J. Vis. Commun. Image R. 21 (2010) 392–403

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Fast color correction for multi-view video by modeling spatio-temporal variation

Feng Shao [a], Gang-Yi Jiang [a,*], Mei Yu [a], Yo-Sung Ho [b]

[a] Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China
[b] Dept. of Inform. & Comm., Kwangju Institute of Science and Technology, Kwangju 500-712, Korea

ARTICLE INFO

Article history:
Received 1 July 2009
Available online 11 March 2010

Keywords:
3DTV
FVV
Multi-view imaging
Color correction
Spatial color discrepancy model
Temporal variation model
Linear regression
Time-invariant detection

ABSTRACT

In multi-view video, a number of cameras capture the same scene from different viewpoints. Color variations between the camera views may deteriorate the performance of multi-view video coding or virtual view rendering. In this paper, a fast color correction method for multi-view video is proposed by modeling spatio-temporal variation. In the proposed method, multi-view keyframes are defined to establish the spatio-temporal relationships for accurate and fast implementation. For keyframes, accurate color correction is performed based on spatial color discrepancy model that disparity estimation is used to find correspondence points between views, and linear regression is performed on these sets of points to find the optimal correction coefficients. For non-keyframes, fast color correction is performed based on temporal variations model that time-invariant regions are detected to reflect the change trends of correction coefficients. Experimental results show that compared with other methods, the proposed method can promote the correction speed greatly without noticeable quality degradation, and obtain higher coding performance.

1. Introduction

Three-dimensional (3D) video and imaging technologies is an emerging trend in the development of digital video systems, as we presently witness the appearance of 3D displays, coding systems and 3D camera setups. New video applications such as three-dimensional television (3DTV) and free viewpoint video (FVV) system have drawn wide attention [1]. 3DTV aims to provide viewer depth perception of the scene by simultaneously rendering multiple images for different viewing angles. Instead, FVV application provides the ability for user to interactively select view of the scene. Since more views of a given scene would be needed in such applications, multi-view acquisition [2], multi-view video coding (MVC) [3] and virtual view rendering [4], etc, are the key technologies for 3DTV and FVV applications.

In a multi-view video system, multiple cameras capture the same scene to form multi-view videos. Many factors may lead to color inconsistency between views during acquisition. For example, it is often impossible to capture an object under perfectly constant lighting conditions across the views. Additionally, with multiple cameras, the internal setting like the exposure or focus setting may be varied. Furthermore, specular reflection may occur that light is reflected in different directions with varying intensity. Therefore, an important problem is how to compensate color differences between different views of images and videos when captured with multiple cameras.

Since multi-view video contains a large amount of inter-view statistical dependencies, both temporal and inter-view predictions are exploited in MVC [5,6]. However, when multi-view video data are recorded, significant variations are often observed between the luminance and chrominance components of different views. These discrepancies will reduce the correlation between different views, and therefore affect compression efficiency. Furthermore, these discrepancies will negatively affect rendering of new virtual views, and lead to subjective quality degradation such as dizziness and unnaturalness especially in random view access. Much works were carried out to compensate the illumination changes among multi-view cameras [7–9]. To improve the compression efficiency, weighted prediction had already existed in H.264/AVC, which supports a multiplicative weighting factor and an additive offset [7]. Sohn et a1. used average and variance in blocks to compensate color and luminance in order to use H.264/AVC for MVC [8]. As part of the standardization efforts of the JVT, a block-based illumination compensation scheme was proposed for MVC [9], and adopted in joint multiview video model (JMVM). However, for these methods, the compensation process must be inversed at the decoder, and the decoded views are still color inconsistent.

Color correction is an effective way to compensate the color inconsistency in multi-view video. Many color correction methods were proposed, in most of which color pattern board was used to extract color mapping relationship between views [10]. However, it is not easy to provide a color pattern in various imaging conditions. Yamamoto et al. corrected luminance and chrominance of views by using lookup tables, the correspondences of which were detected by scale invariant feature transform [11]. Chen used

* Corresponding author.
E-mail address: jianggangyi@126.com (G.-Y. Jiang).

1047-3203/$ - see front matter Crown Copyright © 2010 Published by Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2010.03.001

linear transformation of YUV channels, the coefficients of which were obtained by simplified color error model [12]. Fecker modified the luminance and chrominance variations by calculating lookup tables from histograms of two views [13]. In our previous method, color correction was treated as an optimization problem, and the effective color mapping relationships were calculated by using dynamic programming technique [14].

On the other hand, there are many works that perform fast implementation of color correction for multi-view images [15]. While extending to multi-view video, how to improve the correction accuracy and reduce computational complexity simultaneously remains an important problem worthy of study. The most direct strategy is to perform color correction in frame-by-frame mode, however the computational complexity is usually intolerable. In our previous work, once color mapping relationship is established at one time, the following color mapping relationship will be the same [16]. In [13], Fecker also supposed that the mapping function is time-constant during a certain time period. In fact, the assumption is not always kept especially when the multi-view imaging is time-varying. In general, it is a more difficult problem to achieve high correction accuracy and low computational complexity simultaneously. More accuracy correction can yield a significant increase in computational complexity, and vice versa. Since the inter-view and temporal correlations are usually strong in multi-view video, if we can model spatio-temporal variation, the above problem can be solved easily.

In this paper, we propose a fast color correction method for multi-view video. The aim of the proposed method is to achieve lower computational complexity and higher accuracy simultaneously when color correction is performed for multi-view video. The paper begins by describing the spatial color discrepancy model and temporal variation model in multi-view imaging. Based on spatial color discrepancy model, a new color correction method for multi-view images is proposed by using disparity estimation and linear regression. Then, based on the temporal variation model, color correction is extended to multi-view video. Finally, experimental results for different test sequences are given and the performances of computational complexity, correction accuracy and coding efficiency are compared to those of other methods.

2. Spatial color discrepancy model in multi-view imaging

Multi-view video is captured by multiple cameras simultaneously from different viewpoints. For illustration, we show an example of a typical multi-view imaging model with two cameras in Fig. 1. In this case, the same object will appear different colors in different views due to various imaging factors, and these inconsistencies will degrade the performance of subsequent MVC or virtual view rendering. On the other hand, once the imaging condition is fixed, the color change trends will be constant during acquisition. Therefore, it is necessary to model the color discrepancy in multi-view imaging.

The image intensity I_k taken from a Lambertian surface by a digital color camera can be described as

$$I_k = \int_\lambda E(\lambda)S(\lambda)R_k(\lambda)d\lambda \tag{1}$$

where $E(\lambda)$ denotes spectral power distribution of the illumination, $S(\lambda)$ denotes the surface spectral reflectance of the object and $R_k(\lambda)$ is spectral sensitivity of the kth camera sensors.

A finite dimensional linear model [17] is used to describe the spectral function of illumination and surface reflectance as a linear combination of several basis functions

$$S(\lambda) = \sum_{j=1}^{n} \sigma_j s_j(\lambda) \quad , E(\lambda) = \sum_{j=1}^{m} \varepsilon_j e_j(\lambda) \tag{2}$$

where m and n denote the numbers of basis functions, here, m and n are set 3. $s_j(\lambda)$ is the jth reflectance basis function, and σ_j is its weighting coefficient. $e_j(\lambda)$ is the jth basis function for the illuminant, and ε_j is its weighting coefficient.

Let $\mathbf{A}$ be the 3×3 spread matrix that corresponds to illumination and surface reflectance, $\mathbf{b}$ be the 3×1 offset vector of the imaging sensors, $\mathbf{x}$ be the 3×1 vector of ideal values at a particular pixel, and $\mathbf{y}$ is the 3×1 vector containing the measured values. Then, under the finite dimensional linear model assumption, the color discrepancy can be modeled as a linear transformation

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{3}$$

Starting from Eq. (3), we can derive color discrepancy model in multiple cameras. As shown in Eq. (4), current and reference view images are obtained at the same time instant from different cameras for the same scene

$$\begin{aligned} \mathbf{y}_{ref} &= \mathbf{A}_1\mathbf{x}_{ref} + \mathbf{b}_1 \\ \mathbf{y}_{cur} &= \mathbf{A}_2\mathbf{x}_{cur} + \mathbf{b}_2 \end{aligned} \tag{4}$$

where $\mathbf{x}_{ref}$ and $\mathbf{x}_{cur}$ are ideal color values for current and reference view images, respectively. $\mathbf{y}_{ref}$ and $\mathbf{y}_{cur}$ are the corresponding actual acquired color values for different cameras, respectively. Supposing that $\mathbf{x}_{ref}$ and $\mathbf{x}_{cur}$ are consistent in the matching pixels, the color discrepancy model between $\mathbf{y}_{ref}$ and $\mathbf{y}_{cur}$ can be described as

$$\begin{aligned} \hat{\mathbf{y}}_{ref} &= \mathbf{A}_1\mathbf{A}_2^{-1}(\mathbf{y}_{cur} - \mathbf{b}_2) + \mathbf{b}_1 = \mathbf{M}\mathbf{y}_{cur} + \mathbf{T} \\ \mathbf{M} &= \mathbf{A}_1\mathbf{A}_2^{-1}, \mathbf{T} = \mathbf{b}_1 - \mathbf{A}_1\mathbf{A}_2^{-1}\mathbf{b}_2 \end{aligned} \tag{5}$$

where $\hat{\mathbf{y}}_{ref}$ denotes the matching pixels in reference view image that corresponds to $\mathbf{y}_{cur}$. However, due to the limited scene information

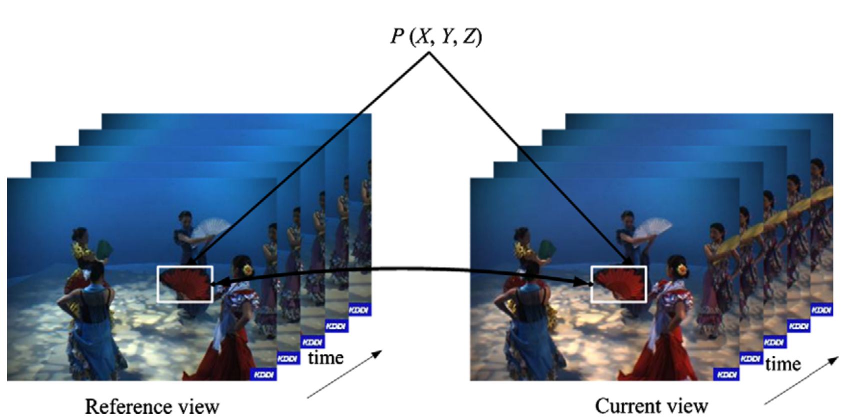


Fig. 1. Illustration of multi-view imaging.

from only two images, the color discrepancy model derived at one specific time instant cannot reflect the actual change trends at all times. Thus, temporal extension of the color discrepancy model is necessary.

3. Temporal variation model in multi-view imaging

In Fig. 2, considering color discrepancy over time, the same object will give different color values from different cameras at different time instant. In multi-camera imaging geometry, the same object will give similar motion trajectory in different cameras, and the disparity in neighboring views at different time stamp will be equivalent. Supposing that $I_s^t(x_0, y_0)$ and $I_{s+1}^t(x_1, y_1)$ are the color values for different views at time t, and $I_s^{t+1}(x_2, y_2)$ and $I_{s+1}^{t+1}(x_3, y_3)$ are the corresponding color values for different views at time $t + 1$, respectively. Let $g_1(\cdot)$ be color discrepancy relationship at t time, and $f_1(\cdot), f_2(\cdot)$ be temporal variation relationships for the view s and view $s + 1$, respectively. If we have derived the variation relationships as

$$\begin{aligned} I_s^t(x_0, y_0) &= g_1(I_{s+1}^t(x_1, y_1)) \\ I_s^t(x_0, y_0) &= f_1(I_s^{t+1}(x_2, y_2)) \\ I_{s+1}^t(x_1, y_1) &= f_2(I_{s+1}^{t+1}(x_3, y_3)) \end{aligned} \tag{6}$$

Thus, the relationship between $I_s^{t+1}(x_2, y_2)$ and $I_{s+1}^{t+1}(x_3, y_3)$ can be easily represented by

$$I_s^{t+1}(x_2, y_2) = f_1^{-1}(g_1(f_2(I_{s+1}^{t+1}(x_3, y_3)))) = g_2(I_{s+1}^{t+1}(x_3, y_3)) \tag{7}$$

where $g_2(\cdot)$ denotes the color discrepancy relationship at time $t + 1$.

Therefore, if we can model the temporal variation $f_1(\cdot)$ and $f_2(\cdot)$ in the above equation, the relationship between color discrepancy models $g_1(\cdot)$ and $g_2(\cdot)$ can be easily established.

4. The proposed fast color correction method for multi-view video

Fig. 3 shows a typical FVV reference model [18]. On the sender side, multi-view images are captured with multiple cameras. The captured images may contain the misalignment and color differences of the cameras. Then image correction, including geometric calibration and color correction, should be performed. The corrected multi-view images are compressed for transmission and storage by the encoder. On the receiver side, free viewpoint images are generated by interpolating the decoded images and displayed on a 2D/3D display. Therefore, color correction is a very important process in the FVV system.

Since it is not practical to calculate the correction matrices in a frame-by-frame mode, in order to find a better tradeoff between its complexity and accuracy, multi-view keyframes are defined and applied. Different with the keyframe concept in computer animation [19], the keyframe in this paper corresponds to a triggering, which hints correction coefficients should be updated. For keyframes, the correction matrices are obtained by performing color correction directly. While for non-keyframes, the correction matrices are obtained from previous frames by utilizing temporal variation model. The interval between keyframes can be decided in fixed or adaptive way. In the proposed method, in order to be compatible with the following MVC, we only consider the fixed keyframes interval, which is similar to the size of group of picture (GOP) in MVC.

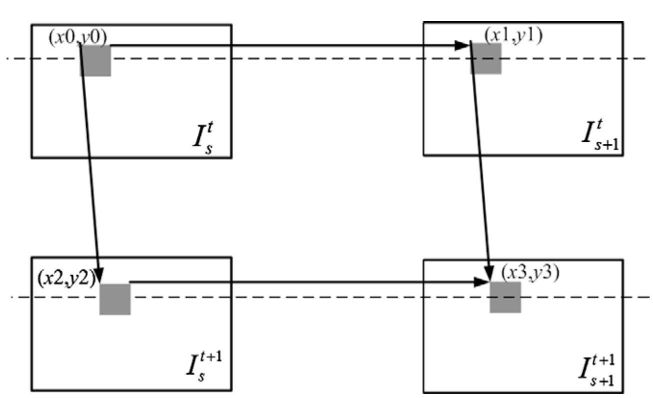


Fig. 2. Relationships of temporal and inter-view correlations.

Thus, based on the above defined multi-view keyframes, a new color correction method for multi-view video is proposed. As shown in Fig. 4, the method is summarized as follows.

Step 1. Check whether the current multi-view frames are keyframes or not. If they are keyframes then go to Step 2, otherwise, go to Step 3.
Step 2. Calculate the disparity vectors between multi-view images by mean-removed sum of absolute differences (MRSAD) based disparity estimation. Then, achieve color correction of multi-view keyframes with linear regression.
Step 3. Detect time-invariant regions between two consecutive frames. Then, achieve fast color correction based on the temporal variation model.

4.1. The proposed color correction method for multi-view images

In the proposed method, keyframes provide a suitable abstraction and description of the color discrepancy information between views, and the color discrepancy model for the subsequent frames is inherited from the keyframes. Therefore, the keyframes should be corrected accurately. The above color discrepancy model is derived from RGB format data. While for YUV video format, a linear color transformation is also satisfied because the transformation from RGB to YUV is also linear. Supposing view s as reference view, color correction is performed for view $s + 1$ to obtain consistent color appearance with the reference view. The color corrected values for each component is expressed as a weighted linear sum of the current YUV values

$$\begin{bmatrix} Y_{cor} \\ U_{cor} \\ V_{cor} \end{bmatrix} = \begin{bmatrix} \alpha_{Y1} & \alpha_{Y2} & \alpha_{Y3} \\ \alpha_{U1} & \alpha_{U2} & \alpha_{U3} \\ \alpha_{V1} & \alpha_{V2} & \alpha_{V3} \end{bmatrix} \begin{bmatrix} Y_{s+1} \\ U_{s+1} \\ V_{s+1} \end{bmatrix} + \begin{bmatrix} a_{Y4} \\ a_{U4} \\ a_{V4} \end{bmatrix} \tag{8}$$

In order to derive the above coefficients, we use MRSAD-based disparity estimation [9] to find matching points between reference and current views, and perform a linear regression to derive the coefficients. For an $N \times N$ block of pixels located at position (x, y), *MRSAD* is defined as

$$MRSAD(i,j) = \sum_{x=x_0}^{x_0+N} \sum_{y=y_0}^{y_0+N} |(Y_{s+1}(x,y) - \mu_{s+1}) - (Y_s(x+i, y+j) - \mu_s)| \tag{9}$$

where $Y_{s+1}(x, y)$ and $Y_s(x, y)$ are luminance values of the view s and the view $s + 1$, respectively, (i, j) represents a candidate disparity vector, and μ_{s+1} and μ_s are the mean values of pixels for each blocks, respectively.

Here, the current view is firstly divided into 8×8 blocks, and for each block, the disparity vector $\mathbf{d}_{s+1\to s}$ from view $s + 1$ to view s is calculated by minimizing the *MRSAD* over a search range

$$\mathbf{d}_{s+1\to s} = \arg\min_{(i,j)\in\Omega} MRSAD(i,j) \tag{10}$$

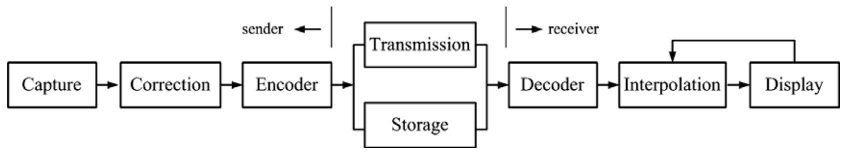


Fig. 3. FVV reference model.

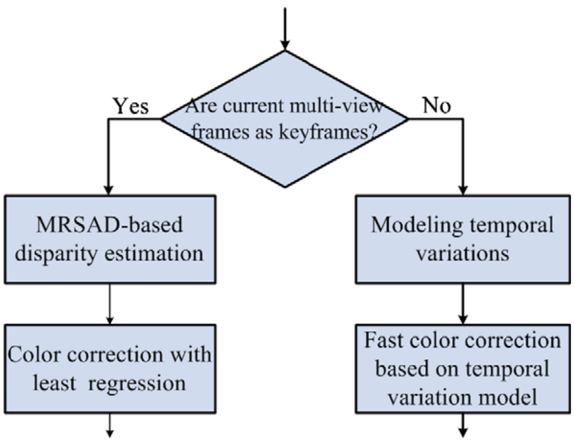


Fig. 4. Flowchart of the proposed method.

where Ω denotes the disparity search range. However, the disparity vectors estimated with the above formula may not be the true disparity due to occlusion, mismatching or other factors. An inverse matching operation from reference view to current view is performed to validate the matching. If the disparity deviation between $\mathbf{d}_{s+1\to s}$ and $\mathbf{d}_{s\to s+1}$ is less than 2, that is, $|\mathbf{d}_{s+1\to s}+\mathbf{d}_{s\to s+1}| < 2$, then the blocks are matched.

Then, all matching pixels in current and reference views are defined in vector forms as $\mathbf{Y}_{s+1}$, $\mathbf{U}_{s+1}$, $\mathbf{V}_{s+1}$, $\mathbf{Y}_s$, $\mathbf{U}_s$, $\mathbf{V}_s$. Let $\mathbf{\Psi} = [\mathbf{Y}_{s+1}, \mathbf{U}_{s+1}, \mathbf{V}_{s+1}, 1]$ and $\mathbf{\Phi} = [\mathbf{Y}_s, \mathbf{U}_s, \mathbf{V}_s]$, the relationships between the vectors of matching points can be expressed as

$$\mathbf{\Phi} = \mathbf{\Psi}\mathbf{C} + \boldsymbol{\varepsilon} \tag{11}$$

where $\boldsymbol{\varepsilon}$ is error vector, and $\mathbf{C}$ is a 4×3 matrix. By minimizing the energy of the error vector $\boldsymbol{\varepsilon}$ in a least square sense [20], the correction matrix $\mathbf{C}$ can be computed by

$$\mathbf{C} = (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\mathbf{\Psi}^T\mathbf{\Phi} \tag{12}$$

In order to have a similar form with the color discrepancy model in Eq. (5), the matrix $\mathbf{C}$ is divided into a 3×3 matrix $\mathbf{M}$ and a 3×1 matrix $\mathbf{T}$. Supposing $\mathbf{H} = [Y, U, V]^T$ for each pixel (x, y), using the matrices $\mathbf{M}$ and $\mathbf{T}$, the color discrepancy model is described as

$$\hat{\mathbf{H}}_s = \mathbf{M}\mathbf{H}_{s+1} + \mathbf{T} + \boldsymbol{\varepsilon} \tag{13}$$

where $\hat{\mathbf{H}}_s$ and $\mathbf{H}_{s+1}$ are the corresponding color vectors of view s and view $s + 1$, respectively.

4.2. The proposed fast color correction method for multi-view video

By performing the above method for multi-view images frame by frame, it can be easily extended to video. However, for low complexity applications, such as wireless multi-view video system [21], it is not practical to carry out the same process for every frame since performing disparity estimation and linear regression would be computationally expensive. In our previous work [15], once the color mapping relationship established at one time, the mapping will be directly applied frame by frame, which may lead to a perceptible flicker. The primary case of flicker comes from scene change. Supposed at time n, object c is mapped to c' as shown in Fig. 5(a). At time $(n + m)$, another object appears with varying illumination, and it is mapped to c'', as shown in Fig. 5(b). If consistent mapping function is performed at time n and $(n + m)$, flicker may occur.

In order to show the influence of flicker, we compare the correction coefficients a_{Y1}, a_{Y2}, a_{Y3} and a_{Y4}, as shown in Fig. 6, which are obtained by performing disparity estimation and linear regression frame by frame. It is obvious that only minor changes occur for the coefficients a_{Y1}, a_{Y2} and a_{Y3}, while for the coefficient a_{Y4}, the changes are drastic. In other words, even though those coefficients can lead to a good fitting between different views at a particular time, it can also result in discontinuous variations between adjacent time instants, which may deteriorate the temporal correlations. Therefore, it is necessary to model the temporal variations to reflect the actual variation trends of the coefficients. In the existing techniques, Bayesian model [22] or Kalman filtering [23] can be used to predict the temporal variations. However, for both Bayesian and Kalman filtering models, it is difficult and time-consuming to obtain accurate model coefficients. Here, we propose a method with temporal information to fast obtain the correction matrices. The proposed method mainly consists of two steps, time-invariant detection and temporal variation modeling.

4.2.1. A. Time-invariant detection

The objective of time-invariant detection is to extract the background regions that are unchanged over time. For single-view video, illumination change is the main reason for discontinuous color variations among frames. However, the illumination in foreground motion regions is instable. First, motion objects may vary with time, which make it hard to measure the illumination intensity for these regions. Second, even for the same objects, the reflected intensity may also be inconsistent when the reflective surface is changed. In order to extract stable temporal variation

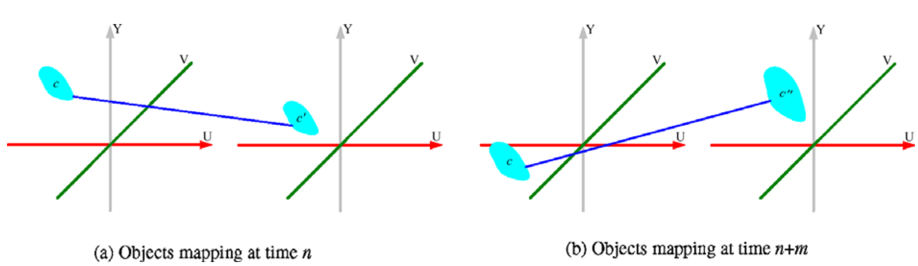
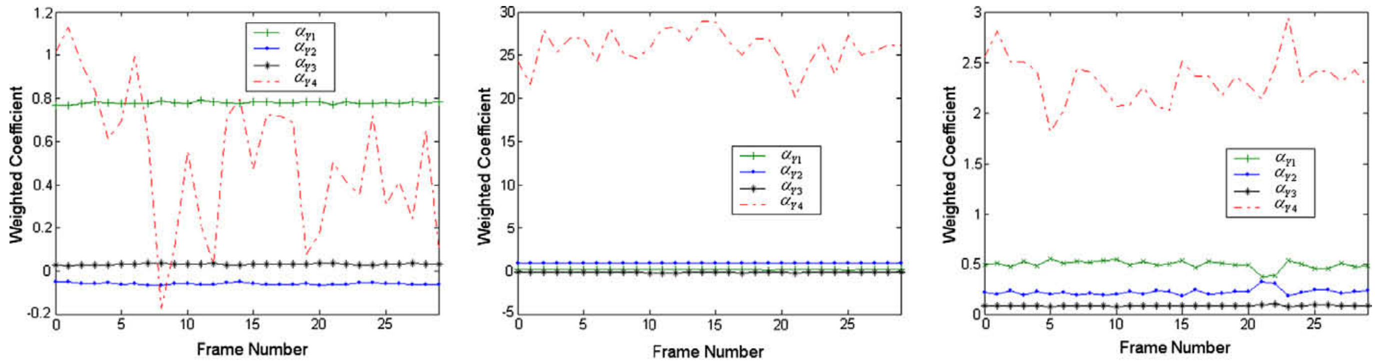


Fig. 5. Relationships of object mapping.



(a) '*Flamenco1*' sequence (b) '*Objects2*' sequence (c) '*Race1*' sequence

Fig. 6. Correction coefficients for Y component.

information in video, we use background as a uniform reference surface between two consecutive frames.

Here, we propose an algorithm that incorporate both motion and edge information to identify the background. First of all, an initial motion detection mask (MDM) at time t, denoted as MDMi_t, between two consecutive frames, is computed by a thresholding operation. The decision rule for obtaining MDMi_t can be expressed as

$$\mathrm{MDM}i_t(x,y) = \begin{cases} \text{not changed} & \text{if } D_t(x,y) \leqslant \beta\sigma_N \\ \text{changed} & \text{otherwise} \end{cases} \tag{14}$$

$$D_t(x,y) = |Y(x,y,t+1) - \mu^{t+1} - (Y(x,y,t) - \mu^t)| \tag{15}$$

$$\sigma_N^2 = \frac{1}{M}\sum_{(x,y)}(Y(x,y,t+1) - \mu^{t+1} - (Y(x,y,t) - \mu^t)^2 \tag{16}$$

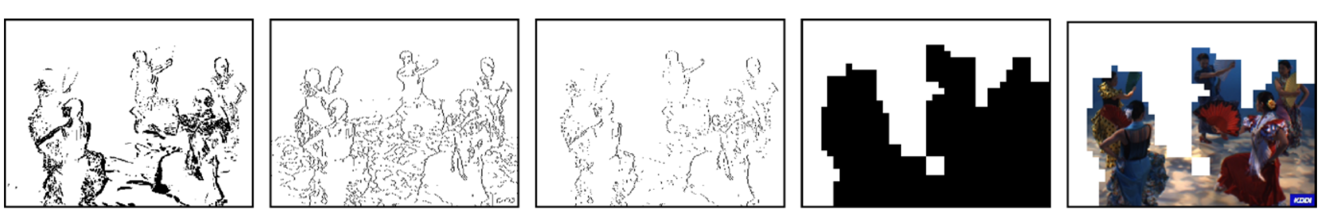
where $Y(x, y, t)$ and $Y(x, y, t + 1)$ are Y components of pixels at the tth and $(t + 1)$th frames at position (x, y), and μ^t and μ^{t+1} are the average luminance of the frames t and $t + 1$, respectively. β is a weigh factor which is set to 2 empirically. M is the number of pixels in a frame. Figs. 7–9(a) show an example of MDMi for '*Flamenco1*', '*Objects2*', and '*Race1*' sequences, respectively.

However, only taking illumination into account for change detection suffers from the problem that unchanged pixels are easily mistaken as changed ones. To overcome the problem and be more robust under varying illumination, moving object edge is incorporated to identify the regions with motion. We define the edge detection mask (EDM$_t$) as a set of all edge points detected by the SUSAN operation in current frame. Figs. 7–9(b) show the EDM$_t$ for '*Flamenco1*', '*Objects2*', and '*Race1*' sequences, respectively.

Then, by combining the above two masks, those unchanged pixels are excluded, and the final MDM (MDMf_t) for current frame is expressed as

$$\mathrm{MDM}f_t(x,y) = \mathrm{MDM}i_t(x,y) \cap \mathrm{EDM}_t(x,y) \tag{17}$$

Note that these unchanged pixels are basically removed after incorporating the extracted edges, as shown in Figs. 7–9(c). Finally, in order to separate the foreground and background, horizontal and vertical scanning with 8×8 block over MDMf_t is first performed. If the block is located in the middlepart of two edge blocks in horizontal and vertical scan line simultaneously, it is regarded as the foreground. Otherwise, it is regarded as the background. Finally, the regions are connected by simple connection templates. By applying the connected MDM (MDMc_t) in Figs. 7–9(d) on color video, the foreground regions are detected, as shown in Figs. 7–9(e).



(a) MDMi_t (b) EDM$_t$ (c) MDMf_t (d) MDMc_t (f) Foreground

Fig. 7. Example of the algorithm for detecting foreground regions of '*Flamenco1*'.

(a) MDMi_t (b) EDM$_t$ (c) MDMf_t (d) MDMc_t (f) Foreground

Fig. 8. Example of the algorithm for detecting foreground regions of '*Objects2*'.

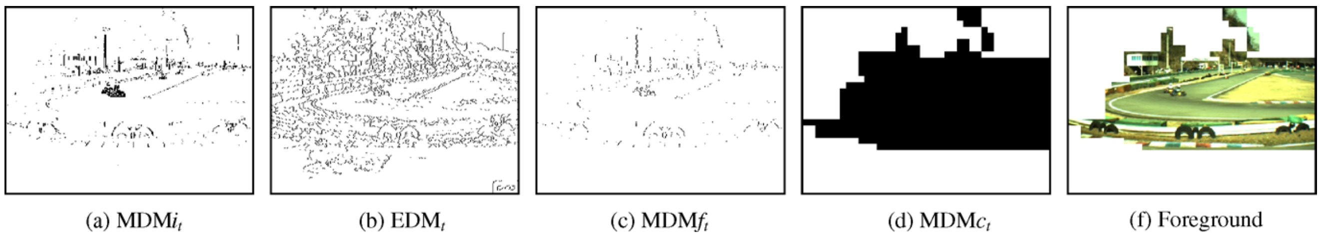


Fig. 9. Example of the algorithm for detecting foreground regions of '*Race1*'.

4.2.2. B. Fast color correction with temporal variation model

For single camera, color distributions in background regions of adjacent frames are assumed to Gaussians, $p(c,t) \sim N(\mu_1, \sigma_1)$ and $p(c,t+1) \sim N(\mu_2, \sigma_2)$, where c is the color value, $p(c,t)$ is the probability density function, and u and σ are the mean and standard deviation, respectively. In order to response the above distributions, the acquired intensity value for Y component between adjacent frames is modeled as

$$Y_s^t(x', y') = a \cdot Y_s^{t+1}(x, y) + b \tag{18}$$

where a is a scaling factor and b is an offset factor. By transferring the parameters of Gaussian models from the tth frame to the $(t+1)$th frame, the coefficients a and b are derived by

$$a = \sqrt{\sigma_s^{t+1}/\sigma_s^t}, \quad b = \mu_s^{t+1} - a \cdot \mu_s^t \tag{19}$$

where μ_s^t and μ_s^{t+1} are the means in Gaussian models of the view s at the tth and $(t+1)$th frames, respectively. ${\sigma_s}^t$ and ${\sigma_s}^{t+1}$ are the corresponding standard deviations. Considering YUV components of view s and view $s+1$, the temporal variation models can be further expressed as

$$\hat{\mathbf{H}}_s^t = \mathbf{Z}_s \mathbf{H}_s^{t+1} + \boldsymbol{\Theta}_s + \boldsymbol{\varepsilon} \tag{20}$$

$$\hat{\mathbf{H}}_{s+1}^t = \mathbf{Z}_{s+1} \mathbf{H}_{s+1}^{t+1} + \boldsymbol{\Theta}_{s+1} + \boldsymbol{\varepsilon} \tag{21}$$

where $\mathbf{Z}_s$, $\mathbf{Z}_{s+1}$ and Θ_s, Θ_{s+1} are the matrices in temporal variation model for view s and view $s+1$, respectively.

Supposing that the color discrepancy model at time t has been known, and it is described as

$$\hat{\mathbf{H}}_s^t = \mathbf{M}^t \mathbf{H}_{s+1}^t + \mathbf{T}^t + \boldsymbol{\varepsilon} \tag{22}$$

where $\mathbf{M}^t$ and $\mathbf{T}^t$ are the correction matrices at time t, respectively. Then, based on the spatio-temporal relationship in Eq. (7), the color discrepancy model at time $t+1$ can be described as

$$\hat{\mathbf{H}}_s^{t+1} = \mathbf{Z}_s^{-1} \mathbf{M}^t \mathbf{Z}_{s+1} \mathbf{H}_{s+1}^{t+1} + \mathbf{Z}_s^{-1}(\mathbf{M}^t \boldsymbol{\Theta}_{s+1} + \mathbf{T}^t - \boldsymbol{\Theta}_s) + \boldsymbol{\varepsilon} \tag{23}$$

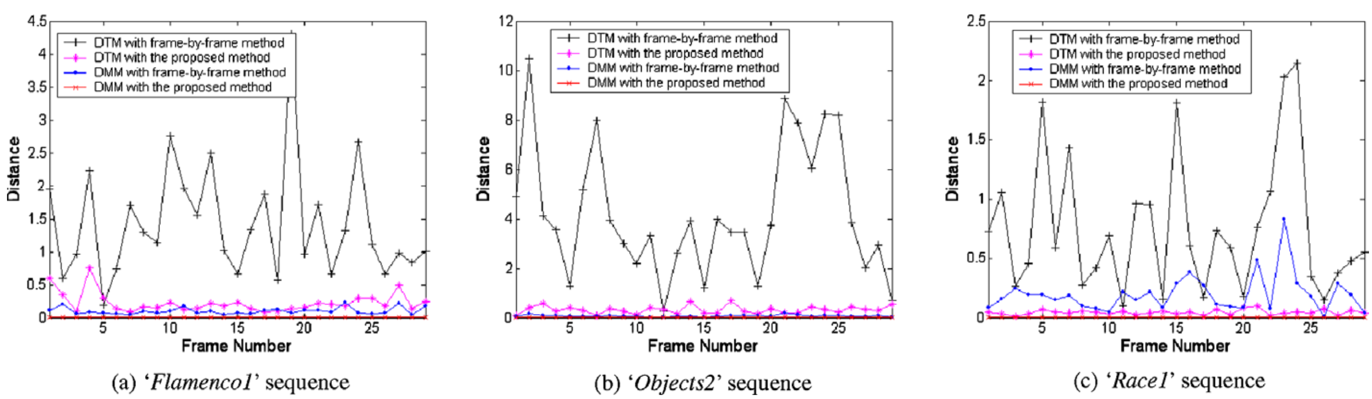


Fig. 10. Distance comparison between adjacent correction matrices.

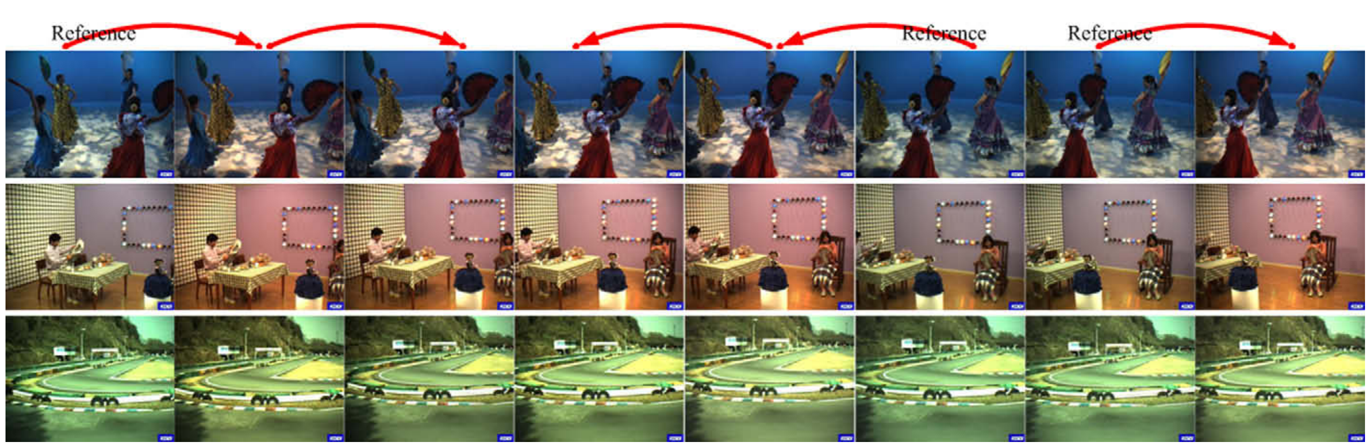


Fig. 11. Eight original viewpoint images of '*Flamenco1*', '*Objects2*' and '*Race1*'.

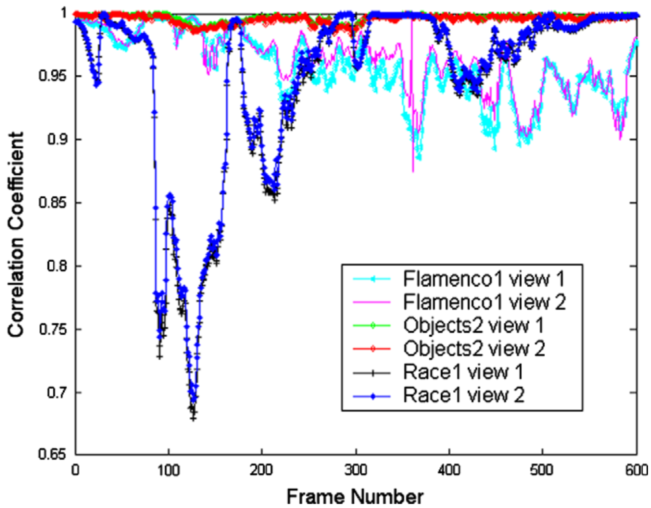


Fig. 12. Statistic results of correlation coefficients.

Supposing $\mathbf{M}^{t+1} = \mathbf{Z}_s^{-1}\mathbf{M}^t\mathbf{Z}_{s+1}$ and $\mathbf{T}^{t+1} = \mathbf{Z}_s^{-1}(\mathbf{M}^t\boldsymbol{\Theta}_{s+1} + \mathbf{T}^t - \boldsymbol{\Theta}_s)$, Eq. (23) is simplified as follows:

$$\hat{\mathbf{H}}_s^{t+1} = \mathbf{M}^{t+1}\mathbf{H}_{s+1}^{t+1} + \mathbf{T}^{t+1} + \boldsymbol{\varepsilon} \tag{24}$$

where $\hat{\mathbf{H}}_s^{t+1}$ and $\mathbf{H}_{s+1}^{t+1}$ are the corresponding color vectors of view s and view $s + 1$ at time $t + 1$, respectively. Thus, the color discrepancy model at time $t + 1$ is derived, and $\mathbf{M}^{t+1}$ and $\mathbf{T}^{t+1}$ are the corresponding correction matrices at time $t + 1$.

It is noteworthy that the proposed method has at least two important advantages. Firstly, by utilizing the temporal variation model, computational complexity of color correction is significantly reduced. Secondly, since the correction matrices are derived from previous frames, discontinuous variations between adjacent frames can be avoided. In order to describe the discontinuous variations, we calculate the distances of **M** matrices (DMMs) and distances of **T** matrices (DTMs) between two adjacent time instants. The distances are computing by adding the total errors for each element in the matrix. As shown in Fig. 10, it is obvious that both DMMs and DTMs with the proposed method are significantly lower than the frame-by-frame method, and the DMMs are almost no change, which guarantees smooth variations between subsequent frames.

5. Experimental results and analyses

In experiments, multi-view video sequences '*Flamenco1*', '*Objects2*' and '*Race1*' are used as the test sequences [24]. The size of the sequences is 320 × 240, and the images are taken by a horizontal parallel camera configuration with eight viewpoints. Fig. 11 shows eight original views of '*Flamenco1*', '*Objects2*', and '*Race1*'. Clearly, the color consistency among these eight views of the sequences is poor. In this case, the first, fifth and sixth views are chosen as the reference, and the other views are corrected by referencing the nearest view. To confirm the performance of the proposed method, we implemented our coding experiments in JMVM7.0 [25]. The test condition is set as follows: four basis QPs 22, 27, 32, 37 are used, the temporal GOP size is 15, and the total number of encoded frames in each view is 600.

Fig. 12 shows statistic results of the correlation coefficients of '*Flamenco1*', '*Objects2*' and '*Race1*' in the view 1 and view 2, respectively. It is evident that there are strong correlations between adjacent time instants since most of the correlation coefficients are larger than 0.9, and for '*Race1*', the coefficients are relatively low for some frames because of rapid scene changes. Therefore, it is necessary to model the temporal variations for multi-view video by exploiting the high correlations.

5.1. Computational complexity comparison

In order to objectively measure performances of the proposed method in computational complexity and correction accuracy, four color correction schemes are discussed

Scheme 1: Traditional color correction method which is performed frame by frame
Scheme 2: The proposed color correction method in which keyframes are put in every 10 frames
Scheme 3: The proposed color correction method in which keyframes are put in every 15 frames
Scheme 4: The proposed color correction method in which keyframes are put in every 30 frames

Table 1 shows experimental results of computational time in which *TS* indicates the average time saving and it is defined by

$$TS = \frac{T_{Scheme1} - T_{Proposed}}{T_{Scheme1}} \times 100\ (\%) \tag{25}$$

where $T_{Scheme1}$ and $T_{Proposed}$ are the total computational time for Scheme 1 and the proposed scheme, respectively. The maximum disparity search ranges in the experiments are 40 and 5 at horizontal and vertical directions, respectively. Table 1 shows the speedup per-

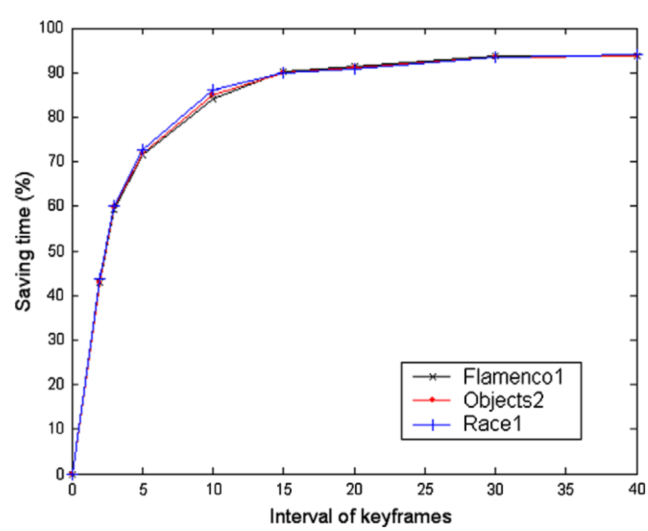
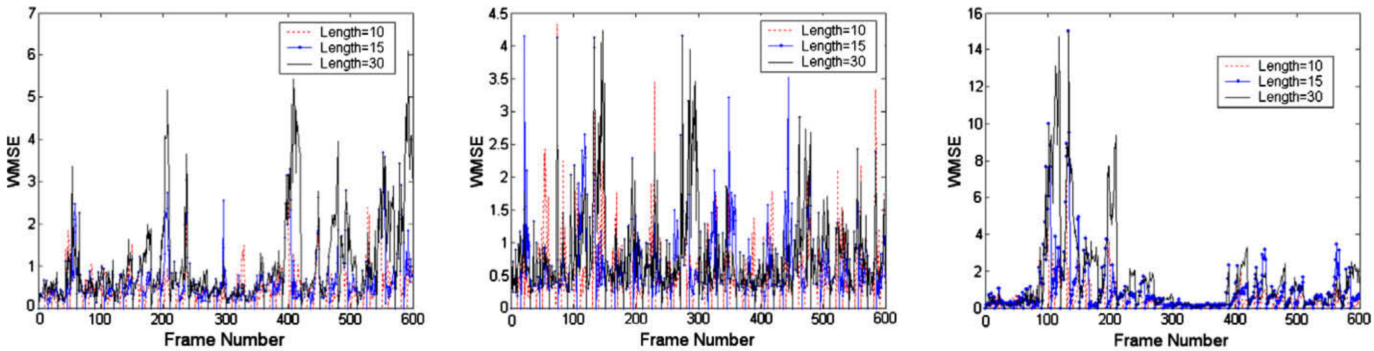


Fig. 13. The relationship between keyframe interval and saved time.

Table 1
Speedup performance comparisons of '*Flamenco1*', '*Objects2*' and '*Race1*'.

	$T_{Scheme1}$ (s)	$T_{Scheme2}$ (s)	*TS* (%)	$T_{Scheme3}$ (s)	*TS* (%)	$T_{Scheme4}$ (s)	*TS* (%)
Flamenco1	1396.875	222.421	84.08	137.250	90.17	89.781	93.57
Objects2	1372.515	205.203	85.05	137.671	89.97	90.546	93.40
Race1	1355.828	189.593	86.02	138.593	89.78	87.328	93.56



(a) '*Flamenco1*' sequence (b) '*Objects2*' sequence (c) '*Race1*' sequence

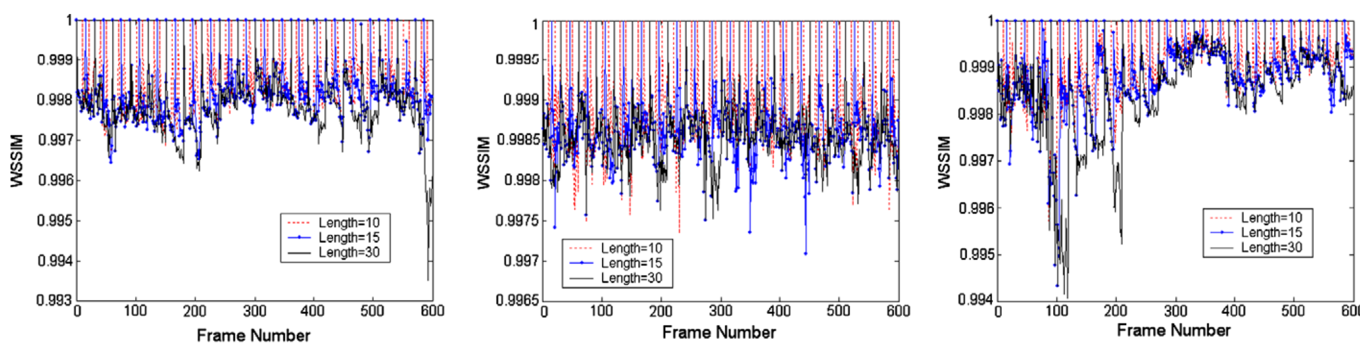
Fig. 14. WMSE comparison results.

formance of the proposed scheme with different keyframes intervals. It is obvious that the proposed scheme can significantly reduces the computational complexity, ranging from 84.08% to 93.57%, when the keyframes interval is ranging from 10 to 30. Fig. 13 shows the relationship between keyframes interval and saved time. From the figure, it is noted that more computational time can be saved with the proposed method, and the saved time increases with increasing keyframes interval length, while the accuracy of color correction will be also decreased with the increasing keyframes interval length. Therefore, an optimal keyframe interval will exist for the best trade-off between accurate correction and fast execution.

5.2. Objective and subjective correction performance comparison

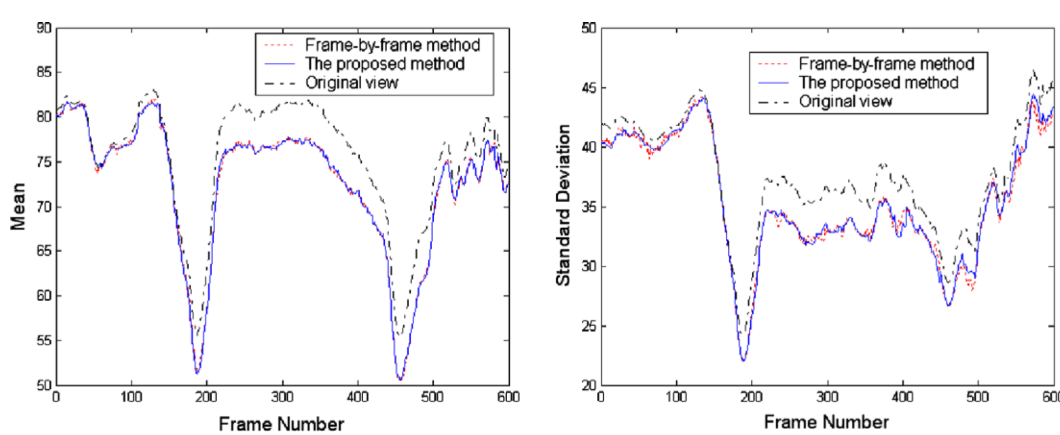
In order to objectively evaluate performance of the proposed method, the structural similarity index (SSIM) metric [26] is used to measure the similarity. We calculate the following weighted mean square errors (WMSE) and weighted SSIM (WSSIM) between the corrected images with frame-by-frame method and the proposed method. WMSE or WSSIM is calculated based on the weighted average on three components

$$WMSE = w_1 MSE_Y + w_2 MSE_U + w_3 MSE_V \tag{26}$$



(a) '*Flamenco1*' sequence (b) '*Objects2*' sequence (c) '*Race1*' sequence

Fig. 15. WSSIM comparison results.



(a) Mean (b) Standard deviation

Fig. 16. Mean and standard deviation comparisons of '*Flamenco1*'.

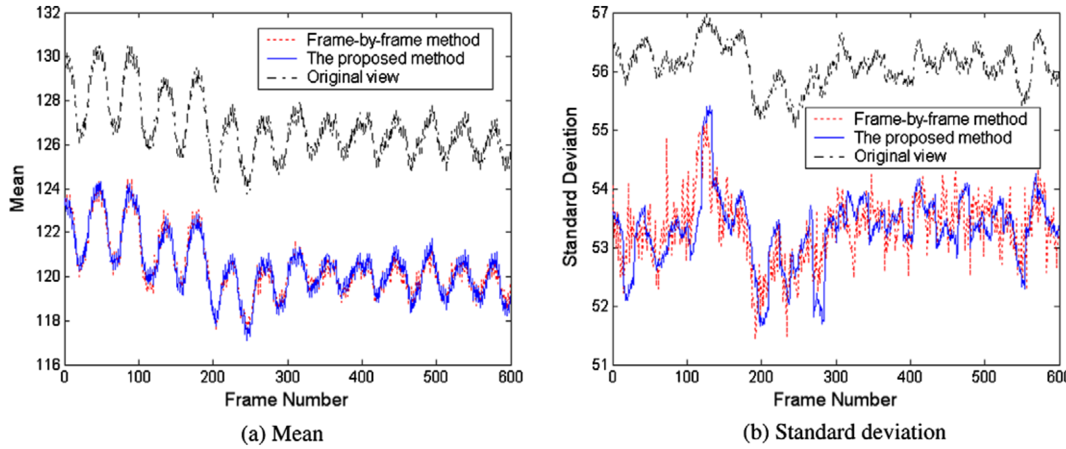


Fig. 17. Mean and standard deviation comparisons of '*Objects2*'.

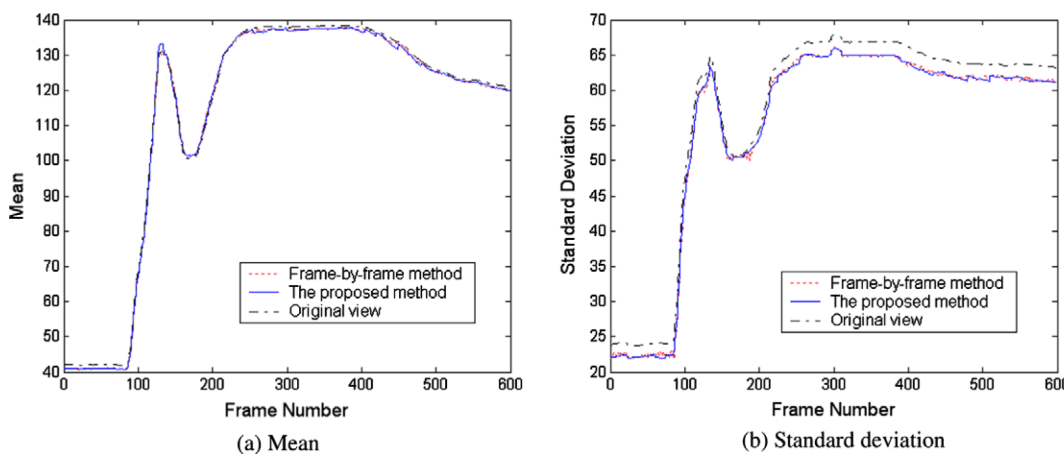


Fig. 18. Mean and standard deviation comparisons of '*Race1*'.

Fig. 19. Color correction results of '*Flamenco1*'. (a) Reference image (view 1, frame #559); (b) current image (view 2, frame #559); (c) the corrected image with HM method in [13]; (d) the corrected image with the proposed scheme.

Fig. 20. Color correction results of '*Objects2*'. (a) Reference image (view 1, frame #559); (b) current image (view 2, frame #559); (c) the corrected image with HM method in [13]; (d) the corrected image with the proposed scheme.

$$WSSIM = w_1 SSIM_Y + w_2 SSIM_U + w_3 SSIM_V \quad (27)$$

where w_1, w_2 and w_3 are weighted parameters, $w_1 = 0.8$, $w_2 = 0.1$, $w_3 = 0.1$ for Y, U and V components, respectively. Here, $0 \leqslant WSSIM \leqslant 1$, where $WSSIM = 1$ if two images are the same. As shown in Figs. 14 and 15, for different keyframe interval, WMSE values are relatively low, and WSSIM values are close to 1, which implies that the differences between the frame-by-frame method and the proposed method are not obvious. In other words, the proposed method can still achieve good correction results even with larger keyframe interval.

Figs. 16–18 show the mean and standard deviation comparisons of Y channel for '*Flamenco1*', '*Objects2*' and '*Race1*', respectively. Here, the keyframe interval is 15. As can be seen from the results, the means and standard deviations of the corrected images with the proposed method are basically fitted to those of the frame-by-frame method. However, for the frame-by-frame method, it is applied independently to each frame, which may lead to discontinuous variations between frames. While for the proposed method, smooth variations between subsequent frames can be obtained by using temporal variation models, which can reflect the actual variation trends of the original view.

Figs. 19–21(a) and (b) show the reference image and original image in the view 1 and view 2 at time 559 of '*Flamenco1*', '*Objects2*' and '*Race1*'. Clearly, the color consistency among the views is poor. Thus, the color correction is necessary if the multi-view images will be used to render new virtual arbitrary view. Figs. 19–21(c) show the corrected images with histogram matching (HM) method presented in [13] and Figs. 19–21(d) show the corrected images with the proposed method. Although the computa-

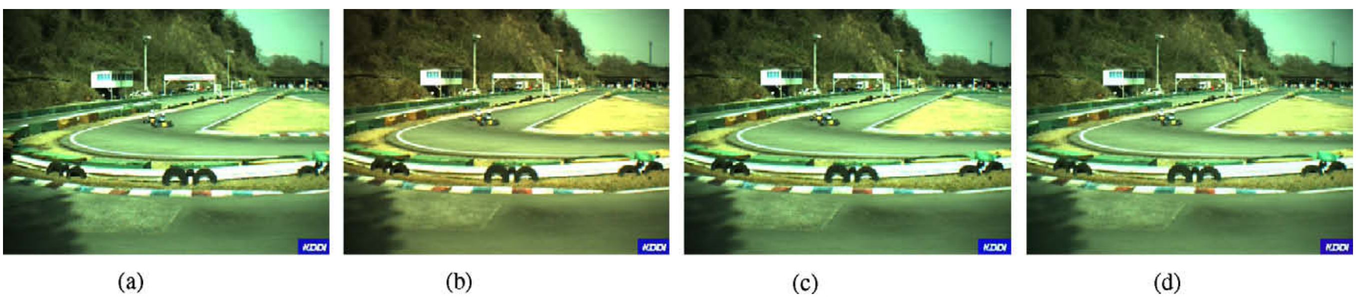


Fig. 21. Color correction results of '*Race1*'. (a) Reference image (view 1, frame #559); (b) current image (view 2, frame #559); (c) the corrected image with HM method in [13]; (d) the corrected image with the proposed scheme.

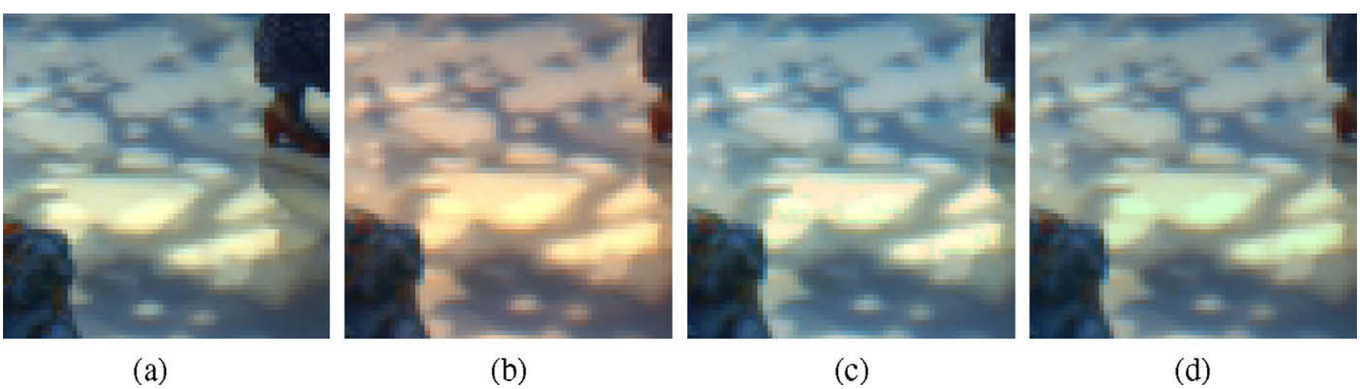


Fig. 22. Enlarged part signified in Fig. 19. (a) Reference image; (b) current image; (c) the corrected image with HM method; (d) the corrected image with the proposed scheme.

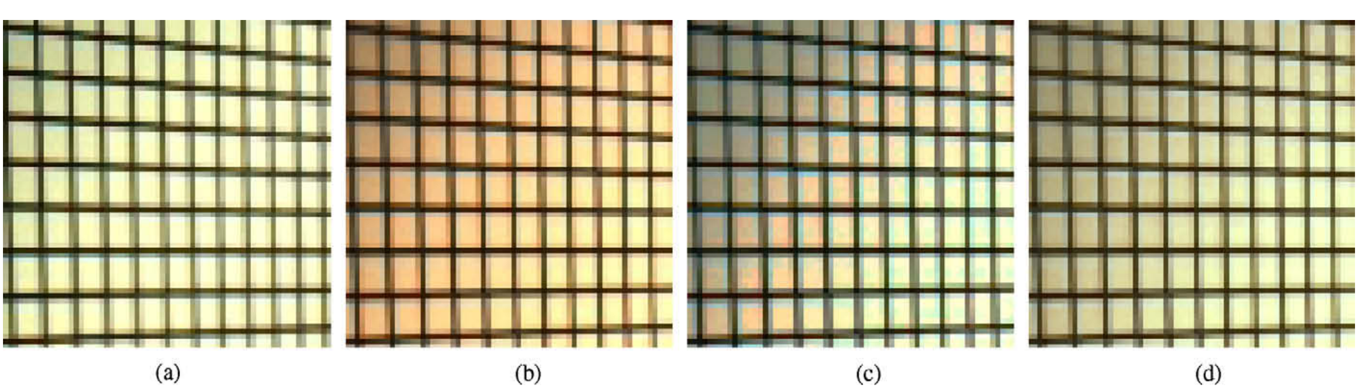
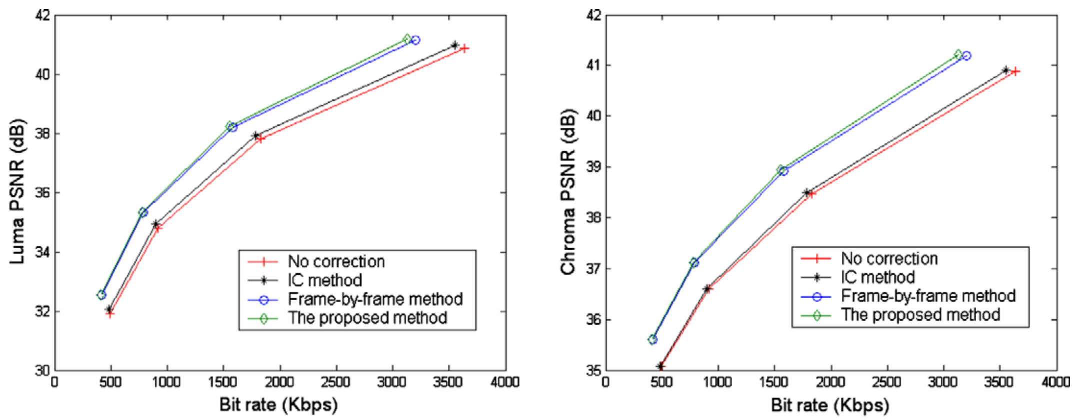


Fig. 23. Enlarged part signified in Fig. 20. (a) reference image; (b) current image; (c) the corrected image with HM method; (d) the corrected image with the proposed scheme.

tional complexity of the HM method is low, content variation between the current image and the reference image will inevitable to cause problem in the histogram related algorithm. We enlarge the part signified in Figs. 19 and 20, and show the detail examples in Figs. 22 and 23. It can be seen more clearly that compared to HM method, the proposed method can not only correct the color of current image to that of reference one, but also preserve the structural and contrast information of original image much better. This is because the proposed method attempts to model the relationship between two data sets that preserve the contrast and structure to be corrected.

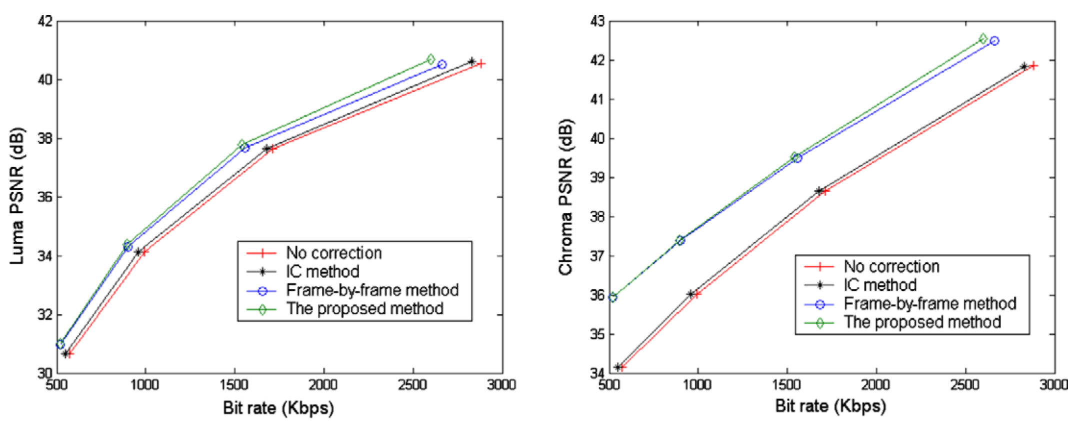
5.3. Coding performance comparison

Then, we compare our method with (1) compressing the original video; (2) the illumination compensation (IC) method adopted



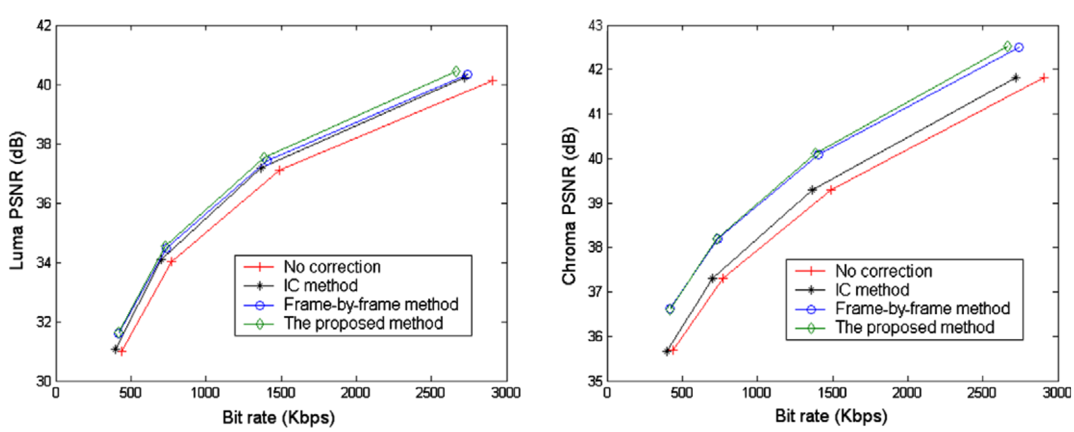
(a) Luma channel (b) Chroma channel

Fig. 24. Coding performance comparisons of '*Flamenco1*'.



(a) Luma channel (b) Chroma channel

Fig. 25. Coding performance comparisons of '*Objects2*'.



(a) Luma channel (b) Chroma channel

Fig. 26. Coding performance comparisons of '*Race1*'.

in JMVM; (3) the frame-by-frame method. Figs. 24–26 show the rate-distortion performance comparisons of '*Flamenco1*', '*Objects2*' and '*Race1*' for all eight views, respectively. The vertical axis in each sub-figure is the PSNR of luma or chroma channel, while the horizontal axis corresponds to the sum of the bitrates used for encoding. The Y component carries the luma channel information, and the chroma PSNR is the average PSNR of the U and V components. As can be seen from the results, the proposed method results in larger PSNR gains compared to compressing the original data with or without IC, especially in chroma channel. Besides, the coding performances of the proposed method are higher than the frame-by-frame method at high bitrate side. The reason lies in that the frame-by-frame method can result in discontinuous variations between adjacent time instants, while the proposed method can eliminate those discontinuous variations by modeling the temporal variations, which increases the temporal correlations in MVC.

6. Conclusions

Color correction is an important issue for multi-view video coding and virtual view synthesis in three-dimensional television and free viewpoint video applications. In this paper, a fast color correction method for multi-view video is proposed by modeling spatio-temporal variation. Experimental results show that the proposed method can achieve better performance in computational complexity, correction accuracy and coding efficiency. In the proposed method, in order to evaluate the correction performance, the corrected images with frame-by-frame method are assumed as reference. In fact, the reference itself is not right. Therefore, how to objectively evaluate the performance of the proposed method is an important problem to be settled. Currently, the proposed method corrects the color information based on a set of fixed keyframes, leaving no adaptability to video content. In future work, we will research on intelligent algorithms that automatically understand the video content to find keyframes.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant 60872094, 60832003, 60902096), 863 Project of China (2009AA01Z327), the Program for New Century Excellent Talents in University (NCET-06-0537), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20093305120002), and the Open Project Program of the State Key Lab of CAD&CG (Grant A0904), Zhejiang University. It was also sponsored by K.C. Wong Magna Fund in Ningbo University.

References

[1] Y. Morvan, D. Farin, P.H.N. de With, System architecture for free-viewpoint video and 3D-TV, IEEE Transactions on Consumer Electronics 54 (2) (2008) 925–932.
[2] A. Kubota, A. Smolic, M. Magnor, et al., Multiview imaging and 3DTV, IEEE Signal Processing Magazine 24 (6) (2007) 10–21.
[3] P. Merkle, A. Smolic, K. Muller, et al., Efficient prediction structures for multiview video coding, IEEE Transactions on Circuits and Systems for Video Technology 17 (11) (2007) 1461–1473.
[4] K. Muller, A. Smolic, K. Dix, et al., View synthesis for advanced 3D video system, EURASIP Journal on Image and Video Processing 2008, Article ID 438148, pp. 11, doi:10.1155/2008/438148.
[5] Y. Chen, Y. K. Wang, K. Ugur, et al. The emerging MVC standard for 3D video services, EURASIP Journal on Advances in Signal Processing, vol. 2009, Article ID 786015, pp. 13, doi:10.1155/2009/786015.
[6] Yun Zhang, Gangyi Jiang, Mei Yu, et al., Adaptive multiview video coding scheme based on spatiotemporal correlation analyses, ETRI Journal 31 (2) (2009) 151–161.
[7] T. Wiegand, G.J. Sullivan, G. Bjntegaard, et al., IEEE Transactions on Circuits and Systems for Video Technology 13 (7) (2003) 560–576.
[8] K. Sohn, Y. Kim, J. Seo, et al., H.264/AVC-compatible multi-view video coding, ISO/IEC JTC1/SC29/WG11, M12874 Bangkok, Thailand, Jan. 2006.
[9] J.H. Hur, S. Cho, Y.L. Lee, Adaptive local illumination change compensation method for H.264/AVC-based multiview video coding, IEEE Transactions on Circuits and Systems for Video Technology 17 (11) (2007) 1496–1505.
[10] S.H. Lee, J.H. Choi, Design and implementation of color correction system for images captured by digital camera, IEEE Transactions on Consumer Electronics 54 (2) (2008) 268–276.
[11] K. Yamamoto, M. Kitahara, H. Kimata, et al., Multiview video coding using view interpolation and color correction, IEEE Transactions on Circuits and Systems for Video Technology 17 (11) (2007) 1436–1449.
[12] Y. Chen, J. Chen, C. Cai. Luminance and chrominance correction for multi-view video using simplified color error model, in: Proc. of Picture Coding Symposium, Beijing, China, April 2006.
[13] U. Fecker, M. Barkowsky, A. Kaup, Histogram-based prefiltering for luminance and chrominance compensation of multiview video, IEEE Transactions on Circuits and Systems for Video Technology 18 (8) (2008) 1258–1267.
[14] Feng Shao, Gangyi Jiang, Mei Yu, Multi-view video color correction using dynamic programming, Journal of System Engineering and Electronics 19 (6) (2008) 1115–1120.
[15] M. Zhang, N.D. Georganas, Fast color correction using principal regions mapping in different color spaces, Real-Time Imaging 10 (1) (2004) 23–30.
[16] Feng Shao, Gangyi Jiang, Mei Yu, et al. A content-adaptive multi-view video color correction algorithm, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 969–972, 2007.
[17] T. Lanrence, A. Brian, Color constancy: a method for recovering surface spectral reflectance, Journal of Optical Society of America (A) 3 (1) (1986) 29–33.
[18] M. Tanimoto, T. Fujii, K. Suzuki. Available Technologies for FTV, ISO/IEC JTC1/SC29/WG11, M15088 Antalya, Turkey, January 2008.
[19] A. Agarwala, A. Hertzmann, D.H. Salesin, et al., Keyframe-based tracking for rotoscoping and animation, ACM Transactions on Graphics 23 (3) (2004) 584–591.
[20] D.C. Montgomery, E.A. Peck, G.G. Vining, Introduction to Linear Regression Analysis, Wiley/InterScience Press, New York, 2006. July.
[21] C. Yeo, J.J. Wang, K. Ramchandran, Robust distributed multi-view video compression for wireless camera networks, in Proc. of SPIE: Visual Communications and Image Processing, vol. 6508, p. 65080P, 2007.
[22] D. Ascher, N.M. Grzywacz, A Bayesian model of temporal frequency masking, Vision Research 40 (16) (2000) 2219–2232.
[23] S.K. Sahu, K.V. Mardia, A Bayesian kriged Kalman model for short-term forecasting of air pollution levels, Journal of the Royal Statistical Society: Series C (Applied Statistics) 54 (1) (2005) 223–244.
[24] R. Kawada, KDDI multi-view video sequences for MPEG 3DAV use, ISO/IEC JTC1/SC29/WG11, M10533 Munich, Germany, 2004.
[25] A. Vetro, P. Pandit, H. Kimata, et al., Joint multiview video model (JMVM) 7.0, ISO/IEC JTC1/SC29/WG11, JVT-Z207, Antalya, Turkey, Jan. 2008.
[26] Z. Wang, L. Lu, A.C. Bovik, Video quality assessment based on structural distortion measurement, Signal Processing: Image Communication 19 (2) (2004) 121–132.