

## Color Transfer between Images with IEC

Qiang LI<sup>1</sup>, Tsuyoshi NAKAMURA<sup>1\*</sup>, Yuyan CHAO<sup>2,4</sup>, Lifeng HE<sup>3,4</sup> and Hidenori ITOH<sup>1</sup>

<sup>1</sup>*Department of Computer Science and Engineering, Nagoya Institute of Technology,  
Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan*

<sup>2</sup>*Faculty of Environment, Information and Business, Nagoya Sangyo University,  
3255-5 Arai-cho, Owariasahi, Aichi 488-8711, Japan*

<sup>3</sup>*Faculty of Information Science and Technology, Aichi Prefectural University,  
1522-3 Ibaragabasama, Kumabari, Nagakute-cho, Aichi-gun, Aichi 480-1198, Japan*

<sup>4</sup>*Guest Professor of Shanxi University of Science and Technology, China*

*\*E-mail address: tnaka@ics.nitech.ac.jp*

(Received December 1, 2004; Accepted January 31, 2005)

**Keywords:** Non-photorealistic Rendering, Color Transfer, Interactive Evolutionary Computation

**Abstract.** Some research has been conducted on applying coloration of painterly art to a photographic image. Most of the proposed methods use some absolute criteria, e.g., the nearest distance in *CIE L\*a\*b\** color space, converting a color to another uniquely. However, we think that basic color transfer depends on the subjective feelings and sensitivities of users themselves. We describe an algorithm where every user can produce an original color-transformed image resulting from two inputs, which are a target photograph and a reference painting. Our algorithm is based on interactive evolutionary computation (IEC) and can produce a variety of color-transformed images semi-automatically. Some color-transformed examples similar to reference paintings are illustrated, and the searching efficiency of our method is discussed. Figures 1, 2, 3, 4, 13, 14 and 15 are originally made of color images. They can be seen in WEB version of FORMA. URL is “<http://www.scipress.org/journals/forma/index.html>.”

### 1. Introduction

In the real world, each painterly artist renders a painting in his/her own style. This style can be distinguished by elements, such as motif, color, shape deformation and texture. Graphic researchers, however, have demonstrated many techniques to produce non-photorealistic rendered images. An aim of non-photorealistic rendering (NPR), which is an alternative to photorealism, is to produce painterly images that feature expression similar to that used in actual painterly arts (GOOCH, 2001). Previously, some methods to paste the texture of a painterly art on a photograph have been suggested (HERTZMANN, 2001; WANG, 2004). Other methods for applying coloration of a painterly art to a photographic image have also been suggested (REINHARD, 2001; CHANG, 2002). Altering colors of an image is also one of the most common tasks in image processing, and it is the main interest of our work.

Fig. 1. Target photograph  $I_{in}$ .Fig. 2. Reference painting  $I_{ref}$ .

When you imagine a method for applying the colors of a given painting to a photograph, you should consider the use of color space. Previous researches used  $l\alpha\beta$  and  $CIE L^*a^*b^*$  color space (REINHARD, 2001; CHANG, 2002). Both  $l\alpha\beta$  and  $CIE L^*a^*b^*$  are a kind of 3D color space representation. In  $l\alpha\beta$  color space,  $l$ -axis represents an achromatic channel, while  $\alpha$  and  $\beta$  channels are chromatic yellow-blue and red-green opponent channels. In  $CIE L^*a^*b^*$ , each axis of  $L$ ,  $a$  and  $b$  represents similar channel as well as  $l\alpha\beta$ . One method is based on simple statistical analysis in color space (REINHARD, 2001), whereas the other is based on color categorization characteristics of human vision (CHANG, 2002). Although these methods have some differences in their approaches, they basically aim at a similar goal.

Algorithms of applying colors of a given painting evaluate some distance in the colors between a painting and a photograph in color space, and they acquire color pairs for the color transformation uniquely. The uniqueness of transformation is indeed useful and



Fig. 3. Resulting image 1.



Fig. 4. Resulting image 2.

effective in some cases. However, algorithms cannot always acquire results that satisfy a user's request, and it is difficult for him/her to estimate the results produced. We believe that the process of detection of color pairs depends not only on the distance of colors but also on other factors, such as the artistic subjectivity of the user.

In this paper we report on our attempt to transfer colors of a painting to a photograph, where the subjectivity of each user was the only criterion for color assignment. The color transfer processing exploits interactive evolutionary computation (IEC) (TOKUI, 2000; TAKAGI, 2001; KATAGAMI, 2002) and finds color correspondences between a painting and a photograph for altering the photograph's color. In our algorithm, IEC is applied, whose fitness function is provided by users. Users can repeat the operations to evaluate the output re-colored images and assign some fitness values, which are either 1 or 0. IEC actually

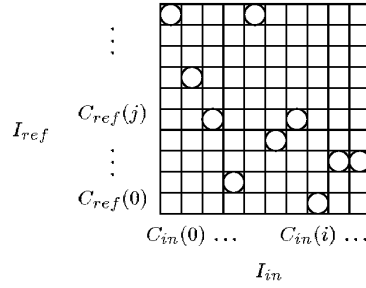


Fig. 5. Color matching table.

requires users to conduct a portion of the color-transfer operations themselves by showing users a variety of candidate images. Thus, IEC does not perform all of the processing automatically. In other words, it is semi-automatic.

Figure 3 illustrates an example of a color-transferred result by exploiting IEC method. Figures 1 and 2 are a target photograph  $I_{in}$  and a reference painting  $I_{ref}$ , respectively. IEC color-transferred Fig. 1 into Fig. 3 based on coloration of Fig. 2. Figure 4 is also another example of color-transferred result, however we had not exploited IEC to produce this image. Instead of IEC, we calculated the Euclid distance in  $CIE L^*a^*b^*$  color space and applied the nearest neighboring color of the painting (Fig. 2) to the corresponding one of the photograph (Fig. 1) in a similar way as was used previously (REINHARD, 2001; CHANG, 2002) as follows.

$$p_{out}(x, y) = p_{ref}(i, j), \quad (1)$$

where  $p_{out}(x, y)$  is a pixel value of  $(x, y)$ -coordinate on an output color-transferred result, and  $p_{ref}(i, j)$  is a pixel value of  $(i, j)$ -coordinate on a reference painting  $I_{ref}$ . A pixel value corresponds to a point in  $L^*a^*b^*$  color space.  $p_{ref}(i, j)$  is calculated as follows:

$$p_{ref}(i, j) = \arg \min_{p_{ref}(i, j)} \left( \text{dist}(p_{in}(x, y), p_{ref}(i, j)) \right), \quad (2)$$

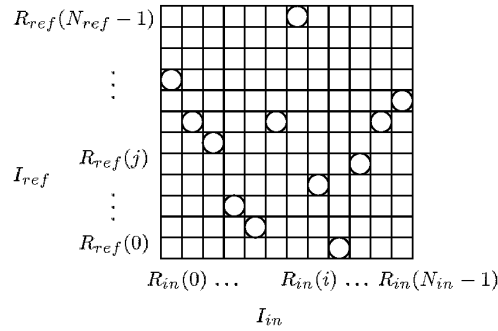
where  $p_{in}(x, y)$  is a pixel value of  $(x, y)$  on a target photograph  $I_{in}$ , and  $\text{dist}(p_{in}(x, y), p_{ref}(i, j))$  represents Euclid distance in  $CIE L^*a^*b^*$  color space as follows. The following  $L_{in}(x, y)$ ,  $a_{in}(x, y)$  and  $b_{in}(x, y)$  are  $L$ ,  $a$  and  $b$  value of  $p_{in}(x, y)$  respectively. Then  $L_{ref}(i, j)$ ,  $a_{ref}(i, j)$  and  $b_{ref}(i, j)$  are  $L$ ,  $a$  and  $b$  value on  $p_{ref}(i, j)$  respectively.

$$\text{dist}(p_{in}(x, y), p_{ref}(i, j)) = d_L + d_a + d_b, \quad (3)$$

$$d_L = (L_{in}(x, y) - L_{ref}(i, j))^2,$$

$$d_a = (a_{in}(x, y) - a_{ref}(i, j))^2, \quad (4)$$



Fig. 6. Region matching table  $T_r$ .

$$d_b = (b_{in}(x, y) - b_{ref}(i, j))^2.$$

The viewer can see and confirm the visual difference between Figs. 3 and 4. We believe that they feel much more similar coloration-look between Fig. 3 and Fig. 2 in comparison with the one between Fig. 4 and Fig. 2.

In Secs. 2 and 3, we present the problems and solutions, where we exploit IEC for color transfer. In Sec. 4, we present an overview of the algorithm, then in Sec. 5, we show some output examples produced by our algorithm and evaluate the usefulness of the IEC approach. Finally, we discuss some future directions for color transfer. Figures 1, 2, 3, 4, 13, 14 and 15 are originally made of color images. They can be seen in WEB version of FORMA. URL is “<http://www.scipress.org/journals/forma/index.html>.”

## 2. Color Transfer

Color transfer is a kind of combinatorial optimization problem. Previous works on this subject dealt with color transfer as a color-matching problem between a photograph and a painting. An example of the color-matching problem is illustrated in Fig. 5.  $C_{in}(i)$  and  $C_{ref}(j)$  represent an  $i$ -th color of a target photograph  $I_{in}$  and a  $j$ -th color of a reference painting  $I_{ref}$  respectively. The circles in the table indicate assigning  $C_{ref}(j)$  to the corresponding  $C_{in}(i)$ . The color-matching table is completed using some criteria, e.g., the nearest distance in color space which is exploited to produce Fig. 4 as we have mentioned.

In this paper we attempt to use a new type of table, called a “region-matching table” and denoted by  $T_r$ . To start, we divided each image into several regions, which indicate place areas in an image, that is to say a set of all regions is equivalent to a whole image. The regions in a target photograph  $I_{in}$  are represented as  $R_{in}(i)$  ( $i = 0, \dots, N_{in} - 1$ ) and the ones in a reference painting  $I_{ref}$  are represented as  $R_{ref}(j)$  ( $j = 0, \dots, N_{ref} - 1$ ). Then we decide a pair  $(R_{in}(i), R_{ref}(j))$ , and assigns a mean color, which is calculated in Eq. (5), to the corresponding  $R_{in}(i)$ . Figure 6 is an example of region-matching tables. The circles in the table indicate assigning a color in  $R_{ref}(j)$  to  $R_{in}(i)$ .

$$\begin{aligned}
\langle L_{\text{ref}}(j) \rangle &= \frac{\sum_{x,y \in R_{\text{ref}}(j)} L_{\text{ref}}(x,y)}{n_{\text{ref}}(j)}, \\
\langle a_{\text{ref}}(j) \rangle &= \frac{\sum_{x,y \in R_{\text{ref}}(j)} a_{\text{ref}}(x,y)}{n_{\text{ref}}(j)}, \\
\langle b_{\text{ref}}(j) \rangle &= \frac{\sum_{x,y \in R_{\text{ref}}(j)} b_{\text{ref}}(x,y)}{n_{\text{ref}}(j)},
\end{aligned} \tag{5}$$

where  $n_{\text{ref}}(j)$  is a total of pixels in  $R_{\text{ref}}(j)$ .

The region-matching table  $T_r$  provides more solutions for the color-transfer problems than the color-matching table. That is to say, even if the regions in  $I_{\text{in}}$  have same color,  $T_r$  can assign different color to each region. Instead, it needs larger searching space, and of course it will take more time to acquire the optimum solution. There is one further problem that we could not ignore. It depends on IEC's features. IEC produces color-transformed images semi-automatically. A fitness function of IEC is provided by the users themselves, and they have to repeat the operations to evaluate the color-transformed candidates until they get the optimum result that satisfies them. The repeated operations force the users to do a lot of work if the searching space is huge. Thus, we have to introduce a method to reduce the number of times of making evaluation. Our approach is based on the intuitive idea that painters roughly assign colors on a canvas first and when they draw and render a painting they decide colors for finer regions.

On the basis of this idea, we first prepared coarse regions allowing us roughly to assign colors to the regions. For acquiring coarse regions, we adopted  $k$ -means, which is a method for image segmentation (TAKAGI, 1991). After that, we assigned fine colors to each region based on a statistical analysis (REINHARD, 2001).

Once IEC finished rough color assignment, the matching pair  $(R_{\text{in}}(i), R_{\text{ref}}(j))$  was decided. Fine color assignment was performed between  $R_{\text{in}}(i)$  and  $R_{\text{ref}}(j)$ . For this purpose we wanted some aspects of the distribution of data points in  $L^*a^*b^*$  space to transfer between  $R_{\text{in}}(i)$  and  $R_{\text{ref}}(j)$ , where means and standard deviations along each of the three axes were sufficient. Thus, we computed these measures for both  $R_{\text{in}}(i)$  and  $R_{\text{ref}}(j)$ .

Precise of process is as follows. In Eq. (6) each of  $L_{\text{in}}(x, y)$ ,  $a_{\text{in}}(x, y)$  and  $b_{\text{in}}(x, y)$  represents a  $L^*a^*b^*$  color member of  $(x, y)$  in  $R_{\text{in}}(i)$ . Then each of  $\langle L_{\text{in}}(i) \rangle$ ,  $\langle a_{\text{in}}(i) \rangle$  and  $\langle b_{\text{in}}(i) \rangle$  represents a mean of  $L^*a^*b^*$  color member in  $R_{\text{in}}(i)$ . We computed the means and standard deviations for each axis separately in the  $L^*a^*b^*$  space. First, we subtracted the mean from the data at each point  $(x, y)$ :

$$\begin{aligned}
L'(x, y) &= L_{\text{in}}(x, y) - \langle L_{\text{in}}(i) \rangle, \\
a'(x, y) &= a_{\text{in}}(x, y) - \langle a_{\text{in}}(i) \rangle,
\end{aligned} \tag{6}$$

$$b'(x, y) = b_{\text{in}}(x, y) - \langle b_{\text{in}}(i) \rangle,$$

$$x, y \in R_{\text{in}}(i).$$

Then, we scaled the data points comprising the synthetic region by factors determined by the respective standard deviations:

$$L_{\text{out}}(x, y) = \frac{\sigma_{\text{ref}}^L(j)}{\sigma_{\text{in}}^L(i)} L'(x, y) + \langle L_{\text{ref}}(j) \rangle,$$

$$a_{\text{out}}(x, y) = \frac{\sigma_{\text{ref}}^a(j)}{\sigma_{\text{in}}^a(i)} a'(x, y) + \langle a_{\text{ref}}(j) \rangle, \quad (7)$$

$$b_{\text{out}}(x, y) = \frac{\sigma_{\text{ref}}^b(j)}{\sigma_{\text{in}}^b(i)} b'(x, y) + \langle b_{\text{ref}}(j) \rangle,$$

where  $L_{\text{out}}(x, y)$ ,  $a_{\text{out}}(x, y)$  and  $b_{\text{out}}(x, y)$  represent a pixel value of  $(x, y)$  on a color-transferred image  $I_{\text{out}}$ .  $\sigma_{\text{ref}}^*(j)$  and  $\sigma_{\text{in}}^*(i)$  ( $*$   $\in \{L, a, b\}$ ) are standard deviations in  $R_{\text{ref}}(j)$  and  $R_{\text{in}}(i)$  for each axis. After these transformations, the resulting data have standard deviations that conform to  $R_{\text{ref}}(j)$ .

### 3. IEC

#### 3.1. Encoding of a chromosome

IEC manages and completes a region-matching table  $T_r$  as shown in Fig. 6, where  $N_{\text{ref}}$  and  $N_{\text{in}}$  indicate the total number of  $I_{\text{ref}}$ 's regions and that of  $I_{\text{in}}$ 's. When we consider the problem for assigning  $R_{\text{ref}}(j)$  ( $j = 0, \dots, N_{\text{ref}} - 1$ ) to  $R_{\text{in}}(i)$  ( $i = 0, \dots, N_{\text{in}} - 1$ ) in the table  $T_r$ , there are  $(N_{\text{ref}})^{N_{\text{in}}}$  total combinations. Our approach uses the table itself as a chromosome<sup>†</sup> for IEC. The expression of a chromosome is as follows:

$$\left[ x_0, x_1, \dots, x_{N_{\text{in}}-1} \right],$$

$$0 \leq x_n \leq N_{\text{ref}} - 1 \quad (n = 0, \dots, N_{\text{in}} - 1). \quad (8)$$

The length of a chromosome will be  $N_{\text{in}} \log N_{\text{ref}}$  bits, which is too long when the table is large. To solve this problem, our approach divides the whole table  $T_r$  to  $K$ -small tables as

---

<sup>†</sup>“Chromosome” is usually a biological term and it indicates a string of DNA. In this paper “chromosome” is a technical term of IEC and a chromosome indicates a solution to complete  $T_r$  and it is represented as a bit string.

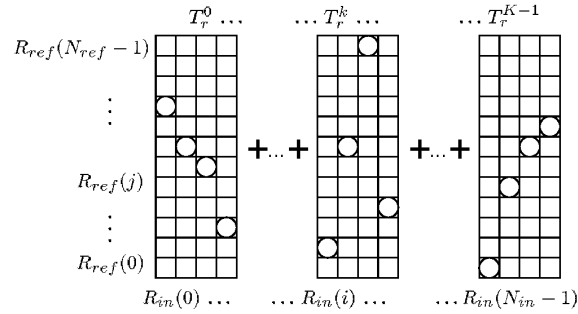


Fig. 7. Divided region-matching table.

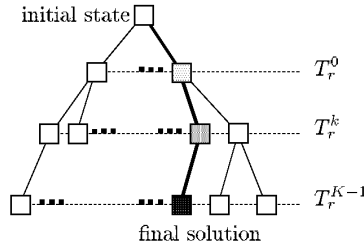


Fig. 8. Search tree to get final solution.

shown in Fig. 7, if  $N_{in}$ , the total number of  $I_{in}$ 's regions, is beyond a threshold value.  $T_r^k$  indicates a  $k$ -th small table ( $k = 0, \dots, K - 1$ ). Thus each  $T_r^k$  has  $(N_{in} \log N_{ref})/K$  bits as the length of a chromosome. Then, a user evaluates and completes  $T_r^0$ , and repeats the same operations for  $T_r^1, \dots, T_r^{K-1}$ . This means that the user decomposes a whole path to a final solution into  $K$ -subgoals. It is a kind of depth-first search shown in the search tree of Fig. 8, and users are allowed to evaluate and complete  $T_r^k$  only on  $k$ -th layer of the tree.

Furthermore, as part of preprocessing before beginning to search, we sorted the order of regions in area order. The order of regions is given by

$$\text{sum}_p(R_{in}(i)) \geq \text{sum}_p(R_{in}(i + 1)), \quad (9)$$

where  $\text{sum}_p(R_{in}(i))$  is the total number of pixels in  $R_{in}(i)$ . Equation (9) determines the order of the horizontal axis in the whole of region-matching table  $T_r$ . Thus, a user begins to evaluate and complete a small table  $T_r^0$  that occupies the largest areas in  $I_{in}$ , then treat smaller tables  $T_r^1, \dots, T_r^{K-1}$ . This sorting is also based on the intuitive idea that painters tend to begin assigning colors to large areas first. We believe that area size is an important factor for color assignment and that larger regions have some kind of higher visual impact. Therefore, our approach mimics the painter's drawing process in the real world.

$$\begin{array}{l}
\textit{Chromosome}(1): [x_0(1), x_1(1), \dots, x_{N_{in}-1}(1)] \\
\textit{Chromosome}(2): [x_0(2), x_1(2), \dots, x_{N_{in}-1}(2)] \\
\vdots \\
\textit{Chromosome}(M): [x_0(M), x_1(M), \dots, x_{N_{in}-1}(M)]
\end{array}$$

Fig. 9. Population.

$$\begin{array}{l}
\textit{Chromosome}(a): [x_0(a), x_1(a), \dots, x_n(a), x_{n+1}(a), \dots, x_{N_{in}-1}(a)] \\
\textit{Chromosome}(b): [x_0(b), x_1(b), \dots, x_n(b), x_{n+1}(b), \dots, x_{N_{in}-1}(b)] \\
\downarrow \\
\textit{Offspring}(a): [x_0(b), x_1(b), \dots, x_n(b), x_{n+1}(a), \dots, x_{N_{in}-1}(a)] \\
\textit{Offspring}(b): [x_0(a), x_1(a), \dots, x_n(a), x_{n+1}(b), \dots, x_{N_{in}-1}(b)]
\end{array}$$

Fig. 10. Crossover.

### 3.2. Operations of GA

Operations of IEC we established are illustrated by the algorithm below which runs for each  $T_r^k$  ( $k = 0, \dots, K - 1$ ). However in the following algorithm, we use  $K = 1$  for simply explanation.

1. [Start] Initialize the initial adaptive probability  $P_0(i, j)$ , which is calculated by Eq. (10).  $P_0(i, j)$  indicates the probability that  $R_{ref}(j)$  is assigned to  $R_{in}(i)$  in the initial generation. Generate a population<sup>†</sup> of  $M$  chromosomes, which is illustrated in Fig. 9, under  $P_0(i, j)$ .

2. [Fitness] A user evaluates and gives the fitness value 0/1 of each chromosome in the population.

3. [New population] Generate a new population by executing the following steps.

(i) [Selection] Select parent chromosomes whose fitness value is 1 from a population, if there are more than two chromosomes whose fitness value is 1.

(ii) [Crossover] Cross over with the crossover probability the parents to form new offsprings<sup>‡</sup>. The crossover probability decides whether executing crossover or not, and in our system it is set to 0.5. Figure 10 illustrates crossover operation. When executing crossover, a crossover point is settled at the locus between  $n$  and  $n + 1$ , which is selected randomly. *Offspring (a)* and *Offspring (b)* are generated from the parent *Chromosome (a)* and *Chromosome (b)*.

(iii) [Mutation (1)] Generate new offsprings randomly under the adaptive probability  $P_g(i, j)$ , which indicates the probability in  $g$ -th generation and is defined in Eqs. (10) and (11), except for the offsprings that the crossover formed.

<sup>†</sup>“Population”, which indicates a set of chromosomes, is a technical term of IEC.

<sup>‡</sup>“Offspring”, which indicates a chromosome in the next generation, is also a technical term of IEC.

$$\begin{array}{c}
\text{Offspring}(a): [x_0(a), x_1(a), \dots, x_n(a), \dots, x_{N_{in}-1}(a)] \\
\Downarrow \\
\text{Offspring}(a): [x_0(a), x_1(a), \dots, y_n(a), \dots, x_{N_{in}-1}(a)]
\end{array}$$

Fig. 11. Mutation (2).

(iv) [Mutation (2)] Mutate all the new offspring at each locus randomly with the mutation probability. The mutation probability is set to 0.025. Figure 11 illustrates [Mutation (2)] operation. When the locus  $n$  is selected with the mutation probability,  $x_n(a)$  changes some random value  $y_n(a)$ .

(v) [Accepting] Place the new offsprings in a new population.

4. [Replace] Use a new generated population for a further run of the algorithm.
5. [Update] Update the adaptive probability  $P_g(i, j)$  for the next generation according to Eq. (11).
6. [Test] Stop and return the best solution if a user finds a chromosome that satisfies her subjective criteria in the current population.
7. [Loop] Go to step 2.

The definition of  $P_0(i, j)$  is given below. It is based on the same idea that a larger area could be some kind of higher visual impact.

$$P_0(i, j) = \frac{\text{sum}_p(R_{\text{ref}}(j))}{\sum_{k=0}^{N_{\text{ref}}-1} \text{sum}_p(R_{\text{ref}}(k))}. \quad (10)$$

The adaptive probability  $P_g(i, j)$  in  $g$ -th generation, is assigned to  $R_{\text{ref}}(j)$  for the  $R_{\text{in}}(i)$ .  $P_{g+1}(i, j)$ , which indicates the  $(g+1)$ -th generation, is given by

$$P_{g+1}(i, j) = \frac{q(i, j)}{\sum_{k=0}^{N_{\text{ref}}-1} q(i, k)}, \quad (11)$$

where  $\omega$  is a coefficient in these equations to update the adaptive probability, and determined as follows:

$$\omega = \begin{cases} \omega_1 & : \text{for fitness 1} \\ \omega_0 & : \text{for fitness 0} \\ 1.0 & : \text{otherwise.} \end{cases}$$



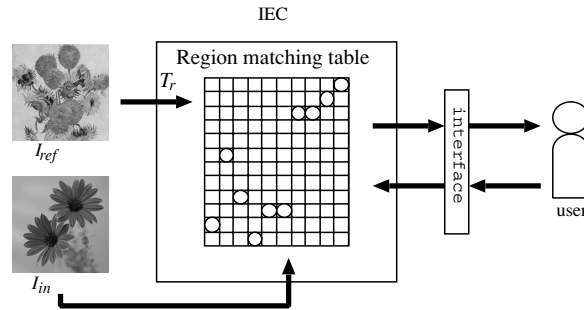


Fig. 12. System overview.

The relation between  $\omega_1$  and  $\omega_0$  is given by  $\omega_1 > 1.0 > \omega_0 \geq 0.0$ . For example,  $\omega_1 = 2.0$  and  $\omega_0 = 0.5$  in our experiments. Thus, we notice the adaptive probabilities of the regions included in a chromosome, which a user gave 1 as a fitness, increase in the next generation. The higher the adaptive probability, the bigger the chance to be selected in [Mutation (1)]. On the other hand, the adaptive probabilities of the regions included in a chromosome, which a user gave 0 as a fitness, decrease in the next generation. The history of user's evaluation is well reflected in the evolution process.

#### 4. System Overview

Figure 12 illustrates the system overview we developed. A user prepares two images which are a target photograph  $I_{in}$  and a reference painting  $I_{ref}$ . The system divides each image into coarse regions by exploiting image segmentation. The regions acquired from  $I_{in}$  and  $I_{ref}$  constitute  $xy$ -axes of the region matching table  $T_r$  (see Fig. 6). The interface of IEC is located between the users and  $T_r$ . The users evaluate and complete  $T_r$  through the interface of IEC. Of course, they do not need to handle  $T_r$  directly. In the interface of IEC, the users see and confirm the color-assigned images, which are candidates of the solution the system has produced, and users just give fitness 0/1 to them. The system produces new candidates based on the fitness, and users evaluate them again. Such processings are repeated until a candidate satisfies their subjective criteria.

#### 5. Experimental Results

Our experiments applied color-transfer processing to 10 pairs of images, a target photograph and a reference painting. Figures 13–15 show examples of our color-transfer algorithm. Figure 13 shows target photographs, and Fig. 14 shows reference paintings. Figure 15 shows examples of color-transformed image resulting from target photographs of Fig. 13. The similarity between Figs. 14 and 15 can clearly be seen.

In the experiments, we required that users assign colors of reference paintings to target photographs by using the IEC system under the condition that they arrange a similar look and feel to the reference paintings. Color-transfer processing finished when an output

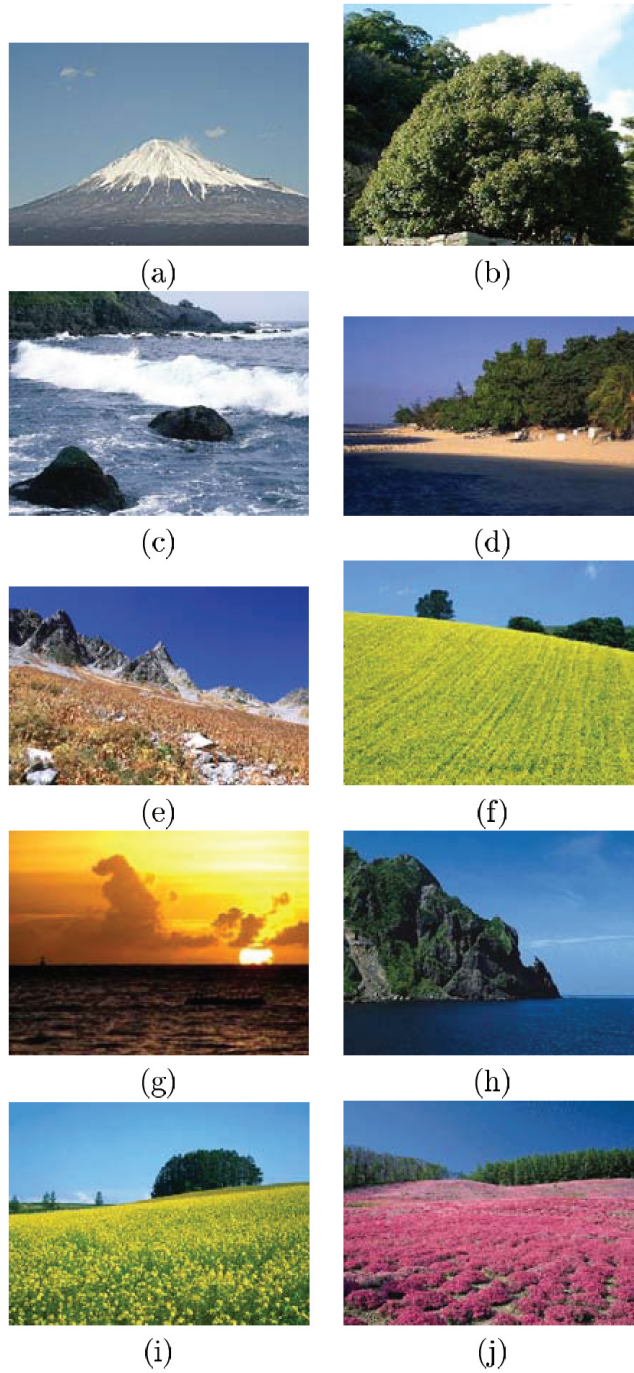


Fig. 13. Target photographs.

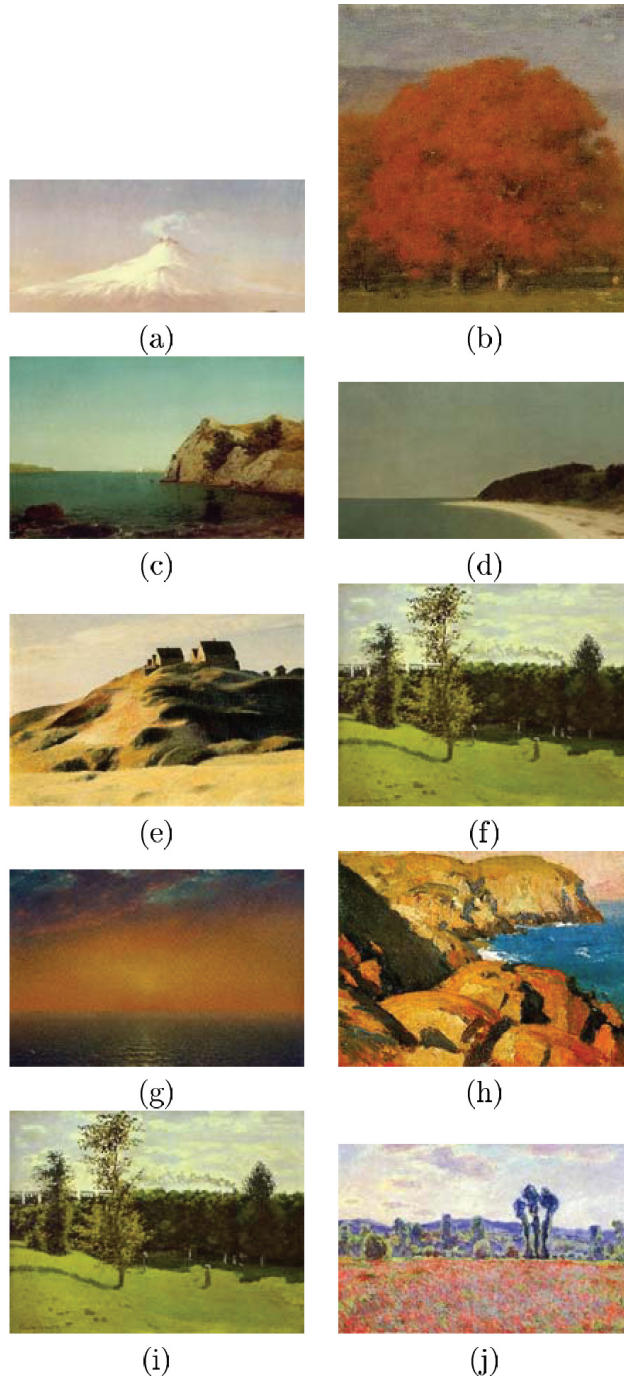


Fig. 14. Reference paintings.

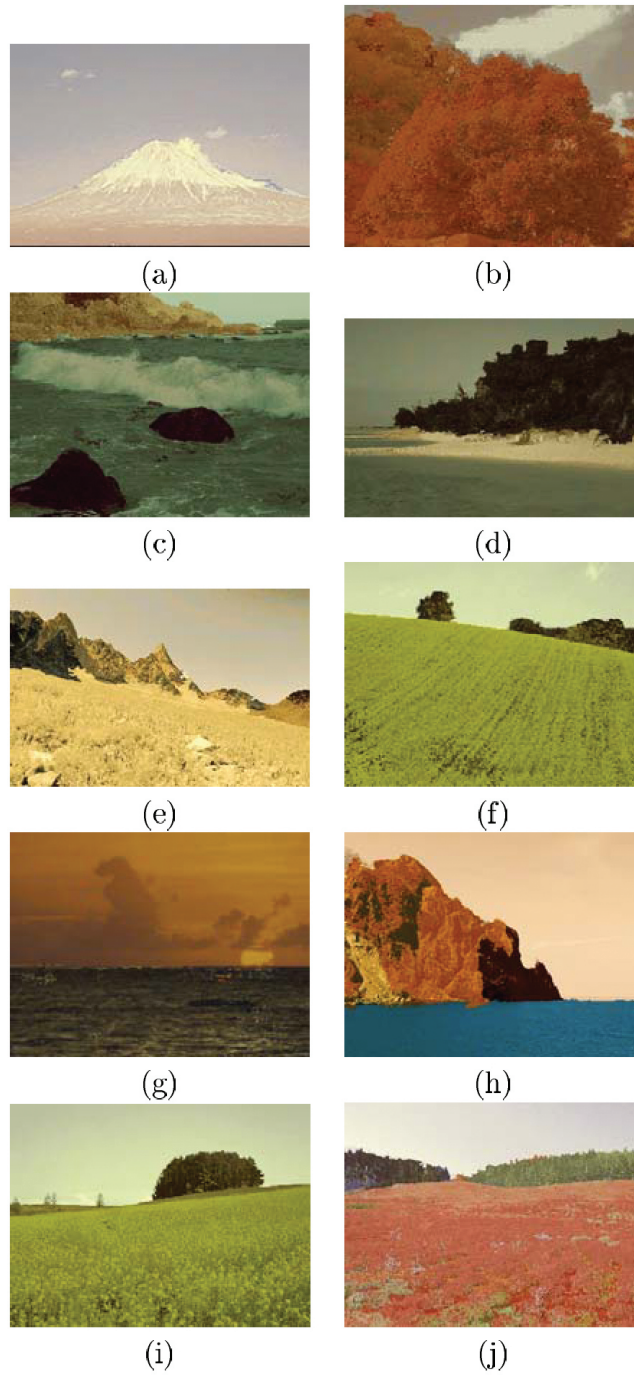


Fig. 15. Resulting images.

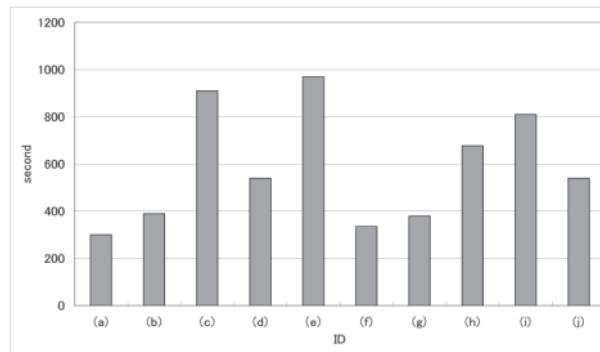


Fig. 16. Processing time to get satisfactory results (Fig. 15).

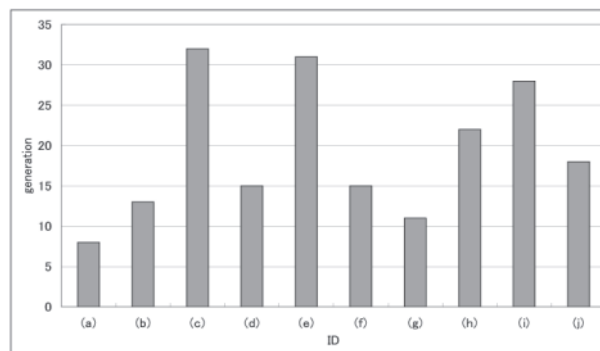


Fig. 17. Generations of IEC to get final satisfactory results (Fig. 15).

result satisfied the user's own subjective criteria. Five trials for each pair of images were made, and five color-transformed resultings were acquired for each pair. The five results might have a little visual differences although they were made using a same pair. Because our system depends on user's subjective criteria. This is one of our system's features that exploits IEC. Figures 16 and 17 show the mean processing time and the mean total generations until the user was satisfied with each final result. Our system is operating on a PC with 512 MB memory, a Celeron 2.40 GHz CPU, running the Sun Microsystem's Solaris 9 operating system.

As you see the targets and references for the experiments shown in Figs. 13 and 14, you can notice that the compositions are comparatively simple. Thus, it would not be a difficult task for a user to consider the combination needed to assign colors to a target photograph. The generation shown in Fig. 17 might represent a user's consideration time. Figure 17 indicates that 8–32 generations are needed until an output result satisfies the user's own subjective criteria. The average is 18 to 19 generations. We think this is not so



high as the conventional IEC system. However Fig. 16 indicates that it takes a considerable time to acquire the final results. It must have been hard work for the users.

## 6. Conclusion

In this paper we presented our attempts to represent a color transfer between a photograph and a painting with IEC. We believe that we acquired the color transferred images that have similar impression to the reference images shown in Figs. 14 and 15. The system using IEC requires a user just to push the buttons of the system window on the display monitor. Such easy operations can give 0/1 as fitness and evolve color transferred images interactively. On the other hand, photo retouch tools, e.g. Adobe Photoshop, are also available to operate colors interactively. However, users must learn and acquire high-level skills, while in automatic color transfer it is difficult to reflect user's subjective criteria is difficult. The position of our method is in the middle between photo retouch tools and automatic color transfer.

Some output examples were illustrated in the experiments, and we evaluated the processing time shown in Fig. 16. The processing time largely depends on the specs of the PC, but our system takes too much time for processing. The processing time is one of the most important factors for IEC system because the faster the processing, the less the user has to work. We have to resolve the problem immediately. In our research, we prepared comparatively simple images in the experiments, and we did not need to divide into many regions. If we succeed in shortening the processing time, we can handle more complex ones. These problems are left for our future work.

## REFERENCES

- CHANG, Y., SAITO, S. and NAKAJIMA, M. (2002) Color transformation based on the basic color categories of a painting, *Siggraph2002 Conference Abstracts and Applications*, pp. 157–157.
- GOOCH, B. and GOOCH, A. (2001) *Non-Photorealistic Rendering*, A. K. Peters.
- HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B. and SALESIN, D. H. (2001) *Image Analogies*, *SIGGRAPH 2001 Conference Proceedings*, pp. 327–340.
- KATAGAMI, D. and YAMADA, S. (2002) Interactive evolutionary computation for real robot form a viewpoint of observation, *7th International Conference on Intelligent Autonomous*, pp. 158–165.
- REINHARD, E., ASHIKRRHMIN, M., GOOCH, B. and SHIRLEY, P. (2001) Color transfer between images, *IEEE Computer Graphics and Applications*, **21**(5), 34–41.
- TAKAGI, H. (2001) Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation, *Proceedings of the IEEE*, **89**(9), 1275–1296.
- TAKAGI, M. and SHIMODA, H. (1991) *Handbook of Image Analysis*, University of Tokyo Press.
- TOKUI, N. and IBA, H. (2000) Music composition with interactive evolutionary computation, *3rd International Conference on Generative Art*.
- WANG, B., WANG, W., YANG, H. and SUN, J. (2004) Efficient example-based painting and synthesis of 2D directional texture, *IEEE Transactions on Visualization and Computer Graphics*, **10**(3), 266–277.