

PAPER

Improvement of Colorization-Based Coding Using Optimization by Novel Colorization Matrix Construction and Adaptive Color Conversion

Kazu MISHIBA^{†a)} and Takeshi YOSHITOME[†], *Members*

SUMMARY This study improves the compression efficiency of Lee's colorization-based coding framework by introducing a novel colorization matrix construction and an adaptive color conversion. Colorization-based coding methods reconstruct color components in the decoder by colorization, which adds color to a base component (a grayscale image) using scant color information. The colorization process can be expressed as a linear combination of a few column vectors of a colorization matrix. Thus it is important for colorization-based coding to make a colorization matrix whose column vectors effectively approximate color components. To make a colorization matrix, Lee's colorization-based coding framework first obtains a base and color components by RGB-YCbCr color conversion, and then performs a segmentation method on the base component. Finally, the entries of a colorization matrix are created using the segmentation results. To improve compression efficiency on this framework, we construct a colorization matrix based on a correlation of base-color components. Furthermore, we embed an edge-preserving smoothing filtering process into the colorization matrix to reduce artifacts. To achieve more improvement, our method uses adaptive color conversion instead of RGB-YCbCr color conversion. Our proposed color conversion maximizes the sum of the local variance of a base component, which resulted in increment of the difference of intensities at region boundaries. Since segmentation methods partition images based on the difference, our adaptive color conversion leads to better segmentation results. Experiments showed that our method has higher compression efficiency compared with the conventional method.

key words: colorization-based coding, image compression, colorization matrix, edge-preserving filtering, adaptive color conversion

1. Introduction

Recently, several colorization-based coding methods have been proposed [1]–[5]. Colorization is the process of adding color to a grayscale image using scant color information, which is called representative pixels (RP). Colorization-based coding methods apply colorization to these decoding process. In the encoding process, an RGB color image is first converted to one base component and two color components, which are often one luminance component and two chrominance components obtained by RGB-YCbCr conversion. Next, the base component is compressed by a conventional compression method, e.g., JPEG 2000, then RP are extracted from the color components using the coded base component. In the decoding process, the color com-

ponents are constructed through colorization using the RP and the coded base component, then a color image is reconstructed using the inverse color conversion. The difficulty of colorization-based coding is how to extract RP in encoder for efficient compression.

Colorization can be formulated as a process that applies a colorization matrix to a vector which consists of RP. In the previous colorization-based coding methods [2]–[5] using Levin's colorization method [6], a colorization matrix is constructed after RP extraction. These methods iteratively add or reduce RP to find a better RP set which reduces the reconstruction error between an original color component and its reconstructed color component. In this approach, obtained RP may redundant or incomplete because it does not guarantee the optimality of RP. Lee et al. [1] took a different approach from the previous methods. Their method constructs a colorization matrix before RP extraction. It obtains the optimal RP with respect to a given colorization matrix in the sense that it minimizes the reconstruction error by solving an optimization problem. Since this approach mathematically guarantees that the obtained RP set is optimal with respect to a given colorization matrix, the compression efficiency depends on the colorization matrix. The study [1] proposed the above optimization-based coding framework, but included little discussion on how to construct an efficient colorization matrix for compression. While the method in [1] outperforms other previous methods in compression efficiency, it suffers from artifacts and still has room for improvement in compression efficiency.

This paper proposes a variant of [1] that offers better compression efficiency. The differences between [1] and our proposed method are at the construction of a colorization matrix and color space conversion.

The construction of a colorization matrix consists of the following two steps: performing a segmentation method on a base component, and then constructing the entries of the colorization matrix using the segmentation results. Although the entries affect on the compression performance, the previous study [1] has little discussion on it. This paper discusses how to construct a better colorization matrix using segmentation results.

Whereas method [1] uses RGB-YCbCr conversion to obtain a base and color components, our method uses adaptive color conversion which maximizes the sum of the local variance of a base component. The adaptive color con-

Manuscript received March 25, 2015.

Manuscript revised July 16, 2015.

Manuscript publicized July 31, 2015.

[†]The authors are with the Department of Electrical and Electronic Engineering, Tottori University, Tottori-shi, 680–8550 Japan.

a) E-mail: mishiba@ele.tottori-u.ac.jp

DOI: 10.1587/transinf.2015EDP7106

version leads to appropriate segmentation results, which resulted in improvement of compression efficiency.

The rest of this paper is organized as follows. Section 2 presents the colorization-based coding framework proposed in [1]. We show the processing flow of our coding method in Sect. 3. Our colorization matrix construction is presented in Sect. 4. Section 5 introduces adaptive color conversion method for efficient compression. Experimental results are presented in Sect. 6. Section 7 concludes the paper.

This paper is an extension of our previous study [7].

2. Colorization-Based Coding Using Optimization

For the following discussions, an image component is expressed as a column vector form ordered in lexicographic order. Let \mathbb{R} , $|\cdot|_0$, $\|\cdot\|$, \cdot^T and $\bar{\cdot}$ denote the set of all real numbers, ℓ^0 -norm, ℓ^2 -norm, transposition of a matrix, and the average of a vector, respectively.

Lee et al. [1] formulate a colorization process as the following matrix form:

$$u = Cx \quad (1)$$

where $x \in \mathbb{R}^m$ is a column vector which contains values only at the positions of RP, $C \in \mathbb{R}^{n \times m}$ is a colorization matrix, where normally $m < n$, and $u \in \mathbb{R}^n$ is a constructed color component. In the colorization process, C and x are given and u is the solution to be determined. In contrast, in the colorization-based coding of [1], C and u are given and x is the solution to be determined. In the context of compression, x needs to be sparse. The optimal solution is formulated as

$$\arg \min_x \|u_0 - Cx\|^2 \text{ s.t. } |x|_0 \leq L \quad (2)$$

where L is a positive integer which controls the number of non-zero components in x and $u_0 \in \mathbb{R}^n$ represents an original color component. Equation (2) can be solved by the Orthogonal Matching Pursuit algorithm [8].

Obviously, the solution of Eq. (2) depends on a colorization matrix C . It is built using a segmentation result performed on a base component. Each segmented region defines non-zero positions of each column vector of C . Values at non-zero positions can be designed arbitrarily. We show an example of the column vector which is designed in a binary manner. An entry of a column vector $b^\omega \in \mathbb{R}^n$ created from a segmented region Ω^ω in a binary manner is expressed by

$$b_i^\omega = \begin{cases} 1 & \text{if } i \in \Omega^\omega \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where ω is an index of a segmented region. To assign a certain value c to an entry of x corresponding to b^ω means to add c to a color component to be reconstructed in a segmented region Ω^ω . Using multiple segmentation results enhances a possibility to obtain column vectors appropriate for reconstruction. To achieve it, a segmentation method is performed at different scales. The meanshift segmentation [9] is a suitable method for this purpose because it can

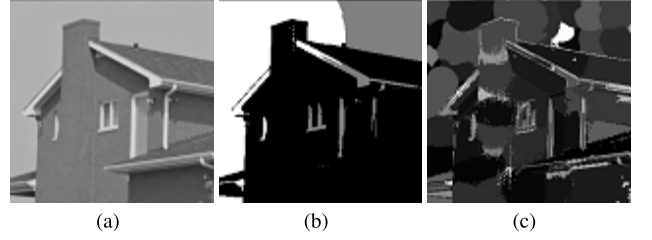


Fig. 1 Segmentation results of base component (a) with two different scales (b) and (c). In each segmentation result, the same luminance means the same region.

obtain different segmentation results with different spatial and range resolutions. Figure 1 represents different segmentation results with two different scales. By performing segmentation at each scale s , the whole image Ω is divided into multiple segmented regions satisfying

$$\bigcup_{t=1}^{t_s} \Omega^{s,t} = \Omega, \quad \bigcap_{t=1}^{t_s} \Omega^{s,t} = \emptyset, \quad \forall s \in \{1, \dots, s_k\} \quad (4)$$

where $\Omega^{s,t}$ is the t -th segmented region at scale s , t_s is the number of segmented regions at scale s , and s_k is the number of scales. Let Φ be a set $\Phi = \{\omega | \omega = (s, t), s \in \{1, \dots, s_k\}, t \in \{1, \dots, t_s\}\}$, $|\Phi|$ be the number of elements of Φ , and Ω^ω be the t -th segmented region at scale s when $\omega = (s, t)$. Given a column vector $v^\omega \in \mathbb{R}^n$ which is designed in a certain manner for a colorization matrix, we denote by $V^\Phi \in \mathbb{R}^{n \times |\Phi|}$ the matrix whose column vectors are v^ω ordered in lexicographic order on Φ , i.e.,

$$V^\Phi = [v^{\omega_1}, \dots, v^{\omega_{|\Phi|}}]. \quad (5)$$

In the study [1], two types of matrix are created using segmentation results. The first matrix[†] is a binary matrix $B^\Phi \in \mathbb{R}^{n \times |\Phi|}$, whose column vectors are b^ω in Eq. (3). The second matrix $W^\Phi \in \mathbb{R}^{n \times |\Phi|}$ is constructed using the Euclidean distance d_i^ω between the center of the mass of Ω^ω and the pixel $i \in \Omega^\omega$. The column vector $w^\omega \in \mathbb{R}^n$ is expressed as

$$w_i^\omega = \begin{cases} \frac{1}{d_i^\omega} & \text{if } i \in \Omega^\omega \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Using the above two matrices B^Φ and W^Φ , the colorization matrix is constructed as

$$C_1 = [B^\Phi, W^\Phi]. \quad (7)$$

In both encoder and decoder, the same colorization matrix can be obtained by performing a segmentation on a base component to be transmitted from encoder to decoder.

In this method, RP consist of the non-zero positions and their values of x for two color components. It is empirically observed that RP for two color components have almost the same non-zero positions. To reduce the amount

[†]This matrix is not described in [1], but is included in the implementation code developed by the authors.

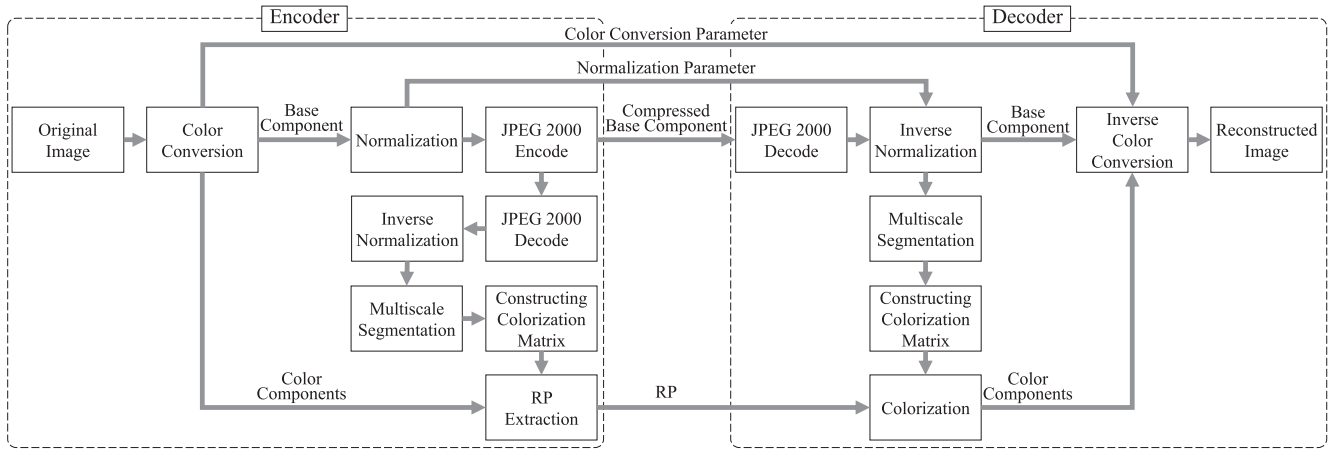


Fig. 2 Processing flow of our method.

of bits required for RP, Eq. (2) is solved with a constraint that the non-zero positions for two color components are the same. Therefore this method transmits the following two components: a base component compressed by a conventional image compression method, and an RP set consisting of the non-zero positions to color components and each values for them. In this paper, base components are compressed by the JPEG 2000 standard. The decoding process is as follows. First, the base component is decompressed, then segmentation is performed on the base component. Next, a colorization matrix is constructed using segmentation results. Finally, color components are reconstructed by the colorization process of Eq. (1) using the colorization matrix and the RP set.

3. Processing Flow

Our compression procedure basically follows to [1] but differs in several ways. Figure 2 shows the processing flow of our method. The color conversion matrix depends on images. Thus its inverse color conversion matrix needs to be given in decoder. The normalization procedure adjusts the range of intensities of a base component to the range of possible values on the following compression of a base component.

Compressed data in our method consist of a compressed base component, RP for color components, variables for color conversion and for normalization. Each RP requires 28 bits to be stored, where 12 bits are for the non-zero positions of x , and 16 bits for the two color component values (each 8 bits). The color conversion matrix requires three variables to be stored because it is a three dimensional rotation matrix. The normalization requires two variables, which are the minimum and maximum values of a base component before normalization. Our method expresses each variable with 8 bits.

4. Proposed Colorization Matrix Construction

In the colorization-based coding framework, the compres-

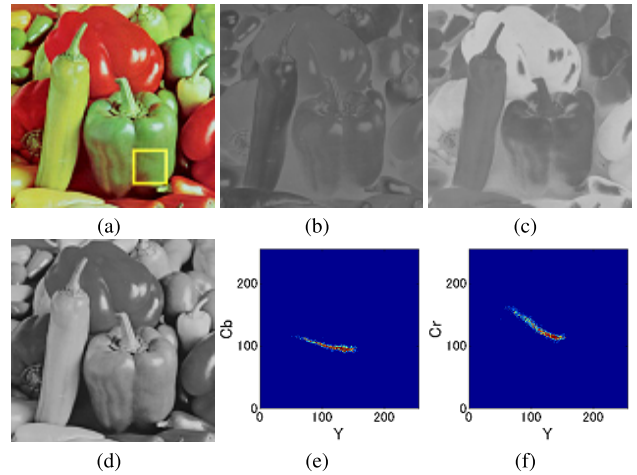


Fig. 3 Correlation between base (Y) and color (Cb and Cr) components. (a) Original RGB image. (b), (c) and (d) Cb, Cr and Y components of (a). (e) and (f) Distributions in the yellow boxed area of (a) between base and color components.

sion efficiency obviously depends on a colorization matrix C . This section describes how to construct a colorization matrix for efficient compression. Our proposed colorization matrix is derived from the correlation between a base and its corresponding color components in a local area, which is described in Sect. 4.1. In addition, we embed an edge-preserving smoothing filtering process into the colorization matrix to reduce artifacts, which is described in Sect. 4.2.

4.1 Correlation of Base-Color Components

Since color components are reconstructed based on their corresponding base component in the colorization-based coding, it is reasonable to use a correlation between the base and color components to construct a colorization matrix. Figure 3 (b) - (d) presents the base and color components obtained by applying RGB-YCbCr conversion to the original image in Fig. 3 (a). Figure 3 (e) and (f) shows the distributions of the values of the base and color components

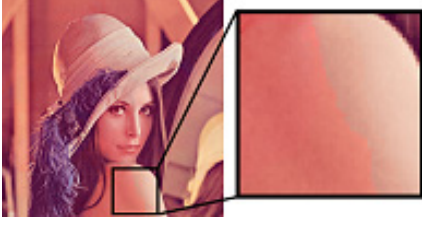


Fig. 4 Reconstruction result with the colorization matrix C_2 .

of a local area. As can be seen from this figure, the values of a color component in a local area can be approximated by a translation and a linear transformation of the values of its corresponding base component. Let y and u be column vectors of a base and color components of a local area, respectively. Here, $\sigma_y^2 \neq 0$ where σ_y^2 is the variance of y . u can be approximated by the following formula:

$$u \simeq \alpha y + \beta \quad (8)$$

where α and β are scalars. In a least squares sense, $\alpha = \sigma_{y,u}/\sigma_y^2$ and $\beta = \bar{u} - \alpha\bar{y}$ where $\sigma_{y,u}$ is a covariance between y and u . Therefore, Eq. (8) can be rewritten as

$$u \simeq [\mathbf{1}, \quad y - \bar{y}] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (9)$$

where $\mathbf{1}$ is a vector in which all of the elements are ones, $x_1 = \bar{u}$ and $x_2 = \alpha$. Let column vectors y^ω and u^ω be a base and color component in Ω^ω , respectively. Since many segmented regions are small, the following expression can be a good approximation for a color component in a segmented region Ω^ω :

$$\begin{bmatrix} b^\omega & g^\omega \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} \quad (10)$$

where $x'_1 = \bar{u}^\omega$, $x'_2 = \sigma_{y^\omega, u^\omega}/\sigma_{y^\omega}^2$, and g^ω is a column vector in G^Φ that has the following entries:

$$g_i^\omega = \begin{cases} y_i - \bar{y}^\omega & \text{if } i \in \Omega^\omega \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where y_i is the i -th entry of y . Based on the discussion above, the following colorization matrix can be expected to reconstruct the color component with less error:

$$C_2 = \begin{bmatrix} B^\Phi & G^\Phi \end{bmatrix}. \quad (12)$$

4.2 Edge-Preserving Smoothing Filtering

Since the meanshift segmentation is a hard segmentation method, which makes hard boundaries, a reconstructed color image may include artificial contours (see Fig. 4). We need to eliminate artificial contours while preserving true contours. As shown in Fig. 3 (b) - (d), contours in a base component and its corresponding color components often

occur at the same position. Therefore, it is highly possible that artificial contours in a reconstructed color component occur at the regions where its corresponding base component is smooth. This premise indicates that an edge-preserving smoothing filter whose coefficients are computed from a base component can eliminate artificial contours in its corresponding color components. To guarantee the optimality of a reconstructed color component, we embed the filtering process into the colorization matrix. Let $R \in \mathbb{R}^{n \times n}$ be a process of edge-preserving smoothing filtering. The colorization matrix can be expressed as

$$C_3 = R \begin{bmatrix} B^\Phi & G^\Phi \end{bmatrix}. \quad (13)$$

We can use any edge-preserving smoothing filter whose kernel weights can be expressed explicitly, such as the joint bilateral filter [10] and the guided image filter [11]. In the joint bilateral filtering process, the entry of R is expressed as

$$R_{i,j} = \frac{1}{N_i} \exp\left(-\frac{\|p_i - p_j\|^2}{\sigma_s^2}\right) \exp\left(-\frac{\|y_i - y_j\|^2}{\sigma_r^2}\right) \quad (14)$$

where N_i is a normalizing factor for the i -th row, p_i and p_j are pixel coordinates at the i -th and j -th pixels, and σ_s and σ_r are geometric and photometric spread, respectively.

As with C_1 and C_2 , the colorization matrix C_3 is of size $n \times 2|\Phi|$ and is constructed from a decoded base component.

5. Color Conversion

Whereas the previous study [1] uses RGB-YCbCr color conversion to make a base and color components, our method uses an adaptive one which maximizes the sum of the local variance of a base component. Thus to find adaptive color conversion can be written as a principal component analysis (PCA) procedure. The adaptive color conversion increases the difference of intensities at region boundaries. Since segmentation methods partition images based on the difference, our adaptive color conversion leads to better segmentation results. Let $\psi, |\psi|, X^\psi \in \mathbb{R}^{|\psi| \times 3}$ be a local area, the number of pixels in ψ , and an RGB color component in ψ , respectively. The column vectors of X^ψ correspond to R, G and B components. Base component y^ψ and two color components u_1^ψ and u_2^ψ in ψ are expressed with a color conversion matrix $T \in \mathbb{R}^{3 \times 3}$ as

$$\begin{bmatrix} y^\psi & u_1^\psi & u_2^\psi \end{bmatrix} = X^\psi T \quad (15)$$

where

$$T = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix} \quad (16)$$

and $t_i \in \mathbb{R}^3$ satisfies $\|t_i\|^2 = 1$ and $t_i \perp t_j (i \neq j)$. Our adaptive color conversion maximizes the following sum of the local variance of y : $\sum_{\psi \in \Psi} \sigma_{y^\psi}^2$ where $\sigma_{y^\psi}^2$ is the variance of y^ψ and Ψ is a set of all local areas on y . Based on a PCA procedure, t_1 , t_2 and t_3 can be calculated as the first, second and third largest eigenvectors of the variance-covariance matrix of $M \in \mathbb{R}^{3 \times 3}$ where the entry is

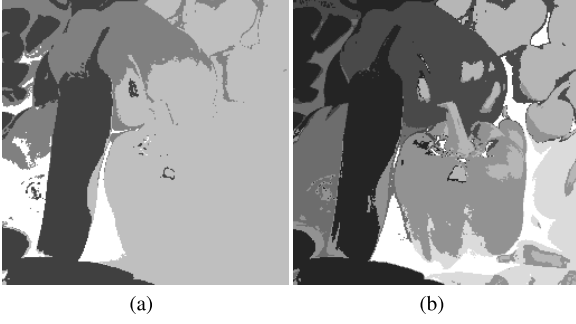


Fig. 5 Segmentation results of base components with same scale parameter using RGB-YCbCr color conversion (a) and adaptive color conversion (b).

$$M_{i,j} = \sum_{\psi \in \Psi} \frac{1}{|\psi|} (X_i^\psi - \overline{X_i^\psi})^T (X_j^\psi - \overline{X_j^\psi}) \quad (17)$$

and X_i^ψ is the i -th column vector of X^ψ .

Figure 5 shows segmentation results of base components converted from an original image shown in Fig. 3 (a) by RGB-YCbCr and our adaptive color conversion. As shown in this figure, the adaptive color conversion produces plausible result compared with RGB-YCbCr color conversion.

6. Experimental Results

Figure 6 shows the test images for the experiment. We used *imwrite* function with a default setting in MATLAB for JPEG 2000 compression. The function controls the compression ratio r by changing ‘*CompressionRatio*’ parameter. The meanshift segmentation was performed with 16 different scales described in [1]. The filtering process R in our method was the joint bilateral filtering with $\sigma_s = 4$, $\sigma_r = 50$ and the window size 11×11 . In our proposed color conversion, a set Ψ of all local areas on an image consists of the subimages obtained by dividing the image into 64×64 pixels.

The number of RP affects compression efficiency. Figure 7 shows the PSNR performance of [1] and our method in several combinations of the compression ratio of base components and the number of RP. In each compression ratio, the number of RP was changed from 40 up to 600 at an interval of 80. The PSNRs are calculated between the RGB channels of the original and its reconstructed images. The data size means the total bits of compressed data, i.e., a compressed base component and RP for color components in [1], and in addition to them, variables for color conversion and for normalization in our method. As can be seen in this result, increment of the number of RP does not always improve PSNRs. The reason is that obtained RP minimize the reconstruction error of color components instead of RGB channels. In the following experiment, we use the following empirically-determined relation:

$$L = \lceil 2000/r \rceil \quad (18)$$

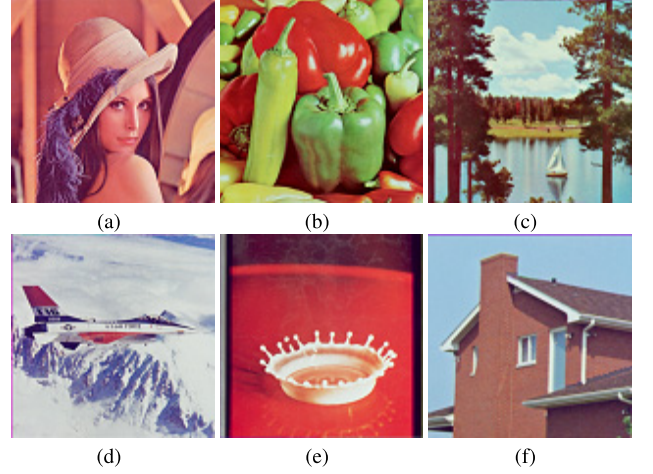


Fig. 6 Test images of size 256×256 ; (a) “Lenna” image, (b) “Pepper” image, (c) “Sailboat” image, (d) “Airplane” image, (e) “Milkdrop” image, and (f) “House” image

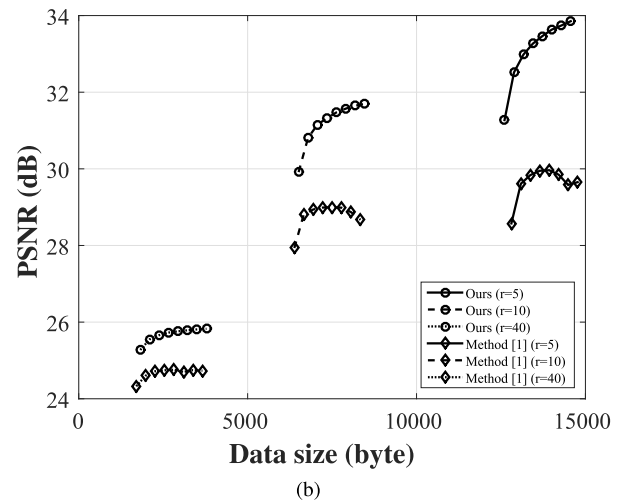
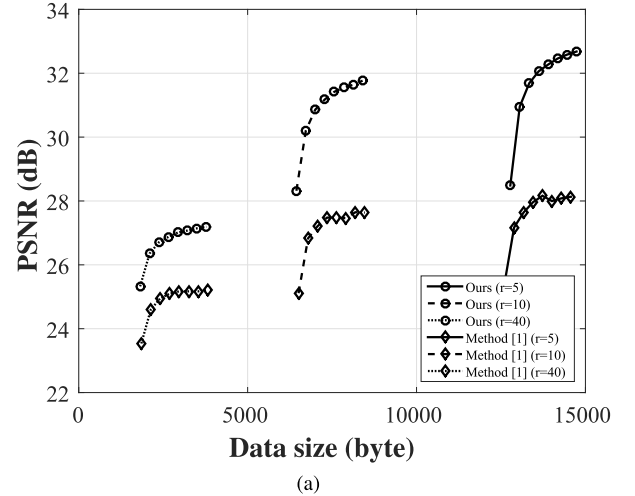


Fig. 7 Performance for method [1] and our method; (a) *Pepper* and (b) *Sailboat*

where $\lceil \cdot \rceil$ is the ceiling function. It is not optimal for all the images, but it produced plausible results in our experiment.

We compared our method with the JPEG 2000 stan-

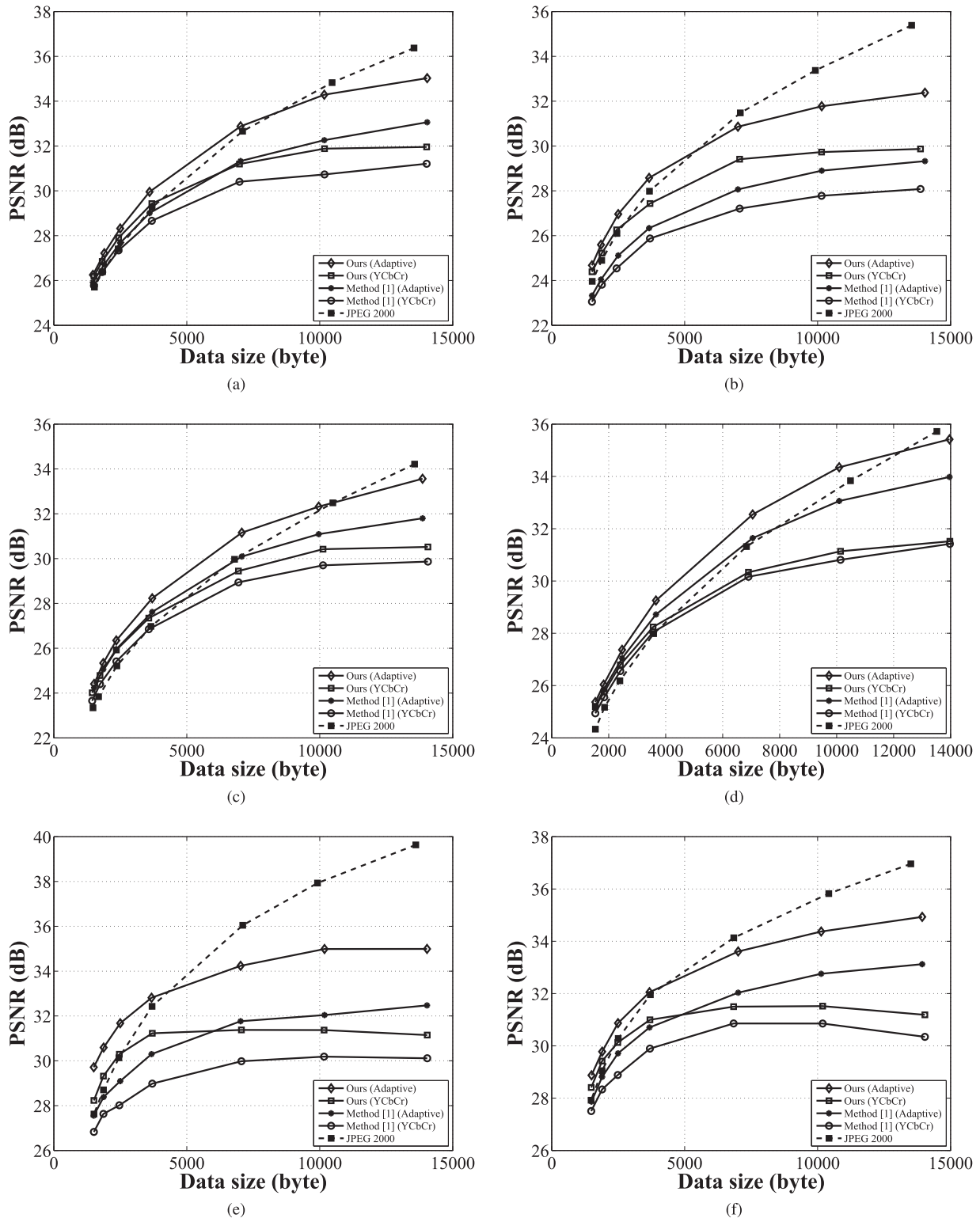


Fig. 8 Compression performance for JPEG 2000, method [1] and our method with RGB-YCbCr and our adaptive color conversion; (a) *Lena*, (b) *Pepper*, (c) *Sailboat*, (d) *Airplane*, (e) *Milkdrop*, and (f) *House*

dard and the conventional method [1]. To show the effect of the combination of our colorization matrix construction and adaptive color conversion, our method and [1] were performed with the proposed and RGB-YCbCr color con-

version. Figure 8 shows the PSNR performance over all test images. As shown by comparison between our method and [1] using RGB-YCbCr color conversion, our method achieved higher performance. This result indicates that our

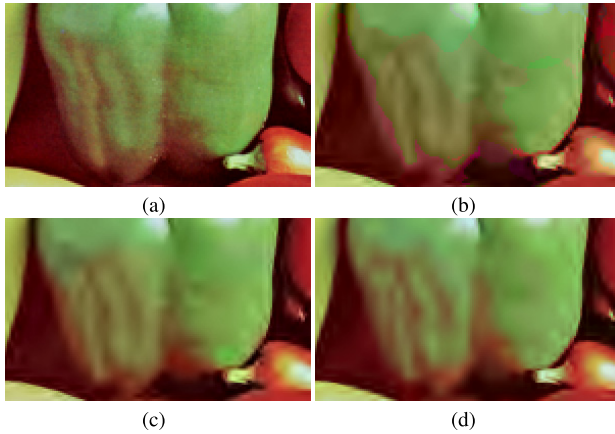


Fig. 9 Reconstructed results of “Pepper” image (zoom-in); (a) original image, (b) method in [1], (c) our method with RGB-YCbCr color conversion, and (d) our method with proposed color conversion

colorization matrix construction can improve compression efficiency. As shown by comparison between the results using the proposed and RGB-YCbCr color conversion in our method and [1], our proposed color conversion greatly contributed to improvement of compression efficiency. As a result, our proposed method with the combination of our colorization matrix construction and adaptive color conversion provided higher PSNR values than [1] for all the test images. Parts of the reconstructed images of *Pepper* with about 2500 bytes are shown in Fig. 9. Whereas [1] reconstructed an image with many artificial contours, our method reconstructed an image with less artificial contours. Compared with the JPEG 2000, our method indicated lower compression efficiency for higher bit rates and higher one for lower bit rates, as shown in Fig. 8. As this result indicates, the colorization-based coding framework proposed in [1] can roughly approximate images with a few data because bases of a colorization matrix cover large areas.

7. Conclusions

In this paper, we proposed a better colorization-based coding method in compression efficiency based on [1]. To improve compression efficiency, we introduced the following two approaches: a colorization matrix construction using a correlation of base-color components and embedding an edge-preserving smoothing filtering process, and the adaptive color conversion which maximizes the sum of the local variance of a base component. The experimental results showed that our method has higher compression efficiency than the conventional method.

Acknowledgments

We are grateful to Sukho Lee and his coauthors for providing the code for the colorization-based coding.

References

- [1] S. Lee, S. Park, P. Oh, and M. Kang, “Colorization-based com-

- pression using optimization,” *IEEE Trans. Image Processing*, vol.22, no.7, pp.2627–2636, 2013.
- [2] L. Cheng and S.V.N. Vishwanathan, “Learning to compress images and videos,” *Proceedings of the 24th International Conference on Machine Learning, ICML ’07*, New York, NY, USA, pp.161–168, ACM, 2007.
- [3] X. He, M. Ji, and H. Bao, “A unified active and semi-supervised learning framework for image compression,” *CVPR*, pp.65–72, IEEE, 2009.
- [4] T. Miyata, Y. Komiyama, Y. Sakai, and Y. Inazumi, “Novel inverse colorization for image compression,” *Proceedings of the 27th Conference on Picture Coding Symposium, PCS’09*, Piscataway, NJ, USA, pp.101–104, IEEE Press, 2009.
- [5] S. Ono, T. Miyata, and Y. Sakai, “Colorization-based coding by focusing on characteristics of colorization bases,” *PCS*, pp.230–233, IEEE, 2010.
- [6] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” *ACM Trans. Graph.*, vol.23, no.3, pp.689–694, Aug. 2004.
- [7] K. Mishiba and T. Yoshitome, “Colorization matrix construction with high compression efficiency for colorization-based coding using optimization,” *Proc. IEEE Int. Conf. Image Processing*, pp.5551–5555, 2014.
- [8] Y.C. Pati, R. Rezaifar, and P.S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, pp.40–44, 1993.
- [9] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.24, pp.603–619, 2002.
- [10] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, “Digital photography with flash and no-flash image pairs,” *ACM Trans. Graph.*, vol.23, no.3, pp.664–672, Aug. 2004.
- [11] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.35, no.6, pp.1397–1409, 2013.



Kazu Mishiba received the B.E. and M.E. in 2004 and 2006, respectively from Keio University, Yokohama, Japan. In 2006, He joined FUJI FILM CO., Ltd. He received the Ph.D. in 2011 from Keio University. He was an assistant professor at Keio University in 2011. He is currently an assistant professor at Tottori University. His research interests are in the areas of image interpolation, content-aware resizing, and image editing.



Takeshi Yoshitome received the B.S. and M.S. degrees in computer science from Tsukuba University, Japan, in 1982 and 1984, respectively. In 1984, he joined the Electrical Communication Laboratories, Nippon Telegraph and Telephone Corporation (NTT), Kanagawa, Japan and has been engaged in research and development of image processing systems. He received the Ph.D. degree in computer science from Tsukuba University in 2010.