

## Data-Driven Image Color Theme Enhancement

Baoyuan Wang\* Yizhou Yu†\* Tien-Tsin Wong‡ Chun Chen\* Ying-Qing Xu§

\*Zhejiang University †University of Illinois at Urbana-Champaign

‡The Chinese University of Hong Kong §Microsoft Research Asia



**Figure 1:** An example of image color theme enhancement using our method. **Left:** Original image; **middle:** recolored result with the “nostalgic” color theme; and **right:** recolored result with the “lively” color theme.

### Abstract

It is often important for designers and photographers to convey or enhance desired color themes in their work. A color theme is typically defined as a template of colors and an associated verbal description. This paper presents a data-driven method for enhancing a desired color theme in an image. We formulate our goal as a unified optimization that simultaneously considers a desired color theme, texture-color relationships as well as automatic or user-specified color constraints. Quantifying the difference between an image and a color theme is made possible by color mood spaces and a generalization of an additivity relationship for two-color combinations. We incorporate prior knowledge, such as texture-color relationships, extracted from a database of photographs to maintain a natural look of the edited images. Experiments and a user study have confirmed the effectiveness of our method.

**CR Categories:** I.4.3 [Image Processing and Computer Vision]: Enhancement; I.4.6 [Image Processing and Computer Vision]: Segmentation—Region Partitioning; I.4.10 [Image Processing and Computer Vision]: Representation—Statistical

**Keywords:** Color Theme, Color Optimization, Histograms, Soft Segmentation, Texture Classes

### 1 Introduction

Designers and photographers carefully conceive the color composition of their posters or images to convey desired color themes.

\*This work was done when Baoyuan Wang was an intern at Microsoft Research Asia

Psychological studies confirm the strong associations between colors and semantic themes. For instance, “graceful” is often associated with a combination of pinkish and purple colors while “sad” is typically defined using a combination of blue and grey colors. In general, a color theme is a template of colors and an associated verbal description. Given a color theme, our goal is to perform *color theme enhancement* of an image, which seeks a global transformation of the colors in the original image so that it maintains the realism of natural images while the new color composition is perceptually close to a desired color theme. This paper develops techniques that can digitally enhance a desired color theme in an image through means of image processing.

Unfortunately, editing the color composition of an image to enhance a color theme can be a painful experience. Powerful commercial software, such as Photoshop, does not explicitly support color theme editing. A user may have to search for suitable image editing operations that can achieve a desired color theme through a large number of trials. Furthermore, such software typically maps the original colors to a new range of colors in a global manner, mimicking the effect of an optical color filter. Such an approach relinquishes control and may result in over-modification and unnatural results (e.g. leaves may be colored in blue).

On the other hand, directly forcing image regions to closely follow colors from a desired color theme may sometimes be over-aggressive and violate commonsense knowledge, especially when the color theme is dramatically different from the original colors in the image. For example, leaves may be colored in blue. As natural materials, such as grass and tree bark, are usually recognized via their textures, we propose to learn the relationships between texture classes and color histograms. With this texture-color knowledge, we can suppress over-aggressive coloring.

Hence, a desired approach should automate color editing according to a user-desired theme, while respecting natural textures and leaving sufficient control to users. In this paper, one of our major contributions lies in casting the problem of color theme enhancement as an optimization problem that searches for a new color composition of the input image that approximately matches the desired color theme while still making the image look natural. Our optimization

tries to balance among the colors of a desired theme, the colors of natural materials as well as user-defined color constraints.

Another contribution of this paper is the quantification of differences between an image and a color theme. Note that an image has a large number of pixels while a color theme is typically defined by a small template of colors and a vague verbal description such as “graceful” and “African painting”. To quantify the differences between the two, we use color mood spaces developed through psychophysical studies [Ou et al. 2004a]. Both the image and color theme are first mapped to points in a color mood space and their differences are computed as the distance between the points.

Our system can automatically modify the colors of input images once the user selects a desired color theme. To achieve flexible local control and facilitate color propagation, we divide the image into multiple soft segments. A user can incrementally refine the color assignment by scribbling on any interested local regions with desired colors. Our system treats such user-specified colors as soft constraints in the optimization. In the rest of this paper, we describe in detail our algorithms and demonstrate the effectiveness of our method with experimental results and a user study.

## 2 Background and Related Work

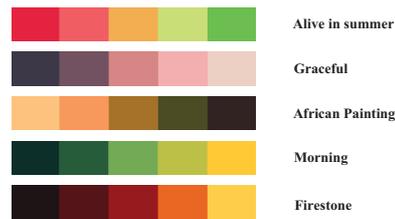
### 2.1 Related Work

**Color Editing and Transfer.** Much work has been performed on image color transfer and editing. Early work on color transfer [Reinhard et al. 2001] performs a statistical analysis to impose the color characteristics of one image onto another. When two images are not compatible, users may have to manually set swatches for each color region to define a match between them. Welsh et al. [2002] introduces a technique for transferring the chromatic information from a color image to a grayscale image. Similarly, the procedure was enhanced by allowing users to match areas of the two images with rectangular swatches.

To avoid this, Chang et al. [2005] proposed an automatic approach to stylize an image. This method captures the statistical characteristics within each image category, and color transfer is only performed between images within the same category. Recently, Piti et al. [2007] extended [Reinhard et al. 2001] by finding a linear mapping to minimize the displacement cost. Their solution is based on the Monge-Kantorovitch (MK) theory of mass transportation and can avoid odd results to a certain extent. Nevertheless, the quality of the results still relies on the compatibility between source and reference images.

Bae et al. [2006] and Lischinski et al. [2006] focus on effective tone adjustment through manipulating the luminance channel only while Cohen-Or et al. [2006] performs color adjustment according to harmonization rules. Recently, Shapira et al. [2009] proposed an interesting method to edit image appearances interactively using Gaussian mixture models (GMM). Though their editing results may potentially convey a certain mood or theme, their technique was not originally designed for steering the colors in a given image towards a specific color theme while our technique in this paper has been designed for image color theme enhancement.

**Edit Propagation.** Manually editing the appearances of images and videos is a labor intensive process even using commercial software such as Photoshop. Recently, intuitive stroke-based methods [Levin et al. 2004; Lischinski et al. 2006; Qu et al. 2006; Luan et al. 2007; Pellacini and Lawrence 2007; An and Pellacini 2008] have been proposed to simplify this process. In these methods, users need to supply edits by drawing scribbles in different regions.



**Figure 2:** A list of popular color themes from Adobe Kuler. **Left:** the color templates. **Right:** the associated verbal descriptions.

These edits automatically propagate to the rest of the image by exploiting the constraint that pixels with high affinities should receive similar edits. Among them, An and Pellacini [2008] develop a general and robust framework by efficiently approximating the all-pairs affinity matrix. In this paper, we adapt this edit propagation method to obtain a soft image segmentation. More recently, an acceleration technique for affinity-based edit propagation has been developed in [Xu et al. 2009] using adaptive clustering based on k-d trees.

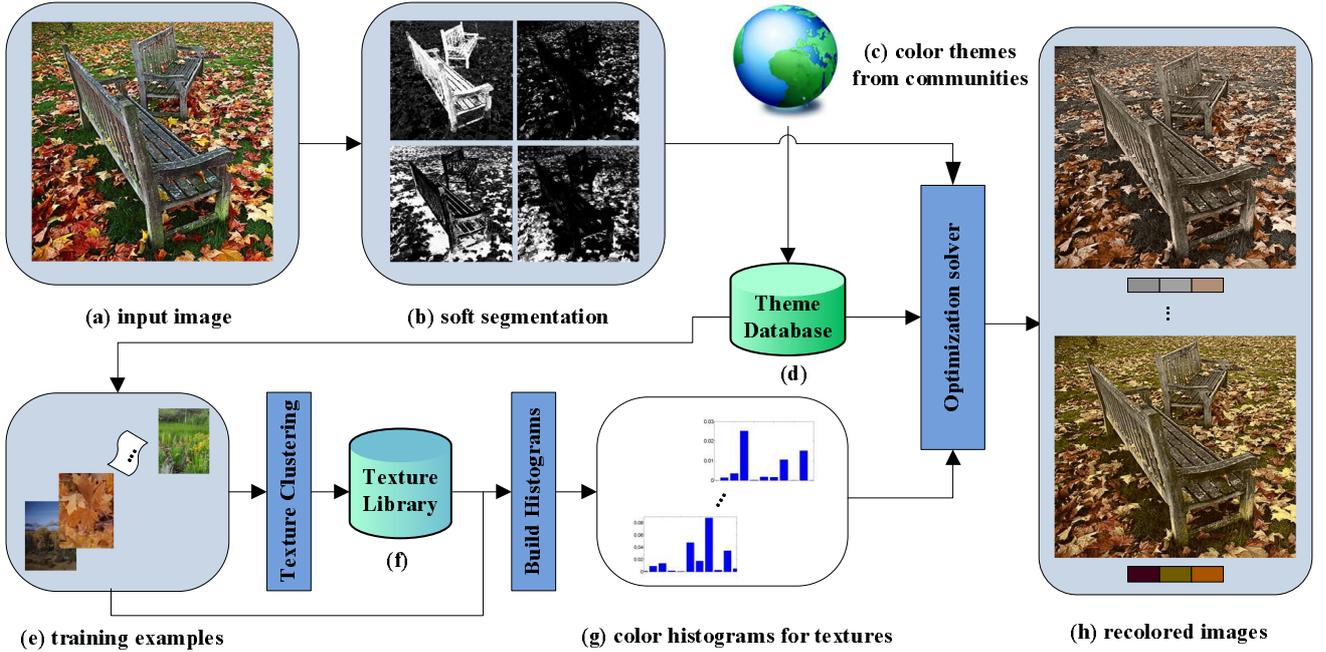
### 2.2 Color and Mood

Research on relationships between colors and their evoked mood or emotions has a long history. Much work [Sato et al. 2000; Ou et al. 2004a] has also been devoted to the development of quantitative color mood models. Such models give specific formulae to calculate mood scales from color appearance attributes such as luminance, hue and chroma. In particular, Ou et al. [2004a] proposes a three-dimensional color mood space which agrees well with findings in [Sato et al. 2000]. In Ou et al. [2004a]’s psychological experiments, 20 colors on 10 color mood scales, such as warm-cool, heavy-light and tense-relaxed, were assessed by a number of observers from different countries. Three major factors were extracted from these ten different scales using Principle Component Analysis (PCA). Each factor corresponds to one of the first three principal components and is partially associated with multiple mood scales. For example, the second factor is partially associated with the following mood scales: hard-soft, heavy-light and masculine-feminine. The three extracted factors are named activity, weight, and heat, respectively, and form the three coordinate axes of their three-dimensional color mood space. Ou et al. [2004a] further proposes empirical formulations that quantitatively connect the color mood space with the CIELAB color space (see Appendix for more details). In this paper, we rely on such a color mood space to compute the differences between an image and a color theme.

In [Ou et al. 2004b], three similar factors have also been identified by PCA for color pairs, indicating a coherent color mood space for a single color and two color combinations. More importantly, the *additivity relationship* [Hogg 1969] was found to hold for color moods evoked by two-color combinations. This means the color mood coordinates of a color pair in the three-dimensional color mood space can be predicted by simply averaging the color mood coordinates of individual colors in the pair. Since pixel colors in an image may form more than two color clusters, inspired by [Ou et al. 2004b], we propose to generalize the additivity relationship to color combinations with more than two individual colors in this paper.

## 3 Overview

The framework of our system is illustrated in Figure 3. It consists of an offline phase (lower row) and a runtime phase (upper row). Prior knowledge of texture-color relationships is obtained in the offline phase and such relationships are further converted to probability



**Figure 3:** The overall pipeline of our framework. Each sub-image in (b) is an influence map of a soft segment. Each soft segment finds the most relevant texture class in (f) and adopt the corresponding color histogram in (g). The histograms in (g) are further converted into continuous probability density distributions using Gaussian Mixture Model(GMM). The user needs to select a desired color theme from (d), then our optimization solver takes the prior knowledge (color probabilities), the desired color theme and the image segments into consideration to generate final recolored images in (h).

density functions (Section 4). Since our color theme representation (3- or 5-color) is compatible with industrial practices, we allow the usage of popular color themes available from online communities, such as Adobe Kuler.

During the runtime phase, we first compute  $K$  soft segments ( $K$  usually is set from 5 to 12, depending on the images). Mathematically, every pixel  $t$  in an image is associated with a  $K$ -dimensional vector,  $\vec{P}_t = (P'_{t1}, \dots, P'_{tK})$ ,  $\sum_{i=1}^K P'_{ti} = 1$ , where  $P'_{ti}$  is the probability of pixel  $t$  belonging to segment  $i$ . The probability vectors over all pixels of a given image define a new image with  $K$  channels. We adapt the edit propagation method in [An and Pellacini 2008] to obtain  $K$  soft segments starting from seed patches, which can be user-specified (via scribbling) or automatically generated based on texture labeling (described in Section 6). The colors of the resulting  $K$  soft segments form the unknown variables in the subsequent optimization.

The user starts an editing session by choosing a desired color theme from a list (obtained from existing literature or online communities). Then our goal is to recolor each soft segment so that the overall color composition of the recolored image approximately matches the chosen color theme while retaining the natural appearance of the original image. This is achieved via an optimization that tries to balance between the colors of a desired theme and the colors of natural materials. For an image segment with texture patterns, we use the texture descriptor of the segment as a key for searching the best matched texture class in our texture library (obtained in the offline phase), and use the corresponding color probability density function to constrain the optimization. Users can fine-tune the color composition by scribbling on selected image segments. User-specified colors are regarded as soft constraints in the optimization. The optimization is detailed in Section 5.

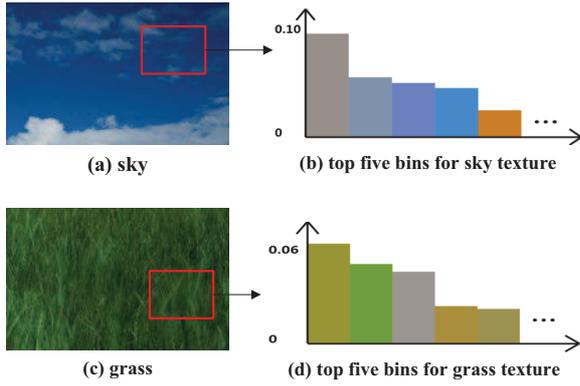
After optimization, the  $i$ -th soft segment is assigned with a target

average color  $c_i^n$ . Hence, the  $i$ -th channel (which corresponds to the  $i$ -th soft segment) of a pixel can be recomputed by multiplying the ratio  $c_i^n/c_i^o$ , where  $c_i^o$  is the original average color of the soft segment. The final color of a pixel is computed by linearly combining colors from all channels. Specifically, the final color of pixel  $t$  is equal to  $\sum_{i=1}^K P'_{ti}(c_i^n/c_i^o)c_t$ , where  $c_t$  is the original color of pixel  $t$ .

## 4 Data-Driven Prior Knowledge Extraction

**Color Theme Based Image Labeling** We start with building two databases in an offline phase, one is a color theme database and the other is an image database. The theme database consists of representative color themes from existing literature and online communities such as Adobe Kuler (Figure 2), and the image database was built by randomly choosing a subset of tens of thousands of color images downloaded from Flickr. We then use the theme database to label each image from the image database. Specifically, we set up discrete histogram bins by quantizing colors from these images in the CIELAB color space using the K-means algorithm. A normalized color histogram was computed for every image. Since a color theme is typically associated with a template of 3 or 5 colors, we use the colors corresponding to the 3 or 5 most frequent bins in a histogram to form a color composition of the corresponding image, and compute the distance between this color composition and every chosen color theme based on Earth Mover’s Distance [Rubner et al. 1998]. Every image in the database is labeled with the closest color theme. In this way, each adopted color theme has at least 500 automatically labeled images.

**Texture-Color Relationship** Textures of natural materials are usually highly correlated with certain color combinations. For example, the “grass” texture is more likely to be associated with the



**Figure 4:** Color histograms for textures “sky” and “grass” among all images labeled with the same color theme. Their top five color bins are visualized by setting  $L=60$  (as luminance is excluded during indexing).

green color than the blue color. Furthermore, such color association may vary across different color themes. For example, in a lively theme, the grass texture may be most frequently associated with the green color while in another bleak theme, it may be more frequently associated with the yellow color. For these reasons, we build color histograms indexed by both color themes and texture classes.

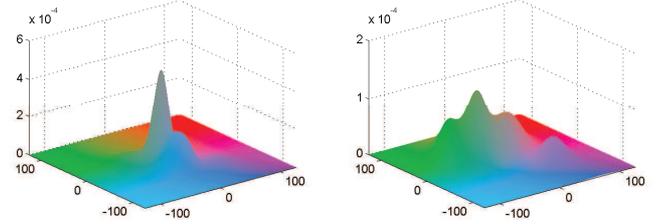
We start with the extraction of a universal texture library. 200 classes of textures are extracted from images across all adopted color themes. We achieve this by first dividing each image into small patches using the method in [Felzenszwalb and Huttenlocher 2004]. Then we calculate the mean and standard deviation of the pixelwise Gabor filter (scale=4, rotation=6) responses within every patch [Manjunath and Ma 1996]. This gives rise to a 48-dimensional texture descriptor for every patch. We cluster these texture descriptors using the K-means algorithms [Leung and Malik 2001]( $K=200$ ). The cluster centers define the texture library. During the indexing of color histograms, only the chrominance channels ( $a^*$  and  $b^*$ ) of the CIELAB space are involved, as the luminance channel is mainly due to the lighting condition instead of the body color of texture. We empirically set up 100 histogram bins by running vector quantization over this 2D subspace only. The number of histogram bins was determined arbitrarily, and a larger number is likely to further improve the results. As the correlation between chrominance values and texture classes may also be dependent on the color theme, for each set of images labeled with the same color theme, we compute a family of color histograms, one for each class of texture. Figure 4 shows two typical histograms for sky and grass textures.

**Continuous Probability Density Estimation** Since much of the statistical knowledge we gathered from the example images is expressed as color histograms with discrete bins, it would be helpful to convert these histograms into continuous probability density functions in the following section. We still represent colors in the CIELAB color space. The set of histogram bins for each class of textures is denoted as  $U$ , and each histogram bin represents a combination of the  $a^*$  and  $b^*$  channels. For an image segment  $s_i$ , we first identify the most likely texture class for  $s_i$ , which is denoted as  $\mathcal{T}_i$ . Let us further use  $\mathcal{H}_{\mathcal{T}_i}$  to denote the pre-computed color histogram indexed by  $\mathcal{T}_i$  and the color theme currently selected by the user, and use  $h_j \in \{1, 2, \dots, |U|\}$  to denote the label of a color bin corresponding to a base color  $(a_j^*, b_j^*) \in U$ . The frequency of the base color can be expressed as  $\mathcal{H}_{\mathcal{T}_i}(h_j)$ . To estimate the underlying probability density, we construct Gaussian Mixture Models

(GMMs) from discrete histograms. For any color  $\vec{c} = (l^*, a^*, b^*)$  ( $l^*$ ,  $a^*$  and  $b^*$  are the three channels in the CIELAB space respectively), its probability density in texture class  $\mathcal{T}_i$  is expressed as follows.

$$\mathcal{H}'_{\mathcal{T}_i}(\vec{c}) = \sum_{j=1}^{|U|} \mathcal{H}_{\mathcal{T}_i}(h_j) \frac{1}{2\pi\sigma_j} \exp\left(\frac{-(a_i^* - a_j^*)^2 - (b_i^* - b_j^*)^2}{2\sigma_j^2}\right), \quad (1)$$

where  $\sigma_j$  is estimated as the average distance between  $(a_j^*, b_j^*)$  and the base colors of its three nearest color bins. Figure 5 shows examples of continuous models based on this construction.



**Figure 5:** Continuous probability density functions for sky (Left) and grass (Right) textures respectively, converted from discrete histograms

## 5 Color Optimization

We cast image color theme enhancement as an optimization problem, which tries to strike a balance among the selected color theme, the precomputed texture-color relationships as well as color constraints. Suppose there are  $K$  soft segments in a given image. The color composition of this image is optimized by searching for an optimal state for every segment so that the edited image exhibits the desired theme. The energy function is defined as follows.

$$E = \alpha E_1 + \beta E_2 + \gamma E_3, \quad (2)$$

where

$$E_1 = \sum_{i=1}^K \psi(\vec{c}_i), \quad E_2 = - \sum_{i=1}^K \log(\mathcal{H}'_{\mathcal{T}_i}(\vec{c}_i)),$$

$$E_3 = \left\| \frac{1}{N} \sum_{t=1}^N \mathcal{F}(\vec{c}_t) - \frac{1}{m} \sum_{k=1}^m \mathcal{F}(\vec{c}_k^*) \right\|^2,$$

$$\psi(\vec{c}_i) = \begin{cases} \|\vec{c}_i - \vec{c}_i'\|^2, & \text{a scribble over segment } s_i \\ \|\vec{c}_i - \vec{c}_i^0\|^2, & \text{otherwise} \end{cases}$$

and  $\vec{c}_i = \vec{c}_{i_{\max}}$ ,  $i_{\max} = \arg \max_i \mathcal{P}_{t_i}$ .

$E_1$  defines the first energy term which indicates how well soft color constraints are satisfied.  $\vec{c}_i$  denotes the new color of segment  $s_i$  that needs to be optimized. It has three channels.  $\vec{c}_i^0$  denotes the average color of a representative patch of  $s_i$  in the original image while  $\vec{c}_i'$  denotes the color of a scribble. We take these two types of colors as soft constraints for scribbled and non-scribbled segments respectively. An important reason why we take the original colors of non-scribbled segments as soft constraints is that we would like to alter the original image as little as possible but still significantly enhance the desired color theme.

Color distance is computed using the Euclidean distance.  $E_2$  incorporates color preferences encoded in the underlying probability

density function  $\mathcal{H}'_{\mathcal{T}_i}$  to ensure naturalness and realism. We take the negative logarithm of the probability density  $\mathcal{H}'_{\mathcal{T}_i}(\vec{c}_i)$  to penalize colors that are not probable in the example images.

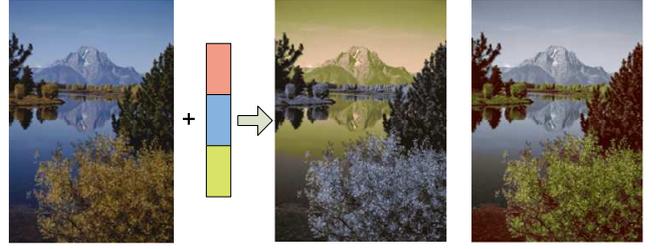
$E_3$  is a least-squares energy term, which steers the color composition of the image towards the desired color theme. Note that an image has a large number of pixels while a color theme is typically defined by a small template of colors. They are not directly comparable. To overcome this problem, we rely on a three-dimensional color mood space. Both the image and color theme are first mapped to three-dimensional points in this color mood space and their difference is then computed as the Euclidean distance between the points. In this paper, we adopt the three dimensional color mood space (*activity*, *weight*, and *heat*) developed in [Ou et al. 2004a]. Note that our optimization formulation does not prevent us from using other color mood spaces.

The first part of  $E_3$  maps the edited image to the color mood space. Here  $N$  denotes the number of pixels in the image. For computational efficiency, instead of linearly combining colors from multiple segments, the current color  $\vec{c}_i$  at pixel  $t$  is approximated by the color of the most probable segment at  $t$ .  $\mathcal{F}$  transforms a color from the CIELAB color space to the color mood space. Detailed formulations of  $\mathcal{F}$  can be found in the appendix. By generalizing the additivity relationship in [Ou et al. 2004b], we calculate the color mood of an image by averaging the color mood of every pixel. The second part of  $E_3$  maps the chosen color theme to the color mood space. By transforming the colors in the color theme to the color mood space and taking average of the transformed results, we are able to represent a color theme as a 3D vector. Here  $m$  is the number of distinct colors in the color theme, and  $\vec{c}_k^*$  represents a specific color in the color theme.  $m = 3, 5$  in our experiments.

$\alpha$ ,  $\beta$  and  $\gamma$  are the weighting coefficients of  $E_1$ ,  $E_2$  and  $E_3$  respectively. In general, we fix one of them and adjust the other two. For example, in our experiments, we set  $\gamma$  to 1 and  $\beta$  10-50 times as large as  $\alpha$ . These three terms altogether try to make the recolored image perceptually consistent with the desired color theme while ensuring naturalness and realism. Figure 6 demonstrates the importance of the optimization. Figure 7 shows the importance of the energy term  $E_3$  in steering the image towards the color theme. Projecting an image to the color mood space discards much information, therefore,  $E_3$  has a low discriminative ability. There exists a large number of distinct color compositions that can minimize  $E_3$ . However, our goal is to find the optimal composition that also makes the image look natural and realistic which cannot be guaranteed only through  $E_3$ . By incorporating  $E_1$  and  $E_2$  into the energy function, we can narrow down our choice.

Figure 8 shows the importance of prior knowledge in enhancing the realism of the result for a texture image. For images with rich textures, automatic results without any user scribbles are often adequate. For textureless image segments, however, prior knowledge can not provide enough constraints. To avoid over-modification and unnatural results, we adopt the average color of the segments in the original image as soft constraints (Figure 9 (b) and (c)). We also support user scribbles to facilitate local adjustment. Users can give a rough hint of their preferred colors by directly scribbling on the regions (Figure 9 (d) and (e)).

In practice, we minimize the energy function in Equation (2) using sequential quadratic programming (SQP) [Lawrence and Tits 1996]. SQP is well suited for optimizing differentiable objective functions subject to linear and/or nonlinear constraints as well as simple bounds on the variables. In our problem, the color channels we wish to optimize have both lower and upper bounds. The actual bounds we use are  $[0, 100]$  for the  $L$  channel and  $[-126, 126]$  for the  $A$  and  $B$  channels. We adopt the FSQP [Lawrence et al.



**Figure 8:** Left: an input image and a color theme. Middle: result image without  $E_2$ . Right: result image with  $E_2$ .

1997] package in our implementation. To apply FSQP, we need to start with an initial feasible solution. We simply generate such an initial solution by sampling near the original image colors for non-scribbled segments and sampling near the scribble colors for scribbled segments. Experiments show that such a selection strategy makes SQP converge more quickly than a purely random initial solution.

### 5.1 Automatic Theme Driven Scribbles

To effectively steering an image towards a given color theme, we automatically generate color scribbles over a subset of image segments. There is an automatic scribble associated with every color from the template of the target color theme. We still need to decide which image segment an automatic scribble should be drawn over. Suppose there are  $K$  segments ( $K > m$ ). Denote  $\vec{c}_i^*$  ( $i \in 1, 2, \dots, m$ ) as one of the colors in the theme,  $w_i = \frac{1}{N} \sum_{t=1}^N \delta_{i, \arg \max_j p_{tj}}$  where  $\delta_{ij} = 1$  when  $i = j$ , and  $\delta_{ij} = 0$  otherwise. Then the segment where  $\vec{c}_i^*$  should be scribbled over is determined as follows.

$$\arg \max_j w_j \vec{c}_i^{*T} \vec{c}_j^0, \quad (3)$$

where  $\vec{c}_j^0$  is the average color of segment  $s_i$  in the original image. Intuitively, a color in the theme is assigned to a segment with a large area and a similar average color. We choose an appropriate segment for each color in the theme one by one using this greedy method. Note that each segment can be chosen only once in this process.

## 6 Soft Segmentation

Our soft segmentation mainly follows the method in [An and Pellacini 2008] except for an extra step at the end to convert edit propagation results to probability channels. Suppose there are  $K$  seed areas, each of which is assigned a distinct scalar number. The seeds along with their assigned numbers serve as edits for the algorithm in [An and Pellacini 2008], which propagates these edits and generates an influence value for every pixel. Suppose the scalar numbers in the  $K$  edits are  $\{v_1, v_2, \dots, v_K\}$  ( $0 \leq v_j \leq 1$ ) and the influence value at pixel  $t$  is  $f_t$  ( $0 \leq f_t \leq 1$ ). We define the probability vector for pixel  $t$  as follows.

$$\mathcal{P}'_{ti} = \exp\left(\frac{(f_t - v_i)^2}{\alpha^2}\right) / \sum_{j=1}^K \exp\left(\frac{(f_t - v_j)^2}{\alpha^2}\right), \quad i = 1, \dots, K \quad (4)$$

where  $\alpha$  is used to control the “softness” of the segmentation. When  $\alpha$  approaches zero, the largest channel approaches 1 while all other channels approach 0, which becomes a “hard” segmentation. When  $\alpha$  approaches infinity, all channels approach  $1/K$ . An appropriate  $\alpha$  should keep one of the channels dominant ( $\mathcal{P}'_{ti} > 0.5$ ) while



**Figure 6:** **Left:** an input image and a color theme. **Middle:** image generated by the greedy initial assignment in section 5.1, without further optimization. **Right:** final image with further optimization.



**Figure 7:** **Left:** an input image and a target color theme. **Middle:** result image without  $E_3$ . **Right:** result image with  $E_3$ .

also maintaining smooth transitions across segment boundaries. We typically set  $\alpha$  between 0.2 and 0.5.

**Automatic Generation of Seed Areas** For complex images, it is tedious for a user to scribble all seeds for soft segmentation. Thus, we provide an option to generate seed areas automatically. We over-segment the input image into many small patches [Felzenszwalb and Huttenlocher 2004] and compute the average color and Gabor-wavelet texture descriptor for every patch. The basic idea for automatic seeding is to label every patch with a texture class so that nearby patches sharing similar appearances should be labeled with “similar” textures. Here “similar” means either the same texture class or different texture classes with similar color histograms. We build a graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ , where  $\mathcal{V}$  is the set of  $M$  nodes corresponding to the set of patches and  $\mathcal{E}$  is the set of edges connecting neighboring nodes. Obviously, this is a graph labeling problem, where every node is associated with an unknown label that needs to be solved. The label refers to one of the texture classes we have.

We define the following energy function for texture labeling.

$$E' = \sum_{i \in \mathcal{V}} E_4(l_i) + \delta \sum_{(i,j) \in \mathcal{E}} E_5(l_i, l_j), \quad (5)$$

$$E_4(l_i) = \|\vec{T}_i^p - \vec{T}_{l_i}^t\|^2, \quad (6)$$

$$E_5(l_i, l_j) = \exp\left(-\frac{\|\vec{\mu}_i - \vec{\mu}_j\|^2}{\eta^2}\right) \|\vec{\mathcal{H}}_{l_i} - \vec{\mathcal{H}}_{l_j}\|^2, \quad (7)$$

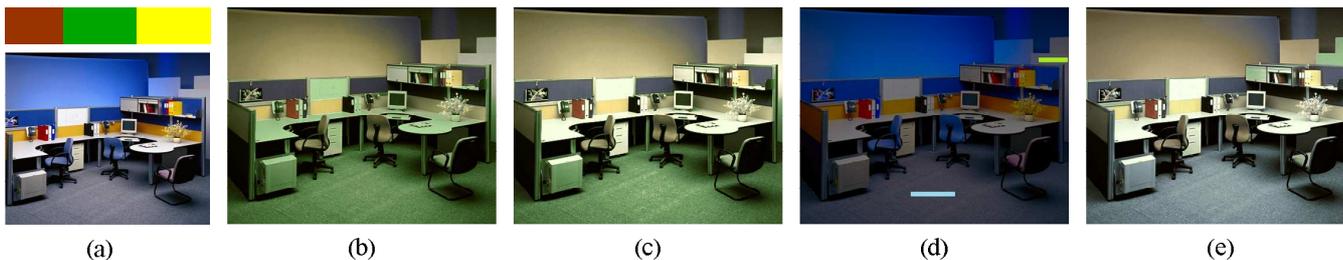
where  $l_i$  is the texture label for patch  $i$ ,  $\vec{T}_i^p$  represents the texture descriptor of patch  $i$ ,  $\vec{T}_{l_i}^t$  represents the descriptor of texture class  $l_i$ ,  $\vec{\mu}_i$  represents the average color of patch  $i$ , and  $\vec{\mathcal{H}}_{l_i}$  represents the color histogram corresponding to texture class  $l_i$ .  $\delta$  controls the relative importance of the two energy terms, and  $\eta$  controls the smoothness of the labeling result in local regions. All the histogram distances are determined by L2 norm.

The first term in (5) means it is more likely for a patch to be labeled with a texture class if the texture descriptors of the two are similar. The second term means the smaller the difference between the average colors of two neighboring patches, the more likely they are labeled with the same texture class or different texture classes with similar color histograms.

**Loopy Belief Propagation** Belief propagation (BP) is an inference algorithm for graph models, such as Bayesian networks and Markov random fields (MRF). Nevertheless, BP has been frequently used for minimizing energy functions defined on a graph as long as the energy function only involves pairwise interactions among the graph nodes. For a graph without any loops, belief propagation guarantees to reach the globally optimal solution in a finite number of iterations. For a graph with loops (e.g. the loops existing in  $\mathcal{G}$  for our case), BP can still be applied using loopy belief propagation (LBP) [Yedidia et al. 2003]. Studies [Freeman et al. 2000] show that BP often achieves a very good approximation of the global solution even for graphs with thousands of loops. Since (5) only involves pairwise interactions between neighboring patches, LBP can be readily applied to minimize (5).

Once we have obtained the texture labeling result, patches with the same texture label are grouped into the same region. Note that the number of regions is influenced by  $\eta$ . A larger  $\eta$  usually gives rise to a smaller number of regions. In our experiments, we typically set  $\delta = 100$  and  $\eta = 50$ .

The above labeling process might lead to over-segmentation. A final step in our algorithm merges any pair of resulting regions if the difference between their average colors is sufficiently small. Within each final region, we choose the patch with the largest belief value as the seed area of an image segment. An example with automatically generated seeds can be found in Figure 10.



**Figure 9:** (a) input image and color theme; (b) recolored image without any color constraints. (c) recolored image with original image colors as soft constraints. Note that the color of the tabletop and whiteboard have been adjusted from green to white, indicating the usefulness of this constraint. (d) user scribbles. (e) recolored image with scribble colors as additional soft constraints.



**Figure 10:** Left: Patches serve as graph nodes for texture labeling. Middle: Labeling result with color-coded regions. Right: Seed patches after the final merging step are indicated in white.

## 7 Results and Analysis

**Performance** We have fully implemented our algorithms and successfully tested them on a variety of images from many different sources, including the Corel Image Database and Flickr. Some of the color theme editing results can be found in Figures 1, 12 and the supplementary materials. If the image size is  $1024 \times 1024$ , and there are up to 10 soft segments, color optimization in Section 5 typically converges around 100 iterations and takes 0.25 seconds on a PC with an Intel Core 2 Duo 2.6GHz processor and 2GB RAM, which allows real-time interactions. We would like to emphasize that many factors contribute to the overall impression a human viewer has about an image. The color composition of the image is just one of them. However, in this section, we only evaluate the color theme of an image.

**Comparison with Color Transfer** To enhance a color theme in an image using color transfer, one must provide a reference image with the desired color theme. Color transfer by histogram matching is the most straightforward method to accomplish this task. However, this simple method suffers from unnatural colorization and artifacts (such as green sky in Figure 11(b)). The MK linear mapping method [Piti and Kokaram 2007] can avoid this problem to a certain extent. However, it still requires highly consistent global color statistics between the source and reference images; otherwise, artifacts may occur (such as the purple color on the mountain in Figure 11(c)). It is extremely hard to locate a reference image with both the desired color theme and the same color statistics as the source image. Furthermore, color transfer is a completely automatic process that does not allow any user intervention and local control. Users often need to try multiple reference images before obtaining a marginally acceptable result. In contrast, our approach takes advantage of color themes available from online communities (e.g. Adobe Kuler), saving the user’s effort in finding suitable reference

images. Besides, soft image segmentation and color scribbling facilitate local color editing.

**Comparison with Simple Filtering** Existing filtering-based approaches (e.g. using Photoshop) can be performed with or without region segmentation. Without segmentation, the user would have to apply a single color filter to the entire image, which can probably produce certain color themes at the expense of reduced quality and diversity of the colors in the original image, i.e. the color of certain regions may be changed in a dramatic way to violate commonsense knowledge (e.g. grass region is colored in purple). On the other hand, with region segmentation, the choice of color filters for individual regions still has to be manually determined. Novice users may have to try a large number of different filter settings in order to achieve a desired color theme. In contrast, our system initializes and optimizes color automatically according to the chosen color theme and texture information, while letting the user set color constraints in individual regions. In summary, our system achieves a global transformation of the image color composition, while ensuring that each segment of the image only undergoes reasonable changes.

**User Study** We have carried out a user study to evaluate the consistency between the color composition of the recolored image and the user desired color theme. For every chosen color theme, a paired T-test was conducted to evaluate the difference between the original and the recolored images.

We invited 25 participants (14 females and 11 males) with ages ranging from 18 to 45. Among them, 5 are professional designers. All of them have normal vision. We chose 20 test images, including both indoor and outdoor scenes. They were divided into 7 groups each of which needs to be recolored to match one of 7 chosen color themes (Table 1). The number of images in each group is {3, 3, 2, 4, 2, 3, 3}, respectively. These 25 participants were also divided into two groups. We randomly chose 5 participants to form the first group and asked each of them to perform theme enhancement on 4 test images using our system. Before the test, a 10-minute training session was conducted to give them instructions and make them familiar with the system environment. Each participant in this group should complete the task as soon as possible, but was encouraged to edit the image until satisfaction.

Once the first group had completed their task, we collected 20 original images and 20 recolored ones. These 40 images were sorted in a random order, and shown one by one to participants in the second group (20 members). Participants rated their confidence on the color theme in each image according to their first impression, and they were not allowed to go back to previously viewed images. Specifically, the color composition of an image was compared with each of the 7 color themes, and an overall confidence



**Figure 11:** Comparisons with histogram matching and color transfer. The source image is shown in Figure 7 Left. (a) a reference image and its corresponding color theme, (b) result image generated by histogram matching, (c) result image generated by MK-based color transfer [Piti and Kokaram 2007], (d) theme enhancement result by our method.

score for each theme was measured by considering two criteria at the same time: (1) how good a recolored image conforms to the color theme; and (2) the visual quality (naturalness) of the recolored image. In short, if a participant feels that the recolored image looks natural and also matches the color theme, he/she should give a high confidence score.

Confidence is quantified into 101 discrete levels ranging from 0 to 100. Every participant performed the same task in a single room, on a computer with a 19-inch monitor (1280×1024 resolution).

Thus, we collected two sets of sample confidence values for the original and recolored images, respectively, for each theme. A paired T-test was performed on these samples to check if  $\mu_a$  is significantly smaller than  $\mu_b$  using the following hypotheses

$$H_0 : \mu_a \geq \mu_b,$$

$$H_1 : \mu_a < \mu_b,$$

where  $\mu_a$  represents the mean confidence value of the original image, while  $\mu_b$  represents the mean confidence value of the recolored image. Hypothesis  $H_0$  means that the original image is more consistent with a given theme, which means our recolorization fails to enhance and express the theme.

Table 1 shows all 7 paired T-test results. Note that all two tailed P values are less than 0.05, and all T values are negative, indicating that  $H_0$  is rejected with statistical significance while  $H_1$  is accepted. This concludes that our algorithm has significantly enhanced the desired color themes.

**Table 1: Paired T-test Results ( $\alpha = 0.05$ )**

No.	Theme	T	P(two-tail)
1		-2.716	.000865
2		-9.309	.000000
3		-2.916	.005843
4		-8.753	.000000
5		-3.135	.003255
6		-11.703	.000000
7		-9.326	.000000

In addition, participant feedback shows that our system is easy to use and very intuitive to recolor an image. For example, tests from the first group show that our system can produce the desired result within half a minute. One user from the first group particularly liked the scribbling support for direct local adjustment. Two designers gave us valuable comments on the user interface. We are considering these and other suggestions for improving the user experience.

## 8 Conclusions and Discussion

We have presented an example-based method for steering the color composition of an image towards a desired color theme impression. We formulate our goal as a unified optimization that simultaneously considers a desired color theme, texture-color relationships as well as color constraints. Experiments and a user study have confirmed the effectiveness of our method.

**Limitations** In previous psychological studies [Hogg 1969; Ou et al. 2004b], the additivity relationship has been found to hold for color moods evoked by two-color combinations. To compute the color mood of an image which may have multiple segments, we have generalized this additivity rule to multiple color combinations. We have found our color optimization algorithm partially based on this rule can steer the color composition of an image towards a desired color theme. Nevertheless, a thorough psychological study is necessary to fully validate our generalization.

Although texture-color relationships learned from examples have proven to be useful in reducing user-supplied color constraints, color scribbles are often necessary over textureless regions. Likewise, out-of-focus image segments, whether they have texture or not, impose a challenge to our method because they appear blurred. In such cases, color inference based on texture-color relationships tend to generate random results. This is shown in Figure 13, where the out-of-focus green leaves have been recolored purple.



**Figure 13:** A failure case for out-of-focus regions. **Left:** an input image. **Right:** the recolored result.

## Acknowledgements

We would like to thank Chen Zhao for her advice on user study design and the anonymous reviewers for their valuable suggestions. Thanks also go to John Wright for video dubbing, Matt Callcut for proofreading, and all participants in our user study for making this paper possible. This work was partially supported by National Science Foundation (IIS 09-14631), National Natural Science Foundation of China (60728204/F020404), and Hong Kong Research

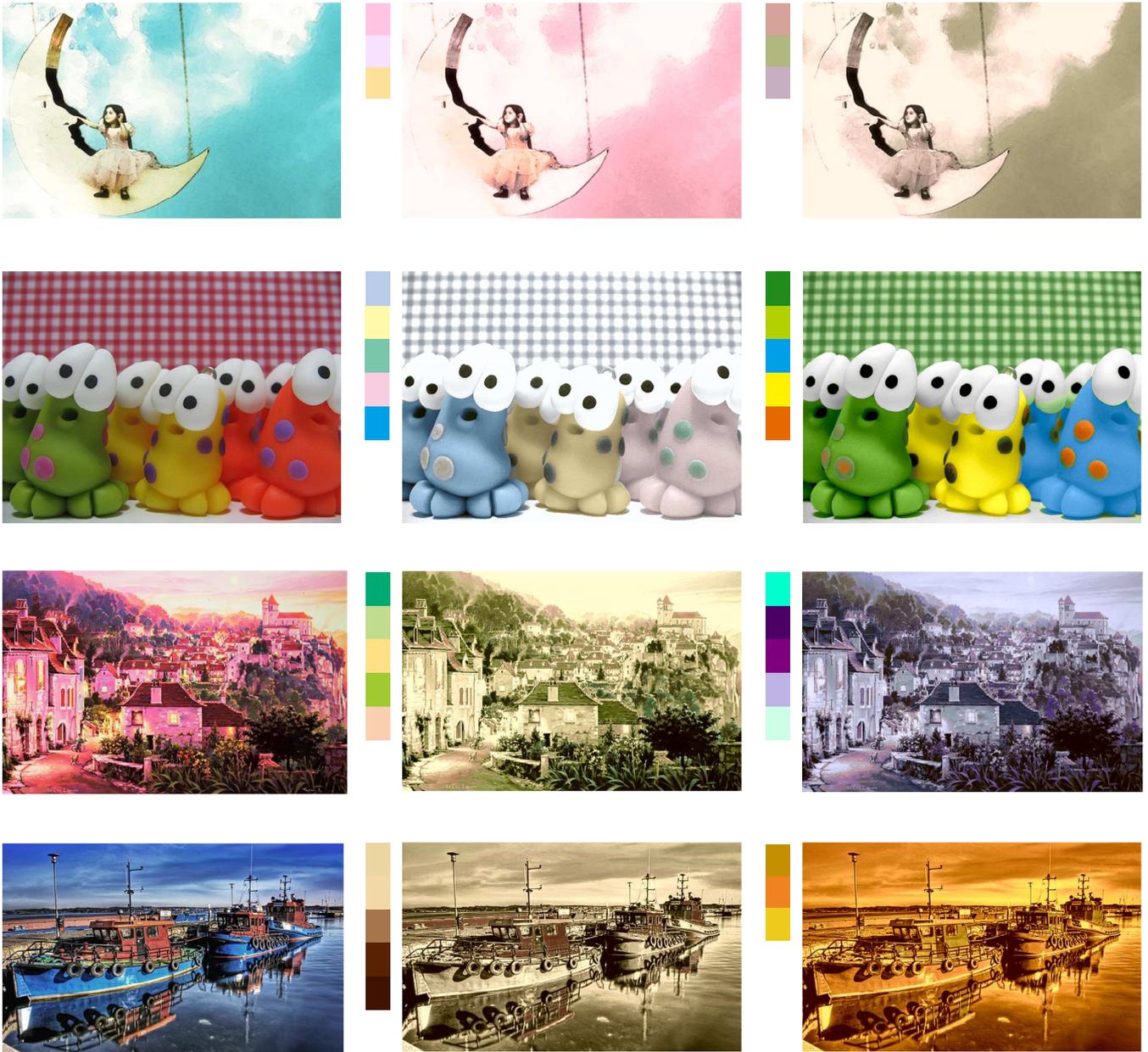


Figure 12: A variety of images and their recolored ones with different color theme enhancements. The first column shows the original images.

Grants Council under General Research Funds (CUHK417107).

## References

- AN, X., AND PELLACINI, F. 2008. Approp: all-pairs appearance-space edit propagation. *ACM Trans. Graph.* 27, 3, 40.
- BAE, S., PARIS, S., AND DURAND, F. 2006. Two-scale tone management for photographic look. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, 637–645.
- CHANG, Y., SAITO, S., UCHIKAWA, K., AND NAKAJIMA, M. 2005. Example-based color stylization of images. *ACM Trans. Appl. Percept.* 2, 3, 322–345.
- COHEN-OR, D., SORKINE, O., GAL, R., LEYVAND, T., AND XU, Y.-Q. 2006. Color harmonization. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, 624–630.
- FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. 2004. Efficient graph-based image segmentation. *Int. J. Comput. Vision* 59, 2, 167–181.
- FREEMAN, W. T., PASZTOR, E. C., AND CARMICHAEL, O. T. 2000. Learning low-level vision. *Int. J. Comput. Vision* 40, 1, 25–47.
- HOGG, J. 1969. The prediction of semantic differential ratings of color combinations. *J Gen Psychol* 80, 141152.
- LAWRENCE, C. T., AND TITS, A. L. 1996. Nonlinear equality constraints in feasible sequential quadratic programming. *Optimization Methods and Software* 6, 265–282.
- LAWRENCE, C. T., ZHOU, J. L., AND TITS, A. L. 1997. User's guide for cfsqp version 2.5: A c code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints. *Institute for Systems Research, University of Maryland, Technical Report TR-94-16r1 College Park, MD 20742*.
- LEUNG, T., AND MALIK, J. 2001. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision* 43, 1, 29–44.
- LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. *ACM Trans. Graph.* 23, 3, 689–694.
- LISCHINSKI, D., FARBMAN, Z., UYTTENDAELE, M., AND SZELISKI, R. 2006. Interactive local adjustment of tonal values. *ACM Trans. Graph.* 25, 3, 646–653.
- LUAN, Q., WEN, F., COHEN-OR, D., LIANG, L., XU, Y.-Q., AND SHUM, H.-Y. 2007. Natural Image Colorization. In *Rendering Techniques 2007 (Proceedings Eurographics Symposium on Rendering)*, J. Kautz and S. Pattanaik, Eds., Eurographics.
- MANJUNATH, B. S., AND MA, W. Y. 1996. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 8, 837–842.
- OU, L.-C., LUO, M. R., WOODCOCK, A., AND WRIGHT, A. 2004. A study of colour emotion and colour preference. part i: Colour emotions for single colours. *Color Research & Application* 29, 3, 232–240.
- OU, L.-C., LUO, M. R., WOODCOCK, A., AND WRIGHT, A. 2004. A study of colour emotion and colour preference. part ii: Colour emotions for two-colour combinations. *Color Research & Application* 29, 4, 292–298.
- PELLACINI, F., AND LAWRENCE, J. 2007. Appwand: editing measured materials using appearance-driven optimization. *ACM Trans. Graph.* 26, 3, 54.
- PITI, F., AND KOKARAM, A. 2007. The linear monge-kantorovitch linear colour mapping for example-based colour transfer. *Visual Media Production, 4th European Conference on Visual Media Production, London, UK*, 1–9.
- QU, Y., WONG, T.-T., AND HENG, P.-A. 2006. Manga colorization. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)* 25, 3, 1214–1220.
- REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. Appl.* 21, 5, 34–41.
- RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. 1998. A metric for distributions with applications to image databases. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, USA, 59.
- SATO, T., KAJIWARA, K., HOSHINO, H., AND NAKAMURA, T. 2000. Quantitative evaluation and categorising of human emotion induced by colour. *Advances in Colour Science and Technology* 3, 53–59.
- SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2009. Image appearance exploration by model-based navigation. *Comput. Graph. Forum* 28, 2, 629–638.
- WELSH, T., ASHIKHMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. *ACM Transactions on Graphics* 21, 3, 277–280.
- XU, K., LI, Y., JU, T., HU, S.-M., AND LIU, T.-Q. 2009. Efficient affinity-based edit propagation using k-d tree. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, ACM, New York, NY, USA, 1–6.
- YEDIDIA, J. S., FREEMAN, W. T., AND WEISS, Y. 2003. Understanding belief propagation and its generalizations. 239–269.

## A Color-Mood Transformation

A three-dimensional color mood space was introduced in [Ou et al. 2004a; Ou et al. 2004b]. The three coordinate axes of the space are called *activity*, *weight*, and *heat*. Empirical formulations of the transformation from the CIELAB color space to the proposed color mood space were also developed. Given a color  $\vec{c} = (L^*, a^*, b^*)$ , its corresponding point,  $\vec{e}$ , in the color mood space is a nonlinear function of  $\vec{c}$ ,  $\vec{e} = \mathcal{F}(\vec{c})$ , where  $\mathcal{F}$  is defined by the following three equations:

*activity*:

$$-2.1 + 0.06[(L^* - 50)^2 + (a^* - 3)^2 + (\frac{b^* - 17}{1.4})^2]^{1/2}, \quad (8)$$

*weight*:

$$-1.8 + 0.04(100 - L^*) + 0.45 \cos(h - 100^\circ), \quad (9)$$

*heat*:

$$-0.5 + 0.02(C^*)^{1.07} \cos(h - 50^\circ), \quad (10)$$

where  $L^*$  = CIELAB lightness;  $C^*$  = CIELAB chroma;  $h$  = CIELAB hue angle;  $a^*$ ,  $b^*$  are the CIELAB coordinates.