ICIC Express Letters
Part B: Applications
Volume **8**, Number **3**, March **2017**
 ISSN 2185-2766
pp. **553–561**

COLOR RECONSTRUCTION OF GREYSCALE IMAGES BASED ON SPARSE REPRESENTATION

Di Zhang[1], Baosheng Kang[1], Lei Zhang[2] and Jing Zhang[1]

[1]School of Information Science and Technology
Northwest University
No. 1, Xuefu Avenue, Xi'an 710127, P. R. China
{ xbdxzd; bskang; mywangwang123 }@163.com

[2]Public Computer Teaching Department
Yuncheng University
No. 1155, West Fudan Street, Yuncheng 044000, P. R. China
zhanglei0359@126.com

Received August 2016; accepted November 2016

Abstract. *Sparse representation has drawn greater attention in color reconstruction. This paper proposes a color reconstruction approach based on sparse representation of classified image patches and non-local sparse coding. Firstly, image patches are classified into smooth, texture and edge classes, and different dictionaries are trained by the relevant information. Then, non-local coefficient is obtained by the calculated local coefficients. Finally, the color of the greyscale image is reconstructed by the dictionary and the non-local coefficient. The experimental results show that the proposed method can achieve competitive performance compared with other state-of-the-art methods.*
Keywords: Color reconstruction, Sparse representation, Patch classification, Non-local sparse coding

1. **Introduction.** Colorization is a computer process of adding three-dimensional (RGB) pixel values to greyscale image under certain constraints [1,2], which can increase the visual appeal of images such as old black and white photos, classic movies and scientific illustrations [2]. Color images have significant application value in image processing. However, greyscale image is easy to ignore some significant information because of the missing of color information. So the color reconstruction is very important.

At present, image rendering methods mainly include color diffusion based on manual stroke [1,3], color transfer based on the reference color image [2,4].

The method of color diffusion is based on a simple premise that nearby pixels in space-time that have similar intensities should have similar colors [1]. Levin et al. [1] proposed the optimization of the weighting functions to add the color in YUV color space, but its efficiency is very low. Yatziv and Sapiro [3] proposed a fast image and video colorization using chrominance blending, but it is strict with the input images.

Welsh et al. [2,4] proposed the method of color transfer based on the reference color image that the entire color "mood" of the reference color image is transferred to the target image by matching luminance and texture information between the images. However, the matching process is time-consuming and may assign vastly different colors to neighboring pixels which have similar intensities.

With the development of compression sense [5], sparse representation theory [6] has emerged as a powerful technique in color reconstruction. Hao et al. [7,8] proposed that reconstructing the color of the greyscale image uses sparse representation.

Unlike the work done in [7,8] that reconstructing the color of all the greyscale image patches applies only one dictionary and encodes each image patch independently, in this paper, our approach reconstructs the greyscale image based on sparse representation of classified image patches and non-local sparse coding to overcome the disadvantages of weak adaptability of the dictionary and artificial blocking effect.

Inspired by [9,10], in which the method of classifying the image into different classes and using non-local sparse coding is used respectively, in our method, the image patches are classified into smooth, edge and texture classes by the threshold of variance and entropy to train different dictionaries, which adapt to different image patches types and have strong adaptivity. In addition, given the non-local self-similarity of images, the algorithm calculates the weighted average of local codes of similar patches to obtain the non-local sparse coding of each image patch [11,12], which can avoid the artificial blocking effect. The strength of the dictionary and the non-local sparse coding can greatly improve the reconstruction results. Numerical experiments demonstrate that the proposed algorithm has a good reconstruction performance.

2. **Principle.**

This section describes the related techniques including the correlation between luminance, feature and color information, the principle of classification, the method of gradient and texture feature extraction, as well as the non-local sparse coding.

2.1. **Correlation.**

According to numerous observations and analysis of natural images, two assumptions [7] are established. They are
(1) If the luminance of spatially adjacent pixels is similar, the color is commonly similar;
(2) The similar features have similar color.

Based on the above assumptions, the joint matrix connected by luminance and features information has the same sparse coefficient with the color matrix in the joint dictionary, so we can recover the desired color image easily.

2.2. **Classification.**

In order to eliminate the defects in [7,8], we take the idea of classification [9] into account. The image patches are classified into smooth, edge and texture patches according to variance σ^2 and entropy E [10]. The variance of the image patch is

$$\sigma^2 = \frac{1}{m}\sum_{j=0}^{m-1}(x_j-\mu)^2 \tag{1}$$

where m is the number of pixels in an image patch, x_j is the grey value of the j-th pixel, and μ is the average value of the grey value of the image patch.

The entropy of the image patch is

$$E = -\sum_{i=0}^{255} p_i \log p_i \tag{2}$$

where p_i is the probability of the i-th pixel level in the image patch.

Let variance threshold H_1 and entropy threshold H_2 be the criteria of classification [10]. The patch is smooth if $\sigma^2 < H_1$, $E < H_2$. If $\sigma^2 > H_1$ and $E > H_2$, it belongs to the edge patches. The remaining patches are classified as the texture patches.

2.3. **Feature extraction.**

Additional features should be extracted to improve the results, because a single luminance value could not represent entirely different parts of an image. The variation of gradient value of the edge patches is obvious [10], so the gradient feature is usually used to describe the feature of the edge patches effectively. We can use high-pass filters to extract the gradient feature [13]. The filters are

$$f_{11} = [1,0,-1],\ f_{12} = f_{11}^T,\ f_{21} = [1,0,-2,0,1],\ f_{22} = f_{21}^T \tag{3}$$

For texture patches, there are varieties of ways to extract them, which can be classified as statistical method, syntactic method and spectrum method [14]. In this paper, extracting the texture feature only depends on luminance [7], so gray-level co-occurrence matrix (GLCM) [7,15], which is the simplest statistical method, is the best choice. Several parameters including the mean and standard deviation of the angular second moment (ASM), entropy, moment of inertia (MOI) and correlation are calculated to reflect the texture feature.

The gradient feature extraction needs four filters, which results in the increases of the dimension of the image patches [16]. Similarly a certain redundancy in the texture feature extraction with eight parameters must exist. In order to reduce the dimension and the complexity of the algorithm, principal component analysis (PCA) is applied.

2.4. **Non-local sparse coding.** The methods in [7,8] encode each patch independently, not taking account of the non-local self-similarity. To overcome the defects, this paper handles them non-locally. The calculation of the non-local coefficients is as follows.

Step 1. Calculate the Euclidean distance D_{ij} between the i-th and j-th image patches by Equation (4)

$$D_{ij} = \|v(x_i) - v(x_j)\|_2 \tag{4}$$

Step 2. Measure the similarity by D_{ij} and set the threshold T.

Step 3. Get the local coefficients α_i of the i-th image patch x_i by Equation (5)

$$\alpha_i = \arg\min_{\alpha_i} \left\{ \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_0 \right\} \tag{5}$$

where D is the dictionary, λ is a small constant denoting the regularization parameter, and α_i is the sparse coefficient of the i-th class.

Step 4. Calculate the weighting factors by searching the similar patches if their distance is less than T. Then get the non-local coefficients β_i of the i-th image patch by the weighted average of local codes of similar patches α_i

$$\beta_i = \sum_j \omega_{i,j}\alpha_j,\ 0 \le \omega_{i,j} \le 1,\ \sum_j \omega_{i,j} = 1,\ \omega_{i,j} = \frac{1}{Z(i)} \exp\left(-\|v(x_i) - v(x_j)\|_2^2\right)/h^2 \tag{6}$$

where h is the attenuation factor,

$$Z(i) = \sum_j \exp\left(-\|v(x_i) - v(x_j)\|_2^2\right)/h^2$$

3. **Algorithm.** This section describes the algorithm in detail. This method differs from other state-of-the-art methods in that it uses the method of classification and non-local sparse coding. The patches are classified into three categories by the variance threshold H_1 and entropy threshold H_2 to train different dictionaries by the K singular value decomposition (K-SVD) algorithm, and the local sparse coefficient α_i of the greyscale image is calculated by the regularized orthogonal matching pursuit (ROMP) algorithm according to the obtained corresponding dictionaries, and then the non-local sparse coefficient β_i is obtained according to their weighted average of local codes of similar patches. The greyscale image can be reconstructed using β_i and the corresponding color dictionary. The method has great superiorities in reconstruction results.

The algorithm can be grouped into two aspects, namely *Dictionary Learning* and *Color Reconstruction*. The input images are the reference RGB image Y and original greyscale image G. The color image X is the desired output image.

3.1. **Dictionary learning.** The notations in the algorithm are as follows.

$$L_i = [l_{i1}, l_{i2}, \ldots, l_{ik_i}];\ C_i = [c_{i1}, c_{i2}, \ldots, c_{ik_i}];\ T = [t_1, t_2, \ldots, t_{k_2}];\ F = [f_1, f_2, \ldots, f_{k_3}]$$

where l_{ij} is the luminance. $c_{ij} = [r_{ij}, g_{ij}, b_{ij}]^T$ is the color value, t_j is the texture feature, and f_j is the gradient feature. Besides, $i = 1, 2, 3$ represent the smooth class, texture class, and edge class respectively and k_i represents the number of patches of the i-th class image patches. D_i is the dictionary of the i-th class; D^L is the luminance dictionary,

D^G is the combinational dictionary of the luminance and features and D^C is the color dictionary.

Step 1. Get the image Y and divide it into patches with overlap.

Step 2. Classify Y into smooth patches, edge patches, and texture patches.

Step 3. Combine l_{1j} and c_{1j} into a column vector y_{1j}, and then get the smooth sample

$$Y_1 = [y_{11}, y_{12}, \ldots, y_{1k_1}]$$

Combine l_{2j}, t_j and c_{2j} into a column vector y_{2j}, and then get the texture sample

$$Y_2 = [y_{21}, y_{22}, \ldots, y_{2k_2}]$$

Combine l_{3j}, f_j and c_{3j} into a column vector y_{3j}, and then get the edge sample

$$Y_3 = [y_{31}, y_{32}, \ldots, y_{3k_3}]$$

Step 4. Learn D_1, D_2 and D_3 of the training samples Y_1, Y_2 and Y_3 by Equation (7) using K-SVD

$$\{D_i, \gamma_i\} = \arg\min_{D_i,\gamma_i} \left\{ \|Y_i - D_i\gamma_i\|_2^2 + \lambda \|\gamma_i\|_0 \right\}, \quad i = 1, 2, 3 \tag{7}$$

where $D_1 = \left[D_1^L, D_1^C\right]^T = \left[D_1^G, D_1^C\right]^T$, $D_2 = \left[D_2^L, D_2^F, D_2^C\right]^T = \left[D_2^G, D_2^C\right]^T$, $D_3 = \left[D_3^L, D_3^T, D_3^C\right]^T = \left[D_3^G, D_3^C\right]^T$.

3.2. **Color reconstruction.** The notations in the algorithm are as follows. l_i is the luminance of the smooth, texture or edge greyscale image patch. t is the texture feature. f is the gradient feature. k_i is the number of patches of the i-th class image patch.

Step 1. Set $G_1 = l_1$, $G_2 = [l_2, t]^T$, $G_3 = [l_3, f]^T$.

Step 2. Calculate the local sparse coefficient α_i by Equation (8)

$$\alpha_i = \arg\min_{\alpha_i} \left\{ \left\|G_i - D_i^G\alpha_i\right\|_2^2 + \lambda \|\alpha_i\|_0 \right\}, \quad i = 1, 2, 3 \tag{8}$$

where $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ij}, \ldots, \alpha_{ik_i}]$, and α_{ij} is the sparse coefficient of the j-th image patch of the i-th class.

Step 3. Get the non-local coefficient β_{ij} by their weighted average of local codes of similar patches.

Step 4. Set $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{ij}, \ldots, \beta_{ik_i}]$.

Step 5. Reconstruct the color of greyscale images according to the following Equation (9)

$$C_i = D_i^C \beta_i, \quad i = 1, 2, 3 \tag{9}$$

4. **Results.** In this section, the reconstructed results are demonstrated. The images are separated into lots of 8×8 patches with 4 pixels overlapping, and the dictionary has $4 \times N_i$ atoms, where N_i is the size of the rows of the training samples. Subjective and objective evaluation criteria are used to verify the effectiveness of the algorithm, and the structural similarity (SSIM) [17] and color-difference formula (ΔE_{00}) CIEDE2000 in L*a*b* color space [18,19] are applied to evaluating the results.

The SSIM formula [17] is

$$SSIM = \frac{4\mu_x\mu_y\sigma_{x,y}}{\left(\mu_x^2 + \mu_y^2\right)\left(\sigma_x^2 + \sigma_y^2\right)} \tag{10}$$

where μ_x and μ_y represent the mean color values of the target reconstructed image and the original color image respectively. σ_x and σ_y are standard deviation respectively. $\sigma_{x,y}$ is the covariance.

The calculation of the CIEDE2000 uses Formula (11) in L*a*b* color space.

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{K_L S_L}\right)^2 + \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{K_C S_C}\right)\left(\frac{\Delta H'}{K_H S_H}\right)} \tag{11}$$

where S_L, S_C and S_H are weighting factors adjusting the uniformity of color space. K_L, K_C and K_H are adjustment factors related to using conditions, influencing the feeling of color difference. The closer to 1 the SSIM value is and the smaller the value ΔE is, the better the reconstruction results are.

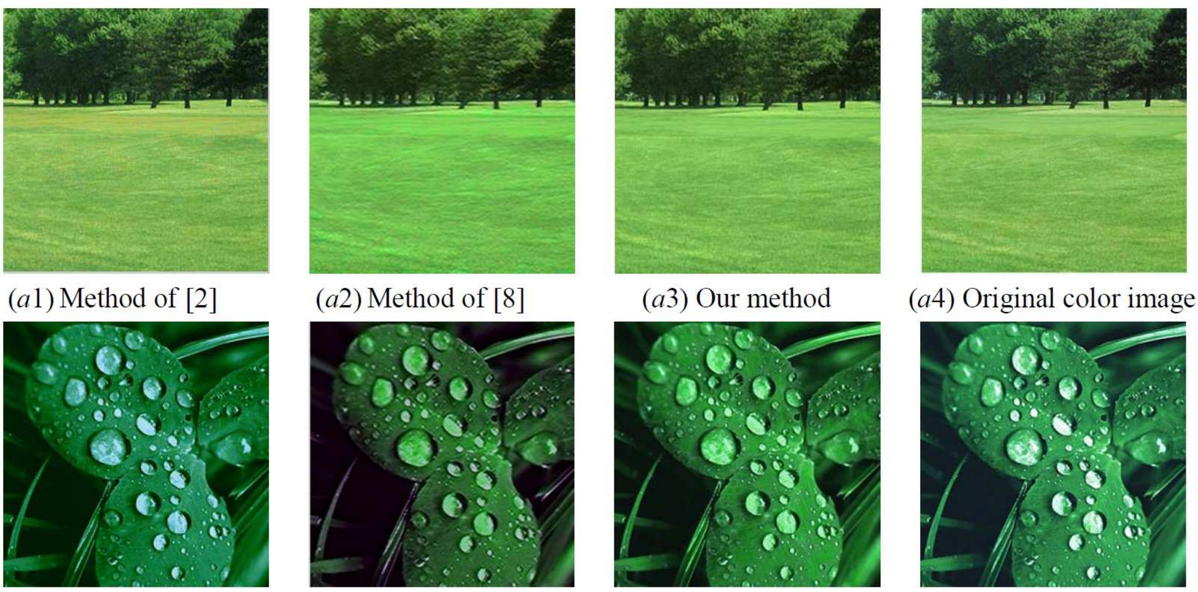
In order to make the comparison more obvious, the images are classified into smooth, texture and edge images. In each experiment, the greyscale images are reconstructed using different methods.

Experiment 1. In the experiment, the smooth images, which include the reference color images Figure 1($a1$), Figure 1($b1$) and original greyscale images Figure 1($a2$), Figure 1($b2$), are presented in Figure 1, named *Grass and Leaf*. Figure 2($a1$) and Figure 2($b1$) are colorized by the method in [2], Figure 2($a2$) and Figure 2($b2$) by the method in [8], and Figure 2($a3$) and Figure 2($b3$) by our method. Figure 2($a4$) and Figure 2($b4$) are the original color images. Objective comparison of experimental results is shown in Table 1.

Experiment 2. In the experiment, the reconstruction results of the texture images are presented, named *Tiger and Wall*. Figure 3 presents the reference color images Figure 3($a1$), Figure 3($b1$). Figure 3($a2$) and Figure 3($b2$) are original greyscale images. Figure

($a1$) Reference image ($a2$) Greyscale image ($b1$) Reference image ($b2$) Greyscale image

Figure 1. The original smooth images



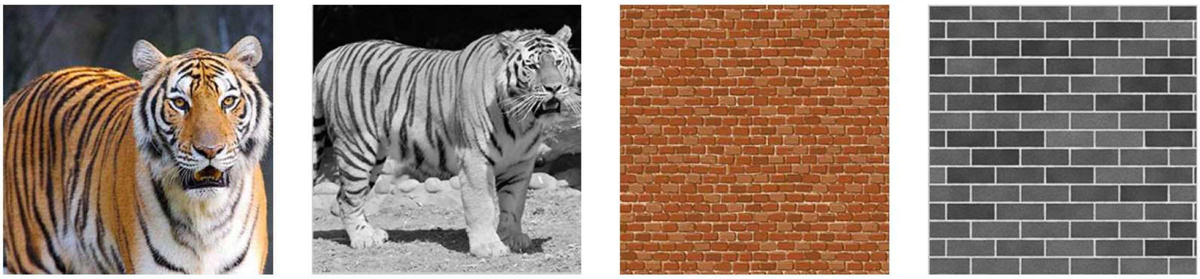
($a1$) Method of [2] ($a2$) Method of [8] ($a3$) Our method ($a4$) Original color image

($b1$) Method of [2] ($b2$) Method of [8] ($b3$) Our method ($b4$) Original color image

Figure 2. The reconstruction smooth images

Table 1. Comparison of reconstruction results

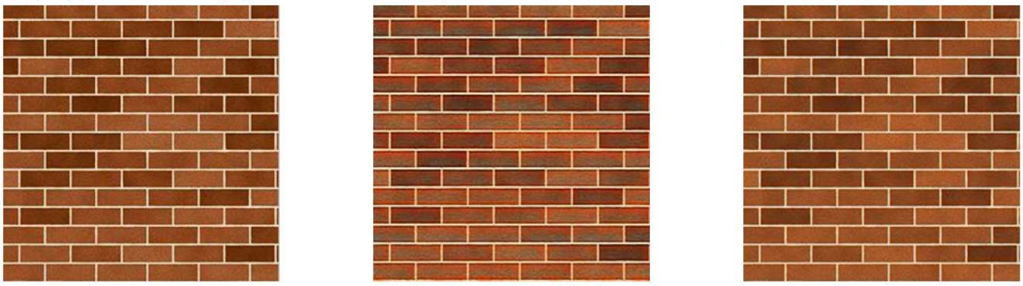
Image	**Criterion**	**Algorithms**		
		Method of [2]	Method of [8]	Our method
Grass	SSIM	0.836	0.961	0.986
	ΔE_{00}	7.425	4.073	2.735
Leaf	SSIM	0.786	0.855	0.943
	ΔE_{00}	8.198	6.894	5.430



(a1) Reference image (a2) Greyscale image (b1) Reference image (b2) Greyscale image

Figure 3. The original texture images

(a1) Method of [2] (a2) Method of [8] (a3) Our method



(b1) Method of [2] (b2) Method of [8] (b3) Our method

Figure 4. The reconstruction texture images

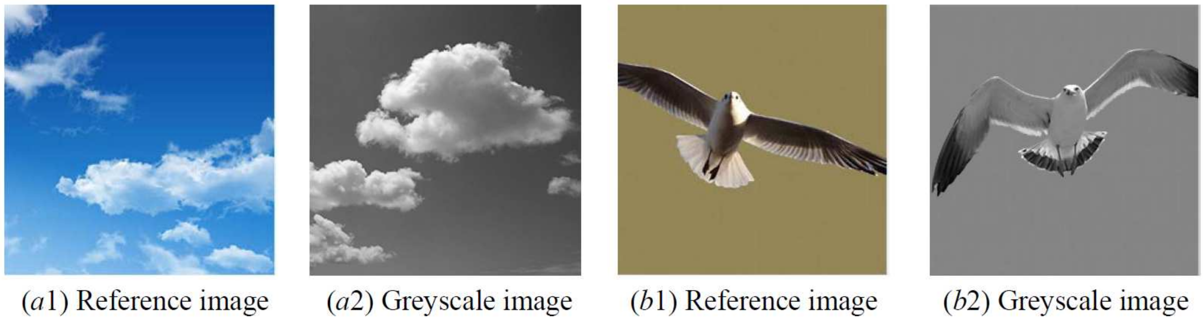
Table 2. Comparison of reconstruction results

Image	**Criterion**	**Algorithms**		
		Method of [2]	Method of [8]	Our method
Tiger	SSIM	0.688	0.910	0.932
	ΔE_{00}	12.559	6.673	6.024
Wall	SSIM	0.863	0.833	0.874
	ΔE_{00}	6.987	7.584	6.738

4 shows the reconstruction images by using different methods, in which Figure 4(a1) and Figure 4(b1) are colorized by the method in [2], Figure 4(a2) and Figure 4(b2) by the

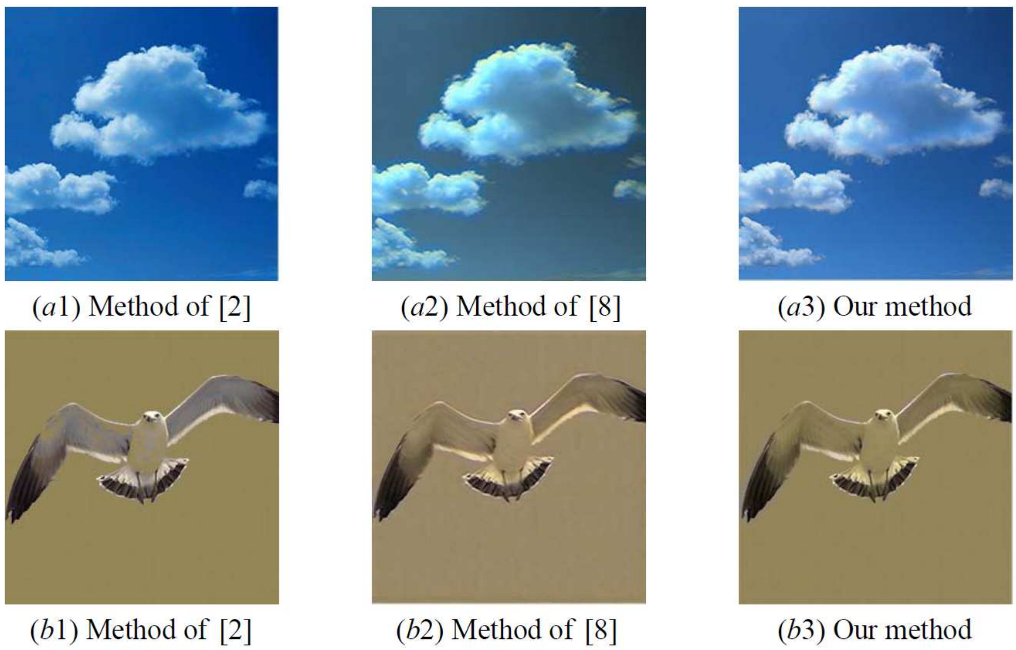
method in [8], and Figure 4($a3$), Figure 4($b3$) by our method. Objective comparison is shown in Table 2.

Experiment 3. In this experiment, the edge images, which include the reference color images Figure 5($a1$), Figure 5($b1$) and original greyscale images Figure 5($a2$), Figure 5($b2$), are presented, named *Sky and Bird*. Figure 6 shows the reconstruction images using different methods, in which Figure 6($a1$) and Figure 6($b1$) are colorized by the method in [2], Figure 6($a2$) and Figure 6($b2$) by the method in [8], and Figure 6($a3$) and Figure 6($b3$) by our method. Objective comparison of experimental results is shown in Table 3.



($a1$) Reference image ($a2$) Greyscale image ($b1$) Reference image ($b2$) Greyscale image

Figure 5. The original edge images



($a1$) Method of [2] ($a2$) Method of [8] ($a3$) Our method

($b1$) Method of [2] ($b2$) Method of [8] ($b3$) Our method

Figure 6. The reconstruction edge image

Table 3. Comparison of reconstruction results

Image	Criterion	Algorithms		
		Method of [2]	Method of [8]	Our method
Sky	SSIM	0.912	0.759	0.965
	ΔE_{00}	6.462	8.687	3.603
Bird	SSIM	0.978	0.981	0.990
	ΔE_{00}	3.553	3.452	2.034

As it can be seen from the experimental results, the reconstruction effect of our method is better than other methods both in subjective and objective evaluation criteria. Other methods show distinct chromatic aberration by the calculation of the CIEDE2000 and the reconstructed results look not smooth and not uniform. The primary reason that our method can obtain better results is that it takes account of the differences of various image patches and the non-local sparse coding.

5. **Conclusions.** Our method makes full use of the sparse representation and the reliable theoretical basis of the correlation among the luminance, feature and the color information to reconstruct the color of the greyscale images. From the comparison of experimental results, as long as the threshold of classification is reasonable, the results are definitely beneficial and favorable whenever they are evaluated from subjective or objective aspects. It classifies the patches into smooth patches, texture patches and edge patches, which can reduce the redundant feature information firstly so that the storage space is greatly reduced compared to the method in [7,8] that the extraction of the feature is untargeted. In addition, it can reflect the differences of various image patches types, so the trained dictionaries have stronger adaptive ability to obtain satisfactory results. Finally, the reconstructed color images have good authenticity and smoothness.

Although the proposed approach achieves better performance than previous approaches, the setting of the threshold is still a very hard problem. Our future work is to develop a better method to improve the accuracy of the results.

Acknowledgment. This work is partially supported by the Project NSFC (61272286), and the Project of Natural Science Foundation of Shaanxi Province, China (2014JM8346). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] A. Levin, D. Lischinski and Y. Weiss, Colorization using optimization, *ACM Trans. Graphics (TOG)*, vol.23, no.3, pp.689-694, 2004.
[2] T. Welsh, M. Ashikhmin and K. Mueller, Transferring color to greyscale images, *ACM Trans. Graphics (TOG)*, vol.21, pp.277-280, 2002.
[3] L. Yatziv and G. Sapiro, Fast image and video colorization using chrominance blending, *IEEE Trans. Image Processing*, vol.15, no.5, pp.1120-1129, 2006.
[4] R. Irony, D. Cohen-Or and D. Lischinski, Colorization by example, *Eurographics Symposium on Rendering Techniques*, Konstanz, Germany, pp.201-210, 2005.
[5] D. L. Donoho, Compressed sensing, *IEEE Trans. Information Theory*, vol.52, no.4, pp.1289-1306, 2006.
[6] J. Yang, J. Wright, T. S. Huang et al., Image super-resolution via sparse representation, *IEEE Trans. Image Processing*, vol.19, no.11, pp.2861-2873, 2010.
[7] K. Hao, M. Song, J. Bu et al., Natural grayscale image colorization via local sparse coding, *Journal of Computer-Aided Design & Computer Graphics*, vol.23, pp.1401-1408, 2011.
[8] X. Zhang, R. Zhao et al., Color reconstruction algorithm for grayscale images based on dictionary learning and sparse representation, *Journal of Computer-Aided Design & Computer Graphics*, vol.26, pp.1092-1098, 2014.
[9] Q.-S. Lian and W. Zhang, Image super-resolution algorithm based on sparse representation of classified image patches, *Acta Electronica Sinica*, vol.40, pp.920-925, 2012.
[10] J. Li, *Super Resolution Reconstruction Algorithm Based on Dictionary Learning Using Multi-Feature and Multi-Scale Description*, Master Thesis, South China University of Technology, 2014.
[11] A. Buades, B. Coll and J. M. Morel, A review of image denoising algorithms, with a new one, *Multiscale Modeling & Simulation*, vol.4, no.2, pp.490-530, 2005.
[12] L. Zhe, Y. Jing et al., Super-resolution image restoration based on non-local sparse coding, *Journal of Electronics & Information Technology*, vol.37, pp.522-528, 2015.
[13] I. Guyon and A. Elisseeff, An introduction to feature extraction, *Feature Extraction*, Springer Berlin Heidelberg, 2006.

[14] M. Sharma and S. Singh, Evaluation of texture methods for image analysis, *The 7th Australian and New Zealand Intelligent Information Systems Conference*, pp.117-121, 2001.
[15] M. Benco and R. Hudec, Novel method for color textures features extraction based on GLCM, *Radioengineering*, vol.4, no.16, pp.64-67, 2007.
[16] H. Zhu, C. Xu et al., Super-resolution reconstruction of image based on improved dictionary learning, *Journal of Nanchang Institute of Technology*, vol.34, pp.1-6, 2015.
[17] Z. Wang, A. C. Bovik, H. R. Sheikh et al., Image quality assessment: From error visibility to structural similarity, *IEEE Trans. Image Processing*, vol.13, no.4, pp.600-612, 2004.
[18] Y. Yang, J. Ming and N. Yu, Color image quality assessment based on CIEDE2000, *Advances in Multimedia*, vol.2012, no.2, 2012.
[19] G. Sharma, W. Wu and E. N. Dalal, The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations, *Color Research and Application*, vol.30, no.1, pp.21-30, 2005.