Signal Processing: Image Communication 74 (2019) 266–279

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

Colorization-based image coding using graph Fourier transform☆

Kazunori Uruma [a,*], Katsumi Konishi [b], Tomohiro Takahashi [c], Toshihiro Furukawa [d]

[a] Kogakuin University, Japan
[b] Hosei University, Japan
[c] Tokai University, Japan
[d] Tokyo University of Science, Japan

ARTICLE INFO

Keywords:
Colorization
Image compression
Signal processing on graphs
Graph Fourier transform

ABSTRACT

This paper deals with the colorization-based image coding algorithm. In this algorithm, a color image is compressed by encoding its luminance image by a standard coding method such as JPEG coding and by storing several color pixels called as representative pixels (RPs). In decoding phase, a color image is restored from a luminance image and some color information of RPs using the image colorization technique. While previous studies have achieved a high coding performance, the compression method of RPs has not been considered because the positions of RPs are inhomogeneous. In order to improve the image coding performance, this paper proposes the RPs compression algorithm using the graph Fourier transform, where the chrominance image is transformed to graph spectrum and compressed. Using this RPs compression algorithm and a colorization technique, a new colorization-base image coding algorithm is proposed. Numerical results show that the proposed algorithm achieves better performance than some previous studies and JPEG2000 coding.

1. Introduction

The image colorization technique provides a color image from a luminance image and several color pixels called as representative pixels (RPs), which have multi-dimensional information of values and their positions [1–3]. Based on the image colorization, some image compression algorithms have been proposed [4–10]. In encoding phase of these algorithms, appropriate RPs are extracted from a color image, then the information of RPs are stored, and the luminance image is compressed by standard coding method such as JPEG coding. In decoding phase, the colorization algorithm recovers a color image from luminance image and RPs. Because the colorization algorithm can recover a full color image from a few number of RPs, the compression algorithms using colorization achieve a high coding performance.

Cheng et al. have proposed the colorization-based image coding algorithm [4], which uses a machine learning to determine which pixels are extracted as RPs. In [6], Ono et al. have proposed an algorithm based on Levin's colorization algorithm. One of its contributions is that the recovery error is minimized by optimizing the values of RPs. However, these algorithms [4,6] do not consider how to store the positions and chrominance values of RPs. In [7], Lee et al. have proposed the colorization-based coding algorithm using sparse optimization. In this algorithm, a new colorization technique has been proposed to compress the chrominance images effectively based on an image segmentation technique, and appropriate RPs are selected by sparse optimization technique. The advantage of this algorithm is that a few volume of information to store the positions of each RP is required. Furthermore, Lee's algorithm has been improved based on the correlation between the luminance and chrominance components in a local area [8]. However the volume of information to store the positions of RPs is not zero, that is, there is room for improvement of the compression performance. In [10], a new colorization technique and its image coding algorithm have been proposed. The most important characteristic is that this algorithm recovers not only the chrominance image but also the luminance image in decoding phase. However its performance is almost the same as that of JPEG2000.

The performance of the colorization-based image coding algorithm depends on its colorization accuracy and compression rate of the information about RPs, which consists of the information of their coordinates and chrominance values.

Simple way to achieve high information compression rate of RPs is to choose pixels at grid points as RPs. If all RPs are on grid points, it is not required to store the positions of all RPs, and the chrominance values can be compressed by applying a standard coding method such as JPEG because the RPs can be represented as a small size image. However this idea leads to a low colorization accuracy because grid points pixels are not appropriate to be used as RPs. Although high colorization

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.image.2018.12.011.

* Corresponding author.
E-mail address: uruma@cc.kogakuin.ac.jp (K. Uruma).

https://doi.org/10.1016/j.image.2018.12.011
Received 21 February 2018; Received in revised form 18 December 2018; Accepted 18 December 2018
Available online 31 December 2018
0923-5965/

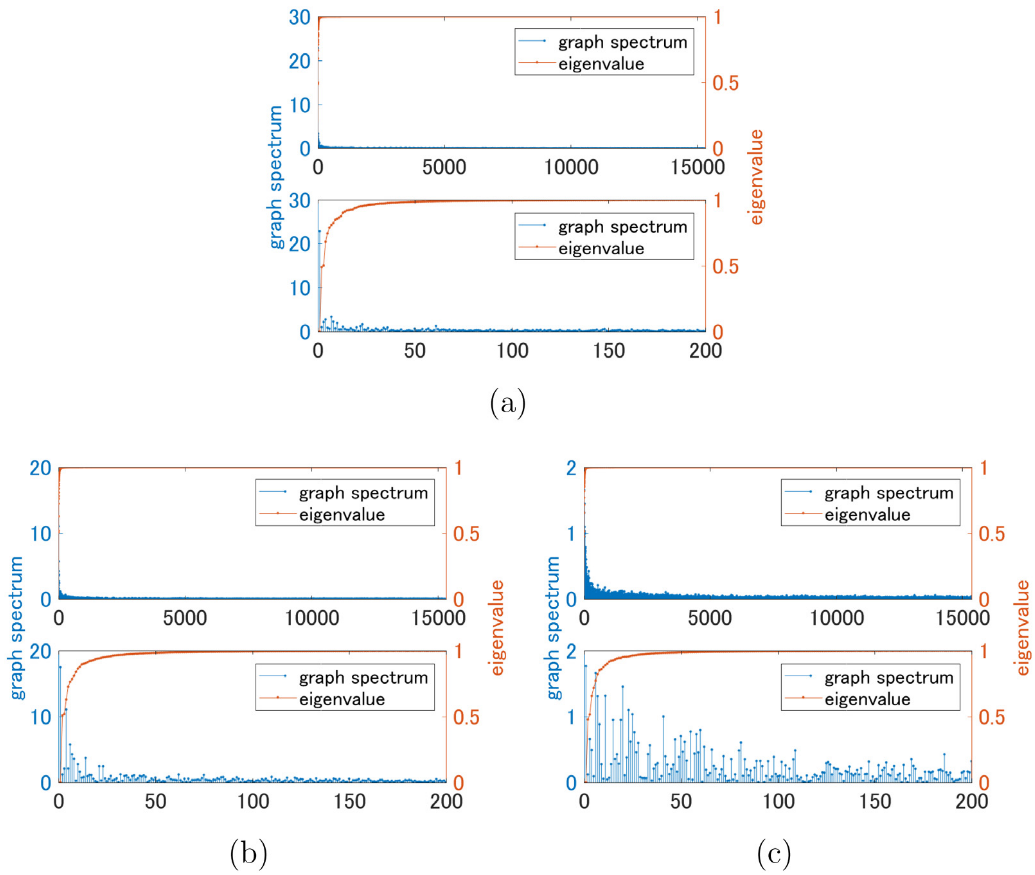

Fig. 1. The graph spectrum of the Cb values of RPs (the lower side shows the region from 1 to 200 of the upper side): (a) Lenna image, (b) Pepper image and (c) Airplane image in Fig. 7.

performance can be achieved by choosing RPs whose positions are not grid points, we cannot compress their chrominance values by any image coding method due to inhomogeneity of their positions. Motivated by these points, the challenge of this work is to propose a colorization-based image coding algorithm achieving higher coding performance than the previous colorization-based image coding algorithm and JPEG2000.

In order to achieve high colorization accuracy with less information of RPs, this paper proposes a colorization-based image coding algorithm using an image segmentation technique and the graph Fourier transform. Empirical results show that the state-of-the-art superpixel method [11] enables us to find good RPs by choosing a central pixel at each segment as RPs after image segmentation. Furthermore, it is not required to store the positions of RPs because we can obtain exact positions of RPs at decoding phase by applying the same image segmentation algorithm to the same luminance image at encoding phase. Although the positions of RPs extracted based on the image segmentation are inhomogeneous, we can compress their chrominance values by using the graph Fourier transform, which has been studied in the field of the graph signal processing [12–16]. A brief scheme of the proposed algorithm is as follows. First, in encoding phase, the luminance image is segmented based on the image segmentation algorithm, and the RPs are extracted as the center pixels of each segment. Next the graph is constructed by utilizing the each RPs as vertex, and the weight of edges are calculated from luminance image. The chrominance values of RPs are represented as the graph signal, which is transformed to the graph spectrum based on the graph Fourier transform. Then the value of graph spectrum corresponding to the low frequency is stored. In decoding phase, the same graph of the encoding phase is constructed by applying the same segmentation method to a luminance image, and a chrominance image is restored from a luminance image and graph spectrum. Contributions of this paper is to propose an RPs compression method based on graph spectrum and to provide a new colorization-based image coding algorithm using graph Fourier transform. Numerical results show that the proposed algorithm achieves better performance than both some previous studies, JPEG2000 and JPEG XR coding.

This paper is organized as follows. In Section 2, we give related works — Levin's colorization method, Lee's colorization-based image coding algorithm and graph Fourier transform. Section 3 proposes the colorization-based image coding using the graph Fourier transform, and numerical examples show the effectiveness of the proposed algorithm in Section 4.

2. Related works

This paper provides a new image coding algorithm based on the image colorization techniques and the graph Fourier transform. For preliminary, this subsection introduces these related works.

2.1. Levin's image colorization technique

Levin et al. have proposed a colorization algorithm in [1]. This algorithm recovers a chrominance image $\boldsymbol{u} \in R^{MN}$ from a luminance image $\boldsymbol{y} \in R^{MN}$ using representative pixels (RPs) $\boldsymbol{x} \in R^{MN}$, where M and N denote the image size. RPs can be represented by their chrominance values x_i^* and an index set Ω corresponding to their positions as follows,

$$x_i = \begin{cases} x_i^* & \text{if } i \in \Omega \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

In order to recover the chrominance image $\boldsymbol{u}$, Levin et al. have defined the cost function J as follows,

$$J(\boldsymbol{u}) = \|\boldsymbol{x} - A\boldsymbol{u}\|_2^2, \quad A = I - B, \tag{2}$$

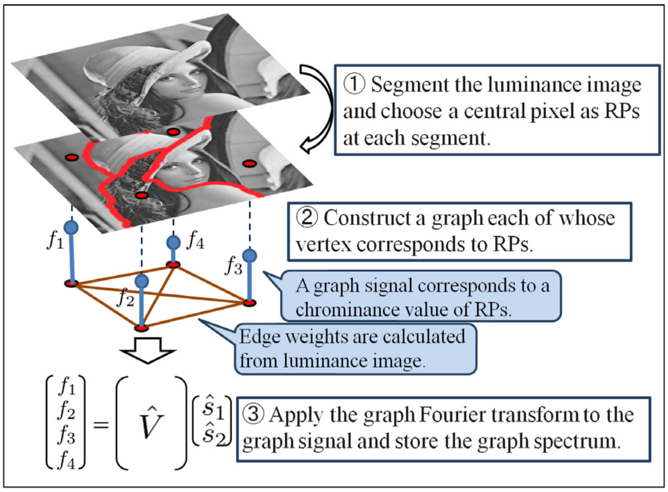

Fig. 2. Illustration of the proposed RPs coding algorithm based on the graph Fourier transform.

where $I \in R^{MN\times MN}$ is an identity matrix and $B \in R^{MN\times MN}$ is matrix containing the weighting components $b'_{i,j}$ defined by

$$b'_{i,j} = \begin{cases} 0 & \text{if } i \in \Omega \text{ or } j \notin N_i \\ b_{i,j} & \text{otherwise} \end{cases}, \tag{3}$$

$$b_{i,j} = \frac{1}{S_i}\exp\left(-\frac{(y_i - y_j)^2}{2\sigma_i^2}\right), \tag{4}$$

and N_i is the index set of 8-neighboring pixels of the ith pixel on an image. In the above equation, S_i is a normalization constant and equal to $\sum_{j\in N_i}\exp\left(-\frac{(y_i-y_j)^2}{2\sigma_i^2}\right)$, y_i denotes the ith value of $\boldsymbol{y}$, and σ^2 is a given variance. Then the chrominance image is recovered according to the following equation,

$$\boldsymbol{u} = C\boldsymbol{x}, \tag{5}$$

where $C = A^{-1} \in R^{MN\times MN}$, and we call C as Levin's colorization matrix.

2.2. Colorization-based image coding algorithm

Several colorization-based image coding algorithms have been proposed using Levin's colorization technique [4–6]. In [6], Ono et al. have proposed the algorithm based on the colorization (5). One of the contributions of Ono's algorithm is that the value of RPs are optimized in encoding phase such that the recovery error is minimized based on the following problem,

$$\min_{\boldsymbol{x}} \|\boldsymbol{u}^* - C\boldsymbol{x}\|_2^2 \quad \text{s.t. } x_i = 0, i \notin \Omega, \tag{6}$$

where $\boldsymbol{u}^* \in R^{MN}$ denote the original chrominance image. However the Ono's algorithm requires $\log_2 MN$ [bits] to represent the position for each RP.

Lee et al. have proposed a colorization-based image coding algorithm using sparse optimization and image segmentation technique in order to reduce the information about RPs [7]. They have proposed the following equation instead of the Levin's colorization (5),

$$\boldsymbol{u} = C_L\boldsymbol{x}_L, \tag{7}$$

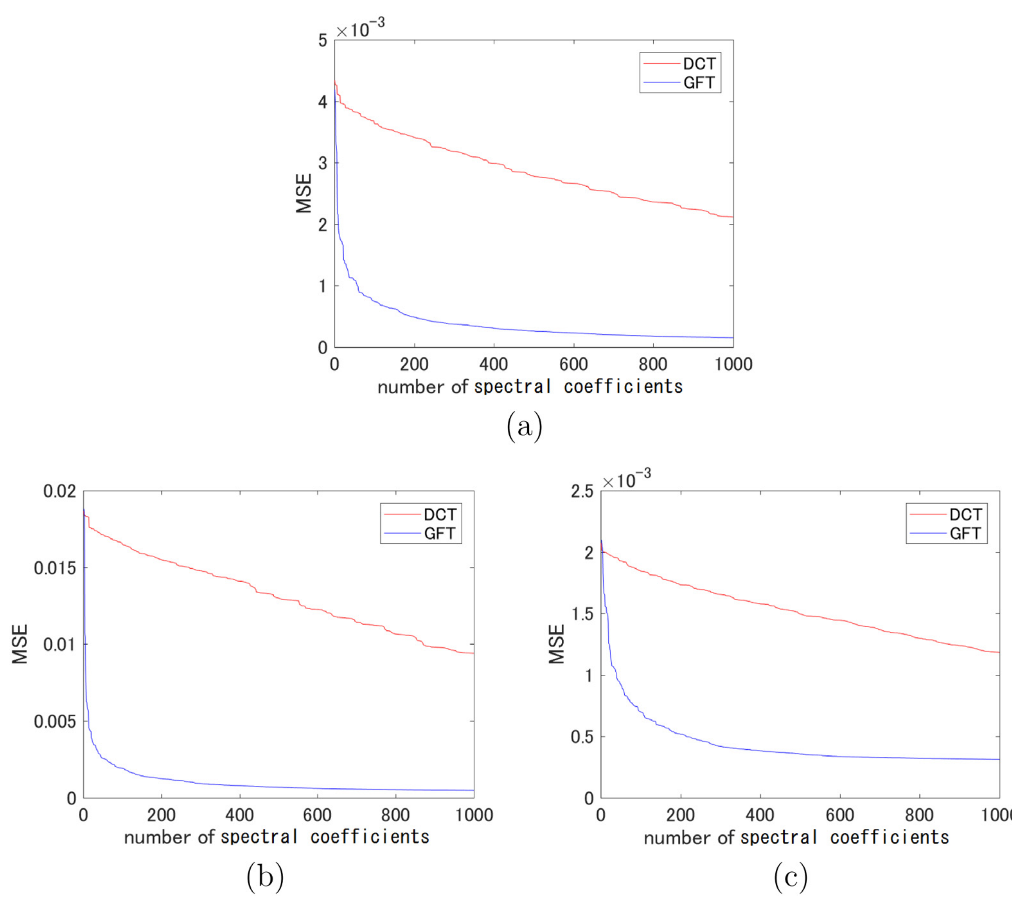

Fig. 3. Comparisons of the proposed GFT with DCT by MSE(mean squared error): (a) Lenna image, (b) Pepper image and (c) Airplane image as shown in Fig. 7.

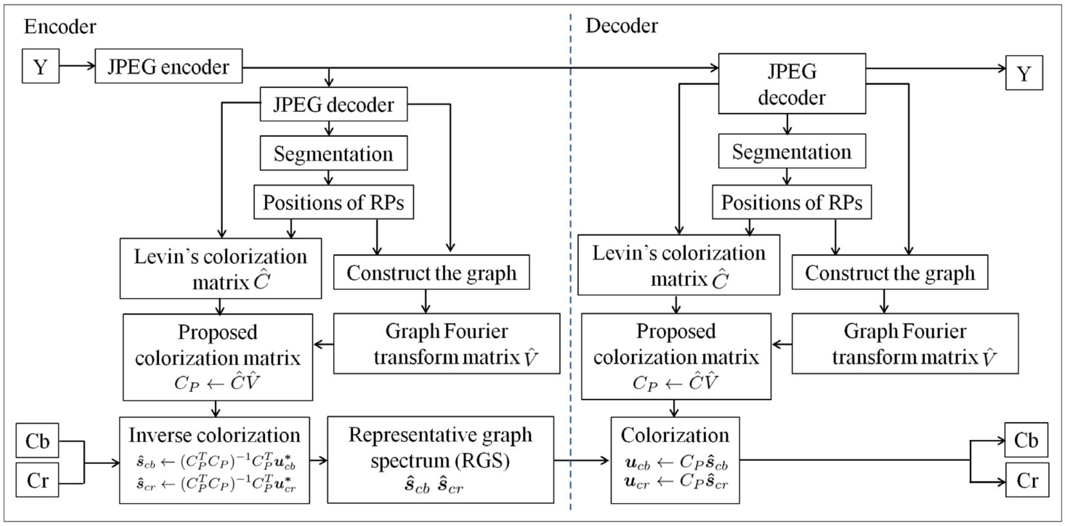

Fig. 4. The flowchart of the proposed colorization-based image coding algorithm using graph signal processing.

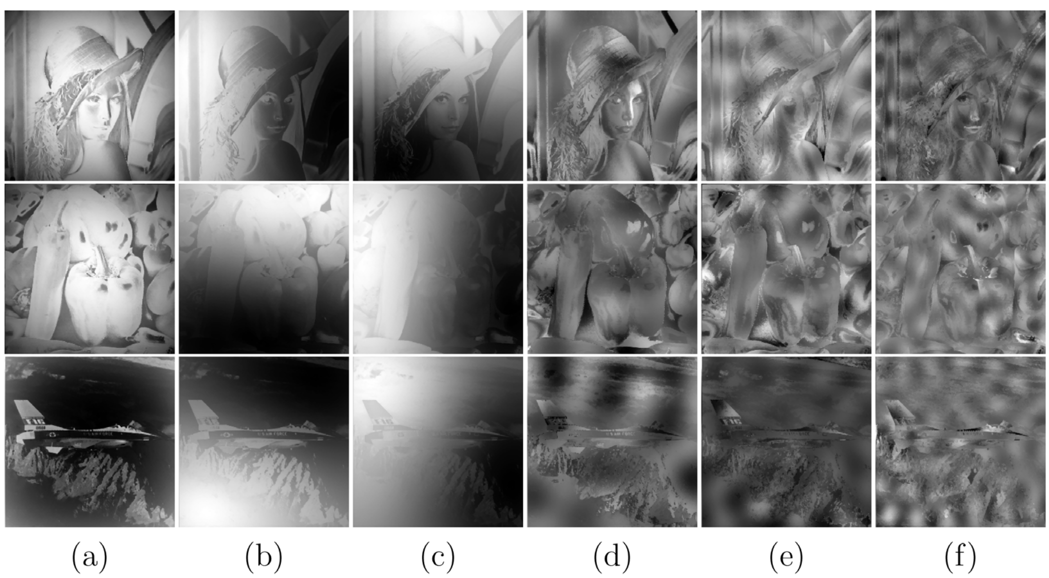

Fig. 5. The basic images of Lenna image (upper side), Pepper image (center side) and Airplane image (lower side) with (a) $i = 1$, (b) $i = 2$, (c) $i = 3$, (d) $i = 50$, (e) $i = 100$ and (f) $i = 200$.

where $C_L \in R^{MN\times K}$ and $\boldsymbol{x}_L \in R^K$ denote a new colorization matrix and RPs, respectively. In their algorithm, C_L is constructed from the luminance image such that it can be represented by less column vectors than Levin's colorization matrix C, that is, less number of RPs are required. The constructing of C_L is based on the image segmentation of the luminance image, and each column vector of C_L represents a sharp edges of each object. In encoding phase, the algorithm selects the small number of columns in C_L based on the following sparse optimization problem,

$$\begin{aligned}&\min_{\boldsymbol{x}_L} \|\boldsymbol{u}^* - C_L\boldsymbol{x}_L\|_2^2\\ &\text{s.t.}\ \ \|\boldsymbol{x}_L\|_0 \le \ell,\end{aligned} \tag{8}$$

where $\boldsymbol{u}^* \in R^{MN}$ and ℓ are the original chrominance image and a given constant, respectively, and $\|\cdot\|_0$ denotes the l_0 norm of a vector. Then ℓ compressed values of RPs and their positions are stored, and, in decoding phase, the chrominance image is recovered using Eq. (7). The advantage of this algorithm is that $\log_2 K$ [bits] is required to store the positions of RPs and that high coding performance is achieved as compared with the other colorization-based image coding algorithms and JPEG2000.

The problems (6) and (8) gives the sparse vectors as the solution. Although, in generally, it is advantage that the data is represented as sparse in the data compression, there is room for improvement because we are required to store the positions of non-zero elements. That is, it is the best if the colorization matrix has only a few column vectors and these vectors can represent a chrominance image well.

2.3. Graph Fourier transform

In order to compress the RPs located inhomogeneously on an image, this paper uses the Fourier transform of functions on a graph [12–15], and this subsection gives its brief description.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, W\}$ denote an undirected connected weighted graph consisting of a finite set of vertices $\mathcal{V}$ with $|\mathcal{V}| = P$, a set of edges $\mathcal{E}$, and a weighted adjacency matrix W whose (i,j)th element $w_{i,j}$ represents the weight of the edge $e = (i,j)$ connecting vertices i and j. The degree matrix D is a diagonal matrix whose ith diagonal element d_i is equal to the sum of the weights of all the edges incident to vertex i, and the normalized graph Laplacian is defined as follows,

$$L = D^{-1/2}(D - W)D^{-1/2}. \tag{9}$$

Because L is a real symmetric matrix, it has a set of orthogonal eigenvectors. Let $\boldsymbol{v}_i, (i = 1,\dots,P)$ and $\lambda_i, (i = 1,\dots,P)$ denote eigenvectors

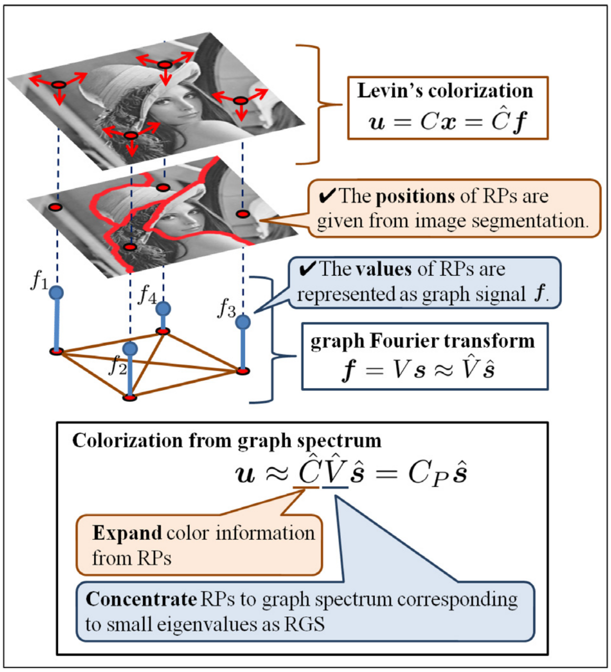

Fig. 6. Illustration of the proposed colorization technique from RGS.

and eigenvalues, respectively, which satisfy $L\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$ and $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_P$. Then we have the following equation,

$$L = V \Lambda V^T, \tag{10}$$

where $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_P]$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_P)$. The Fourier transform of the graph signal $\boldsymbol{f} \in R^P$ is defined by

$$\boldsymbol{s} = V^T \boldsymbol{f}, \tag{11}$$

where $\boldsymbol{s} \in R^P$ is called a graph spectrum, and the inverse graph Fourier transform is given as follows,

$$\boldsymbol{f} = V\boldsymbol{s}. \tag{12}$$

Here, this paper explains some characteristics of the eigenvectors and eigenvalues. We obtain the following equations concerned with kth eigenvector and eigenvalue,

$$\lambda_k = \boldsymbol{v}_k^T L \boldsymbol{v}_k = \frac{1}{2}\sum_{i,j} w_{i,j} \left(\frac{v_{k(i)}}{\sqrt{d_i}} - \frac{v_{k(j)}}{\sqrt{d_j}} \right)^2, \tag{13}$$

where $v_{k(i)}$ denotes ith element value of $\boldsymbol{v}_k$. The Eq. (13) gives the fact that $\frac{v_{1(i)}}{\sqrt{d_i}} = \frac{v_{1(j)}}{\sqrt{d_j}}$ for all combinations of i, j since $\lambda_1 = 0$. Therefore, we obtain $\boldsymbol{v}_1 = S[\sqrt{d_1}, \ldots, \sqrt{d_P}]^T$, where S is a normalization constant. Moreover, the elements of each eigenvector tend to have various values according to increasing eigenvalues because (13) implies that the kth eigenvalue is equal to the sum of the difference values of all entries of the kth eigenvector. The eigenvalues of the normalized Laplacian matrix satisfies $0 \leq \lambda_k \leq 2$, and $\lambda_P = 2$ is given when a graph is the bipartite graph. On the other hand, the fully graph connected by the same value for all weights gives the eigenvalues as $\lambda_2 = \cdots = \lambda_P = 1 + \frac{1}{P-1}$. This paper proposes an RPs compression method based on these characteristics.

3. Proposed algorithm

3.1. Representative pixels compression using graph Fourier transform

This paper focuses on the compression of RPs in order to achieve a high coding performance. As mentioned in Section 1, the performance of the colorization-based image coding algorithm depends on its colorization accuracy and compression rate of the information about RPs. To achieve high colorization accuracy with less information of RPs, this subsection proposes an RPs compression algorithm for a colorization-based image coding algorithm using an image segmentation technique and the graph Fourier transform.

First we consider the extraction method of RPs. Because colorization algorithm colorizes an image by expanding the color information from

Fig. 7. Test images: (a) Lenna, (b) Pepper, (c) Airplane, (d) Sailboat, (e) Milkdrop and (f) Earth.

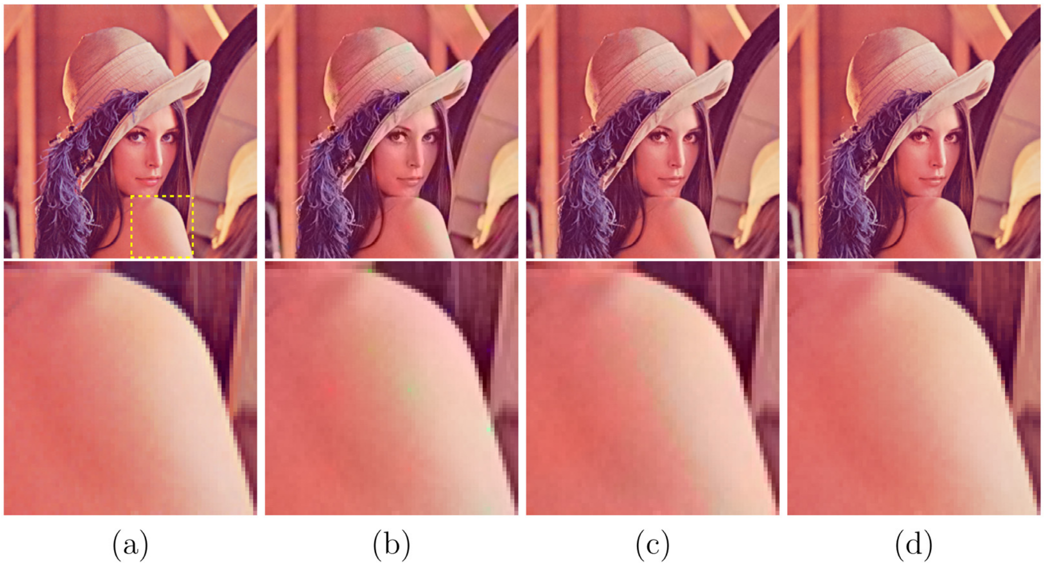

Fig. 8. Visual comparisons of Lenna image (upper side shows whole images, and lower side shows their zoomed images): (a) original image, (b) algorithm proposed in [6], (c) algorithm proposed in [7] and (d) proposed algorithm.

RPs, RPs are preferably to be in the center of objects in the image rather than in regions near the edges. To select desirable RPs, this paper introduces the image segmentation technique. In encoding phase, the state-of-the-art superpixel method proposed in [11] is applied to the luminance image, and the center pixel in each segment is extracted as RPs. The advantage of this method is that the volume of information storing the positions of RPs is not required because we can obtain the positions of RPs by applying the same segmentation method to the same luminance image in decoding phase. Note that the accuracy of this segmentation algorithm is lowered by applying it to a luminance image instead of a color image. In particular, this lowering of accuracy is caused at edge regions. However, it is not serious problem because the purposes of the segmentation are to select the positions of RPs from the pixels except for the edges and to select the center pixels of each segment. That is, a center pixel of each region is rarely located at edge even when the segmentation is failed at edge regions.

Next we focus on the method to store the chrominance values of RPs. Let $\bar{\Omega} = \{(m_i, n_i) | i = 1, \ldots, P\}$ and $X^*_{m_i,n_i}$ respectively denote the set of coordinates of RPs and a chrominance value of the (m_i, n_i)th pixel, where P is the number of RPs. Constructing the graph where the pixels in $\bar{\Omega}$ are used as its vertices, this paper provides a method to compress the chrominance values of RPs as the graph signal. Let us define the graph signal $\boldsymbol{f} \in R^P$ as follows,

$$\boldsymbol{f} = [X^*_{m_1,n_1}, X^*_{m_2,n_2}, \ldots, X^*_{m_P,n_P}]^T. \tag{14}$$

Then, in order to compress the graph signal effectively, we consider the weighted adjacency matrix W such that the graph signal is represented by a few eigenvectors of the graph Laplacian. This paper makes an assumptions that the pixels have similar chrominance values when their distance is small or when they have similar luminance values. It is not guaranteed that all images satisfy this assumption. However, as shown in numerical examples of Section 4, this assumption is appropriately for general images. Based on this assumption, we define a graph of RPs whose (i, j)th element of the weighted adjacency matrix as follows,

$$w_{i,j} = \exp(-\alpha \Delta d_{(i,j)}) \exp(-\beta \Delta y_{(i,j)}), \tag{15}$$

where α and β are given constants. In this equation, $\Delta d_{(i,j)}$ and $\Delta y_{(i,j)}$ denote the distance between vertices and the difference of the luminance values between vertices defined by

$$\Delta d_{(i,j)} = \sqrt{\left(\frac{m_i}{M} - \frac{m_j}{M}\right)^2 + \left(\frac{n_i}{N} - \frac{n_j}{N}\right)^2}, \tag{16}$$

and

$$\Delta y_{(i,j)} = |Y_{m_i,n_i} - Y_{m_j,n_j}|. \tag{17}$$

In (17), $Y_{m_i,n_i} \in [0, 1]$ denotes the luminance value of the (m_i, n_i)th pixel. Based on the equations from (9) to (11), the graph signal $\boldsymbol{f}$ is transformed to its graph spectrum $\boldsymbol{s}$.

Next we discuss the characteristics of the graph constructed by Eq. (15). The Eq. (15) gives similar weights $w_{i,j}$ close to 1 when ith and jth vertices are in the same object of the image since the luminance values in the same object are similar. Therefore, the non-zero eigenvalues of the normalized Laplacian matrix consisting of the vertices in the same object is approximated to $1 + \frac{1}{P^*-1}$, where P^* is the number of vertices in one object, because it is a fully connected graph in which weights of all edges are similar to each other. If we set the weights $w_{i,j} = 0$ when ith and jth vertices are in different objects, the eigenvalues of the normalized Laplacian matrix consisting of the vertices on the whole image is given as $\lambda_1 = \cdots = \lambda_{ob} = 0$ and $\lambda_{ob+1} \simeq \cdots = \lambda_P \simeq 1 + \frac{1}{P^*-1}$, where ob denotes a number of objects and regions in an image. However, Eq. (15) makes the values of $w_{i,j}$ larger than 0 even when ith and jth vertices are in different objects. Hence only one of eigenvalues of the normalized Laplacian matrix constructed by the proposed weight (15) is equal to 0, most of them are equal to $1 + \frac{1}{P^*-1}$, and the others have small value.

Fig. 1 shows an example of the eigenvalues of graph Laplacian, the absolute values of its corresponding graph spectrum transformed from the graph with Cb values of all RPs, where the parameter of superpixel segmentation algorithm is selected such that the number of segments is 12 000. $\alpha = 3.5$ and $\beta = 2.5$ are used for the weight (15). We can see that there are a single 0, several small eigenvalues, and a lot of large eigenvalues close to 1. This results agree with the above discussion, and we can assume that the number of small eigenvalues represents the number of the objects and regions in image. On the other hand, the values of spectrum tend to decrease according to increasing eigenvalues, and there is no spectrum having significantly large value except for low eigenvalues. Therefore this paper proposes an RPs compression algorithm to store the information of the graph spectrum corresponding to the c smallest eigenvalues. Here we call the graph spectrum corresponding to the c smallest eigenvalues as a representative graph spectrum (RGS), which is denoted as a c-dimensional vector $\hat{\boldsymbol{s}} \in R^c$.

In decoding phase, we can construct the same graph of the encoding phase from the luminance image, and the graph Laplacian eigenvectors

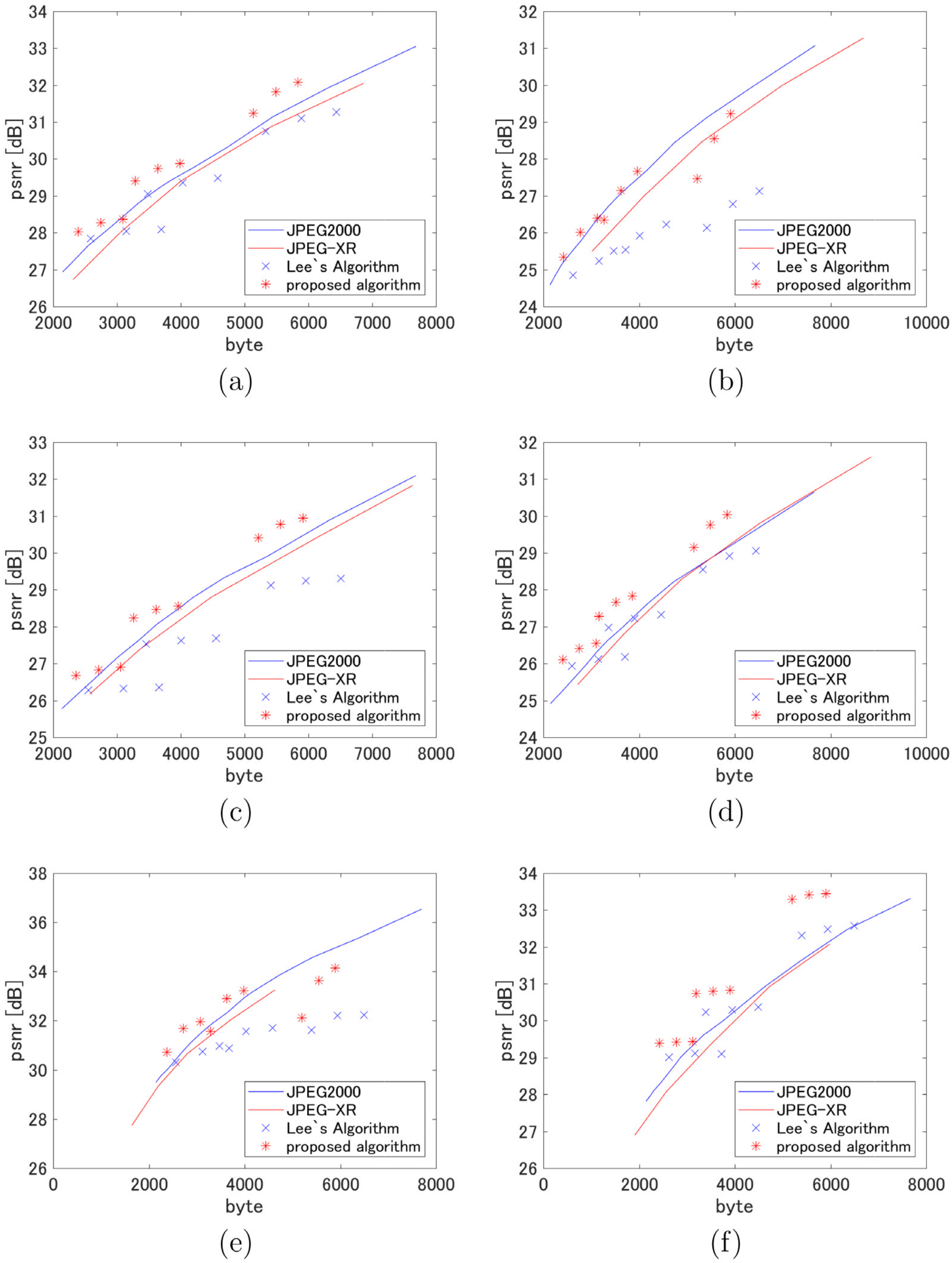

Fig. 9. Comparisons of the proposed algorithm with JPEG2000, JPEG XR and Lee's algorithm by PSNR: (a) Lenna image, (b) Pepper image, (c) Airplane image, (d) Sailboat image, (e) Milkdrop and (f) Earth image.

corresponding to the c small eigenvalues are calculated. Then the stored RGS is transformed to the chrominance values of RPs using the inverse graph Fourier transform (12). Finally, RPs compression method based on the graph Fourier transform is proposed as illustrated in Fig. 2.

Here we confirm that the chrominance values of RPs can be represented by a few basis of GFT in Fig. 3, which shows the relationship of the mean squared error (MSE) and the number of the stored spectrum. Note that MSE is calculated by the original chrominance values of RPs and recovered chrominance values of RPs. As can be seen, a few basis of the proposed GFT can represent the chrominance values of RPs appropriately, that is, the proposed GFT is appropriate basis to store the RPs. Fig. 3 also shows the result of the DCT based RPs compression. The DCT based RPs compression algorithm is that the selected RPs are organized as a matrix, and the chrominance values are transformed to the spectrum by the 2D-DCT basis. We can see that the proposed GFT compresses the chrominance values of RPs more effectiveness than DCT basis.

The colorization-based image coding is achieved by using the Levin's colorization technique and RPs compression algorithm proposed in this subsection. In order to improve the coding performance, next subsection modifies Levin's colorization (5) and provides a new equation which enables us to directly transform the RGS to the chrominance image, that is, a new colorization technique from RGS is proposed based on the graph Fourier transform.

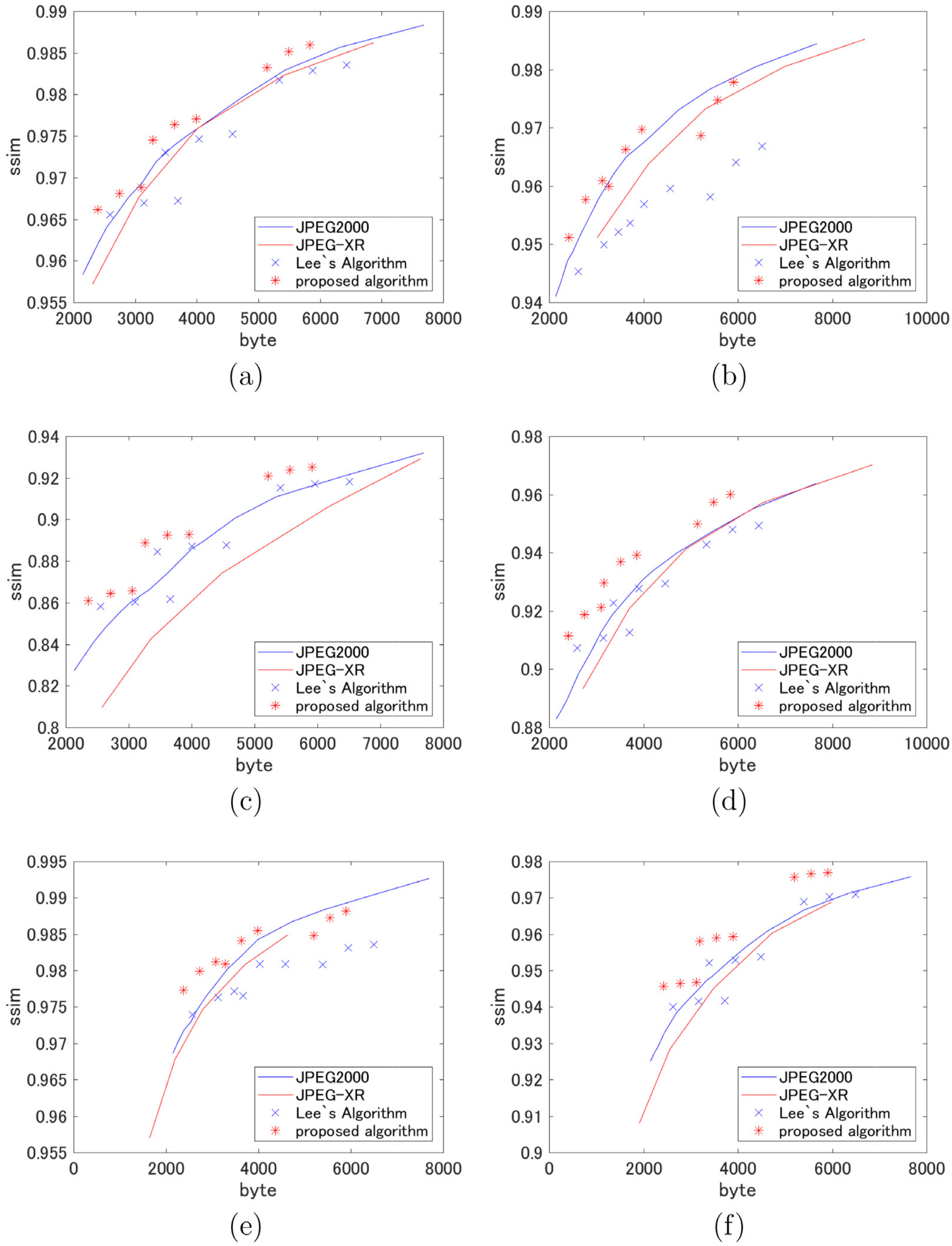

Fig. 10. Comparisons of the proposed algorithm with JPEG2000, JPEG XR and Lee's algorithm by SSIM: (a) Lenna image, (b) Pepper image, (c) Airplane image, (d) Sailboat image, (e) Milkdrop and (f) Earth image.

3.2. Colorization-based image coding using graph Fourier transform

This subsection introduces a new colorization technique directly using RGS based on the graph Fourier transform, and a colorization-based image coding algorithm is proposed. Let $\hat{C} \in R^{MN\times P}$ denote the submatrix consisting of the ith column vectors of Levin's colorization matrix C for all $i \in \Omega$. Then we have the following equations from Levin's colorization (5),

$$\boldsymbol{u} = C\boldsymbol{x} = \hat{C}\boldsymbol{f}. \tag{18}$$

Let $\hat{V} \in R^{P\times c}$ denote the submatrix consisting of the ith column vectors of V for all $i \in \{1, 2, \ldots, c\}$. Then we have that

$$\boldsymbol{f} = V\boldsymbol{s} \approx \hat{V}\hat{\boldsymbol{s}}. \tag{19}$$

Substituting (19) to (18), we obtain the following equation,

$$\boldsymbol{u} \approx \hat{C}\hat{V}\hat{\boldsymbol{s}}. \tag{20}$$

Let $C_P = \hat{C}\hat{V}$, and then we have that

$$\boldsymbol{u} \approx C_P\hat{\boldsymbol{s}}. \tag{21}$$

Based on the above approximate equation, this paper provides a colorization-based image coding algorithm as shown in Algorithms 1

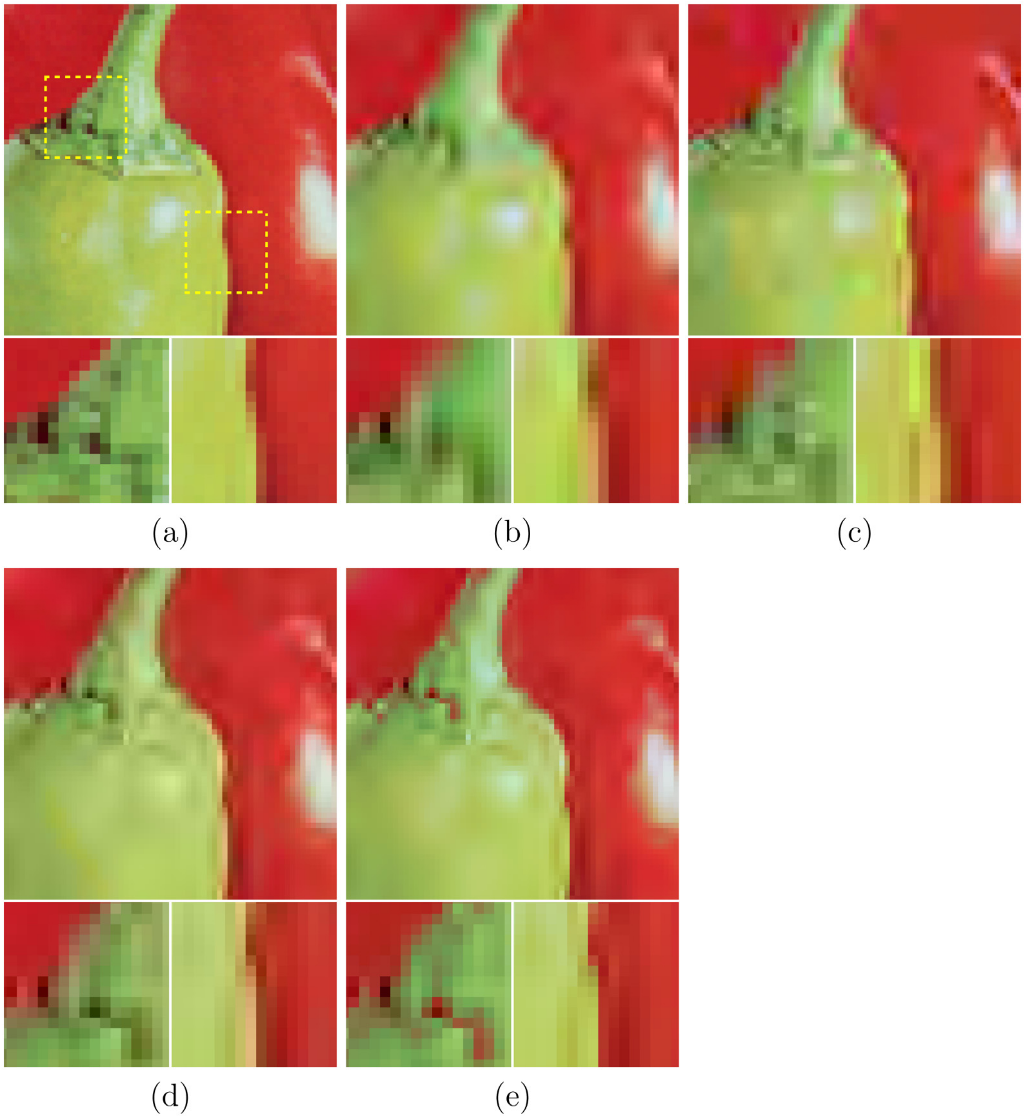

Fig. 11. Visual comparisons of Pepper image (zoom in): (a) original image, (b) JPEG2000, (c) JPEG XR, (d) algorithm proposed in [7] and (e) proposed algorithm.

and 2, where $\boldsymbol{u}^*_{cb}$ and $\boldsymbol{u}^*_{cr}$ denote the original chrominance images, and $\hat{\boldsymbol{s}}_{cb}$ and $\hat{\boldsymbol{s}}_{cr}$ denote the RGS of $\boldsymbol{u}^*_{cb}$ and $\boldsymbol{u}^*_{cr}$, respectively. In encoding phase, we construct the matrix C_P from a luminance image, and the following problem is considered in order to obtain the optimal RGS minimizing the recovery error derived from

$$\min_{\hat{\boldsymbol{s}}} \|\boldsymbol{u}^* - C_P \hat{\boldsymbol{s}}\|_2^2. \tag{22}$$

Because C and V are full-rank matrices, $\hat{C}$ and $\hat{V}$ are column full-rank matrices. Therefore C_P is a column full-rank matrix, and the solution of the above problem is calculated as follows,

$$\hat{\boldsymbol{s}} = (C_P^T C_P)^{-1} C_P^T \boldsymbol{u}^*. \tag{23}$$

Then RGS $\hat{\boldsymbol{s}}$ is stored. In decoding phase, C_P is constructed from the luminance image, and the chrominance image is recovered from Eq. (21). Fig. 4 shows the flowchart of the proposed algorithm.

Algorithm 1 Encoding algorithm.

Require: $\boldsymbol{y}$, $\boldsymbol{u}^*_{cb}$, $\boldsymbol{u}^*_{cr}$, α, β and c

Generate the Levin's colorization matrix C using a luminance image.

Segment the luminance image, and generate the weighted adjacency matrix using (15).

Generate the graph Fourier transform matrix V from graph Laplacian L.

Generate the submatrices $\hat{C}$ and $\hat{V}$ from C and V.

$C_P \leftarrow \hat{C}\hat{V}$.

$\hat{\boldsymbol{s}}_{cb} \leftarrow (C_P^T C_P)^{-1} C_P^T \boldsymbol{u}^*_{cb}$.

$\hat{\boldsymbol{s}}_{cr} \leftarrow (C_P^T C_P)^{-1} C_P^T \boldsymbol{u}^*_{cr}$.

Ensure: $\hat{\boldsymbol{s}}_{cb}$ and $\hat{\boldsymbol{s}}_{cr}$.

3.3. Volume of information to store representative graph spectrum

The range of values of RGS $\hat{s}_i$ $(i = 1, \dots, c)$ is large, which can be seen in Fig. 1. Hence, in order to store the values of spectrum appropriately, we take the logarithm of each value of spectrum as follows,

$$\bar{s}_i = \log(|\hat{s}_i| + 1). \tag{24}$$

Then each value of $\bar{s}_i$ for $i = 1, \dots, c$ is stored. The following volume of information Q [bits] is required to restore $\hat{s}_i$ $(i = 1, \dots, c)$,

$$Q = 2\{cq_1 + q_2 + c\}, \tag{25}$$

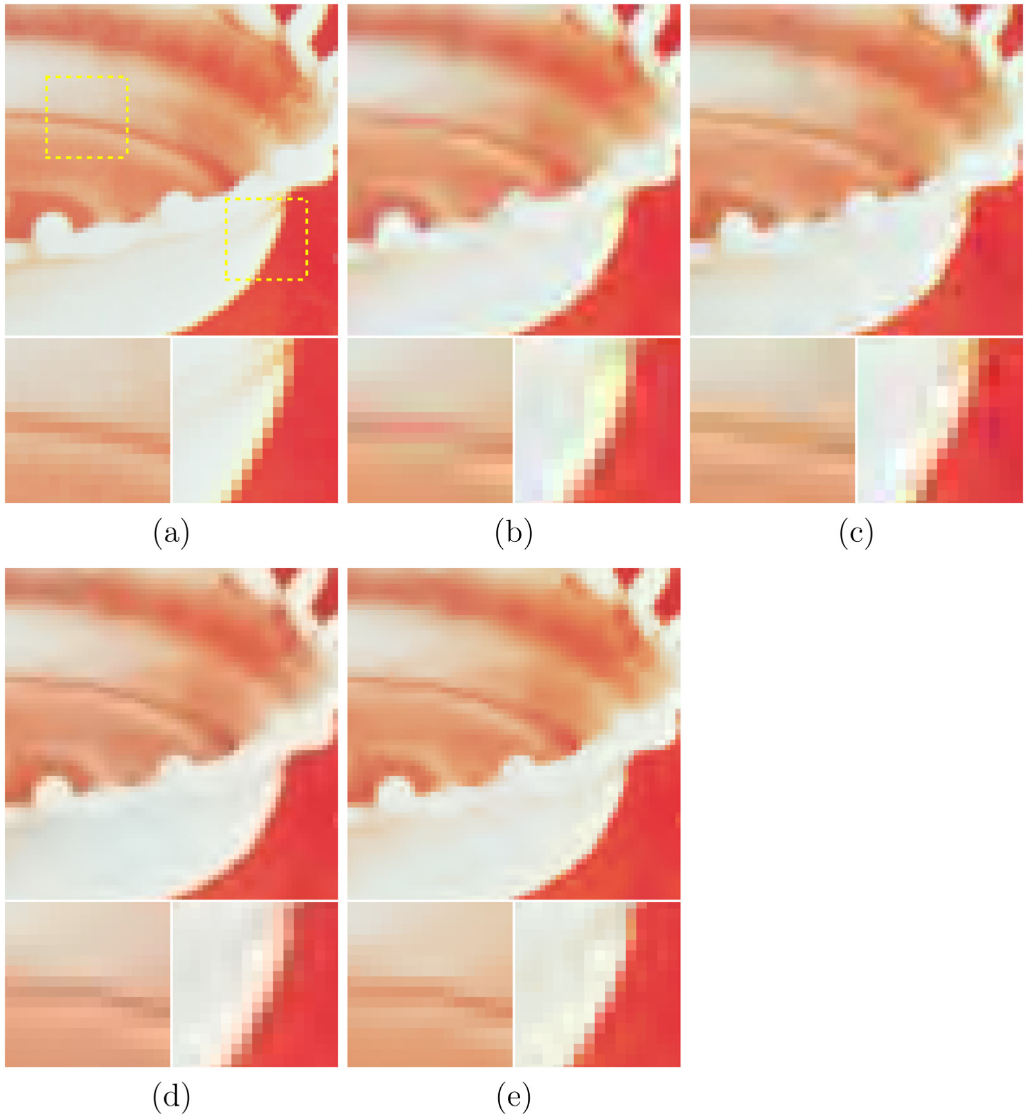

Fig. 12. Visual comparisons of Milkdrop image (zoom in): (a) original image, (b) JPEG2000, (c) JPEG XR, (d) algorithm proposed in [7] and (e) proposed algorithm.

Algorithm 2 Decoding algorithm.

Require: $\boldsymbol{y}$, $\hat{\boldsymbol{s}}_{cb}$, $\hat{\boldsymbol{s}}_{cr}$, α, β and c
Generate the Levin's colorization matrix C using a luminance image.
Segment the luminance image, and generate the weighted adjacency matrix using (15).
Generate the graph Fourier transform matrix V from graph Laplacian L.
Generate the submatrices $\hat{C}$ and $\hat{V}$ from C and V.
$C_P \leftarrow \hat{C}\hat{V}$.
$\boldsymbol{u}_{cb} \leftarrow C_P\hat{\boldsymbol{s}}_{cb}$.
$\boldsymbol{u}_{cr} \leftarrow C_P\hat{\boldsymbol{s}}_{cr}$.
Ensure: $\boldsymbol{u}_{cb}$ and $\boldsymbol{u}_{cr}$.

where q_1 and q_2 denote the quantization bit size to represent the value of each spectrum and the maximum value of graph spectrum, respectively. The third term of the right-hand side of (25) represents the sign of each $\hat{s}_i$. The minimum value of graph spectrum is not required to store because it can be assumed to be zero. In decoding phase, the spectrum is recovered by the following equation,

$$\hat{s}_i = sign(\hat{s}_i)\exp(\bar{s}_i) - 1, \tag{26}$$

where $sign(\cdot)$ denotes the signum function.

Next we investigate how the proposed colorization matrix represents an image. The recovered chrominance image $\boldsymbol{u}_{cb}$ is written as follows,

$$\begin{aligned}\boldsymbol{u}_{cb} &= C_P\hat{\boldsymbol{s}}_{cb}\\ &= \hat{C}\hat{V}\hat{\boldsymbol{s}}_{cb}\\ &= \sum_{i=1}^{c}\hat{s}_{cb_i}\hat{C}\boldsymbol{v}_i,\end{aligned} \tag{27}$$

where $\hat{s}_{cb_i}$ denotes ith value of $\hat{\boldsymbol{s}}_{cb}$. We can see that the colorized image is represented by the linear combination of $\hat{C}\boldsymbol{v}_i$, which is called a basic image in this paper. As mentioned in the last paragraph of Section 2, the elements of an eigenvector corresponding to a low eigenvalue tend to have similar values, and hence this eigenvector gives smooth basic image. By contrast, non-smooth basic images are given by eigenvector corresponding to a high eigenvalue. Fig. 5 shows the basic images $\hat{C}\boldsymbol{v}_i$ for $i = \{1, 2, 3, 50, 100, 200\}$ for three images, where the values of basic images are adjusted in the range of $[0, 1]$. We can see that the basic images have the following two characteristics.

(i) Each basic image has a wavy pattern based on the graph, and the basic image of higher eigenvalues has higher frequency waves. The recovered chrominance image is represented as the sum of various wavy pattern images.

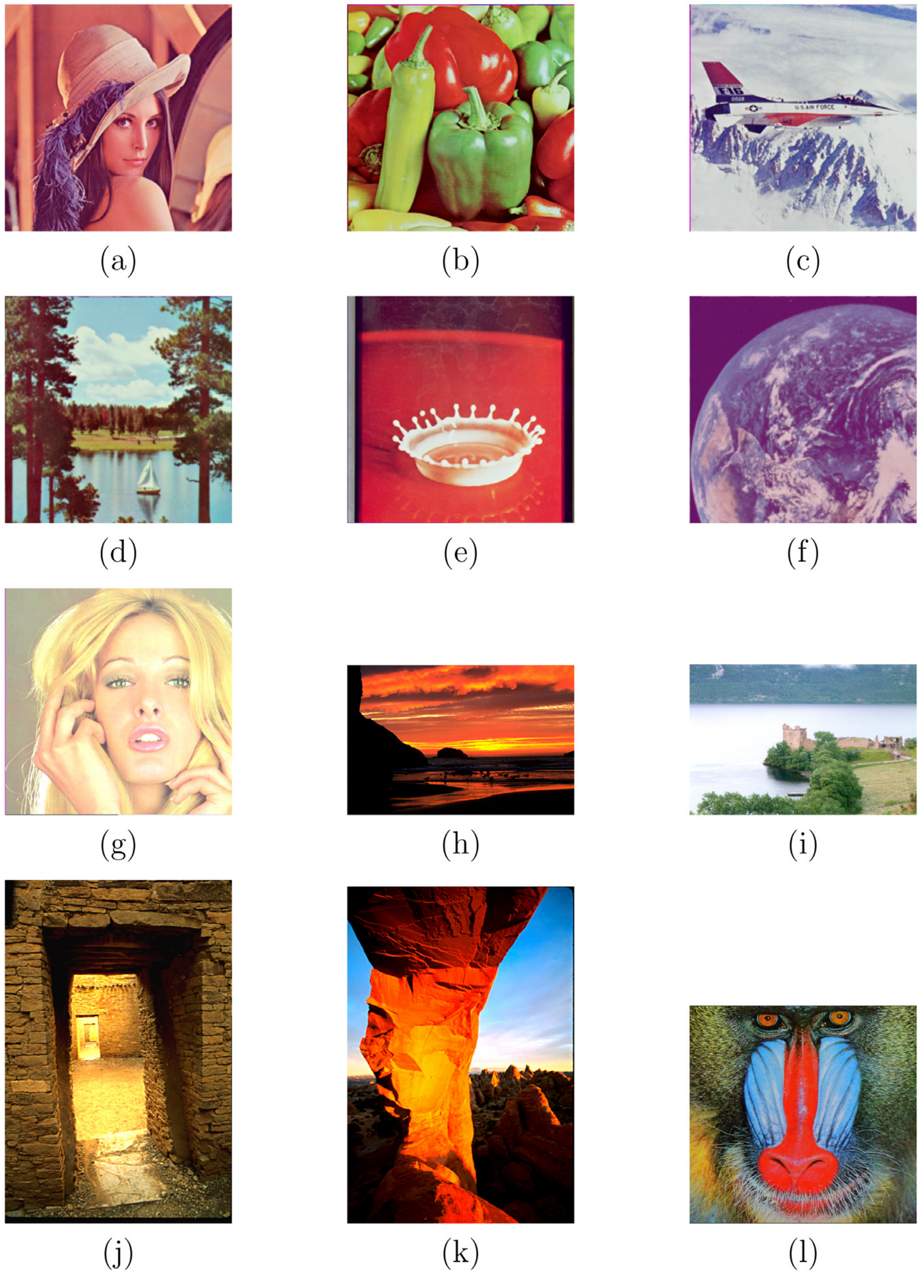

Fig. 13. Test images for the comparison using large size image: (a) Lenna (512 × 512), (b) Pepper (512 × 512), (c) Airplane (512 × 512), (d) Sailboat (512 × 512), (e) Milkdrop (512 × 512), (f) Earth (512 × 512), (g) Tiffany (512 × 512), (h) Bandon (403 × 610), (i) Lochness (559 × 841), (j) Pueblo bonito (610 × 403), (k) Skyline arch (594 × 400) and (l) Baboon (480 × 500).

(ii) Each basic image has the edges of objects, and therefore, the recovered chrominance image can represent the sharp edges of each object.

The first characteristic is analogous to that of image coding algorithms in frequency domain such as JPEG and JPEG2000, the second one is similar to that of Lee's algorithm, and hence the proposed colorization matrix C_P has almost both characteristics of these algorithms. As shown in [7], the performance of Lee's algorithm is improved by utilizing the wavelet basis vectors to construct C_L, which implies that the proposed algorithm is expected to achieve good performance.

Now we focus on the differences of three colorization matrices Levin's C in (5), Lee's C_L in (7) and the proposed C_P in (21), which are constructed from a luminance image and transform the RPs or RGS to a chrominance image. While C and C_L are designed to represent the chrominance image as the small number of RPs, C_P is designed to require a few information using a graph spectrum based compression technique, and RGS requires less information to store the chrominance values of RPs than the same number of RPs of C and C_L since the chrominance values are compressed for generating RGS. In the other words, the proposed algorithm can represent the greater number of RPs than Lee's algorithm when the number of RGS is equal to that of Lee's RPs. Furthermore, since the proposed algorithm selects the same RPs from the same luminance image both in encoding and decoding phase, the information of RPs' positions is not stored. Therefore the proposed algorithm requires less information to store RGS than the same number of RPs of Lee's algorithm and achieves high colorization performance. This is the major characteristic and the advantage of the proposed colorization algorithm. Fig. 6 shows the illustration of the proposed colorization from RGS.

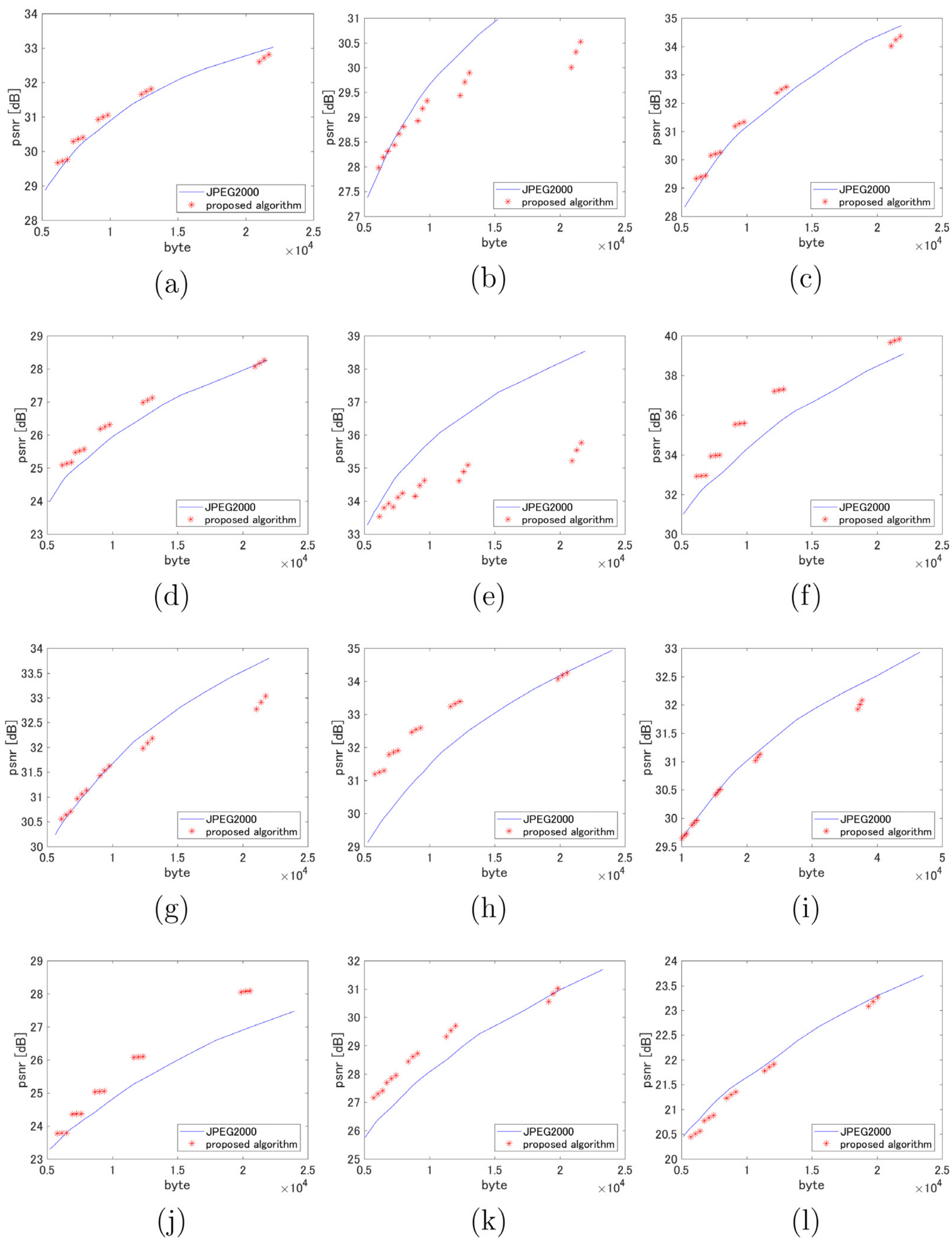

Fig. 14. Comparisons of the proposed algorithm with JPEG2000 by PSNR: (a) Lenna image, (b) Pepper image, (c) Airplane image, (d) Sailboat image, (e) Milkdrop, (f) Earth image, (g) Tiffany image, (h) Bandon image, (i) Lochness image, (j) Pueblo bonito image, (k) Skyline arch and (l) Baboon image.

4. Numerical examples

This section provides numerical examples to show the effectiveness of the proposed algorithm. Except for the last experiment, we use test images as shown in Fig. 7, all of which are 256 × 256 size images. In order to evaluate the quality of image compression, we measure the differences between an original image and recovered images using peak signal to noise ratio (PSNR) and the structural similarity (SSIM) [17]. To calculate them, the recovered chrominance images and the luminance image are transformed to the RGB color image, and they are compared using MATLAB functions `psnr` and `ssim`. All source codes except for JPEG XR are implemented in MATLAB 2017a on a PC with an Intel Core i7 3.0 GHz CPU, 128 GB of RAM memory. We use Pixillion Image Converter software for JPEG XR. All experiments use $\alpha = 3.5$, $\beta = 2.5$, $q_1 = 6$ and $q_2 = 12$. The proposed algorithm utilizes the image segmentation algorithm proposed in [11] with $\gamma = 12000$ and $\delta = 10$, which are parameters to determine the number of segments and to balance color similarity and spatial proximity, respectively. These parameters are chosen empirically based on numerical experiments whose results are shown in Table 1. In these experiments, Lenna image is used, the value of RPs is set to 200, and the luminance image is uncompressed.

The results show that the above parameters give the highest coding performance of several set of the parameters. Because the same values of parameters are used for all images in encoding and decoding phases, these values are not required to be stored.

This paper shows the coding performance of the proposed algorithm comparing with Ono's algorithm [6] and Lee's algorithm [7]. In order

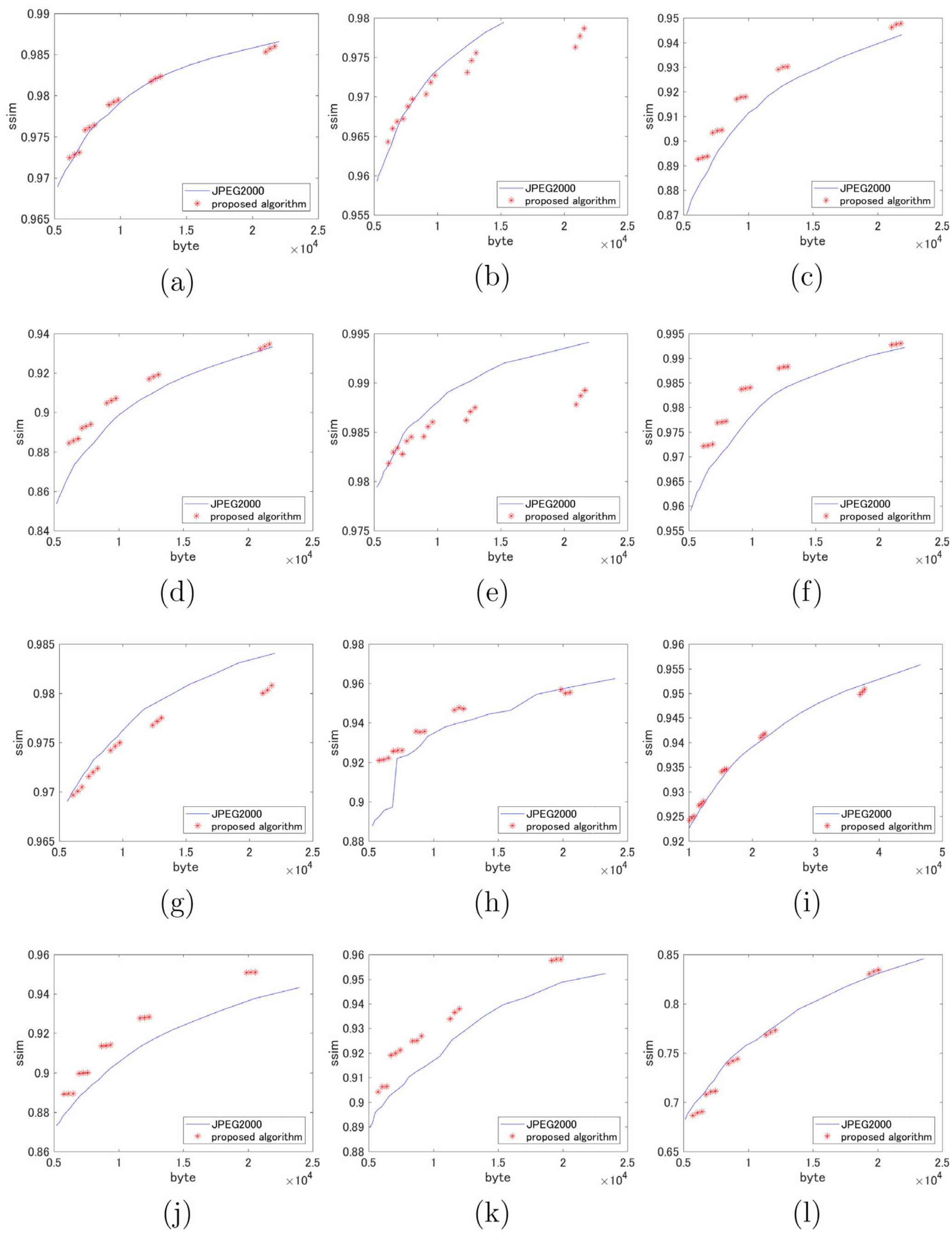

Fig. 15. Comparisons of the proposed algorithm with JPEG2000 by SSIM: (a) Lenna image, (b) Pepper image, (c) Airplane image, (d) Sailboat image, (e) Milkdrop, (f) Earth image, (g) Tiffany image, (h) Bandon image, (i) Lochness image, (j) Pueblo bonito image, (k) Skyline arch and (l) Baboon image.

to show the performance of colorization-based image coding, these algorithms use uncompressed luminance image. Because Ono's algorithm cannot be fixed the number of RPs exactly, the number of RGS in the proposed algorithm and that of RPs in Lee's algorithm are set to be equal to the number of RPs in Ono's algorithm. Table 2 shows the number of RPs/RGS, the volume of information to store the RPs/RGS, PSNR and SSIM. As can be seen, the proposed algorithm achieves the best recovery performance of these algorithms. Furthermore, the proposed algorithm requires the lowest volume of information storing the RPs/RGS. Fig. 8 shows the recovery images of Lenna image and their zoomed images. We can see that the result of Lee's algorithm has redundant edge on smooth regions and that the result of Ono's algorithm has some false-color pixels while the proposed algorithm gives smooth color gradation on smooth regions and has no false-color pixels. However, the calculation time of the proposed algorithm is higher than the other algorithms because the calculation of the eigenvector requires a high calculation cost. In encoding, the calculation time of the algorithm [6,7] and the proposed algorithm are required about 30, 40 and 140 [sec], respectively. In decoding, the calculation time of the algorithm [6,7] and the proposed algorithm are required about 2, 20 and 140 [sec], respectively.

Next the proposed algorithm is compared with JPEG2000 coding, JPEG XR and Lee's algorithm. We use `imwrite` function of MATLAB for JPEG2000, where the parameter of `compression ratio` is set as $[25, 100]$ with interval of 5, and other parameters are used as default values. The luminance image of the proposed algorithm and Lee's algorithm is compressed by JPEG2000, where the parameter of `compression ratio` is set as $\{13, 23, 33\}$, and the number of RPs/RGS is set as $\{200, 400, 600\}$. Fig. 9 shows the relationship of PSNR and the volume of information to

Table 1
Results using some parameters.

The value of α ($\beta = 2.5$ and $\gamma = 12\,000$ are fixed)	PSNR [dB]
0.35	32.81
3.5	**34.01**
35	32.11
The value of β ($\gamma = 12\,000$ and $\alpha = 3.5$ are fixed)	PSNR [dB]
0.25	33.44
2.5	**34.01**
25	31.79
The number of segments γ ($\alpha = 3.5$ and $\beta = 2.5$ are fixed)	PSNR [dB]
6 000	33.96
12 000	**34.01**
18 000	34.00

Table 2
Comparison of colorization-based coding algorithms using uncompressed luminance image.

Image	Algorithm	RPs/RGS	Byte	PSNR [dB]	ssim
Lenna	Ono's algorithm [6]	242	968	31.17	0.9842
	Lee's algorithm [7]	242	666	33.20	0.9896
	Proposed algorithm	242	**427**	**34.44**	**0.9918**
Pepper	Ono's algorithm [6]	240	960	24.81	0.9460
	Lee's algorithm [7]	240	660	27.17	0.9678
	Proposed algorithm	240	**423**	**28.99**	**0.9787**
Airplane	Ono's algorithm [6]	234	936	28.00	0.9349
	Lee's algorithm [7]	234	644	31.61	0.9695
	Proposed algorithm	234	**413**	**34.42**	**0.9793**
Sailboat	Ono's algorithm [6]	248	992	28.03	0.9398
	Lee's algorithm [7]	248	682	31.04	0.9667
	Proposed algorithm	248	**437**	**32.46**	**0.9760**
Milkdrop	Ono's algorithm [6]	242	968	27.17	0.9538
	Lee's algorithm [7]	242	666	32.34	0.9835
	Proposed algorithm	242	**427**	**33.88**	**0.9904**
Earth	Ono's algorithm [6]	236	944	32.59	0.9741
	Lee's algorithm [7]	236	649	36.42	0.9872
	Proposed algorithm	236	**416**	**40.26**	**0.9944**
Average	Ono's algorithm [6]	240.3	961.3	28.63	0.9555
	Lee's algorithm [7]	240.3	661.2	31.96	0.9774
	Proposed algorithm	240.3	**423**	**34.07**	**0.9851**

store the RGB color image, and Fig. 10 shows the relationship of SSIM and the volume of information to store the RGB color image. Figs. 11 and 12 show the zoomed result images of Pepper and Milkdrop, where the volume of information to store image is about 3.6 Kbytes. As can be seen in Figs. 9 and 10, the proposed algorithm achieves the higher PSNR and SSIM than Lee's algorithm, JPEG2000 and JPEG XR coding in a lot of cases. Figs. 11 and 12 indicate that natural images are recovered by the proposed algorithm comparing with Lee's algorithm, JPEG2000 and JPEG XR coding. In particular, colors in the edge region of the proposed algorithm is clear. However, Fig. 11 shows that the results of the proposed algorithm have false-color pixels, that is, red color pixels are recovered while they should be green. This false-colorization is occurred because the proposed algorithm has assumed that the pixels have similar chrominance values when they have similar luminance values and because the luminance values of these false-color pixels is more similar to the red pepper on the background than the green pepper on the front side.

Finally, the proposed algorithm is compared with JPEG2000 coding using 12 large size images as shown in Fig. 13 (we use some images available at: https://www.petitcolas.net/watermarking/image_database/). $\gamma = 48\,000$ is used for the proposed algorithm, and the other parameters are set to the same parameters of the experiments for 256×256 size image. Fig. 14 shows the relationship of PSNR and the volume of information to store the RGB color image, and Fig. 15 shows the relationship of SSIM and the volume of information to store the RGB color image. As can be seen, in eight images except for Pepper, Milkdrop, Tiffany and Baboon images, the proposed algorithm shows the effectiveness.

5. Conclusions

This paper proposed the colorization-based image coding algorithm using the graph Fourier transform. We focused on that the chrominance values of RPs can be represented as a few graph spectrum based on the graph Fourier transform, and a new colorization technique using graph spectrum was proposed. Based on the proposed colorization technique, a new colorization-based image coding algorithm was proposed. Numerical results showed that the proposed algorithm achieves higher coding performance than some previous colorization-based image coding algorithms, JPEG2000 and JPEG XR coding.

Acknowledgments

We are grateful to Prof. Sukho Lee and his co-authors for providing the source code of the colorization-based coding proposed in [7]. This work was supported by JSPS KAKENHI Grant Numbers 17H07129.

References

[1] A. Levin, D. Lischinski, Y. Weiss, Colorization using optimization, ACM Trans. Graph. 23 (3) (2004) 689–694.
[2] L. Yatziv, G. Sapiro, Fast image and video colorization using chrominance blending, IEEE Trans. Image. Process. 15 (5) (2006) 1120–1129.
[3] S. Iizuka, E.S. Serra, H. Ishikawa, Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification, ACM Trans. Graph. 35 (4) (2016) 1101–1111.
[4] L. Cheng, S. Vishwanathan, Learning to compress images and videos, in: ACM Proc. of the 24th Int. Conf. Maching Learning (ICML), 2007, pp. 161–168.
[5] T. Miyata, Y. Komiyama, Y. Inazumi, Y. Sakai, Novel inverse colorization for image compression, in: Proc. IEEE Picture Coding Symp, 2009, pp. 1–4.
[6] S. Ono, T. Miyata, Y. Sakai, Colorization-based coding by focusing on characteristics of colorization bases, in: Proc. IEEE Picture Coding Symp, 2010, pp. 230–233.
[7] S. Lee, S. Park, P. Oh, M. Kang, Colorization-based compression using optimization, IEEE Trans. Image Process. 22 (7) (2013) 2627–2636.
[8] K. Mishiba, T. Yoshitome, Colorization matrix construction with high compression efficiency for colorization-based coding using optimization, in: Proc. IEEE Int. Conf. Image Process, 2014, pp. 5551–5555.
[9] K. Uruma, K. Konishi, T. Takahashi, T. Furukawa, Fast colorization based image coding algorithm using multiple resolution images, EURASIP J. Image Video Process. 2016 (7) (2016) 1–15.
[10] P. Peter, L. Kaufhold, J. Weickert, Turning diffusion-based image colorization into efficient color compression, IEEE Trans. Image Process. 26 (2) (2017) 860–869.
[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.
[12] F.R.K. Chung, Lectures on Spectral Graph Theory, CBMS Lectures, Fresno, 1996.
[13] U. Von Luxburg, A tutorial on spectral clustering, Statist. Comput. 17 (4) (2007) 395–416.
[14] A. Sandryhaila J. M. F. Moura, Discrete signal processing on graphs, IEEE Trans. Signal Process. 61 (7) (2013) 1644–1656.
[15] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs:Extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Process. Mag. 30 (3) (2013) 83–98.
[16] M. Onuki, S. Ono, M. Yamagishi, Y. Tanaka, Graph signal denoising via trilateral filter on graph spectral domain, IEEE Trans. Signal Inf. Process. Netw. 2 (2) (2016) 137–148.
[17] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment:from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.