We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,900
Open access books available

124,000
International authors and editors

140M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Pixel Value Adjustment for Digital Watermarking Using Uniform Color Space

Motoi Iwata, Takao Ikemoto, Akira Shiozaki and Akio Ogihara
Osaka Prefecture University
Japan

1. Introduction

In this chapter, we propose a pixel adjustment procedure for digital watermarking methods using uniform color space. Many digital watermarking methods use uniform color space due to the convenience of controlling image quality (Yoshiura & Echizen, 2006). However, there is the problem that watermarked RGB integer pixel values in such methods are not quite corresponding to the watermarked real-number pixel values in uniform color space because the color transform between RGB space and uniform color space is non-linear, where we assume that the watermarked RGB integer pixel values are obtained by rounding off the RGB real pixel values transformed from the real pixel values in uniform color space. Note that RGB space means sRGB color space in this chapter. In sRGB color space, each color component is stored as an integer value and the range of pixel values is [0, 255]. This problem often causes erroneous extraction.

We have already found that the simple rounding off procedure used in many existing watermarking methods is not enough to control robustness. Then we didn't use the simple rounding off procedure but used RGB neighboring search procedure for calculating watermarked RGB integer pixel values in our watermarking method for uniform color space (Iwata et al., 2010). In RGB neighboring search, firstly we obtain watermarked RGB real-number pixel values from the watermarked real pixel values in uniform color space. Next we obtain eight ($= 2^3$) candidates by rounding up or rounding down each pixel value in R, G and B components. Finally we select suitable one for extracting correct watermark bits. We confirmed that we could improve the robustness by using RGB neighboring search procedure. In fact, RGB neighboring search procedure is the most suitable for watermarking methods using linear color transformations such as YCbCr.

In case of watermarking methods using uniform color space, the most suitable pixel value for extracting correct watermark bits is not always in the eight candidates in RGB neighboring search because color transformation between RGB space and uniform color space is non-linear. Therefore we propose a new pixel value adjustment procedure based on a color transformation table. The color transformation table represents the relationship between RGB integer pixel values and the corresponding real pixel values in uniform color space. It is the advantage by using the color transformation table that we can select the most suitable real-number pixel values in uniform color space from the candidates which correspond to RGB integer pixel values. Note that each candidate can be extracted as the same value as

the original value even after transforming it into RGB space and then transforming it into uniform color space again. Therefore we can obtain the selected real-number pixel values in uniform color space from watermarked images. We adopt two approaches to use the color transformation table. One is that the color transformation table is applied to watermarked real pixel values in uniform color space after embedding. In this approach, we search the pixel value nearest to the watermarked real pixel value. The other is that the color transformation table is applied to original pixel values in uniform color space before embedding. In this approach, we select the most suitable pixel value for embedding from the set of colors near the original pixel value, where original pixel value means the pixel value of an image which is going to be watermarked but has not yet.

In experiments, we demonstrate the advantage of the proposed pixel value adjustment based on color transformation table by comparing with the simple rounding off procedure and RGB neighboring search procedure. Moreover we demonstrate the advantage of the latter approach described above.

The rest of this chapter consists of six sections. Firstly We describe color spaces in Sec. 2. Next we introduce existing pixel value adjustments and propose a new pixel value adjustment procedure based on color transformation table in Sec. 3. Then we describe two approaches utilizing pixel value adjustments in Sec. 4. Moreover we describe watermarking procedure for evaluating pixel value adjustments in Sec. 5. Then we show and discuss the performance of the proposed pixel value adjustment and the proposed approach in Sec. 6. Finally we conclude this chapter in Sec. 7.

2. Color spaces

In this section, we describe sRGB color space, XYZ color space and L*a*b*color space in Sec. 2.1, Sec. 2.2 and Sec. 2.3 respectively.

2.1 sRGB color space

sRGB color space is a standard RGB color space used for general monitors and so on. In sRGB, one color is composed of red, green and blue components. Each component is represented by an integer value of size 8 bits. The range of each component is [0, 255]. Then one color is represented by 24 bits.

2.2 XYZ color space

XYZ color space is a color space established by CIE (Commission Internationale de l'Eclairage) in 1931. The transformation of sRGB color space into XYZ color space is as follows (JSA, 2007):

2.2.1 sRGB to XYZ

First we obtain gamma-transformed sRGB color space by the following equations.

$$R_\gamma = \begin{cases} \dfrac{\hat{R}}{12.92}, & \hat{R} \le 0.04045 \\ \left(\dfrac{\hat{R} + 0.055}{1.055}\right)^{2.4}, & \text{otherwise} \end{cases} \tag{1}$$

$$G_\gamma = \begin{cases} \dfrac{\hat{G}}{12.92}, & \hat{G} \leq 0.04045 \\ \left(\dfrac{\hat{G} + 0.055}{1.055}\right)^{2.4}, & \text{otherwise} \end{cases} \tag{2}$$

$$B_\gamma = \begin{cases} \dfrac{\hat{B}}{12.92}, & \hat{B} \leq 0.04045 \\ \left(\dfrac{\hat{B} + 0.055}{1.055}\right)^{2.4}, & \text{otherwise} \end{cases} \tag{3}$$

$$\hat{R} = \frac{R}{255} \tag{4}$$

$$\hat{G} = \frac{G}{255} \tag{5}$$

$$\hat{B} = \frac{B}{255} \tag{6}$$

where R_γ, G_γ and B_γ are the values in gamma-transformed sRGB color space, and R, G and B are the values in sRGB color space.

Then we obtain XYZ color space from gamma-transformed sRGB color space by the following equations.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412453 & 0.35758 & 0.180423 \\ 0.212671 & 0.71516 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} \begin{pmatrix} R_\gamma \\ G_\gamma \\ B_\gamma \end{pmatrix} \tag{7}$$

where X, Y, Z represents the value of X component, Y component and Z component respectively.

2.2.2 XYZ to sRGB as real-number

The pixel value (X, Y, Z) in XYZ color space can be transformed into the corresponding pixel value (R, G, B) in sRGB color space by the following equations. Note that R, G and B in Eqs.(9) $\sim$ (11) are not integers but real-numbers.

$$\begin{pmatrix} R_\gamma \\ G_\gamma \\ B_\gamma \end{pmatrix} = \begin{pmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.415 \\ 0.0557 & -0.2040 & 1.0570 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{8}$$

$$R = \begin{cases} 255 \times 1.055 R_\gamma^{\frac{1}{2.4}} - 0.055, & R_\gamma > 0.0031308 \\ 255 \times 12.92 R_\gamma, & \text{otherwise} \end{cases} \tag{9}$$

$$G = \begin{cases} 255 \times 1.055 G_\gamma^{\frac{1}{2.4}} - 0.055, & G_\gamma > 0.0031308 \\ 255 \times 12.92 G_\gamma, & \text{otherwise} \end{cases} \tag{10}$$

$$B = \begin{cases} 255 \times 1.055 B_\gamma^{\frac{1}{2.4}} - 0.055, & B_\gamma > 0.0031308 \\ 255 \times 12.92 B_\gamma, & \text{otherwise} \end{cases} \tag{11}$$

2.3 L*a*b* color space

L*a*b*color space is one of uniform color spaces established by CIE in 1976 (*Lab color space*, n.d.). In a uniform color space, the distances between colors are fixed based on the perceptual differences between the colors (JSA, 2007; Oyama, 2000).

2.3.1 XYZ to L*a*b*

The pixel value (X, Y, Z) in XYZ color space can be transformed into the corresponding pixel value (L^*, a^*, b^*) in L*a*b*color space by the following equations.

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) - 16 \tag{12}$$

$$a^* = 500 \left\{ f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right\} \tag{13}$$

$$b^* = 200 \left\{ f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right\} \tag{14}$$

where

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & \text{if } \quad t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{16}{116}, & \text{otherwise} \end{cases} \tag{15}$$

where X_n, Y_n and Z_n are coefficients which depend upon the illuminant (for daylight illuminant D65, $X_n = 95.045$, $Y_n = 100$ and $Z_n = 108.892$).

2.3.2 L*a*b*to XYZ

The reverse transformation is as follows:

$$Y = Y_n f^{-1}\left(\frac{1}{116}(L^* + 16)\right) \tag{16}$$

$$X = X_n f^{-1}\left(\frac{1}{116}(L^* + 16) + \frac{1}{500}a^*\right) \tag{17}$$

$$Z = Z_n f^{-1}\left(\frac{1}{116}(L^* + 16) - \frac{1}{200}b^*\right) \tag{18}$$

where

$$f(t) = \begin{cases} t^3, & \text{if} \quad t > \dfrac{6}{29} \\ 3\left(\dfrac{6}{29}\right)^2\left(t - \dfrac{4}{29}\right), & \text{otherwise} \end{cases} \tag{19}$$

3. Pixel value adjustment

In this section, we use L*a*b*color space and RGB color space as pre-transform color space and transformed color space respectively. Let (L^*, a^*, b^*) be a pixel value in L*a*b*color space and let (R, G, B) be the corresponding pixel value in RGB color space, where the pixel values of pre-transformed color space are real numbers and those of transformed color space are integers. Pixel value adjustments transform (L^*, a^*, b^*) to (R, G, B) in each manner described in the following sections.

3.1 Simple adjustment

Firstly (L^*, a^*, b^*) is transformed to (R', G', B') by the equations described in Sec. 2, where R', G' and B' are real numbers calculated by the equations. Then R, G and B are obtained by rounding R', G' and B' off to integers respectively.

3.2 Adjustment by RGB neighboring search

Firstly (L^*, a^*, b^*) is transformed to (R', G', B') by the equations described in Sec. 2 as same as simple adjustment. Next the candidates $\xi_i (0 \le i < 8)$ are calculated as the following:

$$\xi_1 = (\lfloor R' \rfloor, \lfloor G' \rfloor, \lfloor B' \rfloor) \tag{20}$$

$$\xi_2 = (\lfloor R' \rfloor, \lfloor G' \rfloor, \lceil B' \rceil) \tag{21}$$

$$\xi_3 = (\lfloor R' \rfloor, \lceil G' \rceil, \lfloor B' \rfloor) \tag{22}$$

$$\xi_4 = (\lfloor R' \rfloor, \lceil G' \rceil, \lceil B' \rceil) \tag{23}$$

$$\xi_5 = (\lceil R' \rceil, \lfloor G' \rfloor, \lfloor B' \rfloor) \tag{24}$$

$$\xi_6 = (\lceil R' \rceil, \lfloor G' \rfloor, \lceil B' \rceil) \tag{25}$$

$$\xi_7 = (\lceil R' \rceil, \lceil G' \rceil, \lfloor B' \rfloor) \tag{26}$$

$$\xi_8 = (\lceil R' \rceil, \lceil G' \rceil, \lceil B' \rceil) \tag{27}$$

where $\lfloor R' \rfloor$ represents the maximum integer which is less than R', and $\lceil R' \rceil$ represents the minimum integer which is larger than R'. Finally one of the above eight candidates is selected as (R, G, B). The general criterion of this selection is the Euclidean distance between (L^*, a^*, b^*) and (L_i^*, a_i^*, b_i^*), where L_i^*, a_i^* and b_i^* are the corresponding pixel value in L*a*b*color space to ξ_i.

3.3 Adjustment based on color transformation table

3.3.1 Color transformation table

Color transformation table represents the relationship between RGB integer pixel values and the corresponding real pixel values in uniform color space. In this chapter, we use color transformation table to transform real-number pixel values in uniform color space into the corresponding integer pixel values in RGB color space. The output of color transformation table is the set of real-number pixel values in uniform color space which are near to the input pixel value. Note that all pixels in the set can be transformed without loss into RGB color space because they have the integer pixel values corresponding to themselves in RGB color space.

3.3.2 Adjustment procedure

Firstly (L^*, a^*, b^*) is transformed to $Z = \{\zeta_i\}$, where ζ_i represents a color in uniform color space near to the input (L^*, a^*, b^*). In the set Z, the nearest color to the input is ζ_0, followed in order by $\zeta_1, \zeta_2, \ldots$. Then one of the colors in the set is selected, and is transformed without loss into the corresponding (R, G, B).

3.3.3 Approximate search

Exhaustive search to obtain Z requires extraordinary computational cost. Thus we use the approximate search to obtain Z as follows:

step1 The input (L^*, a^*, b^*) is given. Z is initialized to $Z = \varnothing$. The parameters δ and n are fixed, where δ represents the range of approximate search around the input, and T_Z represents the minimum number of $|Z|$.

step2 Add the colors ζ satisfying $|L^* - L^*(\zeta)| \leq \delta$ to Z, where $L^*(\zeta)$ represents the L^* component at ζ.

step3 Remove the colors ζ satisfying $|a^* - a^*(\zeta)| > \delta$ from Z, where $a^*(\zeta)$ represents the a^* component at ζ.

step4 Remove the colors ζ satisfying $|b^* - b^*(\zeta)| > \delta$ from Z, where $b^*(\zeta)$ represents the b^* component at ζ.

step5 If $|Z| \geq T_Z$, go to step6. Otherwise, increase $\delta = \delta + d\delta$, initialize $Z = \varnothing$, and go to step2, where $d\delta$ is a pre-determined constant value.

step6 Sort ζ_i in Z so that the Euclidean distance between the input (L^*, a^*, b^*) and ζ_0 is the shortest, followed in order by $\zeta_1, \zeta_2, \ldots$.

Finally we obtain the set Z of colors which are near to the input (L^*, a^*, b^*). All colors ζ_i in Z calculated by this approximate search satisfy $|L^* - L^*(\zeta_i)| \leq \delta$, $|a^* - a^*(\zeta_i)| \leq \delta$, $|b^* - b^*(\zeta_i)| \leq \delta$ and $|Z| \geq T_Z$.

4. Approaches to use pixel value adjustment procedures

We adopt two approaches to use the color transformation table. One is that the color transformation table is applied to watermarked real-number pixel values in uniform color
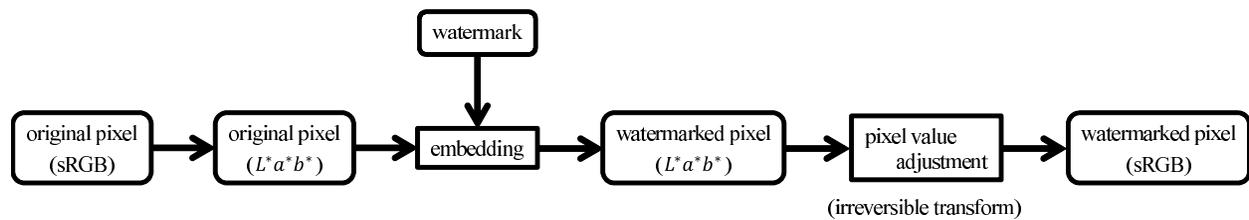
Fig. 1. Former approach : Adjustment after embedding

space after embedding. In this approach, we search the pixel value nearest to the watermarked real-number pixel value. Generally, simple adjustment and adjustment by RGB neighboring search are also adopt this approach. The other is that the color transformation table is applied to original pixel values in uniform color space before embedding. In this approach, we select the most suitable pixel value for embedding from the set of colors near the original pixel value, where original pixel value means the pixel value of an image which is going to be watermarked but has not yet. The former approach is existing one, and the latter approach is new one we proposed.

4.1 Adjustment after embedding

Figure 1 shows the flow of the former approach described here. This approach is generally used for many existing methods using a uniform color space.

step1 Original pixels in sRGB color space are transformed into the corresponding original pixels in $L^*a^*b^*$ color space by using the equations in Sec. 2.

step2 Watermarks are embedded into the original pixels in $L^*a^*b^*$ color space. Then the watermarked pixels in $L^*a^*b^*$ color space are obtained.

step3 The watermarked pixels in $L^*a^*b^*$ color space are transformed into the real-number watermarked pixels in sRGB color space by using the equations in Sec. 2.

step4 The real-number watermarked pixels in sRGB color space are adjusted to integers by using the adjustment procedure described in Sec. 3.

Note that the pixel value adjustment used in step4 is irreversible because the real-number watermarked pixels in sRGB color space have difference of maximum size 0.5 from integers. This fact sometimes causes erroneous extraction. The origin of this problem is that the watermarked pixels in $L^*a^*b^*$ color space are calculated without consideration of the existence of the corresponding integer pixels in sRGB color space. Moreover the influence of adjustments from real-number sRGB pixels to integer sRGB pixels on the corresponding pixel values in $L^*a^*b^*$ is not constant because the transformation between $L^*a^*b^*$ and sRGB is non-linear.

4.2 Fusion of adjustment and embedding

Figure 2 shows the flow of the latter approach described here. It is a new approach we describe in this chapter.

step1 Original pixels in sRGB color space are transformed into the corresponding original pixels in $L^*a^*b^*$ color space by using the equations in Sec. 2.
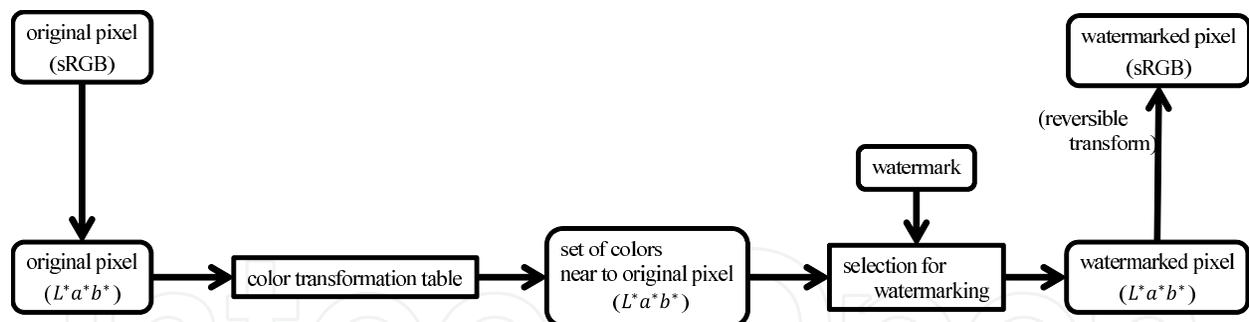
Fig. 2. Latter approach : Fusion of adjustment and embedding

step2 The original pixels in L*a*b*color space are transformed into the sets of colors near to themselves by using color transformation table in the way described in Sec. 3.3

step3 Each watermarked pixel in L*a*b*color space is selected from the set of colors near to the corresponding original pixel in L*a*b*color space so that the watermarked pixel satisfies the watermarking condition according to the watermark.

step4 The watermarked pixels obtained in step3 are transformed without loss into the watermarked pixels in sRGB color space.

Note that the transformation in step4 is reversible because the watermarked pixel selected from the set in step3 has the corresponding pixel in integer sRGB color space. This fact causes correct extraction because extractors can obtain the very same watermarked pixel values as embedders produce in L*a*b*color space although the pixels have been transformed into integer sRGB color space.

5. Watermarking procedure for evaluation

We use the watermarking method proposed by Chou and Wu (Chou & Wu, 2003) to evaluate each pixel value adjustment procedure. In this section, we briefly describe the method. The method proposed by Chou and Wu embeds 3 bits into a pixel in L*a*b*color space.

5.1 Embedding procedure

We describe how to embed the i-th watermark vector $\boldsymbol{W_i} = (w_i^L, w_i^a, w_i^b)$ into the i-th pixel value $\boldsymbol{P_i} = (L_i, a_i, b_i)$ in the following, where $w_i^L \in \{0,1\}$ and $w_i^a \in \{0,1\}$ and $w_i^b \in \{0,1\}$ and L_i, a_i and b_i represent the pixel values of L^*, a^* and b^* components respectively.

step1 The quantization index $\boldsymbol{q_i}$ is calculated from $\boldsymbol{P_i}$ by the following equation.

$$\boldsymbol{q_i} = \left(\frac{L_i}{Q_L}, \frac{a_i}{Q_a}, \frac{b_i}{Q_b} \right) \tag{28}$$

where Q_L, Q_a and Q_b represent the quantizer stepsizes of L^*, a^* and b^* components respectively. Note that this quantization adopts rounding off.

step2 The watermarked quantization index $\boldsymbol{q_i^W}$ is obtained by the following equation.

$$q_i^W = q_i + (\nu_i \times (\hat{q}_i \oplus W_i)) \tag{29}$$

where

$$\hat{q}_i = q_i \bmod 2 \tag{30}$$

Here $\nu_i = (\nu_i^L, \nu_i^a, \nu_i^b)$ is a random sequence, where $\nu_i^L \in \{-1, 1\}$, $\nu_i^a \in \{-1, 1\}$ and $\nu_i^b \in \{-1, 1\}$. Note that $q_i^W = q_i$ when $\hat{q}_i = W_i$.

step3 The watermarked pixel value $P_i^W = (L_i^W, a_i^W, b_i^W)$ is calculated from q_i^W by inverse quantization as follows:

$$P_i^W = (Q_L, Q_a, Q_b) \times q_i^W$$

$$= (Q_L q_i^{LW}, Q_a q_i^{aW}, Q_b q_i^{bW}) \tag{31}$$

5.2 Extracting procedure

We describe how to extract the i-th watermark vector W_i' from the i-th watermarked pixel value P_i' as follows:

step1 The quantization index q_i' is calculated from P_i' with the same quantizer stepsizes used for embedding by the following equation.

$$q_i' = \left(\frac{L_i'}{Q_L}, \frac{a_i'}{Q_a}, \frac{b_i'}{Q_b} \right) \tag{32}$$

step2 The i-th watermark vector is extracted by the following equation.

$$W_i' = q_i' \bmod 2 \tag{33}$$

5.3 Watermarking condition for latter approach

At step3 in Sec. 4.2, the watermarking condition to select watermarked pixels from the set of colors is arbitrarily decided. Then we decide the watermarking condition based on the watermarking procedure for evaluation as the following equation.

$$q(\zeta_i) \equiv W_i \pmod 2 \tag{34}$$

where

$$q(\zeta_i) = \left(\frac{L^*(\zeta_i)}{Q_L}, \frac{a^*(\zeta_i)}{Q_a}, \frac{b^*(\zeta_i)}{Q_b} \right) \tag{35}$$

Here ζ_i represents the i-th color in the set of colors near to the original pixel in L*a*b*color space, and $L^*(\zeta_i)$, $a^*(\zeta_i)$ and $b^*(\zeta_i)$ represent the pixel values of ζ_i in L^*, a^* and b^* components respectively.

We select ζ_i which has minimum index i and satisfy Eq.(34) as the watermarked pixel.

6. Experiments

6.1 Environments

Firstly we investigated the image quality of watermarked images for each pixel value adjustment, that is, simple adjustment, adjustment by using RGB neighboring search, adjustment based on color transformation table after embedding and fusion of adjustment and embedding. In this section, adjustment by using RGB neighboring search, adjustment based on color transformation table after embedding and adjustment based on color transformation with fusion of adjustment and embedding are represented by "RGB neighboring search," "former approach" and "latter approach." Then we confirmed the number of error bits in each pixel value adjustment. Finally we investigated the computational time for former approach and latter approach.

Shown in Fig. 3 we used twelve color bitmap images "aerial," "airplane," "balloon," "couple," "earth," "girl," "lena," "mandrill," "milkdrop," "parrots," "pepper" and "sailboat" as original images. They were standard images widely used for experiments. The size of all original images was 256×256 pixels. They are mainly in standard image database (SIDBA) (*The USC-SIPI Image Database*, n.d.), and all of them are often used for experiments in technical papers.

In watermarking procedure, we used $Q_L = 1.5$, $Q_a = 2$ and $Q_b = 2$ as the quantizer stepsizes. The watermark is pseudo-random vector sequence of length 65536 ($= 256 \times 256$), where the element of the watermark is 3-dimensional vector as described in Sec. 5. Thus the length of watermark bits is 196608 ($= 65536 \times 3$) bits.

In the proposed adjustment based on color transformation table, we used $\delta = 2.25$, $T_Z = 1000$ and $d\delta = 0.25$. In ideal manner, δ gradually increases for the appropriately fixed T_Z and then the appropriate δ is fixed. However this manner requires great computational cost. Thus we fixed δ so that the computational cost is convenient.

We used PSNR for the metric of the amount of change of pixel values in RGB space. PSNR was calculated by the following equation.

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}} \tag{36}$$

$$\text{MSE} = \frac{1}{3N_x N_y} \sum_{i=1}^{3N_x N_y} (\text{img}_i - \text{oimg}_i)^2 \tag{37}$$

where img_i and oimg_i represent the pixels in one image and the other image respectively.

The CPUs of the personal computer used for this experiment are two Xeon E5520(2.27GHz). We used Java for implementation. The size of the memory of Java Virtual Machine is 4.0GB.

6.2 Image quality

Figures 4∼6 show the watermarked images by using RGB neighboring search, former approach and latter approach. We omitted the watermarked images by simple adjustment because they are similar to those by RGB neighboring search. This fact can be confirmed by
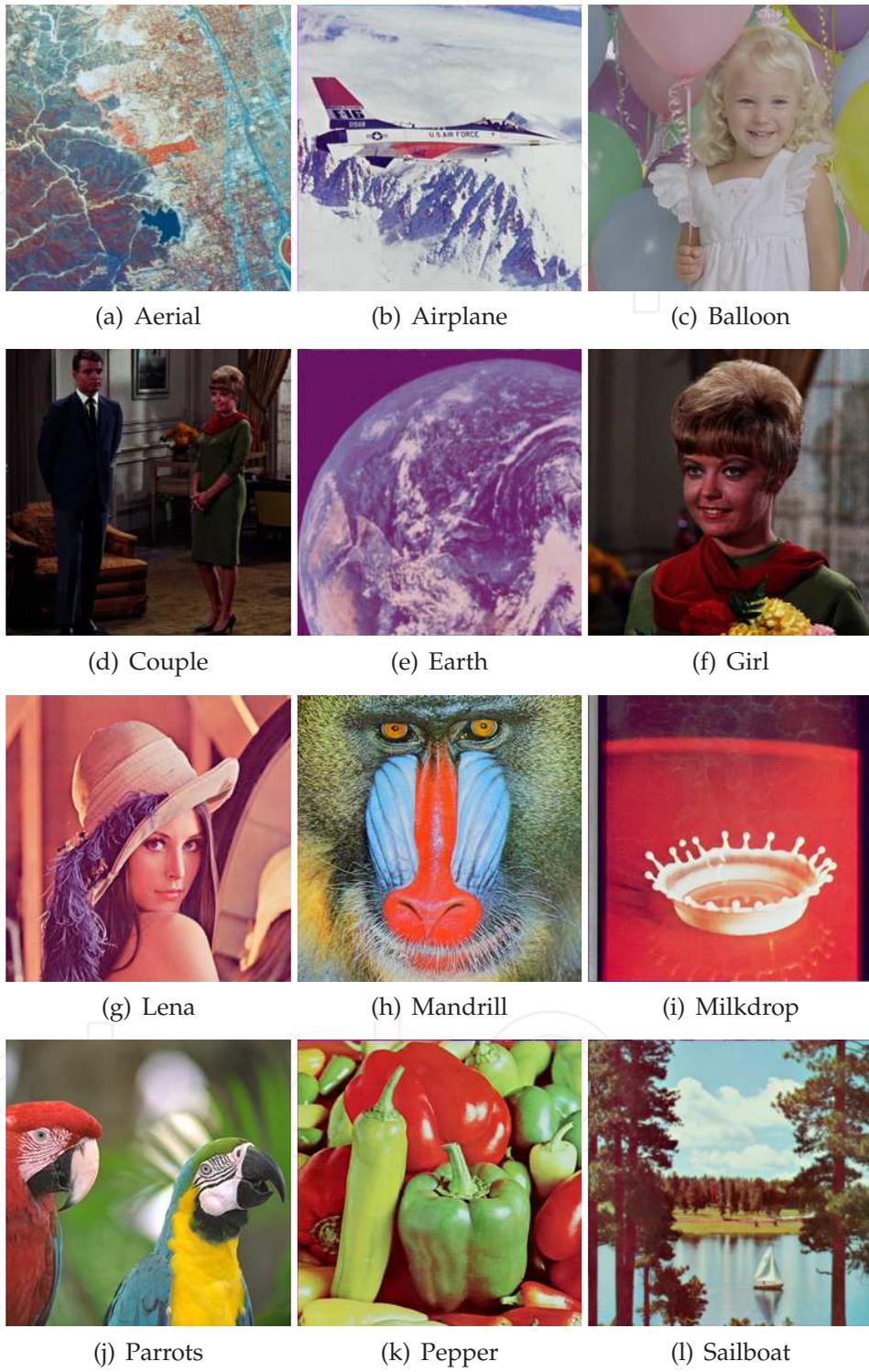
(a) Aerial (b) Airplane (c) Balloon

(d) Couple (e) Earth (f) Girl

(g) Lena (h) Mandrill (i) Milkdrop

(j) Parrots (k) Pepper (l) Sailboat

Fig. 3. Original images

the PSNRs described below. Shown in Figs. 4∼6, the degradation of the watermarked images is imperceptible.

Table 1 shows the PSNRs in each pixel value adjustment. The PSNRs of latter approach are the highest in all PSNRs.

original images	simple adjustment	RGB neighboring search	color transformation table (former approach)	color transformation table (latter approach)
Aerial	34.16	34.16	34.16	44.71
Airplane	33.95	33.88	33.84	44.52
Balloon	35.44	35.42	35.42	45.10
Couple	36.34	36.35	36.32	44.64
Earth	35.37	35.37	35.38	45.06
Girl	36.89	36.89	35.40	45.24
Lena	36.39	36.38	36.36	45.24
Mandrill	34.60	34.59	34.55	44.90
Milkdrop	35.68	35.69	35.01	44.47
Parrots	34.95	34.94	33.57	43.89
Pepper	35.21	35.20	33.58	43.93
Sailboat	34.98	34.97	34.97	45.14

Table 1. The PSNRs in each pixel value adjustment.

6.3 Number of error bits

Table 2 shows the number of error bits from the watermarked images in each pixel adjustment. The number of error bits of latter approach is the least for all original images, followed in order by that of former approach, RGB neighboring search and simple adjustment.

The number of error bits of latter approach is theoretically zero. However, there are some cases that there is no color satisfying the watermarking condition in the set of colors near to original pixels in "Aerial," "Airplane," "Parrots" and "Pepper" when $\delta = 2.25$. In such cases, δ should be larger so that there is a color satisfying the watermarking condition in the set for all original pixels, although the computational cost becomes greater.

6.4 Computational time

Table 3 shows the computational time [ms] for pixel value adjustment in former approach and latter approach when $\delta = 2.25$. The computational time for simple adjustment and RGB neighboring search is omitted because it is extremely fast compared with that for former approach and latter approach.

Compared the computational time of former approach with that of latter approach, former approach is slightly faster than latter approach. However the performance of former approach is almost same as that of simple adjustment and RGB neighboring search. Then it is desirable that the pixel value adjustment based on color transformation table is used by latter approach.

(a) Aerial : RGB neighboring search, former approach, latter approach

(b) Airplane : RGB neighboring search, former approach, latter approach

(c) Balloon : RGB neighboring search, former approach, latter approach

(d) Couple : RGB neighboring search, former approach, latter approach

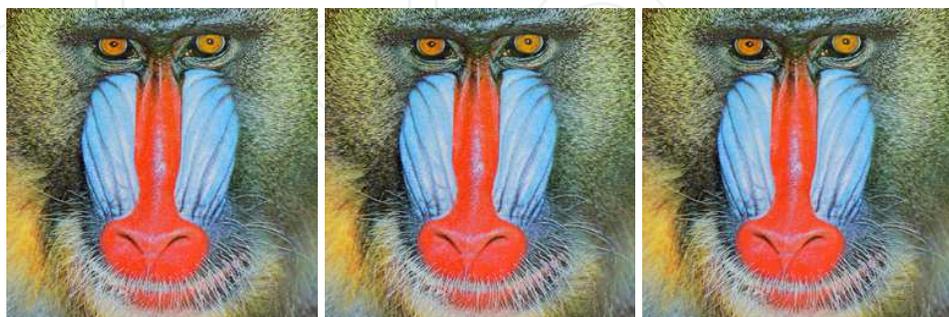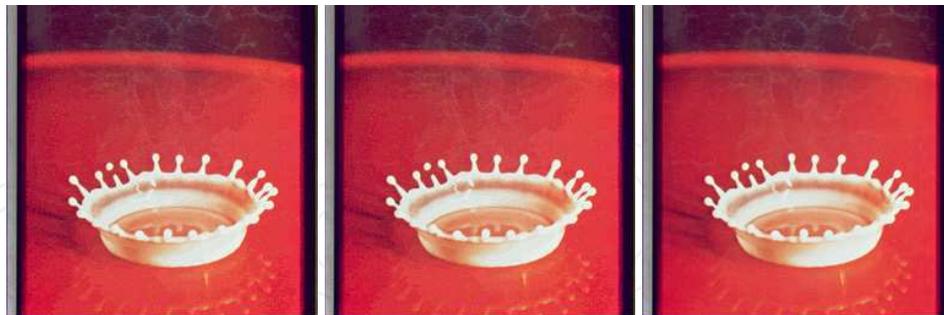Fig. 4. Experimental results(Aerial, Airplane, Balloon, Couple)

(a) Earth : RGB neighboring search, former approach, latter approach

(b) Girl : RGB neighboring search, former approach, latter approach

(c) Lena : RGB neighboring search, former approach, latter approach

(d) Mandrill : RGB neighboring search, former approach, latter approach

Fig. 5. Experimental results(Earth, Girl, Lena, Mandrill)

(a) Milkdrop : RGB neighboring search, former approach, latter approach

(b) Parrots : RGB neighboring search, former approach, latter approach

(c) Pepper : RGB neighboring search, former approach, latter approach

(d) Sailboat : RGB neighboring search, former approach, latter approach

Fig. 6. Experimental results(Milkdrop, Parrots, Pepper, Sailboat)

original images	simple adjustment	RGB neighboring search	color transformation table (former approach)	color transformation table (latter approach)
Aerial	195	196	104	3
Airplane	181	166	57	7
Balloon	0	0	0	0
Couple	4237	3839	4215	0
Earth	0	0	0	0
Girl	2516	2424	1347	0
Lena	83	79	42	0
Mandrill	119	118	36	0
Milkdrop	2163	2050	837	0
Parrots	1911	1899	1352	3
Pepper	2674	2569	1427	2
Sailboat	15	15	6	0

Table 2. The number of error bits from watermarked images in each pixel value adjustment.

original images	former approach	latter approach
Aerial	3893365	5173605
Airplane	3371501	4443929
Balloon	4403430	5759862
Couple	1224645	1534407
Earth	3536644	4889056
Girl	1947603	2319508
Lena	3648970	5725525
Mandrill	4180504	5310248
Milkdrop	3779653	5622682
Parrots	3699138	4604142
Pepper	3666946	4965765
Sailboat	2921615	3673465

Table 3. Computational time in the pixel value adjustment based on color transformation table of former approach and latter approach when $\delta = 2.25$.

6.5 Discussion

Shown in Tab. 1, the PSNRs of latter approach are the highest in all PSNRs. The PSNRs of other three pixel adjustments are similar to each other because they modify pixel values only around the corresponding watermarked pixel values (real-number). From this fact, we guess that the original pixel values in L*a*b*color space are often modified to the watermarked pixel values far from the pixel values which have the corresponding integer pixel values in sRGB color space. Moreover latter approach is more suitable than other pixel value adjustments from viewpoint of less change in L*a*b*color space because the modification of latter approach for watermarking in L*a*b*color space is the least under condition of satisfying the watermarking condition. Then Human Visual System is hard to perceive the degradation by watermarking method adopting latter approach.

Shown in Tab. 2, the number of error bits of latter approach is the least in all pixel value adjustments. In case of "Aerial," "Airplane," "Parrots" and "Pepper," the number of error bits is not zero. Even in such cases, the number of error bits must be zero by increasing δ although the computational time also increases. This result confirms the contribution of fusion of pixel value adjustment and watermarking toward the reduction of erroneous extraction.

Shown in Tab. 3, the computational time for the pixel value adjustment based on color transformation table is very long. Thus the proposed adjustment of latter approach reduces error bits and improves image quality at a computational cost. Note that the computational time for extraction is independent from the pixel value adjustment for watermarking. The computational time for extraction is usual even if the proposed adjustment of latter approach is used for watermarking.

The discussion demonstrates that the proposed adjustment of latter approach contributes to both of few error bits and high PSNRs although the long computational time for watermarking is required.

7. Conclusion

We have proposed a new pixel value adjustment based on color transformation table for watermarking methods using a uniform color space. The proposed adjustment fuses a pixel value adjustment into a watermarking procedure. This fusion produces both of few erroneous extraction from watermarked images and high PSNRs. We have investigated the image quality of watermarked images and the number of error bits from watermarked images for all pixel value adjustments. Moreover we have investigated the computational time of the pixel value adjustment based on color transformation table. We have confirmed the contribution of the fusion of a pixel value adjustment and a watermarking procedure toward the property of watermarking method using a uniform color space through the experiments and the discussion.

Our future works should be to reduce the computational cost of using color transformation table and to expand color transformation table so as to be able to apply watermarking method using frequency transform.

8. References

Chou, C. H. & Wu, T. L. (2003). Embedding color watermarks in color images, *EURASIP Journal on Applied Signal Processing* Vol.1: 32–40.

Iwata, M., Kanaya, T., Shiozaki, A. & Ogihara, A. (2010). Digital watermarking method warranting the lower limit of image quality of watermarked images, *EURASIP Journal on Advances in Signal Processing* Vol.2010: Article ID 426085, doi:10.1155/2010/426085.
    URL: *http://www.hindawi.com/journals/asp/2010/426085/*

JSA (2007). *JIS handbook 61 Color 2007*, Japanese Standards Association.

Lab color space (n.d.). Wikipedia.
    URL: *http://en.wikipedia.org/wiki/Lab_color_space*

Oyama, T. (2000). *Invitation to visual psycology*, Saiensu-sha Co., Ltd.

The USC-SIPI Image Database (n.d.). University of Southern California.
    URL: *http://sipi.usc.edu/database/database.php*

Yoshiura, H. & Echizen, I. (2006). Maintaining picture quality and improving robustness of color watermarking by using human vision models, *IEICE Trans. Inf.&Syst.* Vol.E89-D: 256–270.
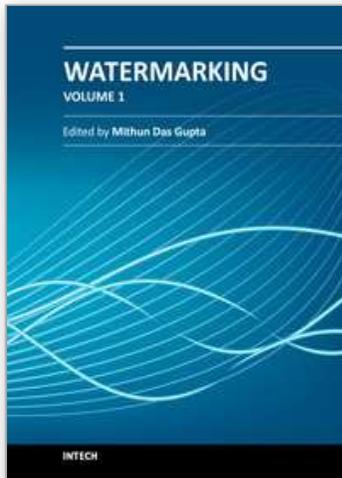
Watermarking - Volume 1
Edited by Dr. Mithun Das Gupta

This collection of books brings some of the latest developments in the field of watermarking. Researchers from varied background and expertise propose a remarkable collection of chapters to render this work an important piece of scientific research. The chapters deal with a gamut of fields where watermarking can be used to encode copyright information. The work also presents a wide array of algorithms ranging from intelligent bit replacement to more traditional methods like ICA. The current work is split into two books. Book one is more traditional in its approach dealing mostly with image watermarking applications. Book two deals with audio watermarking and describes an array of chapters on performance analysis of algorithms.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Motoi Iwata, Takao Ikemoto, Akira Shiozaki and Akio Ogihara (2012). Pixel Value Adjustment for Digital Watermarking Using Uniform Color Space, Watermarking - Volume 1, Dr. Mithun Das Gupta (Ed.), ISBN: 978-953-51-0618-0, InTech, Available from: http://www.intechopen.com/books/watermarking-volume-1/pixel-value-adjustment-for-digital-watermarking-using-uniform-color-space