise disparity vectors (DV) and the corresponding predictor set $P$ for blocks in the current frame $S$. By noting that the disparity field provides an estimation of scene depth, we use disparity vectors to classify each video frame, $S$, into regions $S^1$, $S^2$, ... $S^k$, each corresponding to different depth classes. Each region (depth level) $S^i$ includes multiple blocks; we denote $P^i$ the set of block-wise predictors corresponding to blocks in region $S^i$. A constrained 2D spatial filter $\psi^i$ is estimated for each depth level $S^i$ to minimize the mean-squared prediction error between $S^i$ and $P^i$:

$$\min_{\psi^i} \sum_{x,y} \left(S^i_{x,y} - \psi^i * P^i_{x,y}\right)^2, \text{ where } (x,y) \text{ is the pixel position, and } * \text{ denotes convolution} \qquad (1)$$

The estimated filters are then applied to the reference frame to create filtered references. Finally in the encoding stage, for each block in $S$, the encoder selects the predictor (filtered or unfiltered) that provides the lowest rate-distortion cost (RD-cost), thus ensuring highest coding efficiency. The gain in coding efficiency from the ARF approach we just described comes at the expense of higher encoding complexity, in particular because this is a *two-pass encoding scheme* (initial disparity estimation and final encoding with filtered references).

Further author information:
Send correspondence to PoLin Lai: polinlai@usc.edu

Without any prior knowledge about the mismatch, the initial search and filter estimation are necessary in order to adaptively design filters.

In this paper, we first analyze focus mismatches among cameras utilized in a multiview system. We show that the differences that appear in the images can be represented in terms of the focus setting mismatches (view-dependency) and the depths of objects (depth-dependency). For 1D parallel camera arrangement, we further demonstrate that the depth-dependency can be associated with the disparity exhibited in the images. This analysis justifies our previously proposed depth-related ARF designed to compensate for focus mismatches. Taking into account the characteristics of multiview systems, the resulting properties can be exploited to achieve a more efficient ARF coding scheme such that the complexity is much reduced while the coding gain is preserved.

Driven by the analytical results, we then study the performance of ARF in MVC inter-view prediction. The gains in coding efficiency provided by ARF show a strong view-wise variation, depending on whether there is focus mismatch, and how severe the mismatch is. Furthermore, for views indeed exhibiting focus mismatch, the estimated filter coefficients at different timestamps demonstrate strong correlation when the objects' depths remain similar in corresponding captured scene. These observations are consistent with the analytical results.

Based on the analysis and the observations, we propose two techniques to improve the computational efficiency of ARF coding for MVC inter-view prediction. First, we propose *view-wise ARF adaptation*, which, based on the already observed RD-cost reduction, allows the encoder to determine whether to apply ARF for the remaining frames in a view over a certain time interval. Second, *filter updating based on depth-composition change* achieves further complexity reduction by allowing the same set of filters to be used by several consecutive frames (rather than updating them for every frame) until there is significant change in the depth-composition within the scene.

The remainder of this paper is organized as follows: The analysis of focus setting mismatch in a multiview system is provided in Section 2. The focus mismatch effect on the images captured by the corresponding cameras is also presented. In Section 3, we study ARF performed on inter-view prediction in MVC and introduce the proposed techniques for efficient ARF coding scheme. Simulation results are presented in Section 4. Finally, we conclude this work in Section 5.

## 2. MULTIVIEW SYSTEM WITH FOCUS MISMATCHES

In this section, we aim to analyze focus setting mismatches in a multiview system, and demonstrate how such mismatches will affect the captured images. We start with a review of the imaging model for a camera equipped with lens to derive the properties of the projected images under the influence of lens focusing effects. Then we consider multiple cameras which capture the scene from different viewpoints. With focus setting difference, the characteristics of the camera imaging systems will be different, leading to mismatches among images captured by different views. In particular, we consider the parameters of focal length and object positions to analytically model such mismatches. Our analysis will also relate the focus mismatches to the disparity exhibited in images from different views. The analytical results provide a better understanding of the focus mismatch problem and can be exploited to design coding tools aiming to compensate for such mismatches.

### 2.1. Analytical characteristics of images captured with a lens

A camera is typically modeled[*] as a system consisting of a film and a lens with focal length $f$ and aperture size $a$. For digital cameras, the film is made up with an array of image sensors. The plane which contains the film is called the "image plane", which is parallel to the lens with distance $d$. In Fig. 1, we construct a coordinate system with its origin located at the center of the lens and its $xy$-plane parallel to the image plane. The $z$-axis, which passes through the lens center, is also called the optical axis. The two points on the optical axis with $|z| = f$ are called the "focal points", which have special properties that will be discussed shortly.

We analyze the effects of light via geometrical optics, which treats light as rays.[4] Let us consider in Fig. 1, a point $P$ at location $(X, Y, Z)$ which is visible to the camera. Light rays passing through the lens center will not be refracted. Therefore, the light ray originates from $P$, passes through $O$, will be projected on the image plane at point $P'$ with coordinates $(X', Y', X')$ such that (based on the "congruence of triangles"):

[*] The material in this sub-section 2.1 serves as an introductory review. For more details refer to[3,4]

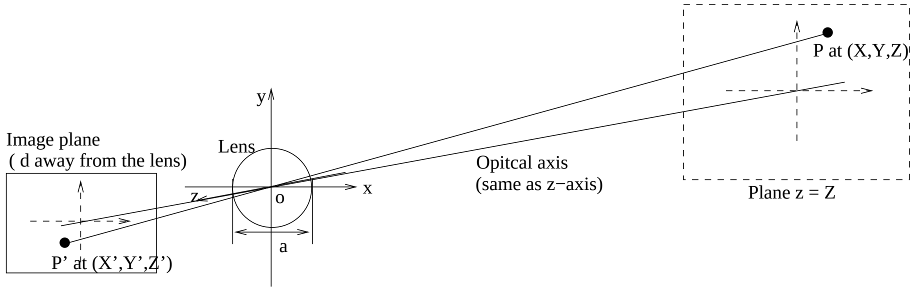


**Figure 1.** The model of a camera equipped with a lens

$$(X', Y', Z') = (\frac{d}{Z}X, \frac{d}{Z}Y, d) \tag{2}$$

The principle in (2), which describes the light rays passing through lens center, is called the "perspective projection". It is widely used in geometric camera models when the focusing effect of the lens is ignored.[5] However, in this work, it is our goal to analyze the effect of focus introduced by the lens. To take it into account, light passing through other parts of the lens have to be considered as well. According to geometrical optics, light rays parallel to the optical axis on one side of the lens will be refracted to pass through the "focal point" on the other side of the lens. Furthermore, light rays originate from a point $P$ with depth $Z$ (negative value for coordinate in Fig.1) will *converge* to a point $\hat{P}$ on the other side of the lens with distance $\hat{Z}$ that satisfies:

$$\frac{1}{|Z|} + \frac{1}{\hat{Z}} = \frac{1}{f} \tag{3}$$

These properties can help us determine the paths of light rays passing through other part of the length. Fig. 2(a) depicts three points $P, P_1, P_2$ at different depths, their corresponding projections $P', P'_1, P'_2$ of light rays passing through the lens center, and their converged image points $\hat{P}, \hat{P}_1, \hat{P}_2$. The dashed light paths are determined after finding the converging points based on the light paths depicted with solid lines. As a result, on the image plane with distance $d$ to the lens, a visible point will produce a point projection (perfectly focused) only if its depth $Z$ satisfies:

$$\frac{1}{|Z|} + \frac{1}{d} = \frac{1}{f} \ \Rightarrow \ |Z| = \frac{d \cdot f}{d - f} \tag{4}$$

When capturing the scene, we can focus on a specified distance by fine tuning $f$. We denote the perfectly focused distance $\frac{d \cdot f}{d-f}$ as $|Z^*|$. Fig. 2(b) shows the relationship between $f$ and $|Z^*|$ under different $d$. It can be seen that, operating close to $d$, a very small change in $f$ can cause significant difference in $|Z^*|$. For points with distances other than $|Z^*|$, their projections on the image plane will be uniform circles with diameter $\beta$, as depicted in Fig. 2(a). Using again the congruence of triangles, $\beta$ can be calculated as:

Depth closer than $|Z^*|$ (Fig. 2(a) $P_1$):

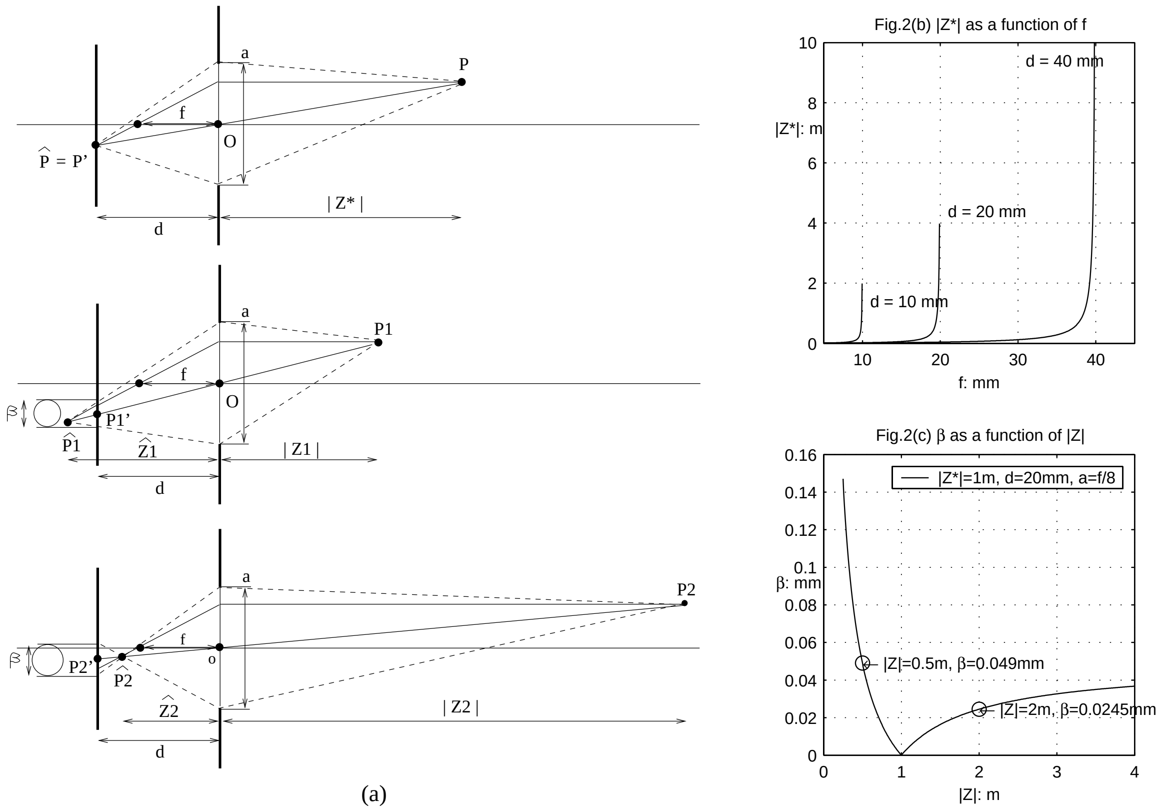
$$\begin{aligned} \frac{\beta}{a} &= \frac{\hat{Z}_1 - d}{\hat{Z}_1} \\ \beta &= \frac{af\,(|Z^*| - |Z_1|)}{|Z_1|\,(|Z^*| - f)} \end{aligned} \tag{5}$$

Depth farther than $|Z^*|$ (Fig. 2(a) $P_2$):

$$\begin{aligned} \frac{\beta}{a} &= \frac{d - \hat{Z}_2}{\hat{Z}_2} \\ \beta &= \frac{af\,(|Z_2| - |Z^*|)}{|Z_2|\,(|Z^*| - f)} \end{aligned} \tag{6}$$

Fig. 2(c) illustrates the variation of $\beta$ under the setting $|Z^*| = 1$ meter, $d = 20$mm (hence $f = \frac{1000*20}{1000+20} = 19.608$mm), and $a = f/8$. The diameter $\beta$ increases as the $|Z|$ moves away from $|Z^*|$. Instead of forming a *point image*, on the image plane, the image intensity is *spread over* a circle with diameter $\beta$ (area $\pi(\beta/2)^2$). Therefore, for a point with a depth $Z$, the *point spread function*, $PSF_Z$, is a circular disk as:

$$PSF_Z(x, y) = \begin{cases} 4/(\pi\beta^2), & \text{if } x^2 + y^2 \leq (\beta/2)^2 \\ 0, & \text{otherwise} \end{cases} \text{, where } \beta = \frac{af\,(|Z - Z^*|)}{|Z|\,(|Z^*| - f)} \tag{7}$$



(a)

**Figure 2.** (a)Projected images via a lens for points at different depths, (b)Variations of $|Z^*|$, (c)Variations of $\beta$

From Fig. 2(a), a visible point $P$ at $(X, Y, Z)$ will produce a circle-shaped projection centered at $P'$ on the image plane with coordinate $(X', Y')$ specified by (2). The intensity of the projection resulting from P, denoted as $I_P$, can then be described as:

$$I_P(x, y) = K_P \cdot PSF_Z(x - X', y - Y') \tag{8}$$

$K_P$ represents the light intensity at converging point $\hat{P}$, produced by $P$. On the image plane, this value is spread over the disk as modeled by (8). The total image intensity at $(x, y)$ on the image plane, denoted as $J(x, y)$, is the *superposition of all the projection circles centered at different locations that contribute non-zero values at position* $(x, y)$. In general, projections centered at near locations can have different diameters, as $\beta$ depends on the depth of the visible point that produces the projection. Within the scene being captured, for points in a visible region with very similar depth $Z$, their corresponding projections can be well approximated by the *same* point spread function (a fixed $Z$). Also, due to approximately fixed $Z$, $P - P' - \hat{P}$ has a one-to-one-to-one mapping. Note that, as depicted in Fig. 1, any point along the path $\vec{P'O}$ will correspond to the same $P'$, if we don't specify the depth $Z$. The converged light intensity of these points with similar depth $Z$, can be represented as a function of the location of P', i.e. a function of X' and Y': $K_P \to K_Z(X', Y')$. The subscript indicates that this representation is under a given $Z$. The projected image of this region can then be derived as:

$$\begin{aligned} J_Z(x, y) &= \int\int K_Z(X', Y') \cdot PSF_Z(x - X', y - Y') dX'dY' \\ &= K_Z(u, v) * PSF_Z(u, v), \text{ where } (u, v) \text{ are dummy variables} \end{aligned} \tag{9}$$

With the setting in Fig. 2(c), the image area of *a visible region at depth 0.5m* will be affected by a $PSF$ with $\beta = 0.049$mm, while the image corresponds to *a region at depth 2m* will be affected by another $PSF$ with $\beta =$ 0.0245mm.

### 2.2. Focus mismatches in multiview system

Now, we will discuss images captured by more than one camera, such as in multiview systems. One of the most common multiview settings uses a 1D horizontal camera arrangement: cameras are put along a horizontal line with equal spacing $b$ between each other, and their optical axes are parallel. Consider two neighboring cameras and denote them V1 and V2. We assume they have the same image plane distances: $d_{V1} = d_{V2} = d$, and their aperture settings are also identical: $a = f/8$. Since V2 is to the right of V1 with distance $b$, as compared with V1, we can regard the scene as shifted by $-b$ along the $x$-axis for the coordinate system centered at the lens of V2. From (2), due to the shift of $-b$, for a visible point with distance $Z$, the *center of its projection* will appear on the image plane of $V2$ with a disparity of $\delta_Z$ along the $x$-axis when we compare with the image of $V1$, :

$$P \text{ at } (X-b, Y, Z)_{V2} \rightarrow P'_{V2} \text{ at } \left(\frac{d}{Z}(X-b), \frac{d}{Z}Y\right) = \left(\frac{d}{Z}X + \delta_Z, \frac{d}{Z}Y\right) \text{ , where } \delta_Z = \frac{d}{Z}(-b) \tag{10}$$

Thus, for a visible region with depth $Z$, in V1 and V2, its *projection centers* have the following relationship:

$$K_{Z,V2}(x,y) = K_{Z,V1}(x-\delta_Z, y) \tag{11}$$

**If the two cameras are set with the same focal length** $f$, such that they have the same perfect-in-focus depth $Z^*_{V1} = Z^*_{V2} = Z^*$, then from (7), their PSF will be identical ($PSF_{Z,V1} = PSF_{Z,V2} = PSF_Z$). For a visible region with depth $Z$, the image intensity of V2, $J_{Z,V2}(x,y)$, can be related to the image of V1 as:

$$\begin{aligned} J_{Z,V2}(x,y) &= K_{Z,V2}(x,y) * PSF_Z(u,v) \\ &= K_{Z,V1}(x-\delta_Z,y) * PSF_Z(u,v) = J_{Z,V1}(x-\delta_Z,y) \end{aligned} \tag{12}$$

That is, the image captured by V2 is simply a shifted version of the image captured by V1. The disparities are different for image portions that corresponds to visible region at different depths. As in (10), regions with smaller depth $|Z|$ will produce larger disparity $\delta_Z$.

**However, if the focal lengths of the two cameras are not identical**, (hence they focus on different depths $Z^*_{V1} \neq Z^*_{V2}$), they will have different $PSF_Z$ due to the different $\beta$. For a visible region with depth $Z$, the corresponding images will be:

$$\begin{aligned} J_{Z,V1}(x,y) &= K_{Z,V1}(x,y) * PSF_{Z,V1}(x,y) \\ J_{Z,V2}(x,y) &= K_{Z,V1}(x-\delta_Z,y) * PSF_{Z,V2}(x,y) \end{aligned} \tag{13}$$

Given $f_{V1}$ and $f_{V2}$, we can determine the difference between $PSF_{Z,V1}$ and $PSF_{Z,V2}$. Consider the following example as in Fig. 3(a): Both cameras have $d = 20$mm and $a = f/8$, V1 is set with $|Z^*_{V1}| = 1$m (as in Section 2.1) but V2 is set with $|Z^*_{V2}| = 1.3$m. From the two $\beta$ curves, for regions at $|Z| = 0.5$m, V2 has a larger $\beta$ than V1; while for regions at $|Z| = 2$m, $\beta_{V2}$ is about half of $\beta_{V1}$.

To illustrate how such difference in $\beta$ will affect the images, we plot the frequency transform of $PSF_Z$, which is called the optical transfer function, $OTF$. That is, $Fr\{PSF_Z(x,y)\} = OTF(v_x, v_y)$, where $Fr\{\cdot\}$ denotes the transform from spatial to frequency domain. The following transform pair can be derived[6] by applying the Hankel transform to obtain a frequency domain representation in the polar coordinates system.

$$PSF_Z(r) = \begin{cases} 4/(\pi\beta^2), & \text{if } r^2 \leq (\beta/2)^2 \\ 0, & \text{otherwise} \end{cases} \quad \rightarrow \quad OTF_Z(q) = \frac{2J_1(\pi\beta q)}{\pi\beta q} \tag{14}$$

In (14), $r = \sqrt{(x^2+y^2)}$, $q = \sqrt{(v_x^2+v_y^2)}$ and $J_1$ is the Bessel function of the first kind of order 1.

Before going to the figure, let us discuss the frequency range we need to consider. For digital cameras, the light intensity is sampled by image sensors: Only frequencies up to the Nyquist rate have to be taken into account. For sensor type 1/2" (H×W = 6.4mm×4.8mm), a resolution of 640×480 pixels leads to 0.01mm sample-spacing between pixels. The Nyquist rate is 100/2=50(cycles/mm). In polar system, $q = \sqrt{50^2+50^2} \approx 70.71$. Thus, to plot $OTF_Z$ as (14), with $\beta$ in the unit of mm, we only need to consider the range up to $q$=70.71. This value corresponds to $\Omega = \pi$ in the digital domain. Fig. 3(b)(c) illustrates the OTF differences between V1 and V2. It can be seen that, within the main-lobe of $OTF_Z$, for image portions correspond to visible regions at $|Z| = 0.5$m, some lowpass has to be performed from V1 to V2. On the other hand, for visible regions at $|Z| = 2$m, the corresponding image portions in V1 need to be enhanced in order to match V2.

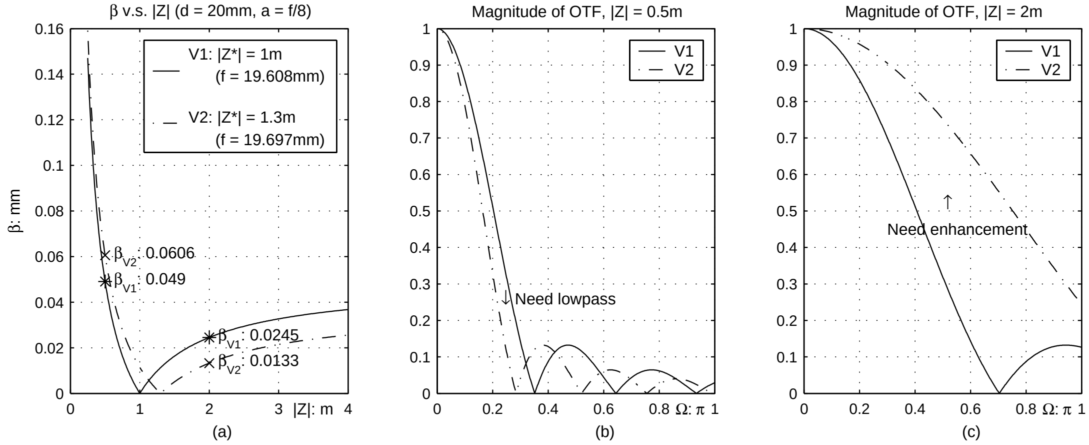


**Figure 3.** An example of depth dependent mismatches in frequency domain

The analytical results provide insight into the focus mismatch problem in multiview systems. In designing coding techniques to operate under these conditions we can take advantage of specific properties of blurring due to focus mismatch. First of all, analysis demonstrates that, in multiview systems, when the cameras have differences in focus settings, their images will exhibit *depth-dependent* mismatches. Knowing the depth-composition within the scene can then help us design compensation kernels optimized for each depth level. Secondly, for a 1D camera arrangement, from (13), the depth-dependent mismatch can also be *associated with the disparity*, which itself is a function of depth, as shown by (10). These results provide theoretical grounding for our previously proposed ARF approach:[1,2] when there is no measurement of depth available, the disparity vectors are exploited as estimation of scene depth to roughly partition an image into different depth levels. Thirdly, these mismatches correspond to differences in $OTF$ which can be represented as *blurring/sharpening filters* that are *circular symmetric* in spatial domain ($\beta$). In our ARF approach, for each depth level, a 2D MMSE filter with symmetrical constraint is estimated in order to compensate for the discrepancy in frequency responses illustrated in Fig. 3.

By combining the analytical results with characteristics of multiview systems, the complexity of two-pass ARF coding scheme can be reduced. Assuming that during the multiview video capturing process, the cameras being used, the spacing and the relative shooting orientations between cameras, and the focus settings, are time invariant (we will refer this as a “time-invariant multiview setting”). Then the characteristic of $OTF_Z$ of each view will also be time-invariant. For a pair of views with larger focus setting mismatch (larger difference in their $Z$-$\beta$ curves), the depth-dependent discrepancy will also be stronger, leading to lower coding efficiency in inter-view prediction and potentially higher coding gain if applying ARF. As for a view pair with no focus setting difference, theoretically the images will only be affected by disparity (as described by (12) ). The potential benefit of ARF would be limited since there is no depth-dependent mismatches to address. It is possible to predict whether ARF can achieve substantial gain for a given view by evaluating its coding performance at certain earlier timestamps, such that we can avoid applying ARF to views that would not achieve much coding gain. Furthermore, since the $OTF_Z$ are time-invariant, for a given pair of views, the mismatch in their images produced by an object at depth $|Z|$, will be the same at different capturing timestamps. Thus, across time, when the captured scene is composed of similar levels of depths, the types of mismatch exhibit in the images will also be alike, leading to similarity in the ARF filters. Instead of estimating ARF filters at every timestamp, a set of filters can be re-used during a certain time interval until there is significant change in depth-composition within the scene. With time-invariant camera spacing (fixed $b$ in (10) ), a given depth $Z$ will correspond to the same disparity $\delta_Z$ even for images captured at different timestamps. Thus to determine changes in depth-composition, we can compare the the distribution of block-wise DVs at different timestamps. A more efficient filter estimation/updating scheme can be developed by exploiting this property.

## 3. COMPUTATIONALLY EFFICIENT ARF FOR INTER-VIEW CODING IN MVC

For the multiview test sequences used by JVT-MVC group,[7] the "time-invariant multiview setting" assumption is being held true, i.e., there is no camera adjustment while the video is being captured. It has been reported, that these sequences exhibit inter-view mismatches that are strongly view dependent.[8] For a given pair of view, whether there exists focus mismatch between them, and the types of focus mismatch (blurriness/sharpness), are consistently observed over different timestamps.

Provided these observations and the analytical results from Section 2, in this section, we study the coding performance of ARF applied to inter-view prediction in MVC. Combining the finding with the analysis in Section 2, we propose techniques to design efficient ARF coding schemes which maintain coding efficient with much reduced complexity.

### 3.1. Rate-distortion analysis and view-wise adaption for ARF

In state-of-art video coding techniques, high coding efficiency is achieved by rate-distortion optimization. To analyze the performance of ARF, we record the frame-wise rate-distortion cost (RD-cost) in the initial disparity estimation (with only unfiltered reference) and in the final encoding (with unfiltered and multiple filtered references). The frame-wise RD-cost is calculated by aggregating macroblock (MB) RD-costs estimated during MB mode decision. We apply ARF to MVC inter-view coding with IPPPPPPP structure. For multiview data with frame rate 24fps, the inter-view coding is performed only at every 12th frame: 0, 12, 24......; for data with 30fps, inter-view coding is performed at every 15th frame: 0, 15, 30...... Thus these timestamps correspond to a half second interval, and we will call them "anchor timestamps" 0, 1, 2... etc. Fig. 4 provides some of the results when we compare the frame-wise RD-cost in the initial and final disparity estimation.

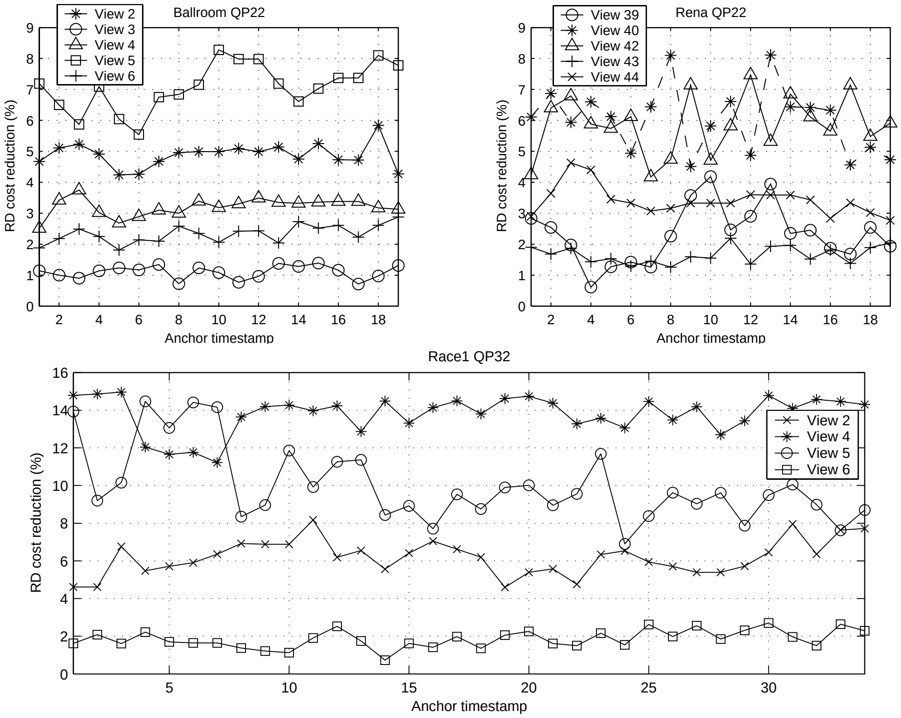


**Figure 4.** RD-cost reduction provided by ARF

According to the analytical results in Section 2, view pair with no focus setting difference will not produce depth-dependent mismatch which ARF is designed to compensate for. However, when there exists a focus setting difference, the type and degree of mismatch will depend on the depths of objects appearing in the scene. This will result in the variation of ARF performance[†]. The RD-cost reduction in Fig.4 shows behavior that is consistent

[†] Consider the setting as in the example of Fig. 3, a scene composed of more objects at depth around 2 meter will have stronger discrepancy for ARF to compensated, as compared to a scene with more objects at depth around 0.5 meter.

with the analytical results. It can be observed that, first, the RD-cost reduction achieved by our ARF approach varies significantly from view to view. These variations conform with the reported mismatches in multiview test sequences:[8] For views that exhibit strong focus mismatch with respect to the views used for prediction, for example Views 4 and 5 of Race1, applying ARF can provide more than 10% reduction in RD-cost. On the other hand, for a view with no perceivable focus mismatch as compared to its reference view, encoding using ARF leads to very limited improvement in coding efficiency. Second, views showing higher gains with ARF (exhibit focus mismatch) tend to have larger variations in RD-cost reductions across different timestamps, due to the change in depth-composition. Note that while Fig.4 only depicts results at a given QP for each sequence, the same behavior (variations cross view/time) is observed for all three sequences across QP 22, 27, 32, and 37. However it's worth mentioning that the RD-cost reductions become smaller as QP increases (low-bitrate scenario).

From the analytical results and observations above, if for a given view the RD-cost reduction achieved by using ARF is consistently very small over multiple anchor timestamps, it is reasonable to consider not applying ARF since the overall coding efficiency is still likely to be preserved. However, when observing small ARF coding gain, we have to consider the the potentially larger variation in RD-cost reduction for views indeed having focus mismatch. It may not be sufficient to apply ARF adaption for all the remaining frames based on a single observation. To be able to address this situation, we propose a predictive ARF adaptation method such that, within a period of $N$ anchor timestamps, encoder evaluates RD-cost reduction provided by ARF in the first $T$ anchor timestamps, and determines whether to apply ARF to the remaining anchor timestamps. In the next period of $N$ anchors, ARF will be tested *again* to determine whether it will be efficient. Let $\mu^V_{(1,T)}$ denote the average RD-cost reduction over anchor timestamps 1 to $T$ (where $T \geq 1$ ) in view $V$ when applying ARF, and $\sigma^V_{(1,T)}$ denote the corresponding standard deviation. This ARF adaptation method can be summarized as:

$$\text{For view } V\text{, apply ARF to anchor timestamp } i \text{ ? } (\ cN+T < i < c(N+1)\ )$$

$$\begin{cases} \text{NO,} & \text{if } \mu^V_{(cN+1,cN+T)} < \kappa \text{ and } \sigma^V_{(cN+1,cN+T)} < \epsilon \\ \text{YES,} & \text{otherwise} \end{cases} \tag{15}$$

We denote this as "View-wise ARF adaptation based on RD-cost prediction". In Section 4, simulation results will be provided with selected settings of $N$, $T$, $\kappa$, and $\epsilon$.

### 3.2. Correlation among ARF filters, and filter updating technique

From the analysis in Section 2, if under the condition of "time-invariant multiview setting", the types of mismatches exhibit in images from a given pair of view will depend on the depth-composition within the captured scene. A portion within the scene at depth $|Z|$ will produce the same type of mismatch at different capturing timestamps, leading to similarity in the corresponding estimated ARF filters. To further investigate the variation of filters, we perform correlation analysis: For a given view, we concatenate filter coefficients from filters estimated at anchor timestamp $t_1$ to form a coefficient vector $\mathbf{A}_{t_1}$, and compared it with the corresponding filter coefficient vector $\mathbf{A}_{t_i}$ at another anchor timestamp $t_i$. Fig. 5 provides results for some of the analyzed sequences.

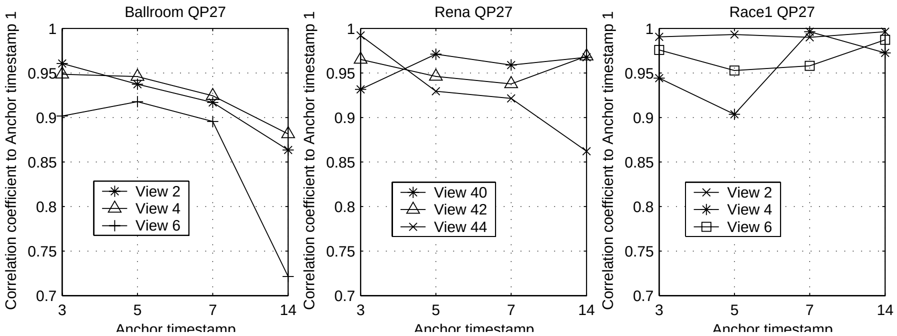


**Figure 5.** Correlation of estimated filter coefficients at different timestamps

For Ballroom, there are multiple dancing couples moving in the scene. A couple may appear in some frames at depth $Z_0$, while in the preceding frames there is nothing at this particular depth. Due to such depth-composition difference, the filter correlation has larger variation as compared to the other sequences tested. On the other hand, in Rena, the scene composition is consistent across time: A foreground girl and the background remain at same distances to the camera for the entire sequence. The estimated filter coefficients demonstrate very high correlation even over 14 anchor timestamps (about 200 frames). The most interesting case is Race1. In this sequence, the *content* in the scene is changing due to the viewing angle shift of the camera-set and the carts driving along the runway. However, frames at different timestamps mostly cover the same range of *depth*. (Refer to Fig.8 in the Appendix section.) That is to say, at different timestamps, there is no new "depth level" being introduced (as compared to the Ballroom case). As a result, filters still demonstrate very strong similarity. These results are consistent with the depth-dependency property derived in Section 2.

When the filters are highly correlated over certain time interval, e.g., when *depth-composition* within the scene remains similar, it is not necessary to estimate them at every single timestamp. Applying same set of filters over multiple timestamps will reduce the effort spent on initial disparity search and filter estimation, while the coding efficiency could be preserved. Moreover, when filters are re-used across time, we do not need to transmit filter coefficients. Our analysis suggests that the time intervals during which the filters are re-used and when to re-estimate/update filter coefficients, can be adaptively determined by comparing the *depth-composition* at different timestamps. In ARF, DVs are exploited as estimation of depth and are classified into classes using Gaussian-Mixture Model (GMM). To determine whether there has been a change in depth-composition, we compare the GMM classification results at different timestamps. Let $\mu_{V,i}^{GMM}(m)$ denote the mean of Gaussion component $m$ in the DV classification for frame $i$ in view $V$, and $P_{V,i}^{GMM}(m)$ be the corresponding percentage of blocks being classified into that class. A Gaussian component is defined as "not being covered", in a reference timestamp $r$, if its mean is at least $D$ pixels away from any Gaussian mean at timestamp $r$, $\mu_{V,r}^{GMM}(n)$. If the summation of $P_{V,i}^{GMM}(m)$ of all these "not covered" component is over a certain percentage $P$, we apply the two-pass ARF coding for the current frame to update filters, otherwise the filters estimated at the reference timestamp will be re-used.

$$W_{V,i}(m) = \begin{cases} 1, & \text{if } \forall n \ |\mu_{V,i}^{GMM}(m) - \mu_{V,r}^{GMM}(n)| > D \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$\text{If } \sum_{m} W_{V,i}(m) \cdot P_{V,i}^{GMM}(m) > P \rightarrow \text{apply two-pass ARF} \qquad \text{Otherwise} \rightarrow \text{re-use filters.} \tag{17}$$

However, there is an issue preventing us from directly using the above scheme: For the current frame being considered, at the point when making the decision to update filters or not, we actually do not have its disparity information yet. Disparity estimation should be performed only after we decided to apply two-pass ARF, otherwise filter will be re-used and initial estimation is skipped. To overcome this problem, we refer to a view in the earlier coding order for GMM disparity information. For example, when encoding View 2, the DV GMM for frames in View 1 are exploited to determine the "change in depth-composition" and then to decide updating filters or not. (Note that in this scheme, we cannot apply filter re-using to the first view being inter-view coded, as there will be no reference disparity.) This method would be most suitable for 1D parallel camera arrangement with equal spacing among cameras, as the disparity values of different views at same timestamp should be very similar in this scenario. We denote such method as "filter updating based on depth-composition change". Fig. 6 shows some filter re-using results when we set $D = 5$ pixels and $P = 15\%$ in (16)(17).

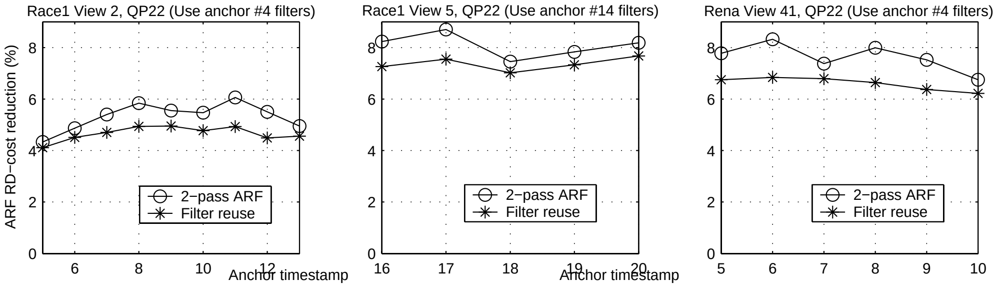


**Figure 6.** Examples of RD performance when filters are re-used over multiple timestamps

Combining with the "ARF adaptation" described in Section 3.1, an efficient ARF coding scheme can be summarized as the pseudo code provided below. In this new scheme, within a period of $N$ anchor timestamps, after evaluating ARF for the first $T$ anchor timestamps, there are three possible encoding options for the remaining frames. If encoder determines not to use ARF, they will be encoded normally, followed by a GMM classification on the DVs to provide disparity information for the next view. If it is decided to apply ARF, (16)(17) will be used to choose between simply re-using filters or performing the two-pass ARF to update filters. Frames encoded by re-using filters also need to undergo the classification on DV to generate disparity information for the next view. (But still we avoid the initial disparity search and filter estimation.)

```
                    Efficient ARF coding scheme for MVC inter-view prediction

loop over views $V$
{
    loop over $c$ (chunks of $N$ anchor timestamps)
    {
        for $(cN+1 \leq i \leq cN+T)$

            • Apply two-pass ARF
            (Initial disparity estimation, classification, filter estimation, encoding)

        • Calculate $\mu^V_{(cN+1,cN+T)}$ and $\sigma^V_{(cN+1,cN+T)}$

        if $(\mu^V_{(cN+1,cN+T)} < \kappa$ and $\sigma^V_{(1,i)} < \epsilon)$

            for $(cN+T < i < c(N+1))$
                • Conventional encoding (No ARF for the following frames)
                • GMM classification for disparity vectors

        else
        {
            • Filter reference timestamp $r$ = $T$

            for $(cN+T < i < c(N+1))$
            {
                • Calculate $W_{V-1,i}(m)$ as in (16)

                if $\sum_m W_{V-1,i}(m) \cdot P^{GMM}_{V-1,i}(m) > P$

                    • Apply two-pass ARF to update filters
                    • Filter reference timestamp $r$ = $i$

                else
                    • Re-use the filters at timestamp $r$
                    • Directly encode with filtered reference
                    • GMM classification for disparity vectors
            }
        }
    }
}
```

## 4. SIMULATION RESULTS

We conduct simulations with the proposed efficient ARF techniques. As described in Section 3.1, IPPPPPPP inter-view coding is performed with half-second time interval (anchor timestamps). We implemented the ARF coding scheme on top of the H.264/AVC framework using reference software JMVM 5.0. We set $N$ = 20 and $T$ = 4, i.e., for a 10 second period (20 anchor timestamps), anchor timestamps in the first 2 seconds will be encoded with ARF to evaluate the RD-cost reduction. To set the thresholds $\kappa$ and $\epsilon$, we observed ARF performance for sequences with very limited improvement, such as Exit and Uli. The achieved RD-cost reductions for these sequences are mostly within $0\% \sim 2\%$. Thus we set $\kappa = 2\%$ and $\epsilon = 1\%$: The encoder will disable ARF coding if it observes the average RD-cost reduction over the first 4 anchor frames is less than 2% with a variation less than 1%. For the remaining frames which require filtering, we again set set $D = 5$ pixels and $P$ =15% in (16) and (17) to determine whether filters will be re-used or updated (thus performing the entire two-pass ARF). The encoding results are provided in Fig. 7 (QP 22, 27, 32, 27) along with Table 1.

It can be seen from the results that the proposed techniques are very efficient in preserving ARF coding gains while the complexity is significantly reduced. After zooming in on the RD curves, we observe a degradation of less than $0.05dB$ as compared to the previously proposed ARF results. Across different QPs, the view-wise ARF adaptation method successfully identifies views with limited achievable coding gain if applied ARF. With higher QP, ARF is applied to fewer views since the achievable gains tend to be smaller under the low bitrate scenario. As for views that indeed utilize ARF, the filters are only updated once or twice over the timestamps tested (excluding the initial ARF testing period, which is the first 4 anchors for Ballroom and Rena, and a total of 8 anchors for Race1 since it has one period of 20 anchors and the remaining 15 anchors). All the other anchor frames are encoded using one-pass coding with filtered references constructed using already estimated filters.

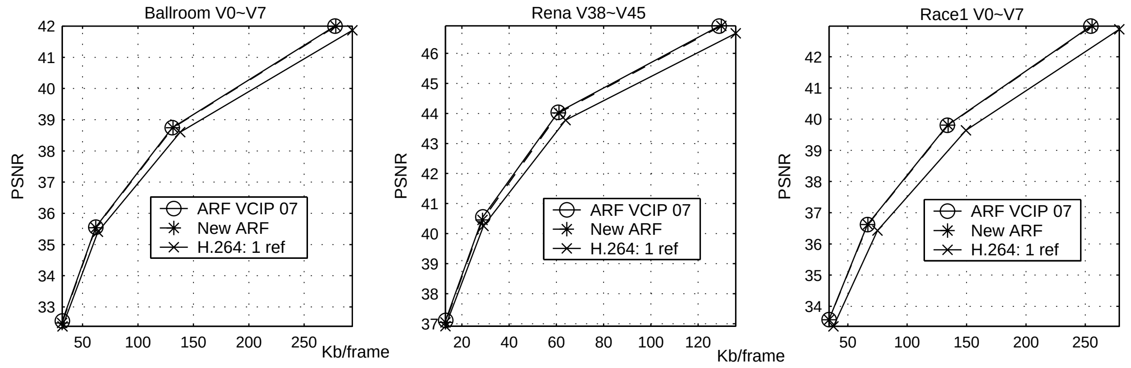


**Figure 7.** Encoding results of the proposed coding scheme

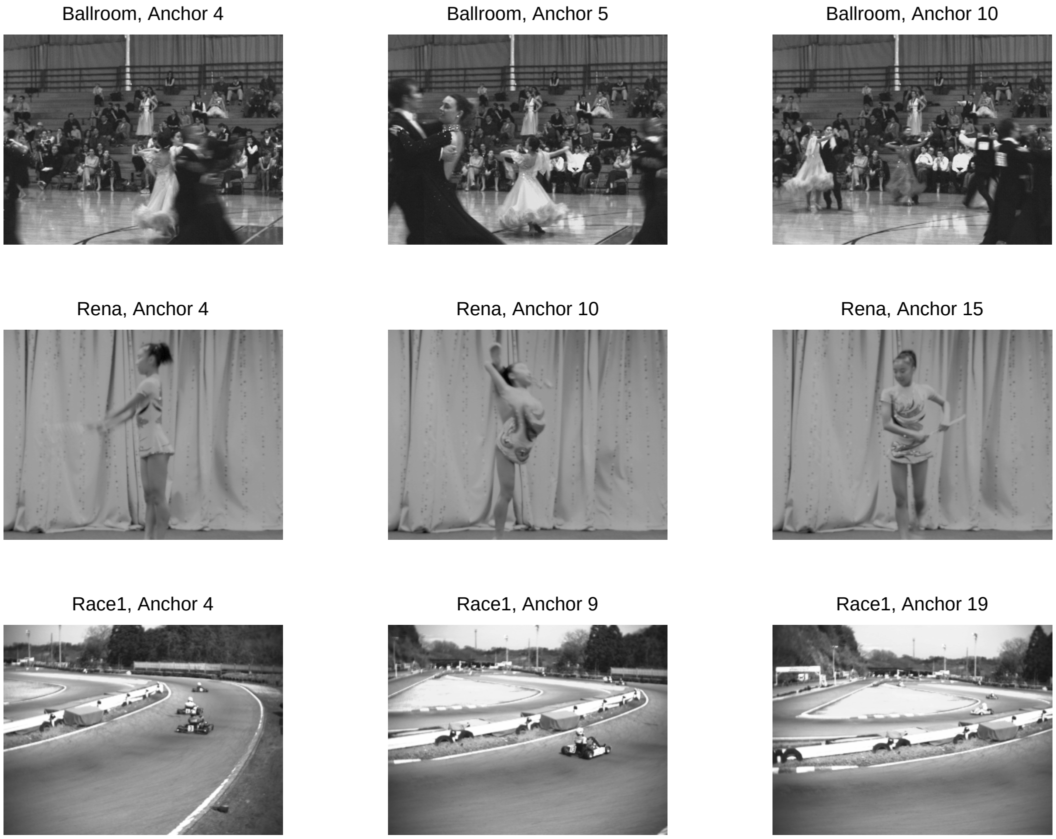
**Table 1.** Encoding selection of the proposed efficient ARF coding scheme

| | | **Ballroom(20 anchors)** | | | | | | | **Rena(20 anchors)** | | | | | | | **Race1(35 anchors)** | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **QP** | **Type** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | 39 | 40 | 41 | 42 | 43 | 44 | 45 | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 22 | 2-pass | 20 | 6 | 4 | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 35 | 9 | 9 | 9 | 9 | 8 | 9 |
| | Re-Use | | 14 | | 14 | 14 | 14 | | | 16 | 16 | 16 | | 16 | | | 26 | 26 | 26 | 26 | | 26 |
| | NoFilt | | | 16 | | | | 16 | 16 | | | | 16 | | 16 | | | | | | 27 | |
| 27 | 2-pass | 20 | 6 | 4 | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 35 | 8 | 8 | 9 | 9 | 8 | 8 |
| | Re-Use | | 14 | | 14 | 14 | 14 | | | 16 | 16 | 16 | 16 | 16 | | | 27 | 27 | 26 | 26 | | 27 |
| | NoFilt | | | 16 | | | | 16 | 16 | | | | | | 16 | | | | | | 27 | |
| 32 | 2-pass | 20 | 4 | 4 | 4 | 6 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 35 | 8 | 8 | 9 | 8 | 8 | 8 |
| | Re-Use | | 16 | | | 14 | | | | 16 | | 16 | | | | | 27 | 27 | 26 | 27 | | 27 |
| | NoFilt | | | 16 | 16 | | 16 | 16 | 16 | | 16 | | 16 | 16 | 16 | | | | | | 27 | |
| 37 | 2-pass | 20 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 35 | 9 | 9 | 9 | 8 | 8 | 8 |
| | Re-Use | | | | | | | | | 16 | | 16 | | | | | 27 | 26 | 26 | 27 | | 27 |
| | NoFilt | | 16 | 16 | 16 | 16 | 16 | 16 | 16 | | 16 | | 16 | 16 | 16 | | | | | | 27 | |

## 5. CONCLUSIONS

In this paper, we analyze focus mismatches in a multiview system, and propose techniques to efficiently apply ARF coding scheme to MVC inter-view prediction. With heterogeneous focus setting, we demonstrate that the differences among images captured by different cameras can be represented in terms of the focus setting mismatches (view-dependency) and the object depths (depth-dependency). Evaluations of the ARF coding performance verifies the analytical results. By exploiting the properties derived from the theoretical and performance analysis, we propose i) view-wise ARF adaptation based on RD-cost prediction to determine whether ARF is beneficial for a given view; and ii) filter updating based on depth-composition change, in which same set of filters will be re-used until the depth-composition within the scene has changed. Simulation results show that the new ARF scheme significantly reduces encoding complexity with less than 0.05 $dB$ degradation.

## APPENDIX A.



Ballroom, Anchor 4

Ballroom, Anchor 5

Ballroom, Anchor 10

Rena, Anchor 4

Rena, Anchor 10

Rena, Anchor 15

Race1, Anchor 4

Race1, Anchor 9

Race1, Anchor 19

**Figure 8.** Images at different anchor timestamps for the sequences tested

## REFERENCES

1. J. Kim, P. Lai, J. Lopez, A. Ortega, Y. Su, P. Yin, and C. Gomila, "New coding tools for illumination and focus mismatch compensation in multi-view video coding," *Submitted to IEEE Trans. Circuits Systems and Video Technologies (CSVT)* **17, no. 11**, pp. 1519–1535, Nov 2007.
2. P. Lai, Y. Su, P. Yin, C. Gomila, and A. Ortega, "Adaptive filtering for cross-view prediction in multi-view video coding," in *Proc. SPIE 2007 Visual Communications and Image Processing (VCIP)*, Jan 2007.
3. P. Mouroulis and J. Macdonald, *Geometrical Optics and Optical Design*, Oxford Series in Optical and Imaging Sciences, 1996.
4. H.-C. Lee, "Review of image-blur models in a photographic system using the principles of optics," *SPIE Optical engineering* **20, issue. 5**, pp. 405–421, May 1990.
5. D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2003.
6. R. N. Bracewell, *The Fourier Transform and Its Applications*, McGRAW-HILL, 3rd ed., 2000.
7. ISO/IEC-JTC1/SC29/WG11, "Call for proposals on multi-view video coding," *MPEG Document N7327* , Jul 2005.
8. Y.-S. Ho, K.-J. Oh, C. Lee, B. Choi, and J. H. Park, "Observations of multi-view test sequences," *JVT Document W084* , Apr 2007.