Histogram-Based Prefiltering for Luminance and Chrominance Compensation of Multiview Video

Ulrich Fecker, Marcus Barkowsky, and André Kaup, *Senior Member, IEEE*

***Abstract*—Significant advances have recently been made in the coding of video data recorded with multiple cameras. However, luminance and chrominance variations between the camera views may deteriorate the performance of multiview codecs and image-based rendering algorithms. A histogram matching algorithm can be applied to efficiently compensate for these differences in a prefiltering step. A mapping function is derived which adapts the cumulative histogram of a distorted sequence to the cumulative histogram of a reference sequence. If all camera views of a multiview sequence are adapted to a common reference using histogram matching, the spatial prediction across camera views is improved. The basic algorithm is extended in three ways: a time-constant calculation of the mapping function, RGB color conversion, and the use of global disparity compensation. The best coding results are achieved when time-constant histogram calculation and RGB color conversion are combined. In this case, the usage of histogram matching prior to multiview encoding leads to substantial gains in the coding efficiency of up to 0.7 dB for the luminance component and up to 1.9 dB for the chrominance components. This prefiltering step can be combined with block-based illumination compensation techniques that modify the coder and decoder themselves, especially with the approach implemented in the multiview reference software of the Joint Video Team (JVT). Additional coding gains up to 0.4 dB can be observed when both methods are combined.**

***Index Terms*—Image-based rendering, multiview video, video coding, video signal processing.**

I. Introduction

MULTIVIEW video is a technique where an object or a scene is recorded using a setup of several synchronous cameras from different positions (see, e. g., [1]). The resulting set of video sequences is called a *multiview video sequence*. Such a dataset can also be referred to as *dynamic light field*. In contrast to *static light fields*, where rigid and nonmoving objects are described by images from various positions, dynamic light fields make it possible to record moving objects or scenes.

The information contained in a multiview sequence can be used to visualize the object or scene from any desired viewpoint and with any viewing angle [2], [3]. Using *image-based rendering* algorithms, photorealistic views can be generated even for intermediate views, where no camera has been positioned during the recording process. Ideally, no geometric information is required, and even objects with complex geometry or difficult surface characteristics can be displayed without additional effort.

Frequently discussed applications for multiview video include *three-dimensional television (3D TV)* [4] as well as *free-viewpoint television (FTV)*, where the user is able to navigate freely through the scene [5], [6]. In addition, image-based rendering techniques can be applied in medicine, where 3-D visualizations of inner organs of the human can be created [7].

The recording of multiview video creates a large amount of data. Therefore, efficient compression techniques are required to store or transmit multiview video streams. A straightforward method for compression is *simulcast coding*, which means that each camera view is coded independently using an existing video codec, e. g., based on the H.264/AVC video coding standard [8]. It is, however, possible to increase the coding efficiency by exploiting not only the temporal correlation between subsequent frames, but also the spatial correlation between neighboring camera views [9]. This can be achieved by modifying the motion compensation step in such a way that references for the prediction are not only searched in temporally preceding or succeeding frames, but also in frames from other camera views. This means that both motion compensation and *disparity compensation* is performed. Based on these considerations, several coding schemes which extend the H.264/AVC coding standard for multiview video have recently been proposed (see, e.g., [10], [11]). A scheme based on hierarchical B pictures is currently being standardized by the Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG [12]–[14]. The corresponding reference software [Joint Multiview Video Model (JMVM)] is used for the coding simulations in this paper.

When multiview video data are recorded, significant variations are often observed between the luminance and chrominance components of the different views. These discrepancies mainly originate from calibration differences between the cameras, leading to global changes between the views. Global miscalibrations can also arise when heterogeneous cameras are used for recording multiview sequences. In addition to global changes, local illumination variations occur because the same object surface can appear differently from different angles [15].

Luminance and chrominance variations may impair the performance of a multiview coder or a renderer. Therefore, it is desirable to take them into account during the coding process. As part of the standardization efforts of the Joint Video Team, a block-based illumination compensation scheme has been proposed which improves the coding efficiency by predictive coding of the DC coefficient of the integer transform applied

Manuscript received May 8, 2007; revised September 7, 2007 and November 28, 2007. First published June 10, 2008; current version published October 8, 2008. This work was funded by the German Research Foundation (DFG) within the Collaborative Research Center "Model-based analysis and visualization of complex scenes and sensor data" under Grant SFB 603/TP C8. This paper was recommended by Associate Editor L, Onural.

The authors are with the Chair of Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg, 91058 Erlangen, Germany (e-mail: fecker@LNT.de; barkowsky@LNT.de; kaup@LNT.de).

Color versions of one or more figures in this paper can be found online at http//ieee.org/ieeexplore

Digital Object Identifier 10.1109/TCSVT.2008.926997

1051-8215/$25.00 © 2008 IEEE

in H.264/AVC [16]. This process is inversed at the decoder side, and, furthermore, the deblocking filter of H.264/AVC is modified to eliminate possible blocking artifacts caused by this approach [17].

In this paper, a different approach is used. In contrast to modifying the coder and decoder, a prefiltering step is introduced which harmonizes the luminance and chrominance components of the different camera views. Thus, the coder and decoder remain unmodified. The aim of this step is to achieve a higher coding efficiency when the adapted sequences are coded compared with the original ones. This prefiltering is also helpful for image-based rendering because differences in luminance and chrominance between the recorded camera views can lead to incorrect illumination and color reproduction after view interpolation. By using histogram-based prefiltering, the user's experience can be improved by freeing the rendering result from disturbing brightness and color variations. Therefore, it is assumed that the filtering does not need to be reversed on the decoder side.

Based on an idea outlined by Hekstra *et al.* in [18], histogram matching is proposed to adapt all camera views to a reference view in the center of the camera setup [19]. This method assumes that a good fit of a distorted image to a reference image may be obtained by adapting the cumulative histogram of the distorted image to the cumulative reference histogram. The advantage of this procedure is that no assumptions on the type of distortion like brightness or contrast variations are made and nonlinear operations may be considered. Since the same correction is applied to the entire image, histogram matching is especially useful to correct global discrepancies in the luminance and the chrominance. However, local illumination changes originating from surface orientation differences are preserved. Because one aim of image-based rendering is to visualize such complex surface characteristics, preserving them is advantageous for a realistic rendering result.

This paper begins by describing the basic histogram matching algorithm. It shows that the use of histogram-based prefiltering improves the prediction across camera views, while in contrast no coding gain is observed using the JMVM coder. Based on these results, different extensions to the basic algorithm are proposed for improving the coding performance. These extensions are then combined in order to achieve the best coding efficiency. Finally, results for different test sequences are presented and the coding performance is compared with that of the illumination compensation algorithm implemented in the JMVM coder. A combination of both algorithms is also discussed.

II. Histogram Matching Algorithm

A. Description of the Algorithm

The basic histogram matching algorithm can be used to adapt a distorted image to a reference image. The calculations are conducted in the YCbCr color space. Here, an example is shown for the luminance component Y, but the algorithm can be applied for both chrominance components Cb and Cr in an analogous manner.

The amplitude of the luminance signal of the reference image is denoted by $y_R[m,n]$. As a first step, the histogram of the reference image is calculated as follows:

$$h_R[v] = \frac{1}{w \cdot h} \sum_{m=0}^{h-1} \sum_{n=0}^{w-1} \delta[v, y_R[m,n]]$$
$$\text{with} \quad \delta[a,b] = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{else.} \end{cases} \tag{1}$$

Here, w denotes the width and h denotes the height of the image. Next, the cumulative histogram $c_R[v]$ of the reference image is created

$$c_R[v] = \sum_{i=0}^{v} h_R[i]. \tag{2}$$

The histogram $h_D[v]$ and the cumulative histogram $c_D[v]$ of the distorted image are calculated in the same manner. An example for a reference image and a distorted image together with their histograms and cumulative histograms is shown in Fig. 1. The distortion that was used on the distorted image included a gamma correction and a decrease in brightness. Therefore, its histogram is shrinked and shifted to the left.

Based on the cumulative histograms $c_R[v]$ and $c_D[v]$, a mapping function M is derived. The mapping is found by matching the number of occurrences in the distorted image to the number of occurrences in the reference image

$$M[v] = u \quad \text{with} \quad c_R[u] \leq c_D[v] < c_R[u+1]. \tag{3}$$

This process is illustrated in Fig. 2. The mapping is then applied to the distorted image $y_D[m,n]$, resulting in the corrected image $y_C[m,n]$

$$y_C[m,n] = M[y_D[m,n]]. \tag{4}$$

B. Correction of the First and Last Active Bin

When the described algorithm is applied, all values of the distorted image below a certain threshold are clipped and remapped to the first active bin in the histogram. This induces a brightness offset in the corrected image for the dark values in the luminance image. In order to avoid this effect, the first active bin value is modified. The following additional step is only applied to the luminance component, because the color components generally do not suffer from clipping artifacts.

The algorithm described so far implies that, in the clipped interval, the highest value of the reference image is used for the values in the corrected image. The quality of the mapping can be improved by using the center of mass of the values for this interval in the reference image. The interval is $[0 \ldots M[0]]$ and the center of mass is calculated as:

$$s_l = \frac{\sum_{i=0}^{M[0]} i \cdot h_R[i]}{\sum_{i=0}^{M[0]} h_R[i]}. \tag{5}$$

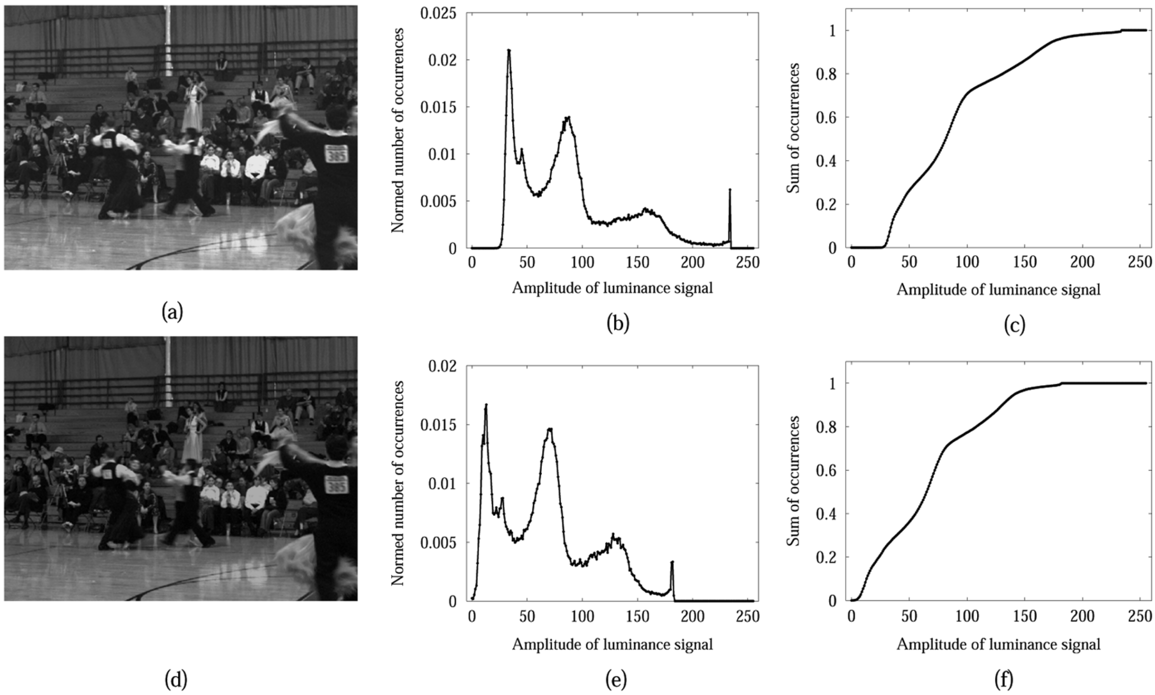


Fig. 1. Example histograms of the luminance component for a reference image (sequence "Ballroom") and a distorted image. (a) Reference image. (b) Histogram of the reference image. (c) Cumulative histogram of the reference image. (d) Distorted image. (e) Histogram of the distorted image. (f) Cumulative histogram of the distorted image.

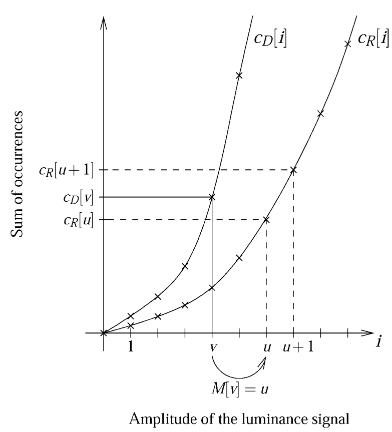


Fig. 2. Details of the mapping algorithm shown in a section of the cumulative histogram.

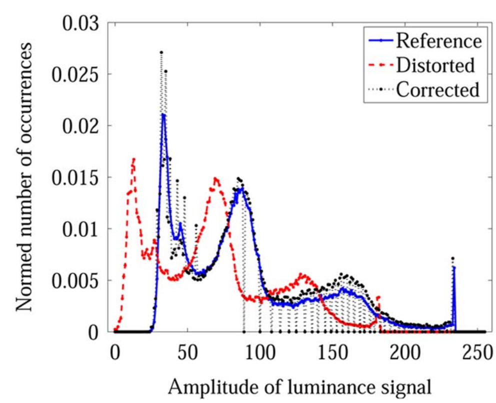


Fig. 3. Example for the histogram of the corrected image compared with the reference image and the distorted image (sequence "Ballroom").

This value is then applied in the mapping

$$M[0] = s_l. \tag{6}$$

To avoid the same effect when clipping occurs at the upper boundary, the last active bin is modified as well. The upper interval is $[M[254] + 1 \ldots 255]$, and the center of mass is calculated and applied in the mapping as follows:

$$\begin{aligned} s_u &= \frac{\sum_{i=M[254]+1}^{255} i \cdot h_R[i]}{\sum_{i=M[254]+1}^{255} h_R[i]} \\ M[255] &= s_u. \end{aligned} \tag{7}$$

An example for the resulting histogram of the corrected image with this mapping applied is shown in Fig. 3. The histogram of the corrected image is closer to the reference histogram than to the histogram of the distorted image. In this example, the histogram is stretched to fit the reference histogram when the algorithm is applied. That is why there are luminance levels which do not occur in the corrected image. This effect leads to sharp drops to zero in the histogram, but it does not impair the visual impression of the corrected image.

C. Application to Multiview Sequences

The described algorithm can be applied to multiview sequences in the following way. One camera view, which is close to the center of the camera setup, is chosen as the reference

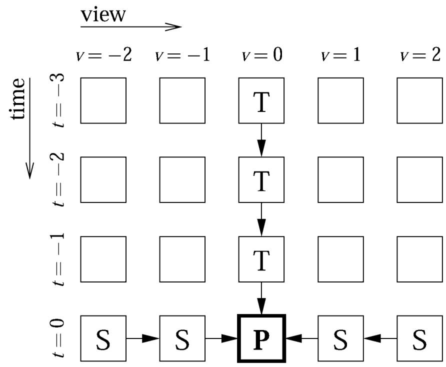


Fig. 4. Prediction scheme assumed for the statistical analysis.

view. All other camera views are corrected so that their histograms fit the histogram of the chosen reference view. This is done frame by frame for the whole sequence, which means that each time step is corrected individually.

III. Effect of Histogram Matching on Multiview Coding

A. Statistical Evaluation of the Prediction Step

The method described in [9] was used to evaluate the effect of histogram matching on multiview video coding independent from a specific coding scheme. It was assumed that an encoder would hold frames from a certain number of time steps for all camera views in its memory (see Fig. 4). The current frame to be predicted is marked as "P"-frame. For each block in the frame, block matching is performed to find the reference block with the minimum mean square error (MSE) compared with the current block. For that, all temporally preceding frames ("T"-frames) as well as frames from the same time step but from different camera views ("S"-frames) are searched. The frame delivering the minimum MSE is then chosen as the best reference for the current block.

The number of temporal (T) and spatial (S) references is counted for all frames in all views of the multiview sequence and converted into percentages. These probabilities provide an estimate for the number of blocks which benefit from prediction across the different camera views. The frames marked by empty rectangles in Fig. 4 are not searched because it could be shown that their individual probabilities are rather small [9].

For the evaluation, several multiview test sequences were used. The described block-matching scheme was applied to the original sequences as well as to the sequences after histogram matching had been performed. The results are shown in Table I.

The results show that the percentage of spatial prediction increases when the sequences have been compensated using histogram matching. The gain in spatial prediction varies between 1.6% to 12.9%, depending on the characteristics of the specific sequence. The results indicate that disparity-compensated prediction across the camera views is improved by the harmonization achieved by histogram matching.

B. Effect on the JMVM Multiview Video Coder

Here, the effect of the described algorithm on the performance of multiview video coding is analyzed. Several test sequences were compensated using histogram matching and then coded using the JMVM reference software (version 2.4). A GOP length of 12 was used and the sequences cut after a length of 180 frames (96 frames for "Breakdancers") to limit the simulation time. Four BasisQP values were tested (22, 27, 32, and 37), as specified in [20]. The average PSNR differences were then calculated according to the method suggested by Bjøntegaard in [21]. The difference between the encoder input and the decoder output was used to calculate PSNR values. This means that these values are derived based on the original sequence when histogram matching is not used, but when histogram matching is used they are derived based on the compensated sequence.

In Fig. 5, the coding performance with and without histogram matching is shown for a typical test sequence ("Ballroom"). From the plot, it can be seen that the coding performance is not improved when the basic histogram matching algorithm is applied. Instead, it deteriorates slightly. The result contradicts the statistical evaluation which showed that the spatial prediction across camera views was improved by using luminance and chrominance compensation. This can be explained by the fact that temporal prediction still plays a very important role for the coding efficiency. Because the histogram matching algorithm is applied individually for each time step, variations may be introduced between subsequent frames in the temporal direction. Therefore, the temporal prediction may be less efficient due to these variations, which leads to a small overall coding loss.

IV. Extensions to the Algorithm

In order to improve the coding performance, different modifications and extensions are introduced to the histogram matching algorithm.

A. Time-Constant Mapping Function

The original algorithm was applied to the whole sequence on a frame-by-frame basis. This meant that, for each time step and camera view, a separate histogram was calculated, leading to a separate mapping function. These mapping functions were then used to correct the different time steps independently from each other. This leads to a good fit of the different camera views at each particular point in time. However, it can also lead to variations between subsequent time steps, which may deteriorate the coding performance (as seen in the previous section) and affect the visual quality.

Therefore, a modified algorithm is introduced in which the histograms are summed up over all time steps for both the reference view and that which shall be corrected [22]. To achieve this, the calculation of the histogram in (1) is modified as follows:

$$h_R[v] = \frac{1}{\ell \cdot w \cdot h} \sum_{t=0}^{\ell-1} \sum_{m=0}^{h-1} \sum_{n=0}^{w-1} \delta[v, y_R[m,n,t]]$$
$$\text{with} \quad \delta[a,b] = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{else.} \end{cases} \tag{8}$$

Here, $y_R[m,n,t]$ denotes the amplitude of the luminance image at time step t of the reference view. The length of the sequence is

TABLE I
RESULTS OF THE STATISTICAL ANALYSIS IF HISTOGRAM MATCHING IS USED COMPARED WITH THE RESULTS WITHOUT HISTOGRAM MATCHING

Sequence	Number of views	Number of frames	*Without Histogram Matching* Temporal	Spatial	*With Histogram Matching* Temporal	Spatial	Increase in Spatial Percentage
Crowd	5	1 002	86.20 %	13.80 %	84.61 %	15.39 %	1.59 %
Flamenco1	8	624	77.43 %	22.57 %	72.42 %	27.58 %	5.01 %
Ballet	8	100	87.20 %	12.80 %	80.29 %	19.71 %	6.91 %
Breakdancers	8	100	63.70 %	36.30 %	50.76 %	49.24 %	12.94 %
Ballroom	8	250	84.91 %	15.09 %	83.26 %	16.74 %	1.65 %
Exit	8	250	87.26 %	12.74 %	83.96 %	16.04 %	3.30 %
Jungle	8	250	96.13 %	3.87 %	93.66 %	6.34 %	2.47 %
Uli	8	250	91.67 %	8.33 %	85.49 %	14.51 %	6.18 %

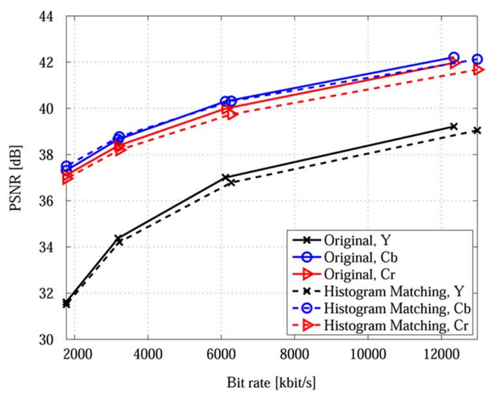


Fig. 5. Coding performance using the basic histogram matching algorithm ("Ballroom", eight views). Average PSNR difference: -0.26 dB (Y), 0.00 dB (Cb), and -0.28 dB (Cr).

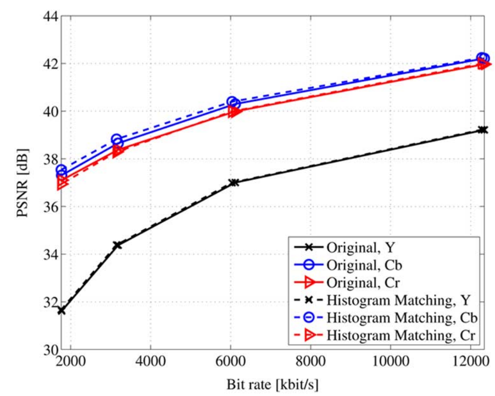


Fig. 6. Coding performance using time-constant histogram matching ("Ballroom," eight views). Average PSNR difference: 0.05 dB (Y), 0.15 dB (Cb), and -0.02 dB (Cr).

denoted by ℓ. If the histogram is calculated based on only a part of the sequence, ℓ may also denote the length of this interval.

The cumulative histogram is calculated according to (2), and, from that, a single mapping function is derived according to (3), which is then applied to all frames. Using this method, variations between subsequent time steps can be avoided.

The algorithm is still applied separately for each camera view of a multiview video sequence with the exception of the reference view, which serves as a correction basis for all other views. Therefore, if N is the number of camera views, $N-1$ mapping functions are derived.

Fig. 6 shows the coding performance when histogram matching with a time-constant mapping function is applied. The plot has also been generated using the "Ballroom" sequence and the JMVM reference coder. Compared with Fig. 5, the coding efficiency has improved and is now slightly better than the coding efficiency without luminance and chrominance compensation.

B. RGB Color Space

The algorithm described so far operates in the YCbCr color space, correcting the Y, Cb, and Cr components individually. This choice is based on the fact that common video codecs operate in this color space. In addition, the test sequences used are for the most part stored in the YCbCr color space with a color subsampling according to 4:2:0. However, the most common cameras that could be used in recording the sequences will operate in the RGB color space. Therefore, the correction algorithm should be modified to work in this color space also.

If the video data are stored as YCbCr sequences, a conversion to RGB needs to be done. As the resolution of the Cb and Cr components is reduced by a factor of 2 horizontally and vertically, their original resolution is restored by bilinear interpolation. After the conversion, the R, G, and B components are processed separately using the histogram matching algorithm. Then, the corrected sequences are converted back to the YCbCr color space to be passed to the multiview encoder.

Although the data are converted from YCbCr to RGB and back in floating-point arithmetic, there is a possibility of introducing a quantization error by using the histogram matching algorithm. However, this effect can be shown to be rather small and hardly affects the quality of the video sequences.

The coding performance for histogram matching in the RGB color space is depicted in Fig. 7. The performance is improved compared with the efficiency of the original algorithm in Fig. 5, especially for the chrominance components.

C. Global Disparity Compensation

Since the distorted sequence and the reference sequence originate from cameras in different positions, they do not show exactly the same content. Instead, there is a certain displacement between the two camera views. It is therefore desirable to calculate the histograms based only on the overlapping area. To achieve this, a phase correlation algorithm is used to determine the global disparity between the two sequences before the histogram calculation is performed.

The phase correlation method computes the cross-correlation, which is efficiently implemented using the discrete Fourier

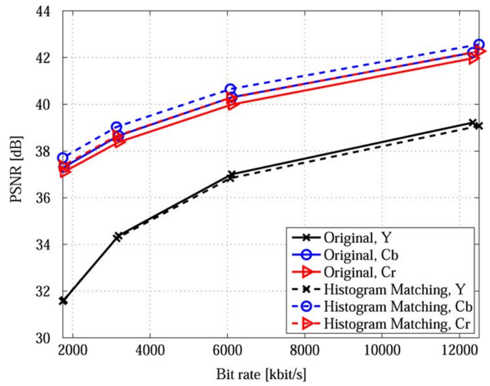


Fig. 7. Coding performance using histogram matching in the RGB color space ("Ballroom," eight views). Average PSNR difference: -0.09 dB (Y), 0.39 dB (Cb), and 0.30 dB (Cr).

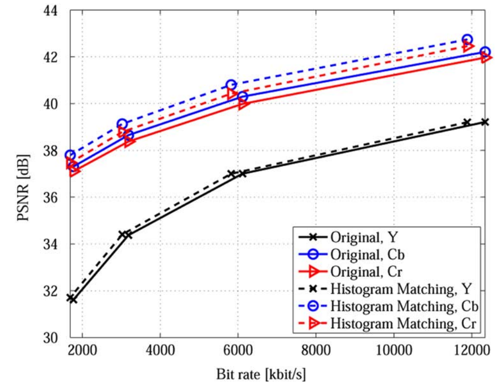


Fig. 8. Coding performance using time-constant histogram matching in the RGB color space ("Ballroom," eight views). Average PSNR difference: 0.21 dB (Y), 0.62 dB (Cb), and 0.55 dB (Cr).

transform (DFT). Only the phase information of the spectrum is considered. After an inverse transformation, the highest peak indicates the spatial offset of the two images [23].

However, applying global disparity compensation did not improve the coding efficiency compared to the original algorithm for the test sequence "Ballroom." For this sequence, and for many other available test sequences, the disparity between the camera views is rather small, indicating that the nonoverlapping area is also very small. Since global disparity compensation hardly has any effect on these sequences, it is not applied in the remainder of this paper. However, it may be useful for sequences with larger disparities where the nonoverlapping area is larger.

D. Time-Constant Histogram Matching in the RGB Color Space

All of the extensions presented here can be used individually. However, the coding performance can be further improved when the extensions are combined. As global disparity compensation has no effect for the sequences with rather small disparities, a combination of only the time-constant histogram matching and RGB color conversion is used.

The resulting coding efficiency is shown in Fig. 8. This combination clearly outperforms all other variations of the algorithm and leads to a significant improvement compared with multiview coding without prefiltering. It is therefore proposed to use time-constant histogram matching with RGB color conversion prior to multiview video coding, such as the process illustrated in Fig. 9. Please note that in a practical setup for capturing and coding, no additional color conversion is necessary for the proposed algorithm since the conversion and downsampling steps are performed anyway when the data is passed from RGB cameras to the video coder, which works on YCbCr 4:2:0 data.

E. Effect on the Visual Quality

Increasing the coding performance by changing the sequences is only useful when the visual quality of the sequences is not degraded [24]–[27]. Based on a subjective quality evaluation of the various views before and after histogram matching, no such degradation is found in this study. For some sequences (e.g., "Ballroom"), the subjective quality remains unchanged and no noticeable differences can be observed. For other sequences, the visual impression is improved. The original sequences "Crowd" and "Race1," for example, contain rather obvious variations between the color of identical objects in neighbouring views, which impairs the visual impression. After prefiltering, the color is harmonized between the views, which is clearly advantegous not only for the subjective quality of the multiview sequence but also for image-based rendering applications.

F. Reversibility of the Algorithm

Depending on the application, luminance and chrominance compensation can also be helpful for the renderer and therefore does not need to be reversed. In this case, no additional data needs to be transmitted. However, if necessary the compensation can be approximately reversed by applying the inverse of the mapping function (Fig. 10) to the decoded data. The amount of additional data in this case is limited, especially for time-constant histogram matching, since only one mapping function per corrected view is involved for the whole sequence.

V. Coding Results

In the previous section, it was shown that time-constant histogram matching with RGB color conversion provides the best possibilities for extending the basic histogram matching algorithm. Coding performance results based on these extensions are shown in Figs. 11–13 (in addition to Fig. 8).

As can be seen from the plots, the coding performance is improved for both the luminance as well as the chrominance components in most cases. The PSNR of the Y component is typically about 0.2–0.7 dB higher using histogram matching. For the Cb and Cr components, even larger gains of up to 1.9 dB can be observed. For the "Breakdancers" sequence, the PSNR of the luminance component is slightly decreased, but the PSNR of both chrominance components is improved.

VI. Comparison to Illumination Compensation in JMVM

Lee *et al.* [16] propose a different approach for illumination compensation, which has been implemented in the JMVM soft-

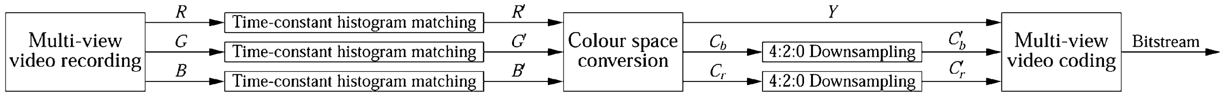


Fig. 9. Block diagram of the proposed algorithm. Time-constant histogram matching is applied in the RGB color space before color conversion, chrominance downsampling, and multiview coding.

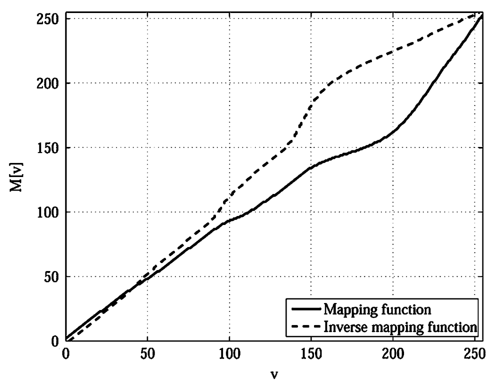


Fig. 10. Example for a mapping function in the RGB color space and its inverse (Race1, view 0, R component).

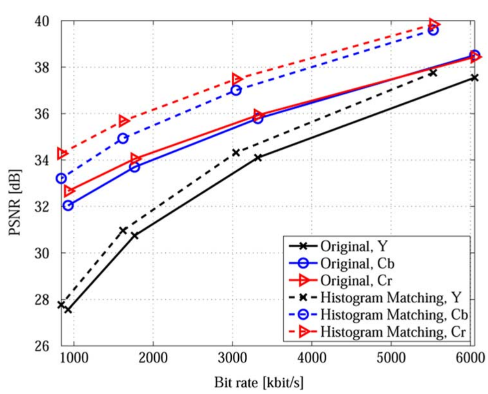


Fig. 11. Coding performance using time-constant histogram matching in the RGB color space ("Crowd," five views). Average PSNR difference: 0.68 dB (Y), 1.50 dB (Cb), and 1.84 dB (Cr).

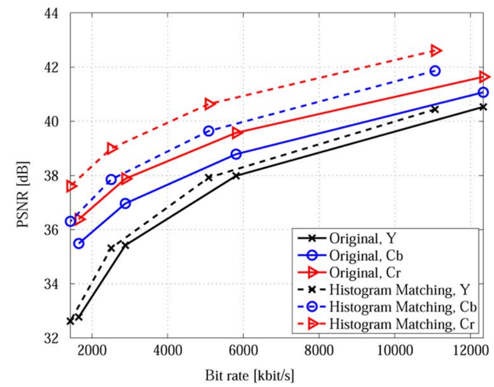


Fig. 12. Coding performance using time-constant histogram matching in the RGB color space ("Race1," eight views). Average PSNR difference: 0.42 dB (Y), 1.22 dB (Cb), and 1.41 dB (Cr).

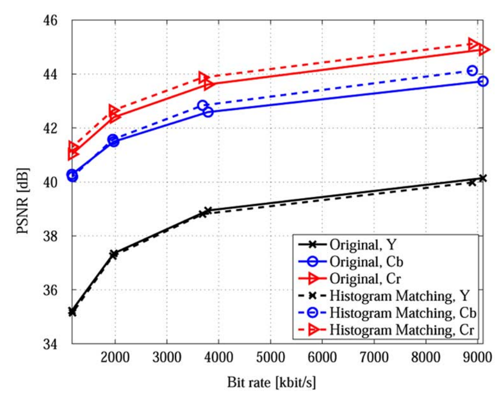


Fig. 13. Coding performance using time-constant histogram matching in the RGB color space ("Breakdancers," eight views). Average PSNR difference: −0.10 dB (Y), 0.22 dB (Cb), and 0.28 dB (Cr).

ware (denoted as "JMVM-IC" in the future). The idea is to modify the coding of the DC coefficient in the transform domain. When Inter prediction across different camera views is used, the DC coefficient corresponds to the illumination difference between the currently coded macroblock and the reference macroblock from the reference view. If there is an illumination offset between the reference image and the currently coded image, the DC coefficients of neighbouring macroblocks will be highly correlated.

Therefore, a predictor is introduced which calculates an estimate for the DC coefficient of the current macroblock from those of the surrounding macroblocks. By using this method, the number of bits needed to code the DC coefficient can be reduced. Resulting PSNR gains of 0.1–0.7 dB are reported in [16]. Since this process modifies the coding process, it has to be reversed in the decoder.

One advantage of JMVM-IC is that both global and local illumination changes can be considered. The method also does not change the output sequence, which is useful if the occurring illumination changes need to be preserved for the desired application. The advantage of histogram matching, on the other hand, is that it can compensate for all kinds of camera calibration errors which do not necessarily lead to a pure DC offset. Since it is used as prefiltering, the changes to the multiview sequence remain at the decoder output. Although this will be beneficial for rendering in most cases, it could sometimes be undesirable. In that case, an approximate inversion can be achieved using the inverse of the mapping function (see Section IV-F).

In the following, we will compare the gain in coding efficiency achieved by histogram matching to JMVM-IC. Since histogram matching is used as a prefiltering step—in contrast to JMVM-IC, which modifies the coding process—both methods can be combined. The resulting PSNR curves are shown in Figs. 14–17 for the same test sequences used before. For an easier comparison, the average PSNR gains are additionally summarized in Table II. Based on the luminance component (Y), both methods are able to achieve a coding gain in most of

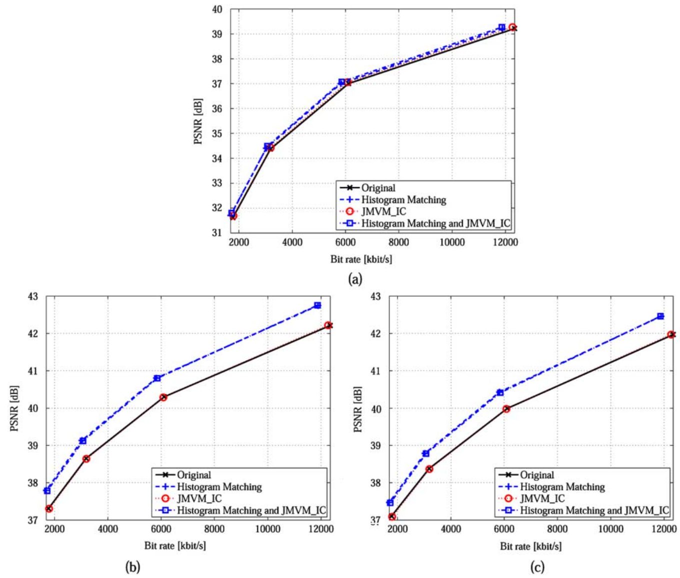


Fig. 14. Coding performance using time-constant histogram matching with RGB color conversion compared with illumination compensation in JMVM ("Ballroom," eight views). (a) Y component. (b) Cb component. (c) Cr component.

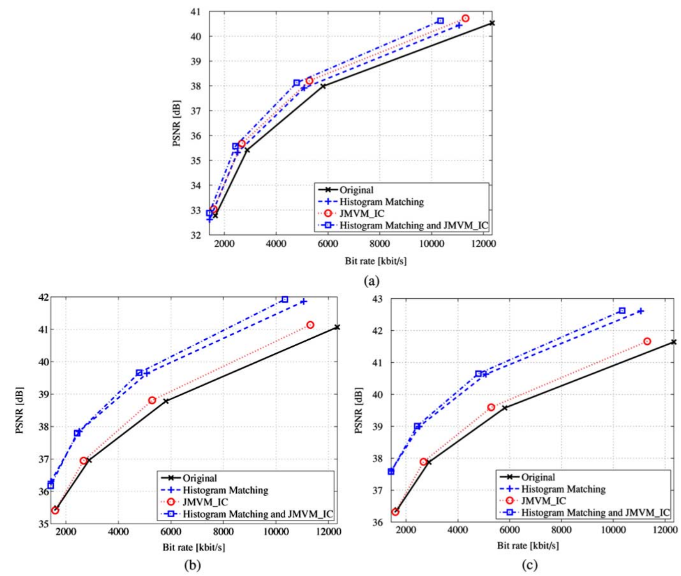


Fig. 16. Coding performance using time-constant histogram matching with RGB color conversion compared to illumination compensation in JMVM ("Race1", 8 views). (a) Y component, (b) Cb component, (c) Cr component.

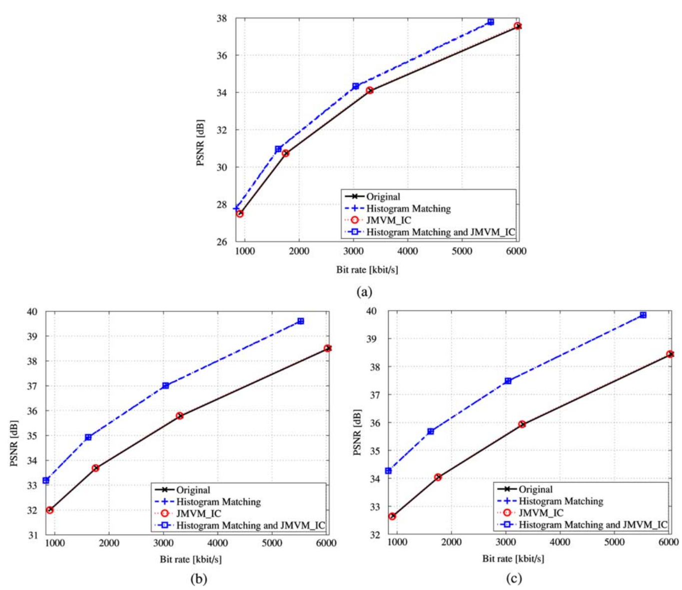


Fig. 15. Coding performance using time-constant histogram matching with RGB color conversion compared to illumination compensation in JMVM ("Crowd", 5 views). (a) Y component, (b) Cb component, (c) Cr component.

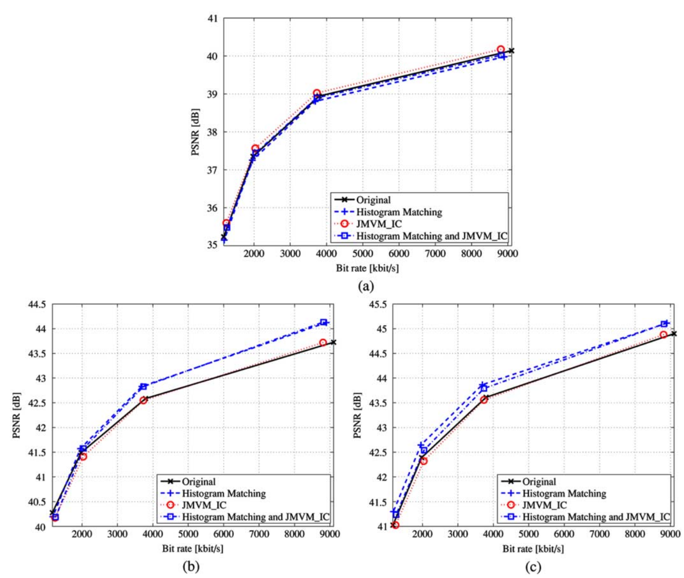


Fig. 17. Coding performance using time-constant histogram matching with RGB color conversion compared to illumination compensation in JMVM ("Breakdancers", 8 views). (a) Y component, (b) Cb component, (c) Cr component.

the sequences. It is only in the sequence "Breakdancers" that histogram matching does not improve the coding result for the Y component. For all other sequences, the combination of both methods shows the best performance.

Regarding the chrominance components (Cb and Cr), JMVM-IC hardly has any effect on the coding performance, except for the "Race1" sequence. Histogram matching, in contrast, significantly improves the performance for all four test sequences.

To summarize, the best performance in most cases is achieved when both methods are combined. In this case, histogram matching as well as JMVM-IC contribute to the coding gain in the luminance component, while the coding gain in the chrominance components is mainly achieved by histogram matching.

VII. Summary and Conclusion

Prefiltering based on a histogram matching algorithm has been proposed to compensate the luminance and chrominance variations between the different camera views of multiview video sequences. A basic algorithm has been presented, and through statistical analysis it was shown that it is able to improve the spatial prediction across the different camera views.

TABLE II
AVERAGE PSNR GAINS OF HISTOGRAM MATCHING, JMVM-IC AS WELL AS A COMBINATION OF HISTOGRAM MATCHING AND JMVM-IC. THE PSNR DIFFERENCES ARE ALWAYS CALCULATED COMPARED TO MULTIVIEW CODING WITHOUT LUMINANCE OR CHROMINANCE COMPENSATION

	Histogram Matching			JMVM-IC			Histogram Matching and JMVM-IC		
Sequence	**Y**	**Cb**	**Cr**	**Y**	**Cb**	**Cr**	**Y**	**Cb**	**Cr**
Ballroom	0.21 dB	0.62 dB	0.55 dB	0.06 dB	−0.01 dB	−0.01 dB	0.24 dB	0.59 dB	0.52 dB
Crowd	0.68 dB	1.50 dB	1.84 dB	0.04 dB	0.02 dB	0.02 dB	0.71 dB	1.51 dB	1.85 dB
Race1	0.42 dB	1.22 dB	1.41 dB	0.54 dB	0.24 dB	0.22 dB	0.82 dB	1.35 dB	1.55 dB
Breakdancers	−0.10 dB	0.22 dB	0.28 dB	0.10 dB	−0.06 dB	−0.06 dB	−0.04 dB	0.18 dB	0.16 dB

To achieve a coding gain, possible extensions to the algorithm were introduced, and it was shown that time-constant histogram matching with RGB color conversion is the best option. Coding tests using the JMVM reference coder were performed for different test sequences. For most of the tested sequences, the coding performance could be significantly increased by up to 0.7 dB for the luminance component and by up to 1.9 dB for the chrominance components.

Histogram-based prefiltering was also combined with the illumination compensation method implemented in the JMVM software. It was shown that the coding performance is improved even further when this combination of both algorithms is used.

If the whole sequence is not available for filtering (e.g., during a real-time transmission), the time-constant histogram matching algorithm will need to be extended so that it can be applied individually on small parts of the sequence. Future studies should focus on different approaches for achieving this. One possibility is a sliding-window approach, where the mapping for each time step is found using a window of a certain size around the time step. It might also be beneficial to detect scene changes and restart the histogram calculation after a scene change has occurred. If the video data is recorded using a fixed camera setup, mapping functions for each camera could also be generated in advance during a calibration step and then applied in real time during the recording and transmission of the multiview video.

Currently, the view in the center of the camera setup is manually chosen as the reference view. However, if this view itself is distorted, it is better to choose another reference. In the future, a more sophisticated selection of the reference view should be carried out prior to histogram matching. For example, a method for identifying and excluding outliers from possible reference views should be developed. The reference view could then be chosen from the remaining views based on histogram differences between views or distance from the center of the camera setup.

ACKNOWLEDGMENT

The authors would like to thank P. Prokein and W. Schnurrer for their helpful assistance with the implementation of the algorithm and the simulations as well as the reviewers for their valuable comments. Furthermore, the authors express their thanks for providing test sequences to KDDI Corporation, the Interactive Visual Media Group at Microsoft Research, Mitsubishi Electric Research Laboratories (MERL) and Fraunhofer HHI.

REFERENCES

[1] B. Wilburn, M. Smulski, K. Lee, and M. A. Horowitz, "The light field video camera," in *Proc. Media Processors SPIE Electron. Imag.*, San Jose, CA, Jan. 19–25, 2002, pp. 29–36.
[2] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH*, New Orleans, LA, Aug. 4–9, 1996, pp. 31–42.
[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH*, New Orleans, LA, Aug. 4–9, 1996, pp. 43–54.
[4] A. Vetro, W. Matusik, H. Pfister, and J. Xin, "Coding approaches for end-to-end 3-D TV systems," in *Proc. Picture Coding Symp. (PCS 2004)*, San Francisco, CA, Dec. 15–17, 2004.
[5] M. Tanimoto, "Free viewpoint television—FTV," in *Proc. Picture Coding Symp. (PCS 2004)*, San Francisco, CA, Dec. 15–17, 2004.
[6] C. Theobalt, G. Ziegler, M. Magnor, and H.-P. Seidel, "Model-based free-viewpoint video: Acquisition, rendering, and encoding," in *Proc. Picture Coding Symp. (PCS 2004)*, San Francisco, CA, Dec. 15–17, 2004.
[7] F. Vogt, S. Krüger, J. Schmidt, D. Paulus, H. Niemann, W. Hohenberger, and C. H. Schick, "Light fields for minimal invasive surgery using an endoscope positioning robot," *Methods Inf. Medicine*, vol. 43, no. 4, pp. 403–408, 2004.
[8] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[9] A. Kaup and U. Fecker, "Analysis of multi-reference block matching for multi-view video coding," in *Proc. 7th Workshop Digital Broadcasting*, Erlangen, Germany, Sep. 2006, pp. 33–39.
[10] H. Kimata, M. Kitahara, K. Kamikura, and Y. Yashima, "Multi-view video coding using reference picture selection for free-viewpoint video communication," in *Proc. Picture Coding Symp. (PCS 2004)*, San Francisco, CA, Dec. 15–17, 2004.
[11] M. Flierl and B. Girod, "Coding of multi-view image sequences with video sensors," in *Proc. IEEE Int. Conf. Image Process. (ICIP 2006)*, Atlanta, GA, Oct. 8–11, 2006, pp. 609–612.
[12] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical B pictures," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Document JVT-P014*, Poznan, Poland, Jul. 23–29, 2005.
[13] K. Mueller, P. Merkle, H. Schwarz, T. Hinz, A. Smolic, T. Oelbaum, and T. Wiegand, "Multi-view video coding based on H.264/MPEG4-AVC using hierarchical B pictures," in *Proc. Picture Coding Symp. (PCS 2006)*, Beijing, China, Apr. 24–26, 2006.
[14] A. Vetro, P. Pandit, H. Kimata, and A. Smolic, "Joint multiview video model (JMVM) 3.0," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Document JVT-V207*, Marrakech, Morocco, Jan. 13–19, 2007.
[15] J. López, J. H. Kim, A. Ortega, and G. Chen, "Block-based illumination compensation and search techniques for multiview video coding," in *Proc. Picture Coding Symp. (PCS 2004)*, San Francisco, CA, Dec. 15–17, 2004.
[16] Y.-L. Lee, J.-H. Hur, Y.-K. Lee, K.-H. Han, S. Cho, N. Hur, J. Kim, J.-H. Kim, P.-L. Lai, A. Ortega, Y. Su, P. Yin, and C. Gomila, "CE11: Illumination compensation," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Document JVT-U052r2*, Hangzhou, China, Oct. 20–27, 2006.
[17] W.-S. Shim, G. H. Park, and J.-H. Yang, "CE5: Illumination compensation," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Document JVT-V305*, Marrakech, Morocco, Jan. 13–19, 2007.
[18] A. P. Hekstra, J. G. Beerends, D. Ledermann, F. E. de Caluwe, S. Kohler, R. H. Koenen, S. Rihs, M. Ehrsam, and D. Schlauss, "PVQM—A perceptual video quality measure," *Signal Process.: Image Commun.*, vol. 17, no. 10, pp. 781–798, Nov. 2002.
[19] U. Fecker, M. Barkowsky, and A. Kaup, "Improving the prediction efficiency for multi-view video coding using histogram matching," in *Proc. Picture Coding Symp. (PCS 2006)*, Beijing, China, Apr. 2006.
[20] Y. Su, A. Vetro, and A. Smolic, "Common test conditions for multiview video coding," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Document JVT-U211*, Hangzhou, China, Oct. 20–27, 2006.

[21] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *ITU-T SG 16 Q. 6 Video Coding Experts Group (VCEG), Document VCEG-M33*, Austin, TX, Apr. 2–4, 2001.

[22] U. Fecker, M. Barkowsky, and A. Kaup, "Time-constant histogram matching for colour compensation of multi-view video sequences," in *Proc. 26th Picture Coding Symp. (PCS 2007)*, Lisbon, Portugal, Nov. 2007.

[23] Y. Wang, J. Ostermann, and Y.-Q. Zhang*, Video Processing and Communications.*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[24] E. Dubois and S. Sabri, "Noise reduction in image sequences using motion-compensated temporal filtering," *IEEE Trans. Commun.*, vol. COM-32, no. 7, pp. 826–832, Jul. 1984.

[25] N. Vasconcelos and F. Dufaux, "Pre and post-filtering for low bit-rate video coding," in *Proc. Int. Conf. Image Process. (ICIP 1997)*, Washington, DC, Oct. 26–29, 1997, vol. 1, pp. 291–294.

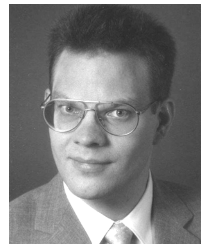
[26] H. Tsuji, T. Sakatani, Y. Yashima, and N. Kobayashi, "A nonlinear spatio-temporal diffusion and its application to prefiltering in MPEG-4 video coding," in *Proc. Int. Conf. Image Process. (ICIP 2002)*, Rochester, NY, Sep. 22–25, 2002, vol. 1, pp. I-85–I-88.

[27] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Perceptual prefiltering for video coding," in *Proc. Int. Symp. Intell. Multimedia, Video and Speech Process.*, Hong Kong, Oct. 20–22, 2004, pp. 510–513.

Ulrich Fecker received the Dipl.-Ing. degree in electrical engineering from the University of Ulm, Germany, in 2003. He is currently working toward the Dr.-Ing. degree at the University of Erlangen-Nuremberg, Germany.

During the winter of 2002/2003, he was an industrial trainee for the R&D Department, British Broadcasting Corporation (BBC), Kingswood, U.K. Since January 2004, he has been a Research Scientist with the Chair of Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg. His research interests include video coding and processing with a focus on multiview video and dynamic light fields. From 2004 to 2007, he worked on a subproject of the DFG Collaborative Research Center 603 "Model-Based Analysis and Visualization of Complex Scenes and Sensor Data".

Mr. Fecker is a member of the German VDE and ITG.



Marcus Barkowsky received the Dipl.-Ing. degree in electrical engineering from the University of Erlangen-Nuremberg, Germany, in 1999. His doctoral work focused on a reliable video quality measure for low bitrate scenarios with special emphasis on mobile transmission.

From 2000 to 2002, he was a Research Scientist with the Fraunhofer Institute for Integrated Circuits (IIS-A), Erlangen. In 2001, he have a lecture at the Technical University of Ilmenau, Germany. In 2002 he joined the Chair of Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg and, since 2007, is with OPTICOM GmbH, Erlangen.

André Kaup (M'96–SM'99) received the Dipl.-Ing. and Dr.-Ing. degrees from RWTH Aachen University, Germany, in 1989 and 1995, respectively, both in electrical engineering.

From 1989 to 1995, he was with the Institute for Communication Engineering, Aachen University of Technology, Aachen, where he was responsible for industrial as well as academic research projects in the area of high-resolution printed image compression, object-based image analysis and coding, and models for human perception. In 1995, he joined the Networks and Multimedia Communications Department, Siemens Corporate Technology, Munich, Germany, where he chaired work packages in several European research projects in the areas of very low-bit-rate video coding, image-quality enhancement, and mobile multimedia communications. In 1999 he was appointed head of the mobile applications and services group in the same department, with research focusing on multimedia adaptation for heterogeneous communication networks. Since 2001, he has been a full Professor and head of the Chair of Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg, Germany. From 1997 to 2001, he was head of the German MPEG delegation and from 1998 to 2001 he also served as an Adjunct Professor with the Technical University of Munich and the University of Erlangen-Nuremberg, teaching courses on image and video communication. From 2005 to 2007 he was vice-speaker of the DFG Collaborative Research Center 603 "Model-Based Analysis and Visualization of Complex Scenes and Sensor Data." He has published over 120 scientific journal and conference papers and holds 25 patents.

Prof. Kaup is member of the German Informationstechnische Gesellschaft. He was elected Siemens Inventor of the Year 1998 and was a recipient of the 1999 ITG Award.