Vis Comput (2010) 26: 933–942
DOI 10.1007/s00371-010-0498-y

ORIGINAL ARTICLE

Example-based painting guided by color features

Hua Huang · Yu Zang · Chen-Feng Li

Published online: 14 April 2010

Abstract In this paper, by analyzing and learning the color features of the reference painting with a novel set of measures, an example-based approach is developed to transfer some key color features from the template to the source image. First, color features of a given template painting is analyzed in terms of hue distribution and the overall color tone. These features are then extracted and learned by the algorithm through an optimization scheme. Next, to ensure the spatial coherence of the final result, a segmentation based post processing is performed. Finally, a new color blending model, which avoids the dependence of edge detection and adjustment of inconvenient tune parameters, is developed to provide a flexible control for the accuracy of painting. Experimental results show that the new example-based painting system can produce paintings with specific color features of the template, and it can also be applied to changing color themes of art pieces, designing color styles of paintings/real images, and specific color harmonization.

Keywords Image processing · Example-based painting · Color features learning

H. Huang (✉) · Y. Zang
School of Electronic and Information Engineering, Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an, China
e-mail: huanghua@xjtu.edu.cn

Y. Zang
e-mail: zangyu7@126.com

C.-F. Li
School of Engineering, Swansea University, Swansea, UK
e-mail: c.f.li@swansea.ac.uk

1 Introduction

In computer painting and image synthesis, an example-based approach creates paintings/images by automatically modifying the source image to imitate some specific features of the reference image. Many existing methods try to learn the texture features of a painting [10, 16] and although they present impressive results, the color features of a painting are seldom considered. In this paper, we mainly focus on how to learn and extract the color features of the template painting, and to transfer to the source image the color theme and color style.

When painting, a artist chooses colors from a certain range to express a specific emotion and, to emphasize the emotion, the use of color is often exaggerated, e.g. the sky may no longer be painted in blue and the meadow may no longer be green. Real images such as photos taken by camera are created to describe the real world as factually as possible and, except for some specific art photos, they often lack vivid emotion. Hence, in order to transfer the artist's emotion from a painting to a real image, a key task is to learn and copy the color features. Choosing the right color to paint may not be difficult for specialized artists, but it can be very confusing for an amateur user who only has a passionate admiration for a reference painting and a vague request "make it like this."

Traditional color transfer methods [1, 6, 12, 13, 15, 17] are not suitable for this task because they aim to directly transfer the colors from one image to the other, but do not address the color features that determine the overall color style of the reference. These direct color transfer approaches often fail to transfer the emotion of the template painting, and some important visual features of the input image (such as the relationship of light and shadow) tend to get damaged. On the other hand, painters never simply copy the colorful world to the canvas and instead they create paintings

with their own choices of colors. The emotion (e.g., passionate or apathetic) of a painting is largely determined by its color features and associated global color style (e.g., warm or cold). Thus, when rendering a real image, these color features should be learned from the template painting and intrinsic structural characteristics of the input image should also be preserved.

The contribution of this paper is a new example-based approach that transfers automatically some key color features from the template to the source image. The result image is created with the emotion of the template painting while preserving the structural content of the input image. First, color features of a given template painting are analyzed in terms of hue distribution and the overall color tone, and these features are then extracted and learned by the algorithm through an optimization scheme. A segmentation based post-processing technique is developed to ensure the spatial coherence of the painting result. When painting on the canvas, a color blending model is designed to control the accuracy of the painting process more flexibly. The paper is organized as follows. Section 2 briefly reviews some representative works related to this study. The measures we defined for the analysis of color features are explained in Sect. 3. In Sect. 4, a new learning algorithm is developed based on an optimization scheme with constraints specific to color features. Section 5 describes how to render a real image towards a painting. Some experimental results are presented in Sect. 6 with conclusions being made in Sect. 7.

2 Related work

Painterly rendering is not a new subject and has been extensively studied in the past decade [7, 8, 11, 19]. Litwinowicz [11] proposed an automatical rendering algorithm to generate impressionistic effects. In this approach, properties of a brush stroke including position, color, size, and orientation, etc. are predefined without interaction and, for an input video, the optical flow is used to ensure the temporal coherence. Hertzmann [8] improved this technique by introducing curved brush strokes and a layered painterly method. This scheme simulates the real painting process in a more scientific manner. Hays and Essa [7] made some modifications to stroke orientation and painting style variation, and extended this work to video rendering. Zhang et al. [19] presented a video-based algorithm for synthesizing animations of running water in the style of Chinese paintings. In this work, the authors made some novel contributions on painting structure analysis and stroke placement. The painterly rendering methods outlined above focus mainly on how to simulate the real painting process while less attention is being paid to learning the high-level features of a painting.

This feature-learning issue is addressed to some extent by some example-based painting methods [4, 10, 16]. Hertzmann et al. [10] synthesized the target image B' from the source image B by imitating the relationship between a pair of reference images A and A'. However, their work is pixel-based and its performance is relatively slow. Wang et al. [16] proposed a patch-based method to speed up the synthesis process. The basic idea of their method is using some patches defined on the example painting to synthesize the target image according to a calculated orientation field. The patches are specified by users to best represent the style of the example painting. Drori et al. [4] introduced an example-based synthesis technique that can extrapolate the style of an given input image. First, the reference image is decomposed into fragments with specific styles and contents, and then the source image with different style and content is partitioned into fragments adaptively, after which these source fragments are stitched together, according to the relationship between the reference fragments, to create an image with the new style. These methods focus on learning the texture features of the template paintings, and they operate mainly on the luminance channel while the color of the input image remain unchanged [10, 16].

Many color transfer methods have also been developed to map colors from one image to the other [12, 13, 15, 17]. Reinhard et al. [13] proposed an efficient global color transfer method which rescales some color statistics (e.g., mean and standard deviation) of the input image according to a given reference image. Pitie et al. [12] proposed a n-dimensional PDF transfer method that can be applied to global color transfer in 3D cases. Tai et al. [15] proposed a soft segmentation method that could locally transfer colors from one image to another with less spatial artifacts. To resolve the fidelity problem, Xiao and Ma [17] presented a color transfer algorithm that preserve the gradient field of the input image via an optimization procedure. However, these methods concentrate on direct color transfer without specifying the key color features that determine the overall color style and emotion of the template. As a result, the emotion of the template painting may not be expressed sufficiently, and some other important structures of the input image (e.g., pattern of light and dark) tend to get lost. Hence, they are not suitable for the task of this paper.

This work is also related to color harmonization [2, 14]. To adjust the colors of a source image based on a predefined harmonic template, Cohen-Or et al. [2] proposed a recoloring scheme that preserves the spatial coherence among neighboring pixels by the graph cut optimization. Sawant and Mitra [14] extended this work to videos. To deal with flickering artifacts, an input video is divided into some subsequences of adjacent frames with overlap, and then an average orientation θ is used on each subsequence to produce harmonic videos with less flicker. The above harmonization

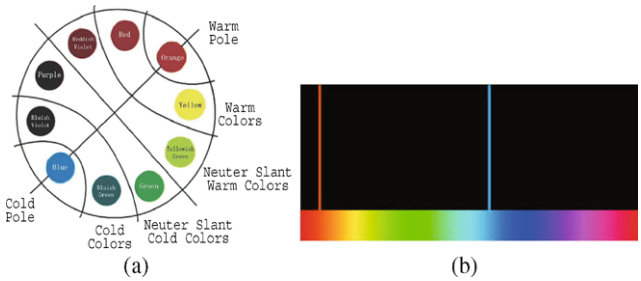


Fig. 1 Color features. (**a**) Warm and cold colors. (**b**) The warm/cold pole in the hue space

work aims to produce images or videos with harmonic perception. To some extent, our work can also be viewed as a harmonization process which harmonizes a real image to imitate some color features of a reference painting. However, general templates that suit for our purpose are difficult if not impossible to define, in that there are countless color styles in painting and they are full of individualities. Hence, our harmonization template is chosen as a specific painting.

3 Color features of paintings

People perceive the outside world through the human perceptual system and in many cases, we rely largely on the visual system, for which the most sensitive perception is often the color. Artists also consider color as the most important feature of a painting. Different choices of colors, in terms of range and composition, assign different emotions to a painting. However, most of the real images such as photos taken by a camera, aim to describe the real world as truthful as possible without the emotional soul. It has long been desired to extract the color features of a painting, but only until recently the issue was addressed in a quantitative manner [2, 5, 14]. Cohen-Or et al. [2] and Sawant and Mitra [14] placed the hue of an image to some predefined harmonic templates to make it visually harmonic, and suggested that the hue of an image is perhaps the most visible color feature since its distribution directly reflects the overall tone and the basic emotion of a painting. Greenfield and House [5] claimed an object in a painting should be represented by structured values, namely the pattern of light and dark, which vary with the content of the painting and should be preserved. This work was motivated by the above ideas and our first goal is to learn the hue distribution of the given template painting while preserving the pattern of light and dark of the input image. That means the learning process of color features is performed on HSV color space (since it represents better the color perception of human), and the algorithm modifies the hue channel only.

Considering the hue distribution alone may not be enough to define the color style and emotion of a painting. For example, two images with opposite main-color area and complem-entary-color area will show similar hue distribution, but have drastically different style and emotion. Therefore, color tone, another important color feature, should also be considered to avoid the above contradictory case. Humans have different feelings, cold or warm, when facing different colors because of psychological reaction. For most people, red, orange, or yellow, etc. are considered as warm colors while green, blue, or royal purple, etc. are considered as cold colors, and yellowish green, fuchsia, etc. are considered to be neutral. The combination of the hue distribution and the color tone provides a simple measure for the overall color style and emotion of a painting.

Of course some other color features such as the contrast of light and shade, complementary relation, and color hyperbole are also very important in a painting. But these features are content related and how to learn these features can be summarized as a semantic segmentation problem. In this paper, we mainly focus on the global color features that are suitable for content-independent learning.

To measure the tone of a painting, we first exam the schematic definition from artists. As shown in Fig 1 (a), different colors give different degrees of visual temperature with blue being the coldest and orange the warmest. Taking into consideration the cold and warm poles in the hue space (as shown in Fig. 1 (b)) we can define a measure for the tone of a painting. For a color, the closer it is to the cold pole the greater cold degree it has and vice versa. Noting that colors with lager saturation often give a greater stimulation, we define the cold/warm degree of a painting as:

$$M(I)_{\text{cold/warm}} = \sum_{h \in \mathscr{H}(I)} sat_h * \frac{area_h}{||h - h_{\text{cold/warm}}||} \tag{1}$$

where $M(I)_{\text{cold/warm}}$ represents the measure of the cold/warm degree of an image I, h denotes the hue value and $h_{\text{cold/warm}}$ the hue value of the cold/warm pole, $\mathscr{H}(I)$ the hue domain of image I, $area_h$ the percentage of the pixels with hue value h in the whole image, and sat_h the average saturation of pixels with hue value h. Based on $M(I)$, the quantitative measure of the cold/warm degree, the tone of an image can be defined as

$$I.tone = \begin{cases} \frac{M(I)_{\text{warm}}}{M(I)_{\text{cold}}} & M(I)_{\text{warm}} \geq M(I)_{\text{cold}} \\ -\frac{M(I)_{\text{cold}}}{M(I)_{\text{warm}}} & M(I)_{\text{warm}} < M(I)_{\text{cold}} \end{cases} \tag{2}$$

Figure 2 shows that paintings with similar color features defined above give us similar feelings. The tone values for (a) and (b) are 24.88 and 28.79, respectively, and their hue distributions are shown in (c) and (d). The two paintings express similar passionate emotion because of their similar color features. This simple observation verifies that the hue distribution and the tone value defined above can be used to represent the global color style and emotion of an image.

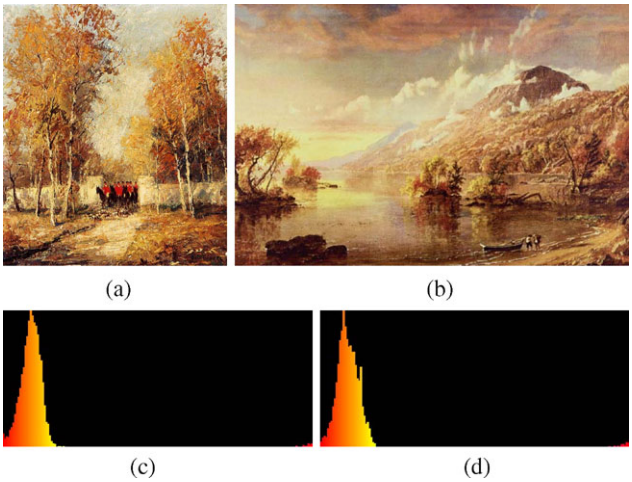


Fig. 2 Paintings with similar color features. Tone values for paintings (**a**) and (**b**) are 24.88 and 28.79, respectively, and their hue distributions are shown in (**c**) and (**d**), respectively. Paintings of different contents can give similar perceptual feelings (*top row*), and their color features measured by the hue distribution (*bottom row*) and the tone are found similar

4 Learning color features

Given a template painting, a straightforward way for an input image to learn the hue distribution from the template image is moving every bin in the hue histogram of the input image to the position of the nearest bin in the template hue histogram. But this simple idea has the following problems:

1. The set of colors of the input image may be reduced, and the color reduction becomes increasingly more visible when the difference between the input image and the template painting increases.
2. The second measure regarding the tone is completely ignored in this simple approach and in some cases, wrong color style and emotion may be created from the input image.
3. It is difficult to avoid pseudo edges and ensure the spatial coherence of the result image.

To overcome the above problems, the hue histograms of the input image and the template image are both divided into blocks and, following an optimization approach with a specially designed objective function, the hue blocks of the input image are moved to the positions of the hue blocks of template painting, after which a block merging operation is performed to produce spatially coherent results. Following the establishment of hue-block mapping, the hue values of pixels in each block are modified correspondingly.

4.1 Hue-space partition

Although it is ideal to operate on every single hue bin in the hue distribution diagram, this will make it too complex to cope with the spatial coherence at the later stage. Therefore, an initial partition is performed to group neighboring hue bins into blocks which then serve as the basic operation unit in our algorithm. After calculating the hue histogram of the input image, decorative colors whose percentage are lower than 0.2% (this value works well for most images in our examples while it can be adjusted to suit other cases) in the whole image are removed in order to preserve the main colors. Then all local minima are detected in a certain window and the size of the window determines indirectly on which scale the histogram is partitioned. In this work, the radius of the window is set to 1, which means if the hue value of a bin is less than its two adjacent bins in the hue histogram, then it is considered to be a local minimum. Once all local minima of a hue histogram are detected, the hue blocks are defined as the parts between two neighboring local minima. The input image and the template image are processed in the same way to form the hue blocks.

Some preliminary concepts regarding the properties of hue blocks are defined here: *gravity* represents the weighted average hue value of a block based on the area of each hue bin; defined in the same way, *sat* represents the weighted average saturation value of a block; *area* denotes the normalized area of a block in the hue histogram; *category* denotes which part of the segmented image a specific block belongs to (more details about this property will be explained in Sect. 4.3); *tone* denotes the tone value of a block. Based on these quantitative properties, a novel objective function is defined to control the leaning scheme via an optimization approach.

4.2 Learning algorithm based on binary graph matching

There are two targets for the learning algorithm for color features, i.e., the hue distribution and the overall color tone of the given template painting. The simple idea mentioned earlier is likely to cause color reduction. To overcome the problem, a basic principle is if a position in the hue histogram of the template painting has already been seated, then no other blocks are allowed to move into it. Based on this principle, a model using binary graph matching can be set up to describe the operation. Let $S(I) = b_{i_1}, b_{i_2}, \ldots, b_{i_n}$ denote the block sets of the input image and $S(T) = b_{t_1}, b_{t_2}, \ldots, b_{t_n}$ denote the block sets of the template painting. A graph G is constructed after connecting each node in $S(I)$ to every node in $S(T)$ by an edge with a cost. A match between node sets $S(I)$ and $S(T)$ is defined as a subgraph of G such that none of its edges share the same vertex. Our task is to find a match M minimizing the total sum of its edge cost, i.e.,

$$\arg\min_{M} \sum_{e_{uv} \in M} c_{e_{uv}} \tag{3}$$

where u and v are the node numbers of $S(I)$ and $S(T)$, and $c_{e_{uv}}$ the cost of each edge e_{uv}. Based on the learning targets, the edge cost is defined as

$$c_{e_{uv}} = E_1(b_{i_u}, b_{t_v}) + \lambda * E_2(b_{i_u}, b_{t_v}). \tag{4}$$

The energy E_1 accounts for the distance between two blocks b_{i_u} and b_{t_v} and is weighted by the block's *area*. Specifically, it can be written as

$$E_1 = b_{i_u}.area * \|b_{i_u}.gravity - b_{t_v}.gravity\| \tag{5}$$

where $b_{i_u}.area$ and $b_{i_u}.gravity$ are the area and gravity of the block b_{i_u}, as defined in Sect. 4.1. Area $b_{i_u}.area$ can be easily obtained from the hue histogram and $b_{i_u}.gravity$ can be calculated as

$$b_{i_u}.gravity = \frac{\sum_{h_k \in \mathscr{H}(b_{i_u})} h_k * area_{h_k}}{\sum_{h_k \in \mathscr{H}(b_{i_u})} area_{h_k}} \tag{6}$$

where h_k is the kth hue value in block b_{i_u}. The property *sat* of a block can be calculated in a similar way and the only difference from the above expression is that h_k is replaced by sat_{h_k} (sat_{h_k} and $area_{h_k}$ are defined in Sect. 3). The energy E_2 is designed to encourage the tone value of the input to be moved towards the direction of the template painting. Specifically, it is defined as follows:

$$E_2(b_{i_u}, b_{t_v}) = \delta(b_{t_v}.tone, T.tone) \times \|b_{i_u}.tone - b_{t_v}.tone\| * b_{i_u}.area. \tag{7}$$

In the above expression, $T.tone$ denotes the tone value of the template image T and can be calculated by (2). Expressions $b_{i_u}.tone$ and $b_{t_v}.tone$ can be obtained in a similar way and the only difference is that they are defined in the hue domain of a block, where the variables sat_h, $area_h$, and h in (2) are replaced by $b_{i_u}(b_{t_v}).sat$, $b_{i_u}(b_{t_v}).area$ and $b_{i_u}(b_{t_v}).gravity$, respectively. Function $\delta(b_{t_v}.tone, T.tone)$ equals -1 if $b_{t_v}.tone$ and $T.tone$ have the same sign and equals 1 if their signs are different. Parameter λ in (4) is a weight to balance the two energy functions, and its value depends mainly on images. A small value (typically about 0.1–0.5) of λ is suitable if the tones of the input and the template are similar and, if the tones of them are very different, a lager λ is desired (typically about 5.0–10.0).

After defining the energy function in (4), the optimization problem defined in (3) can be solved by the classic Kuhn–Munkres algorithm (also known as the Hungarian method). It should be noted that the number of blocks in the input image is usually different from the template painting. If $i_n < t_n$ then the operating procedure mentioned above is performed without change. If $i_n \geq t_n$, then the set $S(T)$ is firstly filled with some special blocks whose *gravity* are set to -1. The weights of the edges connected to these blocks are set to infinite. The blocks matched onto these special blocks after the first matching will be matched again or moved to the nearest positions in $S(T)$.

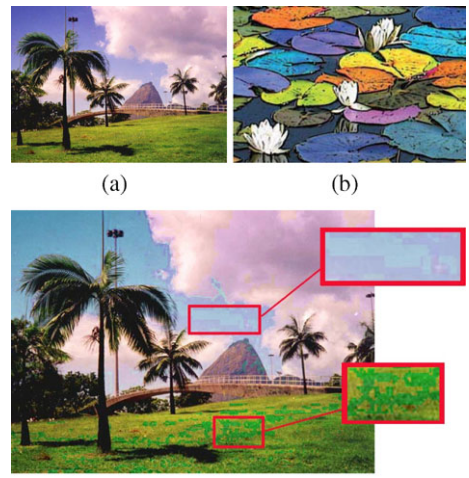


Fig. 3 An example to show the pseudo edges. (**a**) The input image. (**b**) The template painting. (**c**) The result without coherence processing

4.3 Spatial coherence processing

Although color features of the template painting are learned and transferred to the input image after optimization, some defects are observed in the result image due to the change of colors. As shown in Fig. 3, in which (a) and (b) show respectively the input image and the template painting, there are some pseudo edges in the result image (c). These pseudo edges arise because the blocks which are initially contiguous in $S(I)$ are moved apart after the learning process. To remove the discontinuity caused by pseudo edges, we propose a block merging scheme, which mainly include two steps: detecting the blocks related to the pseudo edges and merging these blocks.

The pseudo edges are formed by many broken regions, each of which is centered around two broken blocks. Blocks b_{i_u} and $b_{i_{u-1}}$ in $S(I)$ are considered as broken blocks when,

$$\begin{aligned} &\|b_{i_u}.gravity - b_{i_{u-1}}.gravity\| < t_1 \quad \text{and} \\ &\|M(b_{i_u}).gravity - M(b_{i_{u-1}}).gravity\| > K * t_1 \end{aligned} \tag{8}$$

where $M(\cdot)$ indicates the matching blocks in $S(T)$ and t_1 and K are tune factors, which are set respectively as $t_1 = 5$ and $K = 2$ throughout the whole paper. The broken region can be detected by expanding the broken blocks from both sides. The expanding process is terminated when one of the following conditions is satisfied.

1. The *category* of the expanded block and the current block are different.
2. A predefined expanding threshold t_2 is reached.
3. The border of the neighboring broken region is reached.

(a) (b)

Fig. 4 Result after coherence processing. (**a**) The segmented image. (**b**) The result obtained after spatial coherence processing

The *category* of a block mainly represents which region a block belongs to in the segmented image, which is created by segmenting the hue channel of the input image. As the hue-space partition procedure in Sect. 4.1 has already taken into account some spatial information, the segmentation process does not need to be very accurate and it is better to perform the segmentation in a larger scale. In this work, an initial mean shift filtering [3] is performed to smooth some fine details, which is followed by a classic pyramid segmentation algorithm. The pyramid segmentation is employed due to its simplicity and if necessary, the segmentation result (especially for domains with gradually changing hue values) can be further improved by using specialized segmentation algorithms, e.g., trapped ball technique [18]. The segmented result is shown in Fig. 4(a). A block belongs to a specific region if most of its pixels are contained in that region, and then a label *category* is assigned to this block to denote the region it belongs to. The tune factor t_2 is defined manually and it sets a hard constraint of the expanding process, for which a larger value of t_2 indicates a stricter coherence processing scheme and vice versa.

After detecting the broken regions, all blocks related to the same broken region are merged into one and the properties of this new block are also recalculated. The matching block of this new one is set as the matching block of its component block with the largest *area*. The pseudo edges can be effectively removed by this post-processing scheme and the result is shown in Fig. 4(b). Once the match M is confirmed, the hue values of pixels in each input block are modified, according to the matching reference block, to complete the learning process.

5 Image painting

After more than ten years of development, image painterly rendering has become a relatively mature subject. This paper employs some latest techniques reported in [7–9]. An edge clipping scheme is often used in traditional methods to preserve significant edges, but the effect of edge clipping greatly depends on edge detection and the adjustment of tune parameters such as the number of layers and the stroke radius etc. Hence, we propose a color blending model for painting, and the accuracy of painting process is controlled more flexibly and conveniently by using a single parameter.

(a) (b) (c)

Fig. 5 Comparison of the color blending model. The color blending model provides a easier way to control the accuracy of the painting process. Shown in (**a**) is the result of Hertzman [8]; (**b**) and (**c**) are our results of the color blending model with different parameters

When an artist is creating a painting, the strokes are always painted to those canvas regions that are more different from the target scene such that the painting procedure is a converging process. In the traditional approaches of computer-aided painting, if an edge is missed in edge detection then the edge clipping scheme can not prevent a bad stroke from covering good strokes. To improve the painting process, we propose a color blending model in which the new color of a pixel C_{new} is determined by $C_{\text{old}}(p)$ (the color of the pixel p before rendering), $C_r(p)$ (the color of the reference pixel at the same position), and C_s (the color of the stroke). Specifically,

$$C_{\text{new}}(p) = (1 - W) * C_{c_{\text{old}}}(p) + W * C_s. \tag{9}$$

The weight W in (9) is

$$\begin{cases} (1-\alpha) * G_\sigma(\|d_c(p) - d_s(p)\|) & \text{if } d_c(p) \le d_s(p) \\ 1 - (1-\alpha) * G_\sigma(\|d_c(p) - d_s(p)\|) & \text{if } d_c(p) > d_s(p) \end{cases} \tag{10}$$

where $d_c(p)$ is the difference between $C_{\text{old}}(p)$ and $C_r(p)$, and $d_s(p)$ is the difference between C_s and $C_r(p)$. The parameter α controls the trueness of the painting, and a lager value of α means a more realistic result. Function $G\sigma \in (0, 1]$ is the normalized Gaussian function with mean 0 and standard deviation σ. Knowing the exact standard deviation σ should be $\max(\|d_c(p) - d_s(p)\|)/2$, we use a simple approximation $\sigma = \sqrt{255^2 + 255^2 + 255^2}/2 = 220$. Figure 5 (a) shows the result of [8], in which several tune parameters need to be adjusted in order to obtain a more realistic result. With the new color blending model, the painting process is controlled more easily and it is also independent of the edge detection result. Figures 5 (b) and (c) show the results of the color blending model with parameters $\alpha = 0.95$ and $\alpha = 0.99$.

6 Results and application

Traditional color transfer methods emphasize a direct transformation of colors between images, and they do not address color features that determine the global color style and emotion of a painting. As a result, the emotion of the template painting may not be expressed sufficiently and some intrinsic visual features such as the lightness and the shade pattern of the input image often change with different reference paintings. The proposed method learns key color features from the template image and preserves structural features of the input image. Figure 8 shows a set of results. The input image with tone value 3.37 is shown in (a) and its hue distribution is shown in the bottom right corner. The two reference paintings were created by *Vincent Van Gogh* and they are shown respectively in (b) (*Night Cafe* with tone value 10.17) and (c) (*Starry Night* with tone value −19.80). The middle row shows the painting results using the proposed method, where painting (d) is created according to reference (b), and painting (e) is created with reference (c). Their hue distributions are shown respectively in the bottom right corner of images (f) and (h), and the tone values are 5.33 and −6.28 respectively. The bottom row shows the difference between the proposed method and traditional color transfer methods, where the latest color transfer approach [17] is chosen for comparison. Figures (f) and (h) are our results before painting and the corresponding results of [17] are shown in (g) and (i), respectively. It is observed that the red cloud in image (g) is visually uncomfortable due to over saturation and the overall lightness of the input image is reduced visibly in image (i). The color distance between (g), (i) and the corresponding reference paintings (b) and (c) may be small, but it is hard to say they have the color style and emotion similar to the references.

Besides paintings, the color features and the learning scheme proposed here are also suitable for some other art pieces. Figure 9 shows the results of modifying the color theme of cartoon images. Shown in (a) and (b) are two input images with neutral tone values −1.32 and −1.11, respectively. These two input images do not have a clear color emotion, and their color theme can be changed to obtain a passionate emotion by using the painting template (c). The corresponding results are shown in (e) and (g), respectively, and their tone values are 9.38 and 8.72, respectively. If a template with similar neutral tone value is used (as shown in (d) whose tone value is −1.83), learning color features can bring some fine adjustments to the input images. As shown in results (f) and (h), the colors do not change much compared to the input images, but the color features of the rocks and sea in the setting sun in the template (d) have been expressed well with their tones stay almost unchanged at 1.39 and −1.69, respectively.

The proposed method also allows users to adjust the color style or the tone value of an image without using a template.

(a) (b) (c)

Fig. 6 Our method applied to color harmonization. (**a**) The disharmonious image from [2], (**b**) The harmonization result from [2], (**c**) The harmonization result using our method

(a) (b) (c)

Fig. 7 A failed example. (**a**) The input image. (**b**) The template painting. (**c**) Our result. The large white area on the beach remains unchanged and it appears disharmonious

Based on the tone value and using a simple linear contraction method, users can control the cold/warm degrees of an image by specifying a desired rough tone value. Figure 10 shows some interesting results. An input painting with tone value 1.31 is shown in (a). If warm tones are desired, the reference tone values can be chosen as 5 and 15, and correspondingly warm images are obtained respectively in (b) (tone value 5.21) and (c) (tone value 18.67). If cold tones are desired, the input can be set to −5 and −25, then cold images can be obtained in (d) (tone value −5.89) and (e) (tone value −29.02). It is apparent that the warm/cold feeling expressed in the image is proportional to the input tone value. This technique can also be used to process real photos, e.g., the indoor color design shown in the bottom row of Fig. 10. The original design is shown in (f). Users can redesign the scheme with warm tone (g) (input tone value 25) or cold tone (h) (input tone value −15). In addition, it is noted that different poles can be defined to obtain arbitrary color styles, and shown in (i) is an example taking green as the desired pole.

The proposed learning technique for color features can also be applied to color harmonization. To some extent, this work can also be treated as a color harmonization process. Specifically, traditional harmonization methods harmonize the colors of the input image based on some predefined templates, and in this work the color template is the reference image. Figure 6 (a) is an example taken from Cohen-Or et al. [2], of which the foreground is considered disharmonious

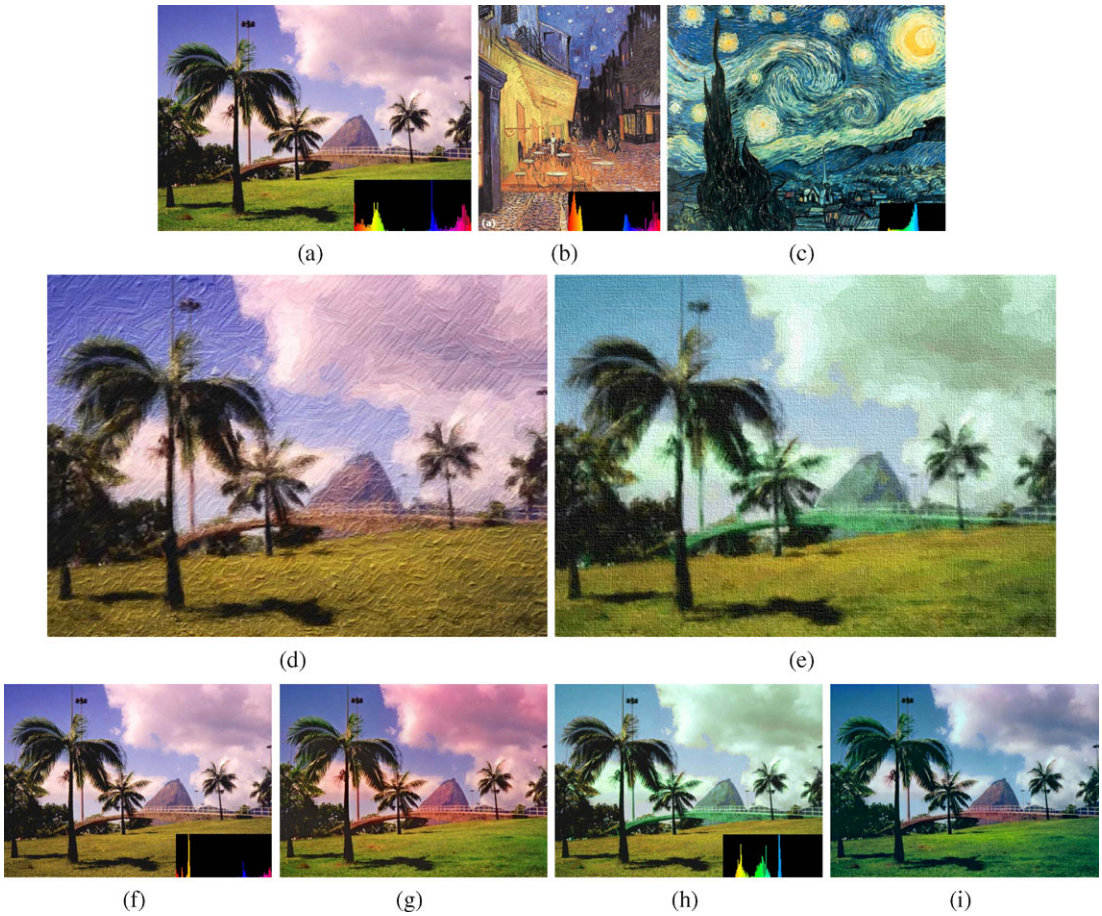


Fig. 8 The proposed example based painting. *Top row*: the input real image (**a**) and two template paintings (**b**) and (**c**) by Vincent Van Gogh. *Middle row*: results obtained respectively from (**b**) and (**c**) using the new method. *Bottom row*: comparison with a latest color transfer method

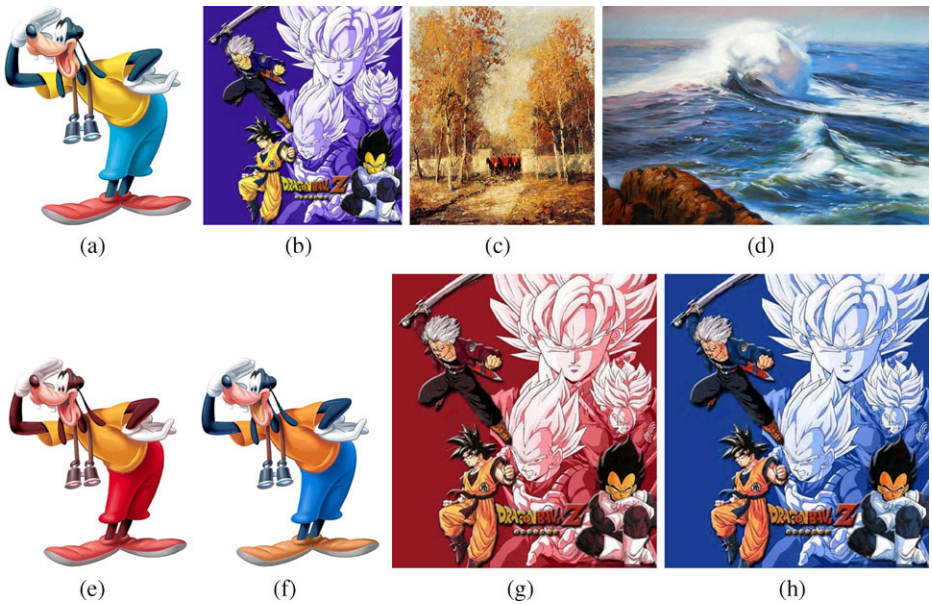


Fig. 9 Our method applied to color theme changing. *Top row*: input images (**a**) and (**b**) and the template paintings (**c**) and (**d**). *Bottom row*: (**e**) and (**g**) are the results learned from (**c**), and (**f**) and (**h**) are the results learned from (**d**)

Fig. 10 Our method applied to basic tone design. *Top row*: (**a**) the original painting, (**b**) and (**c**) paintings designed with warm tone, (**d**) and (**e**) paintings designed with cold tone. *Bottom row*: (**f**) the input photo, (**g**) warm style design, (**h**) cold style design, (**i**) custom style design

with the background. Shown in (b) is the result presented by Cohen-Or et al. [2] which harmonized the foreground with respect to the background. As shown in (c), the harmonization effect can also be archived using the proposed method by taking input image as the foreground object (the girl) and the template image as the background. As the graph-cut technique is used in Cohen-Or et al. [2], the separated areas belonging to the same object may not be deduced, as highlighted by the red ellipse in (b). As one object is likely to have similar colors, this unsatisfactory situation is avoided to some degree in the proposed method, as shown in (c).

Performance We performed our algorithm on a PC with an Intel 3.0 GHz dual-core CPU and a GeForce 9600 GT video card. The computation time depends on the image size and experimental parameters, and the typical value for a 600×800 image is about 8–10 seconds.

7 Conclusion and limitation

In this paper, we propose an example-based painting scheme guided by color features. After defining two key color features for paintings, an optimization based learning scheme is presented to transfer the color features from the template paintings to the input image. In order to obtain results without discontinuity artifacts, a spatial coherence processing scheme is also developed. When painting, a color blending model is designed to control more flexibly the accuracy of the painting process, which avoids the dependence of edge detection and adjustment of inconvenient tune parameters. Comprehensive examples are presented to demonstrate the performance of the proposed approach in different applications, including example-based image painting, example-based color theme changing and color styles design, and color harmonization, etc.

The new technique allows the input image to learn color features from the template image while preserving its own structural features. However, in some extreme cases, colors like white and black could not be handled well. This is because these two colors have the minimum saturation and lightness such that stay unchanged in the learning process. Figure 7 shows a failed example, where (a) is the input image and (b) is the template painting. As a large area in the beach are rendered with white, this area stays almost unchanged in the result (c) and appears disharmonious. For future work, the problem might be tackled by combining the new example-based approach and the traditional color transfer method.

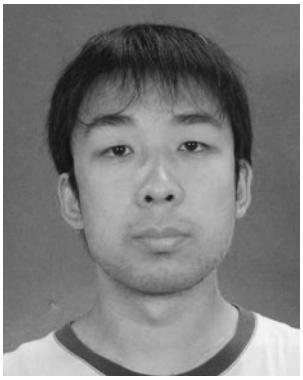
Acknowledgements This work is partially supported by the National Natural Science Foundation of China (Grant No. 60970068), the Key Project of Chinese Ministry of Education (Grant No. 109142) and the MOE-Intel Joint Research Fund (Grant No.MOE-INTEL-09-07).

References

1. Chang, Y., Saito, S., Nakajima, M.: A framework for transfer colors based on the basic color categories. In: Computer Graphics International, pp. 176–183 (2003)
2. Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., Xu, Y.: Color harmonization. ACM Trans. Graph. **25**(3), 624–630 (2006)
3. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. (5), 603–619 (2002)

4. Drori, I., Cohen-Or, D., Yeshurun, H.: Example-based style synthesis. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 143–150 (2003)
5. Greenfield, G., House, D.: Image recoloring induced by palette color associations. J. Winter Sch. Comput. Graph. **11**(1), 189–196 (2003)
6. Grundl, M., Dodgson, N.A.: Color search and replace. In: Workshop on Computational Aesthetics, pp. 101–109 (2005)
7. Hays, J., Essa, I.: Image and video based painterly animation. In: Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering, pp. 113–120 (2004)
8. Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes. In: Proceedings of SIGGRAPH, pp. 453–460 (1998)
9. Hertzmann, A.: Fast paint texture. In: Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering, pp. 91–96 (2002)
10. Hertzmann, A., Jacobs, C., Oliver, N., Curless, B., Salesin, D.: Image analogies. In: Proceedings of SIGGRAPH, pp. 327–340 (2001)
11. Litwinowicz, P.: Processing images and video for an impressionist effect. In: Proceedings of SIGGRAPH, pp. 407–414 (1997)
12. Pitie, F., Kokaram, A., Dahyot, R.: N-dimensional probability density function transfer and its application to colour transfer. In: Proceedings of International Conference on Computer Vision, pp. 1434–1439 (2005)
13. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Comput. Graph. Appl. (5), 34–41 (2001)
14. Sawant, N., Mitra, N.: Color Harmonization for Videos. In: Proceedings of the Indian Conference on Computer Vision, Graphics & Image Processing, pp. 576–582 (2008)
15. Tai, Y., Jia, J., Tang, C.: Local color transfer via probabilistic segmentation by expectation-maximization. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 747 (2005)
16. Wang, B., Wang, W., Yang, H., Sun, J.: Efficient example-based painting and synthesis of 2d directional texture. IEEE Trans. Vis. Comput. Graph. **10**(3), 266–277 (2004)
17. Xiao, X., Ma, L.: Gradient-preserving color transfer. Comput. Graph. Forum **28**(7), 1879–1886 (2009)
18. Zhang, S., Chen, T., Zhang, Y., Hu, S., Martin, R.: Vectorizing cartoon animations. IEEE Trans. Vis. Comput. Graph. **15**(4), 618–629 (2009a)
19. Zhang, S., Chen, T., Zhang, Y., Hu, S., Martin, R.: Video-based running water animation in Chinese painting style. Sci. China Ser. F **52**(2), 162–171 (2009b)

Hua Huang is an associate professor in School of Electronics and Information Engineering, Xi'an Jiaotong University. He received his B.S. and Ph.D. degrees from Xi'an Jiaotong University in 1996 and 2006, respectively. His main research interests include image and video processing, pattern recognition, and machine learning.



Yu Zang is currently a doctoral student of School of Electronics and Information Engineering, Xi'an Jiaotong University. He received the Bachelor's degree in Xi'an Jiaotong University in 2008. His main research interests include image and video processing.

Chen-Feng Li is currently a lecturer at the School of Engineering of Swansea University, UK. He received his B.Eng. and M.Sc. degrees from Tsinghua University, China in 1999 and 2002, respectively, and received his Ph.D. degree from Swansea University in 2006. His research interests include computer graphics and numerical analysis.