Volume 31 (2012), Number 7 pp. 1–12 COMPUTER GRAPHICS *forum*

Content-Based Color Transfer

Fuzhang Wu[1] Weiming Dong[1†] Yan Kong[1] Xing Mei[1] Jean-Claude Paul[2] Xiaopeng Zhang[1‡]

[1]LIAMA-NLPR, Institute of Automation, Chinese Academy of Sciences, China [2]INRIA, France

Abstract

This paper presents a novel content-based method for transferring the color patterns between images. Unlike previous methods that rely on image color statistics, our method puts an emphasis on high level scene content analysis. We first automatically extract the foreground subject areas and background scene layout from the scene. The semantic correspondences of the regions between source and target images are established. In the second step, the source image is re-colored in a novel optimization framework, which incorporates the extracted content information and the spatial distributions of the target color styles. A new progressive transfer scheme is proposed to integrate the advantages of both global and local transfer algorithms, as well as avoid the over-segmentation artifact in the result. Experiments show that, with a better understanding of the scene contents, our method well preserves the spatial layout, the color distribution and the visual coherence in the transfer process. As an interesting extension, our method can also be used to re-color video clips with spatially-varied color effects.

1. Introduction

Color transfer is a practical method to change the appearance of a source image/video according to the color pattern of a target image. It has various applications in movie post-production, artistic design and photo enhancement. An ideal image color transfer algorithm should keep the scene of the source image and accurately apply all the dominant color styles of the target. In many cases, the spatial distribution of the target image should also be presented in the result.

In automatic color transfer techniques such as [RAGS01, TJT05, PKD07, PR11], they extract various color statistics from both images and establish specific mapping functions, such that the re-colored source scene shares similar color theme with the target. In many cases, the low level color statistic information cannot fully describe the real content of the image and some regions may be incorrectly colored during the mapping process. To avoid such visual artifacts, some interactive methods utilize user-specified color strokes to ensure correct color transfer between different image regions, therefore achieve desirable results [LWX07, AP10, LSZ12]. However, it can be difficult for an amateur to specify those strokes in a compatible and perceptually coherent manner. Another problem with previous methods is that they do not consider the spatial distribution characteristics when transferring colors, which might damage the specific visual coherence (Figure 1(d), 1(e)) or even lose important color styles of the target image (Figure 2(d), 2(e)) in the result.

To accurately transfer colors between images with minimal manual labor, we present a system which benefits from the image content analysis. The basic inspirations of our method is that the transfer process should be performed between image regions with similar semantic contents, especially when the obviously different color patterns appear in different regions of the target image (Figure 1). Moreover, previous methods will easily fail if the number of dominant color styles of the source image is much less than the one of the target image (Figure 2). To address those problems, in this paper we present a content-based color transfer technology. We first perform scene analysis to detect the subject areas (salient objects or clarity areas) and recover the background surface layout (sky, ground, vertical). With this content information, the system transfers colors between the corresponding regions on a super-pixel level. For example, our system will separately transfer the sky colors of the target to the sky in the source (if exists) without affecting the color patterns of other regions. For such corresponding regions, we present a novel local color transfer algorithm that integrates spatial distribution of target dominant colors

† Email: Weiming.Dong@ia.ac.cn
‡ Email: Xiaopeng.Zhang@ia.ac.cn

(a) Source Image (b) Target Image (c) Our Result (d) [PR11] (e) [LE07]

Figure 1: *Our method firstly automatically analyzes the scene content structure and divide it into different regions such as salient objects, sky and ground. After that, a novel distribution-aware color transfer process is performed on the corresponding pairs (e.g. transfer the color style from target sky to source sky). Our method well preserves the color spatial distribution and the visual coherence in the result (c). We can see that the color layer effect of the target sky is lost in (d) and (e).*

(a) Source Image (b) Target Image (c) Our Result (d) [TJT05] (e) [PKD07]

Figure 2: *Our result accurately presents the spatial distribution of the colorful tulips in the target image.*

in an optimization framework. We also develop a new progressive transfer scheme to integrate the advantages of both global and local transfer algorithms. That can also avoid the over-segmentation artifact which often occurs in the results of local color transfer methods. As a result, all the obvious color styles, as well as their corresponding spatial distribution in the target image, can be reproduced in the output image without damaging scene contents. To the best of our knowledge, our method is the first trial to challenge this kind of problems in color transfer research, especially for the example like Figure 2. Moreover, our image color transfer approach can be efficiently extended to re-color videos to generate spatially varied color patterns.

2. Related work

Automatic color transfer algorithm was firstly presented by Reinhard et al. [RAGS01]. The pixel color values of the source image are transformed by matching the global color mean and standard deviation of the target in an uncorrelated color space $l\alpha\beta$. Chang et al. [CSUN05] proposed a perception-based scheme for transferring colors based on the basic color categories, which are derived through a psychophysical experiments. This framework was extended to stylize the video frame sequences in [CSN07]. Neumann and Neumann [NN05] used 3D histogram matching to transfer the color style of a source image into an arbitrary given target image having a different distribution. Tai et al. [TJT05] modeled the image color distribution as Gaussian mixture models and solved the regional color transfer problem by expectation maximization. Pitié et al. [PKD07] proposed automated color grading method by transferring an N-dimensional probability distribution function to another. Dong et al. [DBZP10] extracted the same number of dominant colors (DCD) from the two input images and find an optimal one-to-one mapping between the dominant color sets. Pouli and Reinhard [PR11] presented a histogram reshaping technique which allowed users to select how well the color palette of the source image should be matched to that of the target. To further optimized the scene details in the result, Xiao and Ma [XM09] employed poisson compositing [PGB03] to preserve the gradients of the source image. However, as shown in Figure 2, the commonly used color transfer algorithms fail to generate a satisfied result if the number of source dominant colors is much less than the target one, especially when the color styles are transferred between images having similar contents.

Interactive methods made color transfer more controllable in the photo editing process. Luan et al. [LWX07] introduced a brush by which the user specifies the source and destination image regions for color transfer. Wen et al. [WHCO08] used strokes in both the source and target image for multiple local transfers. An and Pellacini [AP10] used a nonlinear constrained parametric model to transfer colors between image regions defined by pairs of strokes. Liu et al. [LSZ12] presented ellipsoid color mixture map to realize selective color transfers. Compared with those interactive frameworks, our image content structure extraction technology can help to maximize the automation.

The concept of color transfer was extended to other applications. Yang et al. [YP08] presented a mood-transferring method between color images by using histogram-matching to preserves spatial coherence. Xue et al. [XWT*08] tried

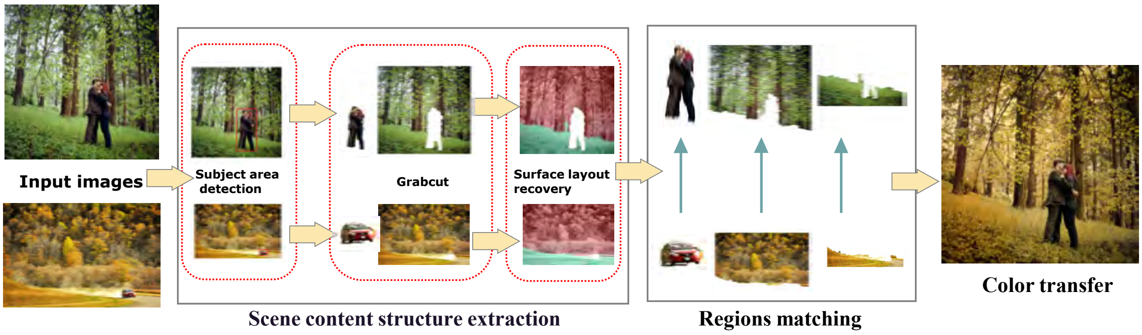


Figure 3: *System pipeline. We separately transfer the color patterns between the corresponding semantic regions.*

to transfer the weathering effects between the input images instead of transferring color. Murray et al. [MSMP11] modified the color content of an input image given only a concept specified by a user in natural language. Data-driven methods were presented for image color enhancement by quantifying image-color theme difference [WYW*10] and non-rigid dense correspondence [HSGL11]. Recently Chia et al. [CZG*11] proposed a semantic colorization system that leverages the image content on the internet.

3. Overview

A block diagram of our system framework is shown in Figure 3. To transfer the color patterns between the source and target image, we first detect the subject area(s) in both images and recover the surface layout of the backgrounds (Section 4). The content information is then used to guide the color transfer process. A novel distribution-aware color transfer algorithm is presented to transfer the color styles between the corresponding regions of the source and target image (Section 5). The spatial distribution and visual details of the target colors can also be preserved in the result. Furthermore, an extension of the method is proposed to transfer spatial color distribution features to the video clips (Section 6).

4. Scene content analysis

Understanding the content structures of the input images will facilitate the color transfer process to generate a natural looking result. For each input image, we detect the subject area(s) and recover the surface layout of the background.

4.1. Subject Area Detection

The detection of subject area depends on the image content. In our system, we set a composition based detection method as the default operator to automatically extract the subject areas from the input images. The user can also choose another two operators to get a more accurate result when the input images contain defocus effects or faces.

(a) Objects (b) Clarity (c) Face (d) Layout

Figure 4: *Image scene content analysis. In (d) we use different colors to indicate different semantic regions, here green for support (ground) and red for vertical surfaces.*

Salient object detection For normal images with clear backgrounds, the salient objects are the subject areas. We first use a composition based method [FWT*11] to detect the windows which contain the salient objects. Based on these windows, we then apply Grabcut method [RKB04] to extract the contour of the objects. Figure 4(a) shows a salient object detection result by our method.

Clarity based detection For a low depth of field image or close-up photo with blurred background, we always treat the clear area(s) as the subject content. We provide a clarity based method to help obtain the clear area(s) in an image. We first use the method in [LT08] to obtain a clarity mask which labels each pixel as clear or blur. Then the image is over segmented by applying mean shift segmentation [CM02]. We label a super-pixel as clear if more than half of its pixels are labelled as clear in the clarity mask. An example of our method is shown in Figure 4(b). Disjoint clear areas are treated as different subject areas in our system.

Face detection Faces are special salient objects in an image, especially for portrait photos. We provide a face detector [XZST07] to help locate the windows which contain the faces. Graph-cut [RKB04] is also employed to extract the contours. Figure 4(c) shows a face detection result.

4.2. Background Layout Recovery

After detecting the subject area(s) in an image, we further segment the background into different semantic regions. We use the method in [HEH07] to recover the surface layout of the background. As a result, every pixel in the image is categorized into one of three classes: *support (ground)*, *vertical*, and *sky*. Based on this categorization, we get a semantic label mask that indicates the pixels' types (e.g. $label = 1$ means *ground*). Specifically, we treat the pixels with the same label as a whole semantic region in our following transfer method, no matter if these pixels are locate in different disjoint areas or not (Figure 5). This treatment ensures that the disjoint sub-regions with identical semantic label are formed together in the transfer process so as to generate an integrated view in the result, which can avoid the possible discontinuity of the color patterns. Therefore, the background image is segmented into three parts: *sky region*, *ground region*, and *vertical region*, as shown in Figure 4(d) as examples.

In most cases, the segmentation results obtained from both the subject detection and the surface layout recovery are sufficient for our color transfer framework. If the automatic segmentation is not satisfied, paint selection technique [LSS09] is used to refine the segmentation results.

(a) Source (b) Target (c) Our result (d) [PKD07]

Figure 5: *The disjoint sky (the two sub-regions outlined by red boundaries) in (a) is treated as a whole semantic region in the matching and transfer process.*

4.3. Semantic Regions Matching

After extracting the content structures of the input images, we match the semantic parts together and transfer the color styles between each pair. For the image backgrounds, we separately match the source sky (vertical, ground) to the corresponding target sky (vertical, ground) if they both exist according to the label masks (e.g. Figure 4(d), "red" vertical is matched to "red" vertical). The matching is straightforward since the semantic regions in the background are unique.

For the salient objects, we can directly construct the matching if there is only one object in both of the input images. However, it will be more complex if there are multiple salient objects in the images. Denote N_s and N_t as the number of objects extracted from the source and target image separately. If $N_s > N_t$, we first randomly sample N_t objects from the source image and then find a matching target object t_i for each sampled object s_i by optimizing the metric:

$$\min_{\{t_i\}} \sum_{i}^{N_t} \|c_{s_i} - c_{t_i}\|, \qquad t_i \neq t_j \quad if \quad s_i \neq s_j, \tag{1}$$

where $c_{s(t)_i}$ is the centroid of the objects in relative coordinates (normalized to $[0,1]$ according to its position in the image). We use the one-to-one EMD mapping framework in [DBZP10] to optimize the metric. For those unsampled source objects, we set the matching target objects to be the same as their nearest sampled objects in the source image.

For the case $N_s \leq N_t$, we instead sample N_s objects from the target image and reformat Equation (1) as $\min_{\{t_i\}} \sum_{i}^{N_s} \|c_{s_i} - c_{t_i}\|$ to find the optimal matching target objects for all of the source objects. Since we use a location constraint, the object matching can guarantee our final color transfer result to preserve the spatial distribution of the target colors in object level. A multiple objects transfer result produced by our method is shown in Figure 10.

Based on these matched semantic regions, we perform our color transfer operations (Section 5) for each pair separately. Moreover, we consider the content structures of input images are different if there is any semantic region in either of the input images that cannot find a matching region in the other image. To solve this problem, we first globally transfer the luminance of the target image to the source by using the method in [RAGS01]. Based on this result, we then perform our color transfer operations between the existing pairs and leaving the other regions unchanged, as shown in Figure 16.

5. Distribution aware color transfer

Utilizing the content structure information of input images, we separately transfer the color style of each semantic region in the target image to the corresponding one in the source. We present a novel distribution-aware color transfer algorithm to preserve the spatial distribution of target colors in the result. For each matched pair, we first segment the regions into super-pixels, where the color and spatial information are used together to measure the pixels' similarity during the segmentation process. Since the color spatial distribution is represented in the segmentation results, we consider it as an additional constraint in the color transfer process to preserve the color spatial consistency. CIE $L^*a^*b^*$ color model is used in all our experiments.

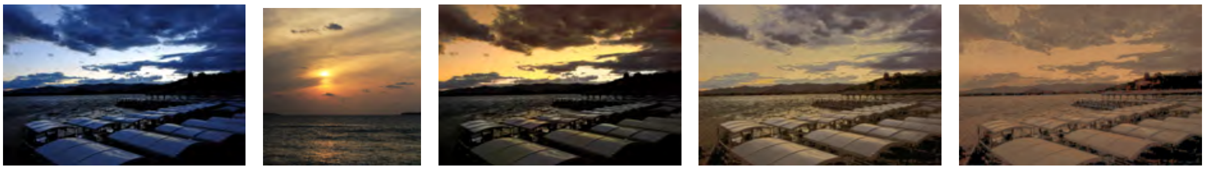
5.1. Segmentation

For each region (composed by the pixels with the same semantic label got in Section 4), we use Mean Shift [CM02] to segment it into super-pixels, which will keep the spatial consistency in the segmentation process. The following features are calculated for each super-pixel P_i:

$$P = \{p, \vec{c}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbb{S}\}, \tag{2}$$

(a) Source Image (b) Target Image (c) Our Result (d) [PR11] (e) [PKD07]

Figure 6: *Our method better preserves the global brightness and local contrast of the original scene. The target color patterns are also better depicted in the result (no blue color bleeding on the buildings and street). The reason is that the color styles in the target sky are only transferred to the source sky and do not affect the other regions (e.g. the roads in (d) and (e)).*



(a) Source Image (b) Target Image (c) Our Result (d) [PR11] (e) [TJT05]

Figure 7: *Our method avoids the foggy artifacts in (d) and (e), while preserving both the global theme (nightfall in a sunny day) and the local color distribution (the spatial layer of the colors in the sky and the shine on the boat).*

where p is the percentage of pixels in the image, $\vec{c}$ is the centroid of the super-pixel, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the color mean and standard deviation value of the pixel colors, $\mathbb{S}$ is a set containing all the pixel colors in a super-pixel. Since the corresponding semantic regions in source and target images may locate in different positions and appear in different geometrical shapes, we calculate their bounding boxes (denote $\boldsymbol{P}_{ul}$ as the upper left position) and align the regions by setting $\boldsymbol{P}_{ul}$ to be the origins of their local coordinate systems. We then set the centroid of each super-pixel as a relative position:

$$\vec{c} = \vec{c}_e - \boldsymbol{P}_{ul}, \tag{3}$$

where $\vec{c}_e$ is the exact centroid of the super-pixel. We normalize $\vec{c}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ all to the range $[0,1]$.

5.2. Soft mapping construction

Based on the segmented super-pixels, we build a connection between the color patterns of the matched source and target semantic regions. Unlike the the one-to-one mapping scheme in [DBZP10], we instead build a soft many-to-many mapping between the two super-pixel sets, which can generate more natural results in most cases. Denote f_{ij} as the probability of the ith source super-pixel maps to the jth target super-pixel, and c_{ij} as the mapping cost between them, the total cost of the mapping is optimized by the metric:

$$\min_{\{f_{ij}\}} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} f_{ij} c_{ij}, \tag{4}$$

where N_s denotes super-pixel number of the source image and N_t denotes target one. The probability f_{ij} is subject to the following constraints:

$$\sum_{j=1}^{N_t} f_{ij} = p_i^s, \quad i = 1,2,\ldots,N_s$$
$$\sum_{i=1}^{N_s} f_{ij} = p_j^t, \quad j = 1,2,\ldots,N_t$$

where p_i^s and p_j^t denote the percentage of pixels in the image corresponding to the super-pixel (labels s and t denote the source and target image respectively). We set a same percentage $p_i = \frac{1}{N}$ to all the super-pixels, which means that they have equal importance in the transfer process. In previous works [FK10, LE07, DBZP10], only color difference is used to measure the mapping cost. In order to preserve the spatial distribution of the target color styles, we modify the mapping cost function c_{ij} by adding the spatial location information as a constraint, which is formulated as:

$$c_{ij} = \exp\left(\frac{\|\mu_i^s - \mu_j^t\|^2}{\delta_c}\right) \cdot \exp\left(\frac{\|\vec{c}_i^s - \vec{c}_j^t\|^2}{\delta_s}\right), \tag{5}$$

where $\vec{c} = (x,y)$ is the super-pixel centroid (Equation (3)), $\mu = (\mu^L, \mu^a, \mu^b)$ is the color mean value, δ_c and δ_s are parameters to control the contributions of color and spatial features respectively. We set $\delta_c = 0.95$ and $\delta_s = 0.6$ as default.

We adopt the EMD framework [RTG00] to solve the linear program Equation (4). After that, we formulate the following transform functions to construct the mappings between the features of the source and target super-pixels:

$$\Phi(\mu_i^s) = \frac{\sum_{j=1}^{N_t} f_{ij}\mu_j^t}{\sum_{j=1}^{N_t} f_{ij}}, \qquad (6)$$
$$\Phi(\sigma_i^s) = \frac{\sum_{j=1}^{N_t} f_{ij}\sigma_j^t}{\sum_{j=1}^{N_t} f_{ij}}.$$

Apparently, the transform functions softly map each source super-pixel to multiple target super-pixels by a weighted average scheme. The weight f_{ij} is solved from Equation (4).

We illustrate our soft mapping scheme between the source and target super-pixels of the corresponding regions in Figure 8. There are 8 super-pixels in the source and 7 in the target, while the number of colors in the source is less than the one of the target. We can see that after optimizing Equation (4) each source super-pixel is mapped to the target ones by a weighted average formulation. The spatial constraint in Equation (5) helps to preserve the spatial distribution of the target color styles in the result.

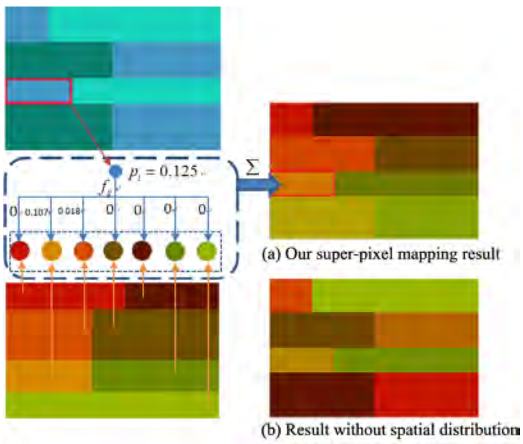


Figure 8: *Illustration of soft mapping between super-pixels. (b) is generated by ignoring the spatial constraint in Equation (5), so the spatial feature of the target colors is lost.*

We use soft boundaries for the source image to avoid the artifacts caused by segmentation. When transforming the pixel color $I(x,y)$, we also consider the influence from its neighbouring super-pixels. In the source image, denote P_i as the super-pixel which pixel $I(x,y)$ belongs to, the set of neighbouring super-pixels of a pixel $I(x,y)$ is defined as:

$$\mathbb{N}(x,y) = \{P_i\} \cup \{P_{i'} \mid \exists I(x,y) \in \mathbb{S}_i, I(x',y') \in \mathbb{S}_{i'} : |x-x'| + |y-y'| = 1\}. \qquad (7)$$

For each neighbouring super-pixel $P_k \in \mathbb{N}(x,y)$, we calculate the probability that the pixel color $I(x,y)$ belongs to it as:

$$p_k'(x,y) = \frac{1}{Z} D(I(x,y), P_k), \qquad (8)$$

where $Z = \sum_{P_k \in \mathbb{N}(x,y)} D(I(x,y), P_k)$ is the normalization factor. $D(I(x,y), P_k)$ is the similarity between the pixel color $I(x,y)$ and its neighboring super-pixel P_k. We adopt the bilateral filter to simultaneously smooth the color and spatial similarity:

$$D(I(x,y), P_k) = \exp\left(-\frac{\|I(x,y) - \mu_k^s\|^2}{\delta_c'}\right) \cdot \exp\left(-\frac{\|(x,y) - \vec{c}_k^{\,s}\|^2}{\delta_s'}\right).$$

We set $\delta_s' = 0.4$ and $\delta_c' = 0.95$ in all our experiments. Users can properly increase the range of neighbourhood in Equation (7) if the input images are large.

5.3. Pixel color transform

For an output pixel $I^o(x,y)$, we utilize the probability $p_k'(x,y)$ to smooth the contributions of the neighbouring super-pixels $\mathbb{N}(x,y)$, which is computed as

$$I^o(x,y) = \sum_k p_k'(x,y)\left(\frac{\Phi(\sigma_k^s)}{\sigma_k^s}(I(x,y) - \mu_k^s) + \Phi(\mu_k^s)\right), \qquad (9)$$

where $p_k'(x,y)$ is the probability that the pixel $I(x,y)$ belongs to the neighbouring super-pixel P_k (Equation (8)), $\Phi(\mu_k^s)$ and $\Phi(\sigma_k^s)$ are the new mapped mean and standard deviation of region P_k respectively (Equation (6)). Since the smoothing operation only involves the neighboring super-pixels, our algorithm is very efficient. In some cases, the local gradient of the source image may be not well preserved in the result and causes scene fidelity problem. To address this problem, we use our output as the intermediate image and apply the gradient-preserving method [XM09] to generate the final result, so as to preserve the scene details of the source image.

5.4. Progressive transfer

To be more robust, we build an iterative framework. The target color pattern is progressively transferred to the source image by modifying the minimum super-pixel size M_k in each step. After the scene content analysis, we first transfer the color styles between the source and target images at the largest scale by segmenting each content region into one super-pixel. The result is used as the source image in the next step. Then we progressively reduce the minimum super-pixel size at each iteration by setting $M_k = 0.2 \cdot M_{k-1}$ and update the pixel color with the following formulations:

$$I_k^{o'}(x,y) = (1-\alpha_k) \cdot I_{k-1}^{o'}(x,y) + \alpha_k \cdot I_k^o(x,y), \qquad (10)$$
$$\alpha_k = 1.0 - \frac{1.0}{1.0 + \exp(0.5 \cdot (k-1))},$$

where $I_k^o(x,y)$ is the result produced by Equation (9) with current super-pixel size, α_k is a weight parameter to control the contribution of each scale. In the first iteration ($k = 1$), the source image is used as the initial value of $I_{k-1}^{o'}(x,y)$. We update the resultant image based on this multi-scale framework until the super-pixel is small enough. In our experiment, we stop this iteration when $M_k < 300$. Users can properly increase the size of M_k if the input images are large.

(a) Source Image (b) Target (c) Result ($k=1$) (d) Result ($k=2$) (e) Result ($k=3$)

(f) Final Result ($k=4$) (g) Without progressive (h) Without soft mapping (i) [TJT05] (j) [PR11]

Figure 9: *The iterative results of our progressive color transfer process. The upper right images in (c)-(f) are the corresponding super-pixels. In (h) we show the result without using soft mapping and soft boundaries, segmentation-like artifacts occur.*

Our progressive approach performs both global and local color transfer during the iteration, which can preserve the global color styles as well as the local details in the result. Figure 9(c)-9(e) shows the intermediate results of our progress transfer process. We can see that the initial results in 9(c) and 9(d) tend to access the target color styles globally, while the final fine scale result 9(f) preserves the local details. Since this progressive framework enables the finer scale transfer to begin with a source image that has the similar global color styles with the target, it can also avoid undesirable over-segmentation artifacts that often occurs when color transfer is performed only in a fine scale. For example in Figure 9(g), over-segmentation artifacts occur in the left haystack and lower part of the sky, which is inconsistent with smooth appearance in the source image. Compared with the results in 9(i) and 9(j), our method better preserves the color patterns of the target image by presenting the sunset glow effect mainly in the middle part of the result. The sunset glow excessively appear in [TJT05] and is almost lost in [PR11].

6. Extension to videos

In this section, we extend our framework to videos. An intuitive way is applying the color styles in the target image to the source video frame by frame. However, this direct operation will cause a perceptible flicker due to the motions in the frame sequences. Moreover, it's more reasonable to preserve the spatial distribution of the target color styles in the whole video scene rather than in each of the single frame.

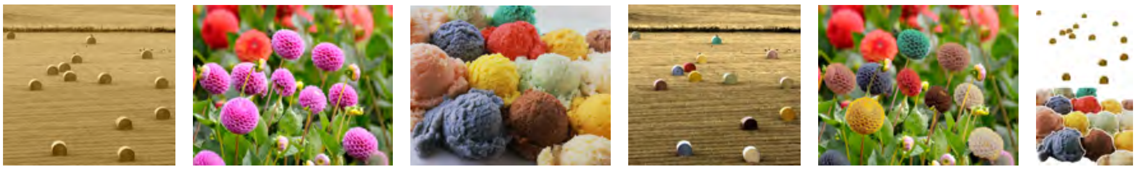
In order to keep the color coherence of the video frame sequences during the transfer process, we first stitch the video frames and restore the scene structure of the video. The SIFT-based stitching algorithm [BL07] is used to create a panoramic image by combining the source video frames. Then we use our color transfer framework to apply the target color styles to the panoramic image. Since there is a projective transformation between each of the video frame and the panoramic image, we can update the frame's color depending on the transferred panorama as follows:

$$\begin{aligned} I^o(x,y,t) &= \sum_k p'_k(x',y')\left(\frac{\Phi(\sigma^s_k)}{\sigma^s_k}(I(x,y)-\mu^s_k)+\Phi(\mu^s_k)\right) \\ (x',y',1)^{\boldsymbol{T}} &= \boldsymbol{H}_t \cdot (x,y,1)^{\boldsymbol{T}} \end{aligned} \tag{11}$$

where $I(,t)$ is the tth image from the source video frame sequence, $\boldsymbol{H}_t$ is the homography matrix representing the projective transformation between the tth image and the panorama, $(x',y',1)^{\boldsymbol{T}}$ is the corresponding homogeneous coordinates in the panoramic image. This function guarantees that the overlapping pixel between two successive frames will present in coherent color styles since they are projected to the same panoramic super-pixel. The examples of video color transfer are shown in the supplemental video, we can see that the color spatial distribution of the target image is separately transferred to different frame sequences to make a coherent visual effect.

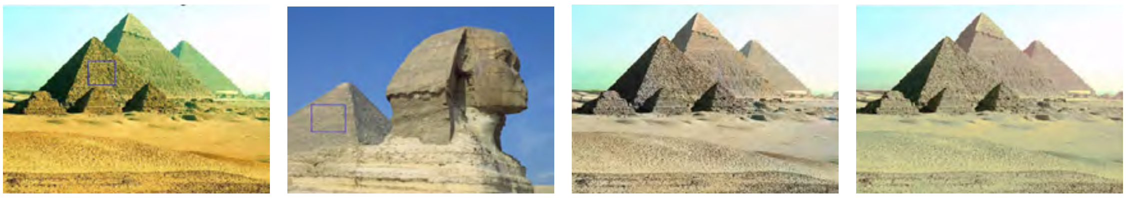
7. Results and discussions

We implement our method on a machine with an Intel Core 7 Quad 2.66GHz processor and 4GB RAM. The amount of time of the subject area detection and background surface layout is typically about 20-30 seconds in total for the images (from 500×333 to 500×500) used in this work. The layout recovery results are generated by the code subscribed by the authors of [HEH07]. We achieve real-time perfor-



(a) Source Image ♯1 (b) Source Image ♯2 (c) Target Image (d) Our Result ♯1 (e) Our Result ♯2 (f)

Figure 10: *The color spatial distribution of the ice cream in (c) is transferred to both the haystacks in (a) and the flowers in (b). We use Paint Selection to extract the haystacks in (a) and separate the ice creams in (c), as shown in (f).*



(a) Source Image (b) Target Image (c) Our Result (d) [LSZ12]

Figure 11: *Our method automatically generate more natural result without any user interaction. The sand in our result depicts the color of the bottom part of the Sphinx in (b), without causing the color distortion artifact in (d).*

(a) Source Image (b) Target Image (c) Our Result (d) [TJT05] (e) [LE07]

Figure 12: *The two buildings separately present the color styles of the two trees in (b). The color spatial information is lost in (d) while the red style is disappeared in (e). Our result also better preserves the brightness and contrast of the scene than (d).*

(a) Source Image (b) Target Image (c) Our Result (d) [PR11] (e) [XM09]

Figure 13: *Our result better preserves both the target color pattern structure and the source scene details.*

mance at our distribution-aware color transfer process after the scene content analysis.

Figure 1 shows a typical example that benefits from our method. The color features of the salient objects (the man and the dog), the sky and the ground of the target image are separately transferred to the corresponding regions of the source image. This scheme is very helpful in accurately presenting the desired visual appearance in the resultant image.

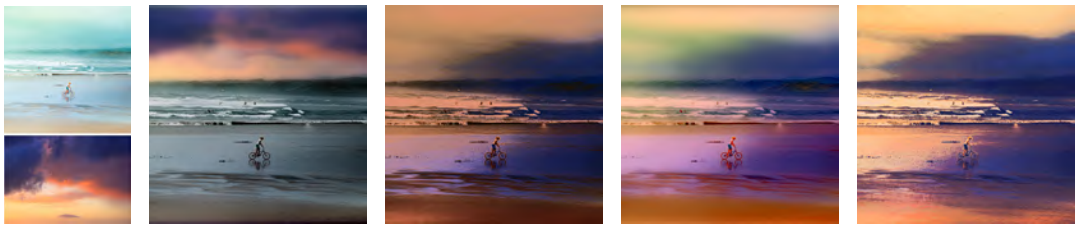
We compare our method with some commonly used color transfer algorithms in Figure 2, 6, 7(a), 12-14. The results of [TJT05] are generated by the code subscribed by the authors of [XM10] (`http://ia.cs.colorado.edu/~wxu/color_correction.htm`). The results of [PKD07], [LE07] , [XM09] and [PR11] are generated from codes which are subscribed by the authors. Results show that our method can both re-produce the local details and global spatial distribution of the target color pattern, while

(a) Source Image (b) Target Image (c) Our Result (d) [PR11] (e) [PKD07] (f) [XM09]

Figure 14: *The color styles of the faces are separately transferred with the help of our face detection operator.*

(a) Source Image (b) Target Image (c) Our Result (d) [PR11] (e) [PKD07]

Figure 15: *Comparison of the color transfer results. We integrate the content information also to the other methods.*

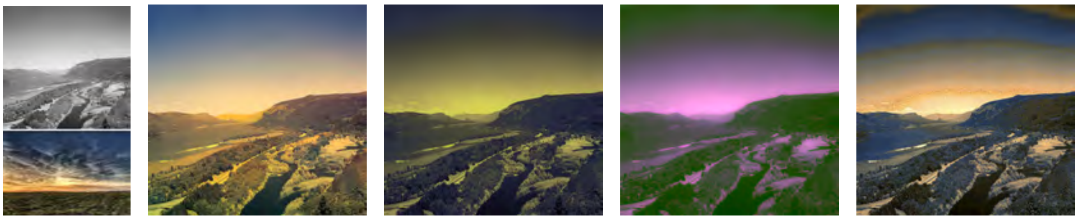


(a) Input (b) Our Result (c) [PR11] (d) [XM09] (e) [PKD07]

Figure 16: *When the input content structures are different, we first globally transfer the luminance and then perform local transfer operation between the skies. The sky in our result is more consistent with the target and the real natural phenomena.*

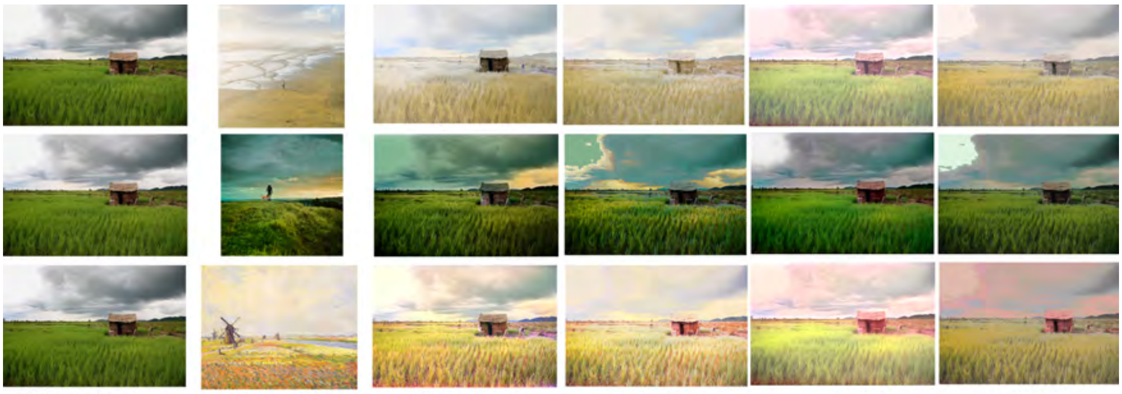
the other methods fail in most cases. The main reason is that the methods based on GMM [TJT05, LE07] or histogram reshaping [PKD07, PR11] usually tend to categorize the similar colors in the source image as a whole and transform them together during the transfer process, without considering their spatial information. Therefore, the similar colors of the source image will also appear to be a similar color style in the result. However, when the number of color styles in the source image is much less than the one of the target image, that scheme will cause the result to be assigned the average of the dominant color styles of the target image, as shown in Figure 1(d), 1(e), 2(d), 2(e), 21(d), 15(d) and 15(e). The integration of spatial constraint in our algorithm can solve the above problem. The similar colors are classified into different categories according to their geometric location, so they can depict different color styles of the target image in the result. On the other hand, our progressive framework combines the advantages of both the global and local color transfer schemes and avoids the over-segmentation artifact in the result which often occurs in most local transfer algorithms.

Figure 10 shows two examples containing multiple salient objects, we can see that the spatial distribution of the target object color styles are well represented in our results. In Figure 11, we compare our method with the manually selective color transfer method [LSZ12]. Figure 6 and Figure 12 exhibits two more examples which are transferred between two images with obvious different contents (city street and wild field). Results show that our method can nicely preserve integrity as well as spatial distribution of the target color styles, while the compared methods cannot. In Figure 14, we detect the faces from the portraits and separately transfer the color styles of the face and the background from the target to the source image. Both the color patterns are well preserved in our result. In Figure 15 we also integrate the content structure information into the other methods. However, with our distribution-aware color transfer algorithm, our result in Fig-



(a) Input (b) Our Result (c) [PR11] (d) [XM09] (e) [PKD07]

Figure 17: *Our method better depicts both the sky (especially the spatial layer of the colors) and the grass (the yellow-green effect caused by the sunlight) color patterns of (b) than other ones. Our result also avoids the segmentation-like artifact in (e).*



(a) Source Image (b) Target Image (c) Our Result (d) [PKD07] (e) [XM09] (f) [PR11]

Figure 18: *Results produced by one source image and different target images. Our algorithm is robust in all examples.*

ure 15(c) is still better than the other two images in preserving the details of the source sky and reproducing the color spatial distribution of the target grass. In Figure 17 we show an example of colourization.

In Figure 18, we use the results generated by using one source image and multiple target images to show the robustness of our method. We can see that our results well preserve the spatial distribution of the target color styles in all results, while also avoiding the sky color bleeding in [PKD07]'s results and color distortion in [XM09] and [PR11]'s results.

(a) Input (b) Our Result (c) [PR11] (d) [PKD07]

Figure 19: *In our result, color styles are separately transferred between clear and smooth regions.*

In Figure 19, we extract the clear regions from the close-up input images and separately transfer the color styles between the source and target foreground objects, as well as the blur backgrounds. The top parts of the flowers in the result image appear green styles because they inherit the color feature of the clear leaves in the target image, which is benefited from our distribution-aware color transfer scheme.

Table 1: *User Study*

	Ours	[PR11]	[TJT05]	[LE07]	[PKD07]	[XM09]
Fig. 1	65.3%	15.3%	5.77%	17.3%	23%	25%
Fig. 2	57.6%	1.92%	17.3%	11.5%	15.3%	13.4%
Fig. 6	61.5%	17.3%	5.77%	0%	11.5%	21.2%
Fig. 7	59.6%	1.92%	3.8%	23%	34.6%	5.77%
Fig. 9	50%	1.92%	9.6%	19.2%	28.5%	11.5%
Fig. 12	55.7%	0%	3.85%	25%	15.3%	21.1%
Fig. 13	75%	1.92%	1.92%	9.62%	17.3%	19.2%
Fig. 14	67.3%	0%	1.92%	32.6%	5.77%	23%
Fig. 15	50%	21.1%	5.77%	13.4%	15.3%	15.3%
Fig. 16	40.3%	11.5%	3.85%	28.8%	9.6%	26.9%
Fig. 17	80.7%	13.4%	0%	3.85%	17.3%	1.92%
Fig. 18-1	53.8%	3.8%	1.92%	40.3%	15.3%	1.92%
Fig. 18-2	61.5%	5.77%	13.4%	3.85%	25%	19.2%
Fig. 18-3	57.6%	3.85%	0%	23%	25%	9.6%
Fig. 19	71.6%	1.92%	0%	11.5%	15.3%	19.6%
Fig. 20	46.1%	3.85%	7.69%	25%	15.3%	36.5%
Fig. 21	48%	0%	11.5%	13.4%	48%	23%

To further evaluate our method, we perform a user study to

(a) Input (b) Our Result (c) Expected Result (d) [LE07] (e) [XM09]

Figure 20: *Our algorithm may generate an unnatural result if the scene content is not precisely extracted. (c) is generated by manually removing the road from the "ground" region of the target and only transfer the color styles of the grass to the corresponding "ground" region (the grass) of the source image.*

compare the results from different methods. 104 users from different ages and backgrounds attended the comparison of 17 sets of the color transfer images. All the stimuli are shown in the supplemental material. In the experiment, we show the source image, the target image, our result and the images of the competitors. For each group, the results are randomly displayed in two rows within one page. In order to avoid the interference of the similar quality results to the statistic data, we ask the participants to choose maximum two images they like from each group. Table 1 shows the statistics. Each row shows the percentages of our method and the competitors been chosen by the participants. From the statistics, our method generally outperforms all competitors.

(a) Input (b) Our result (c) [PKD07] (d) [XM09]

Figure 21: *The spatial feature of target flowers is preserved. However, different color styles may appear in a single flower.*

8. Conclusion and future work

We propose a novel color transfer scheme that utilizes subject area detection and surface layout recovery to minimize user effort and generate accurate results. A new local color transfer algorithm is presented, which integrates spatial distribution of target color styles in the optimization process. An iterative framework is proposed to increase the robustness of the method, which can ensure the transferring of both the global and local color styles, while also eliminating the over-segmentation artifacts. Our method can achieve a good and reasonable result even though the number of the source dominant color styles is much less than the target one. The spatial distribution of the target dominant color styles are also nicely preserved in the output image. We also extend our algorithm to handle video-image color transfer. The special color distribution of the target image can be reproduced in an panoramic-like video.

There are limitations of our content-based color transfer framework that we plan to investigate in future work. One is that the subject area detection methods may be inaccurate with fine-scale structure and unapparent boundary. For example in Figure 10(a) and 10(c) we need to use Paint Selection to help to extract some small haystacks and ice creams. The surface layout recovery method is not accurate enough for some examples which contain complex scene structures. As shown in Figure 20, the road and the grass in the target image are both categorized as "ground", so the color styles of the road are also transferred to the grass of the source image. However, the image semantic segmentation is still an open problem and we believe that our system will be more robust by using some more precise image understanding algorithms. Our distribution-aware color transfer scheme may break the wholeness of the clustered objects. As shown in Figure 21(b), sometimes different color styles appear in one single flower. Therefore, from the user study we can see that many participants also choose 21(d) as a favorite result. Some of the target colors may be lost in the result if the content of the source image is "less" than the target, especially when the salient objects in the source image is less than the ones of the target (e,g, Figure 10(e)). Another is that color transfer for background regions will not always exactly reproduce the spatial distribution of the target color styles in the result, especially when the corresponding region of the source image contains an apparently different spatial color distribution (an example is shown in the supplemental material). However, in most cases those results are actually more semantic-reasonable than simply duplicating the color distribution from the target image. Moreover, videos with large dynamic objects are generally less accurate than for static scenes, since the motion of dynamic objects will affect the accuracy of panoramic stitching. Additionally, we also plan to extend our algorithm to incorporate with internet photos which can subscribe more information to the input images.

Acknowledgement

We thank anonymous reviewers for their valuable input. We thank Tania Pouli for helping to make some results. This work is supported by National Natural Science Foundation of China under No. 61172104, 61271430, 61201402 and 61202324, by Beijing Natural Science Foundation (Content-Aware Image Synthesis and Its Applications, No.4112061), and by SRF for ROCS, SEM.

References

[AP10] AN X., PELLACINI F.: User-controllable color transfer. *Computer Graphics Forum 29*, 2 (2010), 263–271. 1, 2

[BL07] BROWN M., LOWE D. G.: Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision 74*, 1 (August 2007), 59–73. 7

[CM02] COMANICIU D., MEER P.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5 (May 2002), 603–619. 3, 4

[CSN07] CHANG Y., SAITO S., NAKAJIMA M.: Example-based color transformation of image and video using basic color categories. *IEEE Transactions on Image Processing 16*, 2 (2007), 329–336. 2

[CSUN05] CHANG Y., SAITO S., UCHIKAWA K., NAKAJIMA M.: Example-based color stylization of images. *ACM Trans. Appl. Percept. 2*, 3 (2005), 322–345. 2

[CZG*11] CHIA A. Y.-S., ZHUO S., GUPTA R. K., TAI Y.-W., CHO S.-Y., TAN P., LIN S.: Semantic colorization with internet images. *ACM Trans. Graph. 30*, 6 (2011), 156:1–156:8. 3

[DBZP10] DONG W., BAO G., ZHANG X., PAUL J.-C.: Fast local color transfer via dominant colors mapping. In *ACM SIGGRAPH ASIA 2010 Sketches* (New York, NY, USA, 2010), SA '10, ACM, pp. 46:1–46:2. 2, 4, 5

[FK10] FREEDMAN D., KISILEV P.: Computing color transforms with applications to image editing. *J. Math. Imaging Vis. 37*, 3 (July 2010), 220–231. 5

[FWT*11] FENG J., WEI Y., TAO L., ZHANG C., SUN J.: Salient object detection by composition. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (nov. 2011), pp. 1028 –1035. 3

[HEH07] HOIEM D., EFROS A. A., HEBERT M.: Recovering surface layout from an image. *Int. J. Comput. Vision 75*, 1 (October 2007), 151–172. 4, 7

[HSGL11] HACOHEN Y., SHECHTMAN E., GOLDMAN D. B., LISCHINSKI D.: Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph. 30*, 4 (Aug. 2011), 70:1–70:10. 3

[LE07] LALONDE J.-F., EFROS A.: Using color compatibility for assessing image realism. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (oct. 2007), pp. 1–8. 2, 5, 8, 9, 10, 11

[LSS09] LIU J., SUN J., SHUM H.-Y.: Paint selection. *ACM Trans. Graph. 28*, 3 (2009), 69:1–69:7. 4

[LSZ12] LIU S., SUN H., ZHANG X.: Selective color transferring via ellipsoid color mixture map. *J. Vis. Comun. Image Represent. 23* (Jan. 2012), 173–181. 1, 2, 8, 9

[LT08] LUO Y., TANG X.: Photo and video quality evaluation: Focusing on the subject. In *Proceedings of the 10th European Conference on Computer Vision: Part III* (Berlin, Heidelberg, 2008), ECCV '08, Springer-Verlag, pp. 386–399. 3

[LWX07] LUAN Q., WEN F., XU Y.-Q.: Color transfer brush. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 465–468. 1, 2

[MSMP11] 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* (New York, NY, USA, 2011), NPAR '11, ACM, pp. 167–176. 2

[NN05] NEUMANN L., NEUMANN A.: Color style transfer techniques using hue, lightness and saturation histogram matching. In *Proceedings of Computational Aesthetics in Graphics, Visualization and Imaging* (2005), pp. 111–122. 2

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Trans. Graph. 22* (July 2003), 313–318. 2

[PKD07] PITIÉ F., KOKARAM A. C., DAHYOT R.: Automated colour grading using colour distribution transfer. *Comput. Vis. Image Underst. 107*, 1-2 (2007), 123–137. 1, 2, 4, 5, 8, 9, 10, 11

[PR11] POULI T., REINHARD E.: Progressive color transfer for images of arbitrary dynamic range. *Computers & Graphics 35*, 1 (2011), 67 – 80. 1, 2, 5, 7, 8, 9, 10

[RAGS01] REINHARD E., ASHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Comput. Graph. Appl. 21*, 5 (2001), 34–41. 1, 2, 4

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23* (August 2004), 309–314. 3

[RTG00] RUBNER Y., TOMASI C., GUIBAS L. J.: The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision 40*, 2 (2000), 99–121. 5

[TJT05] TAI Y.-W., JIA J., TANG C.-K.: Local color transfer via probabilistic segmentation by expectation-maximization. In *Proceedings of CVPR'05 - Volume 1* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 747–754. 1, 2, 5, 7, 8, 9, 10

[WHCO08] WEN C.-L., HSIEH C.-H., CHEN B.-Y., OUHYOUNG M.: Example-based multiple local color transfer by strokes. *Computer Graphics Forum 27*, 7 (2008), 1765–1772. 2

[WYW*10] WANG B., YU Y., WONG T.-T., CHEN C., XU Y.-Q.: Data-driven image color theme enhancement. *ACM Trans. Graph. 29*, 6 (2010), 146:1–146:10. 3

[XM09] XIAO X., MA L.: Gradient-preserving color transfer. *Computer Graphics Forum 28*, 7 (2009), 34–41. 2, 6, 8, 9, 10, 11

[XM10] XU W., MULLIGAN J.: Performance evaluation of color correction approaches for automatic multi-view image and video stitching. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 0* (2010), 263–270. 8

[XWT*08] XUE S., WANG J., TONG X., DAI Q., GUO B.: Image-based material weathering. *Computer Graphics Forum 27*, 2 (2008), 617–626. 2

[XZST07] XIAO R., ZHU H., SUN H., TANG X.: Dynamic cascades for face detection. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (oct. 2007), pp. 1 –8. 3

[YP08] YANG C.-K., PENG L.-K.: Automatic mood-transferring between color images. *IEEE Comput. Graph. Appl. 28* (March 2008), 52–61. 2