Fast Color Removal Method
Considering Differences between Colors

Go Tanaka, Noriaki Suetake, and Eiji Uchino

Graduate School of Science and Engineering, Yamaguchi University

1677–1 Yoshida, Yamaguchi, 753–8512 Japan

E-mail: tanaka@ic.sci.yamaguchi-u.ac.jp, {suetake, uchino}@sci.yamaguchi-u.ac.jp

Abstract—When a color image is converted into a monochrome one, luminance components of the pixels have been used as gray-levels for the representation of the monochrome image in HDTV standard. However, saliencies of the image embedded only in the chrominance components are disappeared in the monochrome image converted by using luminance components. To cope with this problem, A.A. Gooch et al. have proposed the salience-preserving color removal method called "Color2Gray." The monochrome image well reflected the impression of an input color image can be yielded by Color2Gray. However, the calculation cost of that algorithm is tremendous, and its utility is not so much. In this paper, fast Color2Gray algorithms are proposed. The effectiveness of the proposed method is illustrated through the experiments.

I. Introduction

It is well known that the colors, which human beings can perceive, are characterized by three elements. Hue, luminance, and chroma are often used as the three elements. On the other hand, a mixture of some of primary colors can produce various colors, and digital color images are recorded by using three primary colors, those are red, green, and blue (RGB), in most cases.

In the monochrome conversion of a color image, the luminance component of pixel has been used as a gray-level in the converted monochrome image so far. In HDTV standard, the human visual sensitivity is considered, and a gray-level Y_i of ith pixel is represented by:

$$Y_i = 0.2126r_i + 0.7152g_i + 0.0722b_i, \qquad (1)$$

where (r_i, g_i, b_i) stand for RGB components [1]. The monochrome conversion represented by the weighted sum of RGB components (Eq.(1)) does not require the complex calculation, and can obtain good monochrome images in many cases. However, the monochrome conversion by Eq.(1) does not always reflect the color information appropriately. For example, an image "Impression" by C. Monet and its monochrome image converted by Eq.(1) are shown in Figs.1(a) and (b), respectively. From Fig.1(b), it is observed that the sun and its reflection in the water almost disappear.

To cope with this problem, A.A. Gooch et al. have proposed the salience-preserving color removal method called "Color2Gray" [2]. In this method, a signed color distance is defined, and is used for the formulation of an optimization problem concerning the color removal, that is, the monochrome conversion. And the color-to-monochrome image
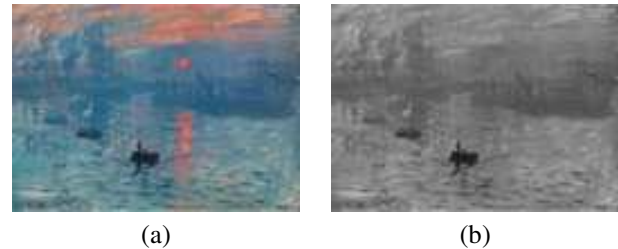

(a)　　　　　　　　　　(b)

Fig. 1.　"Impression" by C. Monet. (a) Original image, (b) Luminance components in HDTV standard.

conversion is achieved by solving the optimization problem. The color difference introduced in Color2Gray is a splendid concept, and resulting images preserve impression of original color images well. However, an algorithm of Color2Gray is complex and its computational cost is tremendous. Hence, it is not suitable for a practical use.

In this paper, firstly, an analytical solution of the optimization problem formulated in Color2Gray is introduced, and a new monochrome conversion method in which calculation cost is reduced comparing with the original Color2Gray is proposed by deforming the analytical solution. The effectiveness of the proposed method is verified by some conversion experiments.

II. Color to Monochrome Conversion Based on Signed Color Distance

A.A. Gooch et al. have proposed the color to monochrome image conversion method called "Color2Gray" which obtains the converted monochrome image by minimizing an optimization problem formulated based on the signed color distance [2]. In Color2Gray, the signed color distance is reflected to gray-levels constructing the monochrome image through the minimization of the optimization problem. Details of Color2Gray are described here.

A. Optimization Problem

In Color2Gray, the following objective function is formulated, at first:

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

where f_i is a gray-level of ith pixel, and $\boldsymbol{f}$ stands for gray-levels of a whole image. δ_{ij} is a signed color distance between ith and jth pixels, and it means the difference of the colors.

σ_ρ is a set of pixel pairs, and its elements are pixel pairs (i, j) satisfying that a chessboard distance between ith and $j$the pixels is equal to ρ or less. That is, pixels pairs (i, j) satisfying $\max(|x_i - x_j|, |y_i - y_j|) \le \rho$ are elements of the set σ_ρ when a spatial coordinate of ith pixel is represented by (x_i, y_i). ρ indicates a range of the neighborhood. In case where all pixel pairs are elements of the set, σ_ρ is expressed as σ_∞ here.

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

$$\tilde{\boldsymbol{f}} = \arg\min_{f_i \in \mathbb{R}} E(\boldsymbol{f}). \tag{3}$$

Equation (3) is solved by using a conjugate gradient method [3] in which luminance components $\boldsymbol{l} = (L_1^*, L_2^*, \ldots, L_n^*)^T$ of an input color image are used as an initial value of $\boldsymbol{f}$. T stands for a transposition.

B. Signed Color Distance

Signed color distance δ_{ij} between ith and jth pixels is defined by:

$$\delta_{ij} = \begin{cases} \Delta L_{ij}^* & |\Delta L_{ij}^*| > \Phi_\alpha(\|\Delta \boldsymbol{C}_{ij}\|) \\ \mathrm{sign}(\Delta \boldsymbol{C}_{ij} \cdot \boldsymbol{v}_\theta)\, \Phi_\alpha(\|\Delta \boldsymbol{C}_{ij}\|) & \text{otherwise} \end{cases} \tag{4}$$

with

$$\Phi_\alpha(x) = \alpha \tanh(x/\alpha), \tag{5}$$
$$\boldsymbol{v}_\theta = (\cos\theta, \sin\theta), \tag{6}$$

and

$$\mathrm{sign}(x) = \begin{cases} +1 & x > 0 \\ -1 & \text{otherwise,} \end{cases} \tag{7}$$

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^*, b_i^*) are components of CIE 1976 $L^*a^*b^*$ color space [4]. As shown in Eq.(4), the signed color distance δ_{ij} is given as ΔL_{ij}^* when the absolute luminance 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 how attachs importance to chrominance difference in the color removal. θ is a parameter to decide a sign of the color distance in the conversion. Concretely, when θ is $\pi/4$, warm-colored and cold-colored pixels become bright and dark in the conversion, respectively.

C. Analytical Solution of the Optimization Problem

In Color2Gray, a range of the neighborhood ρ is usually set to ∞ [2] because distances of pixel pairs, which cannot be discriminated in the monochrome image conversion, are unknown beforehand. Moreover, the condition $\rho = \infty$ guarantees that a certain color in an input image is always converted into a gray-level.

In case where the number of pixels of an input image is n, the number of elements of σ_∞ becomes $_nC_2 = n(n-1)/2$, and setting ρ to ∞ seems to be not appropriate from a view point of the computational complexity. However, the optimization problem of Color2Gray can be solved analytically when ρ is set to ∞ [5]. An analytical solution given in Ref.[5] is represented by:

$$\tilde{f}_i = \bar{L}^* + \frac{1}{n} \sum_{j=1}^{n} \delta_{ij}, \tag{8}$$

where $\bar{L}^*$ stands for an average luminace of an input image. And, n means the number of pixels.

III. FAST COLOR REMOVAL ALGORITHM

When the monochrome image conversion is achieved by using an analytical solution of Color2Gray, calculations for the second term of Eq.(8) occupy most of a whole calculation cost. In this paper, we propose the fast Color2Gray algorithm by using the analytical solution in Eq.(8).

A. Calculation Using Color Number

Here, it is assumed that the color number i' is given for colors appeared in an input image, and number of pixels whose color number is i' is also given as $s_{i'}$. By using Eq.(8), the output gray-level $\tilde{f}_{i'}$ for i'th color can be given by:

$$\tilde{f}_{i'} = \bar{L}^* + \frac{1}{n} \sum_{j'=1}^{m} s_{j'} \delta_{i'j'}, \tag{9}$$

where m means number of colors included in an input image. The output gray-level $\tilde{f}_i$ for ith pixel is obtained by using a look-up table T_1 of i's and i''s.

m is always less or equal to n, and in most cases, m is sufficiently small against n. Therefore, the calculation cost of Eq.(9) is quite less than that of Eq.(8). However, the calculation cost of making T_1 and counting $s_{i'}$ must be small for high speed processing. The way to calculate the T_1 and $s_{i'}$ is described below.

At first, the label l_i for ith pixel is given by:

$$l_i = 256^2 r_i + 256^1 g_i + 256^0 b_i. \tag{10}$$

Here, it is assumed that the RGB values recorded in 256 levels. In case where these are recorded in x levels, 256 in Eq.(10) should be change into x.

Then, pixels are to be sorted in l-order. The fast sort algorithm, quick sort [6], is used here. For example, sorted pixels in l-order become like as $\{l_{34}, l_{52}, l_3, l_4, l_{127}, l_{85}, l_{63}, \ldots\}$. And the values of l become like as $\{0, 0, 1, 1, 1, 3, 7, \ldots\}$. 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 do not exist in an input image, such as $(r, g, b) = (0, 0, 2)$ in this example, are not assigned the color number.

A color number i' can be assigned by scanning the sorted pixels. On that occasion, $s_{i'}$ and the relationship between i and i', that is, the look-up table T_1 can be also acquired simultaneously.

B. Quantization of colors

Now, we propose the calculation cost reduction by quantization of colors.

In this paper, the RGB values after a quantization are given by:

$$r'_i = \beta \lfloor r_i/\beta \rfloor + \beta/2, \qquad (11)$$

$$g'_i = \beta \lfloor g_i/\beta \rfloor + \beta/2, \qquad (12)$$

and

$$b'_i = \beta \lfloor b_i/\beta \rfloor + \beta/2, \qquad (13)$$

where β stands for the width of quantization and it is positive integer number. $\lfloor x \rfloor$ is a floor function and means a maximum integer number less or equal to x.

Though it depends on color distribution of an input image, the number of colors after quantization is usually significantly smaller than that of the bare input image.

C. Changing the Signed Color Distance

A new signed color distance δ'_{ij} is defined by changing Eq.(5), which is one of definition of δ_{ij}, as follows:

$$\Phi'_\alpha(x) = \alpha \tanh(\lfloor x \rfloor/\alpha). \qquad (14)$$

In practical, a look up table T_2 of $\lfloor x \rfloor$'s and $\Phi'_\alpha(x)$'s is used. The cost of calculating "tanh" is reduced by using T_2 while the quality of resulting image is hardly deteriorated.

D. Proposed algorithm

Finally, we propose a fast color removal method named "Fast Color2Gray." This algorithm is constructed by using all methods explained in Sects.III-A–III-C. In Fast Color2Gray, the output gray-level $\tilde{f}_{i'}$ for i'th color is given by:

$$\tilde{f}_{i'} = \bar{L}^* + \frac{1}{n} \sum_{j'=1}^{m'} s_{j'} \delta'_{i'j'}, \qquad (15)$$

where m' stands for the number of colors after quantization. Colors in an input image are quantized before δ' is calculated. The output gray-level $\tilde{f}_i$ for ith pixel is obtained by using T_1.

IV. EXPERIMENTAL RESULTS

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

In the experiment, images "Impression" and "Big Ben" by C. Monet, "Map" in Yahoo!/NAVITEQ, and "Voiture" by L. Bli, in which each image is 24 bits/color-scale, are employed. The sizes of them are illustrated in Table I. Figure 2 shows three images except "Impression" and those monochrome conversion results in HDTV standard. As you can see from Fig.2, a contour of a building in "Big Ben," an island in "Map," and a boundary of a grassy plain and a mountain in "Voiture" are hard to see in those monochrome images consisting of luminance components.

Color2Gray algorithm has three parameters, ρ, α, and θ. As mentioned in Sect.II-C, $\rho = \infty$ is the condition to acquire

TABLE I
SIZES OF IMAGES EMPLOYED IN THE EXPERIMENTS. A UNIT IS PIXEL.

Image	Size of image [pixels]
Big Ben	128×113
Impression	128×92
Map	172×220
Voiture	125×125

(a)   (b)

(c)   (d)

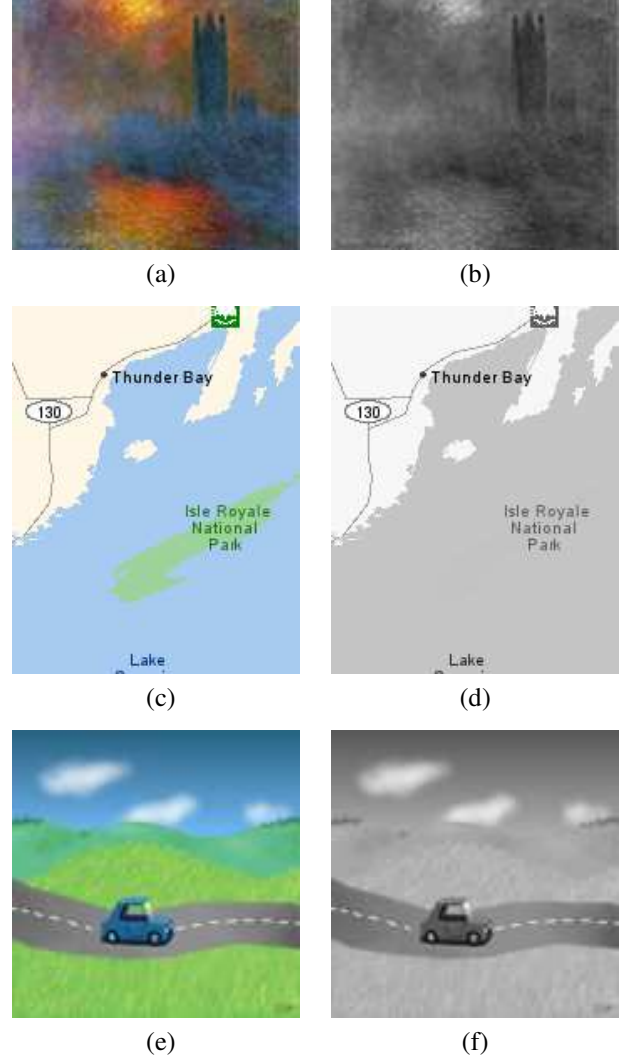
(e)   (f)

Fig. 2. Images employed in the experiments and those luminance components in HDTV standard. (a) "Big Ben," (b) Luminance components of "Big Ben," (c) "Map," (d) Luminance components of "Map," (e) "Voiture," (f) Luminance components of "Voiture."

a good resulting image. In this paper, ρ is set to ∞, and a method represented by Eq.(8) is called "Original Color2Gray." The rest parameters α and θ are irrelevant to the calculation cost. Therefore, the values producing good results in many images were employed, and concretely, these were $\alpha = 15$ and $\theta = \pi/4$.

Firstly, the number of colors after quantization are shown in Table II. And, calculation times of Fast Color2Gray are shown in Table III. The CPU employed in the experiments

TABLE II
RELATIONSHIPS BETWEEN β AND NUMBER OF COLORS AFTER QUANTIZATION.

		Big Ben	Impression	Map	Voiture
β	1	13690	10752	132	4988
	2	11272	7291	132	3126
	4	6073	2784	132	1694
	8	1924	758	127	749
	16	444	198	97	290

TABLE III
CALCULATION TIMES. A UNIT IS SECOND.

		Big Ben	Impression	Map	Voiture
Original Color2Gray		15.3	10.2	63.7	17.8
β	1	2.71	1.73	0.01	0.47
	2	1.88	0.89	0.01	0.20
	4	0.62	0.16	0.01	0.07
	8	0.08	0.01	0.01	0.02
	16	0.01	0.01	0.01	0.01

TABLE IV
MSE'S BETWEEN RESULTING IAMGES OBTAINED BY FAST COLOR2GRAY AND THOSE OBTAINED BY ORIGINAL COLOR2GRAY.

		Big Ben	Impression	Map	Voiture
β	1	0.49	0.53	0.67	0.48
	2	0.58	0.62	0.73	0.61
	4	1.1	1.0	1.3	1.2
	8	3.2	2.7	6.0	4.1
	16	10.8	9.6	1.9	14.8

(a)   (b)

(c)   (d)

Fig. 3.   Resulting images obtained by Fast Color2Gray. (a) "Big Ben," (b) "Impression," (c) "Map," (d) "Voiture."

was Intel®Core™2 Duo 3.0GHz. It can be easily understood that the color numbers and the calculation times are reduced as the value of β increases.

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

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (I_i - J_i)^2, \tag{16}$$

where I and J stand for the images obtained by Fast Color2Gray and by Original Color2Gray, respectively. Table IV shows MSE's of between the images obtained by Fast Color2Gray and Original Color2Gray.

From Tables III and IV, we decided that $\beta \approx 4$ is appropriate for both the computational time and the image quality.

Finally, the resulting images obtained by Fast Color2Gray, in which β was set to 4, are shown in Fig.3. From Fig.3, it is observed that a contour of a building in "Big Ben," the sun and its reflection in the water in "Impression," an island in "Map," and a boundary of a grassy plain and a mountain in "Voiture" are easy to see in those monochrome images obtained by Fast Color2Gray. All images obtained by Fast Color2Gray are visually good, and its computational time is significantly smaller than that of Original Fast Color2Gray.
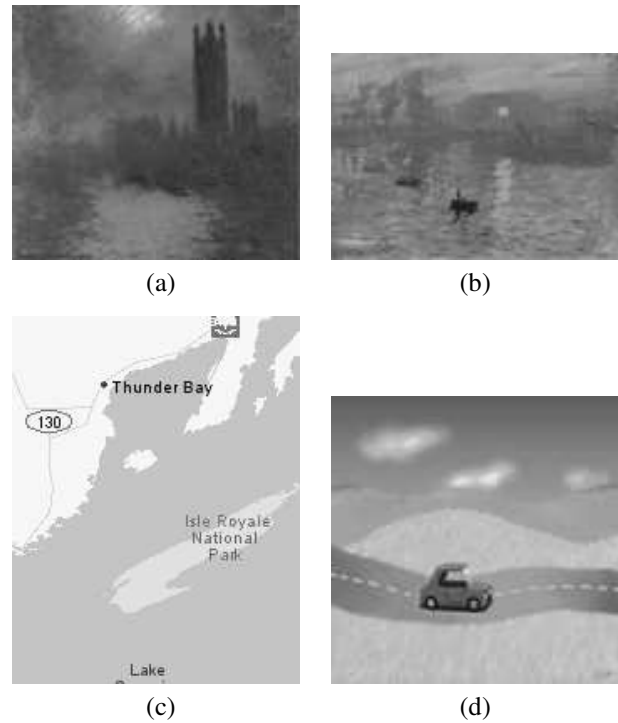
Therefore, it can be said that the effectiveness of the proposed algorithm is confirmed.

V. CONCLUSIONS

In this paper, firstly, the salience-preserving color removal method called "Color2Gray" proposed by A.A. Gooch et al. was introduced. Then, fast Color2Gray algorithm, which reduces the processing time significantly, was proposed by using the analytical solution of the optimization problem of Color2Gray. Some experiments proved the effectiveness of the proposed algorithm.

REFERENCES

[1] A.K. Jain, Fundamentals of digital image processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[2] A.A. Gooch, S.C. Olsen, J. Tumblin, and B. Gooch, "Color2Gray: salience-preserving color removal," Proc. SIGGRAPH 2005, pp.634–639, 2005.
[3] J.R. Shewchuk, An introduction to the conjugate gradient method without the agonizing pain, Tech. Rep., Pittsburgh, PA, 1994.
[4] F. Viénot, H. Brettel, and J.D. Mollon, "Digital video colourmaps for chechking the legibility of displays by dichromats," COLOR research and application, vol.24, no.4, pp.243–252, 1999.
[5] G. Tanaka, N. Suetake, and E. Uchino, "A study of color removal method based on signed color distance and its calculation cost reduction," IEICE Trans. Fundamentals, vol.J91-A, no.8, pp.817–822, Aug. 2008.
[6] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, Numerical Recipes in C, 2nd ed., Cambridge University Press, 1992.