

Color Adjustment Based on Support Vector Regression for Multi-View Video

Huadong Sun, Xuesong Jin, Zhipeng Fan, Lizhi Zhang and Qian Wu

*School of Computer and Information Engineering, Harbin University of
Commerce, Harbin, 150028, China
kof97_sun@163.com*

Abstract

Significant color discrepancies between the different camera views can be observed in multi-view video sequences. In this paper, a color adjustment algorithm based on support vector regression is proposed. A mapping function is established by extracted feature points from original-image and target-image. Then the mapping function is applied to the original image to obtain the corrected image. Experimental results show that the proposed method can produce good correction result. It also shows that the color differences between multi-view video can be effectively reduced by SVR.

Keywords: *Color adjustment, Multi-view video, Support vector regression, Disparity estimation*

1. Introduction

Multi-view video sequence consists of same scene with different views, which are attained by many cameras. These cameras follow specific arrangement. The key factor in the coding scheme of multi-view video is multi-view coding system JMVM [1, 2] based on H.264/AVC, which was presented by the united video team JVT (Joint Video Team). In the process of collecting multi-view videos, the color difference of different view of multi-view video sequence is unavoidable, because objective existence of the noise environment (different lighting condition) and the changes of hardware (radiational features of different cameras, also the different color they product). These differences will reduce the effect of decorrelation between different views, and then, affect the coding efficiency of multi-view video sequences. Figure 1 show the prediction structure of JMVM, a block of frame can be predicted by that of pre, post frame and adjoining view frames, the predicting residual value can be processed by the following procedure. Therefore, to enhance the performance of the successive processing procedure, like the compression of multi-view videos, *etc.*, we need to use suitable algorithm to correct the color difference between different views.

To reduce the color difference between views, the algorithms to adjust the color of views are built on the mapping relationship between objective images and original images, which is used to adjust the color of original images and to obtain the processed images. Based on the assumption about the similarity between original images and respectively objective images' histogram, Fecke [3] presented a color correction algorithm based on histogram matching. This algorithm calculates the histogram of original images and objective images on the parameters of Y, U, V, and then, calculates the accumulating histogram according to it. After that, the algorithm will establish a mapping function to correct color based on the accumulating histogram. The advantage of this algorithm is the low calculation complexity. But, in the condition of big color difference, the subject performance of this algorithm is

worse after the color correction. Chen and *etc.*, [4] presented a linear model used scaling and migration parameters, which are obtained by the calculating Y, U and V's histogram of original images and objective images. In [5], the paper presented a method to adjust the color by checking the workspace color table of R, G, and B. In [6], the paper presented a method to correct the difference by using cubic polynomial, which calculates the average color of multi-views images, firstly, and then, make parallax motion prediction on the block between every view and average color.

In this paper, we present a color correction algorithm based on support vector regression. The algorithm uses the macro-block in the objective video frames and original video frames to build the training sets, which support vectors, and trains regression model in the CIELAB color workspace.

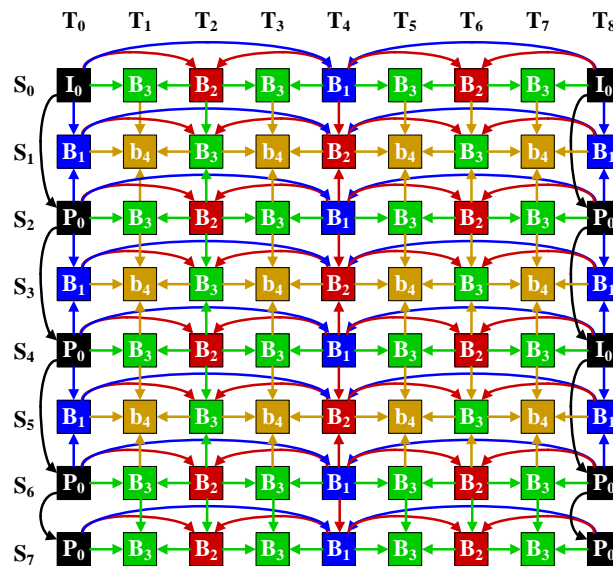


Figure 1. Inter-views Time Prediction Structure, in JMVM View

2. Support Vector Regression

Support vector machine (SVM) is a machine learning method based on the principles of statistics, which is suitable to the small samples training [7]. The Support vector regression (SVR) is an application of SVM in the regression learning.

The basic ideas of SVR and SVM are the same. The purpose of SVR is to look for the best plane in the sample space, which can be used to predict the sample distribution precisely, but, the SVM is to look for the plane which can be used to divide the sample space. There is a set of experimental data:

$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_i, y_i), \dots, (\mathbf{x}_n, y_n)\}, \quad (\mathbf{x}_i, y_i) \in \mathfrak{R}^n \times \mathfrak{R} \quad (1)$$

x_i is the characteristic vector of the input. y_i is the regression value of this character.

$f(x) = w \cdot x + b$, $w \in \mathfrak{R}^d$, $b \in \mathfrak{R}$, if the difference value of $f(x_i)$ and y_i is small in every characteristic vector, $f(x)$ can predict y precisely from x , w is the normal vector in the plane which the SVR is looking for. ε is insensitive function applied in the loss function.

$$L_\varepsilon(y) = \begin{cases} 0 & , \quad |f(\mathbf{x}) - y| < \varepsilon \\ |f(\mathbf{x}) - y| - \varepsilon & , \quad \text{otherwise} \end{cases} \quad (2)$$

Optimized regression function can be obtained through minimize the following function

$$\min \frac{1}{2} \|\mathbf{w}\|^2, \text{ s. t. } \|y_i - (\mathbf{w} \cdot \mathbf{x}_i - b)\| \leq \varepsilon \quad (3)$$

where $\varepsilon \geq 0$ means the maximum difference between SVR prediction value and real value. Under the situation that ε is suitable, equation (3) can solve out the result which meets the standard. In the real situation, big difference exists between the samples because of the noise. So, we have to add ξ_i and ξ_i^* as additional relaxation to permit some of samples out of the limitation of ε . Equation (1) can be expressed as following

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*), \quad \text{s. t. } \begin{cases} y_i - \mathbf{w} \cdot \mathbf{x} - b \leq \varepsilon + \xi_i \\ \mathbf{w} \cdot \mathbf{x} + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (4)$$

C is a specific equilibrium factor used to adjust whether the training model is overfitting or underfitting. ξ_i and ξ_i^* are relaxation factor, which presents the upper bound and lower bound of output, and decide whether the respective sample can be out the of the scope ε decided. To solve the optimization problem with condition, we introduce Lagrangian multipliers - α_i, α_i^* and η_i, η_i^* , equation (4) can be rewrited into following function:

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) - \sum_{i=1}^l (\eta_i \xi_i + \eta_i^* \xi_i^*) - \sum_{i=1}^l \alpha_i (\varepsilon + \xi_i - y_i + \mathbf{w} \cdot \mathbf{x} + b) - \sum_{i=1}^l \alpha_i^* (\varepsilon + \xi_i^* + y_i - \mathbf{w} \cdot \mathbf{x} - b) \quad (5)$$

Solving (4) is equivalent to solve the following function:

$$\min_{\mathbf{w}, b, \xi} \max_{\alpha, \eta} L \quad \text{s. t. } \alpha, \eta \geq 0 \quad (6)$$

The method to calculate the extreme value is to solve out its partial derivative, then, the parameter needed can be obtained:

$$\frac{\partial L}{\partial b} = \sum_{i=1}^l (\alpha_i^* - \alpha_i) = 0 \quad (7)$$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^l (\alpha_i^* - \alpha_i) \mathbf{x}_i = 0 \quad (8)$$

$$\frac{\partial L}{\partial \xi^{(*)}} = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \quad (9)$$

If we turn (7), (8) and (9) into $\eta_i^{(*)} = C - \alpha_i^{(*)}$ and $\mathbf{w} = \sum_{i=1}^l (\alpha_i^* - \alpha_i) \mathbf{x}_i$, and substitute into (6), \mathbf{w} , b , ξ can be eliminated, only α and α^* left.

$$\begin{aligned}
 W(\alpha, \alpha^*) &= -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \mathbf{x}_i \cdot \mathbf{x}_j - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \\
 s.t. \sum_{i=1}^l (\alpha_i - \alpha_i^*) &= 0 \quad \text{and} \quad \alpha_i, \alpha_i^* \in [0, C]
 \end{aligned} \tag{10}$$

Through the maximized (10), we can calculate the optimized α_i and α_i^* ($i = 1, \dots, l$). By using training statistic, parameter b can be obtained from $f(\mathbf{x}_i) = \mathbf{w} \cdot \mathbf{x}_i + b$. Then we can predict the new input data by using $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$. From the deducing process above: \mathbf{w} can be expressed as the linear combination of \mathbf{x} , the calculation complexity of $f(\mathbf{x})$ is unrelated to the dimensionality of \mathbf{x} , only related to the number of support vectors.

To deal with the nonlinear problem, we can introduce kernel function to raise the dimension of the characteristic vector. The $\mathbf{w} \cdot \mathbf{x}$ mentioned before can be revised to a kernel function $K(\mathbf{w}, \mathbf{x})$, $K(\mathbf{w}, \mathbf{x}) = \Phi(\mathbf{w}) \cdot \Phi(\mathbf{x})$, $\Phi(\mathbf{x})$ is the mapping function to map the characteristics vector into higher dimension.

3. Color Correction Based on the Support Vector Regression

To build the correction model based on the SVR, firstly, we look for the suitable training data between objective images and original images, and optimize some learning parameters of support vector regression during using the SVR to correct the color.

3.1. CIELAB Color Workspace

In image processing, the RGB color workspace is used frequently. In RGB color workspace, if the value of blue channel is big, the same situation will happen on the channel of red and green. This means if we want to change the expression method of pixel, we have to change the value of all channels. All of these turn the modification of color into a complicate thing. So, we need a channel which is unrelated to all other color channels, also is independent on the each color component.

Through suitable conversion, we can turn RGB into CIELAB color. There are two steps in the conversion: Equation (11) is the conversion about RGB linear into XYZ; Equation (12), (13), (14) is the conversion about XYZ nonlinearity into CIELAB.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4306 & 0.3415 & 0.1784 \\ 0.2220 & 0.7067 & 0.0713 \\ 0.0202 & 0.1295 & 0.9394 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{11}$$

$$L = \begin{cases} 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3 \left(\frac{Y}{Y_n} \right) & \text{if } \frac{Y}{Y_n} \leq 0.008856 \end{cases} \tag{12}$$

$$a = 500 \left(f \left(X / X_n \right) - f \left(Y / Y_n \right) \right) \tag{13}$$

$$b = 200 \left(f \left(Y / Y_n \right) - f \left(Z / Z_n \right) \right) \tag{14}$$

where

$$f(t) = \begin{cases} \frac{1}{t^3} & \text{if } t > 0.008856 \\ 7.787t + 16/116 & \text{if } t \leq 0.008856 \end{cases} \quad (15)$$

3.2. Selection of Training Data

In the collection of multi-view video sequence, different views are collected by the different collection equipment; the different location of collection equipment will result in the existence of unmatched area between the videos they have collected. The motion prediction algorithm can be used to find their optimized matching area and calculate their view difference under overall situation. Motion prediction algorithm is a technique used in the video coding standard frequently (like MPEG, ITU, and JVT). If we don't consider the calculation complexity, the full search motion prediction can be used to find the optimized view difference vector.

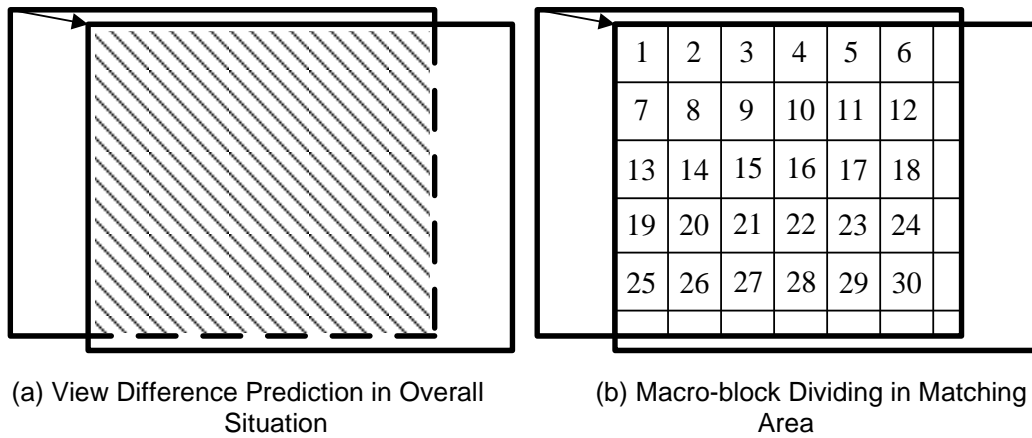


Figure 2. Matching Area and Macro-block Dividing

Suppose A and B represents the objective images and original images respectively, the view difference vector of overall situation- $DV(d_x, d_y)$ can be obtained by letting residual energy- MSE_{global} minimized. The calculation equation can be defined as following

$$MSE_{global}(d_x, d_y) = \frac{1}{(M - d_x)(N - d_y)} \sum_{y=1}^{N-d_y} \sum_{x=1}^{M-d_x} [A(x, y) - B(x - d_x, y - d_y)]^2 \quad (16)$$

So, the view difference of overall situation is:

$$DV(d_x, d_y) = \arg \min_{d_x, d_y} MSE_{global}(d_x, d_y) \quad (17)$$

The matching areas have been divided into fixed sized macro-block pair- $A(i)$ and $B(i)$, and then, we calculate the average value and variance of them. If we set $\sigma_{A(i)}^2 = \text{var}(A(i))$ and $\sigma_{B(i)}^2 = \text{var}(B(i))$, T is the threshold value, we can choose the macro-block meet the following standards

$$\left| \sigma_{A(i)}^2 - \sigma_{B(i)}^2 \right| / \sigma_{A(i)}^2 < T \quad (18)$$

The average value of macro-blocked marked adds into the sample set S.

3.3. Color Correction Algorithm Based on SVR

Based on Visual C++ 6.0, the system realizes the color correction algorithm based on SVR. In the overall multi-view video sequence, to ensure the training samples have fully presented the view difference between two views, every SVR model should be trained and used separately, because the changing tendency of video sequence is inhomogeneous. SVR model updates dynamically along with coding process. The realization of training and usage of each SVR model is as following:

Step 1: calculating the overall view difference vector between objective images and original images, and searching the overall matching area.

Step 2: dividing the objective images and original images into fixed size macro-blocks, the macro-blocks with same location in the matching area is a set of statistic.

Step 3: calculating the variance and average value of every set of macro-blocks. That the difference of the two variances compares to their average value is under the threshold, the average value of macro-blocks pairs met these standards can be added into the training set as training data.

Step 4: the training data of the training set transform from YUV color workspace into RGB, and then, to the CIELAB space.

Step 5: training the three SVR models by using the parameters L, a and b of training set.

Step 6: every pixel value of original images transforms into CIELAB space, the three components obtained will be converted by the respective SVR model. The Y, U and V components of respective pixel of adjust pictures can be obtained, after the converted components-L, a and b convert through color reverse conversion.

4. Experimental Results and Analysis

To demonstrate the performance of the color adjustment, the presented algorithm is used to correct the color of multi-view video of race1 and objects2 (KDDI co. provided, resolution ratio is 640×480). Figure 3 shows the eight views of first frame of race1 sequence before color adjustment, where the standard view is the first view. There is a little color difference between standard view and other views. Figure 5 shows the eight views of first frame of object2 sequence before color adjustment, where the standard view is the sixth view. Compared with race1 sequence, the color difference between standard view and other views is larger. Figure 4 and figure 6 are the results after color adjustment by presented algorithm, from which we can see clearly that the color difference of every view reduce obviously.

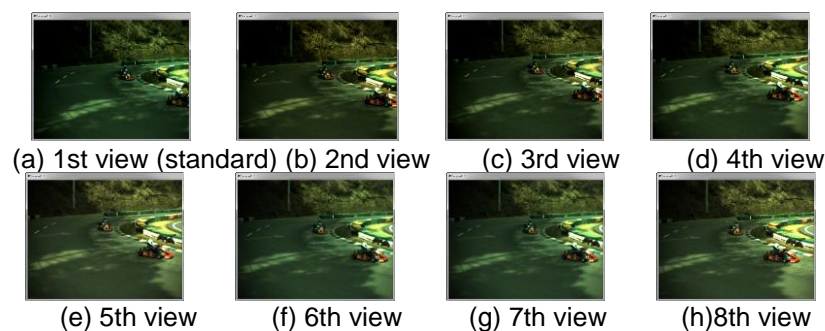


Figure 3. Eight Views of the of First Frame of Race1 Sequence before Color Adjustment

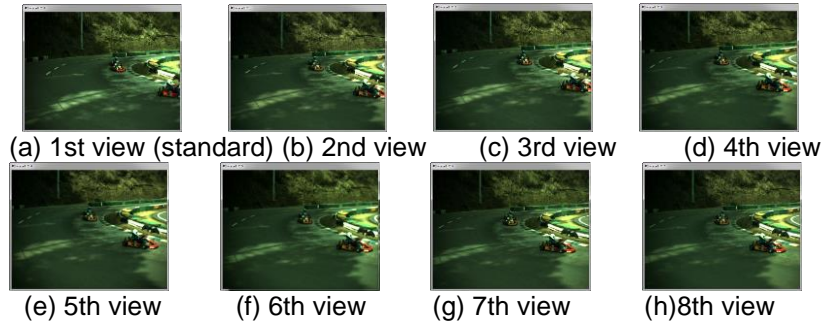


Figure 4. Eight Views of the of First Frame of Race1 Sequence after Color Adjustment

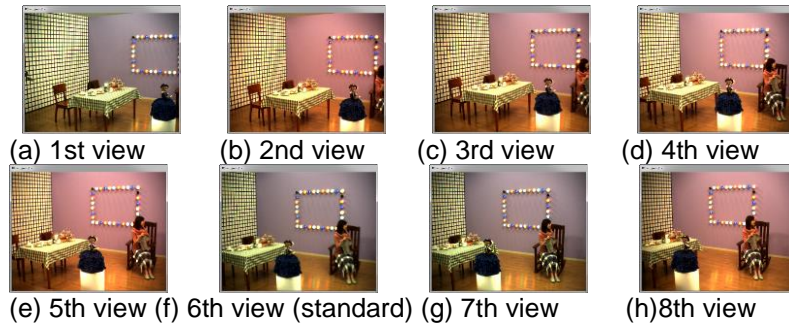


Figure 5. Eight Views of the of First Frame of Object2 Sequence before Color Adjustment

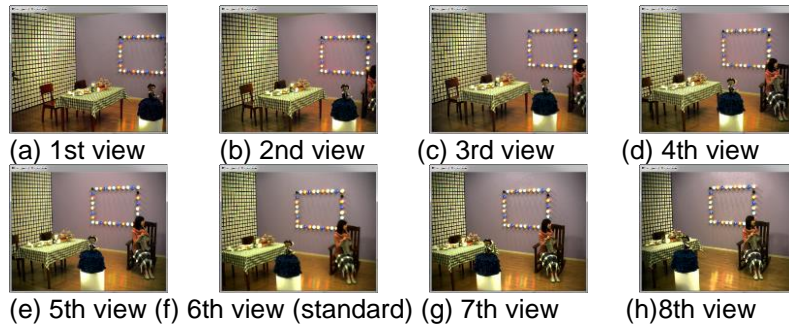


Figure 6. Eight Views of the of First Frame of Object2 Sequence After Color Adjustment

After view difference compensates the view filtering, the residual of matching macro-blocks should be coded. Reducing the residual value can improve the efficiency of compression between views. Comparison is made in the Table 1 and 2 about the residual value of matching macro-blocks of race1 sequence and objects sequence. In the experiment, using the objective view as criterion respectively, we find matching macro-blocks and calculate the residual value of it. The residual value can be expressed by using the SAD (summation of the absolute differences), the calculating equation of SAD is:

$$\begin{aligned}
 SAD = \frac{1}{M \times N} & \left\{ \sum_{x=1}^M \sum_{y=1}^N \left| Y_r(x, y) - Y_p(i + m v_x, y + m v_y) \right| \right. \\
 & + \sum_{x=1}^{M/2} \sum_{y=1}^{N/2} \left| U_r(x, y) - U_p(i + m v_x / 2, y + m v_y / 2) \right| \\
 & \left. + \sum_{x=1}^{M/2} \sum_{y=1}^{N/2} \left| U_r(x, y) - U_p(i + m v_x / 2, y + m v_y / 2) \right| \right\} \quad (19)
 \end{aligned}$$

From tables, the reduction of residual value between view matching macro-blocks is obvious, after the color adjustment.

Table 1. The SAD Value between Objective View and other Views (race1 sequence)

Other views	2	3	4	5	6	7	8
Pre-adjusting	35	35	39	45	41	45	46
Pro-adjusting	35	35	36	36	35	36	36

Table 2. The SAD Value between Objective View and other Views (object2 sequence)

Other views	1	2	3	4	6	7	8
Pre-adjusting	57	57	59	57	54	55	56
Pre-adjusting	48	47	47	47	48	48	50

5. Conclusion

A color adjustment algorithm based on support vector regression is presented in this paper. This algorithm builds color mapping relationship between views through statistical learning to the training sets, and adjusts the color of other views used a specific view as the standard. The color difference of each view reduces obviously, after the using of the algorithm proposed. Through the analysis of experiments, the correlation between views can be enhanced after the color adjustment, and the redundancy data of different views can be erased better by the compensation filtering of for view different views. The experimental result shows the proposed algorithm is effective.

Acknowledgement

This work is supported by technological innovation foundation of Harbin (2012RFQXG090) and Natural Science Foundation of Heilongjiang Province (F201245).

References

- [1] A. Vetro, T. Wiegand and G. J. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard", Proceedings of the IEEE, vol. 99, no. 4, (2011), pp.626-642.
- [2] M. Flierl, A. Mavlankar and B. Girod, "Motion and disparity compensated coding for multi-view video", IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, (2007), pp.1474-1484.
- [3] U. Fecker, M. Barkowsky and A. Kaup, "Histogram-based prefiltering for luminance and chrominance compensation of multiview video", IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 8, (2008), pp. 1258-1267.

- [4] Y. Chen, J. Chen and C. Cai, "Luminance and chrominance correction for multi-view video using simplified color error model", Proc. Picture Coding Symposium, (2006); Beijing, China.
- [5] K. Yamamoto, M. Kitahara, H. Kimata, T. Yendo, T. Fujii, T. M. Tanimoto, S. Shimizu, K. Kamikura and Y. Yashima, "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.
- [6] C. Doutre and P. Nasiopoulos, "Color correction preprocessing for multiview video coding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 9, (2009), pp. 1400-1405.
- [7] V. N. Vapnik, "Statistical Learning Theory", Wiley, (1998).
- [8] E. Reinhard, M. Ashikhmin, B. Gooch and P. Shirley, "Color Transfer between Images", IEEE Computer Graphics and Applications, vol. 21, no. 5, (2001), pp. 34-41.

Authors

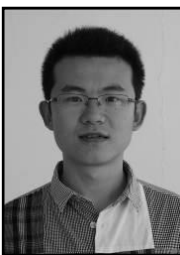


Huadong Sun was born in Heilongjiang, China, in 1980. He received the Doctor degree in communication engineering from Harbin Institute of Technology, Harbin, China, in 2009.

He joined Harbin University of Commerce in 2010. His research areas are image processing and signal processing.



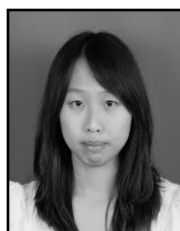
Xuesong Jin was born in 1975. He received the Doctor degree in communication engineering from Harbin Institute of Technology, Harbin, China, in 2014. His research areas are image processing and image compression.



Zhipeng Fan was born in 1984. He received the Master degree in Computer and Information Engineering, Harbin University of Commerce, in 2012. His research areas are image and processing, communications and networking.



Lizhi Zhang was born in Heilongjiang, China, in 1968. He received the Master degree in communication engineering from Harbin Institute of Technology, Harbin, China, in 1995. His research areas are image processing, vector quantization, and information theory



Qian Wu was born in 1990. She is a graduate student in Computer and Information Engineering, Harbin University of Commerce. Her research areas are image and processing, communications and networking.

