

New Color Correction Method of Multi-view Images for View Rendering in Free-viewpoint Television

FENG SHAO, GANGYI JIANG and MEI YU
Faculty of Information Science and Engineering
Ningbo University
315211 Ningbo
CHINA

shaofeng@nbu.edu.cn yumei@nbu.edu.cn jianggangyi@nbu.edu.cn

Abstract: - Color inconsistency between views is an important problem to be solved in multi-view video imaging application, such as free viewpoint television. Up to now, some color correction methods have been proposed mainly for consistent color appearance or high coding efficiency, in which the mean, variance, or covariance information are used to transfer color information. In this paper, by using color restoration and linear regression technique, a new color correction method of multi-view images is proposed for view rendering. We first separate foreground and background from scene by mean-removed disparity estimation. Then color restoration is used to reconstruct the original color information for foreground and background. And then linear regression technique is used to estimate correction parameters for color restored image. Finally, color correction and view rendering between reference image and color corrected image are implemented. Experimental results show that the proposed method can achieve better performances of color correction and view rendering.

Key-Words: - Multi-view image, free viewpoint television, color correction, view rendering, expectation-maximization, linear regression, CIEDE2000.

1 Introduction

Free viewpoint television (FTV) is a new type of natural video media that expand the user's sensation far beyond what is offered by traditional media. The user can choose an own viewpoint and viewing direction within a visual scene, meaning interactive free viewpoint experience. Although FTV is capable to provide an exciting viewing experience, it is challenging to put it into practical applications. Multi-camera capturing system for FTV might not be perfectly calibrated. Although some standard geometric calibration methods exist for calibrating array of cameras^[3], much less attention has been paid to color correction for multiple cameras^[4-5]. Joshi et al. proposed an iterative scheme to calibrate the sensors to each other^[6]. This yields better color consistency between cameras, which is more suitable for multi-view imaging applications. In practical imaging, camera parameters in multi-camera capturing system may be inconsistent, and exposure or focus may be variable for different views. The heterogeneous cameras can lead to global or local mismatches across different views when virtual views are synthesized at the client of

FTV system. In addition, it is often impossible to capture an object under perfectly constant lighting conditions at different spatial positions within an imaging environment. Those variations provide serious challenge for realization of FTV system, and degrade the performance of subsequent multi-view video coding (MVC)^[7] or virtual view rendering^[8].

For CCD or CMOS sensor camera arrays, it is difficult to guarantee color consistency for all cameras when capture the same object. Color inconsistency across cameras may cause incorrect view-dependent color variation during virtual view rendering. The important issue in color correction is how to measure the accurate correspondence points between source and reference images. Yamamoto et al. proposed to obtain color correspondence map with color pattern board^[9] or without color pattern board^[10] (detecting the correspondences by scale invariant feature transform). While the available color correction methods are independent of other operations (such as MVC or view rendering), even though illumination compensation methods have been proposed to improve coding efficiency^[11].

Color correction directly for view rendering application has not been studied in the literature.

Many foreground-background separation methods had been proposed for monoview video, mainly based on object motion activity. Kyungnam et al. proposed real-time foreground-background segmentation using codebook to represent background model^[12]. Chen, et al. combined pixel-based and block-based representation to build hierarchical backgrounds^[13]. Lee used adaptive Gaussian mixture to model nonstationary distributions of pixel in video^[14]. However, foreground-background separation in multi-view is different with the traditional monoview methods, mainly based on the scene distance from imaging plane.

In this paper, an effective color correction method of multi-view images is proposed for view rendering. The rest of paper is organized as follows: Section II describes how to separate foreground-background from images. In Section III, an effective color correction method of multi-view images is proposed, and an objective performance metric is given. Experimental results about color correction, view rendering and MVC are shown in Section IV. Finally, the conclusions are given and future work is suggested.

2 Foreground-background Separation

As we have known that color image is the result of a complex reflection between three components of the optical properties of the scene, the illuminant source and the sensor^[15], which is described as

$$I_k(x, y) = \int_{\omega} R_k(\lambda) E(x, y, \lambda) S(x, y, \lambda) d\lambda \quad (1)$$

where x and y are the coordinates of a pixel in an image, subscript k indicates a type of color sensor of the camera, integral range ω is visible spectrum, R_k is spectral sensitivity function of the k -th sensor, S is a spectral surface reflectance of an object, and E is a spectral power distribution of a illuminant source.

In multi-view imaging, it is often impossible to capture an object under perfectly constant lighting conditions. When the scene depth is changed, the illumination intensity will be changed accordingly. In addition, Lambertian reflectance assumption is not always satisfied in the scene. Under the Lambertian reflectance assumption, the illuminated region of the surface emits the entire light equally in all direction. Therefore, the imaging is inconsistent in different scene depth even for identical objects. It

is necessary to compensate the color change in different depth. In the simplest strategy, depth is divided into same intervals as shown in Fig.1(a). However, the same region in an object may be classified into different depth level, increasing the classification error. And, the linear relation is not always satisfied between scene depth and the scene distance from camera array. For this reason, we use foreground and background for depth division, as shown in Fig.1(b).

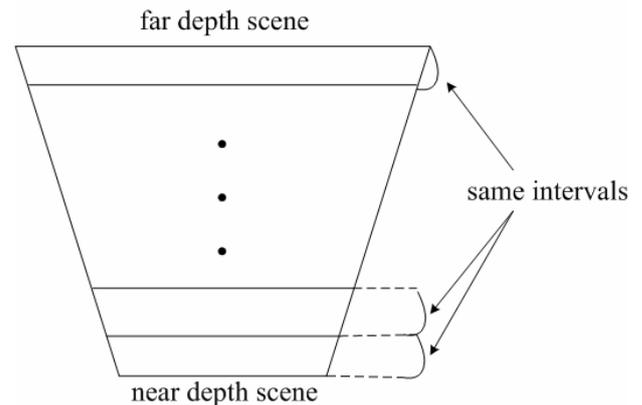


Fig.1(a): depth separation into same interval

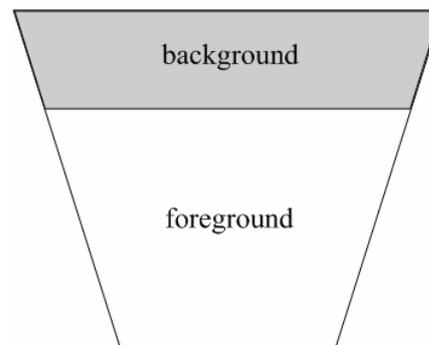


Fig.1(b): foreground-background separation

In this paper, a depth-based foreground-background separation method is proposed. In multi-view imaging, the object depth can be estimated from the disparity between two cameras. Thus disparity vectors provide additional clues for depth estimation of scene. Supposed $(k=0)$ and $(k=1)$ denote foreground and background class for scene, respectively. And $p(k|d(x,y))$ denotes the probability that a pixel (x,y) belongs to foreground ($k=0$) or background ($k=1$). By defining a threshold T , a block belongs to background regions with $p(k=1|d(x,y))=1$, if horizontal and vertical disparities satisfy $(d_x^2 + d_y^2)^{1/2} < T$. Otherwise, the block belongs to foreground regions with $p(k=0|d(x,y))=1$. Thus coarse results of foreground-background separation are obtained. While some blocks in the background

or foreground may be isolated because of occlusion, exposure or other reasons. In order to obtain continuous background contour, we propose a smoothing mechanism with the correlation of adjacent blocks. If a current block belongs to foreground or background, $p(k|d(x,y))=1$, for the adjacent left, top, right and bottom blocks, if at least three blocks belong to a reverse classification, $\sum_{(x',y') \in N} p(k|d(x',y')) \leq 1$, we determine that the current block also belongs the reverse classification. Here, N denotes four neighbourhood of (x,y) .

For one pixel (x,y) and its corresponding disparity vector $d(x,y)$, we adopted expectation-maximization (EM) algorithm based on the Gaussian mixture model to refine the probability belonging to the foreground or background region^[16]. Let $\theta_k = (\mu_k, \sum_k)$ denote mean and variance of Gaussian distribution, and $p_k(d(x,y)|\theta_k)$ denote probability density function of disparity vector $d(x,y)$ with θ_k . Then the probability that $d(x,y)$ belongs to the foreground ($k=0$) or background ($k=1$) is calculated by

$$p(k|d(x,y),\theta_k) = \frac{p_k(d(x,y)|\theta_k)}{\sum_{j=1}^2 p_j(d(x,y)|\theta_k)} \quad (2)$$

3 Color Correction of Multi-view Images for View Rendering

Fig.2 shows a FTV reference model^[17]. On the sender side, multi-view images are captured by 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 and multi-view disparities are compressed for transmission and storage by the encoder. On the receiver side, the multi-view images and multi-view disparities are reconstructed by the decoder. Free viewpoint images are generated by interpolating the reconstructed images using multi-view disparity information and displayed on a 2D/3D display. Therefore, color correction is a very important process in FTV system.

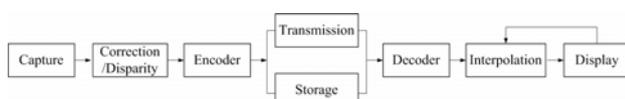


Fig.2: FTV reference model

The block diagram of the proposed color correction method is given in Fig.3. First, disparity estimation with mean-removed sum of absolute differences (MRSAD) is used to separate the foreground and background from scene. Then color restoration is performed to reconstruct original color information for foreground and background, respectively. And then linear regression technique is used to estimate the correction parameters for color restored image. Finally, color correction for color restored image and view rendering between reference image and color corrected image are implemented.

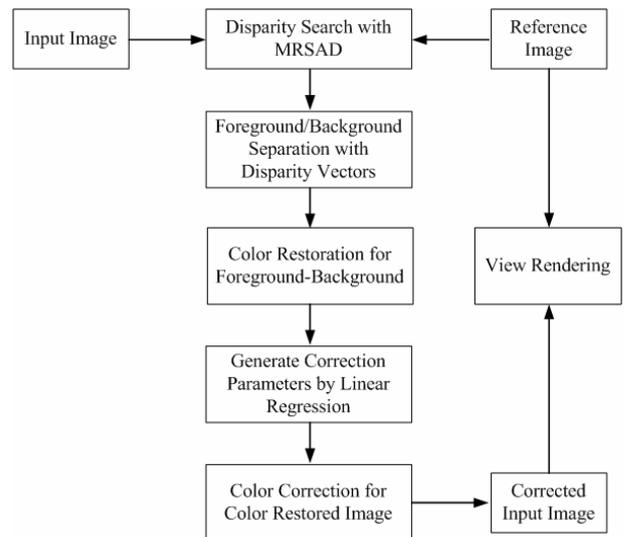


Fig.3: Block diagram of the new color correction method

3.1 Color restoration

The conventional matching metric for disparity estimation is SAD calculation of $S \times T$ blocks. In order to compensate the illumination change, a mean-removed sum of absolute differences (MRSAD) metric^[18] is defined. MRSAD produces block correspondence with the best matched patterns after mean removal. Therefore, disparity vector for each candidate block can be well preserved even with serious cross-view illumination mismatch. And MRSAD is calculated by

$$MRSAD(x,y) = \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} |(S(i,j) - \mu_s) - (R(i+x,j+y) - \mu_r)| \quad (3)$$

where $S(i,j)$ and $R(i,j)$ are pixel values for source image and reference image with spatial coordinates (i,j) , respectively. μ_s and μ_r are the average values for all the pixels in the source block and the reference block, respectively.

Color restoration is the process of obtaining the original color information from the degraded image given the knowledge of the degrading factor^[19]. Degradation of images can have many cases: defects of optical lenses, non-linearity of the electro-optical sensor, motion blurriness, noise, or illumination intensity discrepancy, etc^[20]. Supposing reference image $R(x,y)$ in one view as benchmark image, current source image $S(x,y)$ in another view as degradation image, the degradation model can be expressed as

$$I(x, y) = H(x, y) * S(x, y) = R(x + dv_x, y + dv_y) \quad (4)$$

here $I(x,y)$ denotes real image for $S(x,y)$, (x,y) denotes the pixel position, and $(x+dv_x, y+dv_y)$ is its corresponding disparity displaced pixel. If we have known the response function $H(x,y)$, color restoration equates to inverse convolution (also call deconvolution). Many techniques, including neural network and Kalman filtering, had been used to achieve color restoration^[21,22]. In our approach, we apply the linear filtering technique to restore the degraded image as

$$R(x + dv_x, y + dv_y) = \sum_{i,j} h_{i,j} S(x + i, y + j) \quad (5)$$

here $h_{i,j}$ is the filter coefficients of a point spread function H . The current source image $S(x,y)$ is filtered by an estimator of H chosen to minimize the error with respect to the reference image $R(x+dv_x, y+dv_y)$. Minimum mean-squared error estimation can be derived by optimizing the following:

$$\min_H \sum_{x,y} (R(x + dv_x, y + dv_y) - H * S(x, y))^2 \quad (6)$$

The filter coefficients $h_{i,j}$ can be determined by solving the Wiener-Hopf equations as:

$$\sum_{j=-n}^n \sum_{i=-m}^m \left(h_{i,j} \sum_{(x,y) \in D_k} S(x+i, y+j) S(x+I, y+J) \right) = \sum_{(x,y) \in D_k} R(x + dv_x, y + dv_y) S(x+I, y+J) \quad (7)$$

where $-m \leq I \leq m$, $-n \leq J \leq n$, respect to each coefficient. This can be represented as a full matrix. For foreground or background, a filter matrix will be obtained as a solution to (7).

Finally, considering the probability $p_k(x,y)$, the color restoration can be described as

$$I(x, y) = \sum_k p_k(x, y) \cdot \left(\sum_{i,j} h_{i,j}^k S(x + j, y + j) \right) \quad (8)$$

3.2 Linear regression

Even though accurate color restoration can be achieved, some color-changing problem, such as spectral reflectance characteristics of vision system components, cannot be solved completely. Therefore, additional color correction must be performed after color restoration.

Regression analysis is a technique for using data to identify relationship between two sets of data and using these relationships to make predictions. The simplest form of regression, linear regression^[23], finds a linear relationship between two sets of data, which minimizes the sum of squared errors. Supposed $(I_1(x,y), I_2(x,y), I_3(x,y))$ is three color components of a pixel in input image and $(R_1(x+d_x, y+d_y), R_2(x+d_x, y+d_y), R_3(x+d_x, y+d_y))$ is the corresponding pixel in reference image. The linear regression model is of the form

$$\begin{aligned} R_1(x + d_x, y + d_y) &= b_1 + a_{11}I_1(x, y) + a_{12}I_2(x, y) + a_{13}I_3(x, y) \\ R_2(x + d_x, y + d_y) &= b_2 + a_{21}I_1(x, y) + a_{22}I_2(x, y) + a_{23}I_3(x, y) \\ R_3(x + d_x, y + d_y) &= b_3 + a_{31}I_1(x, y) + a_{32}I_2(x, y) + a_{33}I_3(x, y) \end{aligned} \quad (9)$$

Supposed

$$\mathbf{Y} = \begin{bmatrix} R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ \vdots & \vdots & \vdots \\ R_{1m} & R_{2m} & R_{3m} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & I_{11} & I_{21} & I_{31} \\ 1 & I_{12} & I_{22} & I_{32} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & I_{1m} & I_{2m} & I_{3m} \end{bmatrix}$$

$$\text{and } \mathbf{C} = \begin{bmatrix} b_1 & b_2 & b_3 \\ a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

where m is the number of terms in the regression model. The coefficient matrix \mathbf{C} , which minimizes the residuals in the least-squares sense, is represented by

$$\mathbf{C} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (10)$$

In the proposed color correction method, the number of terms in the regression model is just the number of the matching blocks, and the means of the corresponding blocks are used as elements in \mathbf{Y} and \mathbf{X} .

Finally, for each pixel in color restored image, we use the correction matrix to get the final corrected image.

3.3 Objective performance evaluations

In order to objectively evaluate the performance of color correction method, the color differences

between reference image and color corrected image are calculated. Two color difference definitions are used. The color differences in the CIELAB space are defined as

$$\Delta E_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (11)$$

The color differences are also calculated with a quite complicated CIEDE2000 formula^[24], because CIELAB space is not uniform enough. The CIEDE2000 color difference ΔE_{00} formula is defined as

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{k_C S_C}\right) \left(\frac{\Delta H'}{k_H S_H}\right)} \quad (12)$$

here $k_L, k_C, k_H, S_L, S_C, S_H, R_T$ are weighting factors. The detail definition of the equation is described in appendix.

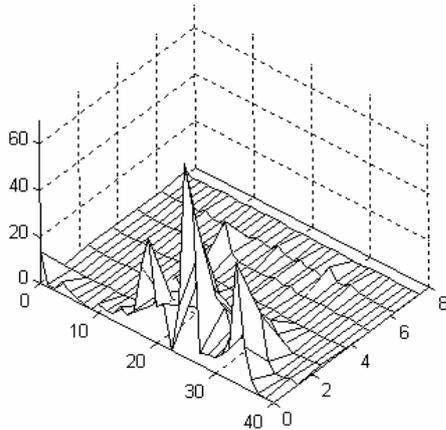


Fig.5(a): Disparity vectors from view 2 to view 1 at the 570 frame in ‘flamenco1’

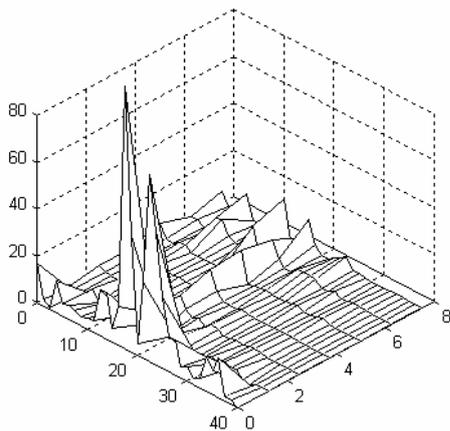


Fig.5(b): Disparity vectors from view 2 to view 1 at the 245 frame in ‘objects2’

4 Experimental Results and Analyses

In experiments, multi-view images sequences ‘flamenco1’ and ‘objects2’, provided by KDDI Corp., are used as the test sequences. The size of

‘flamenco1’ and ‘objects2’ is 320×240, and the images are taken by a horizontal parallel camera configuration with eight viewpoints and 200mm camera interval. Fig.4 shows eight original viewpoint images of the ‘flamenco1’ at the 570th frame and ‘objects2’ at the 245th frame. Clearly, the color consistency among these eight original viewpoint images is poor. Thus, the color correction is necessary if these multi-view images will be used to render new virtual arbitrary viewpoint image.



Fig.4 Eight original viewpoint images of ‘flamenco1’ and ‘objects2’.

Fig.5 shows the disparity vectors from view 2 to view 1 in ‘flamenco1’ and ‘objects2’ multi-view image sequences. The peaks of the histogram in Fig.5 correspond to different depth levels. This accounts for that disparity vectors can provide additional clues for depth estimation of scene. In our classification result, only foreground or background regions are used to represent all depth levels, and blocks classified into a certain level will be associated with different filter matrix and coefficient matrix.

Figs.6-7(a) and (b) show the reference image and input image of ‘flamenco1’ at the 570th frame and ‘objects2’ at the 245th frame in the first and second viewpoints, respectively. Figs.6-7(c) show the horizontal disparity images, which the block size is set as 8, the maximum horizontal and vertical disparities are set as 40 and 8, respectively. Figs.6-7(d) show the final foreground-background separation results with $T=25$ and $T=20$ for ‘flamenco1’ and ‘objects2’, respectively. Because the human eyes are less sensitive to chrominance than luminance, based on the same disparity vectors with luminance, color correction for chrominance is also performed. In the proposed method, we use 5×5 filters in color restoration, with 25 coefficients to be solved in Wiener-Hopf equations (7). The filter matrices of ‘objects’ in foreground and background

are
$$\begin{bmatrix} -0.021 & -0.003 & -0.010 & -0.024 & 0.028 \\ 0.053 & -0.000 & 0.149 & 0.011 & 0.018 \\ 0.014 & 0.066 & 0.514 & 0.108 & -0.049 \\ -0.033 & -0.007 & -0.006 & -0.014 & 0.001 \\ 0.031 & -0.001 & 0.042 & -0.026 & -0.008 \end{bmatrix}$$
 and

$$\begin{bmatrix} 0.053 & 0.021 & 0.041 & -0.048 & -0.042 \\ -0.008 & -0.031 & 0.211 & 0.004 & 0.030 \\ -0.040 & 0.060 & 0.600 & -0.017 & 0.086 \\ -0.018 & 0.005 & 0.029 & -0.053 & 0.000 \\ 0.045 & -0.003 & 0.017 & 0.004 & -0.010 \end{bmatrix}, \text{ respectively.}$$

Obviously, the filter matrices for the foreground and background are inconsistent, which explains the fact that the imaging is different in the foreground and background. The corrected images after color restoration in Figs.6-7(e) have comparatively consistent color appearance with the reference images, while the floor in ‘flamenco1’ and the wall in ‘objects2’ are obviously inconsistent. The reason is that color restoration is likely to fail in these high-light regions due to insufficient information in calculating surface reflectance, lead to loss of the physical properties. With the proposed method, the corrected images, as shown in Figs.6-7(f), can overcome this shortcoming with full consistence with their reference image in color appearance.

In order to objectively evaluate color correction performance, we calculate the color difference ΔE_{ab} and ΔE_{00} between reference image and input image, and compare with the color difference between the reference image and the corrected image. For the color corrected image, there is no original signal for evaluation method, therefore, the average CIELAB values are first calculated, then the color difference between the average CIELAB values of reference image and corrected image are calculated. Table.1 shows the color difference comparison results without correction, with color restoration and with the proposed method. From the table, it is noted that color correction with color restoration can achieve smaller color difference, and the proposed method can further reduce the color difference compared with color restoration method, which is consistent with the subjective color appearance evaluation in Figs.6-7(e) and (f).

Although the proposed color correction method primarily aimed at improving the performance of view rendering, they can easily be used to compensate illumination mismatches in cross-view prediction in MVC. Using the reference codec JMVM 5.0, we encode 300 frames multi-view video in four different basicQP values (22, 27, 32, 37). Figs.8-9 demonstrate coding results of second viewpoint with P-frame prediction for cross-view prediction and hierarchical B for temporal prediction. For ‘flamenco1’, color correction can achieve 0.3dB gain for luminance component Y, almost consistent performance for chrominance



Fig.6(a): reference image of ‘flamenco1’

Fig.6(b): input image of ‘flamenco1’



Fig.6(c): horizontal disparity image

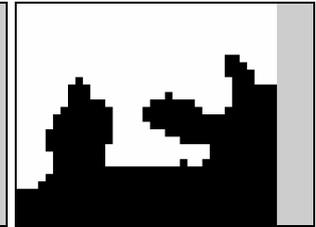


Fig.6(d): foreground-background separation result



Fig.6(e): corrected image after color restoration



Fig.6(f): corrected image with the proposed method



Fig.7(a): reference image of ‘objects2’

Fig.7(b): input image of ‘objects2’



Fig.7(c): horizontal disparity image

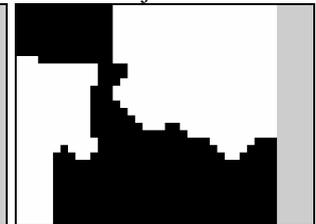


Fig.7(d): foreground-background separation result



Fig.7(e): corrected image with color restoration



Fig.7(f): corrected image with the proposed method

Table 1: Objective color difference comparison results

Test images \ methods	without correction		correction with color restoration		the proposed method	
	ΔE_{ab}	ΔE_{00}	ΔE_{ab}	ΔE_{00}	ΔE_{ab}	ΔE_{00}
flamenco1	0.662	0.807	0.139	0.133	0.089	0.079
objects2	0.871	1.152	0.333	0.362	0.151	0.153

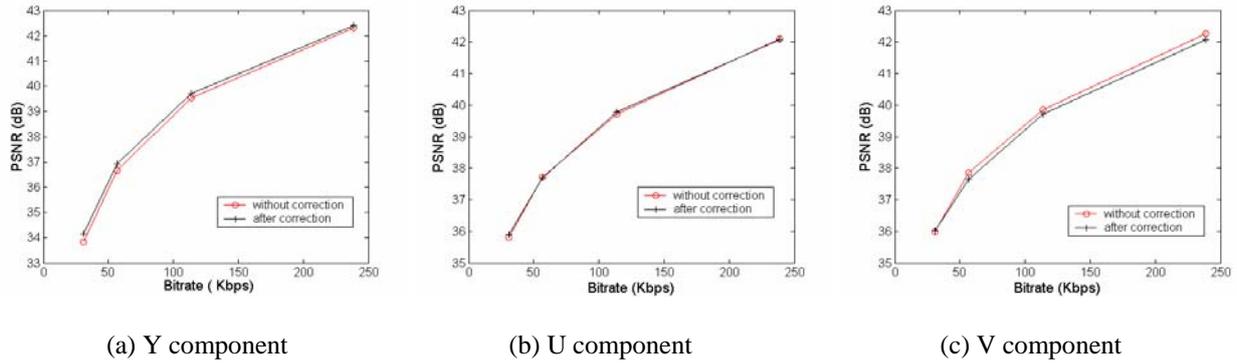


Fig.8 Rate-distortion performance comparison of ‘flamenco1’

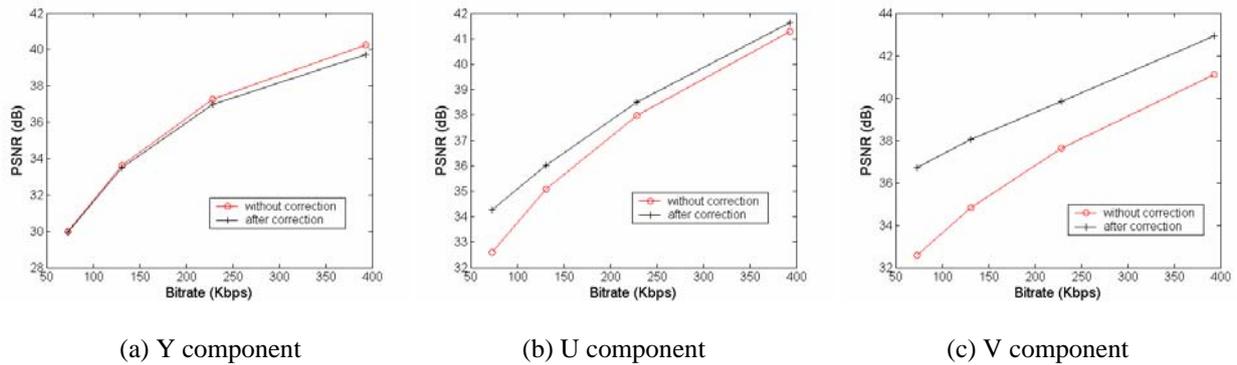


Fig.9 Rate-distortion performance comparison of ‘objects2’

component U, and 0.1dB decrease for chrominance component V. For ‘objects2’, even though the coding performance is reduced for Y component, large gains are obtained for U and V components.

However, ‘flamenco1’ and ‘objects2’ test sets are not geometric calibration, which are not suitable for effective view rendering. In order to quantitatively evaluate the performance of view rendering, ‘cup’ multi-view images, captured by our multi-camera equipment with image size 640×480, are used. The cameras are precisely controlled in the horizontal direction. The distance between neighbouring cameras is 1mm and 120 images with the scene were taken to simulate the densely sampling case. Fig.10(a)-(c) show successive three

viewpoint images, as reference image, middle image and input image, respectively. For some unpredictable internal or external factors, the color appearance in one view may be consistent with others. Fig.10(d) shows the color deflection input image by manual adjustment. Fig.10(e) shows the rendering middle image between reference image and color deflection input image. There is obvious difference in their color appearance between Fig.10(e) and Fig.10(b). Then color correction is performed for the color deflection input image. Fig.10(f) shows the color corrected result with the proposed method. The middle view rendering result between reference image and color corrected image is shown in Fig.10(g). It is obvious that color

appearance is essentially consistent with original middle image in Fig.10(b). We measured the following $PSNR_{rgb}$ between the original middle image in Fig.9(b) and the rendering image in Fig.9(e) or (g) by

$$PSNR_{rgb} = \frac{1}{3}(PSNR_r + PSNR_g + PSNR_b) \quad (13)$$

where $PSNR_i, i \in \{r, g, b\}$ represents the PSNR of R, G, and B channels. The $PSNR_{rgb}$ is improved from 23.64dB to 34.99dB by the proposed method, which indicates that color correction can largely improve the performance of view rendering.



Fig.10(a): reference image of 'cup'



Fig.10(b): original middle image of 'cup'



Fig.10(c): input image of 'cup'



Fig.10(d): color deflection input image



Fig.10(e): rendering result between reference and color deflection input image



Fig.10(f): corrected image with the proposed method



Fig.10(g): rendering result between reference and corrected image

5 Conclusion

Color correction is an important issue for virtual view synthesis and multi-view video coding in FTV. In this paper, an effective multi-view images color correction method for view rendering is proposed. Foreground-background is first separated from scene based on disparity vectors and color restoration and linear regression technique is used to achieve color correction. Experimental results show the effectiveness of the proposed method in color correction and view rendering.

The existing color correction methods are based on the same assumption that the color values after correction are consistent with the reference. However, this assumption is not always satisfied during multi-view imaging. In future work, we will do further research on how to derive partial differential equations (PDE)^[25] as imaging model in color restoration.

Acknowledgment This work was supported by the Natural Science Foundation of China (grant 60672073), the Program for New Century Excellent Talents in University (NCET-06-0537), the Scientific Research Fund of Zhejiang Provincial Education Department (grant 20070962) and the Key Project of Chinese Ministry of Education (grant 206059).

References:

- [1] M. Tanimoto, Overview of free viewpoint television, *Signal Processing: Image Communication*, vol. 21, no. 6, 2006, pp. 454-461.
- [2] M. Tanimoto, T. Fujii, H. Kimata, et al, Proposal on requirements for FTV, *ISO/IEC JTC1/SC29/WG11*, Doc. M13612, Klagenfurt, Austria, July 2006.
- [3] Z. Y. Zhang, A flexible new technique for camera calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, 2002, pp. 1330-1334.
- [4] F. Shao, G. Y. Jiang, M. Yu, Color correction for multi-view images combined with PCA and ICA, *WSEAS Transactions on Biology and Biomedicine*, vol. 4, no. 6, June 2007, pp. 73-79.
- [5] F. Shao, G. Y. Jiang, M. Yu, et al, A content-adaptive multi-view video color correction algorithm, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 15-20, 2007, Honolulu, Hawaii, pp. 969-972.

- [6] N. Joshi, B. Wilburn, V. Vaish, M. Levoy, et al, Automatic color calibration for large camera arrays, UCSD CSE Tech. Rep. CS2005-0821, May 2005.
- [7] A. Smolic, K. Mueller, N. Stefanoski, et al, Coding algorithms for 3DTV- a survey, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, 2007, pp. 1606-1621.
- [8] M. Tanimoto, T. Fujii, K. Suzuki, Improvement of depth map estimation and view synthesis, *ISO/IEC JTC1/SC29/WG11*, Doc. M15090, Antalya, Turkey, January 2008.
- [9] K. Yamamoto, T. Yendo, T. Fujii, et al, Colour correction for multiple-camera system by using correspondences, *The Journal of the Institute of Image Information and Television Engineers*, vol. 61, no. 2, 2007, pp. 213-222.
- [10] 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*, vol. 17, no. 11, 2007, pp. 1436-1449.
- [11] J. H. Hur, S. Cho, Y. L. Lee, Adaptive Local illumination change compensation method for H.264/AVC multiview video coding, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, 2007, pp. 1496-1505.
- [12] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis. Real-time foreground-background segmentation using codebook model, *Real-time Imaging*, vol.11, no.3, pp. 172-185, 2005.
- [13] Y. T. Chen, C. S. Chen, C. R. Huang and Y. P. Hung, Efficient hierarchical method for background subtraction, *Pattern Recognition*, vol. 40, no. 10, pp. 2706-2715, 2007.
- [14] D. S. Lee, Effective Gaussian mixture learning for video background subtraction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27. no. 5, 2005, pp. 827-832.
- [15] G. Unal, A. Yezzi, S. Soatto, G. Slabaugh, A variational approach to problem in calibration of multiple cameras, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, 2007, pp. 1322-1338.
- [16] K. Y. Wong, M. E. Spetsakis, Motion segmentation by EM clustering of good features, in *Proceedings of IEEE Workshop on Image and Video Registration*, July 2004, Washington DC, pp. 166-173.
- [17] M. Tanimoto, T. Fujii, K. Suzuki, Available technologies for FTV, *ISO/IEC JTC1/SC29/WG11*, Doc. M15088, Antalya, Turkey, January 2008.
- [18] J. H. Kim, P. Lai, J. Lopez, et al, New coding tools for illumination and focus mismatch compensation in multiview video coding, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, 2007, pp. 1519-1535.
- [19] Y. Hatakeyama, K. Kawamoto, H. Nobuhara, et al, Color restoration algorithm for dynamic images under multiple luminance conditions using correction vectors, *Pattern Recognition Letters*, vol. 26, 2005, pp. 1304-1305.
- [20] M. Sonka, V. Hlavac, R. Bovle, Image processing, analysis and machine vision, Third Edition, Thomson Press, 2008, pp. 199-211.
- [21] Y. T. Zhou, A. Vaid, Image restoration using a neural network, *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 36, no. 7, 1988, pp. 1141-1151.
- [22] N. P. Galatsanos, R. T. Chin, Restoration of color images by multichannel Kalman filtering, *IEEE Transactions on Signal Processing*, vol. 39, no. 10, 1991, pp. 2237-2252.
- [23] D. C. Montgomery, E. A. Peck, G. G. Vining, Introduction to linear regression analysis, Wiley-Interscience Press, July 2006.
- [24] M. R. Luo, G. Cui, B. Rigg, The development of the CIE2000 colour-difference formula: CIEDE2000, *Color Research and Application*, vol. 26, no. 5, 2001, pp. 340-350.
- [25] M. Messaoud, F. Z. Nouri, Image restoration by partial differential equations, *Geometric Modeling and Imaging--New Trends*, July 2006, pp. 250-254.

Appendix

Given two CIELAB color values $\{L_i^*, a_i^*, b_i^*\}_{i=1}^2$ and parametric weighting factor K_L , K_C and K_H , the process of computation of the color difference is summarized the following equations, grouped as three main steps.

1. Calculate C'_i, h'_i .

$$C_{i,ab}^* = \sqrt{(a_i^*)^2 + (b_i^*)^2}$$

$$\bar{C}_{ab}^* = \frac{C_{1,ab}^* + C_{2,ab}^*}{2}$$

$$G = 0.5 \left(1 - \sqrt{\frac{\bar{C}_{ab}^{*7}}{\bar{C}_{ab}^{*7} + 25^7}} \right)$$

$$a_i' = (1 + G)a_i^*$$

$$C_i' = \sqrt{(a_i')^2 + (b_i')^2}$$

$$h_i' = \begin{cases} 0 & b_i^* = a_i' = 0 \\ \tan^{-1}(b_i^*, a_i') & \text{otherwise} \end{cases}$$

$$R_C = 2\sqrt{\frac{\bar{C}^{\prime 7}}{\bar{C}^{\prime 7} + 25^7}}$$

$$S_L = 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}}$$

$$S_C = 1 + 0.045\bar{C}'T$$

$$S_H = 1 + 0.015\bar{C}'T$$

$$R_T = -\sin(2\Delta\theta)R_C$$

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2} + R_T \left(\frac{\Delta C'}{k_C S_C}\right) \left(\frac{\Delta H'}{k_H S_H}\right)$$

2. Calculate $\Delta L'$, $\Delta C'$, $\Delta H'$:

$$\Delta L' = L_2^* - L_1^*$$

$$\Delta C' = C_2' - C_1'$$

$$\Delta h' = \begin{cases} 0 & C_1' C_2' = 0 \\ h_2' - h_1' & C_1' C_2' \neq 0; |h_2' - h_1'| \leq 180^\circ \\ (h_2' - h_1') - 360 & C_1' C_2' \neq 0; |h_2' - h_1'| > 180^\circ \\ (h_2' - h_1') + 360 & C_1' C_2' \neq 0; |h_2' - h_1'| < -180^\circ \end{cases}$$

$$\Delta H' = 2\sqrt{C_1' C_2'} \sin\left(\frac{\Delta h'}{2}\right)$$

3. Calculate CIEDE2000 Color Difference ΔE_{00} :

$$\bar{L}' = (L_1^* + L_2^*) / 2$$

$$\bar{C}' = (C_1^* + C_2^*) / 2$$

$$\bar{h}' = \begin{cases} \frac{h_1' + h_2'}{2} & |h_1' - h_2'| \leq 180^\circ; C_1' C_2' \neq 0 \\ \frac{h_1' + h_2' + 360}{2} & |h_1' - h_2'| > 180^\circ; |h_1' + h_2'| < 360^\circ; C_1' C_2' \neq 0 \\ \frac{h_1' + h_2' - 360}{2} & |h_1' - h_2'| > 180^\circ; |h_1' + h_2'| \geq 360^\circ; C_1' C_2' \neq 0 \\ h_1' + h_2' & C_1' C_2' = 0 \end{cases}$$

$$T = 1 - 0.17 \cos(\bar{h}' - 30^\circ) + 0.24 \cos(2\bar{h}')$$

$$+ 0.32 \cos(3\bar{h}' + 6^\circ) - 0.20 \cos(4\bar{h}' - 63^\circ)$$

$$\Delta\theta = 30 \exp\left\{-\left[\frac{\bar{h}' - 275^\circ}{25}\right]^2\right\}$$