OPTICAL REVIEW Vol. 16, No. 6 (2009) 601–612

Derivation of the Analytical Solution of Color2Gray Algorithm and Its Application to Fast Color Removal Based on Color Quantization

Go TANAKA, Noriaki SUETAKE*, and Eiji UCHINO

Graduate School of Science and Engineering, Yamaguchi University, Yamaguchi 753-8512, Japan

(Received June 11, 2009; Accepted September 25, 2009)

In order to convert a color image into a monochrome one, the lightness components of pixels have to date been used as gray-levels for the representation of the monochrome image. However, saliencies of an image embedded only in the chrominance components are eliminated in such conversion. To cope with this problem, "Color2Gray" algorithm, which excels in the color removal of digital images, has been proposed by Gooch *et al.* [ACM Trans. Graphics **24** (2005) 634]. In this paper, the algorithm is first analyzed and its mathematical property is revealed. Then a fast Color2Gray algorithm is proposed by using the mathematical property. Finally, the validity and the effectiveness of the proposed algorithm are proven by some experiments.

Keywords: color-to-monochrome conversion, color removal, Color2Gray algorithm, color image, monochrome image, optimization problem, conjugate gradient method, computational cost reduction

1. Introduction

In recent years, digital color images have become very familiar to us, and color printers have also become a common item. However, to save printing costs, the monochrome printing of color images is still popular. Thus the color-to-monochrome image conversion is an important color transformation technique.

In a usual and typical color-to-monochrome conversion, only the lightness components of an input color image are extracted and used to represent the monochrome one. For example, in the high- and standard-definition television (HDTV and SDTV) standard,[1,2] the lightness component of i-th pixel Y_i^{HDTV} and Y_i^{SDTV} are obtained as follows:

$$Y_i^{\mathrm{HDTV}} = 0.2126r_i + 0.7152g_i + 0.0722b_i, \quad (1)$$

$$Y_i^{\mathrm{SDTV}} = 0.299r_i + 0.587g_i + 0.114b_i, \quad (2)$$

where (r_i, g_i, b_i) stand for the linear RGB components of i-th pixel.

The color-to-monochrome conversions represented by the weighted sum of RGB components, such as eqs. (1) and (2), do not require complex calculation, and in many cases good monochrome images can be obtained. However, the conversion using only lightness components does not always reflect the color information in a monochrome image appropriately. For example, the image of "Impression" drawn by Monet and its monochrome images obtained by eqs. (1) and (2) are shown in Figs. 1(a), 1(b), and 1(c), respectively. From Figs. 1(b) and 1(c), it is observed that the sun and its reflection on the water almost disappear.

To cope with this problem, color removal methods which consider differences of colors in an input image have been proposed in recent years.[3–19] For example, Fig. 2 shows the color removal results of Impression obtained by several methods.[3,4,8] The sun and its reflection on the water can be recognized though the appearances of the images differ. Though a definitive method has not yet been proposed[20] because each one has some shortcomings, we consider that the algorithm "Color2Gray" proposed by Gooch *et al.*[3] is relatively good in the image quality of the resulting image. In Color2Gray, differences of colors in an input image are first quantitatively expressed and an optimization problem concerning the color removal is constructed using them. Then the monochrome image, which reflects differences of colors, is obtained by solving the problem using a conjugate gradient (CG) method.[21] Although the Color2Gray can yield visually good monochrome images in many cases, it is well known that its computational cost is tremendous for practical use. The cost of the algorithm is $O(n^2)$ when the number of pixels in an input image is expressed as n.

In this paper, the analytical solution of the optimization problem of Color2Gray is first revealed. However, the computational cost of calculating the bare analytical solution is $O(n^2)$ and is equivalent to the cost of an original Color2Gray employing the CG method. A color quantization is introduced to reduce the cost of calculating the solution, and we then propose a method to compensate the deterioration of image quality of the resulting image caused by the quantization. Finally, a fast and efficient color-to-monochrome conversion algorithm is proposed. The computational cost of this algorithm is $O(n \log n + m'^2)$, where m' is the number of colors in an input image after a quantization.

The rest of this paper is organized as follows. An overview of the color removal methods proposed so far is given in §2. Details of the Color2Gray algorithm are introduced in §3. Then, the analytical solution of the optimization problem formulated in Color2Gray is derived, and a new color-to-monochrome conversion method, in which computational cost is drastically reduced compared with the original Color2Gray, is proposed by transforming the analytical solution in §4 and §5, respectively. Finally, the effectiveness of the proposed method is verified through

*E-mail address: suetake@sci.yamaguchi-u.ac.jp

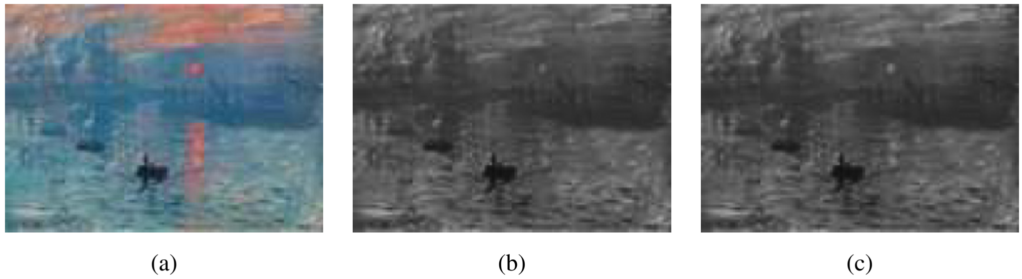


Fig. 1. (Color online) "Impression" drawn by Monet and its monochrome image: (a) original color image, (b) lightness component in HDTV standard, (c) lightness component in SDTV standard.

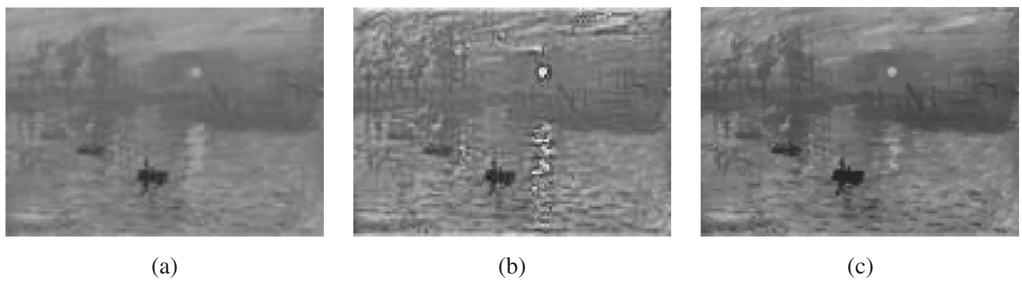


Fig. 2. Color removal results of Impression obtained by several methods: (a) Gooch *et al.* method,[3] (b) Bala and Eschbach method,[4] (c) Grundland and Dodgson method.[8]

some conversion experiments in §6, and the conclusions are described in §7.

2. Related Work

In this section, a brief overview of the color removal methods is given.

Bala and Eschbach[4] proposed a technique which was based on unsharp masking[22] (UM). UM enhances the high frequency components (edges) in an input image, and their method can be regarded as a kind of selective UM in which the strength of edge enhancement effect is controlled based on the value of the chrominance components of each pixel. Alsam and Kolas,[5] and Smith *et al.*[6] also proposed techniques which obtained the enhanced monochrome images by selective UM based on the chrominance components.

However, the methods which employ the UM technique have a common shortcoming: the edge enhancement effect cannot be realized when the two iso-lightness regions are separately placed. Moreover, the enhanced edges are sometimes observed as artifacts. In other words, the "global consistency", which refers to the identical gray-level being assigned to the same colors in an input image, is not satisfied in UM-based methods.

Ways that change the projection coefficients of eq. (1) or (2) for each input image appropriately have also been proposed.[7–13] A simple means is to first use a principal component analysis[23] (PCA) to obtain the coefficients of color-to-monochrome projection. However, the image quality of the resulting image obtained by a simple PCA is insufficient in many cases.[9,14] Zhang *et al.*[7] proposed a method based on the kernel PCA, and Grundland and Dodgson[8] proposed a "predominant component analysis", which resembles PCA, and the color removal method "Decolorize" based on it. On the other hand, Rasche *et al.*[9] proposed a method to obtain the projection coefficients by solving an optimization problem considering the color differences in an input image. However, the computational cost of their method is huge and is not suitable for practical use. Though they introduced color quantization and interpolation to reduce the cost,[14] it is still tremendous. Note that the method proposed in ref. 14 does not use a linear projection, but the algorithm is very similar to the algorithm proposed in ref. 9.

An advantage of the projection-based methods is that the global consistency is automatically satisfied. However, the ability of the color-to-monochrome conversion of the projection-based methods is not high, and it sometimes happens that an identical gray-level is assigned to different colors. Moreover, the "average lightness consistency", which means preservation of the average lightness of an input image in the color-to-monochrome conversion, is not realized because the methods alter the coefficients used in eq. (1) or (2). When the average lightness of an output image is different from that of the input image, the color removal cannot be considered appropriate because the impression given by the image is changed.

Although there are other methods than those mentioned above,[15–19] they also have respective shortcomings.

Consequently, the method proposed by Rasche *et al.* in ref. 14 most closely resembles the Color2Gray algorithm. Both methods obtain the output gray-levels by solving the optimization problem. The most significant difference between them in reference to this is whether or not their objective functions contain absolute values. The function employed by Rasche *et al.* contains the absolute value, and the optimization problem must therefore be solved by an iterative method. On the other hand, the objective function formulated in Color2Gray does not contain the absolute value, which allows the problem to be solved analytically under a certain condition, which is a natural and default condition in Color2Gray.[3)] The analytical solution described in this paper reveals that the global and average lightness consistency are actually realized in the color removal of Color2Gray under this condition.

3. Color2Gray: Color Removal Method Considering the Difference of Colors

Gooch *et al.*[3)] proposed a color-to-monochrome image conversion method also called Color2Gray which obtains a converted monochrome image by minimizing an objective function formulated based on the "signed color distance". In Color2Gray, the signed color distance, which expresses the difference of colors in an input image, is reflected onto the gray-levels constructing the monochrome image through a minimization of the objective function. Details of Color2Gray are described below.

3.1 *Formulation of the optimization problem in Color2Gray*

In Color2Gray, the following objective function is formulated:

$$E(\boldsymbol{f}) = \sum_{(i,j)\in\sigma_\rho} [(f_i - f_j) - \delta_{ij}]^2, \quad (3)$$

where f_i is a gray-level of i-th pixel, and $\boldsymbol{f}$ stands for gray-levels of a whole image. That is, $\boldsymbol{f}$ is $(f_1, f_2, \ldots, f_n)^{\mathrm{T}}$. T and n stand for a transposition and the number of pixels in an input image, respectively. δ_{ij} is a signed color distance between i-th and j-th pixels, and means the difference of the colors between those pixels. σ_ρ is a set of pixel pairs, and its elements are the pixel pair (i,j)s satisfying that a chessboard distance between the pixels is less than or equal to ρ. That is, the pixel pairs (i,j) satisfying $\max(|x_i - x_j|, |y_i - y_j|) \le \rho$ are the elements of σ_ρ when a spatial coordinate of i-th pixel is represented by (x_i, y_i). ρ indicates a range of the neighborhood. In the case where all pixel pairs in an input image are elements of the set, ρ is expressed as ∞ in this paper.

Then a converted monochrome image $\tilde{\boldsymbol{f}}$ is obtained by solving the following optimization problem:

$$\tilde{\boldsymbol{f}} = \arg\min_{f_i\in\mathbb{R}} E(\boldsymbol{f}). \quad (4)$$

Equation (4) is solved using the CG method in which lightness components $\boldsymbol{l}$ of an input color image are used as an initial value of $\boldsymbol{f}$. However, $E(\boldsymbol{f}) = E(\boldsymbol{f} + \boldsymbol{c})$ is satisfied where $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)^{\mathrm{T}}$ and the values of all c_is are identical, so that eq. (4) has an infinite number of solutions. Hence, after obtaining a solution $\tilde{\boldsymbol{f}}$ by the CG method, the gray-level shifting of the solution is achieved to get a valid average lightness of an output converted image. Concretely, a final converted image is obtained by the gray-level shifting which attracts the average lightness of an output monochrome image to that of the input color one.

3.2 *Definition of the signed color distance*

The signed color distance δ_{ij} between i-th and j-th pixels used in eq. (3) is defined by

$$\delta_{ij} = \begin{cases} \Delta L^*_{ij} & |\Delta L^*_{ij}| > \Phi_\alpha(\|\Delta \boldsymbol{C}_{ij}\|) \\ \mathrm{sign}(\Delta \boldsymbol{C}_{ij} \cdot \boldsymbol{v}_\theta)\Phi_\alpha(\|\Delta \boldsymbol{C}_{ij}\|) & \text{otherwise} \end{cases} \quad (5)$$

with

$$\Phi_\alpha(x) = \alpha \tanh(x/\alpha), \quad (6)$$

$$\boldsymbol{v}_\theta = (\cos\theta, \sin\theta), \quad (7)$$

$$\mathrm{sign}(x) = \begin{cases} +1 & x > 0 \\ -1 & \text{otherwise} \end{cases}, \quad (8)$$

where ΔL^*_{ij} is $L^*_i - L^*_j$ and $\Delta \boldsymbol{C}_{ij}$ is $(\Delta a^*_{ij}, \Delta b^*_{ij})$, that is, $(a^*_i - a^*_j, b^*_i - b^*_j)$. L^*_i, a^*_i, and b^*_i are color components of i-th pixel which are transformed onto the CIE 1976 $L^*a^*b^*$ color space.[22)] "$\cdot$" indicates an inner product. As shown in eq. (5), the signed color distance δ_{ij} is given as ΔL^*_{ij} when the absolute lightness difference is more dominant than the chrominance difference. Otherwise the value related to the chrominance difference is assigned to δ_{ij}. α is a parameter which gives importance to the chrominance difference in the color removal. θ is a parameter to determine a sign of the color distance in the conversion. That is, the value of θ determines which colors should be brighter or darker in the color-to-monochrome conversion; for example, when θ is about $\pi/4$, warm-colored and cool-colored pixels become brighter and darker in the conversion, respectively.

4. Analytical Solution of the Optimization Problem in Color2Gray

In Color2Gray, a range of the neighborhood ρ is usually set as ∞[3)] because the distances of pixel pairs, which cannot be discriminated in the ordinal monochrome image, are unknown beforehand. In the case where the number of pixels of an input image is n, the number of elements within σ_∞ becomes ${}_nC_2 + n = n(n+1)/2$, where ${}_nC_2$ means the combination n choose 2, and the condition $\rho = \infty$ seems inappropriate from the viewpoint of computational complexity. However, the optimization problem of Color2Gray can be solved analytically and directly without numerous iterations when ρ is set as ∞, though this is not described in ref. 3.

Here we solve the optimization problem analytically, and give the analytical solution directly in §4.1. Additionally, the characteristics of the color-to-monochrome conversion of Color2Gray are discussed using the solution in §4.2.

4.1 *Derivation of the analytical solution*

The solution of the optimization problem defined by eqs. (3)–(8) is obtained by a first iteration in the CG method when ρ is set as ∞. Though eq. (4) has an infinite number of solutions as mentioned above, in fact, a unique solution is obtained depending on an initial value of the iteration process. Now let's solve the optimization problem.

To solve the optimization problem is equivalent to solving the following simultaneous equations:

$$A_{(n)}\boldsymbol{x} = \boldsymbol{b} \quad (9)$$

with

$$A_{(n)} = nI_{(n)} - J_{(n)}, \tag{10}$$

$$\boldsymbol{x} = (f_1, f_2, \ldots, f_n)^{\mathrm{T}}, \tag{11}$$

$$\boldsymbol{b} = \left(\sum_{j=1}^{n} \delta_{1j}, \sum_{j=1}^{n} \delta_{2j}, \ldots, \sum_{j=1}^{n} \delta_{nj}\right)^{\mathrm{T}}, \tag{12}$$

where $I_{(n)}$ is an $n \times n$ identity matrix, and $J_{(n)}$ is an $n \times n$ matrix of which all the elements are 1. n is the number of pixels of an input image, and is equal to the number of dimensions of the simultaneous equations. Equations (9)–(12) are obtained by rearranging the equations derived from differentiation of eq. (3) for each f_i and regarding them as 0.

By using eqs. (10) and (12), it can be proven that there are the following relationships concerning $A_{(n)}$ and $\boldsymbol{b}$:

$$A_{(n)}^2 = nA_{(n)}, \tag{13}$$

$$A_{(n)}\boldsymbol{b} = n\boldsymbol{b}. \tag{14}$$

The proof of eqs. (13) and (14) is shown in Appendices A and B, respectively. Then the analytical solution can be obtained based on the procedure of the CG method and the relationship of eqs. (13) and (14).

In the procedure of the CG method, an approximate solution vector in k-th iteration $\boldsymbol{x}_k$ is updated as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k, \tag{15}$$

where λ_k is an increment and $\boldsymbol{p}_k$ is a search direction vector. On the other hand, a residual vector $\boldsymbol{r}_k$ is defined as

$$\boldsymbol{r}_k = \boldsymbol{b} - A_{(n)}\boldsymbol{x}_k. \tag{16}$$

If $\boldsymbol{r}_k$ becomes $\boldsymbol{0}$, $\boldsymbol{x}_k$ is identical with an exact solution of the optimization problem. And from eqs. (15) and (16), the following equation concerning $\boldsymbol{r}_{k+1}$ is obtained:

$$\begin{aligned} \boldsymbol{r}_{k+1} &= \boldsymbol{b} - A_{(n)}\boldsymbol{x}_{k+1} \\ &= \boldsymbol{b} - A_{(n)}(\boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k) \\ &= \boldsymbol{r}_k - \lambda_k A_{(n)} \boldsymbol{p}_k. \end{aligned} \tag{17}$$

When k is 0, an initial search direction vector $\boldsymbol{p}_0$ is set as an initial residual vector $\boldsymbol{r}_0$ in the CG method procedure, and $\boldsymbol{r}_1$ is obtained by using eqs. (13), (14), (16), and (17) as follows:

$$\begin{aligned} \boldsymbol{r}_1 &= \boldsymbol{r}_0 - \lambda_0 A_{(n)} \boldsymbol{r}_0 \\ &= \boldsymbol{r}_0 - \lambda_0 (A_{(n)}\boldsymbol{b} - A_{(n)}^2 \boldsymbol{x}_0) \\ &= \boldsymbol{r}_0 - \lambda_0 (n\boldsymbol{b} - nA_{(n)}\boldsymbol{x}_0) \\ &= \boldsymbol{r}_0 - \lambda_0 n \boldsymbol{r}_0. \end{aligned} \tag{18}$$

In the case where λ_0 is set as $1/n$, $\boldsymbol{r}_1$ becomes $\boldsymbol{0}$. That is, the procedure is completely accomplished in the first iteration. In this case, an exact solution $\tilde{\boldsymbol{f}}$ of the optimization problem is obtained using eqs. (10), (15), and (16) as follows:

$$\begin{aligned} \tilde{\boldsymbol{f}} &= \boldsymbol{x}_1 \\ &= \boldsymbol{x}_0 + \lambda_0 \boldsymbol{p}_0 \\ &= \boldsymbol{x}_0 + \frac{1}{n}(\boldsymbol{b} - A_{(n)}\boldsymbol{x}_0) \\ &= \frac{1}{n}(nI_{(n)} - A_{(n)})\boldsymbol{x}_0 + \frac{1}{n}\boldsymbol{b} \\ &= \frac{1}{n} J_{(n)} \boldsymbol{x}_0 + \frac{1}{n}\boldsymbol{b}. \end{aligned} \tag{19}$$

In Color2Gray, the lightness component of an input image $\boldsymbol{l} = (L_1^*, L_2^*, \ldots, L_n^*)^T$ is given as the initial approximate solution vector $\boldsymbol{x}_0$. Then, i-th component of $\tilde{\boldsymbol{f}}$, that is, $\tilde{f}_i$ can be written as follows:

$$\begin{aligned} \tilde{f}_i &= \frac{1}{n}\sum_{j=1}^{n} L_j^* + \frac{1}{n}\sum_{j=1}^{n} \delta_{ij} \\ &= \langle L^* \rangle + \frac{1}{n}\sum_{j=1}^{n} \delta_{ij}, \end{aligned} \tag{20}$$

where $\langle L^* \rangle$ stands for the average lightness of an input image.

4.2 Discussion of the analytical solution expressed in eq. (20)

The first term of eq. (20) is a common constant for all $\tilde{f}_i$s, and corresponds to the $\boldsymbol{c}$ portion in the case where a general solution is represented by $\boldsymbol{c} + \tilde{\boldsymbol{f}}$. In Color2Gray, the first term of eq. (20) has a role to retain the average lightness of an input image in an output image. This is described in the following. $\langle \tilde{f} \rangle$, that is, the average of the components of $\tilde{\boldsymbol{f}}$ is written as follows:

$$\begin{aligned} \langle \tilde{f} \rangle &= \frac{1}{n}\sum_{i=1}^{n} \tilde{f}_i \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\langle L^* \rangle + \frac{1}{n}\sum_{j=1}^{n} \delta_{ij}\right) \\ &= \langle L^* \rangle + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \delta_{ij}. \end{aligned} \tag{21}$$

Due to $\delta_{ij} + \delta_{ji} = 0$ and $\delta_{ii} = 0$ being satisfied, the following equation is satisfied:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \delta_{ij} = 0. \tag{22}$$

According to eqs. (21) and (22), it can be seen that $\langle \tilde{f} \rangle$ is identical with $\langle L^* \rangle$. As a result, it is understood that the gray-level shifting is originally and appropriately achieved in eq. (20) because the average lightness of the input image is $\langle L^* \rangle$. Therefore, it is proved that the gray-level represented by eq. (20) is the final output in Color2Gray with $\rho = \infty$. The color-to-monochrome conversion using eq. (20) is termed "Original Color2Gray" in this paper.

From eq. (20), it is confirmed that the average lightness and global consistencies are realized in Original Color2Gray. The average lightness consistency is confirmed in eqs. (21) and (22); and it can be easily understood that the identical gray-level is assigned to the same colors in an input image by the conversion of eq. (20), which means the global consistency is realized. Although the optimization problem defined by eqs. (3)–(8) has these characteristics inherently, the analytical solution of it represents those characteristics explicitly.

In Original Color2Gray, an output $\tilde{f}_i$ is given by adding the average of δs concerning i-th pixel to the average lightness as shown in eq. (20). On the other hand, eq. (20) can be rewritten as follows:

$$\tilde{f}_i = \frac{1}{n}\sum_{j=1}^{n} L^{*\prime}_{i,j}, \tag{23}$$

where $L^{*\prime}_{i,j}$ means the modified lightness of i-th pixel calculated in consideration with j-th pixel, and is defined by

$$L^{*\prime}_{i,j} = L^*_j + \delta_{ij} = \begin{cases} L^*_i & |\Delta L^*_{ij}| > \Phi_\alpha(\|\Delta \boldsymbol{C}_{ij}\|) \\ L^*_j + \mathrm{sign}(\Delta \boldsymbol{C}_{ij} \cdot \boldsymbol{v}_\theta)\Phi_\alpha(\|\Delta \boldsymbol{C}_{ij}\|) & \text{otherwise} \end{cases}. \tag{24}$$

As shown in eq. (23), the output gray-level $\tilde{f}_i$ in Original Color2Gray is considered the average of the modified lightness $L^{*\prime}$s concerning i-th pixel. Equation (24) shows that $L^{*\prime}_{i,j}$ becomes L^*_i, that is, the modification is not applied when the lightness difference between i-th and j-th pixel is adequately large. The modification is only applied when the chrominance difference is dominant.

Here, the case where the lightness difference is dominant in all pixel pairs is considered. In other words, when the signed color distance δ_{ij} is always given by ΔL^*_{ij}, eq. (23) becomes as follows using eq. (24):

$$\tilde{f}_i = \frac{1}{n}\sum_{j=1}^{n} L^*_i = L^*_i. \tag{25}$$

Thus, it is confirmed that the Color2Gray output is identical to the lightness component of an input image L^* when the lightness difference is more dominant than the chrominance difference in an input image, that is, the lightness modification is not needed.

The analytical solution of the optimization problem of Color2Gray enables us to consider the properties of global and average lightness consistencies. However, the computational cost to obtain the output gray-levels is not greatly reduced using the solution. The computational cost of the Original Color2Gray expressed in eq. (20) is $O(n^2)$ because n times addition of δ_{ij} is needed to obtain $\tilde{f}_i$ and there are n pixels ($\tilde{f}_i$s) in the image of concern. Though the computational cost of the calculation of eq. (20) is slightly less than that in solving the optimization problem of Color2Gray by the CG method, that cost is still $O(n^2)$. Hence, the computational cost reduction of Original Color2Gray is achieved in the next section, and a fast color removal method is proposed based on it.

5. Proposed Fast Color Removal Algorithm

When the decoloring of a color image is carried out by Original Color2Gray, the calculations for the second term of eq. (20) account for most of the entire computational cost. In this section, we propose the fast Color2Gray algorithm using the analytical solution in eq. (20). The following steps are implemented:

A. modification of the definition of the signed color distance,
B. consideration of the number of same color pixels,
C. color quantization,

and

D. lightness compensation.

5.1 *Method A: modification of the definition of the signed color distance*

A new signed color distance δ'_{ij} is defined by modifying eq. (6), which is one of the definitions of δ_{ij}, as follows:

$$\Phi'_\alpha(x) = \alpha \tanh(\lfloor x \rfloor/\alpha). \tag{26}$$

$\lfloor x \rfloor$ is a floor function and means the maximum integer number less than or equal to x. In practice, a look-up table (LUT) for $\lfloor x \rfloor$s and $\Phi'_\alpha(\lfloor x \rfloor)$s is used. The cost of calculating the hyperbolic tangent value is reduced by using the LUT while the quality of the resulting image is hardly deteriorated. Furthermore, using an LUT for $\lfloor x \rfloor$s and $\sqrt{\lfloor x \rfloor}$s is proposed here. To calculate the signed color distance defined in eqs. (5)–(8), it is necessary to calculate the norm of $\Delta \boldsymbol{C}_{ij}$ which contains the calculation of a square root value defined by

$$\|\Delta \boldsymbol{C}_{ij}\| = \sqrt{\Delta a^{*2}_{ij} + \Delta b^{*2}_{ij}}. \tag{27}$$

The cost of calculating square root values can also be reduced by using an LUT.

Concretely, $\Phi'_\alpha(\|\Delta \boldsymbol{C}_{ij}\|)$ is calculated using the LUTs T_1 and T_2 as follows:

$$\Phi'_\alpha(\|\Delta \boldsymbol{C}_{ij}\|) = T_2(T_1(\lfloor \Delta a^{*2}_{ij} + \Delta b^{*2}_{ij} \rfloor)) \tag{28}$$

with

$$T_1(z) = \lfloor \sqrt{z} \rfloor, \tag{29}$$

$$T_2(z) = \alpha \tanh(z/\alpha), \tag{30}$$

where z in eqs. (29) and (30) refers to integer numbers.

5.2 *Method B: consideration of the number of same color pixels*

Here, it is assumed that the color number i' is given for colors appearing in an input image, and the number of pixels whose color number is i' is also given as $s_{i'}$. Then, by using eq. (20), the output gray-level $\tilde{f}'_{i'}$ for i'-th color can be given by:

$$\tilde{f}'_{i'} = \langle L^* \rangle + \frac{1}{n}\sum_{j'=1}^{m} s_{j'}\delta_{i'j'}, \tag{31}$$

where m means the number of colors included in an input image. The output gray-level of i-th pixel $\tilde{f}_i$ is obtained by:

$$\tilde{f}_i = \tilde{f}'_{T_3(i)}, \tag{32}$$

where T_3 is an LUT concerning pixel numbers and color numbers. $T_3(i)$ indicates the color number of the i-th pixel.

m is always smaller than or equal to n, and in most cases, it is substantially smaller. Hence the computational cost of eq. (31) is quite a bit less than that of eq. (20). Note that the computational cost of building T_3 and counting $s_{i'}$ must be modest for high speed processing. The way to carry them out is described below.

First, the label l_i for i-th pixel is given by

$$l_i = 256^2 r_i + 256^1 g_i + 256^0 b_i. \tag{33}$$

In eq. (33), it is assumed that RGB values are recorded in 256 levels. In the case where these are recorded in κ levels, 256 in eq. (33) should be changed to κ. Then the pixels in an input image are sorted in the numerical order of l. The fast sorting algorithm, quicksort,[24,25] is used here. For example, sorted pixels in l's order become $\{l_{34}, l_{52}, l_3, l_4, l_{127}, l_{85}, l_{63}, \ldots\}$. And the values of l become $\{0, 0, 1, 1, 1, 3, 7, \ldots\}$. In this example, the color number $i' = 1$ is assigned to the color of l_{34} and l_{52}, that is, $(r, g, b) = (0, 0, 0)$. Similarly, the color number $i' = 2$ is assigned to the color $(r, g, b) = (0, 0, 1)$, the color number $i' = 3$ is assigned to the color $(r, g, b) = (0, 0, 3)$, and so on. The colors excluded from an input image, such as $(r, g, b) = (0, 0, 2)$ in this example, are not assigned a color number.

The color number i's can be assigned by scanning the sorted pixels, and the number of pixels $s_{i'}$s can also be acquired simultaneously.

5.3 Method C: color quantization

In this section, we propose a computational cost reduction by the color quantization. The RGB values after the quantization (r', g', b') are given by

$$r'_i = \beta\lfloor r_i/\beta \rfloor + (\beta - 1)/2, \tag{34}$$

$$g'_i = \beta\lfloor g_i/\beta \rfloor + (\beta - 1)/2, \tag{35}$$

$$b'_i = \beta\lfloor b_i/\beta \rfloor + (\beta - 1)/2, \tag{36}$$

where β stands for the width of quantization and is a positive integer number. The computational cost reduction can be achieved by applying Method B to the color quantized image.

The number of colors m' after the quantization is usually significantly smaller than that of a bare input image, though it depends on the color distribution of the input image. And m' is always smaller than or equal to m.

The degree of the computational cost reduction is dependent on the color distribution of an input image and the value of β. An efficient reduction is achieved by assigning β a large number. However, the difference between the output image obtained by the Color2Gray with color quantization and that of Original Color2Gray is increased as β increases. On the other hand, when β is set as 1, it is identical with the case in which only Method B is used.

5.4 Method D: lightness compensation

Method C (color quantization) sometimes causes serious deterioration in the quality of an output image because the colors located in a cubic quantization region are transformed into an identical representative color. The means of compensating the deterioration is proposed here as "Method D". The compensation values concerning the input colors and quantized colors are calculated, and are used to decrease the deterioration.

The final gray-level of i-th pixel $\tilde{f}_i$ using method D is obtained as follows:

$$\tilde{f}_i = \tilde{f}'_{T_3(i)} + p_i, \tag{37}$$

where p_i stands for a compensation value for i-th pixel. Color2Gray output can be used for the colors in the same quantization cubic as a candidate of p_i. However, the output becomes similar to the lightness component L^* of the colors because the differences of hue and chroma in a cubic are small. Especially, when the lightness differences are more dominant than the chrominance differences between the colors of concern, Color2Gray output is identical with L^* as mentioned in §4.2. Therefore, it is useful that the lightness component is simply used as the compensation value as follows:

$$p_i = L^*_i - L^{*\prime}_{T_3(i)}, \tag{38}$$

where $L^{*\prime}_{T_3(i)}$ means the lightness component of the quantized color of which the number is $T_3(i)$, and $L^*_i - L^{*\prime}_{T_3(i)}$ stands for the lightness difference between the color of i-th pixel and its quantized color.

5.5 Proposed fast algorithm for the color removal

The algorithm proposed in this paper, that is, the fast Color2Gray algorithm, is constructed using Methods A–D, and is explained here. First, the color quantization is carried out using eqs. (34)–(36); then, color numbers i's are assigned to quantized RGB values (r', g', b'). The output gray-level for i'-th quantized color $\tilde{f}'_{i'}$ is obtained by

$$\tilde{f}'_{i'} = \langle L^* \rangle + \frac{1}{n} \sum_{j'=1}^{m'} s_{j'} \delta'_{i'j'}. \tag{39}$$

Finally, the output gray-level of i-th pixel $\tilde{f}_i$ is obtained as follows:

$$\tilde{f}_i = \tilde{f}'_{T_3(i)} + (L^*_i - L^{*\prime}_{T_3(i)}). \tag{40}$$

Here, $T_3(i)$ indicates the color number i' concerning the color of i-th pixel after the quantization. We call this fast algorithm "Fast Color2Gray".

6. Experimental Results

The attempt is made to verify the validity and the effectiveness of the proposed Fast Color2Gray by applying it to some images.

6.1 Conditions of experiments

In the experiments, Impression shown in Fig. 1(a), "Lenna" and "Parrots" included in the standard image database SIDBA,[26,27] "Map" and "Voiture" drawn by Bli, in which each image is 24 bits/pixel and color-scale, are employed. Their sizes are shown in Table 1. Figure 3 shows the four images without Impression.

The methods used for comparison were those by Bala and Eschbach,[4] and Decolorize,[8] which can remove the color very fast. The Bala and Eschbach method is representative of the UM-based methods and is here referred to as "selective unsharp masking" (SUM). The parameters of SUM are set as $(K, B1, B2, N) = (1, 15, 40, 5)$ based on ref. 4. Among them, a parameter related to the calculation time is only N, and $N = 5$ guarantees that the method works within a short time. On the other hand, Decolorize is representative of the projection-based methods. Its parameters are set as $(\lambda, \sigma, \eta) = (0.3, 25, 0.001)$ based on ref. 8. In Decolorize, the parameter setting is related only to the quality of the resulting image and has no relevance to the calculation time. Moreover, Original Color2Gray expressed in eq. (20) is also employed for comparison. In Original Color2Gray, there are

Table 1. Sizes of images employed in experiments.

	Impression	Lenna	Map	Parrots	Voiture
Image size (width × height)	128 × 92	512 × 512	172 × 172	256 × 256	125 × 125
Number of pixels	11776	262144	29584	65536	15625

Table 2. Calculation times of various methods. A unit is second.

		Impression	Lenna	Map	Parrots	Voiture
HDTV standard		0.01	0.13	0.02	0.05	0.01
SUM		0.02	0.36	0.07	0.15	0.03
Decolorize		0.01	0.25	0.04	0.09	0.02
Original Color2Gray		10.2	4860	30.8	300	17.8
Fast Color2Gray	$\beta = 1$	1.33	120	0.02	4.75	0.26
	$\beta = 2$	0.49	15.9	0.02	4.04	0.11
	$\beta = 4$	0.09	1.17	0.02	1.00	0.04
	$\beta = 8$	0.02	0.25	0.02	0.19	0.02
	$\beta = 16$	0.01	0.22	0.02	0.07	0.02

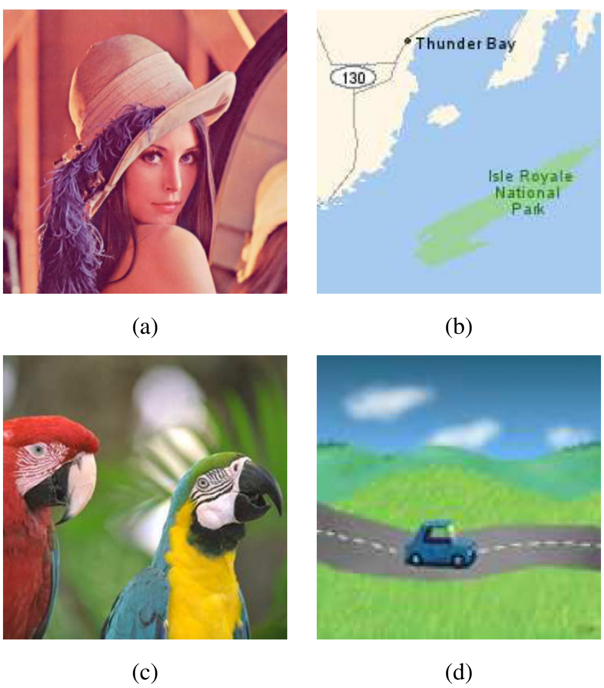


Fig. 3. (Color online) Images employed in experiments: (a) "Lenna", (b) "Map", (c) "Parrots", (d) "Voiture" drawn by Bli.

two parameters α and θ, and these are scarcely related to the computational cost. Therefore, the values producing good results for many images are employed, and these are actually $\alpha = 15$ and $\theta = \pi/4$. In Fast Color2Gray, α and θ are also set as 15 and $\pi/4$, respectively.

In the experiments, we mainly use the HDTV standard expressed in eq. (1) to calculate lightness components of images because its primaries are identical with those of the standard RGB color space,[28] which is a common color space for the displays of personal computers. However, only Decolorize employs the SDTV standard expressed in eq. (2) to calculate lightness components; hence we also show

Table 3. The number of colors after the quantization with various quantization widths β.

β	Impression	Lenna	Map	Parrots	Voiture
1	10752	148279	105	27632	4988
2	7291	53812	105	25766	3126
4	2784	13018	105	12215	1694
8	758	2596	101	3621	749
16	198	498	77	811	290

the monochrome images obtained by this standard in the experimental results.

The CPU employed in the experiments is Intel® Core™ 2 Duo 3.0 GHz.

6.2 *Calculation time*

The calculation times of the methods employed in the experiments are shown in Table 2. From Tables 1 and 2, it is observed that the calculation times of the comparison methods except Original Color2Gray are roughly in $O(n)$, and that of Original Color2Gray is $O(n^2)$. The number of pixels of Lenna is four times larger than that of Parrots, and the calculation time of Original Color2Gray for Lenna is about sixteen times larger than that for Parrots. Moreover, the calculation time of Original Color2Gray is very long in comparison with other methods.

Concerning the Fast Color2Gray, Table 3 shows the number of colors after the quantization with various quantization width β. From Tables 2 and 3, it can be easily understood that the number of colors m' and the calculation times are reduced as the value of β increases, and Fast Color2Gray can process quickly and equivalent to the other methods when β is set as 8. The computational cost of Fast Color2Gray can be expressed as $O(n \log n + m'^2)$. $O(n \log n)$ and $O(m'^2)$ are related to the sorting of pixels by the quicksort and the calculation of eq. (39), respectively. The computational cost of quicksort is independent of β, and theoretically becomes a constant for the image consisting of

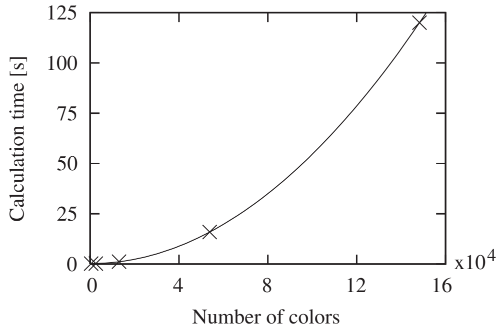


Fig. 4. Relationship between the number of colors after quantization and the calculation time of Fast Color2Gray for Lenna. A line is a quadratic curve.

n pixels. Figure 4 shows the relationship between m' and the calculation time of Fast Color2Gray for Lenna. The line in Fig. 4 is a quadratic curve. Thus it is actually confirmed that the computational cost of Fast Color2Gray is almost $O(m'^2)$. Moreover, the calculation time of the sorting is admittedly not great in comparison with the calculation of eq. (39).

6.3 *Effects of Methods A–D and the parameter setting of Fast Color2Gray*

From the viewpoint of the computational cost, β should be set as a large value in Fast Color2Gray. However, the quality of a resulting image is deteriorated under large β. In this paper, we regard the resulting image obtained by Original Color2Gray as an ideal monochrome image. The difference between the ideal image and the image obtained by Fast Color2Gray is evaluated by a mean square error (MSE) defined as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(f_i^{\mathrm{Fast}} - f_i^{\mathrm{Ori}})^2, \tag{41}$$

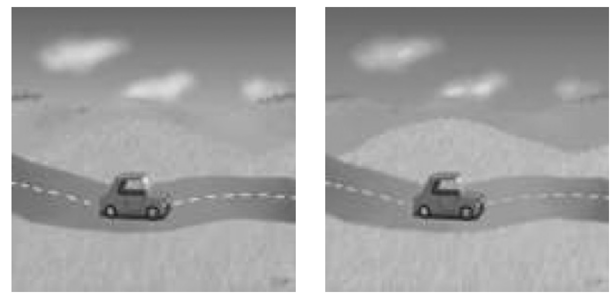
where f^{Fast} and f^{Ori} stand for the resulting monochrome images obtained by Fast Color2Gray and Original Color2Gray, respectively. MSEs concerning the five test images are shown in Table 4.

The case where β is set as 1 is considered first. In this case, the number of colors m' after quantization is identical with the number of colors m in an input image. From Table 2, it is observed that the calculation time of Fast Color2Gray is about 0.1 times less than that of Original Color2Gray for Impression. This reduction in calculation time is mainly due to Method A because m is nearly equal to n as shown in Tables 1 and 3. When β is set as 1, the MSEs concerning the results are less than 1 as shown in Table 4, and such small differences cannot be detected by human vision. Hence it can be said that Method A reduces the calculation time without deterioration of the image quality. On the other hand, as shown in the results for other images, a huge calculation time reduction is achieved because the values of m are very small compared with those of n; this reduction is due to Method B.

Then, the case where β is larger than 1 is considered, that is, the effect of Method C. From Tables 2 and 4, it is observed that the large calculation time reduction is achieved by increasing the value of β though deterioration of the quality of the resulting images also increases.

Table 4. MSEs between the resulting images obtained by Fast Color2Gray and Original Color2Gray.

β	Impression	Lenna	Map	Parrots	Voiture
1	0.26	0.14	0.0006	0.08	0.12
2	0.62	0.49	0.08	0.57	0.57
4	0.97	0.66	0.58	0.96	0.88
8	2.41	1.38	1.10	2.55	2.48
16	9.02	4.94	1.40	8.09	5.42



(a) (b)

Fig. 5. Color removal results of Voiture: (a) lightness component in HDTV standard; (b) Original Color2Gray.

Finally, the effect of Method D is confirmed here. For example, the converted monochrome images of Voiture obtained by HDTV standard and Original Color2Gray are shown in Figs. 5(a) and 5(b), respectively. In Fig. 5(a), it is hard to see the boundary of the grassy plain and mountains which can be discriminated in the original color image. That is, Voiture contains iso-lightness different colors. The boundary is, however, clearly seen in Fig. 5(b). The resulting images obtained by Fast Color2Gray with various β are shown in Figs. 6 and 7. Figure 6 shows the results when Method D is not used, and Fig. 7 shows the results when it is employed. (a), (b), (c), and (d) shown in Figs. 6 and 7 are the resulting images when β is set as 2, 4, 8, and 16, respectively. Especially in Fig. 6(d), deteriorations in the image quality are apparent in the road and the sky portions, while these deteriorations are reduced when Method D is used as shown in Fig. 7(d). Therefore, the effectiveness of Method D is confirmed.

From Table 2, $\beta = 8$ appears to be sufficient for the calculation time in comparison with SUM and Decolorize. Note that the lightness extraction by HDTV standard is the simplest and fastest method. Moreover, $\beta = 8$ is also appears appropriate for the image quality because MSEs between the resulting images obtained by Fast Color2Gray and Original Color2Gray are about 2 when β is set as 8 as shown in Table 4. Indeed, the image quality of Fig. 7(c), which is the resulting image of Fast Color2Gray with $\beta = 8$ for Voiture, is acceptable. Therefore, β is set as 8 in the following experiments.

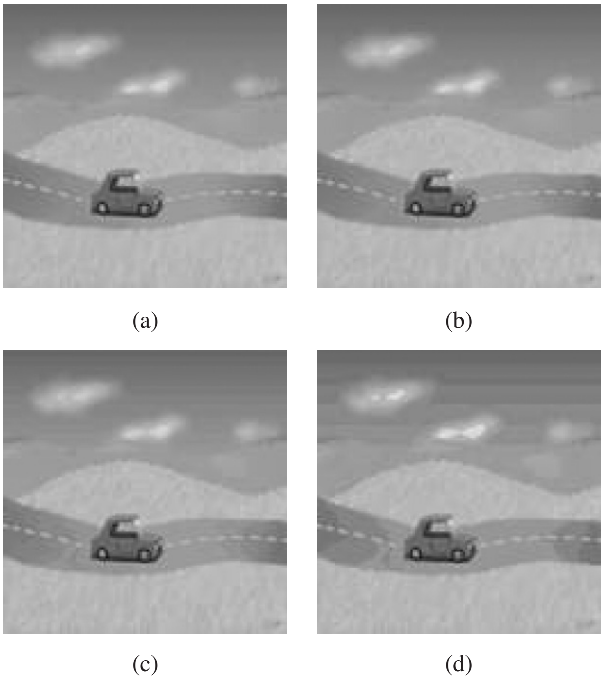


Fig. 6. Color removal results of Voiture obtained by Fast Color2Gray without Method D: (a) $\beta = 2$; (b) $\beta = 4$; (c) $\beta = 8$; (d) $\beta = 16$.

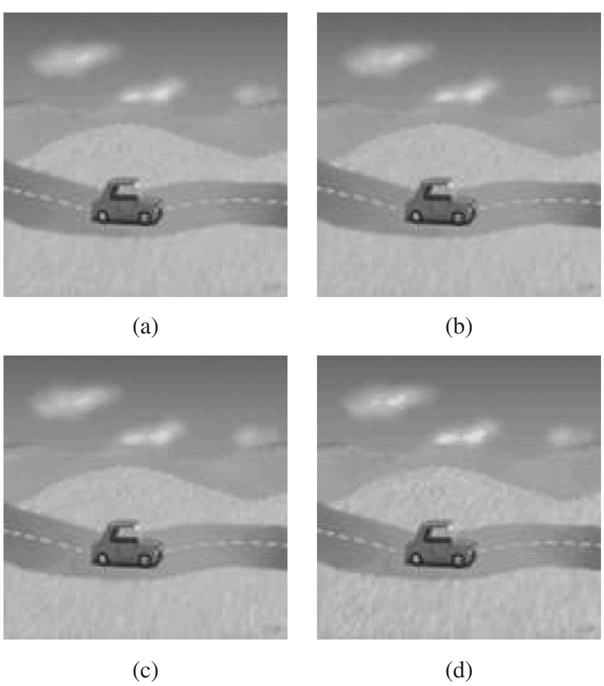


Fig. 7. Color removal results of Voiture obtained by Fast Color2Gray with Method D: (a) $\beta = 2$; (b) $\beta = 4$; (c) $\beta = 8$; (d) $\beta = 16$.

6.4 Subjective evaluation

In this section, the resulting images obtained by Fast Color2Gray with $\beta = 8$ and other methods are shown and discussed subjectively.

Figure 8 shows the resulting image of Impression obtained by Fast Color2Gray. Little deterioration of the image quality caused by the color quantization is seen in comparison with that obtained by Original Color2Gray shown in Fig. 2(a). This result is deemed adequate.

Fig. 8. Resulting image of Impression obtained by Fast Color2Gray.

The monochrome images of Voiture obtained by the methods for comparison are shown in Fig. 9. Figure 9(a) shows the lightness component obtained by SDTV standard, while (b) and (c) show the resulting images obtained by SUM and Decolorize, respectively. Though the boundary of the grassy plain and mountains can be discriminated in the result obtained by SUM, quality of the image is insufficient; the resulting image obtained by Decolorize, however, is visually good. The conversion result of Decolorize can be viewed as superior to that of Original Color2Gray and Fast Color2Gray from the viewpoint of the contrast of the plain and mountains, and of the mountains and sky.

Figure 10 shows the resulting monochrome images of Map, where (a) and (b) show the lightness component obtained by HDTV and SDTV standard, respectively. Figures 10(c), 10(d), 10(e), and 10(f) show the resulting images obtained by SUM, Decolorize, Original Color2Gray, and Fast Color2Gray, respectively. An island in Map has completely disappeared in Fig. 10(a) and is hardly seen in (b). In the result obtained by SUM, the island can be distinguished. However, the quality of the image is not good because artifacts are observed around the island, and they break the global consistency. In the resulting image obtained by Decolorize, the visibility of the island is poor. This conversion result shows one limitation of the projection-based method: that the ability of the color-to-monochrome conversion by the linear projection is low as mentioned in §2. In contrast, excellent resulting images are shown in Figs. 10(e) and 10(f), which are obtained by Original Color2Gray and Fast Color2Gray, respectively.

Figures 11 and 12 show the monochrome conversion results for Parrots and Lenna, respectively. The images are shown in the same manner as in the case of Map. For these images, the lightness components are seen to be a sufficient gray-level as shown in Figs. 11(a), 11(b), 12(a), and 12(b). Hence, the special conversion is not needed for these images. However, artifacts are observed in the results obtained by SUM as shown in Figs. 11(c) and 12(c). As shown in Figs. 11(c), 11(d), 12(c), and 12(d), though the image quality of the results obtained by Decolorize is relatively good in comparison with SUM, unnecessary conversions have been carried out. As shown in Fig. 11(d), the body of the right parrot and the head and body of the left

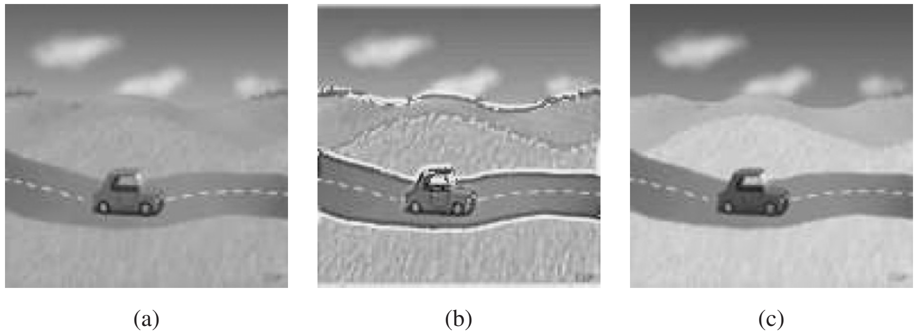


Fig. 9. Color removal results of Voiture obtained by several methods: (a) lightness component in SDTV standard; (b) SUM; (c) Decolorize.

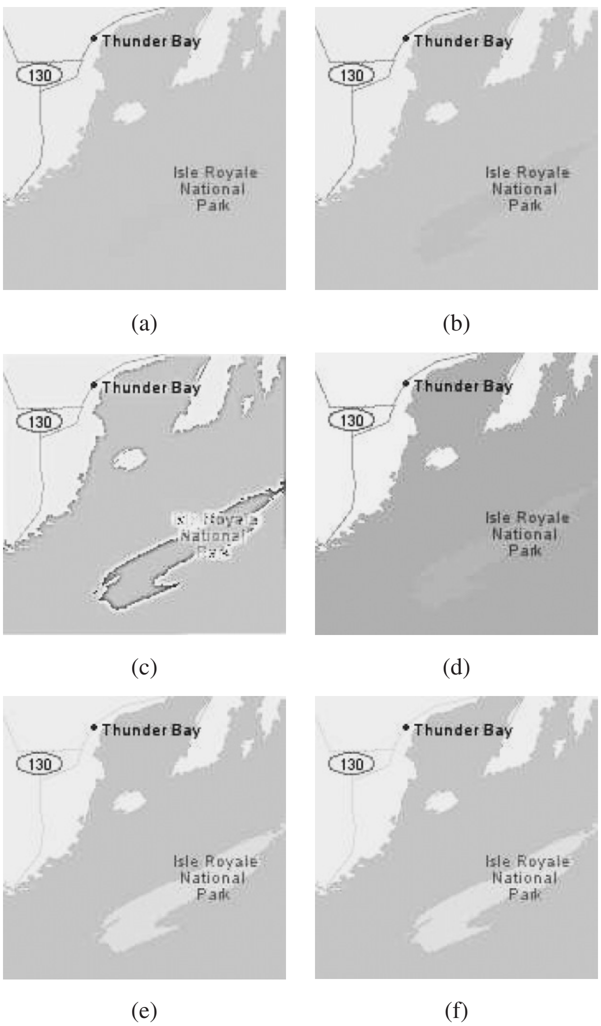


Fig. 10. Color removal results of Map obtained by various methods: (a) lightness component in HDTV standard; (b) lightness component in SDTV standard; (c) SUM; (d) Decolorize; (e) Original Color2Gray; (f) Fast Color2Gray.

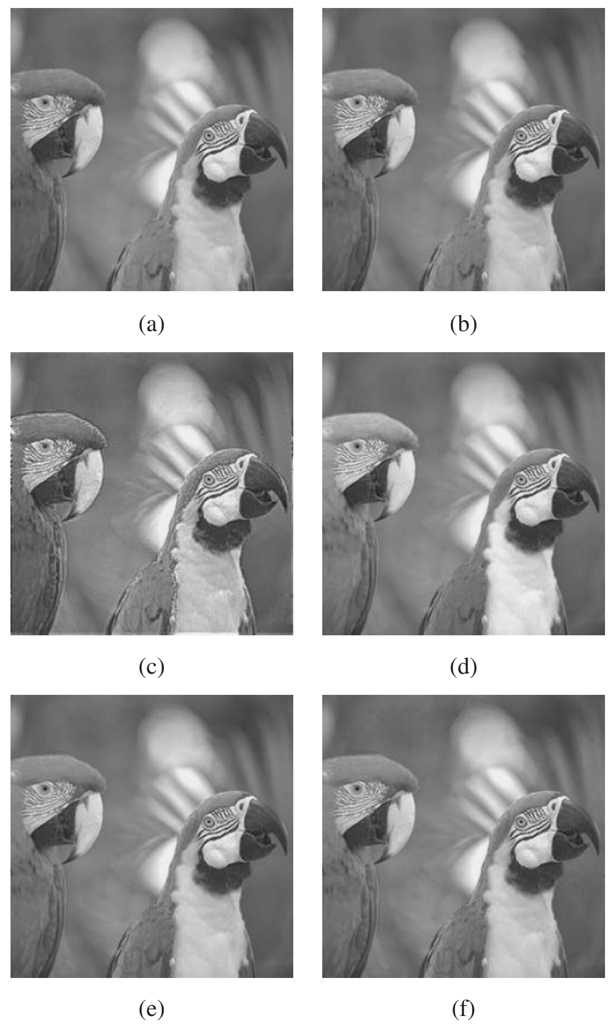


Fig. 11. Color removal results of Parrots obtained by various methods: (a) lightness component in HDTV standard; (b) lightness component in SDTV standard; (c) SUM; (d) Decolorize; (e) Original Color2Gray; (f) Fast Color2Gray.

one (yellow and red colors) are converted into brighter gray-level than their lightness component in Decolorize. And for the result of Lenna shown in Fig. 12(d), the average gray-level is obviously higher than the lightness components shown in Figs. 12(a) and 12(b). The average lightness consistency is not realized in this case. On the other hand, in the Original Color2Gray and Fast Color2Gray, the monochrome images which are similar to the lightness components in HDTV standard are obtained as shown in Figs. 11(e), 11(f), 12(e), and 12(f). These results show that the unnecessary conversions are not executed in the Original Color2Gray and Fast Color2Gray.

Fig. 12. Color removal results of Lenna obtained by various methods: (a) lightness component in HDTV standard; (b) lightness component in SDTV standard; (c) SUM; (d) Decolorize; (e) Original Color2Gray; (f) Fast Color2Gray.

6.5 Quantitative evaluation

Finally, the conversion results shown above are evaluated quantitatively.

Table 5 shows the average gray-levels of the resulting monochrome images. SUM, Original Color2Gray, and Fast Color2Gray satisfy the average lightness consistency. However, this consistency is not satisfied in Decolorize, as indicated especially in the result of Lenna.

Table 6 shows MSEs between the images obtained by the means considering with difference of colors and the lightness components. Concerning Decolorize, the lightness component obtained by the SDTV standard is employed to calculate MSE, and that obtained by the HDTV standard is employed for the other methods. From Table 6, the resulting images obtained by Original Color2Gray and Fast Color2Gray are seen to be relatively similar to the lightness component in comparison with other methods. Thus the color-to-monochrome conversion giving importance to lightness components is realized in Color2Gray. That saves the unnecessary conversion, which is especially observed in the resulting images of Parrots and Lenna obtained by SUM and Decolorize.

These results confirm the validity and effectiveness of Fast Color2Gray.

Table 5. Average gray-levels of resulting images obtained by various methods.

	Impression	Lenna	Map	Parrots	Voiture
HDTV standard	121.9	124.5	207.1	123.4	156.0
SDTV standard	121.1	132.6	206.0	123.4	151.1
SUM	123.3	125.7	207.9	124.3	157.7
Decolorize	115.7	150.3	193.7	131.1	161.8
Original Color2Gray	122.0	124.5	207.7	123.4	156.0
Fast Color2Gray	122.5	124.9	207.8	123.9	156.5

Table 6. MSEs between lightness components and resulting images obtained by various methods. Concerning the Decolorize, the lightness component obtained by SDTV standard is employed to calculate MSE, and that obtained by HDTV standard is employed for the other methods.

	Impression	Lenna	Map	Parrots	Voiture
SUM	250.6	180.4	278.0	112.8	521.7
Decolorize	41.1	432.3	289.5	199.5	908.7
Original Color2Gray	88.8	51.6	87.2	81.4	113.1
Fast Color2Gray	89.2	50.5	91.8	80.9	122.1

7. Conclusions

This paper first introduced the salience-preserving color removal method called Color2Gray proposed by Gooch *et al.*[3] Then Color2Gray was analyzed mathematically, and the analytical solution of its optimization problem was reported. Thereafter, Fast Color2Gray which reduced the processing time significantly using color quantization and various LUTs was proposed based on this analytical solution. Finally, the validity and the effectiveness of the proposed Fast Color2Gray algorithm were proven by experiments.

Appendix A: Proof of Eq. (13)

Using eq. (10), $A_{(n)}^2$ is obtained as follows:

$$\begin{aligned} A_{(n)}^2 &= (nI_{(n)} - J_{(n)})(nI_{(n)} - J_{(n)}) \\ &= n^2 I_{(n)} - 2nJ_{(n)} + J_{(n)}^2. \end{aligned} \tag{A·1}$$

Furthermore, $J_{(n)}^2$ is can be represented as

$$\begin{aligned} J_{(n)}^2 &= \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \\ &= \begin{pmatrix} n & \cdots & n \\ \vdots & \ddots & \vdots \\ n & \cdots & n \end{pmatrix} \\ &= nJ_{(n)}, \end{aligned} \tag{A·2}$$

and then eq. (A·1) can be rewritten as

$$\begin{aligned} A_{(n)}^2 &= n^2 I_{(n)} - 2nJ_{(n)} + nJ_{(n)} \\ &= n^2 I_{(n)} - nJ_{(n)} \\ &= n(nI_{(n)} - J_{(n)}) \\ &= nA_{(n)}. \end{aligned} \tag{A·3}$$

Therefore it is proved that eq. (13) is always satisfied. □

Appendix B: Proof of Eq. (14)

Using eq. (10), $A_{(n)}\boldsymbol{b}$ is represented as follows:

$$\begin{aligned} A_{(n)}\boldsymbol{b} &= (nI_{(n)} - J_{(n)})\boldsymbol{b} \\ &= n\boldsymbol{b} - J_{(n)}\boldsymbol{b}. \end{aligned} \tag{B·1}$$

Here $J_{(n)}\boldsymbol{b}$ is substituted as $\boldsymbol{b}'$. Then k-th element of $\boldsymbol{b}'$, that is, b'_k is represented as follows:

$$b'_k = \sum_{i=1}^{n}\sum_{j=1}^{n} \delta_{ij}. \tag{B·2}$$

b'_k is independent of k. Further, from the definition of the signed color distance δ_{ij}, the relationships $\delta_{ij} + \delta_{ji} = 0$ and $\delta_{ii} = 0$ are always satisfied, and b'_k for all k becomes

$$b'_k = 0. \tag{B·3}$$

Hence the relationship $J_{(n)}\boldsymbol{b} = \boldsymbol{0}$ is satisfied. Then, by substituting the relationship into eq. (B·1), it is proved that eq. (14) is satisfied. □

References

1) International Telecommunication Union Recommendation BT.709-5 (2002).
2) International Telecommunication Union Recommendation BT.601-6 (2007).
3) A. A. Gooch, S. C. Olsen, J. Tumblin, and B. Gooch: ACM Trans. Graphics **24** (2005) 634.
4) R. Bala and R. Eschbach: Proc. IS&T/SID 12th Color Imaging Conf., 2004, p. 82.
5) A. Alsam and O. Kolas: Proc. IS&T/SID 14th Color Imaging Conf., 2006, p. 263.
6) K. Smith, P.-E. Landes, J. Thollot, and K. Myszowski: Comput. Graphics Forum **27** (2008) 193.
7) Y. Zhang, C. Chen, J. Bu, M. Song, C. Pang, and Z. Sun: Proc. 2008 Congr. Image and Signal Processing, 2008, Vol. 3, p. 451.
8) M. Grundland and N. A. Dodgson: Pattern Recognition **40** (2007) 2891.
9) K. Rasche, R. Geist, and J. Westall: IEEE Comput. Graphics Appl. **25** (2005) 22.
10) K. Inoue and K. Urahama: Denshi Joho Tsushin Gakkai Ronbunshi A **89** (2006) 80 [in Japanese].
11) S. Matsumura and I. Kumazawa: Denshi Joho Tsushin Gakkai Ronbunshi D **90** (2007) 1977 [in Japanese].
12) G. Tanaka, N. Suetake, and E. Uchino: Denshi Joho Tsushin Gakkai Ronbunshi A **90** (2007) 673 [in Japanese].
13) G. Tanaka, N. Suetake, and E. Uchino: Proc. 2007 Int. Symp. Intelligent Signal Processing and Communication Systems, 2007, p. 112.
14) K. Rasche, R. Geist, and J. Westall: Comput. Graphics Forum **24** (2005) 423.
15) Y. Bai, S. J. Harrington, and J. Taber: Proc. SPIE **4300** (2000) 444.
16) R. Bala and K. Braun: Proc. SPIE **5293** (2004) 196.
17) R. L. de Queiroz and K. M. Braun: IEEE Trans. Image Process. **15** (2006) 1464.
18) L. Neumann, M. Čadík, and A. Nemcsics: in *Computational Aesthetics 2007*, ed. D. W. Cunningham, G. Meyer, L. Neumann, A. Dunning, and R. Paricio (A. K. Peters, Wellesley, MA, 2007) p. 73.
19) G. R. Kuhn, M. M. Oliveira, and L. A. F. Fernandes: Visual Comput. **24** (2008) 505.
20) M. Čadík: Comput. Graphics Forum **27** (2008) 1745.
21) J. R. Shewchuk: http://www-2.cs.cmu.edu/~jrs/jrspapers.html#cg (1994).
22) A. K. Jain: *Fundamentals of Digital Image Processing* (Prentice-Hall, Upper Saddle River, NJ, 1989).
23) I. T. Jolliffe: *Principal Component Analysis* (Springer, Heidelberg, 2002) 2nd ed.
24) C. A. R. Hoare: Comput. J. **5** (1962) 10.
25) W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery: *Numerical Recipes in C* (Cambridge University Press, Cambridge, U.K., 1992) 2nd ed.
26) M. Sakauchi, Y. Ohsawa, M. Sone, and M. Onoe: Television Gakkai Gijutsu Hokoku (1984) Vol. 8, No. 38, p. 7 [in Japanese].
27) http://www.ess.ic.kanagawa-it.ac.jp/app_images_j.html
28) M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta: http://www.w3.org/Graphics/Color/sRGB.html (1996).