Example-based Multiple Local Color Transfer by Strokes

Chung-Lin Wen[†] Chang-Hsi Hsieh[†] Bing-Yu Chen[‡] Ming Ouhyoung[§]

National Taiwan University

Abstract

This paper investigates a new approach for color transfer. Rather than transferring color from one image to another globally, we propose a system with a stroke-based user interface to provide a direct indication mechanism. We further present a multiple local color transfer method. Through our system the user can easily enhance a defect (source) photo by referring to some other good quality (target) images by simply drawing some strokes. Then, the system will perform the multiple local color transfer automatically. The system consists of two major steps. First, the user draws some strokes on the source and target images to indicate corresponding regions and also the regions he or she wants to preserve. The regions to be preserved which will be masked out based on an improved graph cuts algorithm. Second, a multiple local color transfer method is presented to transfer the color from the target image(s) to the source image through gradient-guided pixel-wise color transfer functions. Finally, the defect (source) image can be enhanced seamlessly by multiple local color transfer based on some good quality (target) examples through an interactive and intuitive stroke-based user interface.

Categories and Subject Descriptors (according to ACM CCS): I.4.3 [Image Processing and Computer Vision]: Enhancement

1. Introduction

Nowadays, digital cameras are very popular, and most people take many photos on their trips, reunions, etc. However, since most end users are not professional photographers, a user usually takes a lot of photos but eventually finds out that only a small portion of them are satisfactory. Indeed, with powerful commercial post-processing software, the experts could enhance these defect photos, but this task can be time-consuming. For those who are not skillful in editing photos, it might even be unachievable. Fortunately, due to the ease of taking photos, people may dispose of several photos of the same object or similar scene captured by using one or more cameras. Thus, it is easy to acquire some good quality photos of the same object or similar scene, and it is possible to use them for enhancing the defect photos.

The direct enhancement of the photos by using some post-processing tools may not be an easy task. For example, in the case of a backlighted photo, one may think that the user can use the brightness/hue adjustment tools as contained in ordinary image processing software. However, such tools usually can only be used to adjust the brightness/hue globally. If such tools are applied to achieve a brighter foreground, the background might be overexposed. Even if the foreground region in interest is carefully specified, the process is not only time-consuming but also may result in artifacts at the foreground and background border. On the other hand, if we can refer to the same object in other good quality photos, it would be much easier. Figure 1 shows three cases and our result along with their reference example(s).

In this paper, we propose an easy-to-learn and easy-to-use system for photo enhancement. To use our system, no extra photographic equipment and knowledge is required and the users also do not have to change their habits in taking photos. What they only have to is to acquire some other good quality photos as targets and draw some strokes on the source and target photos to indicate the corresponding regions and the regions they want to preserve. Then, a multiple local color transfer operation is applied to transfer the color from the targets to the source photo through gradient-guided pixel-

[†] e-mail:{jonathan, isaddo}@cmlab.csie.ntu.edu.tw
[‡] e-mail:robin@ntu.edu.tw
[§] e-mail:ming@csie.ntu.edu.tw

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

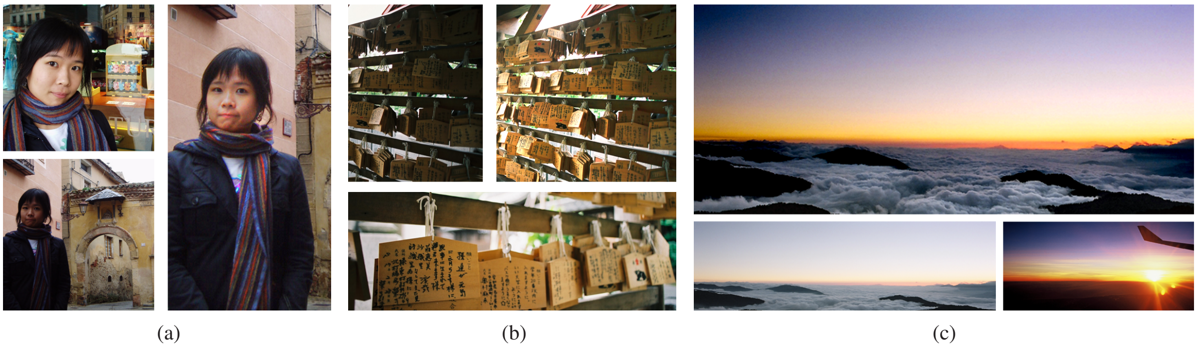


Figure 1: *Three photo enhancement examples. (a) The left two photos are captured at two different places at the same day. The upper one is used to enhance the lower one. The right photo shows the result. (b) The left upper and lower photos depict the same scene captured from different viewpoints and the lower one is used to enhance the left upper one. The right upper photo shows the result. (c) The lower two photos are captured on a mountain and from an airplane, respectively, and the right one is used to enhance the left one. The upper photo shows the result.*

wise color transfer functions. Thus, our contribution is to provide a new simple yet effective interactive multiple local color transfer system that provides an intuitive mechanism to enhance photos without tedious manual work.

2. Related Work

For transferring color from one image to another, Reinhard *et al.* [RAGS01] developed a simple yet effective method. Welsh *et al.* [WAM02] presented another global color transfer system which is applicable for most scenic photos. For photos that contain foreground objects of specific interest such as portraits of persons the system is not very suitable. Following research work [TJT05,CSN07] proposes to solve this issue by presenting a probabilistic approach to conduct local color transfer. However, the user still does not have enough direct control to specify the regions that should be modified and the colors to be transferred. The only way to fine-tune the result is to make use of a reference image then apply the algorithm again in a try-and-error fashion.

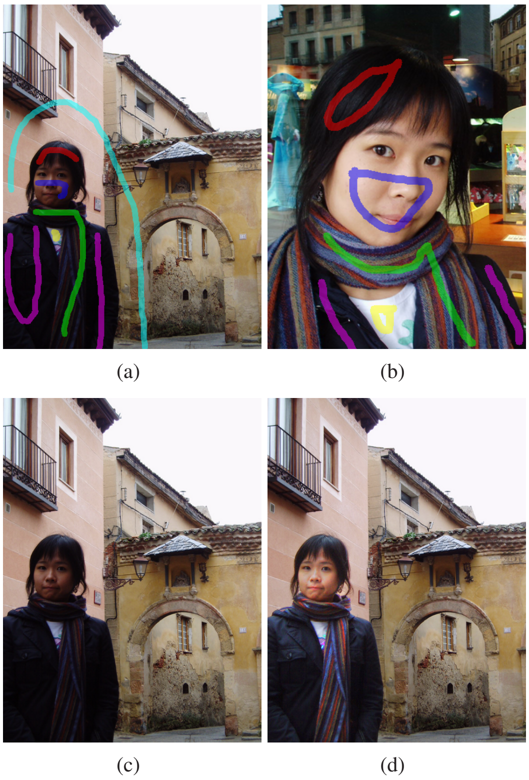
To supply more user control, Levin *et al.* [LLW04] presented a scribble-based color transfer method. With their user-interface, the user can specify the regions without providing an accurate segmentation. However, since the user has to choose the color for colorization by himself, without the hints that a reference image can provide, it is very difficult to acquire the color of the real photo or what the user expected. Besides this, under the assumption that neighboring pixels with similar intensities should be colored by similar colors, the system can only be used to color simple textures with only one color, which constricts itself from being suitable in many real cases, such as the scarf in Figure 1 (a).

On the other hand, Lischinski *et al.* [LFUS06] proposed a scribble-based local parameter editing tool. By using some strokes, the user can divide the image into several regions and locally adjust parameters such as brightness. However, in some cases the "feeling" of a photo is far too subtle to be adjusted by separate parameter. In our system we want to transfer the "feeling" by referring to one or more other images without adjusting individual parameters. In contrast, Bae *et al.* [BPD06] proposed a system that enables the user to transfer the "feeling" from one example to a target image, but it lacks of the ability of local control which our system provides.

Several methods have been proposed to enhance images by combining multiple photographs of the same object in one image that has a better quality. Jia *et al.* [JSTS04] proposed a method that uses a pair of photos taken with different exposure conditions to enhance one of them with motion blur. Similarly, Eisemann and Durand [ED04], Petschnigg *et al.* [PSA*04], and Agrawal *et al.* [ARNL05] presented other approaches to achieve an enhanced photo by combining flash and non-flash photo pairs. However, these methods usually need extra equipment, such as a tripod or a high-quality flash lamp. Furthermore, there are various prerequisites for taking the photos, i.e., the user needs to know what methods he will apply after taking the photos, and sometimes professional photographic knowledge is also required.

3. Stroke-based User Interface

The goal of our system is to enable the user to easily enhance a defect photo by referring to other good quality photos. Hence, an interactive and intuitive stroke-based user interface is provided for specifying corresponding regions on both the source and target photos. Besides this, the user can draw a stroke only on the source photo to preserve a region as the example shown in Figure 8. Figure 2 (a) and (b) show



(a) (b) (c) (d)

Figure 2: *(a) Strokes on the source photo. The light blue stroke was drawn for preserving the background, since there is no such stroke on the target photo (b). (b) Corresponding strokes on the target photo. (c) The source photo. (d) Our result. The background is preserved completely, and the scarf is successfully recovered.*

some strokes drawn by the user. The color of the strokes indicates correspondence between the source and target photos.

After the user has drawn some strokes, we first analyze them. We first collect the pixels under the strokes that specify the regions to be preserved (background) and to be edited (foreground) on the source photo to build the pixel sets $P^s_{\mathcal{B}}$ and $P^s_{\mathcal{F}}$, respectively, where $\mathcal{B}$ and $\mathcal{F}$ are the labels used to denote the background and foreground regions, respectively. In addition to this, we use p to denote a pixel. c_p and l_p are the color and label of pixel p and I_s and I_t are the source and target photos.

4. Background Preservation

Before applying the color transfer, in order to avoid unexpected modification of the background regions, we conduct a background preservation process to segment the source image into foreground and background regions based on the strokes $P^s_{\mathcal{F}}$ and $P^s_{\mathcal{B}}$ as described in Section 3. The background preservation process is performed by an improved version of the graph cuts algorithm [BVZ01]. It is used to minimize the following energy function:

$$E(l_p) = \sum_{p \in I_s} E_c(p)E_p(p) + \alpha \sum_{p,q \in I_s, q \in N_p} E_s(p,q), \qquad (1)$$

where l_p is a possible labeling of pixel p and N_p denotes the neighboring pixels of p, i.e., $q \in N_p$ means that p and q are neighboring pixels.

$E_c(p)$ is the color term which is used to measure the conformity of the color of pixel p and is defined in the same way as in most of the previous work [WYC*06, RKB04, LSS05]. The foreground strokes $P^s_{\mathcal{F}}$ and the background ones $P^s_{\mathcal{B}}$ are used to build 3D GMMs (Gaussian Mixture Models) to describe their color distributions. These are used to estimate whether the color of a pixel p is closer to the foreground or the background region.

$E_s(p,q)$ is the smoothness term which is intended to maintain the edge in the image. It uses the L^2-norm distance of the neighboring pixels p and q in $L^*a^*b^*$ color space to measure the smoothness of the two pixels. If the color distance of the two pixels is small, it aims for making the labels of two pixels as same as possible. Otherwise, two different labels can be assigned to the two pixels.

Besides the color and smoothness terms as used in the traditional graph cuts algorithm, we introduce a new position term $E_p(p)$ in Eq. (1), which utilizes the spatial information specified by the strokes in order to avoid discontinuous segmentation due to similar colors in different regions of the image as shown in Figure 3 (a). If a pixel is closer to the foreground/background strokes, it is more likely that it should belong to the foreground/background regions. Furthermore, if a pixel lies between the foreground and background strokes, it means that the user expects the system to decide where to place the border. We use the following equation to model this:

$$E_p(p) = \frac{1}{2}\mathrm{sign}(r) \times |r|^n + 0.5, \qquad (2)$$

where $r \in [-1,1]$ indicates the normalized distance difference from pixel p to the foreground or background strokes $P^s_x, x \in \{\mathcal{F}, \mathcal{B}\}$ and the opposite ones $P^s_{\bar{x}}$ which is defined as:

$$r = \frac{Dist(p, P^s_x) - Dist(p, P^s_{\bar{x}})}{Dist(p, P^s_{\mathcal{F}}) + Dist(p, P^s_{\mathcal{B}})},$$

and the distance function $Dist(p, P^s_x)$ is defined as:

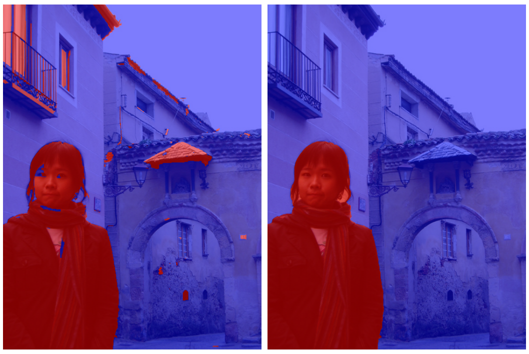
$$Dist(p, P^s_x) = \min_{p'} \|p - p'\|, \forall p' \in P^s_x.$$

Moreover,

$$n = a\exp(-b\frac{Dist(p, P^s_{\mathcal{F}}) + Dist(p, P^s_{\mathcal{B}})}{\max(w_s, h_s)})$$

is used to decide on the importance of the spatial information. w_s and h_s denote the width and height of the source image I_s, and are used to normalize the distance functions.

With the new position term $E_p(p) \in [0,1] \propto r$, we achieve



(a) (b)

Figure 3: *Background preservation result for Figure 2 (a). (a) The result of progressive cut [WYC*06] in the first step. Since the color of the wall is similar to part of the face, the shadow looks like the clothing, and the T-shirt is white, the result shows some errors. (b) Our editing (foreground) regions, which are obtained in one step.*

much cleaner segmentation results as shown in Figure 3 (b), and the background regions can be preserved precisely according to the background strokes drawn by the user. Finally, after the graph cuts process, each $p \in I_s$ is labeled by one $l_p \in \{\mathcal{F}, \mathcal{B}\}$.

5. Multiple Local Color Transfer

Our approach for multiple local color transfer is to set a suitable local (pixel-wise) color transfer function for each pixel. Since the color transfer is operated in the $l\alpha\beta$ color space, it can be performed separately in the three channels. The following description refers to only one of the channels. The same process is applied to the other two channels in the same way.

Just like in the original global color transfer method [RAGS01], we treat the pixel-wise local color transfer functions (Section 5.1) as three linear processes: shifting, scaling and then shifting again denoted by $u(p)$, $f(p)$ and $v(p)$ for updating the pixel $p \in I_s$, respectively. Then, the gradient of the original source image I_s is used to improve the pixel-wise local color transfer functions and obtain the functions (Section 5.2) $\hat{u}(p)$, $\hat{f}(p)$ and $\hat{v}(p)$. Finally, the multiple local color transfer is defined as:

$$c'_p = (c_p - \hat{u}(p))\hat{f}(p) + \hat{v}(p), \quad \forall p \in I_s. \tag{3}$$

5.1. Pixel-wise color transfer function

By using the user's strokes, the source image can be divided into two regions: the regions to be edited (foreground) and to be preserved (background). For the edit regions, since there are corresponding strokes P_j^s and P_j^t on both of the source and target images, we first build the Gaussian color model pairs $G_j^s(\mu_j^s, \sigma_j^s)$ and $G_j^t(\mu_j^t, \sigma_j^t)$ by using corresponding strokes with the same color (label) $j \in \mathcal{F}$, respectively, where μ_j is the mean and σ_j is the standard deviation in the Gaussian color model G_j, so there are $|\mathcal{F}|$ local color transfer functions. Furthermore, we also build the background Gaussian color model $G_{\mathcal{B}}^s(\mu_{\mathcal{B}}^s, \sigma_{\mathcal{B}}^s)$ for the preservation (background) regions on the source image based on the preservation (background) strokes $P_{\mathcal{B}}^s$.

Then, we need to decide by which ratio a pixel should be influenced by each local color transfer function. We use the following equations to set pixel-wise constraints to accumulate the influences of each local color transfer function on each pixel $p \in I_s$:

$$u(p) = \begin{cases} \sum\limits_{j\in\mathcal{F}} C(c_p,j)\mu_j^s + \sum\limits_{j\in\mathcal{B}} C(c_p,j)c_p, & \text{if } l_p = \mathcal{F} \\ c_p, & \text{if } l_p = \mathcal{B} \end{cases} \tag{4}$$

$$f(p) = \begin{cases} \sum\limits_{j\in\mathcal{F}} C(c_p,j)\dfrac{\sigma_j^t}{\sigma_j^s} + \sum\limits_{j\in\mathcal{B}} C(c_p,j), & \text{if } l_p = \mathcal{F} \\ 1, & \text{if } l_p = \mathcal{B} \end{cases} \tag{5}$$

$$v(p) = \begin{cases} \sum\limits_{j\in\mathcal{F}} C(c_p,j)\mu_j^t + \sum\limits_{j\in\mathcal{B}} C(c_p,j)c_p, & \text{if } l_p = \mathcal{F} \\ c_p, & \text{if } l_p = \mathcal{B} \end{cases} \tag{6}$$

where $C(c_p, j)$ indicates by which ratio the color c_p should be influenced from the j-th local color transfer function and is defined as:

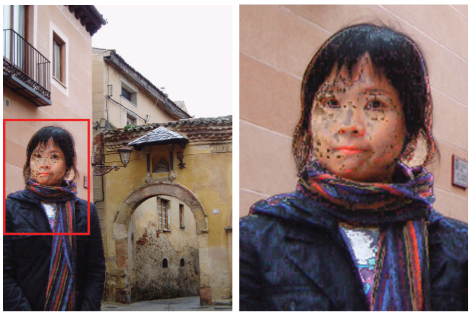
$$C(c_p, j) = \frac{P(c_p|G_j^s)}{\sum\limits_{j' \in \mathcal{F}\cap\mathcal{B}} P(c_p|G_{j'}^s)},$$

where $P(c_p|G_j^s)$ is a Gaussian probability distribution function which is used to estimate the probability that the pixel's color c_p belongs to the Gaussian color model G_j^s of the j-th stroke on the source image I_s.

In Eq. (4)∼(6), if a pixel p is in the edit (foreground) regions (i.e., $l_p = \mathcal{F}$), the weighted average of each function is set to p as a pixel-wise local color transfer function. Otherwise (i.e., $l_p = \mathcal{B}$), we set the shifting parameters $u(p)$ and $v(p)$ to the original color c_p and set the scaling one $f(p)$ to 1 to preserve the original color in the preservation (background) regions. In addition, the latter term (i.e., $j \in \mathcal{B}$) of the case $l_p = \mathcal{F}$ sums up the probability that a given color c_p appears in the background Gaussian color model $G_{\mathcal{B}}^s$. This is designed to prevent segmentation errors in the background regions which produce unexpected change, especially for the edges and some fragmentary background regions between the foreground ones.

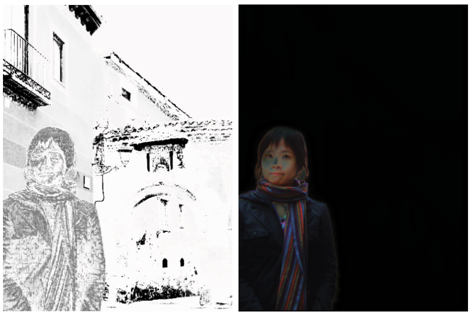
5.2. Gradient-guided color transfer function

Applying only the pixel-wise local color transfer functions for color transfer may result in some artifacts as shown



(a) (b)

Figure 4: *(a) The result of directly applying the pixel-wise local color transfer functions for transferring color. (b) The close-up view of the red rectangle in image (a).*



(a) (b)

Figure 5: *(a) The weight of the gradient-guided constraints. (b) The difference between Figure 2 (c) and (d).*

in Figure 4. Thus, before performing the color transfer, we need to use the gradient of the original source image I_s to improve the pixel-wise local color transfer functions $u(p)$, $f(p)$ and $v(p)$ and obtain the gradient-guided color transfer functions $\hat{u}(p)$, $\hat{f}(p)$, and $\hat{v}(p)$ based on the the following quadratic energy function proposed by Lischinski *et al.* [LFUS06]:

$$\hat{f} = \arg\min_{\hat{f}} \left\{ \sum_{p\in I_s} w(p)(\hat{f}(p) - f(p))^2 + \lambda \sum_{p\in I_s} h(\nabla \hat{f}(p), \nabla L) \right\}, \tag{7}$$

where the first term constraints the solution $\hat{f}(p)$ to be as close to the original $f(p)$ as possible and $w(p)$ is defined as:

$$w(p) = \max_{j\in x} C(c_p, j), \quad \forall l_p = x \in \{\mathcal{F}, \mathcal{B}\}.$$

The second term in Eq. (7) is the smoothness term which is defined as:

$$h(\nabla \hat{f}(p), \nabla L(p)) = \frac{|\hat{f}_x(p)|^2}{|L_x(p)|^\alpha + \varepsilon} + \frac{|\hat{f}_y(p)|^2}{|L_y(p)|^\alpha + \varepsilon},$$

where L is the log-luminance channel of the source image I_s, and the subscripts x and y denote spatial differentiation. This ensures consistency between the neighbor values in $\hat{f}(p)$, but allows for rapid changes across significant edges. Although Eq. (7) is used for enhanced the scaling parameter $f(p)$, the shifting parameters $u(p)$ and $v(p)$ are improved by applying the same process. Figure 5 (a) shows the weight of the gradient-guided constraints.

5.3. Color transfer

Finally, we conduct the color transfer via Eq. (3). Hence, each pixel is altered by different and suitable transfer parameters. Furthermore, the color transferring process does not influence the preservation (background) regions. Figure 5 (b) shows the difference between the source image and the final result. The background is preserved completely and no detail is lost in the photo enhancement process.

6. Results

Figure 2 (c) shows a common case where the defect photo is taken against the light source. We used another photo (the left upper photo of Figure 1 (a)) taken on the same day to enhance it. The source defect photo is successfully enhanced by our color transfer method and both the clothing and scarf in the source photo are successfully recovered as shown in Figure 2 (d). Figure 2 (a) and (b) show the strokes drawn on the source and target photos respectively.

Figure 1 (c) shows a situation where is difficult to take a satisfying photo by a specific exposure. The source photo (the left lower photo of Figure 1 (c)) is taken by the exposure which is suitable to capture the sea of clouds. For the overexposed sky, we chose another photo (the right lower photo of Figure 1 (c)) taken on an airplane at sunrise and transfer the color of the sky to enhance it. The result is shown in the upper photo of Figure 1 (c).

Our multiple local color transfer method can also be used as for global color transfer. Figure 6 (a) shows an overexposed photo and Figure 6 (b) shows another photo taken by a different person on the same trip, which served as target photo. Our result (Figure 6 (c)) is compared with Reinhard *et al.*'s [RAGS01] one (Figure 6 (d)).

Figure 7 shows the comparison of our result and that of Tai *et al.*'s local color transfer method [TJT05]. The source photo is taken with the red tree, so the surface of the river reflects a little bit of red light. Tai *et al.*'s result as shown in Figure 7 (c) successfully separates the trees and the river when transferring the green color, however, the river still includes a little bit of red. Our river as shown in Figure 7 (d) correctly reflects the green light on the river.

Sometimes it is hard to figure out how to achieve a combination of effects in a single step. In this case, it is better to have a progressive editing feature in the system, so that we can concentrate on one part first and then processing the others. Figure 8 demonstrates that our system is capable to

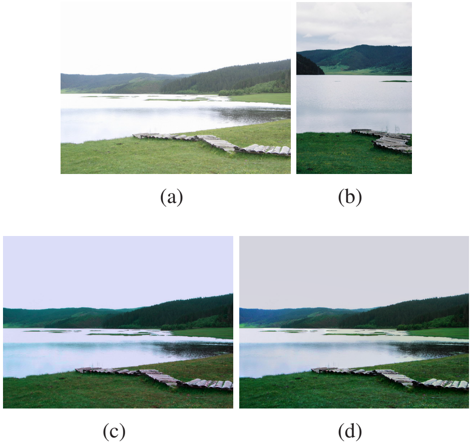


Figure 6: *The comparison of our result and that of Reinhard* et al.*'s global color transfer method [RAGS01]. (a) The source overexposed photo. (b) The target photo taken by another person. (c) Reinhard* et al.*'s [RAGS01] result. (d) Our result.*

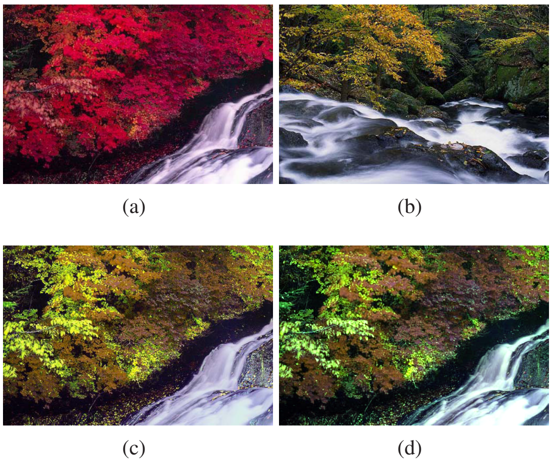


Figure 7: *The comparison of our result and that of Tai* et al.*'s local color transfer method [TJT05]. (a) The source photo. (b) The target photo. (c) Tai* et al.*'s [TJT05] result. (d) Our result.*

cope with this situation. The source photo as shown in Figure 8 (a) is taken under bright sunlight and high contrast. In this situation, it is hard to decide on the exposure to take a photo with that is satisfying for the whole region. In this example, the background is overexposed, but the face is still a little bit underexposed. The user can use our system to easily create an enhanced photo. To achieve this, we first chose a photo with a beautiful sea and coast in the same album, then perform background enhancement to obtain a first result as shown in Figure 8 (b). After having enhanced the background, the face is still not clear enough, so we chose another photo to enhance the skin color of the face and obtain the final result as shown in Figure 8 (c). To exploit the preservation stroke, the sea and coast are not influenced in the further editing. Finally, we have a photo with living expression and a beautiful background.

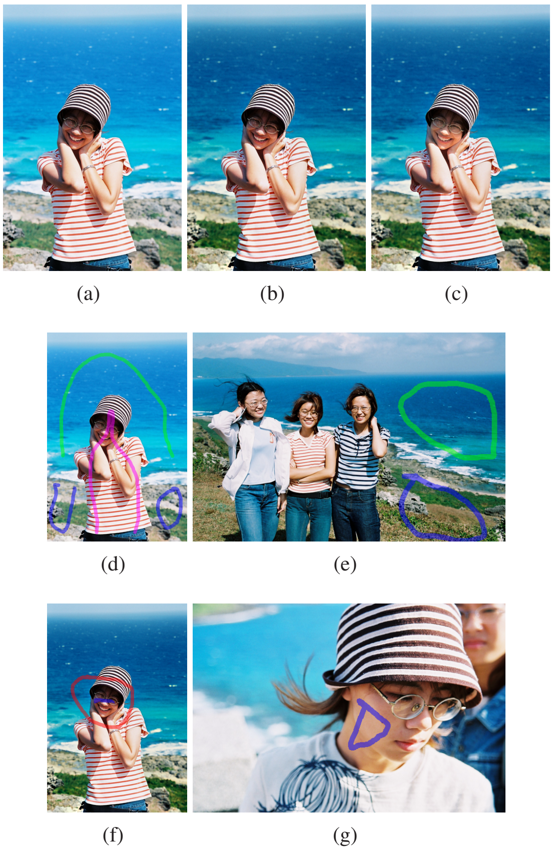


Figure 8: *The progressive editing example. (a) The source photo taken in a high contrast scene. (b) The result of enhancing the background. (c) The result of enhancing the face further. (d)∼(g) The strokes for editing, where (e) is used to enhance (d) and (g) is used to enhance (f). The red stroke in (d) and (f) successfully preserves the person in (a) and the background in (b).*

Our system is also able to seamlessly extract colors in different regions of different images. Figure 9 shows the case of using multiple target photos to enhance one source image. The source photo as shown in Figure 9 (a) is taken at dusk and the sunlight only lightens the mountain top. We acquired three target photos to enhance the sky, mountain, and forest parts, respectively, as shown in Figure 9 (d)∼(f).

Figure 10 (a) and (b) show two photos taken in the early evening. One photo (Figure 10 (a)) is taken with long exposure time for recording the track of the cars' lights, darker

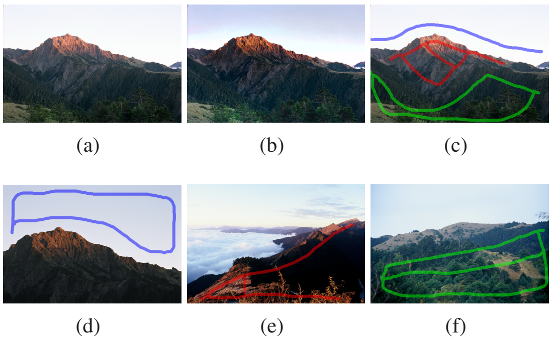


Figure 9: *An editing example of using multiple target photos. (a) The source photo. (b) Our result. (c) The strokes on the source photo (a). (d)∼(f) The target photos and their corresponding strokes, including the target photo for the (d) sky, (e) mountain, and (f) forest.*

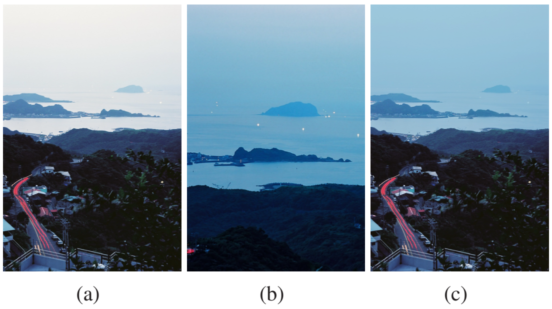


Figure 10: *(a) The source photo taken by long exposure for recording the track of the cars' lights. (b) The target photo record the mood in the evening. (c) Our result is created by combining the two subjects.*

roads, and houses, but the sea and sky become too bright because of overexposure. However, the user takes another photo (Figure 10 (b)) to capture the mood of the sea in the evening. Hence, we can use our system to combine these two subjects to create a better photograph.

Figure 2 (c) and Figure 11 show a result of enhancing the foreground which is mixed with the background, therefore it requires tedious segmentation in traditional approaches if partial editing is preferred. Using our stroke-based user interface, the underexposed foreground is successfully enhanced, but the small background regions surrounded by the foreground are not influenced. In addition, although the bookshelf in the source and target photos are not specified by the user, since its color is similar to the tablets, it can also be enhanced by the tablets in the target photo.

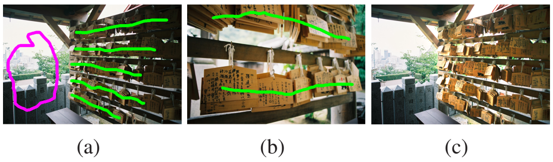


Figure 11: *(a) The strokes on the source photo. (b) The strokes on the target photo. (c) Our result.*

Table 1: *Processing time in each stage. (in seconds)*

	Scarf (Figure 2)	Coast (Figure 8)	Sunrise (Figure 1 (c))
Image dimension	480 × 640	428 × 640	640 × 424
Color space conversion	1.133	1.133	0.978
Graph cuts	2.984	3.094	3.454
Gradient-guided improvement	3.969	2.046	3.781
Upsampling	3.453	3.579	2.891
Other processes	1.148	0.992	0.830
Total	12.687	10.844	11.953

7. Implementation

In our method, the biggest problem with obtaining the gradient-guided color transfer function is performance. Although downsampling is quite useful for speeding up the process, upsampling is another problem. We use Kopf *et al.* [KCLU07]'s joint bilateral upsampling method to solve this problem. In their method, they combine region information of the full resolution source photo in the upsampling process, and the result is smooth and edges are preserved. Therefore, we can achieve an interactive photo enhancement system with a stroke-based user interface and still have good results.

In our system, the solution calculated on one eighth size in width and height is enough to get acceptable results, and the upsampling takes about 4 seconds for a 0.3 mega-pixel image in our implementation. Table 1 lists the processing times at each stage of some results in this paper. All results in this paper were produced by following parameter settings: $\alpha = 16$ in Eq. (1); $a = 5$ and $b = 5$ in Eq. (2); $\lambda = 0.2$, $\alpha = 1$ and $\varepsilon = 0.0001$ in Eq. (7).

8. Conclusion and Future Work

In this paper, we have proposed an example-based photo enhancement system with an interactive and intuitive stroke-based user interface. Furthermore, a multiple local color transfer method is presented and applied to transfer color from the target examples to the source defect photo through gradient-guided local (pixel-wise) color transfer functions. Our method can produce accurate results that match the users' expectations. What is more important, our system is very easy to learn and use; no detailed photographic or photo editing knowledge is required. Along with convenience and efficiency, it is also a powerful way to complete creative tasks. In future work, we would like to develop other editing features under the same scenario.

Although our system works well for almost photos, it still suffers from some limitations. First, a reference photo is necessary, although it should be easy to be acquired. If there is no reference photo at hand, it is hard to enhance the defect one by our system. Besides this, it might not be possible to recover the details of the defect photo if the details were not captured in strongly over-exposed or under-exposed photos.

Acknowledgments

We would like to thank Johanna Wolf for proofreading the manuscript and anonymous reviewers for their valuable comments. This work was partially supported by the National Science Council of Taiwan under NSC95-2622-E-002-018 and NSC96-2622-E-002-002, and also by the Excellent Research Projects of the National Taiwan University under NTU95R0062-AE00-02.

References

[ARNL05] AGRAWAL A., RASKAR R., NAYAR S. K., LI Y.: Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Transactions on Graphics 24*, 3 (2005), 828–835. (SIGGRAPH 2005 Conference Proceedings).

[BPD06] BAE S., PARIS S., DURAND F.: Two-scale tone management for photographic look. *ACM Transactions on Graphics 25*, 3 (2006), 637–645. (SIGGRAPH 2006 Conference Proceedings).

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1222–1239.

[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.

[ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics 23*, 3 (2004), 673–678. (SIGGRAPH 2004 Conference Proceedings).

[JSTS04] JIA J., SUN J., TANG C.-K., SHUM H.-Y.: Bayesian correction of image intensity with spatial consideration. In *Proceedings of 2004 European Conference on Computer Vision* (2004), vol. 3, pp. 342–354.

[KCLU07] KOPF J., COHEN M. F., LISCHINSKI D., UYTTENDAELE M.: Joint bilateral upsampling. *ACM Transactions on Graphics 26*, 3 (2007), 96. (SIGGRAPH 2007 Conference Proceedings).

[LFUS06] LISCHINSKI D., FARBMAN Z., UYTTENDAELE M., SZELISKI R.: Interactive local adjustment of tonal values. *ACM Transactions on Graphics 25*, 3 (2006), 646–653. (SIGGRAPH 2006 Conference Proceedings).

[LLW04] LEVIN A., LISCHINSKI D., WEISS Y.: Colorization using optimization. *ACM Transactions on Graphics 23*, 3 (2004), 689–694. (SIGGRAPH 2004 Conference Proceedings).

[LSS05] LI Y., SUN J., SHUM H.-Y.: Video object cut and paste. *ACM Transactions on Graphics 24*, 3 (2005), 595–600. (SIGGRAPH 2005 Conference Proceedings).

[PSA*04] PETSCHNIGG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics 23*, 3 (2004), 664–672. (SIGGRAPH 2004 Conference Proceedings).

[RAGS01] REINHARD E., ASHIKHIMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics and Applications 21*, 5 (2001), 34–41.

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics 23*, 3 (2004), 309–314. (SIGGRAPH 2004 Conference Proceedings).

[TJT05] TAI Y.-W., JIA J., TANG C.-K.: Local color transfer via probabilistic segmentation by expectation-maximization. In *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), vol. 1, pp. 747–754.

[WAM02] WELSH T., ASHIKHMIN M., MUELLER K.: Transferring color to greyscale images. *ACM Transactions on Graphics 21*, 3 (2002), 277–280. (SIGGRAPH 2002 Conference Proceedings).

[WYC*06] WANG C., YANG Q., CHEN M., TANG X., YE Z.: Progressive cut. In *ACM Multimedia 2006 Conference Proceedings* (2006), pp. 251–260.