Computers & Graphics 36 (2012) 622–634

Contents lists available at SciVerse ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on CANS

Toward automatic and flexible concept transfer

Naila Murray [a,*], Sandra Skaff [b], Luca Marchesotti [c], Florent Perronnin [c]

[a] Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain
[b] Xerox Research Center Webster, Webster, NY, USA
[c] Xerox Research Centre Europe, Meylan, France

ARTICLE INFO

Article history:
Received 30 October 2011
Received in revised form
8 January 2012
Accepted 20 January 2012
Available online 25 February 2012

Keywords:
Color transfer
Concept transfer
Image representation
Gaussian Mixture Model
Earth Mover's Distance

ABSTRACT

This paper introduces a novel approach to automatic, yet flexible, image concept transfer; examples of concepts are "romantic", "earthy", and "luscious". The presented method modifies the color content of an input image given only a concept specified by a user in natural language, thereby requiring minimal user input. This method is particularly useful for users who are aware of the message they wish to convey in the transferred image while being unsure of the color combination needed to achieve the corresponding transfer. Our framework is flexible for two reasons. First, the user may select one of two modalities to map input image chromaticities to target concept chromaticities depending on the level of photo-realism required. Second, the user may adjust the intensity level of the concept transfer to his/her liking with a single parameter. The proposed method uses a convex clustering algorithm, with a novel pruning mechanism, to automatically set the complexity of models of chromatic content. Results show that our approach yields transferred images which effectively represent concepts as confirmed by a user study.

1. Introduction

The manipulation of the chromatic content of images has applications ranging from color correction in movie post-production to document or user profile personalization and graphic design. In particular, color transfer modifies image color content by transferring the chromatic characteristics of a target image to an input image. *Concept* transfer involves modifying an image such that it adheres to a concept, where a concept is typically represented by a collection of colors referred to as a *color scheme*. We use the term concept to refer to emotions or moods which are amenable to a chromatic representation. Examples of concepts include "romantic," "serene," and "cool." Therefore, concept-based color transfer modifies the chromatic appearance of an image or document such that the transferred version assumes a specific conceptual message. Fig. 1 shows an example of transferring the "earthy" concept to an image.

There has been very little work in the literature on the topic of transferring concepts to images. In [1] Hou and Zhang proposed a concept transfer method which required manual color editing and in which the concepts include semantic ones such as "beach". Yang and Peng [2] described an automatic method of transferring moods to an input image by extracting the color information from a suitable image chosen from a database, where each image is associated with a particular mood. Because each mood is represented by a single color, the concept has a rather narrow interpretation. In addition, it requires each of the target and input images to contain a dominant color, in order to successfully convey the concept. The work described in this paper is closest to the method proposed by Wang et al. [3] in which the colors of an input image are modified according to a color combination representing a specific theme such as "graceful". However, their approach requires user scribbles in order to enhance theme transfer.

Unlike the state of the art methods, our proposed method is automatic and uses abstract concepts specified in natural language [4]. The method is particularly appropriate for the unskilled user, who is often aware of the message he or she wants to convey, but uncertain about the colors to use for his/her document or image. This is the main contribution of our method with respect to other transfer methods such as [3]. The second contribution concerns the modelling of the color profiles of both the concept and the input image, in which model complexity is determined through a novel convex clustering approach.

Our approach preserves flexibility in two ways:

1. It employs one of two proposed techniques for mapping input chromatic content to target chromatic content. In the first technique, a probabilistic model of the input image is learnt, following which the flow between input colors and target

* Corresponding author.
E-mail addresses: nmurray@cvc.uab.es (N. Murray), sandra.skaff@xerox.com (S. Skaff), luca.marchesotti@xrce.xerox.com (L. Marchesotti), florent.perronnin@xrce.xerox.com (F. Perronnin).

0097-8493/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2012.01.008

Fig. 1. Example of transferring the "earthy" concept to an image. (a) Original image and (b) image after transfer of "earthy" concept.

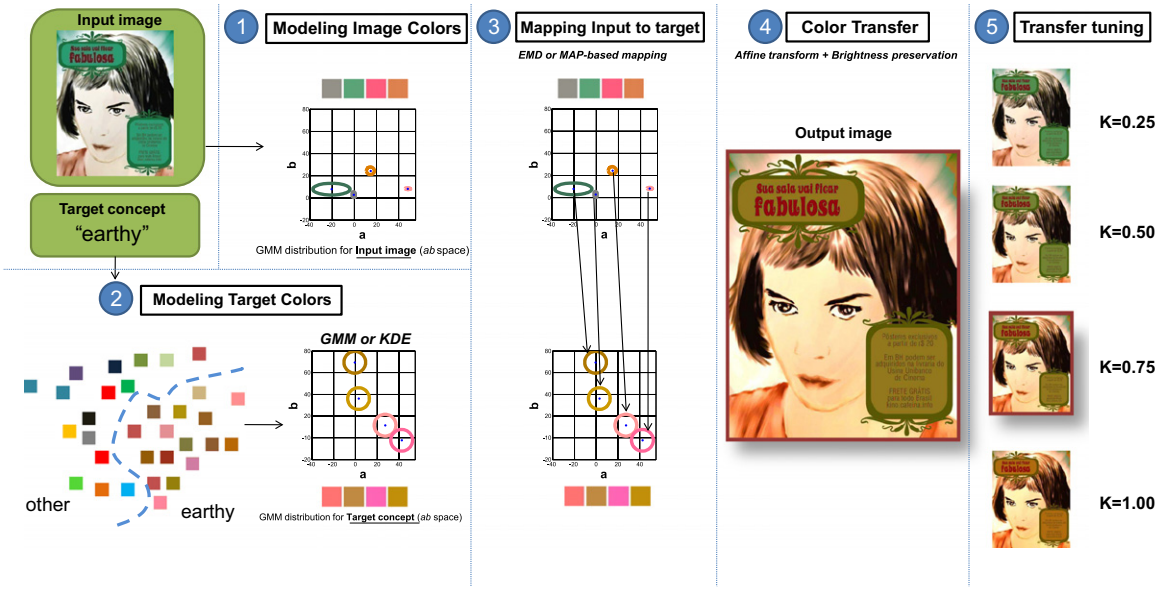

Fig. 2. Our concept transfer framework. *K* is the parameter used to adjust the transfer level. See Section 1 for details.

colors, computed using the Earth Mover's Distance (EMD), is used as a mapping. This technique typically produces non-photorealistic results. In the second technique, Bayesian adaptation is used to create an input image model by adapting the concept model using the chromaticities present in the input image. In this case the adaptation inherently maps input colors to similar concept colors resulting in a more natural-looking image.

2. The final stage of our approach, where color transfer is performed, includes a mechanism which allows the user to tune the level of concept transfer to his or her liking. Fig. 2 summarizes our proposed concept transfer framework.

The remainder of the paper is organized as follows. In Section 2 the state-of-the-art in image-based and concept-based color transfer is described. Section 3 describes the convex clustering algorithm used to set color model complexity. In Section 4 we describe the stages of our automatic concept transfer framework. Section 5 shows and discusses the results obtained using this framework and compares our results to those of a state-of-the-art approach. In addition, this section describes a test performed on non-expert subjects who were asked to identify, between a pair of images, which image is more consistent with a concept. To the best of our knowledge, this is the first time a user study of this nature is performed. Conclusions are given in Section 6.

2. Previous work

There is considerable work in the literature on color transfer. With the recent trend of labeling color combinations, or color schemes, with abstract concepts [5,6], the problem of concept transfer is being increasingly addressed. We first start with an overview of color transfer and previous work. We then give an overview of the work on concept transfer.

Color transfer: Color transfer methods typically consist of three main steps which are described below along with the related work.

Color model representation: The colors present in input and target images are modeled using representations such as probability density functions, particularly Gaussian Mixture Models (GMMs), or 3D color histograms. GMMs and color histograms contain components which are representative of the modeled color data, specifically the means and covariances of Gaussian components and the centers of (non-empty) histogram bins, respectively. We refer to these representative color components as *swatches*. Reinhard et al.'s seminal work on color transfer models the color profile of an image by the mean and covariance matrices of the pixel representation values in the color space channels [7]. To avoid computing a full covariance matrix, the color representation of an image is usually transformed to the CIELAB (hereafter referred to as the *Lab*) color space to decorrelate the channels. Other methods use full covariance matrices and operate in RGB space [8]. For a more representative profile, color histograms [2,9–11] and GMMs [12,13] have been employed. A short coarse color histogram for an image, termed a color palette, has also been used [14]. A target color model is usually obtained from a reference image supplied either by the user [7,10,14], or via image retrieval from a database [2,15]. The composition of this reference, or target image, must be similar to that of the input image, in order for color transfer to occur on a region-to-region basis. For complicated scenes, more than one reference image may be required [16] as finding a reference image with similar content to the input image is difficult.

Input-to-target mapping: Input model colors are mapped to suitable target model swatches using various criteria, most notably distance metrics and EMD-based flows. When input and target color models are created using a single Gaussian, mapping is straightforward [7,8]. For histogram-based color models, the EMD is a popular choice for which the transportation problem between the input and target histograms is solved. When the mapping is one-to-many, the final color used for transfer is a combination of target model swatches [10,15,17,18]. Colorization may be seen as a special case of color transfer from a colored target image to a greyscale input image. In this case, input-to-target mapping is often performed based on texture similarity [16,19]. Artificial color boundaries tend to appear when histogram-based models are used, as spatially close input pixels with similar colors may be placed in different bins and consequently mapped to very different target colors [2]. This phenomenon is less common with GMMs but may be noticeable if the occupancy probability for a pixel is significantly different from that of a neighboring pixel. To improve spatial coherence, constraints on the color differences which are allowed between neighboring pixels were introduced [10,16]. In object-to-object or background-to-foreground color transfer applications, segmentation is required, which puts the final result at the mercy of the quality of the segmentation [10,15].

Color transfer: Transformations between mapped input and target model swatches are first determined. The affine transformation technique introduced by Reinhard et al. is the most common one for transferring color to an input pixel x^{in} to produce an output pixel x^{out} [7,10,12]. First, the pixel colors are represented in a decorrelated color space. This allows the use of diagonal covariance matrices for the input and target color representations, Σ^{in} and Σ^{t}, respectively, which are computed along with the means μ_{in} and μ_{t}. Using these statistics, the transform for which the statistics of x^{out} match the target statistics is $x^{out}=\mu^{t}+(\Sigma^{t})^{1/2}(\Sigma^{in})^{-1/2}(x^{in}-\mu^{in})$. For input and target representations with multiple model swatches, a linear combination of such affine transformations is used to transfer color to x^{in}. Pitié and Kokaram introduced a linear transformation derived from the *Monge-Kantorovich theory of mass transportation, which minimizes the amount of changes in the image colors due to the transfer [20]. Another common color transfer technique involves transforming the input image such that the resultant output image has the same color distribution as that of the target image. Pitié et al. [9] transformed the probability density function of the input image into that of the target image. As opposed to automatic color transfer, the approach in [21], following [22], requires the user to specify regions of the input and target images whose colors should be matched by drawing strokes across these regions. The cumulative density functions of the input regions are then modified to match that of their target regions using a model whose parameters are determined by solving a constrained optimization problem. For intricate images, many region pairs may need to be defined, making this type of user interaction laborious.

Concept Transfer: One of the earlier approaches is that of Hou and Zhang on color conceptualization [1]. Their method extracts category-specific concept representations. Therefore, the representation for a concept such as *warm* differs for the semantic categories "forest" and "beach". First, reference images from a database were manually tagged as belonging to a semantic category. Images from a category were then clustered and each resulting cluster was manually labeled with a concept such as *warm*. A cluster is represented by an average of the hue histograms of its associated images. Histogram matching is performed to alter the colors of an image in order to match those of a desired concept. Since this method does not transfer the saturation and intensity components of the concept colors, the corresponding components of the input image must be similar for the concept to be perceptible in the output image. Another method which performs concept transfer is that of Yang and Peng [2]. Their method is automatic as it does not require user intervention. However, in their method the concept needs to be specified through a target image which contains a dominant color, thus making the identification of a suitable target image tedious for a user. In addition, a concept is represented by one color in this method, unlike in our case where it is a combination of colors. More details on Yang and Peng's approach are provided in Section 5.

The work described in this paper is closest to that of Wang et al. who propose a technique for color transfer based on concepts [3]. Their approach uses a database of natural images to learn relationships between color and texture. For example, "grass" is typically associated with "green" or "yellow". These relationships serve as constraints during color transfer, limiting the appeal of this approach for non-photo-realistic color-modification applications. In addition, prior knowledge is important for forming these relationships, and user input is necessary to impose realism on content which lacks texture. Finally, in the user test, the authors of [3] do not ask the subjects to rate images according to their consistency with a *concept* but rather according to their consistency with colors associated with a concept.

3. Complexity by convex clustering

In our concept transfer method, we model both the target concept and the input image using Gaussian Mixture Models (GMMs). Setting the complexity of a GMM model (i.e. the number of Gaussian components) is a challenging task. In general, there is no one number of components which fits all images or concepts. Moreover, in the specific case of color transfer, determining a suitable number is fundamental. If, for example, the number of components in the input image GMM is too small, perceptually different colors may be associated with the same component.

On the other hand, if the number of components is too large, perceptually similar colors will be associated with different components. The convex clustering algorithm proposed below is an efficient solution for determining the model complexity.

3.1. The convex clustering algorithm

This algorithm involves the optimization of an objective function using the Expectation-Maximization (EM) algorithm, and includes a novel pruning method that dramatically increases efficiency.

The objective function: Let $x_t, t=1,\ldots,T$ be a set of T pixels to be clustered. We choose to represent the color values of the pixels in the *Lab* color space, where the Euclidean distance correlates with the human perception of color. Our concept transfer framework will only use chromatic content, and as such we use only the *ab* components. Therefore, each of the x_t's is a 2D point, represented in the *ab* space. In convex clustering, a kernel is centered on each point, with an assigned weight w_t, and all points are potential cluster centers. The objective function to be optimized is [23]:

$$L=\sum_{t=1}^{T}\log\left(\sum_{s=1}^{T}w_s k_s(x_t)\right) \tag{1}$$

with respect to the w_t's under the following set of constraints:

$$0\le w_t\le 1;$$

$$\sum_{t=1}^{T}w_t=1. \tag{2}$$

The kernel values $k_s(x_t)$ express a similarity between points x_s and x_t. Since we use the Euclidean distance, we compute the similarity measure using the Gaussian kernel:

$$k_s(x_t)=\frac{1}{2\pi^{D/2}\sigma^D}\exp-\frac{\|x_t-x_s\|^2}{2\sigma^2}, \tag{3}$$

where σ is the standard deviation of the kernel and D is the dimensionality of the vectors (2 in this case). Using a normalized kernel k ($\int_x k(x)\,dx=1$) provides a probabilistic interpretation of convex clustering: $\sum_{s=1}^{T}w_s k_s$ is a probability density function which is a GMM in the case of the Gaussian kernel and the function of Eq. (1) is a log-likelihood objective function.

Note that in this formulation of the problem, σ is the only parameter that must be set. Because it affects the similarity measures between points, it controls the degree of color granularity present in the model. Therefore, by using this formulation, a fixed degree of color granularity is set, rather than a fixed number of model components. In this way, the formulation provides an intuitive proxy for setting the model complexity that is consistent across all images.

The solution: We solve the convex clustering problem using EM [23]. This algorithm consists of an E-step (Expectation) and an M-step (Maximization). The E-step computes the T^2 assignments γ_{st} of point x_s to the Gaussian centered on x_t as such: $\gamma_{st}=w_t k_t(x_s)/\sum_{j=1}^{T}w_j k_j(x_s)$. The M-step re-estimates the w_t values as such: $w_t=(1/T)\sum_{s=1}^{T}\gamma_{st}$. To ensure convergence to the optimal solution, all w_t's should be initialized to non-zero values. In our experiments, we initialize all w_t to $1/T$. As the number of iterations increases, most of the w_t's converge to zero and the number of clusters, or equivalently Gaussians, is therefore reduced.

Pruning clusters: If we knew beforehand which w_t values would converge to zero, then we could speed-up the EM algorithm significantly by initializing these values to zero. As this is not the case, we attempt to predict which values will converge to zero and prune them at the early stage of the algorithm. Such pruning implies removing unlikely cluster candidates by adding a third step to the EM algorithm. The authors in [23] proposed setting all the w_t's below a certain threshold to zero after the M-step. However, such a modification would have a limited impact on the speed of convergence as only a very small fraction of w_t's are zeroed at each pruning step. We propose a novel pruning scheme which significantly improves the convergence rate by zeroing a large percentage of w_t's without compromising the optimality of the solution. We note that if $w_t\neq 0$, then x_t is a cluster center and we expect x_t to be assigned with higher probability to its own cluster than to any other one. This can be translated as follows: $s\neq t$, $\gamma_{tt}\ge\gamma_{ts}$ or $w_t=0$. Therefore, we propose the following pruning step:

if there exists an index s such that $\gamma_{tt}<\gamma_{ts}$,

then set $w_t=0$. (4)

3.2. Comparative results

On synthetic data: When a pruning step is introduced into the algorithm, convergence to the optimal solution (in a maximum-likelihood sense) is not guaranteed. However, experiments show that the log-likelihood of the solutions reached are on par with those obtained when running the EM algorithm without the pruning step. In addition, convergence is sped up with pruning as it sets a large number of w_t's to zero. We illustrate the performance of the convex clustering approach on synthetic data, which consists of 4000 points drawn at random from four isotropic Gaussians. Each Gaussian produced 1000 points and was isotropic with $\sigma=1$ in a 2D space. Results are shown in Fig. 3. The left plot of the figure shows that the solution using the EM algorithm of [23] contains 150 clusters, and the solution is obtained after 2 min with a log-likelihood value of $-6.2847\mathrm{e}{+03}$. The right plot of the figure shows that the solution using the EM algorithm with our proposed pruning mechanism contains four clusters, and the solution is reached after 3 s with a log-likelihood value of $-6.2814\mathrm{e}{+03}$. Therefore, the EM algorithm using our proposed pruning method as compared with that of [23] results in faster convergence with a similar log-likelihood value.

On real data: We illustrate the performance of our approach on real data in the context of concept transfer. In Fig. 4, we compare concept-transferred images when an input image GMM is initialized using: (a) agglomerative clustering where the number of Gaussians N is set to an ad-hoc value of 16, which is a good general value; (b) agglomerative clustering where N is set to 1, after visual inspection of the points distribution of the input image showed 1 to be a reasonable value; and (c) convex clustering where the number of Gaussians is determined automatically using our proposed pruning mechanism. In each case, the resulting GMM is used to initialize a MLE estimation of the GMM, which re-estimates the means of the Gaussians and estimates their covariances matrices. The first row of Fig. 4 shows the input image and its GMM for (a), (b) and (c). The second row shows the target concept, represented by a sample of associated colors, and its GMM for (a), (b) and (c). The last row shows the concept transfer results in each case. Notice that when an input image GMM of $N=16$ is used, artifacts occur in the transfer due to the over-segmentation of the image regions resulting from too many Gaussian components. The convex clustering approach automatically sets $N=1$ and consequently no artifacts in the transfer results can be seen. Therefore, this method can be seen as reducing redundancy by retaining only those sample points which are sufficient for explaining the data.

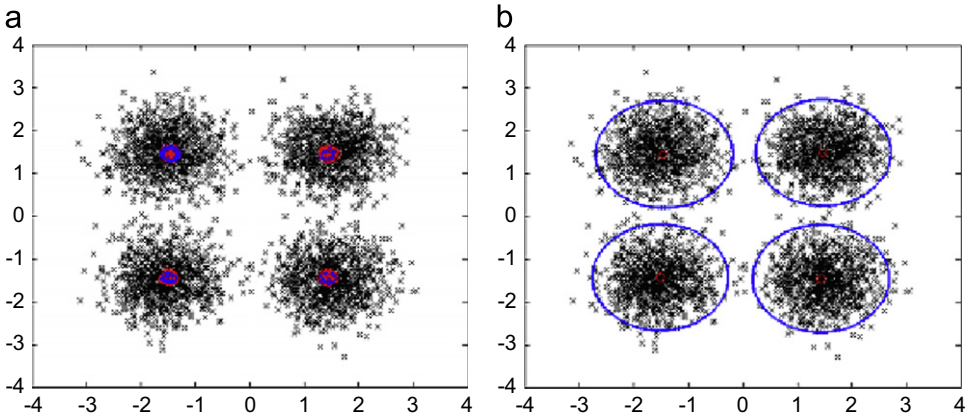

Fig. 3. Solutions found by the convex clustering algorithm using (a) the pruning mechanism of [23] after 2 min; (b) the proposed pruning mechanism after 3 s. The small red circles show the cluster centers, i.e. those points x_t with a non-zero w_t. The blue circles centered on the cluster centers have a radius which is proportional to the w_t value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

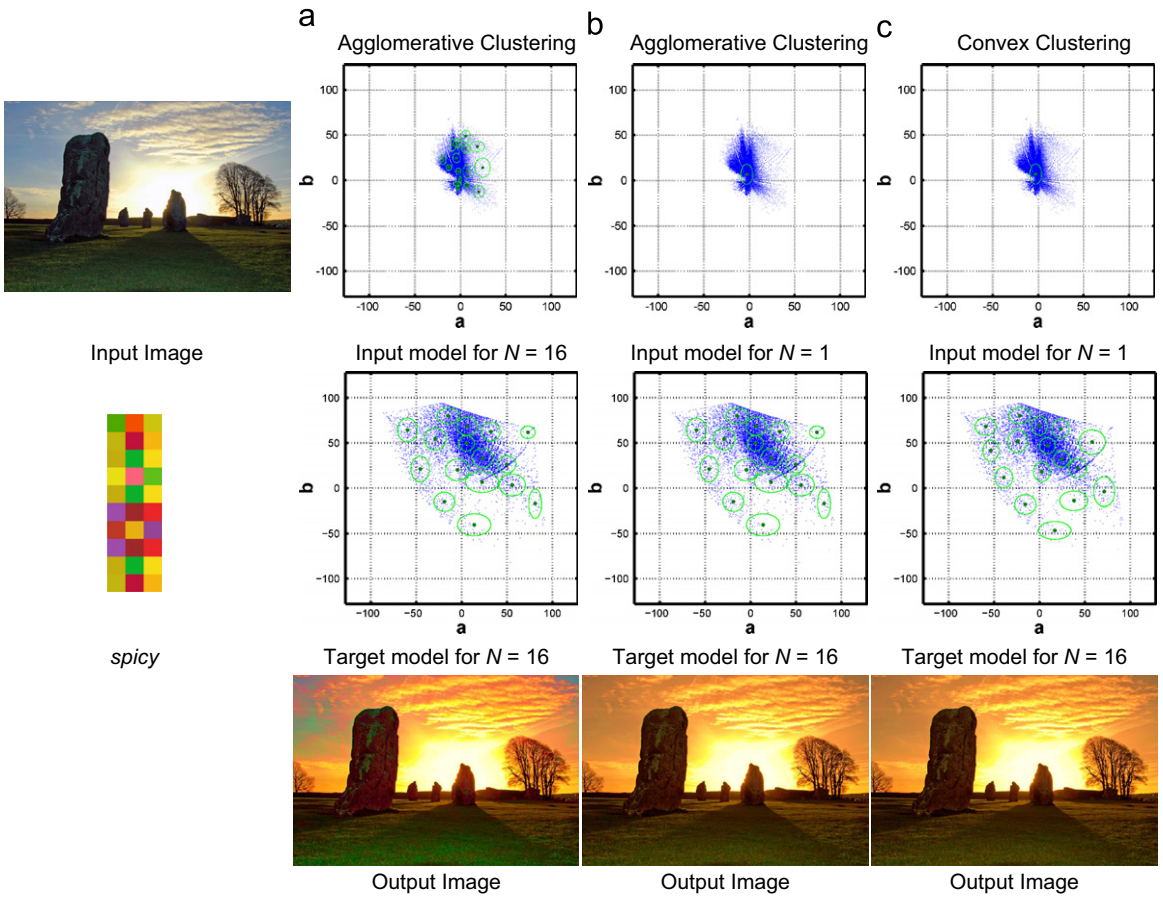

Fig. 4. Comparison between agglomerative and convex clustering: (a) shows agglomerative clustering for an ad-hoc number of $N=16$ clusters. The output image in this case contains significant artifacts due to an overly granulated clustering of the input image chromatic data. The distribution of the data must be known beforehand in order to set $N=1$ for the input image data (shown in (b)). As shown in (c), convex clustering automatically generates $N=1$ clusters for this data.

4. Automatic concept transfer

Our method for automatic concept transfer aims to enable a user to modify any input image such that the modified version conveys a desired concept. The concept is chosen by the user from a selection of concepts described in natural language, such as "earthy". As illustrated in Fig. 2, our method consists of four main stages:

1. modeling the target concept (described in Section 4.1);
2. modeling the input image (described in Section 4.2);
3. mapping input colors to target colors (described in Section 4.3);
4. color transfer (described in Section 4.4).

As a fifth optional stage, the user may adjust the level of concept transfer. We describe these stages next.

4.1. Modeling the target concept

One of the main advantages of concept-based color transfer is that the user does not have to find an appropriate target or

reference image. Instead, he/she only needs to specify the target concept with a natural language tag. However, a non-negligible challenge is represented by finding the right colors to represent such concepts.

Research in this domain is hindered by the shortage of data annotated with abstract concepts arising from emotional, mood, stylistic or aesthetic labels. Furthermore, due to the nature of these concepts, such data tend to be 'noisy' in the label space: user preferences differ, the concepts themselves are difficult to define in a consistent manner, and any data actually labeled in this way is often incomplete. Despite these difficulties, research has been conducted into linking color to abstract categories [5,6]. Most methods, however, have so far been aimed at associating color name tags to single colors. This was performed either through user tests in constrained environments [24–27] or by training statistical models using data collected on the web [28]. To our knowledge, only very few studies [2,3,29] explored the relationship between perceived colors and the emotion induced, but the results again involved a limited set of individual colors and categories. We first start by describing how we annotate colors in Section 4.1.1 and then we move on to describe how the target concept models are created in Section 4.1.2.

4.1.1. Annotating colors

In the presented work, we defined a vocabulary of natural language tags (typically adjectives) which describe 15 target concepts (such as capricious, classic, cool or delicate; see Fig. 5 for the complete list) that are indicative of some of the moods which people may associate to an asset when visualizing combinations of colors in images or documents.

We use two databases of color schemes as model data. The first database, which we call "CC", was obtained from the book "Communicating with Color" [30], an authoritative guide to color combinations and color principles. These schemes are labeled with 1 of the 15 concepts. The schemes are annotated by experienced color consultants. There are 24 schemes assigned to each concept, and each palette contains three swatches. Therefore, for this database, each concept has 72 exemplar colors. Fig. 5 shows examples of schemes from CC. As can be seen, the visual coherence is high. However, for each category/concept we have a small number of examples.

The second database, which we call "CL", comprises 22,000 schemes downloaded from Color Lovers, a popular social network for graphic designers [31]. These schemes were downloaded if they were tagged with keywords associated with a concept contained in CC. The schemes were tagged by amateur designers and are therefore weakly annotated.

This leads to two very distinct data sets with two key challenges: the first set has a minimal number of examples per class; the second is richer, but more noisy in the textual annotations.

Since we want to build a model to synthesize the colors of the target concepts, we used an approach similar to that described in [6]. Specifically, we first trained a supervised learning method [32] using the color schemes of CC. We then employed the resulting model to classify the color schemes of CL according to the 15 target concepts. We characterized each color palette as a sequence of colors, which are referred to as color *swatches*. Each swatch is represented by a 3D vector in a given color space, such as RGB, HSV or CMY. A palette of c color swatches is represented by a $c \times 3$ dimensional vector as such: $[x_1,..,x_k,..,x_c]$, where x_k is a 3D vector denoting a single color swatch. In the case of the palette databases used, the CC and CL schemes have three and five swatches, respectively. After examining several feature combinations, we found that a feature vector resulting from the concatenation of the means of the swatches for each channel in HSV, *Lab* and RGB space performed well.

We trained with these feature vectors 15 SVM classifiers in a one-vs-all configuration. A radial basis function kernel with default parametrization was used and the models were selected using 5-fold cross-validation on CC. The resulting 15 models were then used to classify and score each palette of CL with respect to the 15 concepts. Finally we retained only those schemes which were assigned to a concept with probability higher than a given threshold, T_p. Fig. 6 shows the change in the distribution of the chromatic content of the "cool" concept for different values of T_p.

After this process was completed, we obtained two datasets of color palettes, where each palette is associated with 1 of the 15 concepts. We now describe how we use the swatches of these palettes to create concept models.

4.1.2. Creating the model

The most straightforward method to model the concepts is to fit a statistical model such as a GMM using the swatches as observations. This ensures that the concept is well represented by a combination of colors as compared with one- or two-color representations of concepts as in [5,29,33]. In a few cases, the number of swatches (and therefore colors) associated with a concept is very small and therefore a GMM cannot be fit to them. In this case a good alternative is to use a non-parametric technique such as the Parzen Window or Kernel Density Estimation (KDE). We detail these two techniques which are developed and tested in our framework.

Kernel-based Density Estimation: We interpret the colors associated with the target concept as a sequence of i.i.d. random samples $x_1,..,x_i,..,x_n$ drawn according to some unknown probability law $p(x)$. To estimate the density distribution of these samples we perform convex clustering on colors x_i using the approach described in Section 3. The number of cluster centroids N is thus automatically determined. The covariance matrix of a Gaussian component, Σ, is diagonal with σ^2 set to 100 for both a and b channels.

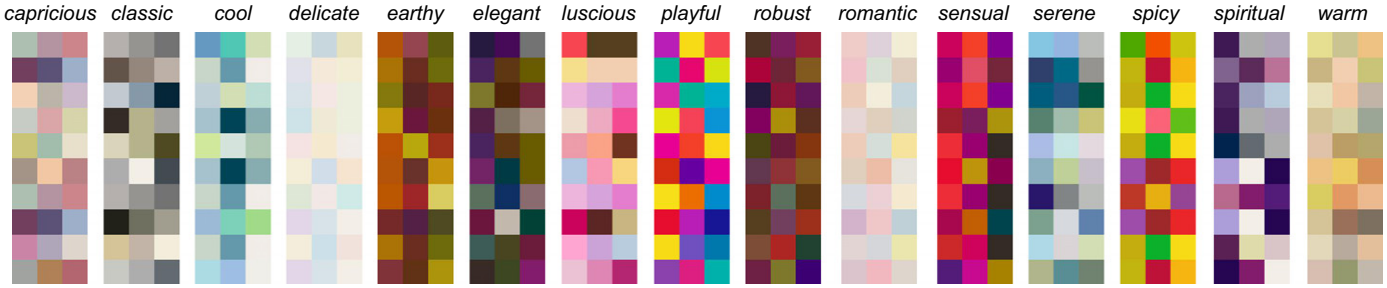

Fig. 5. Examples of schemes from the Communicating with Color database for each of the 15 concepts considered. Each scheme comprises three swatches. Each concept has 24 associated schemes, 10 of which are shown.

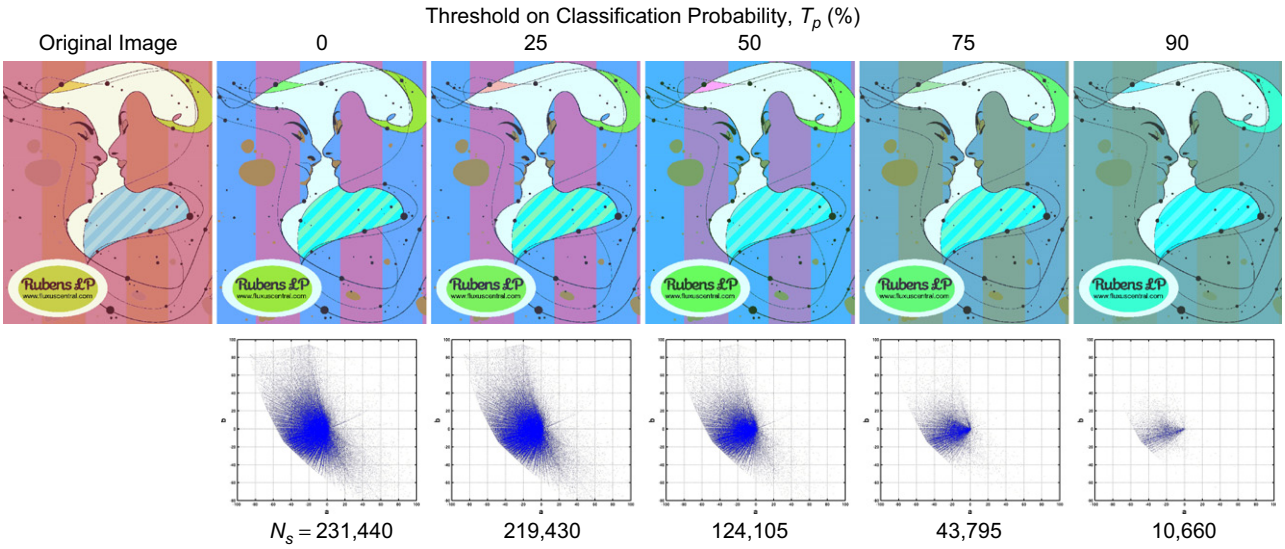

Fig. 6. Effect of varying T_p: the bottom row lists the number of swatches, N_s, belonging to palettes associated with the "cool" concept for T_p ranging from 0% to 90% probability; the middle row shows the distribution of the swatches' *ab* values; the top row shows the result of transferring each version of the concept to an image. For higher values of T_p the variety of colors in the concept and therefore in the output image is decreased. We chose a threshold of 75% for all concepts, as a good compromise between concept distributions that are too broad and those that are too narrow. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Gaussian Mixture Model Estimation: For concepts with a sufficient number of distinct color schemes, we learn a probabilistic model, namely a GMM. The parameters of the GMM are denoted by $\lambda^{tgt}=\{w_i^{tgt},\mu_i^{tgt},\Sigma_i^{tgt},i=1,\ldots,N\}$ where w_i^{tgt}, μ_i^{tgt}, and Σ_i^{tgt} are, respectively, the weight, mean vector and covariance matrix of Gaussian i.

Let $X=x_t,t=1,\ldots,T$ denote the set of observations which are all the swatches of the schemes associated with the target concept in the CL database. If q denotes which Gaussian emitted x_t, the likelihood that x_t was generated by the GMM is

$$p(x_t|\lambda^{tgt})=\sum_{i=1}^{N} w_i^{tgt} p_i(x_t|\lambda^{tgt}), \tag{5}$$

where $p_i(x_t|\lambda^{tgt})=p(x_t|q=i,\lambda^{tgt})$. The weights are subject to the constraint: $\sum_{i=1}^{N} w_i^{tgt}=1$. The components p_i are given by

$$p_i(x_t|\lambda^{tgt})=\frac{e^{-1/2(x_t-\mu_i^{tgt})'(\Sigma_i^{tgt})^{-1}(x_t-\mu_i^{tgt})}}{(2\pi)^{D/2}|\Sigma_i^{tgt}|^{1/2}}, \tag{6}$$

where $|.|$ denotes the determinant operator and D denotes the dimensionality of the feature space. The parameters of the GMM can be obtained by maximizing the log-likelihood function $\log p(X|\lambda^{tgt})$, which is referred to as maximum likelihood estimation (MLE). The standard approach to MLE uses the EM algorithm. This algorithm is only guaranteed to converge to a local optimum not to a global one. The location of convergence is dependent on the initialization parameters. In other words, different initialization conditions will, in general, lead to different schemes. We set the complexity of the GMM model using the approach described in Section 3.

4.2. Modeling the input image

In order to model the colors of a natural image, it is traditionally assumed that the pixel values represented in a given color space are generated by a probabilistic mixture model. This model can be either discrete or continuous. In the discrete case, an image is modeled through a color histogram. However, such a representation is sensitive to quantization errors and setting the number of bins is not trivial. Therefore we use a mixture model, namely a GMM, which is the most common continuous model used in this context. As images produced by today's cameras typically contain millions of pixels, observation data for the GMM are abundant. The GMM is trained in an analogous fashion to that described in Section 4.1, the only difference being that the input image pixel values are used as the observation data in the maximum likelihood estimation.

4.3. Mapping input colors to target colors

We relate the chromaticities of the target and input image models through a mapping which finds soft correspondences between their Gaussian components. An example of a mapping is shown in Fig. 2 for a sample input image and the concept "earthy". The circles with a point at their centers represent Gaussian components. The points are at the mean values of the components, while the length of the axes of the ellipses corresponds to the standard deviation along their dimensions. Correspondences between the components are represented by arrows.

To perform the association we propose two methods: the first is based on the EMD [34], which we refer to as EMD-based mapping (EBM), while the second is a learning technique which we refer to as adaptation-based mapping (ABM).

EMD-based mapping (EBM): First, we create the concept and input color models using the methods described in Sections 4.1 and 4.2, respectively. To associate input and concept model swatches, we turn to the EMD algorithm, which attempts to solve the following optimization problem:

$$\begin{aligned}
&\min_{\{f_{i,j}\}} \quad \sum_{i=1}^{M}\sum_{j=1}^{N} f_{i,j} Dist(s_i^{in},s_j^c);\\
&\text{subject to} \quad \sum_{j=1}^{N} f_{i,j}=w_i^{in}, \quad i=1,\ldots,M \quad (C1),\\
&\sum_{i=1}^{M} f_{i,j}=w_j^c, \quad j=1,\ldots,N \quad (C2),
\end{aligned} \tag{7}$$

where M is the number of swatches in the input model and *Dist* is the cost matrix containing the Euclidean distance between each input swatch s_i^{in} and concept swatch s_j^c. The quantities w_i^{in} and w_j^c are the weights for the i-th input and concept model swatch,

respectively. The flow $f_{i,j}$ can be considered to be the part of swatch s_i^{in} which is mapped to s_j^c. The target swatch s_i^{tgt} for the input swatch s_i^{in} is computed as a weighted average of the concept swatches, where the weight for s_j^c is $f_{i,j}$. Freedman and Kisilev [10] introduced the idea of mapping input swatches to concept swatches using EMD-derived flows. However, in their case, input and target models were represented by color histograms. Constraint (C1) requires that each input model swatch has flows that sum to its weight. Similarly, constraint (C2) requires that each concept model swatch has flows that sum to its weight. Therefore each concept swatch is guaranteed to be associated with at least one input swatch and vice versa. This ensures that each input color is transferred to a concept color and, in addition, that all concept colors are used.

Adaptation-based mapping (ABM): Rather than the method described in Section 4.2, we obtain the input image model by adapting the concept GMM using the Maximum *A Posteriori* (MAP) criterion. This representation of the input model is motivated by the fact that the Gaussian components of an adapted GMM maintain a one-to-one correspondence with the original GMM, as described by Reynolds et al. [35]. The input GMM is initialized using the parameters of the concept GMM and the set of input image pixels $X=\{x_t, t=1,\ldots,T\}$ are used as new observations for the input GMM estimation. The EM algorithm is used for the MAP estimation and consists of two steps:

- *An Expectation* (*E*) *step* that computes for observation x_t the occupancy probability $\gamma_i(x_t)$, which is the probability that x_t has been generated by Gaussian *i*. This probability may be formulated as $\gamma_i(x_t)=p(q=i|x_t,\lambda^{in})$, where q is the Gaussian which has generated x_t. The posterior occupancy probability is computed based on the current estimates of the parameters as such:

$$\gamma_i(x_t)=\frac{w_i^{in}p_i(x_t|\lambda^{in})}{\sum_{j=1}^{N}w_j^{in}p_j(x_t|\lambda^{in})}. \tag{8}$$

- *A Maximization* (*M*) *step* where the parameters are updated based on the expected complete-data log-likelihood given the occupancy probabilities computed in the E step:

$$\hat{w}_i^{in}=\frac{\sum_{t=1}^{T}\gamma_i(x_t)+\tau}{T+N\tau}; \tag{9}$$

$$\hat{\mu}_i^{in}=\frac{\sum_{t=1}^{T}\gamma_i(x_t)x_t+\tau\mu_i^{tgt}}{\sum_{t=1}^{T}\gamma_i(x_t)+\tau}; \tag{10}$$

$$\hat{\Sigma}_i^{in}=\frac{\sum_{t=1}^{T}\gamma_i(x_t)x_tx_t'+\tau\{\Sigma_i^{tgt}+\mu_i^{tgt}\mu_i^{tgt}\}}{\sum_{t=1}^{T}\gamma_i(x_t)+\tau}-\hat{\mu}_i^{in}(\hat{\mu}_i^{in})'. \tag{11}$$

Here τ is the adaptation factor which balances the new observations against the *a priori* information. We set this value to 10. Gaussian components from the concept model for which there are no new observations from the image pixels will remain unchanged in the final input model, and will have negligible weights. Due to the one-to-one correspondence between input and concept swatches, only those input swatches with non-negligible weights, and thus their corresponding target concept swatches, will affect the color transfer.

4.4. Color transfer

The color-transferred image is obtained in the *Lab* color space by applying an affine transformation to the *ab* channel values of the input image pixels. The lightness channel values of the output image pixels are set to be the same as those of the input image pixels. The transformation is calculated given the *ab* values of the swatch colors of the input and target schemes. We match the statistics of the *i*-th input swatch to those of the *i*-th target swatch using a linear transform of the form:

$$A_i=(\Sigma_i^{tgt})^{1/2}(\Sigma_i^{in})^{-1/2};\quad B_i=\mu_i^{tgt}-\mu_i^{in}A_i, \tag{12}$$

where Σ_i^{tgt} and Σ_i^{in} denote the covariance matrices of the *i*-th components of the models representing the target and input schemes, respectively, and μ_i^{tgt} and μ_i^{in} denote their means. Note that since we choose the covariances to be diagonal, $(\Sigma_i^{tgt})^{1/2}$ and $(\Sigma_i^{in})^{-1/2}$ are uniquely defined. The color transfer functions used in the affine transformation are

$$A(x)=\sum_{i=1}^{N}\gamma_i(x)A_i;\quad B(x)=\sum_{i=1}^{N}\gamma_i(x)B_i. \tag{13}$$

Finally, the transformation is applied to the *ab* channel values of the input image pixel x^{in} to obtain the corresponding values in the output image pixel, denoted by x^{out}:

$$x^{out}=A(x^{in})x^{in}+B(x^{in}). \tag{14}$$

Different levels of transfer can be performed by introducing a parameter K into the affine transformation using the following formulations of A_i and B_i:

$$A_i=K(\Sigma_i^{tgt})^{1/2}(\Sigma_i^{in})^{-1/2}+(1-K)I;$$

$$B_i=K\mu_i^{tgt}+(1-K)\mu_i^{in}-\mu_i^{in}A_i, \tag{15}$$

where I is the identity matrix. Using these formulations for $K=1$, A_i and B_i revert to the original case (Eq. (12)), while for $K=0$, A_i and B_i revert to the identity matrix and the null matrix, respectively, so that the input image pixels are unchanged. Therefore, by varying K from 0 to 1, the level of concept transfer may be controlled in a linear fashion. Fig. 7 illustrates the effect of K on color transfer. Note that $K=1$ for all the other results shown in this paper.

Since the image pixel representation values in the channels of the *Lab* color space are not completely decorrelated [36],

Fig. 7. The effect of varying K on concept transfer for "warm". For $K=0$ the image is unchanged while for $K=1$ full transfer has been applied, producing the "warmest" version of the output image. Transfer was performed using GMM-based concept models.

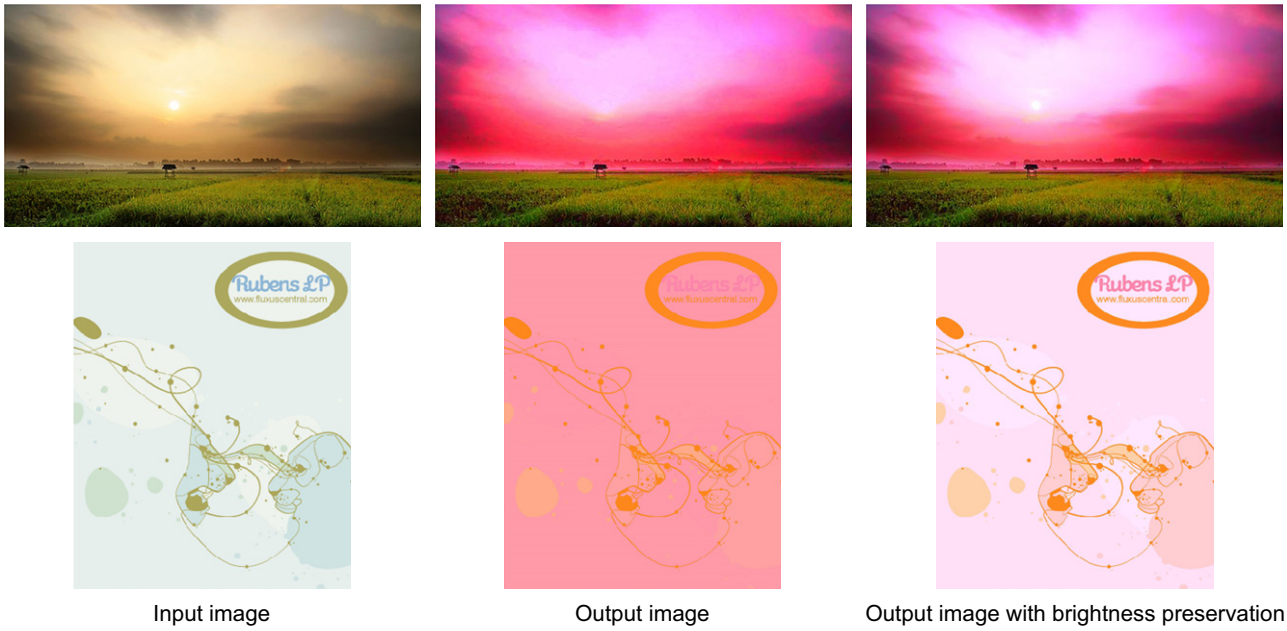

Fig. 8. Row 1: after concept transfer with "playful", the sun in the original image has been recovered from the output image and fewer color gradient artifacts are visible. Row 2: after concept transfer with "sensual", the contrast between various design elements is lost in the output image. It is then recovered after the brightness preservation step. Transfer was performed using KDE-based concept models. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

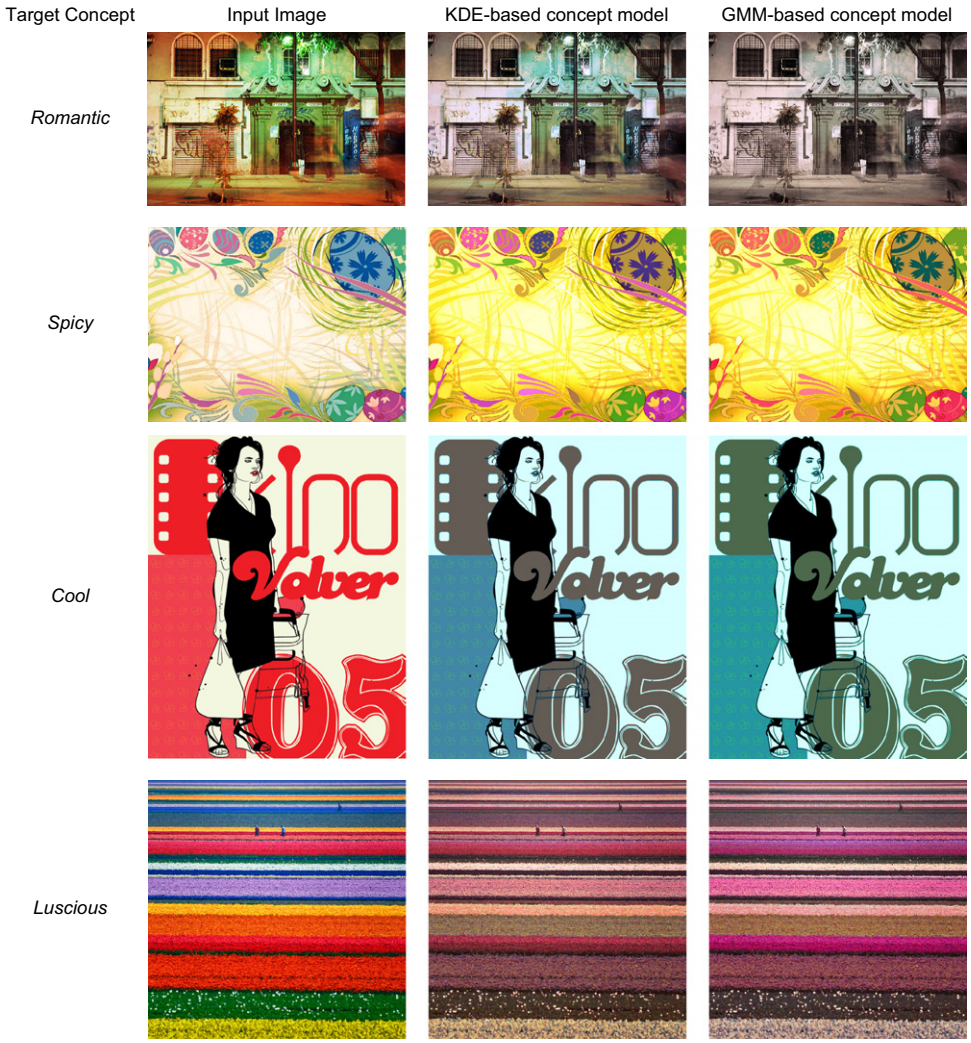

Fig. 9. Comparing our method in the cases of KDE- and GMM-based models for the concepts "romantic", "spicy", and "luscious" with EMD-based transfer. The output images differ depending on the method used to create the concept model. However, all results are consistent with the color scheme of the associated concept. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

our transfer method may still modify the brightness of pixels in the output image. These modifications might involve important features or design elements of the image thus altering its semantic content (see Fig. 8) or impacting the feel and meaning of the input image. In addition, the converse problem exists wherein features not present in the input image, such as sharp color gradients, are observed in the output image as artifacts. We alleviate these issues by preserving the brightness of the input image within the color transferred image as follows. First we scale the transferred image pixel representation values in the R, G, and B channels to attain the brightness of the input image. However, a few pixels might be assigned to values >255 for one or more channels. We cap such out-of-gamut pixels to 255. We then increase the remaining in-gamut pixel values by amounts proportional to their contribution to the brightness channel, so as to compensate for the capping. This cap-and-increase procedure is repeated until all pixels are in-gamut. Fig. 8 shows the improved results obtained by preserving the image brightness.

5. Results

We show concept transfer results using graphic design, photobook template, and natural images. We choose images with a variety of scene content and with both smooth and intricate textures. For a MATLAB implementation running on an Intel Core 2 Duo CPU at 3.00 GHz with 2 GB RAM, a typical run time for an image of size 768×1024 pixels is 12 s.

5.1. KDE versus GMM concept models

Fig. 9 shows results for our approach using KDE- or GMM-based concept models for EMD-based transfer. More results are provided in the supplementary material. While there are differences between the results achieved with these two concept modelling techniques, all results are consistent with the color schemes of the concepts.

5.2. EBM versus ABM-based color transfer

Fig. 10 compares results using EMD-based mapping (EBM) and adaptation-based mapping (ABM). With ABM, concept swatches that correspond to input swatches with negligible weights do not affect the color transfer. This means that only the concept colors with some similarity to input colors are used to perform concept transfer. With EBM on the other hand, constraint (C2) ensures that all concept colors are used in proportion to their weights in the concept model. Therefore EBM is a more aggressive color

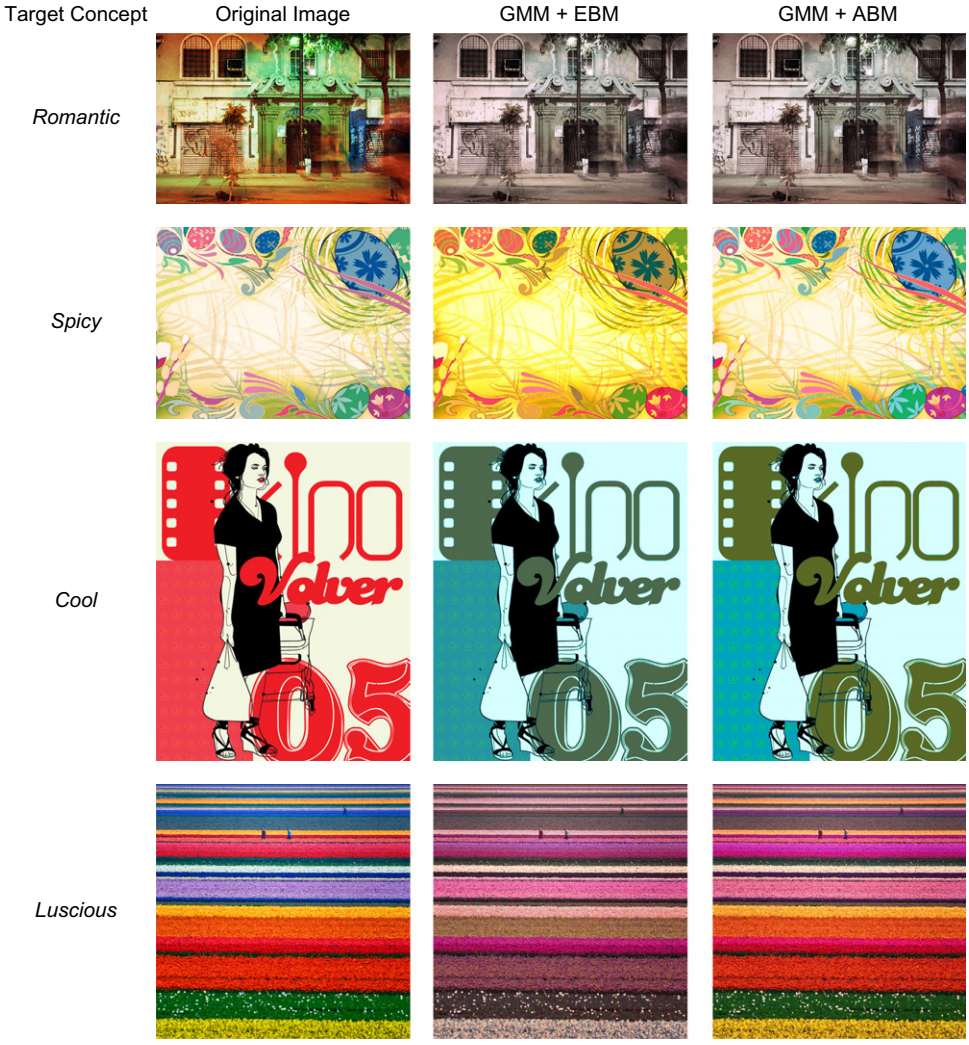

Fig. 10. Comparison between EBM and ABM using GMM-based concept models. Note that for the examples showing transfer of the "spicy" and "luscious" concepts, the EBM-based results are more aggressive. This is due to constraints C1 and C2, described in Section 4.3, which require all concepts colors to be mapped to an input color and vice versa. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

transfer, which may be desirable for non-photorealistic applications. However if a more natural result is required, Bayesian adaptation is a suitable mapping alternative.

5.3. Comparison to state-of-the-art

In order to compare to previous work, we choose Yang and Peng's mood-transferring method [2] as a baseline because it is automatic and does not require user scribbling as in [3]. Fig. 11 shows results for a comparison between our method, with KDE-based concept models and EBM, and Yang and Peng's mood-transferring method [2]. For more results the reader is referred to the supplementary material. The method of [2] defines a mood or concept with one associated color. The mood of an image is taken to be the one whose predefined associated color is most similar to the most frequent color of the image's mood color histogram. The latter comprises 24 bins corresponding to the 24 color moods defined. To transfer the mood of a target image to an input image, the colors of the input image in RGB space are rotated such that the principal color axis is aligned to that of the target image. A matching technique between the modified input image and target image histograms is then applied. This technique iteratively switches colors in input bins that are over-represented with respect to the target bin with colors in bins that are under-represented, by adjusting the colors of the pixels assigned to these bins appropriately. To select a target image, a database of images with pre-computed mood definitions are used. The number of pixels in the second highest frequency bin must be less than 75% of the pixels in the highest frequency bin of the histogram of each image. This is referred to as a dominance constraint and is imposed on both the input and the target images used. It ensures that the mood colors dominate in the images, thus leading to an unambiguous mood transfer. However, this constraint greatly restricts the input and target images available to a user of this method. The target images used in Fig. 11 satisfy this constraint. For the "welcoming"/"warm" example, the input image satisfies the dominance constraint, while the "calm"/"serene" and "romantic" examples illustrate the more general case in which the input images do not satisfy the constraint. As the "romantic" example shows the output image has many artifacts due to the histogram matching and it contains a smaller variety of colors as compared to the input image. In addition, the non-probabilistic concept model of Yang and Peng produces results which, though conforming with the target image, are often particularly unsuccessful with color gradients. The "welcoming/warm" example is one such case. Our concept model, however, is multi-modal probabilistic, and therefore allows for a richer and more extensive variety of concept colors. For this reason the approach is flexible enough for transfers involving input images with varied color content.

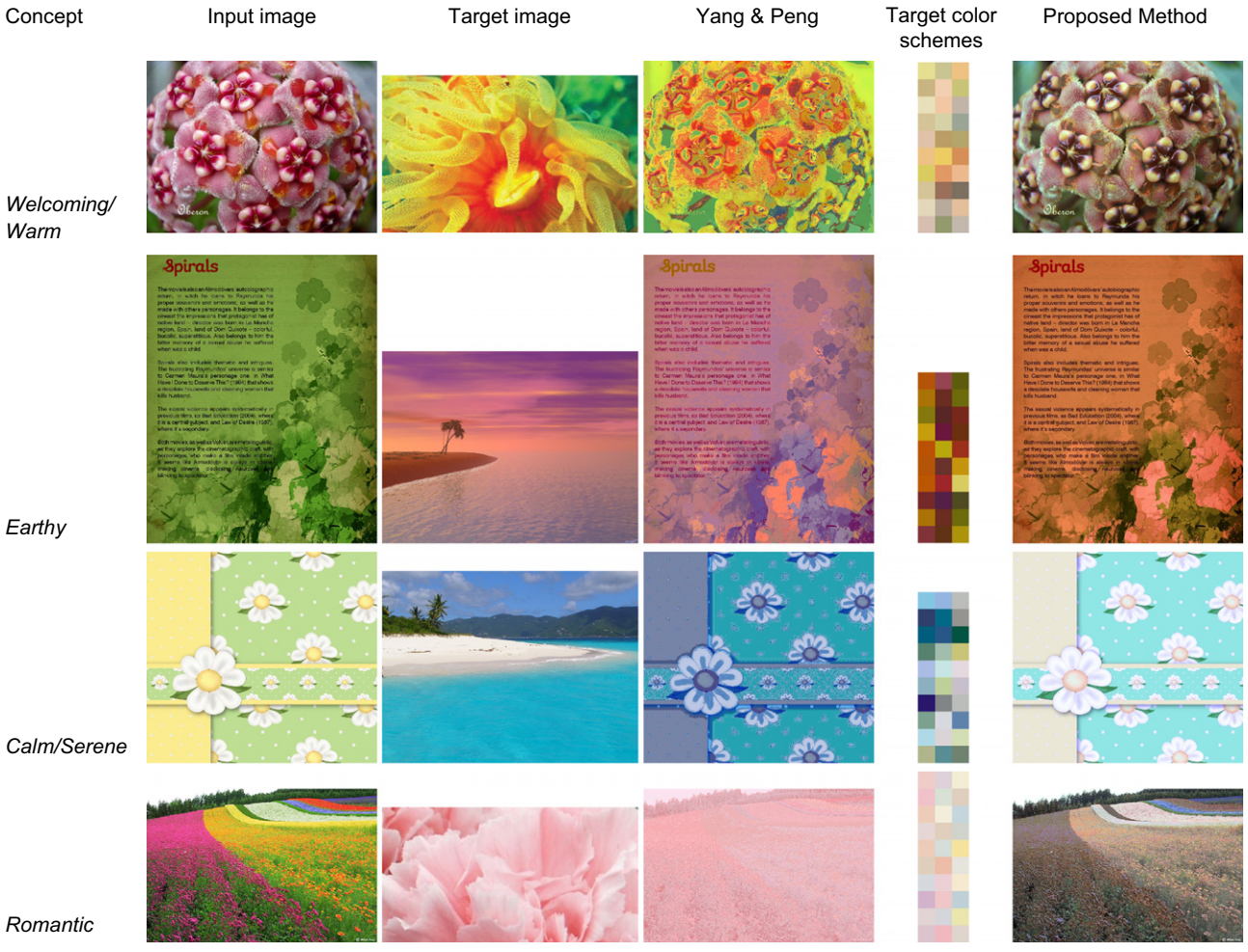

Fig. 11. Comparing to [2] when our method uses KDE-based concept models with EBM. A few schemes from each concept are shown in the fourth column. For the welcoming/warm and earthy examples, both input images satisfy the dominance constraint, while the constraint is not satisfied for the calm/serene and romantic ones. All target images satisfy the dominance constraint.

5.4. User study

To evaluate our method's ability to automatically transfer concepts to images, we performed a user study. We first collected two test images for each of the 15 concepts listed in Section 4.1. These images were chosen such that their semantic content was consistent with their concept. Concept transfer was performed on these 30 images, using GMM-based concept models with EMD-based color transfer. This resulted in 30 pairs of images, each pair containing an original and a color-transferred image. These image pairs constituted the stimuli of the user study. The pairs were displayed side-by-side on a 14″ computer screen with 1440×900 resolution. The positions of the images in each pair, either left or right, were randomly assigned.

We invited 20 participants, all with normal or corrected-to-normal color vision, to participate in the user study, which was conducted in the same room and on the same computer for each participant. None of the participants were design professionals. For each of the 30 image pairs, these users were instructed to select the image they considered to be more consistent with the concept. Note that the users were provided only with the name of the concept, and were not shown the color schemes associated with that concept. Fig. 12 illustrates the interface of the user study, showing 2 of the 30 slides presented to the participants. The users were not informed or aware of the manner in which the images were obtained, that is, that one of the images in each pair was a version of the other image, in which the color content had been modified.

Each image pair was therefore compared 20 times, resulting in 40 pair-wise comparisons per concept. We used these comparisons to fit a probabilistic choice model, namely the Bradley–Terry–Luce (BTL) model, that allows for a quantitative evaluation of our method's performance for each concept. The BTL model for a concept gives a measure of the degree of preference for the concept-transferred image. This measure is set relative to the original image, such that values above 1 indicate that the transferred image is more consistent with the concept than the original image and values below 1 indicate the contrary. The value of this measure, denoted by u, and the probability, P, of a user preferring the concept-transferred image are reported in Table 1, for each concept. For 10 of the 15 concepts, the concept-transferred image was preferable. The preference for the transferred image was especially high for concepts which are strongly associated with certain colors. For instance, our algorithm performed best for the "romantic", "sensual" and "serene" concepts, which are associated with pink, red and blue, respectively. As shades of these colors were dominant in our learned color schemes for these three concepts, the users consistently chose the transferred image. The five concepts for which the original image was preferred were "elegant", "robust", "luscious", "delicate" and "spiritual". None of these concepts are strongly associated with specific colors, such that it is difficult to transfer these concepts to an image solely by manipulating its color content.

We approximated the likely performance of ABM-based concept transfer in the user study by comparing its output images both to the 30 input images used in the user study and the transferred images used in the study, which were obtained with EBM-based concept transfer. First, we computed GMM color models for the output images obtained with both techniques, as well as for the input image. We computed the Bhattacharyya distance between the input image color model and each of the output image color models. The mean distance between the input models and the EBM-based output models for the 30 images was 0.7219, while the mean distance between the input models and the ABM-based output models was 0.6362. This indicates that, as expected, the ABM-based output images are in general closer to the input images in color composition than the EBM-based output

Fig. 12. Interface of the user study. Two examples of the 30 image pairs presented to user study participants are shown. Users were instructed to tick the box corresponding to the image they considered to be more consistent with the concept.

Table 1
BTL model parameters for each concept: u is a measure of the ability of the algorithm to transfer the concept; P is the probability of the user choosing the transferred image as more consistent with the concept.

Concept	u	P
Romantic	9.0000	0.9000
Sensual	7.0000	0.8750
Serene	5.6667	0.8500
Capricious	4.0000	0.8000
Cool	3.4444	0.7750
Earthy	3.0000	0.7500
Classic	3.0000	0.7500
Spicy	3.0000	0.7500
Playful	2.0769	0.6750
Warm	1.5000	0.6000
Spiritual	0.7391	0.4250
Delicate	0.6667	0.4000
Luscious	0.4815	0.3250
Robust	0.3333	0.2500
Elegant	0.2903	0.2250

images. We also computed the distance between the models of the output images obtained for the two mapping methods and found the mean distance to be 0.2476. Therefore, because the output images are in general much closer to each other than they are to the input image, it is reasonable to assume that the results of the user study are applicable to ABM-based transfer.

5.5. Limitations of the method

Because the lightness of pixels in the input image are maintained in their corresponding output pixels, concepts whose colors are particularly light or dark, such as "delicate", or "robust" are sometimes not well transferred.

6. Conclusions

We presented a novel framework for transferring concepts, specified by natural language, to images. The framework used a convex clustering algorithm to automatically set the complexity of models of the chromatic content of input images and target concepts. A pruning mechanism was introduced to ensure fast convergence of the clustering algorithm. The framework is flexible in that it may use either an EMD-based method or a MAP adaptation-based method to map input image chromaticities to target concept chromaticities. The mapping method is chosen based on whether the user desires a more or less photorealistic result. Our approach also provides flexibility to the user with the option of adjusting the level of concept transfer to his/her liking through a single parameter.

A user study showed the efficacy of the transfer of concepts to images using our framework. The user study is different from previous ones in the literature in that it studies the transfer of the concepts themselves rather than the color schemes, such as in [3].

An unresolved challenge in color and concept transfer is ensuring that contiguous regions of an image, which may have the same underlying color but appear differently due to illumination conditions, are recolored to the same color. This is especially difficult when shadows or highlights are present. In the future, we would like to address this challenge by introducing color constancy techniques and spatial constraints into our framework.

Acknowledgments

The authors thank those who served as subjects in the user studies and Tommaso Colombino for his invaluable advice regarding the user study.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.cag.2012.01.008.

References

[1] Hou X, Zhang L. Color conceptualization. In: ACM international conference on multimedia; 2007. p. 265–8. http://doi.acm.org/10.1145/1291233.1291288.
[2] Yang C-K, Peng L-K. Automatic mood-transferring between color images. IEEE Comput Graph Appl 2008;28(2):52–61.
[3] Wang B, Yu Y, Wong TT, Chen C, Xu Y-Q. Data-driven image color theme enhancement. ACM Trans Graph (SIGGRAPH Asia 2010 issue) 2010;29(6):146:1–10.
[4] Murray N, Skaff S, Marchesotti L, Perronnin F. Towards automatic concept transfer. In: Proceedings of the ACM SIGGRAPH/eurographics symposium on non-photorealistic animation and rendering, NPAR'11. New York, NY, USA: ACM; 2011. p. 167–76. http://doi.acm.org/10.1145/2024676.2024703.
[5] Ou L-C, Luo MR, Woodcock A, Wright A. A study of colour emotion and colour preference. Part II: colour emotions for two-colour combinations. Color Res Appl 2004;29:292–8.
[6] Csurka G, Skaff S, Marchesotti L, Saunders C. Learning moods and emotions from color combinations. In: Indian conference on computer vision, graphics, and image processing; 2010.
[7] Reinhard E, Ashikhmin M, Gooch B, Shirley P. Color transfer between images. IEEE Comput Graph Appl 2001;21(5):34–41.
[8] Xiao X, Ma L. Color transfer in correlated color space. In: ACM international conference on virtual reality continuum and its applications; 2006. p. 305–9. http://doi.acm.org/10.1145/1128923.1128974.
[9] Pitié F, Kokaram AC, Dahyot R. N-dimensional probability density function transfer and its application to colour transfer. In: IEEE international conference on computer vision, vol 2; 2005. p. 1434–9.
[10] Freedman D, Kisilev P. Object-to-object color transfer: optimal flows and smsp transformations. In: IEEE conference on computer vision and pattern recognition; 2010. p. 287–94.
[11] Pouli T, Reinhard E. Progressive histogram reshaping for creative color transfer and tone reproduction. In: Proceedings of the 8th international symposium on non-photorealistic animation and rendering, NPAR'10. New York, NY, USA: ACM; 2010. p. 81–90. http://doi.acm.org/10.1145/1809939.1809949.
[12] Tai Y, Jia J, Tang C. Local color transfer via probabilistic segmentation by Expectation-Maximization. In: IEEE international conference on computer vision and pattern recognition, vol 1; 2005. p. 747–54.
[13] Shapira L, Shamir A, Cohen-Or D. Image appearance exploration by model-based navigation. In: Eurographics, vol 28; 2009. p. 629–38.
[14] Greenfield G, House D. Image recoloring induced by palette color associations. In: International conference on computer graphics, visualization and computer vision, vol 11; 2003. p. 189–96.
[15] Lalonde J-F, Efros A. Using color compatibility for assessing image realism. In: IEEE international conference on computer vision; 2007. p. 1–8. 10.1109/ICCV.2007.4409107.
[16] Charpiat G, Hofmann M, Schölkopf B. Automatic image colorization via multimodal predictions. In: European conference on computer vision; 2008. p. 126–39.
[17] Huang T-W, Chen H-T. Landmark-based sparse color representations for color transfer. In: IEEE international conference on computer vision; 2009. p. 199–204. 10.1109/ICCV.2009.5459165.
[18] Dong W, Bao G, Zhang X, Paul J-C. Fast local color transfer via dominant colors mapping. In: ACM SIGGRAPH ASIA 2010 sketches, SA'10. New York, NY, USA: ACM; 2010. p. 46:1–2. http://doi.acm.org/10.1145/1899950.1899996.
[19] Qu Y, Wong T-T, Heng P-A. Manga colorization. ACM Trans Graph 2006;25(3):1214–20. http://doi.acm.org/10.1145/1141911.1142017.
[20] Pitié F, Kokaram A. The linear Monge–Kantorovitch linear colour mapping for example-based colour transfer. In: European conference on visual media production. London, UK; 2007. p. 1–9.
[21] An X, Pellacini F. User-controllable color transfer. Comput Graph Forum (Eurographics) 2010;29(2):263–71. (9).
[22] Levin A, Lischinski D, Weiss Y. Colorization using optimization. ACM Trans Graph 2004;23(3):689–94. http://doi.acm.org/10.1145/1015706.1015780.
[23] Lashkari D, Golland P. Convex clustering with exemplar-based models. In: Advances in neural information processing systems, vol 20; 2008. p. 825–32.
[24] Berlin B, Kay P. Basic color terms: their universality and evolution. Berkeley: University of California Press; 1969.
[25] Conway D. An experimental comparison of three natural language colour naming models. In: East-West international conference on human–computer interactions; 1992. p. 328–39.
[26] Lammens JMG. A computational model of color perception and color naming. PhD thesis. University of Buffalo; 1994.
[27] Benavente R, Vanrell M, Baldrich R. Parametric fuzzy sets for automatic color naming. J Opt Soc Am A 2008;25(10):2582–93.
[28] van de Weijer J, Schmid C, Verbeek J, Larlus D. Learning color names for real-world applications. IEEE Trans Image Process 2009;18(7):1512–23.
[29] Ou L-C, Luo MR, Woodcock A, Wright A. A study of colour emotion and colour preference. Part I: colour emotions for single colours. Color Res Appl 2004;29(3):232–40.
[30] Eiseman L. Pantone guide to communicating with color. Graffix Press, Ltd.; 2000.
[31] ColorLovers, 〈http://www.colourlovers.com/〉.
[32] Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. New York, NY, USA: Cambridge University Press; 2004.
[33] Ou L-C, Luo MR, Woodcock A, Wright A. A study of colour emotion and colour preference. Part III: colour preference modeling. Color Res Appl 2004;29:381–9.
[34] Rubner Y, Tomasi C, Guibas LJ. The Earth Mover's Distance as a metric for image retrieval. Int J Comput Vis 2000;40(2):99–121. http://dx.doi.org/10.1023/A:1026543900054.
[35] Reynolds DA, Quatieri TF, Dunn RB. Speaker verification using adapted Gaussian Mixture Models. In: Digital signal processing, no. 10; 2000. p. 19–41.
[36] Reinhard E, Khan E, Akyz A, Johnson G. Color imaging: fundamentals and applications. A.K. Peters, Ltd.; 2008.