Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 158273, 9 pages
doi:10.1155/2008/158273

Research Article

Color Image Coding by Colorization Approach

Takahiko Horiuchi and Shoji Tominaga

Division of Information Sciences, Graduate School of Advanced Integration Science, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan

Correspondence should be addressed to Takahiko Horiuchi, horiuchi@faculty.chiba-u.jp

Received 24 August 2007; Accepted 23 January 2008

Recommended by Konstantinos Plataniotis

This paper proposes a new color image coding scheme called "colorization image coding." The scheme is based on the colorization technique which can colorize a monochrome image by giving a small number of color pixels. We develop algorithms useful for color image coding. First, the luminance component is separated from an input color image. Then, a small number of color seeds are selected as chrominance information. The luminance image component is coded by a lossy coding technique and the chrominance image component is stored as color seeds. The decoding is performed by the colorization algorithm. It is shown that this colorization technique is effective to image coding, especially for high compression rate, through the experiments using different types of images.

1. INTRODUCTION

With the recent spread of the internet and multimedia technologies, digital color images play more and more important role in human visual communications. Digital color imaging has an enormous impact on industrial applications and scientific projects. Transmission of images and video over very low-bit-rate channels has been a highly challenging task of the information technology during the last decade. Despite many technical problems related to very high compression needed in such applications, international standards have been already accepted. The JPEG image coding standard has enjoyed widespread acceptance, and the industry continues to explore its various implementation issues. Efforts are underway to incorporate recent research findings in image coding into a number of new standards, including those for image coding JPEG 2000 [1], video coding ISO *MPEG-4* [2, 3] and MPEG07 [4], and video teleconferencing ITU-T *H.263+* [5]. Very low-bit-rate image encoders allocate most bits to luminance, and very small number of bits is available for chrominance if the compression ratio is high. Therefore, small number of bits allocated results in numerous artifacts like discontinuities between neighboring blocks or false colors.

In this century, the study of colorization begins to attract attention. Colorization is a technology of coloring monochrome images by giving simple hints of color. Welsh et al. proposed a semiautomatic algorithm by transferring color from a reference color image [6]. Levin et al. proposed an interactive colorization method by giving some color scribbles [7]. The authors also proposed a few colorization algorithms by sowing color pixels and propagating colors spatially to the remaining pixels [8–11]. This paper applies the colorization technique to image data coding, and we call the technique "colorization image coding." Our colorization techniques have an advantage for image coding, since the other colorization techniques, such as reference images and color scribbles, have difficulty in extracting color information. So, the image coding algorithm based on our colorization technique improves color reproduction for high-compression image. The proposed image coding technique meets the human visual system well in which the human eye is far more sensitive to luminance than to chrominance.

Figure 1 shows the procedure of colorization coding scheme. First, luminance component is separated from an input color image. Then, color pixels are selected as seeds from the input color image and colorize the luminance components. The color seed selection is continued until the

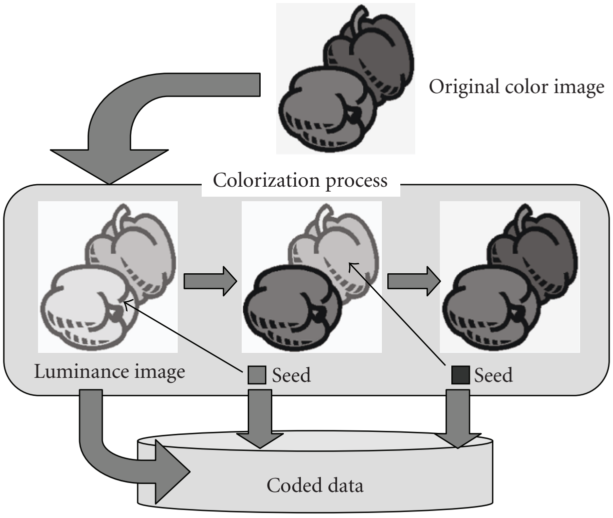
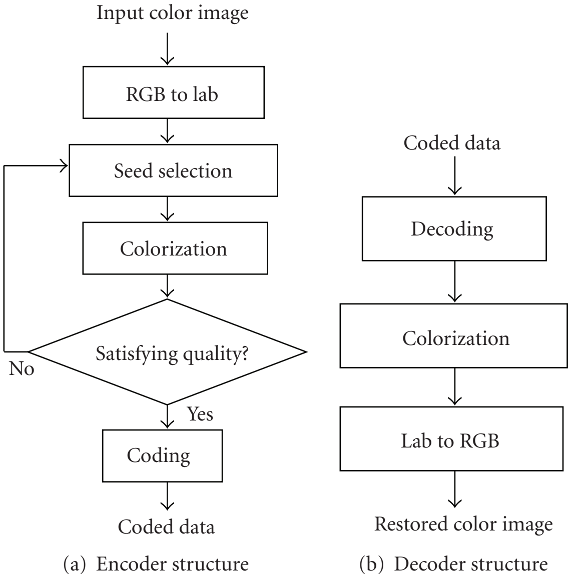


Figure 1: Procedure of colorization image coding.



(a) Encoder structure (b) Decoder structure

Figure 2: Flowchart of the proposed algorithm.

colorized image satisfies a desired quality. Finally, both the luminance component and information of color seeds are stored as coded data. In our coding, most bits are allocated to luminance coding, and only a small number of bits are used for chrominance coding. Thus, the proposed method is suitable for a high compression rate coding.

We compare our results to the standard JPEG compression and show the effectiveness of the proposed coding algorithm.

2. COLORIZATION IMAGE CODING

2.1. Overview

Figure 2 shows the flow of the proposed colorization coding and decoding algorithm. In the coding algorithm, as shown in Figure 2(a), the RGB color first converts to a luminance-chrominance color space. We have investigated many luminance-chrominance spaces such as YUV, YCbCr, and l$\alpha\beta$ [12] under the same compression rate. Then, we found that the use of the CIELAB color space is the best in the quality of colorized images. Therefore, the CIELAB color space is used for expressing color image.

Second, color seed pixels are selected automatically. The selection method is described in Section 2.2. By blending the chrominance components of selected color seeds, all pixels with only luminance component can be colorized by the proposed colorization technique in Section 2.3. The color seed selection and colorization processes are continued until the colorized image satisfies a desired quality by changing the number of color seeds. Finally, the luminance component is coded by a standard lossy coding technique and information of color seeds, which is position and chrominance values, is stored as coded data. Section 2.4 shows this coding scheme.

Figure 2(b) shows the decoding scheme. The decoding process is performed by backward tracing the coding algorithm. The luminance image component and the respective color seeds are decoded and coloring all pixels by the same colorization technique as the coding process. Then, the reproduced color image is in the CIELAB color space converted to the corresponding RGB output image.

2.2. Color seed selection

An input RGB color image is transformed into the CIELAB color space. In order to colorize the luminance image, color pixels from the original color image must be seeded on the luminance image domain. Although, in the general colorization scheme, a user selects the color seeds, in the image coding scheme, the appropriate color seeds are selected automatically.

We have examined the following typical algorithms for selecting color seeds: (a) random selection, (b) selection from high luminance histogram, and (c) selection from box center at high pixel density in the CIELAB space. It is obvious in (a) that a random setting of seeds results in the worst and huge seeds are required to colorize all pixels, because it is independent of image color distribution and seeds are sown on isolated regions. It is required to propagate many pixels by each color seed for reducing data size.

The case (b) constructs luminance histogram for all pixels, in which seeds are selected from the set of pixels with the highest frequency. This method assumes that pixels in the same region have almost the same luminance. According to the assumption, a pixel with a high frequency on the luminance histogram can belong to a large region on the image.

In the case (c), we generate a set of rectangular boxes in the CIELAB color space so that the pixel distribution is covered with the regular lattice points inside the min-max color ranges of the distribution as shown in Figure 3. Each axis of the gamut is divided to m equal intervals depending on the gamut size of the input image. Figures 3(a) and 3(b) show the pixel distribution of the original color image in the CIELAB color space and the generated rectangular boxes covering

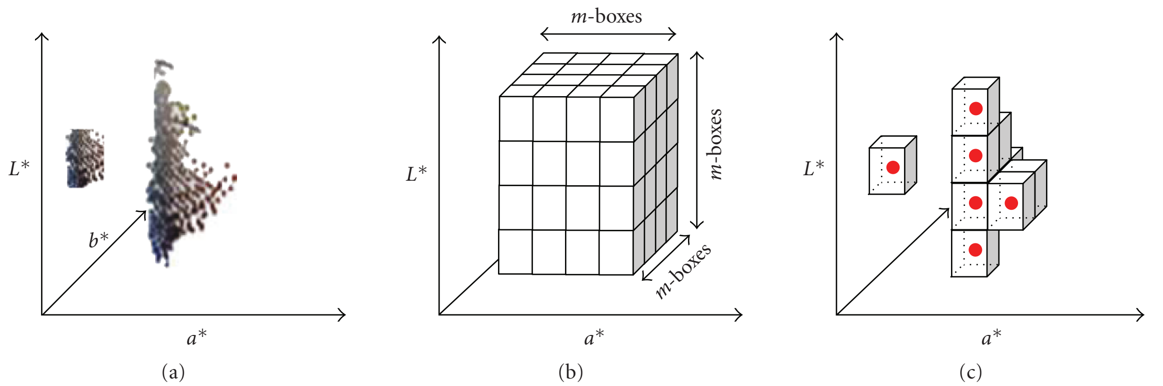


FIGURE 3: Seed selection algorithm from box center at high pixel density in the CIELAB space. (a) Color pixel distribution of original image. (b) Rectangular boxes ($m = 4$). (c) Selected boxes and seeds (indicated as red circles).

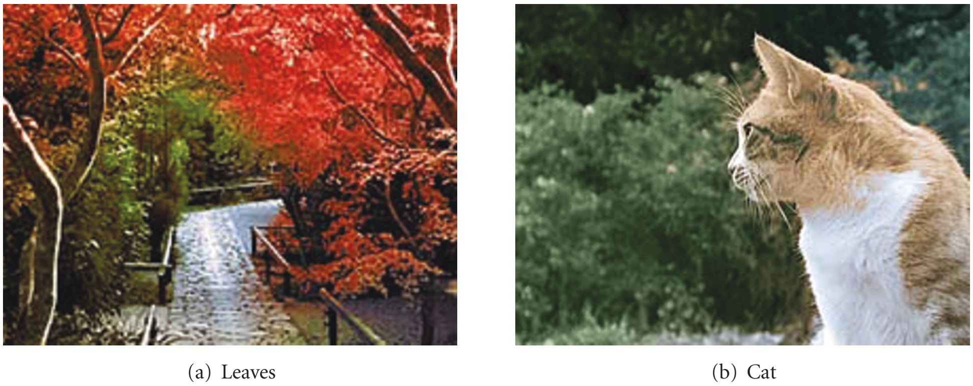


FIGURE 4: Examples of natural images.

the image gamut, respectively. Figure 3(c) shows the selected boxes with high pixel density and color seeds indicated by red circles. The location of each seed is not placed at the center of each cluster but placed at each body center in equally divided unit boxes. This method is much better than the other seed selection methods. We adopt the K-means clustering algorithm to place these seeds at the suitable positions in the CIELAB space.

The K-means algorithm partitions N data points into K disjoint subsets $D_k (k = 1, 2, \dots, K)$ containing N_k data points, so $(N = N_1 + N_2 + \cdots + N_K)$ as to minimize the sum-of-squares criterion,

$$J = \sum_{k=1}^{K} \sum_{n \subseteq D_k} \left| \mathbf{X}_n - \boldsymbol{\mu}_k \right|^2, \qquad (1)$$

where $\mathbf{X}_n$ is a pixel belonging to a class D_k and $\boldsymbol{\mu}_k$ is the geometric centroid of the pixels in D_k. First, the initial seed points $\boldsymbol{\mu}_{\text{seed}}(k)$ are assigned to the respective k classes. Then, coordinates of the centroid are recomputed after clustering and the seed points are renewed. The renewal is continued until no further change occurs in the centroid by iteration. Although the K-means is used as an unsupervised classifier by setting the initial seeds in random, the present algorithm uses this technique for relocating the initial seeds to the more reliable center of gravity in clusters. Finally, pixels near to the converged centroids of K-classes are determined as color seeds. If there are two or more pixels with the same color as the centroid, only one pixel is selected at random as a color seed. Since the restored image quality depends on the parameter K, in our coding scheme, the parameter K is determined by a user as a quality parameter. However, it is possible to determine K automatically by the process of increasing the number of K until the colorized image quality $\text{PSNR}_{\text{org}}(k)$ satisfies a certain desired quality PSNR_{des}.

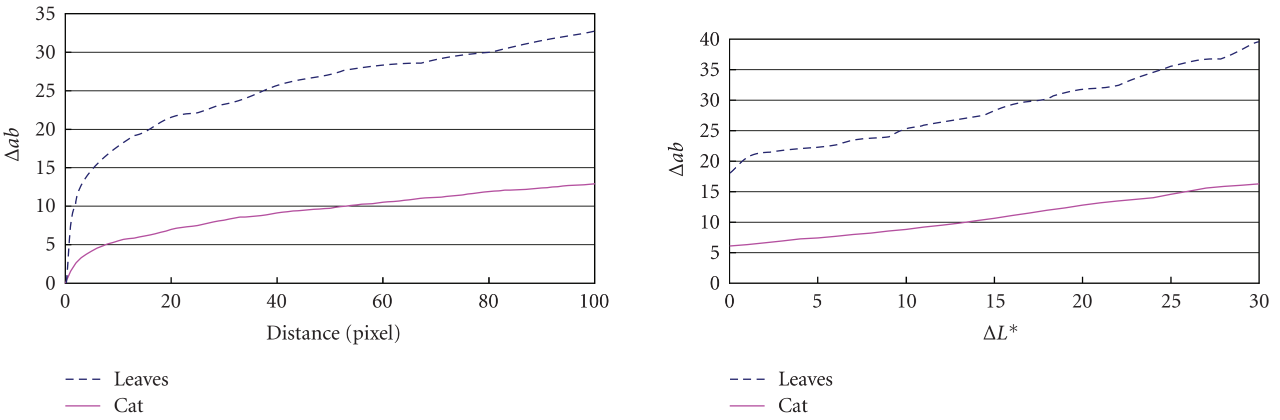
2.3. Colorization

Colorization algorithm is essential in the colorization coding. We propose a colorization algorithm utilizing two properties of natural images.

The first property claims that if the spatial distance between two arbitrary pixels in an image is close in the Euclidean distance, the chrominance distance Δab is small. The chrominance distance Δab between colors (L_1, a_1, b_1) and (L_2, a_2, b_2) is defined in CIELAB space as

$$\Delta ab = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}. \qquad (2)$$

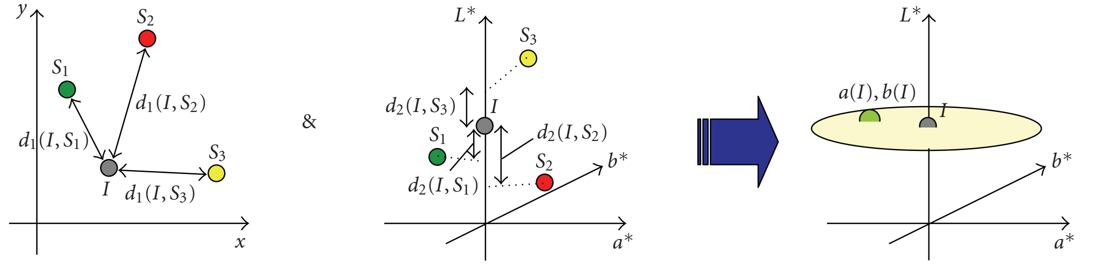
We confirm the property using natural images. Figure 4 shows two examples of typical natural images. Figure 5(a) shows the relation between the Euclidean distance and the



(a) The relation between spatial distance and chrominance distance

(b) The relation between luminance distance and chrominance distance

Figure 5: Properties of natural images.



(a) NSD on x-y plane

(b) NLD in CIELAB space

(c) Estimated chroma on a^*-b^* plane

Figure 6: Geometric expression of the proposed colorization process.

average of the chrominance distance Δab for all pair of pixels in Figure 4. Horiuchi's colorization algorithm in [8] was constructed based on this property. It should be noted in both images that the chrominance distance increases as the spatial distance between pixels increases.

The second property is that pixels with similar luminance values have similar color. This property was used for solving colorization problem in Levin's algorithm [7]. Figure 5(b) shows the relation between the luminance distance ΔL^* and the average of the chrominance distance Δab for all pair of pixels in the images in Figure 4. The luminance distance ΔL^* between colors (L_1, a_1, b_1) and (L_2, a_2, b_2) is defined in CIELAB space as

$$\Delta L^* = \sqrt{(L_1 - L_2)^2}. \qquad (3)$$

Note that Δab increases with ΔL^*.

Let $I = (x, y)$ be a pixel in an input monochrome image, and let $\{S_p \mid S_p = (x_p, y_p)\}_{p=1}^{K}$ be a set of color seeds, where K is the total number of the selected seeds, and S_p is a coordinate of a color seed selected in Section 2.2. Each monochrome pixel I is transformed into the luminance component $L(I)$. Each color seed S_p is represented by the coordinates $L(S_p), a(S_p), b(S_p)$ in the CIELAB color space. Then, we define two distances called NSD and NLD for expressing the above two properties as follows.

Normalized spatial distance (NSD)

The first distance $d_1(I, S_p) \in [0, 1]$ between I and S_p is defined in the spatial domain as

$$d_1(I, S_p) = \frac{\|I - S_p\|^2}{\sqrt{\text{width}^2 + \text{height}^2}}, \qquad (4)$$

where width and height mean the horizontal size and vertical size of the monochrome image, respectively. Symbol $\|\cdot, \cdot\|^2$ means the Euclidean distance.

Normalized luminance distance (NLD)

The second distance $d_2(I, S_p) \in [0, 1]$ between I and S_p is defined in the luminance domain as

$$d_2(I, S_p) = \frac{|L(I) - L(S_p)|}{100}. \qquad (5)$$

Then, we define a combined distance $d(I, S_p) \in [0, 1]$ between I and S_p by the weighted sum of the both distances NSD and NLD as

$$d(I, S_p) = \alpha d_1(I, S_p)^r + (1 - \alpha) d_2(I, S_p)^r, \qquad (6)$$

where $\alpha \in [0, 1]$ indicates a constant for weighting distances. If $\alpha = 1$ in (6), only the NSD, which is a spatial factor, is used for colorization process and it is equivalent to the algorithm in [9]. The algorithm colorizes neighboring pixels with the same chrominance components, and it is impossible to express texture regions. On the other hand, if $\alpha = 0$ in (6), only the NLD, which is a luminance factor, is used for colorization process. The algorithm can colorize neighboring pixels with remarkably different colors, because pixels with the same luminance components in an image will be colorized with the same colors. Therefore, an appropriate blending parameter α is required. The optimum parameter value depends basically on the image contents. However, we found empirically that $\alpha = 0.5$ is useful for any images. Symbol $r \in \mathfrak{R}^+$ indicates a factor which can change the influence of the color seeds. According to the investigation of property for natural images, the influence decreases rapidly as the distance increases, and the influence is nonlinear for the distance. This parameter was determined as $r = 5$ empirically in our experiments.

We consider the chrominance components $a(I)$ and $b(I)$ of the target pixel I by blending the chrominance of seeds. As blending the seeds chrominance, we use the combined distance. If the combined distance $d(I, S_p)$ in (6) is small, the target pixel I should be colorized with similar chrominance components of the seed S_p. So, we transform the combined distance $d(I, S_p)$ into a weight $W(I, S_p) \in [0, 1]$ for blending chrominance of seeded color as

$$W(I, S_p) = \frac{d(I, S_p)^{-1}}{\sum_p d(I, S_p)^{-1}}. \tag{7}$$

Equation (7) shows that the weight for blending becomes large when the combined distance in (6) is small. Then, the color for the pixel I can be represented by $L(I)$, $a(I)$, and $b(I)$ by weighting chrominance of all seeds as follows:

$$\begin{aligned} L(I) &= L(I), \\ a(I) &= \sum_p W(I, S_p) a(S_p), \\ b(I) &= \sum_p W(I, S_p) b(S_p), \end{aligned} \tag{8}$$

where the luminance component $L(I)$ maintains the luminance value of the original image.

Figure 6 illustrates the above algorithm. We use two distances NSD and NLD in the (x, y) image plane and the CIELAB color space. The chrominance components $a(I)$ and $b(I)$ are determined based on the weighted distances on the luminance plane. Thus, the color value is determined based on the two properties with keeping the luminance value of the original image.

The proposed method determines the chrominance components $a(I)$ and $b(I)$ by calculating the weighted average of them for color seeds in (8). If the directions of seeds' vectors $(a(S_{p1}), b(S_{p1}))$ and $(a(S_{p2}), b(S_{p2}))$ are greatly different in hue, the chroma $(a(I), b(I))$ may become different color from the seeds because $(a(I), b(I))$ is the weighted average vector. A solution to this irregular color problem is only to set appropriately seed pixels. The proposed algorithm can increase color seeds automatically until satisfying a desired quality $\mathrm{PSNR}_{\mathrm{des}}$ by using the K-means algorithm. Since the

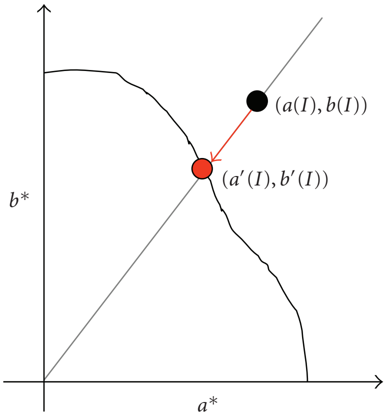
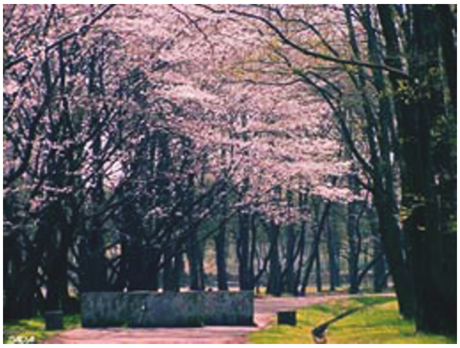


Figure 7: Gamut mapping on a^*-b^* plane by clipping.

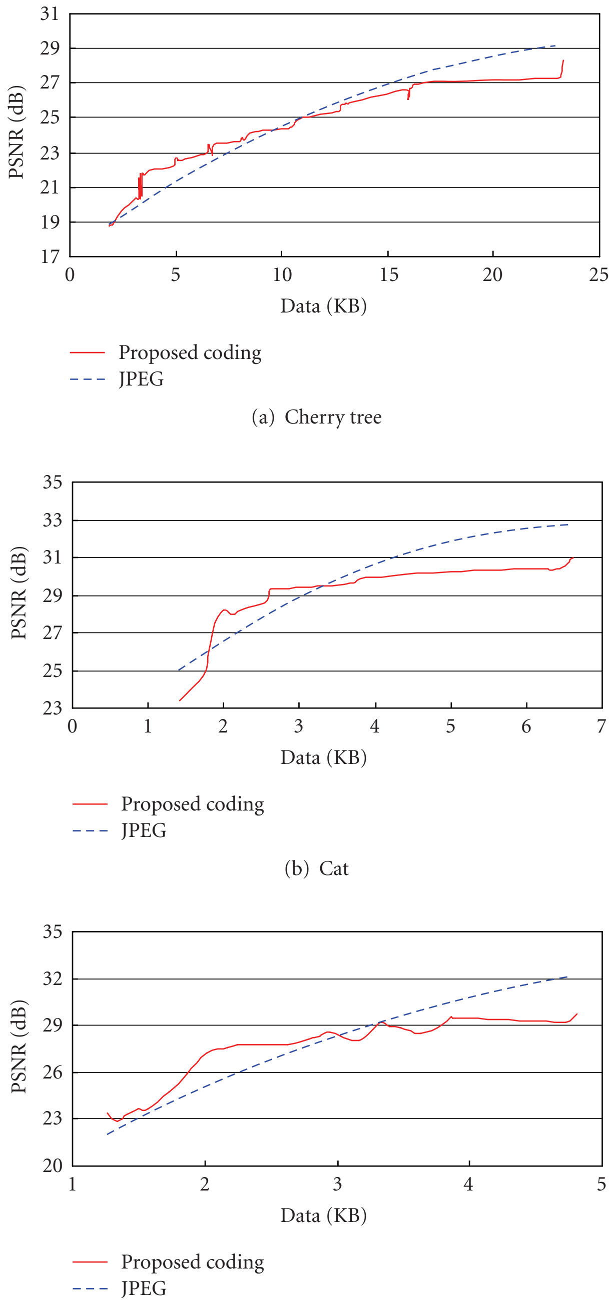


(a) Cherry tree

(b) Cat

(c) Buckingham

Figure 8: Test images.



(a) Cherry tree

(b) Cat

(c) Buckingham

Figure 9: Data size versus PSNR.

insufficient accuracy in $\mathrm{PSNR}_{\mathrm{org}}(k)$ means that a lot of false colors are generated, it can be solved by increasing the seed pixels until false color is improved. Concretely, we have many pixels with different chroma from the already seeded pixels in an image region where there are remarkably false colors. Then, we seed the region with the same chroma. In practice, as shown in Section 2.2, because an initial seed in the K-means algorithm is added to a rectangular box with high pixel density in the CIELAB space, there is high possibility of assigning a seed pixel to the false color region.

Then, the coordinate data of $(L(I),\ a(I),$ and $b(I))$ are transformed into the standard RGB (sRGB) coordinate system for printing or displaying. The sRGB standard, originally suggested by Ralf Kuron, is backward compatible and is based on a calibrated colorimetric RGB color space well suited to CRT monitors, television, scanners, digital cameras, and printing systems. We use the sRGB color space to reproduce a correct color with various devices. The transformed color coordinates determined color may exist outside of the sRGB gamut because $L(I)$, $a(I)$, and $b(I)$ are determined independently. Since we observe the colorized image on a printing device or display device, the color coordinates must exist within the sRGB gamut. For solving such a problem, many gamut mapping algorithms have been proposed in the field of color science [13]. They are mostly designed to work in 2D L-C (Lightness-Chroma) planes. However, a big assumption in the general colorization scheme is to keep the luminance value L^* of the original monochrome image. So, we consider a gamut mapping on a^*-b^* plane in this paper. Figure 7 illustrates the gamut mapping used in our algorithm. We use a typical clipping algorithm in which a point out of gamut moves to the direction of origin with keeping the hue angle. The points within the gamut do not move to anywhere. In Figure 7, the estimated chrominance coordinates $(a(I),\ b(I))$ are mapped to the boundary of the sRGB gamut $(a'(I), b'(I))$. The gamut clipping has an advantage of keeping the entire color reproduction accuracy because the clipping algorithm does not change the color within the gamut [13]. This property is important in the color image coding, although color discontinuity can be caused around clipping part as an artifact.

2.4. *Data coding*

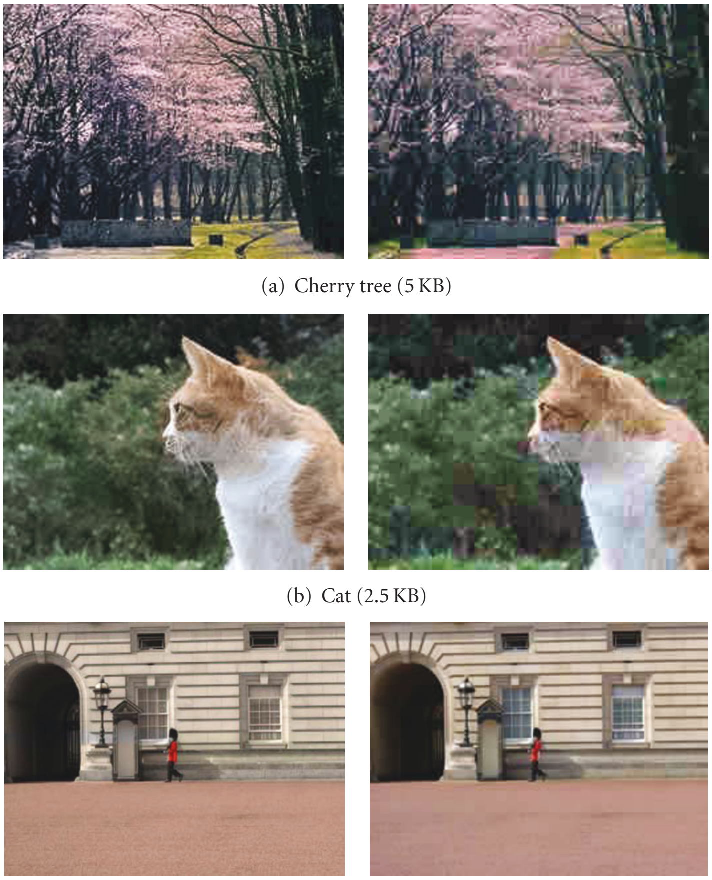
The above algorithms convert an input color image into luminance components and a small number of seed pixels. According to the human visual perception, the luminance component is coded more precisely than the chrominance. In our algorithm, the luminance L is coded by an orthogonal transform such as DCT and wavelet with high quality. The data of each seed pixel are coded as a four-dimensional vector which consists of the spatial coordinate $I = (x, y)$ and the chrominance coordinate $(a(I),\ b(I))$. If 2 bytes are assigned for each element of the vector, it takes 8 bytse for each color seed. So, the total data become

$$\text{Total data (byte)} = \text{coded}\,(L) + 8\,K, \tag{9}$$

where K means the number of color seeds.

3. EXPERIMENTAL RESULTS

In order to verify the performance of the proposed method, we have compared the present coding method with JPEG standard compression using natural images. Figure 8 shows test images. These are 24-bit color images and their size is 256 × 192. Generally speaking, it is not easy to precisely represent color images including high-frequency components with a small number of seeds [8–11]. Figure 8(a) is an


(a) Cherry tree (5 KB)

(b) Cat (2.5 KB)

(c) Buckingham (7 KB)

Figure 10: Restored images under the same data size. Left: results by the proposed method. Right: results by the JPEG coding.

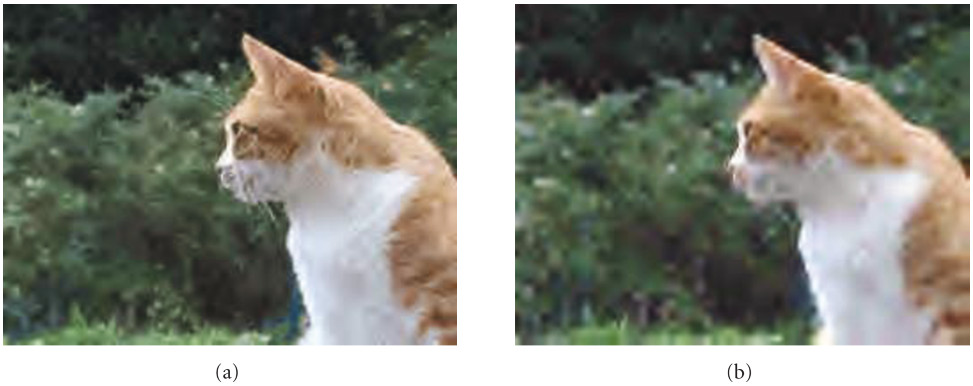


Figure 11: Restored images under the same data size 1.5 Kbyte. (a) results by the proposed method. (b) results by the JPEG 2000 coding.

example of image that includes large areas of high-frequency textures. Figure 8(c) is an example with large areas of uniform colors. Figure 8(b) is an intermediate example. Our experiments used the JPEG coding for the luminance coding. The data size of the original image is 144 KB. We investigated the relation between data size and image quality by changing parameters. Note that the image quality changes greatly on the basis of the number of seeds.

The following peak signal-to-noise ratio (PSNR) [dB] was used as a measure of quality verification:

$$\begin{aligned} \text{PSNR} &= 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right), \\ \text{MSE} &= \frac{1}{3N}\sum_{\text{color}}\sum_{i=1}^{N}\left(x_i^{\text{color}} - y_i^{\text{color}}\right)^2, \end{aligned} \tag{10}$$

where pixels x_i^{color} and y_i^{color} in an original image and a restored image have integer value from 0 to 255 at pixel i in a channel color = {red, blue, and green}. N means the total number of pixels.

Figure 9 shows the relation between the PSNR value and the data size for the test images. The red and blue curves represent the results by the proposed coding and the JPEG coding, respectively. Comparison between red and blue curves suggests that the relative relationship between PSNR and data size looks similar for three images. The performance of the proposed method wins in the part of the high compression rate, while the performance of JPEG is higher than that of the proposed method in the part of low compression rate. However, in Figure 9(b), the performance of JPEG is higher again in the part which is lower than 2 KB. This result means that the coding performance of the proposed method decreases when the number of color seeds is insufficient, because the proposed method cannot perform an appropriate coloring by extremely small number of color seeds.

Images in Figures 8(a) and 8(b) include a lot of high-frequency components, and so many seeds are required for expressing texture parts. Nevertheless, a better performance than JPEG coding is obtained in such a region, as shown in Figures 9(a) and 9(b). Thus, the proposed coding gets better results in high-compression coding under the PSNR measure. For each image, almost 99% of the coded data occupy the luminance component, and around 1% data is assigned for chrominance components which correspond to seed pixels.

We show the restored images by the proposed coding method and JPEG coding method in Figure 10. For each pair of images, both images have the same data size which is high compression rate from 2% to 5% of the original data. Our experiments showed that terrible block noise appeared in the JPEG result, but it tends to have fewer blocky artifacts by the proposed method. However, the washed out color was recognized by using the proposed algorithm. In Figure 10(c), the PSNR measure of the JPEG coding is better than that of the proposed coding. However, the texture of the road is destroyed by block noises for JPEG coding, and a fall of quality is recognized visually. In contrast, as for the proposed coding, the color changes at the window by the lack of seeds, but the details of the original image are clearly restored.

Moreover, we compared the proposed algorithm with the JPEG2000. In this experiment, we used the JPEG2000 for the monochrome encoding in our algorithm. The relationship between the proposed method and the JPEG2000 was almost the same as in Figure 9, though the total performance improved by using the JPEG2000. Figure 11 shows images by the proposed coding method and JPEG2000 coding method under the same data size 1.5 KB. The detailed inspection suggests that the image quality by the proposed method is better than JPEG2000. For instance, the restored image by JPEG2000 with high compression rate loses the detailed textures of cat's hairs and trees in Figure 11, while the image by the proposed algorithm expresses the details more clearly.

4. CONCLUSIONS

This paper proposed a new color image coding algorithm based on the colorization technique. In our algorithm, at first, luminance component was separated from an input color image. Then, color pixels were selected as seeds from the input color image and colorize the luminance components. The color seed selection was continued until the colorized image satisfies a desired quality. Finally, both the luminance component and information of color seeds were stored as coded data. Thus, the proposed coding algorithm is essentially different from techniques based on signal transforms such as PCA and DCT.

The proposed algorithm has the following characteristics.

(1) Most bits are allocated to luminance coding, and only a small number of bits are used for chrominance coding.

(2) The proposed method is effective for a high compression rate coding.

(3) Fewer blocky artifacts are recognized.

Since most of the coded data is luminance component, the data quantity of the luminance component is important for total performance. In our experiment, we used JPEG for luminance coding. We have to try to use other orthogonal transforms and investigate more suitable coding method for luminance component.

REFERENCES

[1] ISO/IEC FDIS 15444-1, “Core Coding System,” 2000.

[2] ISO/IEC DIS 14495, “Generic coding of audio-visual objects”.

[3] F. Pereira and T. Alpert, “MPEG-4 video subjective test procedures and results,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 32–51, 1997.

[4] ISO/IEC JTC1/SC29/WG11, “MPEG07/ M14493,” 2007.

[5] ITU-T Rec.H.263, “Video coding for narrow telecommunication channels at < 64kbit/s”.

[6] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to grayscale image,” in *Proceedings of the 29th ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’02)*, vol. 20, pp. 277–280, San Antonio, Tex, USA, July 2002.

[7] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 689–694, 2004.

[8] T. Horiuchi, “Estimation of color for gray-level image by probabilistic relaxation,” in *Proceedings of the 16th International Conference on Pattern Recognition (ICPR ’02)*, vol. 3, pp. 867–870, Quebec, Canada, August 2002.

[9] T. Horiuchi and S. Hirano, “Colorization algorithm for grayscale image by propagating seed pixels,” in *Proceedings of the International Conference on Image Processing (ICIP ’03)*, vol. 1, pp. 457–460, Barcelona, Spain, September 2003.

[10] T. Horiuchi, “Colorization algorithm using probabilistic relaxation,” *Image and Vision Computing*, vol. 22, no. 3, pp. 197–202, 2004.

[11] T. Horiuchi and H. Kotera, “Colorization for monochrome image based on diffuse-only reflection model,” in *Proceedings of the 10th Congress of the International Colour Association (AIC ’05)*, pp. 353–356, Granada, Spain, May 2005.

[12] D. L. Ruderman, T. W. Cronin, and C.-C. Chiao, "Statistics of cone responses to natural images: implications for visual coding," *Journal of the Optical Society of America A*, vol. 15, no. 8, pp. 2036–2045, 1998.

[13] J. Morovic, "To develop a universal gamut mapping algorithm," Ph.D. Dissertation, University of Derby, Derby, UK, 1998.