A COLLECTION OF DIGITAL PHOTO EDITING METHODS

GUO DONG

NATIONAL UNIVERSITY OF SINGAPORE

2010

A COLLECTION OF DIGITAL PHOTO EDITING METHODS

GUO DONG
(B.Sc., Fudan University, 2005)

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

SINGAPORE, 2010

© 2010, Guo Dong

To my parents...

Acknowledgements

I am deeply grateful to Terence Sim for his thoughtful supervision in last 5 years. His patient guidance, encouragement was precious and helpful.

I really appreciate my colleagues in Computer Vision Lab (known as Media Research Lab 4 now), Zhang Xiaopeng, Miao Xiaoping, Zhuo Shaojie, Ye Ning, Li Hao, Cheng Yuan *etc*. for their help, advice, discussion, and/or collaboration. It was my beautiful memory working with them.

I would like to thank my beloved friends, Sun Jing, Wang Wenxu, Wang Xianjun, Chen Su, Qi Yingyi, *etc*. I really enjoyed the life in Singapore with them.

I also thank my friends who have appeared in my experiment photos or provided photos to support my works: Sun Jing, Lu Han, Zhuo Shaojie, Su Wenzhe *etc*.

Abstract

This thesis addresses three self-contained photo editing methods. First, we introduce a method to correct over-exposure in an existing photograph. Over-exposure is unavoidable when the dynamic range of a scene is much larger than that of a camera sensor. Our method attempts to solve this problem by recovering the lightness and color separately. Second, we introduce a method of creating face makeup upon a face image with another image as the style example. The face makeup process of our method is analogous to physical makeup. The color and skin details are modified accordingly while the face structure is preserved. One major advantage lies in that only one example image is required. This renders face makeup by example very convenient and practical. Some additional makeup effects, *e.g.* makeup by a portraiture, aging effects, beard transfer *etc.* are also easily achievable by our method with slightly different parameter settings. Last, we introduce a method of creating image composite by seamlessly blending a region of interest from an image onto another one while faithfully preserving the color of regions specified by user markup.

These three methods are provided as standalone solution. They could potentially be integrated as add-ons into existing photo editing software, or else serve as standalone software.

Contents

List of Figures

List of Tables

Chapter 1

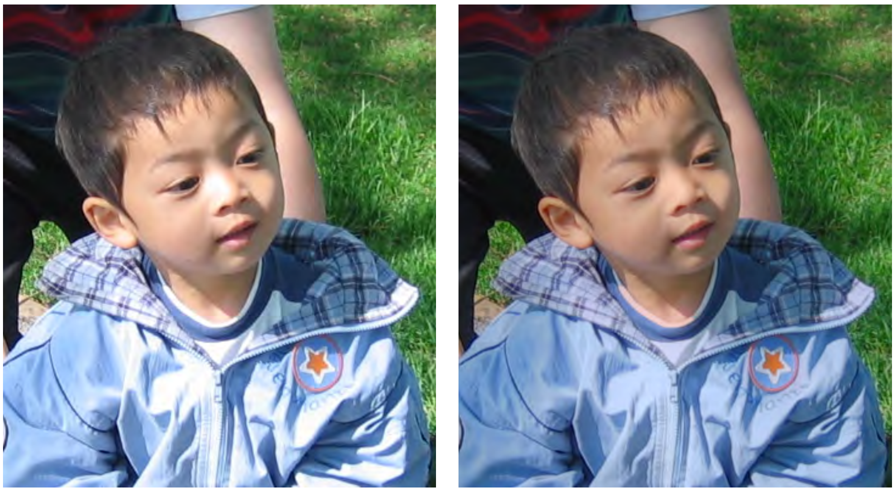
Introduction

1.1 Overview

Photo editing is as old as photography itself. Along with photography was invented over one hundred years ago, photo editing techniques were applied for various purposes, such as enhancing visual appearance.

Traditional photo editing techniques on film photography involved ink, paint, as well as airbrushes. These techniques were manually applied either on film in darkroom or on printed photos. They were applied mostly before digital cameras and computers came out. Nowadays, with the help of particular software on computer, photo editing has become much more accessible. The photo editing discussed in this thesis refers only to digital photo editing.

Digital photography is rapid and cheap, very popular among common users. A large number of photos are being taken at every second. Photo sharing with friends is another rising demand of people. As a result, there are dozens of online photo sharing websites *e.g.* ImageShack, Facebook, Photo Bucket, Flickr, each of



(a) Input Image (b) Result after Over-Exposure Correction

Figure 1.1: Illustration of Photo Editing: Over-Exposure Correction. Left: A photograph taken in an outdoor scene, with some portion over-exposed. Right: The result after over-exposure correction. The highlight of over-exposed regions is successfully reduced while the color is faithfully corrected.

which hosts over one billion photos.

Although digital camera has been developed for decades, current digital camera is still far from perfect. A lot of problems still exist. For example, a very common problem is over-exposure in photograph. The sensor of a camera has its limit of light range that it can capture. If the light falling on the sensor exceeds its limit, there would be a loss of high digital signals. This results in a loss of highlight details in bright regions in photograph. For example, in Figure 1.1[1], a photo of a child is taken under an outdoor lighting condition. The light was so strong that the nose and coat of the child appear over-exposed, *i.e.* too bright and color desaturated. This is a common reason that people would like to edit their photographs.

[1] All the images in this thesis are colored and high-resolution. In case of black/white hard-copy of this thesis, the reader may be interested in viewing electronic version on a monitor for a better understanding.

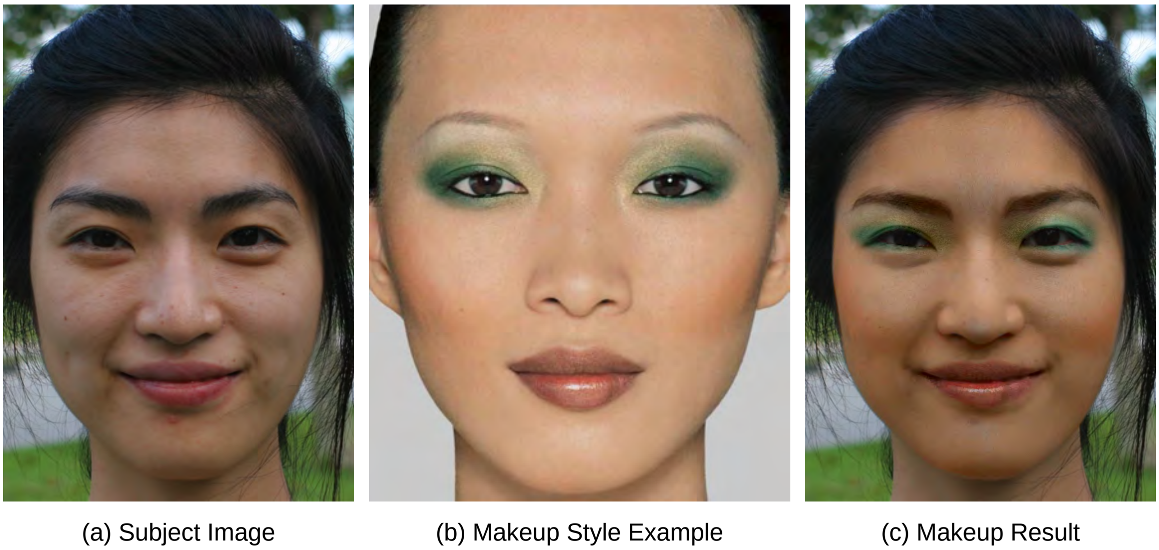
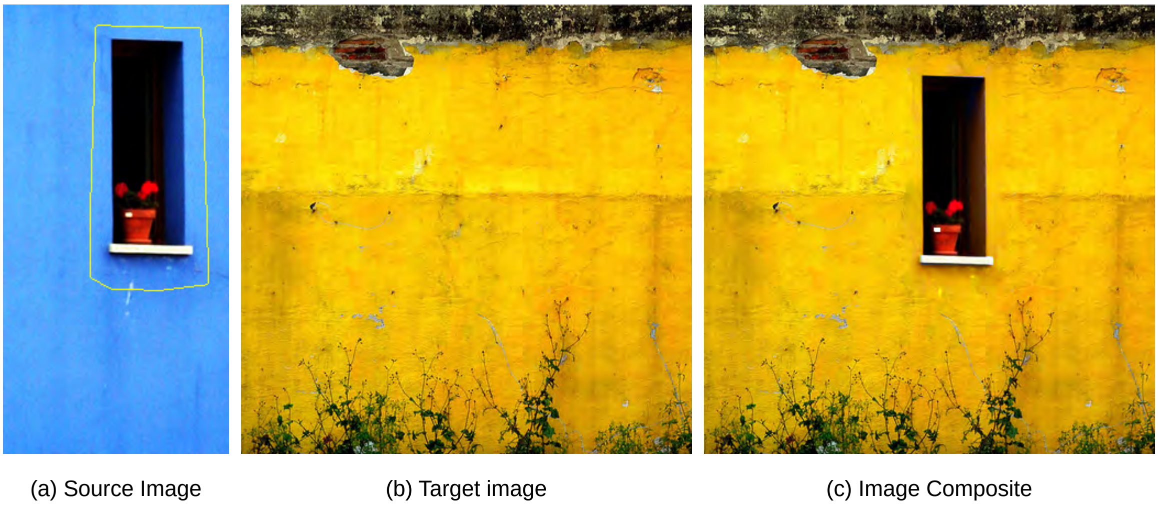

(a) Subject Image (b) Makeup Style Example (c) Makeup Result

Figure 1.2: Illustration of Photo Editing: Face Makeup by Example. Left: A subject image, taken by a common user. Middle: An example style image, taken from a professional makeup book. Right: The result of digital face makeup introduced in this thesis.

Currently, there exist a collection of photo editing software. Some well known ones are Adobe Photoshop, GIMP, Paint.net *etc*. However, the photo editing functions that they provide are mostly pixel oriented, like adjusting contrast, tone *etc*. It requires a lot of effort and expertise of the user to achieve an overall human-perception-based goal, like over-exposure correction. This thesis will discuss how to achieve such a goal fully automatically with mathematical models.

Another demand of photo editing is changing facial appearance. A large number of people, most of whom, females, would like to beautify their faces before sharing photos with friends. Like over-exposure correction, face makeup on photos is a difficult task for common users and also takes patience and time for photo editing experts. In this thesis, we will introduce a method to make the face makeup process easy and fast. The user only need to provide another face photo with makeup as the



(a) Source Image

(b) Target image

(c) Image Composite

Figure 1.3: Illustration of Photo Editing: Seamless Image Compositing. (a) The source image. The window frame is selected by the user within yellow line. (b) The target image. (c) The window frame is pasted on the yellow wall seamlessly with its color faithfully preserved.

makeup example. Our method can transfer the makeup effects from that example image to the subject. People can use this method to beautify their faces or even preview the makeup styles on their own faces. Like over-exposure correction, the makeup process is also achieved fully automatically with mathematical models. The user does not need any expertise or training before using our method. An example is shown in Figure 1.2. A photo of a girl (Figure 1.2(a)) is taken by an amateur without any special equipment. The girl did not wear makeup on her face. Another photo with professional makeup (Figure 1.2(b)) was taken from a professional makeup book [Nars 2004]. Our method successfully transferred the makeup style from the example to the subject, shown in Figure 1.2(c). Besides beauty makeup, additional makeup functions, such as makeup from portraiture, aging makeup, and beard transfer are also easily achievable by our method with different parameter settings. Details will be discussed in Chapter 3.

Besides these two, image compositing is another common need in photo editing. Image compositing is usually used to remove or replace an object. The process is very simple: a user selects a region containing an object from an image (the source image) and then choose a new position in another image (the target image) to paste the region there. To complete image composite, Adobe Photoshop provides a tool called clone stamp. With clone stamp, one can first pick a source position and then paste pixels onto a new location with alpha blending at the boundaries. However, merely blending boundary produces artifacts most of the time. To obtain a seamless result, Pérez *et al*. [2003] introduced the idea of Poisson Image Editing (PIE). With this idea, seamless compositing can be achieved by pasting the region of the source image onto the target image in gradient domain. Then a Poisson solver can be used to integrate from the modified gradient to obtain the composite result. Although the composite result is rather seamless, PIE, directly copying gradient, may result in a color shift of the pasted region. The color shift could be very large if the source and target images are of different colors. The shift is undesirable most of the time and not controllable by users. Thus, we propose a new method that provides seamless composite with the color of the pasted object preserved. For example, in Figure 1.3, the window frame is selected by the user as the region to be pasted. The yellow wall (Figure 1.3(b)) is provided as the target image. An image composite by our method is shown in Figure 1.3(c). The window frame is seamlessly pasted on the yellow wall and there is no color shift at all inside the window frame. The result of PIE, comparison, and additional results will be shown in Chapter 4.

Other than these three problems, there are various purposes for editing a photo. In this thesis, we will focus on aforementioned three needs of photo editing, namely, over-exposure correction, face makeup by example, and color-preserving image compositing. Over-exposure correction can largely correct and repair the over-exposed region of photo. With face makeup, the users can do face makeup on a photo with another photo as the example. Image compositing can seamlessly blend an object from a photo to another with its color well preserved. These methods are fully automatic and achieved by mathematical models. The methods are provided as a complete solution rather than a piece of tool as in most photo editing software. They could potentially be integrated as add-ons into existing photo editing software, or else serve as standalone software.

1.2 Thesis Contributions

In this thesis, three self-contained works of photo editing are introduced. We analyze each particular problem and formulate it into mathematical optimization problem. All these problems can be eventually solved by a sparse banded linear equation, which has many well studied and fast solver implementation. The experiment results have demonstrated the effectiveness of our methods.

Over-exposure correction. We introduce a method that is effective in correcting over-exposed regions in existing photographs. The method is fully automatic and only requires a single input photo. This makes the method applicable to existing photographs. The method is effective in correcting fully over-exposed regions, while previous methods could only handle partially over-exposed regions. In addition, the user has the flexibility to decide the amount of over-exposure correction. This work has been published in CVPR '10 [Guo et al. 2010].

Face makeup by example. Our method is effective in transferring face makeup from an example image to a subject image. This significantly reduces the effort and time for creating makeup effects using traditional photo editing tools. Moreover, only one single example image is required. This renders face makeup by example much more convenient and practical, compared to previous work which usually requires a pair of “before”-“after” makeup images as examples. Additional makeup effects, such as makeup by portraiture, aging effects, beard transfer, are also easily accessible by our method with slightly different parameter settings. This work has been published in CVPR ’09 [Guo and Sim 2009b].

Seamless image compositing. For image compositing, we present a method to produce a seamless image compositing. The color of the pasted region is well preserved, unlike Poisson image editing. Besides, users have the flexibility to choose different regions whose color being preserved. This work has been published in CAIP ’09 [Guo and Sim 2009a].

Chapter 2

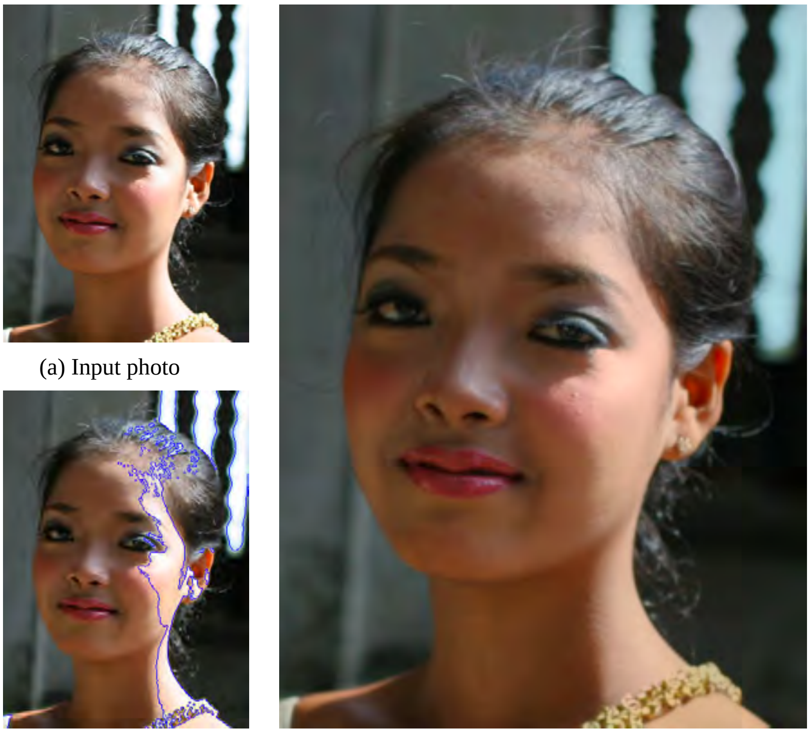
Over-Exposure Correction

2.1 Overview

Digital cameras use a sensor [1] to convert light into digital signal. Every sensor has its limit of light range that it can capture. As a result, if the light falling on the sensor exceeds its limit, there would be a loss of signal and the output signal would be capped at a particular maximum value. In a digital photograph, it appears as a loss of highlight details in the bright regions of the digital photo. This is called *over-exposure*.

Over-exposure happens very often in daily-life photography because the dynamic range of the scene is usually larger than that of camera's sensor. In photography, the term "dynamic range" refers to the ratio between the brightest and darkest measurable light intensities. The dynamic range of common digital cameras is very limited, usually 1000:1, which is much less than that of the real-world scenes. High contrast scenes, such as outdoor environment under direct sun light, may have a

[1] Two common used types of sensor are CCD (charge coupled device) and CMOS (complementary metal oxide semiconductor).



(a) Input photo

(b) Over-exposed region

(c) Result of our method

Figure 2.1: Over-exposure correction. (a) Input photograph, taken in an outdoor scene, with some portion over-exposed. (b) Over-exposed regions, marked in blue lines. (c) The result of our method, where the highlight of over-exposed regions is successfully reduced while the color is faithfully corrected.

very high dynamic range, from 10^5 to 10^9. In such high dynamic range scenes, it is impossible to make everything well-exposed; over-exposure is inevitable.

Usually, the photographer uses built-in auto-exposure function or external illumination meter to adjust the exposure such that the subject is well-exposed. However, over-exposure may be still inevitable on subject if the dynamic range is high. An example is shown in Figure 2.1 (another example was shown in Figure 1.1 in Chapter 1). The girl's face in Figure 2.1 has a large over-exposed region (shown in (b)). In practice, some photographers decrease exposure value to reduce over-exposure. But this cannot prevent over-exposure; decreasing exposure value will make the photo dim and suffer from sensor noise.

In contrast to over-exposure, under-exposure refers to loss of details of dark region due to lack of enough light. In this work, we only consider over-exposure correction as the under-exposed dark regions are usually much less important than the subject of the photo. The subject is usually over-exposed rather than under-exposed.

Some works on High Dynamic Range (HDR) image capturing, such as [Debevec and Malik 1997] [Mitsunaga and Nayar 1999], aim to fully capture the whole dynamic range. With tone mapping techniques, such as [Fattal et al. 2002] [Reinhard et al. 2002] [Chen et al. 2005], the HDR images are mapped to Low Dynamic Range (LDR) ones, thus avoiding over-exposure. However, HDR cameras are too expensive[2] , while other existing HDR capturing solutions usually require multiple shots with different exposure values. HDR imaging techniques with multiple shots are restrictive as they require the scene to be static. Furthermore, HDR capturing works

[2] Recently, Apple Inc. announced an HDR photography function in iPhone 4, which makes HDR imaging less expensive and popular in common users. However, the HDR imaging technique used in iPhone requires multiple images.

only for new photographs; it cannot correct over-exposure in existing photographs.

In this chapter, we present a method to correct over-exposed regions (as shown in Figure 2.1(b)) in a single existing photograph (Figure 2.1(a)). In our result (Figure 2.1(c)), the highlight of over-exposed regions is greatly reduced yet the contrast is still preserved. Meanwhile the color of these regions is faithfully corrected.

The intensity of over-exposed regions is clipped at the maximum value (*e.g.* 255 in images with 8-bit per channel), thus appearing uniformly white. Therefore, a natural way to recover the over-exposed regions is to first estimate the actual value, *e.g.* a work on estimating HDR from LDR [Rempel et al. 2007], and then compress the estimated HDR back to LDR image. However, in most cases, it is difficult to accurately estimate the actual value from a region if its information is completely lost. This is because the actual value might be slightly higher than the maximum value or boost up to a huge one (such as light from the sun). Instead of estimating the actual value and re-mapping to LDR, we present a method that slightly compresses the dynamic range of well-exposed regions while expanding the dynamic range of over-exposed regions. This directly produces an image with the over-exposure corrected.

2.2 Related Work

There is not much previous work directly addressing over-exposure correction. The closest are works by Zhang and Brainard [2004] and Masood *et al.* [2009]. In Zhang and Brainard's work, the ratios between different color channels are used to recover the over-exposed channels. The ratios are estimated based on that of pixels around over-exposed regions. However, the assumption of spatially invariant ratios in their work is inapplicable in real cases. Thus, Masood *et al*. utilize spatial-variant ratios in estimating pixel values in the over-exposed channels. Both the two works can only handle partial over-exposure *i.e*. one or two color channels are over-exposed. Regions of full over-exposure *i.e*. all three channels are all over-exposed are left untouched. However, in real photographs, full over-exposure exists most of time. We need an algorithm working with both partial over-exposed and full over-exposed regions.

Some previous works focused on hallucinating HDR from an LDR image, such as [Wang et al. 2007] and [Rempel et al. 2007]. Wang *et al*. used texture synthesis algorithm to fill the detail texture in over-exposed regions. Users have to specify the clue where the texture of the over-exposed region is similar. The lightness of the over-exposed region is estimated by a Gaussian ellipsoid based on the neighbors around the over-exposed region. In the fashion of texture synthesis it is always required that similar regions should be available in the same photograph or other possible photographs. Also users' hints for texture synthesis requires a lot labor work if there are too many over-exposed regions. In contrast, the work of Rempel *et al*. aims to enhance the visualization of an LDR image on an HDR display device. A smooth brightness enhancement function is applied on and around the over-exposed region to boost up the dynamic range of the original image. However, color correction was not considered in this work.

HDR imaging can be used to capture an HDR scene without over-exposure. With HDR compression, such as [Reinhard et al. 2002] [Fattal et al. 2002] [Chen et al. 2005], an HDR image can be compressed into an LDR image. This kind of tone-mapped image could be well-exposed anywhere, depending on the tone-mapping function. However, HDR cameras are priced too high. Other systems such as

Table 2.1: A summary of comparison of related works and our method.

	Fully Over-exposure	**Scene with motion**	**Color correction**	**Automatic**	**Correct existing photo**
Zhang and Brainard	×				
Massod *et al.*	×				
HDR imaging with multiple shots		×			×
Rempel *et al.*			×		
Wang *et al.*				×	
Zhang and Sim	×(if NIR is also over-exposed)				×
Our method	✓	✓	✓	✓	✓

[Debevec and Malik 1997] [Mitsunaga and Nayar 1999] can composite multiple LDR photographs of the same scene with different exposure values. Thus, they require both the camera and the scene be static and the illumination be unchanged.

Instead of HDR capturing, some other works tackle the over-exposure with additional information. [Zhang et al. 2008] proposed a method that can recover over-exposed regions by transferring details from a corresponding Near-Infrared (NIR) image. Thus their method may deal with scene with motion. However, it is still quite possible that both visible and near-infrared images are over-exposed simultaneously. Also special equipment (dual camera system) is needed.

Both HDR and NIR imaging techniques are designed for capturing new photographs. They cannot correct over-exposed region in existing photographs. In contrast, we focus on correcting over-exposure with only one existing photograph. A summary of comparison of related works and our method is shown in Table 2.1.

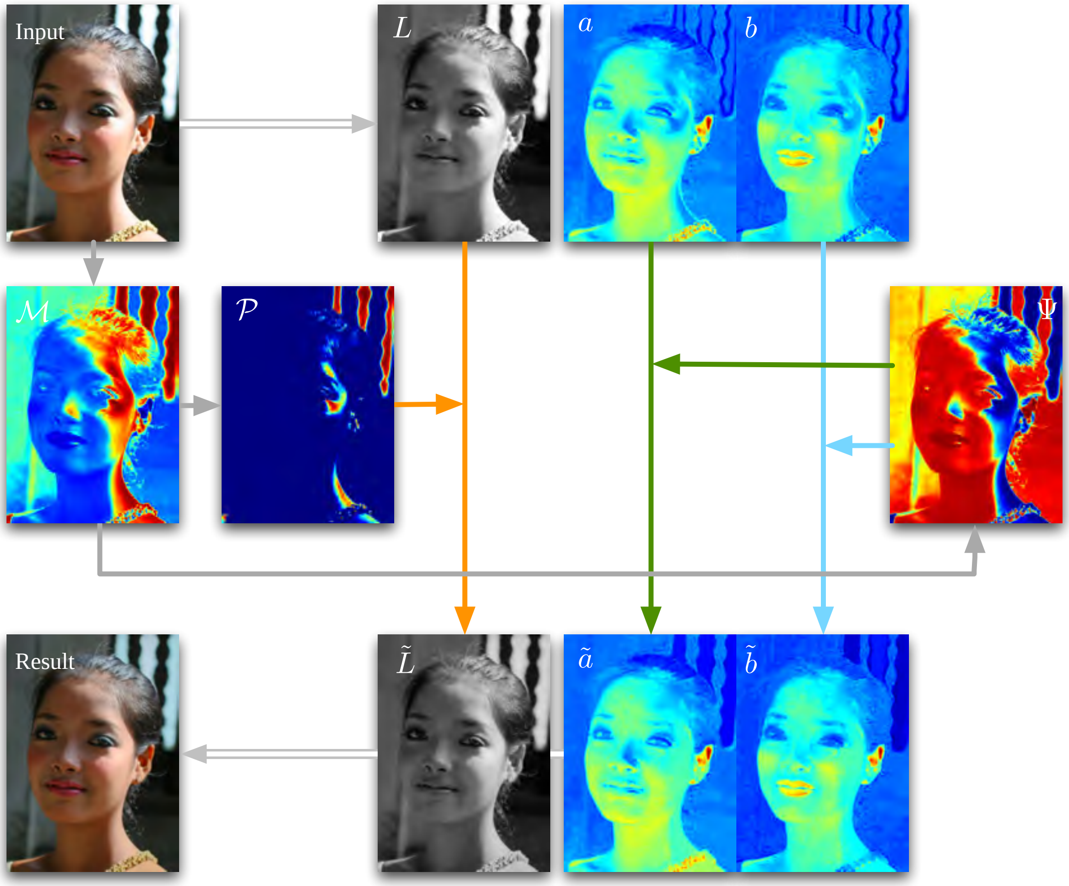


Figure 2.2: The workflow of our method. Two aspects are included, lightness recovery (orange line) and color correction (blue and green lines). Lightness recovery is through L^* channel using over-exposure likelihood $\mathcal{P}$. Color correction is through a^*, b^* channels using color confidence Ψ. Both $\mathcal{P}$ and Ψ are derived from an over-exposure map $\mathcal{M}$, which is generated from the input. Some images are rendered in false color for better visualization. Warmer color (red) denotes higher value.

Table 2.2: Notation used in this chapter.

Notation	Meaning
Ω	Over-exposed region.
$\neg\Omega$	Non-over-exposed region. All except Ω.
L	L^* channel of the input image.
C	Color channels (a^*, b^*) of the input image.
$\tilde{L}$	L^* channel of the output image.
$\tilde{C}$	Color channels (a^*, b^*) of the output image.
$\mathcal{M}$	Over-exposed map (how much a pixel is affected by over-exposure).
$\mathcal{P}$	Over-exposure likelihood (how likely a pixel in Ω is still over-exposed in the output image).
$\mathbf{z}(.)$	Attenuation function of gradient of the input image.
Ψ	Color confidence map (how confident a pixel color is).

2.3 Methodology

In an over-exposed region, the pixel values are clipped at the maximum value, such as 255. Thus, an over-exposed region becomes uniform at about that value in all or some channels[3]. Figure 2.1(a) shows an example. The girl's portrait was taken in an outdoor scene with strong sun-light from her left side. Although taken with auto exposure mode, her left face is still over-exposed, marked in blue lines in Figure 2.1(b). In the following part, we use Ω to denote the over-exposed region, and $\neg\Omega$ to denote the rest of image. A summary of notation used in this chapter is shown in Table 2.2.

There are two aspects in correcting over-exposure in Ω, lightness recovery and color correction. The actual lightness of over-exposed regions should be at least the maximum value. Thus, a natural way to correct over-exposure is to "hallucinate"

[3] Sometimes, due to the compression algorithm of JPEG or other format, this value might be slightly lower than the maximum value.

the lightness first. As introduced in [Wang et al. 2007], a Gaussian ellipsoid is used to fit the boundary of an over-exposed region so as to guess the lightness inside. However, in actual fact, it might be incorrect if strong light sources exist *e.g.* the sun in outdoor scene. The lightness level is rather difficult to estimate. Yet, for subsequent display purpose, the hallucinated lightness should be compressed back to LDR. This inspired us to design an algorithm to recover the over-exposed regions directly in a low dynamic range image. This, on one hand, avoids directly estimating the lightness, and on the other hand, directly makes good use of the original information captured by cameras.

During color correction, color in Ω is corrected via neighborhood propagation based on both neighborhood similarity and the confidence of the original color.

To separately deal with lightness and color, the input image is first converted to CIELAB colorspace, where the L^* channel represents lightness and a^*, b^* channels represent the color. In the rest of this chapter, we use L to represent L^* channel and

$$C = \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.1}$$

to represent a^* and b^* channels of the input image. $\tilde{L}$, $\tilde{C}$ are defined similarly to represent the L^* and a^*, b^* channels of the result image, respectively.

The workflow of our algorithm is shown in Figure 2.2. The input is a photograph with over-exposed regions. First, over-exposed regions are detected, denoted by an over-exposed map. The over-exposed likelihood and color confidence are generated from the map. Lightness recovery and color correction are performed using the two probability maps. The recovered lightness and corrected color are then combined forming the output image.

2.3.1 Over-exposure detection

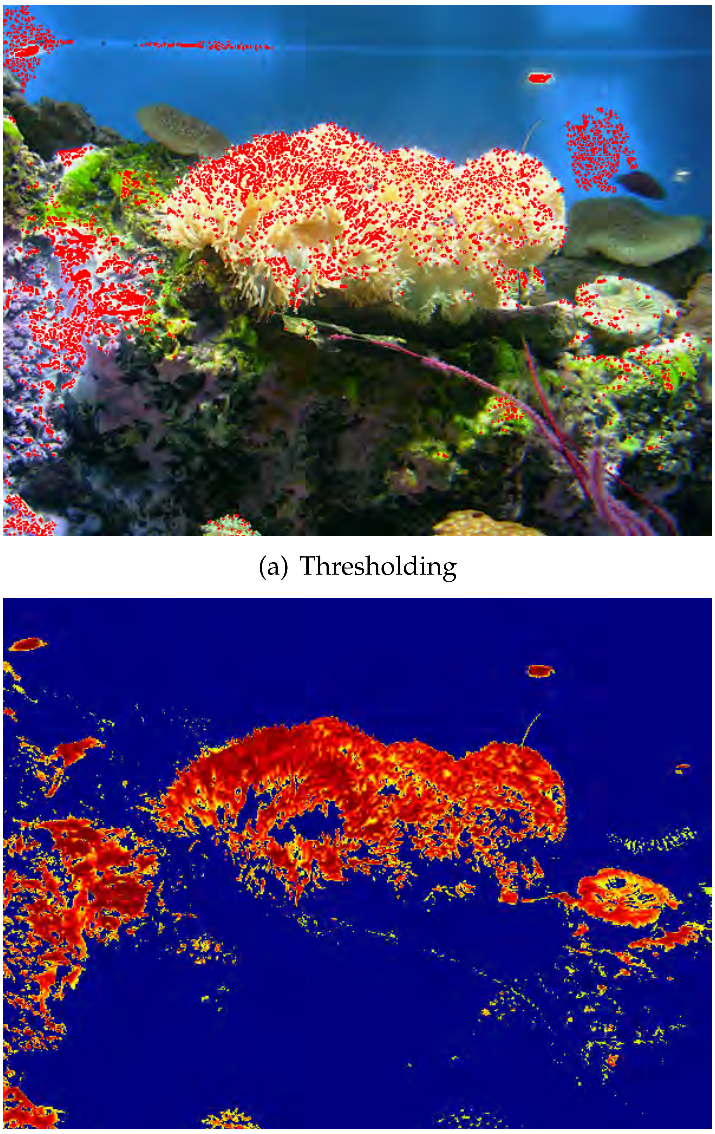
Previous works usually use a simple scheme to detect over-exposure: If the value of a pixel is equal or larger than a threshold, the pixel is considered over-exposed. Usually, the threshold is set to 254 to eliminate the effects of the error due to the compression algorithm. However, such a hard threshold does not handle well the gradual transition from over-exposure regions to their immediate neighbors. The color of these neighbors is desaturated ($||C||_2$ becomes smaller) and lightness increases (L becomes larger). Thus, we create an over-exposed map $\mathcal{M}$ denoting how much a pixel is affected by over-exposure. If L of a pixel is larger or its $||C||_2$ is smaller, the more the pixel is affected by over-exposure. Thus, we define M_i first

$$M_i = (L_i - L_T) + (C_T - ||C_i||_2) \quad , \tag{2.2}$$

L_T and C_T denote the boundary value of the over-exposure region. We consider pixel i is affected by over-exposure if $M_i > 0$. Furthermore, we define

$$\mathcal{M}_i = \frac{1}{2}\Big(\tanh\Big(\delta \cdot M_i\Big) + 1\Big), \tag{2.3}$$

where tanh denotes hyperbolic tangent. An illustration of the hyperbolic tangent function is shown in Figure 2.4. Then, $\mathcal{M}_i$ is inside $(0, 1)$ and $\mathcal{M}_i$ increases dynamically with small M_i and increases slowly toward 1 with large M_i. Thus, $\mathcal{M}_i$ is almost 1 in fully over-exposed region and larger than 0.5 in regions that are affected by over-exposure.



(a) Thresholding

(b) Over-exposed Map

Figure 2.3: Illustration of over-exposure map. (a) Over-exposure detection by simple thresholding (red dots). (b) Over-exposed map (only showing $\mathcal{M} > 0.5$). The input image is shown in Figure 2.8(a). (b) is rendered in false color. Warmer color (red) denotes higher value.

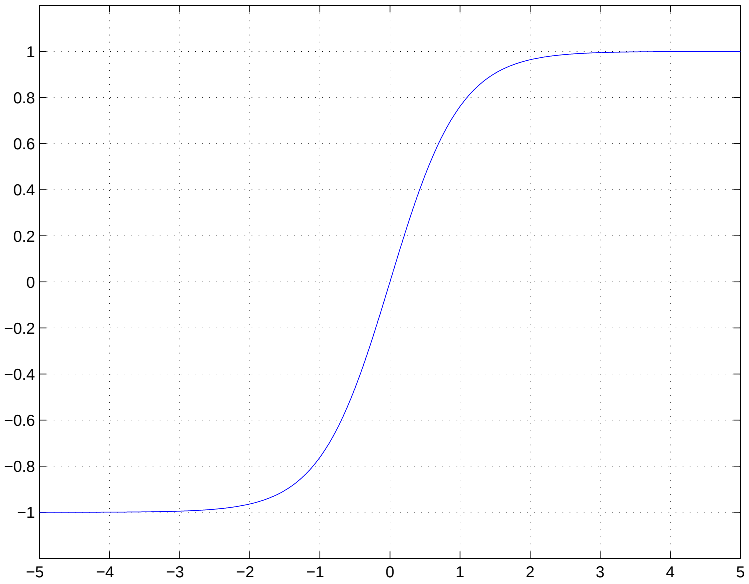


Figure 2.4: Illustration of the the hyperbolic tangent function over the domain $-5 \leq x \leq 5$.

A parameter δ is used to control the curve of speed that $\mathcal{M}_i$ grows with M_i. In experiments, we use $\delta = 1/60$, $L_T = 80$ and $C_T = 40$. We show an example of the area with $\mathcal{M} > 0.5$ in Figure 2.3(b). We consider these regions seriously affected by over-exposure and thus need correction. We use Ω to donate regions whose $\mathcal{M}$ is larger than 0.5 and $\neg\Omega$ for those whose $\mathcal{M}$ is less than 0.5. Ω used in our method covers much more area than the detection result by simple thresholding method (Figure 2.3(a)).

2.3.2 Lightness recovery

To make room for the recovered lightness of Ω, we use a tone mapping technique to compress the dynamic range of $\neg\Omega$. Once the dynamic range of $\neg\Omega$ is compressed, Ω could be expanded to fill the gap.

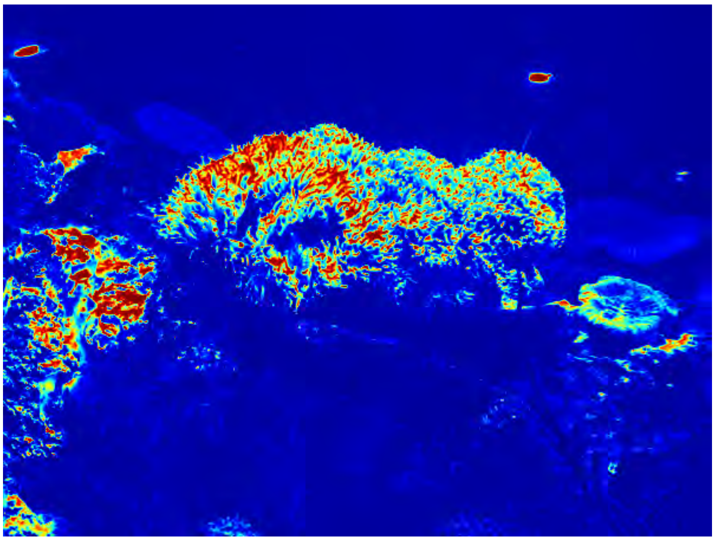


Figure 2.5: Illustration of over-exposure likelihood $\mathcal{P}$. Warmer color (red) denotes higher value.

Here we introduce *over-exposure likelihood* $\mathcal{P}$ to measure how likely the pixel in region Ω is still over-exposed in the output image. $\mathcal{P}$ is defined based on $\mathcal{M}$, *i.e.*

$$\mathcal{P}_i = \frac{1}{K} \cdot \frac{1}{1 - \mathcal{M}_i}, \tag{2.4}$$

where K is a normalization factor to make

$$\max_i \mathcal{P}_i = 1. \tag{2.5}$$

In a sense, $\mathcal{P}$ reflects relative value of the actual lightness in Ω. An example of $\mathcal{P}$ is shown in Figure 2.5. $\mathcal{P}$ is small in most part, and close to 1 only when $\mathcal{M}$ is close to the maximum of $\mathcal{M}$. In other words, only the regions whose $\mathcal{M}$ is close to the maximum of $\mathcal{M}$ have high likelihood to keep the high lightness in the output image.

For dynamic range compression in $\neg\Omega$, we adapt the method proposed by Fattal *et al.* [2002]. Specifically, the gradient of the image is attenuated non-linearly: larger gradient is compressed more than smaller ones. The attenuation factor in $\neg\Omega$ is a power of the magnitude of the gradient. Gradient in Ω is kept unchanged to make the recovered lightness smooth in most places and keep details if any. The attenuation function $\mathbf{z}(.)$ is defined as

$$\mathbf{z}(\nabla L_i) = \begin{cases} \frac{\alpha}{\|\nabla L_i\|} \left(\frac{\|\nabla L_i\|}{\alpha}\right)^{\beta} \cdot \nabla L_i & \text{if } i \in \neg\Omega \\ \nabla L_i & \text{otherwise} \end{cases}, \tag{2.6}$$

where α and β are two parameters to control the compression ratio of the image. α is to control the minimal gradient that is compressed. It is usually set to the 0.1 times the average gradient magnitude. So, β is left to control the compression ratio. This is a user adjustable parameter to control the overall effect of the our results. We use $\beta = 0.9$ in most of our experiments. The choice of β will be discussed in Section 2.4.

Now that the desired gradients are obtained, the first objective is to keep the gradients of result image as similar as possible with these gradients. This leads to the energy of

$$\mathcal{E}_1 = \sum_i \left\|\nabla \tilde{L}_i - \mathbf{z}(\nabla L_i)\right\|_2^2 \tag{2.7}$$

should be minimized.

On the other hand, we need to keep the lightness in Ω as close as possible to the original value, with different likelihood $\mathcal{P}$. Thus, we define the second energy

$$\mathcal{E}_2 = \frac{1}{|\Omega|} \sum_{i \in \Omega} \mathcal{P}_i |\tilde{L}_i - L_i|^2, \tag{2.8}$$

where $|\Omega|$ denotes the number of elements in Ω.

The lightness of $\neg\Omega$ tends to be lower due to the compression of its dynamic range, while the lightness in Ω tends to keep its original high value with different likelihood. As a result, the lightness in Ω is modified according to $\mathcal{P}$, which represents the relative lightness in Ω. To recover the lightness, an overall energy

$$\mathcal{E}_L = \mathcal{E}_1 + \lambda\mathcal{E}_2, \tag{2.9}$$

is to be minimized with a hard constraint that

$$\tilde{L} = L \quad \text{if} \quad L < \min(L) + r(\max(L) - \min(L)) \tag{2.10}$$

where $r = 0.1$. λ is to balance gradient energy $\mathcal{E}_1$ and value energy $\mathcal{E}_2$. Smaller λ means the lightness of Ω is more affected by dynamic range compression in $\neg\Omega$. We use $\lambda = 5$ in our experiments which produce good results. The hard constraint (2.10) means the low-lightness regions are kept unchanged.

According to the Variational Principle, $\tilde{L}$ that minimizes $\mathcal{E}_L$ must satisfy the corresponding Euler-Lagrange equation, after simplifying,

$$\frac{\lambda\mathcal{P}}{|\Omega|} \cdot \tilde{L} - \Delta\tilde{L} = \frac{\lambda\mathcal{P}}{|\Omega|} \cdot L - \mathrm{div}\mathbf{z}. \tag{2.11}$$

The reader may refer to a detailed derivation provided in Appendix B.1. Since in discrete domain both Laplacian operator Δ and div are linear operators, (2.11) is a linear system with unknowns being $\tilde{L}$ in Ω. In this linear system, there is one equation for each pixel. For each equation, there are only five unknowns, which are that pixel and its four neighbors. Thus, this system is a sparse banded linear

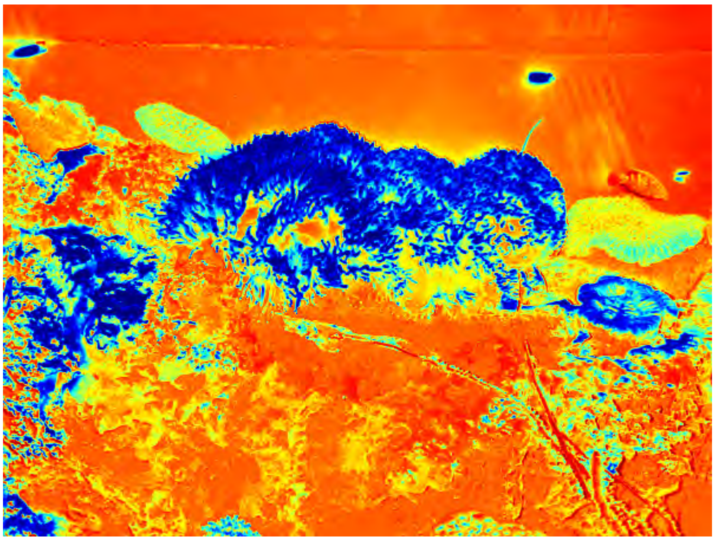


Figure 2.6: Illustration of color confidence Ψ. Warmer color (red) denotes higher value.

system, which could be solved efficiently.

An example of recovered lightness is shown in Figure 2.2, labeled with $\tilde{L}$.

2.3.3 Color correction

The color in or around Ω is more or less affected by over-exposure. We can use the over-exposed map $\mathcal{M}$ to represent how confident a pixel color is. Ψ is defined as

$$\Psi_i = 1 - \mathcal{M}_i. \tag{2.12}$$

An example of Ψ is shown in Figure 2.6.

We attempt to estimate less confident color from more confident one, propagating pixel from pixel via neighborhood similarity. A similar work by Levin *et al.* [2004] aims to colorizing a gray-scale image. The color of each pixel is propagated from the user strokes (indicating color) via neighbor pixels. The similarity in color is based on the gray value similarity. For color correction in over-exposure correction problem, the similarity is based on the lightness difference as well as the original color information that is confident.

The color of each pixel is similar to its neighbors with a similarity weight. Also, it is similar to its original color with a confidence Ψ. Thus, the color is corrected by minimizing

$$\mathcal{E}_c = \sum_i \left[(1 - \Psi_i) \left\| \tilde{C}_i - \sum_{j \in \mathcal{N}_i} w_{i,j} \tilde{C}_j \right\|_2^2 + \Psi_i \left\| \tilde{C}_i - C_i \right\|_2^2 \right], \tag{2.13}$$

where $\mathcal{N}_i$ denotes the neighborhood of a pixel i.

If the color confidence Ψ_i of a pixel $i \in \Omega$ is very low, the second term of $\mathcal{E}_C$ is less important, *i.e.* the original color value of this pixel does not have much effect on the result. Thus the color of this pixel would depend on information propagated from its neighboring pixels. In contrast, if a pixel is less affected by over-exposure, its color confidence Ψ_i increases; both two terms would contribute to the color in result $\tilde{C}_i$. For a pixel in $\neg\Omega$, the original color is dominating; its color tends to be unchanged in the result.

Our strategy for setting the weights w_{ij} is inspired by bilateral filter [Tomasi and Manduchi 1998]. The weight is product of several Gaussian functions of different distance measurements. Specifically,

$$w_{ij} = G(i - j)G(D_{\tilde{L}}(i, j))G(D_a(i, j))G(D_b(i, j)), \tag{2.14}$$

where $D_{\tilde{L}}(\cdot, \cdot)$, $D_a(\cdot, \cdot)$ and $D_b(\cdot, \cdot)$ denote the distance of corrected L^* *i.e.* $\tilde{L}$, and

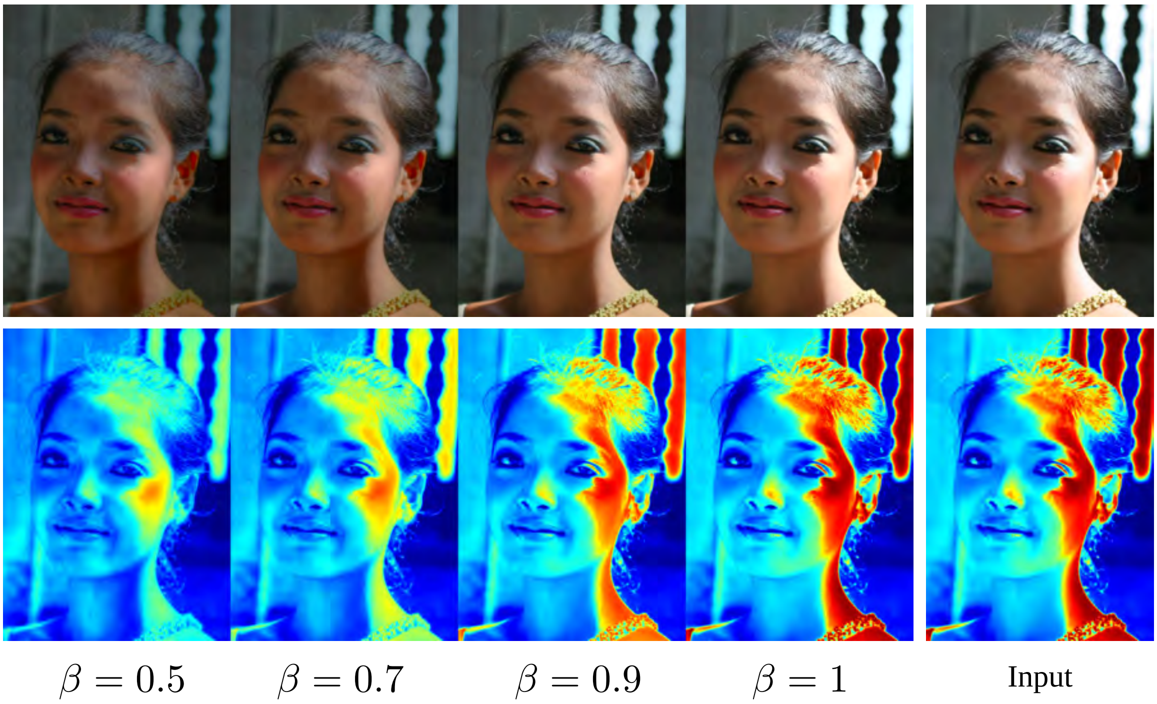


Figure 2.7: Results comparison with different β. Top row, recovered and input image. Bottom row, recovered and original L channel.

original a^* and b^* channels, respectively. The first Gaussian measures the spatial distance, while the second one measures the lightness difference. In other words, pixels that are nearer tend to have more similar color; pixels whose lightness is similar tend to have more similar color. The third and forth Gaussian functions measure the influence of the original color difference. In our implementation, If the original color is not confident enough ($\Psi < 0.6$, we omit these two Gaussian functions.

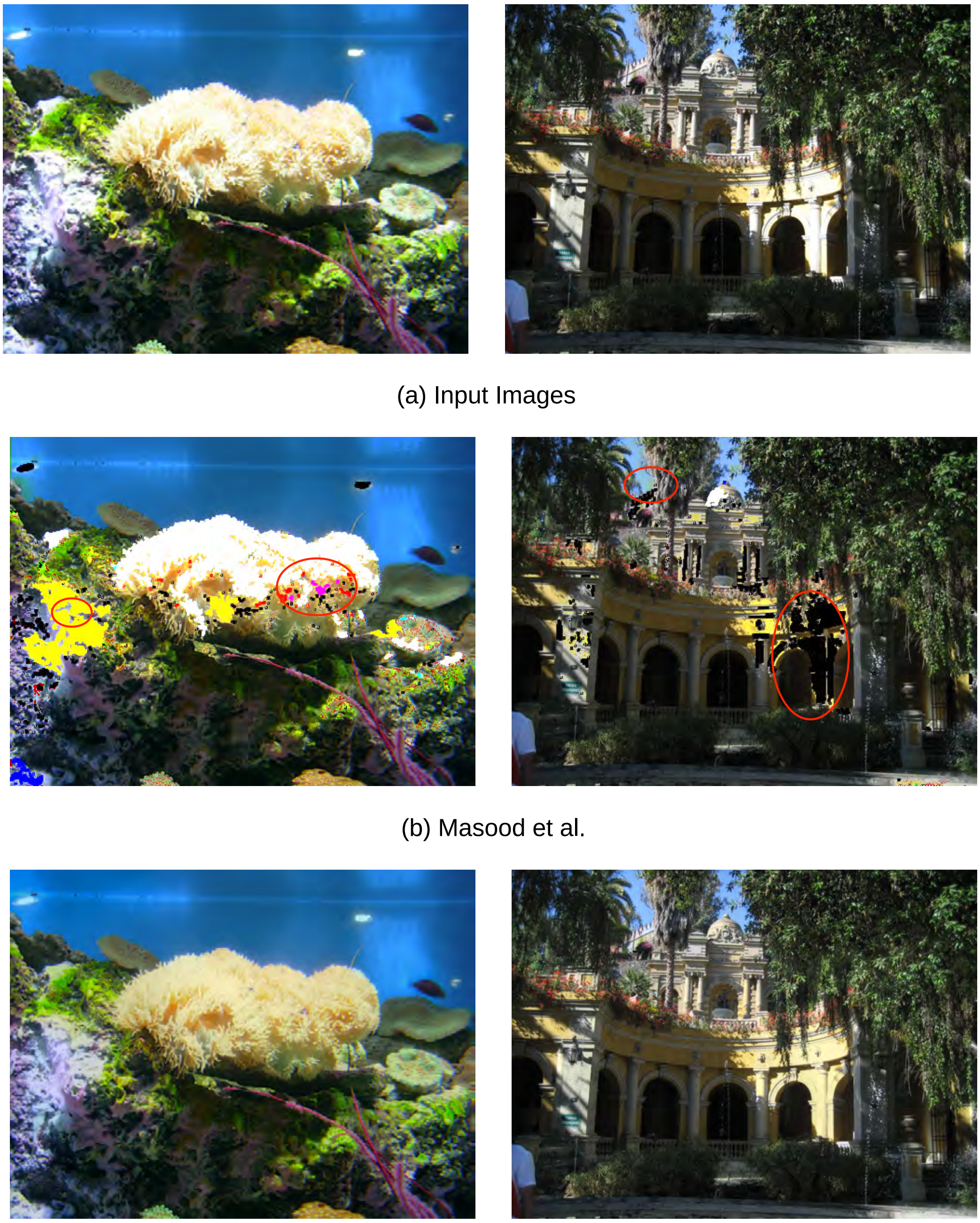
2.4 Experiment and Results

With different amount of well-exposed region being compressed, different levels of over-exposed region is corrected. In Figure 2.7, we show a series of results based on different β values. Smaller β results in more compression on $\neg\Omega$ and thus makes more space for recovering Ω. As a result, $\neg\Omega$ in result appears darker. When $\beta = 1$, the L channel stays untouched, while color channels are still corrected. The result is slightly better than the input. As β decreases, the exposure reduces yet still keeping the relative contrast. A too low β may cause the $\neg\Omega$ too dim, which is also undesirable. Usually, β ranging from 0.8 to 0.9 yields a good result. $\beta = 0.9$ is the value we used to obtain most results.

A comparison of our results with those of Masood *et al.* are shown in Figure 2.8. In the coral example (left), the body of coral has large over-exposed areas. The color becomes pale. In the castle example (right), there are some over-exposed regions on the right side wall. As some regions are fully over-exposed, Masood *et al.*'s method failed to recover these regions[4]. In contrast, our method successfully reduced the strong light and corrected the color of the over-exposed regions. The coral and the wall of the castle look natural and well-exposed in our results.

More results are shown in Figure 2.9 and Figure 2.10. In the flower example, many petals are over-exposed, while the bee on the flower is well-exposed. In our result, the strong reflection on petals is suppressed and the color is perfectly corrected. The bee still appears well-exposed. In the yellow leave and plant example, some parts of the leaves appear over-exposed. They becomes natural in the results. In the girl example, her face under strong sunlight become too bright.

[4] The result images by Masood *et al.* were generated by the code provided on the authors' website.



(a) Input Images

(b) Masood et al.

(c) Our Results

Figure 2.8: Comparison of our results with those of [Masood et al. 2009]. Artifacts in (b) are indicated in red circles, due to their limitation in handling regions with all three color channels over-exposed. In our result (c), the over-exposed regions are corrected successfully.

In our results, the lightness is reduced and color of skin is faithfully corrected. The arm of the Buddhist statue reflects strong sunlight, resulting in over-exposure in the photo. We successfully corrected the over-exposure while still keeping the shinning effects on the arm.

The running time is depending on the size of the over-exposed regions. A typical running time is about 1 second for one million pixel photo with about a quarter over-exposed region. The code was written in matlab and tested on a Core 2 Duo 2.33GHz computer.

2.5 Summary and Discussion

In this chapter, we have presented a method of correcting over-exposure on an existing photograph. Instead of recovering the actual lightness of the over-exposed regions and then compressed back into the image range, we directly estimate the value in the output image. The compression of gradients in well-exposed regions makes room for the over-exposed regions to expand the dynamic range. An over-exposure likelihood is employed to derive the lightness of over-exposed regions in the result image. Color correction is based on the color from well-exposed regions, the similarity of pixel neighborhood, and the confidence of the original color. Good results have demonstrated the effectiveness of our method.

Limitation: For severely over-exposed photographs, the boundary may become unclear between two adjacent objects. Our color correction method may propagate the color across objects, which is undesirable. Figure 2.11 shows such an example. Due to severe over-exposure, the old man's face and the window in background are connected. As a result, the window is colored by red from the old man's face.

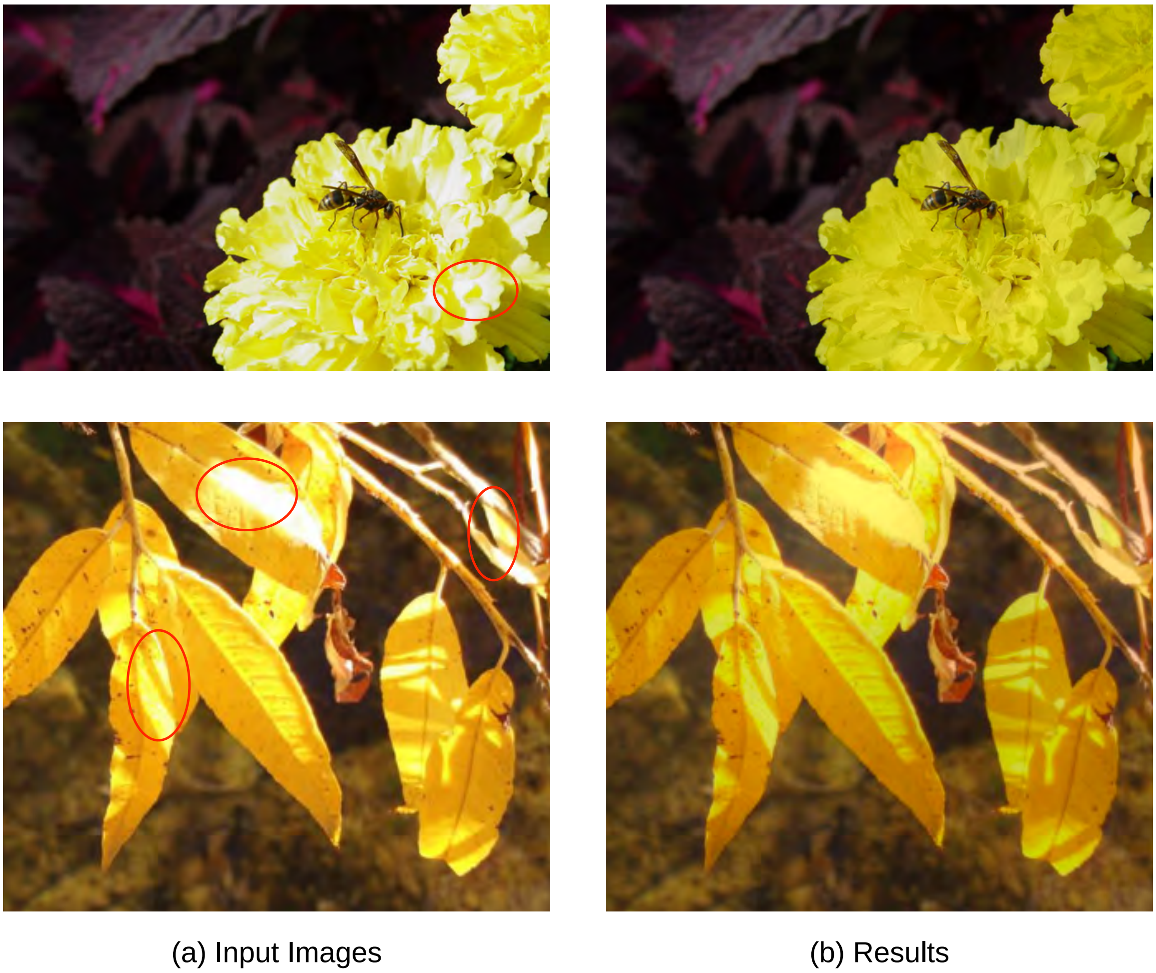


Figure 2.9: Results of correcting over-exposure. (a) Input images. (b) Results. Top row: coral example. Bottom row: yellow leaves. Over-exposed regions are indicated by red circles.

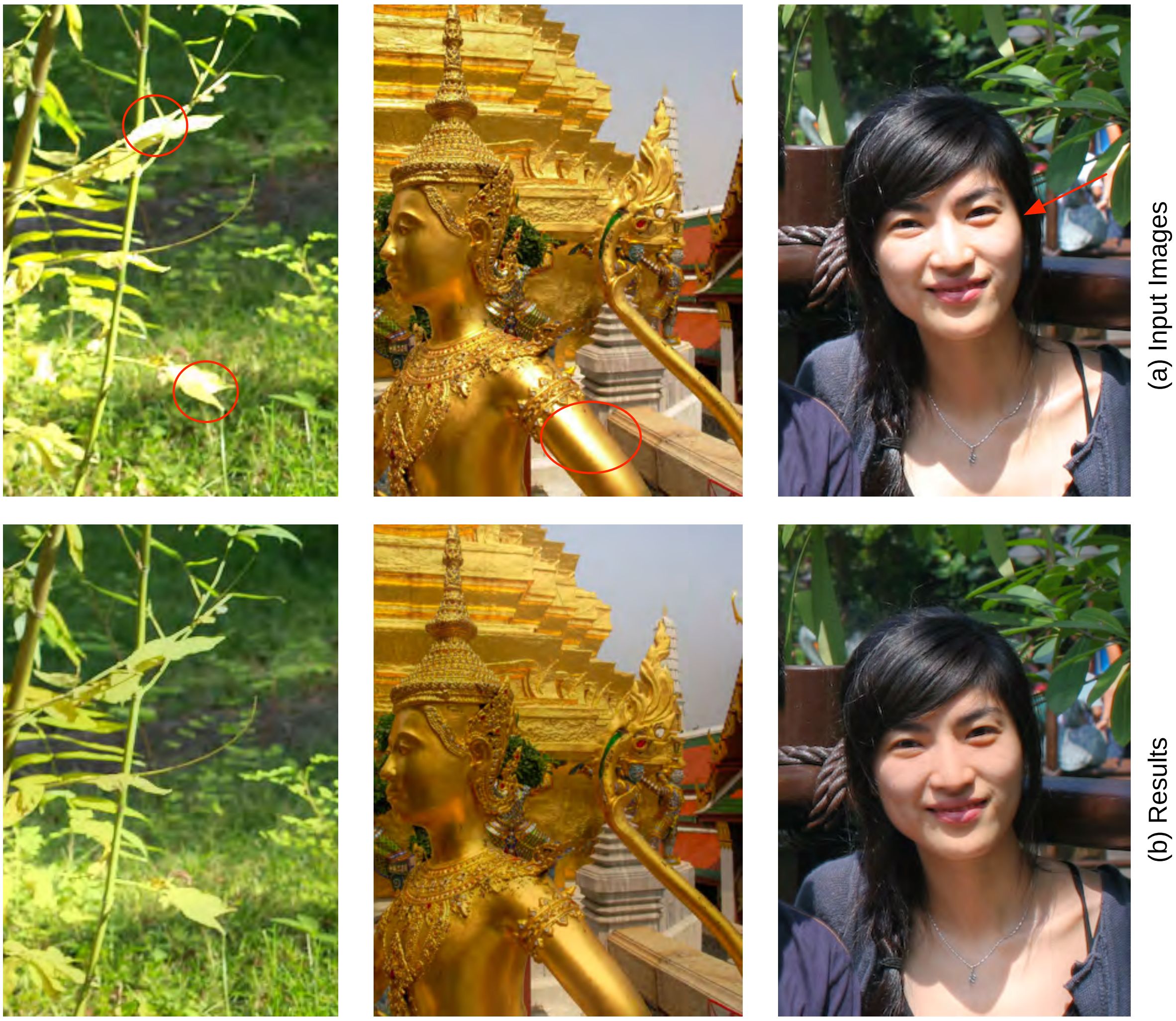
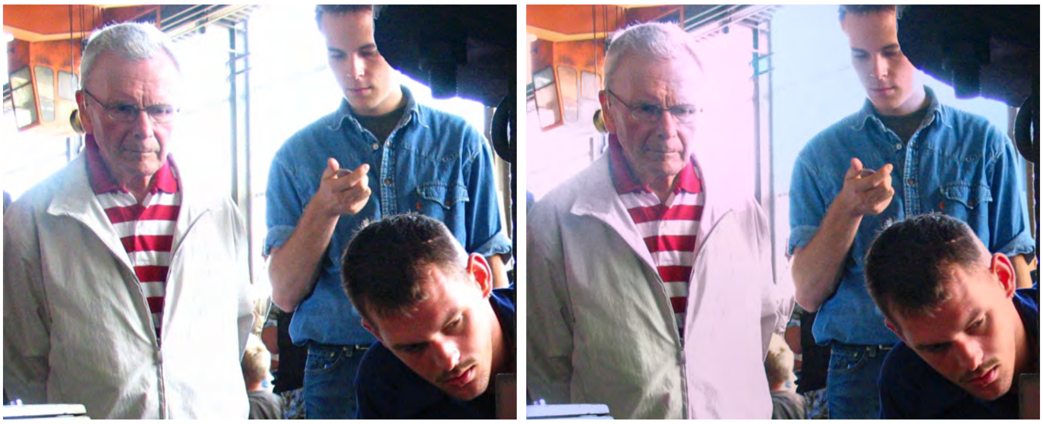


Figure 2.10: Results of correcting over-exposure. (a) Input images. (b) Results. Left: plant example. Middle: Buddhist statue. Right: girl. Over-exposed regions are indicated by red circles.



(a) Input

(b) Result

Figure 2.11: Illustration of limitation of our method. The boundary between the face and the background is missing due to serious over-exposure of the photograph. Our color correction method may propagate the color from the face into the background.

In contrast, although in the same photo, the face at right bottom is successfully corrected because the boundary of the face is very clear.

Chapter 3

Digital Face Makeup

3.1 Overview

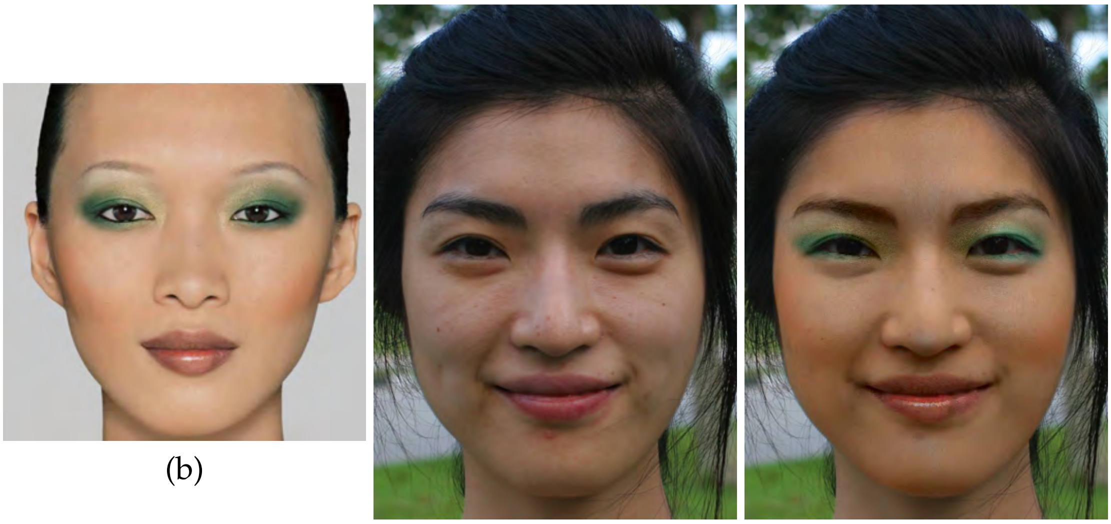
Changing, especially enhancing, facial appearance in photo is a demand of a lot of people. In this chapter we introduce a method to make over a face with another image as the makeup example. As shown in Figure 3.1, with a prototype of an image (Figure 3.1(b)) as the style example, our method could successfully make over a face image (Figure 3.1(a)).

Face makeup is a technique to change one's appearance with special cosmetics such as foundation, powder, cream *etc*. In most cases, especially for females, makeup is used to enhance one's appearance. With physical face makeup, the foundation and loose powder are usually used to change the texture of face's skin. Foundation is mainly used to conceal flaws and cover the original skin texture, while the loose powder is for introducing new, usually pleasant, texture to skin. Afterwards, applications of other color makeup, such as rouge, eye liner and shadow, etc., follow on top of the powder.

Inspired by the process of physical makeup, our method applies makeup in a similar way digitally. First, we decompose the subject and example images into three layers separately: face structure layer, skin detail layer, and color layer. Ideally, the face structure layer contains only the structure of every face component, such as the eyes, nose, mouth, *etc*. The skin detail layer contains the skin texture, including flaws, moles, as well as any wrinkles. The color layer represents color alone. After the three layers are decomposed, skin detail layer is transferred from makeup example to the subject image by analogy with the effects of foundation and loose powder in physical face makeup; The color layer is transferred from makeup example to the subject by analogy with the effects of color makeup physically. The face structure layer of the subject image is preserved.

Digital face makeup is useful in a lot of applications. Consider this scenario: when a customer enters a beauty salon, she selects an example image from a catalog and tells the makeup artist to apply the same makeup on her. Before actual task, it would be extremely helpful if she can preview the makeup effects on her own face. However, this is difficult. Traditionally, people have two choices for trying out makeup. One is to physically apply the makeup, which is time-consuming and requires the patience of the participants. Alternatively, one may try on makeup digitally by way of digital photography and with the help of photo editing software, such as Adobe Photoshop$^{\text{TM}}$. But using such photo editing software is tedious and relies heavily on the users' expertise and effort.

Besides makeup preview, digital face makeup is also extremely useful for photo retouching. One could choose her own well-taken photo (with slight makeup probably) previously as the makeup example to "make over" another photo of hers. We show an example later in Figure. 3.11, a girl use a photo taken in a



(b)

(a) (c)

Figure 3.1: Face makeup by example. (a) A subject image, taken by a common user. (b) An example style image, taken from a professional makeup book [Nars 2004]. (c) The result of our method, where foundation effect, eye shadow, and lip highlight in (b) are successfully transferred to (a).

professional studio as the makeup example to retouch her own faces in other daily-life photos.

In addition to beauty makeup, face makeup has special purposes in some cases, such as making actors appear either older or younger in dramas or movies, or as a component in Halloween makeup or cosplay[1]. We will introduce two special effects that could be easily achieved by our methods in Section. 3.4.4 and 3.4.5, aging effects, adding beard.

3.2 Related Work

There is not much previous work addressing digital face makeup. The most closely related work is that of Tong *et al.* [2007]. In their work, the way makeup changes

[1] Costume play, people wears costumes acting as a character in a cartoon or game

the appearance is learned from a pair of example images of the same face "before" and "after" makeup. The quotient of the "after" divided by the "before" is used to represent the change. Then the quotient is multiplied by another image in order to achieve the makeup result. In contrast, our method requires only one "after" example. This is more convenient and practical, as providing the "before" image is rather difficult in most cases. In addition, it is quite common in the makeup to change the texture of face skin (conceal the original and introduce new one). Because the original texture varies from person to person, the change from "before" to "after" is different between different faces. Thus, it is inappropriate to apply the change across two faces. In contrast, our method directly transfer the skin texture of the example to the subject image, concealing the original texture. Note that our method can also keep the texture of the subject image if needed.

Another method [Ojima et al. 1999] also uses a pair of "before"-"after" makeup examples. However, only the foundation effect is addressed in their work. In contrast, Tsumura *et al.* [2003] employed a physical model to extract hemoglobin and melanin components. Changes in facial appearance are simulated by adjusting the amount of hemoglobin and melanin. The effects they demonstrated include tanning, reddening due to alcohol consumption, aging and cosmetic effects. However, the cosmetic effects are quite limited, and are much simpler than real makeup. Besides, an online commercial software, Taaz [Taaz.com], provides users with virtual makeup on face photos by simulating the effects of specified cosmetics.

Some existing works focused on the beautification of face photos. For example, Brand and Pletscher [2008] proposed an automatic face photo retouching method aiming to detect and remove flaws, moles, and acne from faces. Another interesting work [Leyvand et al. 2008] introduced a technique of modifying face structure to enhance the attractiveness. However, this may also change the identity of the subject, as face structure is usually considered to be a key representation of identity. Conversely, we achieve the goal of beautification by modifying only skin detail and the color, while faithfully preserving the face structure.

Besides makeup, our work can also be applied to aging makeup or beard transfer. [Liu et al. 2001] shows a similar effect of wrinkles transfer. Some other existing works have focused on sumlating face aging process, *e.g*. [Lanitis et al. 2002], [Suo et al. 2007]. Aging simulation aims to predict the appearance of a face in a different age, which may involve face structure change. In constrast, our work is creating makeup effects on a face which could be achieved by physical makeup.

The idea of image processing by example can be found in image analogies [Hertzmann et al. 2001]. Image analogies provide a general framework of rendering an image in different styles. This method learns how pixel values change from a pair of "before"-"after" images as example. This idea was used in Tong *et al*.'s work [Tong et al. 2007]. As mentioned earlier, the difference is that our method learns the effects after alteration, while their method learns the way of altering image pixels.

Our method can also be considered as texture and color transfer. Some previous work on texture transfer is also related. Shan *et al*. [2001] proposed a method to transfer the texture across two images. Their method of transferring fine texture is similar to ours in spirit. They used the quotient of the original image and a Gaussian smoothed version of the image to represent the texture. The quotient is multiplied to the smoothed version of another image. This is similar to our layer decomposition. However, there are many differences. First, the Gaussian blur they used may produce halo effect at the strong edges. In contrast, we use an edge-preserving smooth [Farbman et al. 2008] to separate the layers, which

Table 3.1: Notation used in this chapter.

Notation	Meaning
$\mathcal{I}$	Subject image.
$\mathcal{E}$	Example image (after warping).
$\mathcal{R}$	Result image.
$\{.\}_s$	$\{.\}$'s face structure layer.
$\{.\}_d$	$\{.\}$'s skin detail layer.
$\{.\}_c$	$\{.\}$'s color layer.
$\delta_{\mathcal{I}}$	Weight controlling the contribution of $\mathcal{I}_d$ in $\mathcal{R}_d$.
$\delta_{\mathcal{E}}$	Weight controlling the contribution of $\mathcal{E}_d$ in $\mathcal{R}_d$.
γ	Weight controlling the amount of blending $\mathcal{E}_c$ in $\mathcal{R}_c$.

successfully suppresses the halo effects. Moreover, they only focus on texture transfer; but it is also important to keep color consistent. We separately transfer the skin detail in lightness channel and color in color channel. In addition, they focused on transferring texture, while our goal is to transfer makeup effect, which is more complex.

3.3 Methodology

The inputs are, a subject image $\mathcal{I}$, a face image to be applied with makeup, and an example image $\mathcal{E}$, providing makeup example. The output is result image $\mathcal{R}$, in which the face structure of $\mathcal{I}$ is retained while the makeup style from $\mathcal{E}$ is applied. The notation used in this chapter is listed in Table 3.1.

The workflow is illustrated in Figure 3.2. There are four main steps. First, face alignment should be done between the subject and example images. Because the information is transferred pixel by pixel, a fully alignment (Section 3.3.1) is necessary before transferring. We adopt Active Shape Search to find the corresponding

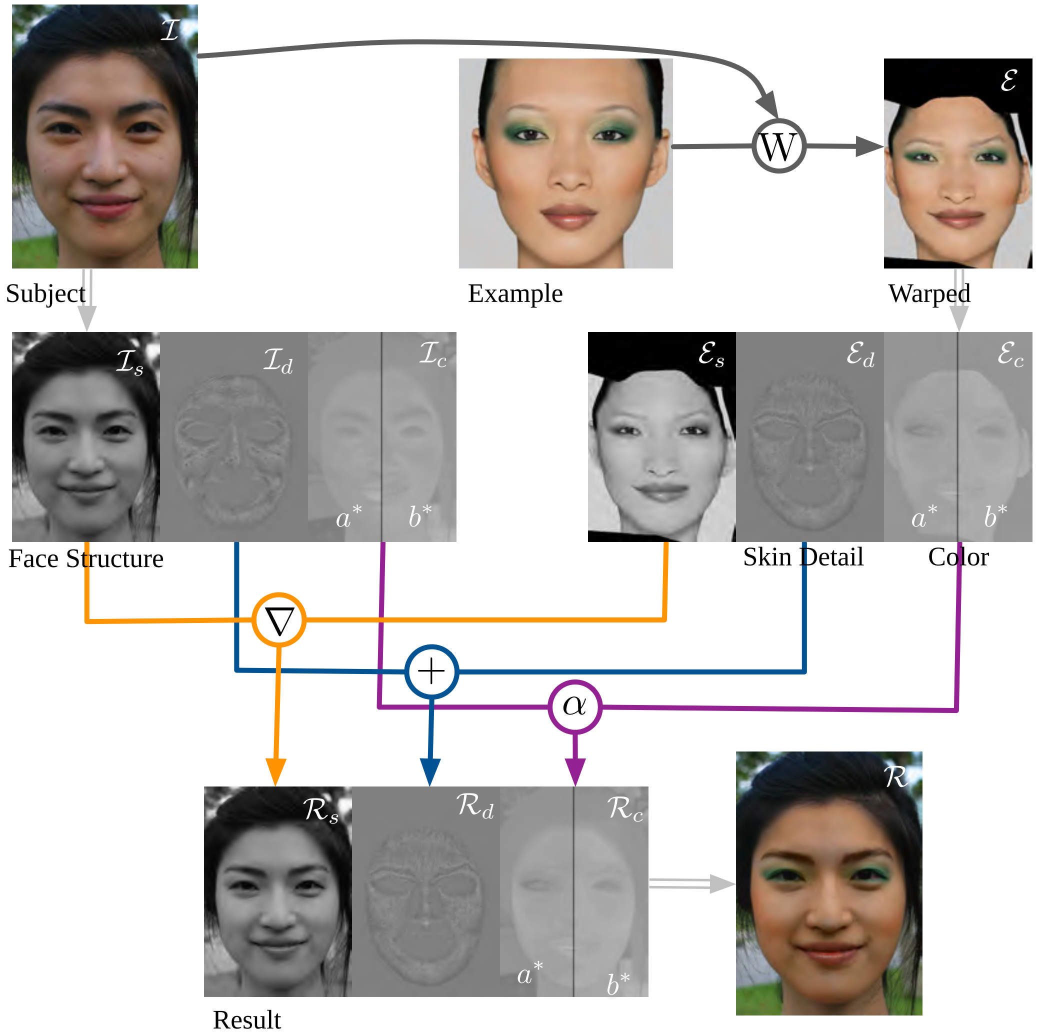


Figure 3.2: The workflow of digital face makeup. W denotes warping. ∇ denotes gradient editing. $+$ denotes weighted addition. α denotes alpha blending. In this figure, the value of skin detail layer, *i.e.* $\mathcal{I}_d$ and $\mathcal{E}_d$, is exaggerated 4 times for better visualization. Our method consists of four main steps: face alignment, layer decomposition, makeup transferring, and layer composition.

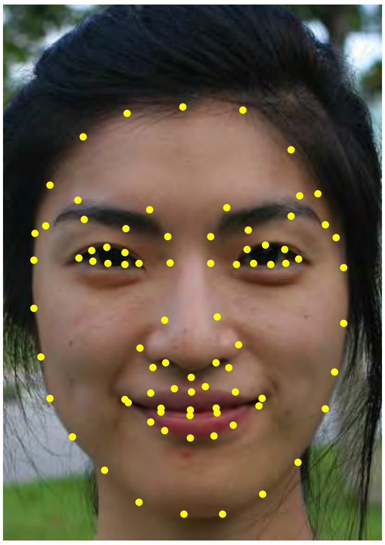


Figure 3.3: Control Points obtained by Active Shape Model.

feature points and thin plate spline to warp the example image $\mathcal{E}$ onto the subject image $\mathcal{I}$. Followed is layer decomposition (Section 3.3.2). Both $\mathcal{I}$ and $\mathcal{E}$ are decomposed into three layers: face structure layer, skin detail layer, and the color layer. Third, information from each layer of $\mathcal{E}$ is transferred to corresponding layer of $\mathcal{I}$ in different fashions: skin detail is transferred in an additive way (Section 3.3.3); color is transferred by alpha blending (Section 3.3.4); highlight and shading effects in face structure layer are transferred in the way of gradient editing (Section 3.3.5). Finally, three resultant layers are composed together.

3.3.1 Face alignment

For face alignment, we adopt the Thin Plate Spline (TPS) [Bookstein 1989] to warp the example image $\mathcal{E}$ to subject image $\mathcal{I}$. The control points required by TPS are obtained using an extended Active Shape Model (ASM) [Milborrow and Nicolls 2008]. Due to the diversity of face appearance under various possible makeup, ASM may not get the accurate position of points. Our system may still require user to slightly refine the position of control points. Since the control points have been roughly located already, the refinement does not require much effort. It usually takes less than 1 minute to refine the control points for a face. An example of refined control points is shown in Figure 3.3. There are 83 points in total on a face.

As shown in Figure 3.4, these control points have defined different face components, viz eyebrows, eyes, nose, nostrils, lips, mouth cavity (the space between lips), and other facial skin (the rest of the face). These components are further divided into three classes to be treated in different ways during makeup. These three classes ($C_1 \sim C_3$) are illustrated in different colors in Figure 3.4. C_1 (the skin region, the entire face excluding C_2 and C_3) follows the workflow illustrated in Figure 3.2. Since the texture of C_2 (lips) varies greatly from person to person and the region of C_2 is easy to deform, we use a special method to transfer the makeup style in this region (discussed in Section 3.3.6). C_3 (eyes and mouth cavity) is kept untouched all the time during entire makeup process.

3.3.2 Layer decomposition

The subject image $\mathcal{I}$ and example image $\mathcal{E}$ (after warping) are first decomposed into color and lightness layers. We then further decompose the lightness layer into face structure and skin detail layers.

In the first step, $\mathcal{I}$ and $\mathcal{E}$ are decomposed into color and lightness layers by converting them to CIELAB colorspace. The L^* channel is considered as lightness layer and a^*, b^* channel the color layer. We choose the CIELAB colorspace because it performs better than other color spaces in separating lightness from color [Wood-

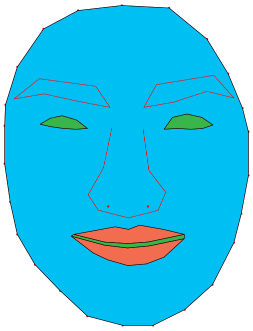


Figure 3.4: Facial components defined by control points by ASM, including eyebrows, eyes, nose, nostrils, lips, mouth cavity, and other facial skin, which are further divided into three classes, blue as C_1, red as C_2, and green as C_3. See text for detailed description.

land and Labrosse 2005], and it is approximately perceptually uniform [Lukac and Plataniotis 2007].

Second, the lightness layer is decomposed to large-scale and detail layers. The large-scale layer is considered as the face structure layer and the detail layer as skin detail layer. Large-scale/detail layer decomposition has been addressed in many works, such as [Eisemann and Durand 2004] and [Zhang et al. 2008]. The main idea is first to perform an edge-preserving smoothing on the lightness layer to obtain the large scale layer, and then to subtract (or divide) the large scale layer from the lightness layer to obtain the detail layer. In this method, we adapt a weighted-least-squares (WLS) operator recently proposed by Farbman *et al*. [Farbman et al. 2008]. An alternative method is bilateral filtering [Tomasi and Manduchi 1998], which was used in many previous works. We choose this WLS operator because of its better performance compared to the bilateral filter, especially when the blur level increases.

Suppose that the lightness layer and the face structure texture layer are denoted by l and s, respectively. The problem of solving s can be formulated as minimization of the energy function:

$$E = |s - l|^2 + \lambda H(\nabla s, \nabla l). \tag{3.1}$$

The first term $|s - l|^2$ is to keep s similar to l, while the regularization term $H(\nabla s, \nabla l)$ is trying to make s as smooth as possible.

The WLS operator described in [Farbman et al. 2008] performs the same level of smoothing all over the image; but we expect different levels of smoothness in different regions. Thus, a spatial-variant coefficient β is added to H. Then H is defined as

$$H(\nabla s, \nabla l) = \sum_p \beta(p) \left(\frac{|s_x(p)|^2}{|l_x(p)|^\alpha + \epsilon} + \frac{|s_y(p)|^2}{|l_y(p)|^\alpha + \epsilon} \right), \tag{3.2}$$

where p indexes the image pixel, ϵ is a small constant preventing division by zero, $\{.\}_x$ and $\{.\}_y$ denote the partial derivative of $\{.\}$ along x and y coordinates respectively, while α is a coefficient for adjusting the effect of ∇l on ∇s. We use $\alpha = 1.2$ and $\lambda = 0.2$ in all our experiments.

According to the Variational Principle, to minimize E, s must satisfy the following Euler-Lagrange equation, after reorganizing,

$$s - \lambda \left(\frac{\partial}{\partial x} \left(\frac{\beta s_x}{|l_x|^\alpha + \epsilon} \right) + \frac{\partial}{\partial y} \left(\frac{\beta s_y}{|l_y|^\alpha + \epsilon} \right) \right) = l. \tag{3.3}$$

The discrete solution to (3.3) is given by

$$\begin{aligned} &\text{For each } p, \\ &s(p) + \sum_{q \in \mathcal{N}(p)} \frac{\beta(p)(s(p) - s(q))}{|l(p) - l(q)|^\alpha + \epsilon} = l(p), \end{aligned} \tag{3.4}$$

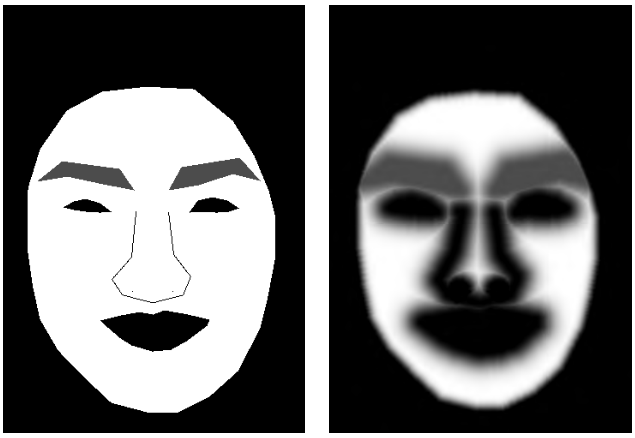


Figure 3.5: Illustration of β used in spatial-variant edge preserving smoothing. (a) Initial definition of β value. (d) Visualization of β value defined by Equation (3.5), brighter pixel denoting higher β value.

where $\mathcal{N}(p)$ denotes the four neighbors of p. A detail derivation is provided in Appendix B.2. (3.4) is a banded linear system with each equation having five unknowns. Such a sparse linear system could be solved efficiently.

We expect that $\beta(p)$ is low inside the facial components, and equal to 1 over the skin area. As shown in Figure 3.5(a), $\beta(p)$ is 0.3 in the eyebrow region, 0 in other facial component region, and 1 in facial skin. In addition, we also expect $\beta(p)$ changes smoothly over the whole image. Thus, we further define $\beta(p)$ as

$$\beta(p) = \min_{q}\left(1 - k(q) \cdot e^{-\frac{(q-p)^2}{2\sigma^2}}\right), \tag{3.5}$$

where q indexes the pixel over the image. $k(q)$ is 0.7 for eyebrows, 0 for skin area, 1 for other facial components. The value of σ^2 is set to $\min(\text{height}, \text{width})/25$. Then, we have $\beta(p)$ as shown in Figure 3.5(b).

Since the L^* channel is approximately perceptually uniform, we use subtraction

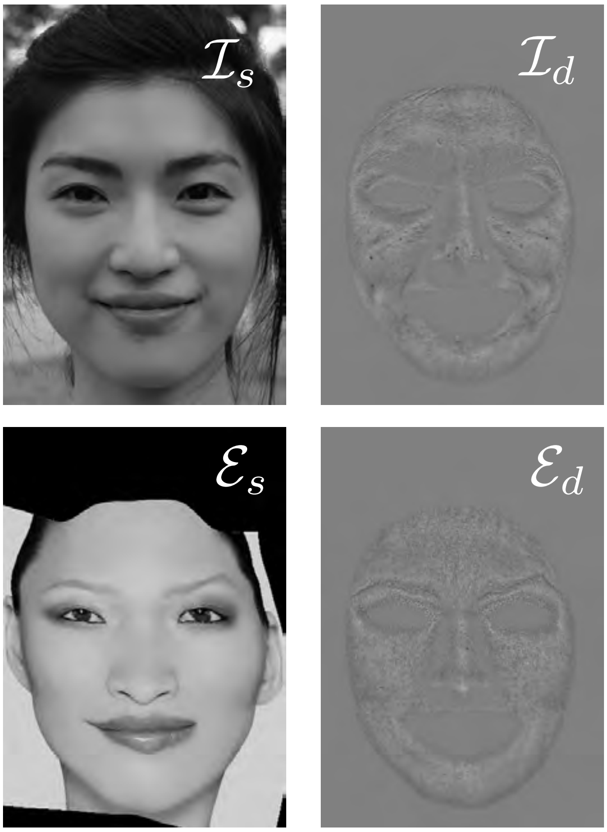


Figure 3.6: Two examples of face structure and skin detail layers. Upper row: Subject image. Bottom row: Example image. Left: Face structure layer. Right: Skin detail layer.

to obtain the skin detail layer d from lightness layer l, *i.e.*

$$d(p) = l(p) - s(p). \tag{3.6}$$

Two examples of our decomposition results are shown in Figure 3.6. We can see that detail is controlled well by $\beta(p)$. Because β is zero in the eyes and mouth regions and outside the facial region, the skin detail layer there is zero.

In the rest of this chapter, we will use $\{.\}_s$, $\{.\}_d$, $\{.\}_c$ to denote $\{.\}$'s face structure layer, skin detail layer, and color layer, respectively.

3.3.3 Skin detail transfer

We use a straightforward method for skin detail transfer that the resultant skin detail layer $\mathcal{R}_d$ is a weighted sum of $\mathcal{I}_d$ and $\mathcal{E}_d$, *i.e.*

$$\mathcal{R}_d = \delta_{\mathcal{I}} \mathcal{I}_d + \delta_{\mathcal{E}} \mathcal{E}_d, \tag{3.7}$$

where $0 \le \delta_{\mathcal{I}}, \delta_{\mathcal{E}} \le 1$. The values of $\delta_{\mathcal{I}}$ and $\delta_{\mathcal{E}}$ control the contribution of each component.

For different applications, different $\delta_{\mathcal{I}}$ and $\delta_{\mathcal{E}}$ values can be used. As we mentioned in Section 3.1, the purpose of foundation and loose powder in physical makeup is to conceal the original skin detail and to introduce new skin detail. Thus, we set $\delta_{\mathcal{I}} = 0$ to conceal $\mathcal{I}_d$, and $\delta_{\mathcal{E}} = 1$ to transfer $\mathcal{E}_d$ to $\mathcal{R}_d$. This is a typical setting for beauty makeup transfer. It is used in all our beauty makeup experiment results except the ones showing different manipulation of makeup effects (Figure 3.7). In some cases, we can also set $\delta_{\mathcal{I}} > 0$ to keep some original skin detail. For example, in adding beard effects, we would keep the original detail ($\delta_{\mathcal{I}} = 1$ and $\delta_{\mathcal{E}} = 0$) in the facial region and transfer details from the example in beard region ($\delta_{\mathcal{I}} = 0$ and $\delta_{\mathcal{E}} = 1$). The detail will be discussed later in Section 3.4.4 and 3.4.5.

Note that the sum of two weights is not required to be 1 because $\mathcal{R}_d$ can come from any amount of $\mathcal{I}_d$ or $\mathcal{E}_d$. In addition, the sum should not be very small, otherwise the face in result image $\mathcal{R}$ would be not realistic due to lack of skin detail.

3.3.4 Color transfer

The resultant color layer $\mathcal{R}_c$ is an alpha-blending of color layers of $\mathcal{I}$ and $\mathcal{E}$, *i.e.*

$$\mathcal{R}_c(p) = \begin{cases} (1-\gamma)\mathcal{I}_c(p) + \gamma\mathcal{E}_c(p) & p \in C_1 \\ \mathcal{I}_c(p) & \text{otherwise} \end{cases} . \tag{3.8}$$

The value of γ is to control blending effect of two color layers. The result in Figure 3.1 is obtained with $\gamma = 0.8$.

3.3.5 Highlight and shading transfer

The highlight and shading effects of makeup lie in the L^* channel. Because the face structure layer is actually the large scale layer of L^* channel, the smooth change of the highlight and shading effects remain in the face structure layer. Since these effects are important for makeup, we should transfer them across this layer.

Because face structure layer contains identity information, we can neither directly copy $\mathcal{E}_s$ over $\mathcal{I}_s$ nor blend them. Instead, we adapt a gradient-based editing method. The idea is to add only large changes of $\mathcal{E}_s$ to $\mathcal{I}_s$. Doing this, we assume that these changes are due to makeup. This assumption holds if the illumination of $\mathcal{E}$ is approximately uniform.

Gradient-based editing can preserve the illumination of $\mathcal{I}$, transfer the highlight and shading effects, and meanwhile yield smooth result. Editing an image in the gradient domain was introduced by Pérez *et al.* [Pérez et al. 2003] and was employed in many later works. The gradient-based method used here is similar. The gradient of R_s is defined as

$$\nabla\mathcal{R}_s(p) = \begin{cases} \nabla\mathcal{E}_s(p) & \text{if } \beta(p)\|\nabla\mathcal{E}_s(p)\| > \|\nabla\mathcal{I}_s(p)\| \\ \nabla\mathcal{I}_s(p) & \text{otherwise} \end{cases} . \tag{3.9}$$

Since only the gradient of the face region (C_1) is changed (but not its boundary or regions outside C_1), the process of solving the resultant face structure layer $\mathcal{R}_s$ from its gradient is equivalent to solving a Poisson equation with Dirichlet boundary condition.

3.3.6 Lip makeup

The makeup effect of lip region (C_2) is quite different from that of face skin (C_1). In physical makeup, cosmetics on lips (*e.g.* lipstick) usually preserve or highlight the texture of lips, rather than conceal it as in face skin. Thus, lips region makeup needs to be treated in a different way.

The main idea is to fill each pixel of $\mathcal{R}$ with pixel value from $\mathcal{E}$ guided by $\mathcal{I}$. Then the makeup effect is similar to $\mathcal{E}$ and the texture is similar to $\mathcal{I}$. Specifically, for each pixel p in $\mathcal{I}$, we search for a pixel q in $\mathcal{E}$ so that $\mathcal{E}(q)$ and $\mathcal{I}(p)$ are as similar as possible, while q and p are as close as possible.

Suppose that lip region after makeup is denoted by $\mathcal{M}$. For each $p \in C_2$, we have

$$\mathcal{M}(p) = \mathcal{E}(\tilde{q}), \tag{3.10}$$

where

$$\tilde{q} = \arg\max_{q \in C_2} \{G(|q - p|)G(|\mathcal{E}(q) - \mathcal{I}(p)|)\} \tag{3.11}$$

where $G(.)$ denotes Gaussian function. For $|\mathcal{E}(q) - \mathcal{I}(p)|$, we use the difference of pixel values in only L^* channel after histogram equalization of $\mathcal{E}$ and $\mathcal{I}$ separately.

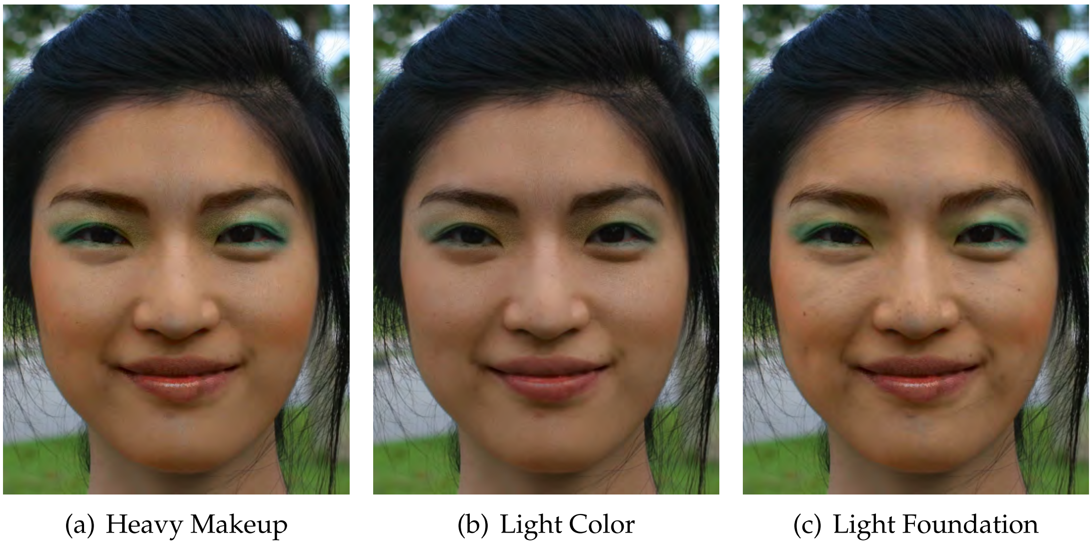
The makeup result of lips region $\mathcal{M}$ is then merged to result image $\mathcal{R}$. The L^* channel of $\mathcal{M}$ is added to L^* channel of $\mathcal{R}$ with a gradient replacement. The a^*, b^* channel of $\mathcal{M}$ replace corresponding region in $\mathcal{R}$. Some examples are shown in 3.4.

3.4 Experiments and Results

3.4.1 Beauty makeup

Our method can manipulate makeup effects in various ways. In case of heavy foundation where the foundation covers the original skin, we use $\delta_{\mathcal{I}} = 0$, $\delta_{\mathcal{E}} = 1$ to transfer all skin detail from $\mathcal{E}$ to $\mathcal{I}$. An example of result is shown in Figure 3.7(a). To simulate light foundation effect, we use $\delta_{\mathcal{I}} = 1$, $\delta_{\mathcal{E}} = 0$ to keep original skin detail. An example is Figure 3.7(c). Adjusting $\delta_{\mathcal{I}}$, $\delta_{\mathcal{E}}$ can simulate different level of foundation effect. For a good result of concealing the skin detail of $\mathcal{I}$, it is required that the skin detail of $\mathcal{E}$ should be only due to makeup, *i.e.* $\mathcal{E}$ should have heavy foundation (usually also with loose powder); otherwise $\mathcal{R}$ may contain imperfection of skin detail from $\mathcal{E}$. Another manipulation is on the level of makeup color by adjusting γ. As shown in Figure 3.7(b), we used $\gamma = 0.5$ for a light makeup color.

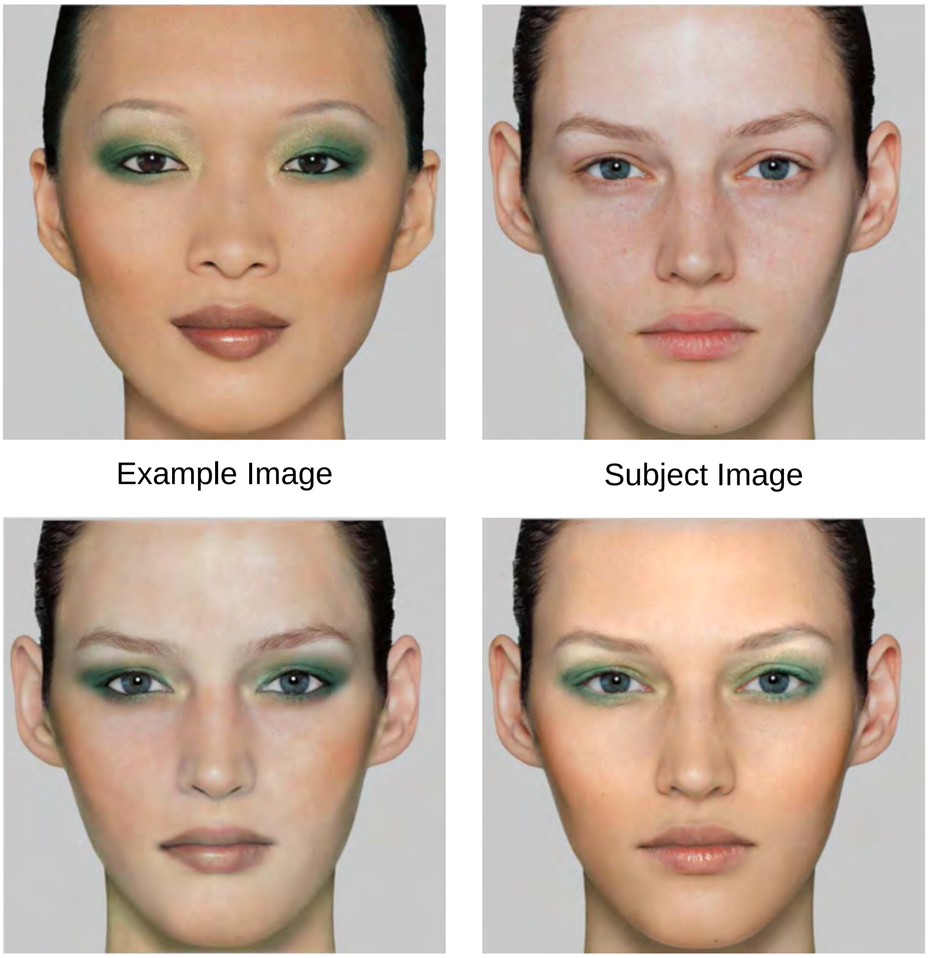
A comparison of our result with that of Tong *et al.* is shown in Figure 3.8. Note that they used an additional image as example of "before" makeup, which is not shown here. In contrast, we only use the "after" image. Our result shows vivid color and more similar makeup style to the example image. From the close-up



(a) Heavy Makeup (b) Light Color (c) Light Foundation

Figure 3.7: Manipulation of makeup effects. (a) Heavy makeup effect (heavy foundation and heavy color). (b) Light color effect. (c) Light foundation effect. The subject and example are the same as in Figure 3.1.

view (Figure 3.10) of the eye, we can see that our method preserves the structure much better than theirs. Moreover, the highlight of lips in our result is successfully transferred while the texture is preserved. It is much more similar to real makeup. For skin tone, we have different representations. Our result takes the skin tone from example image, while their result keeps the original tone. As we presented before, our aim was to apply the makeup style faithfully from the example image, including color, skin detail. In the physical makeup, tone color is usually also considered to be a part of makeup. Another consideration is that color used in makeup should match the skin tone [Yamaguchi 2004]; makeup transferring without skin tone may destroy the original harmony.



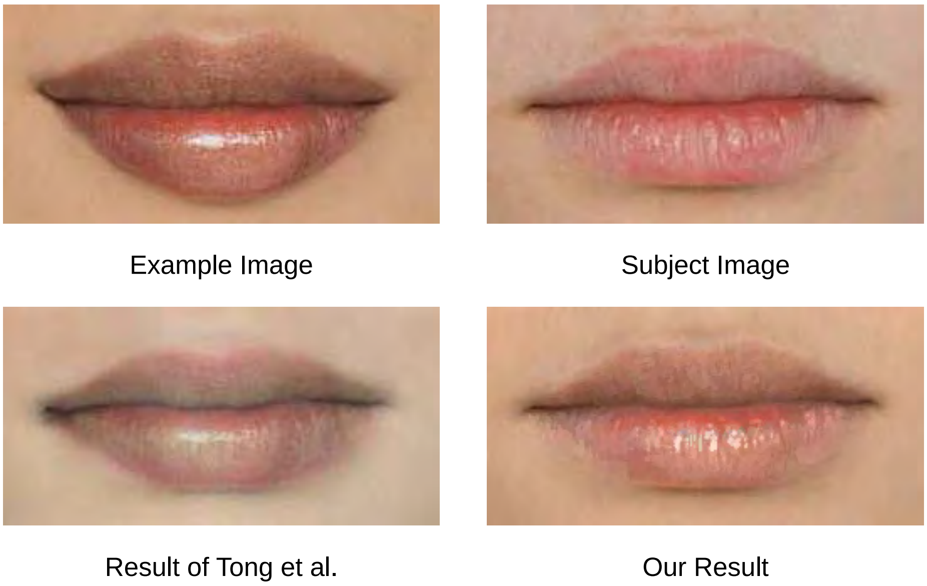
Example Image

Subject Image

Result of Tong et al.

Our Result

Figure 3.8: Comparison of our result with that of Tong *et al*. [Tong et al. 2007]. Note that Tong *et al*. employed an additional "before"-makeup example (not shown here), unlike our method.



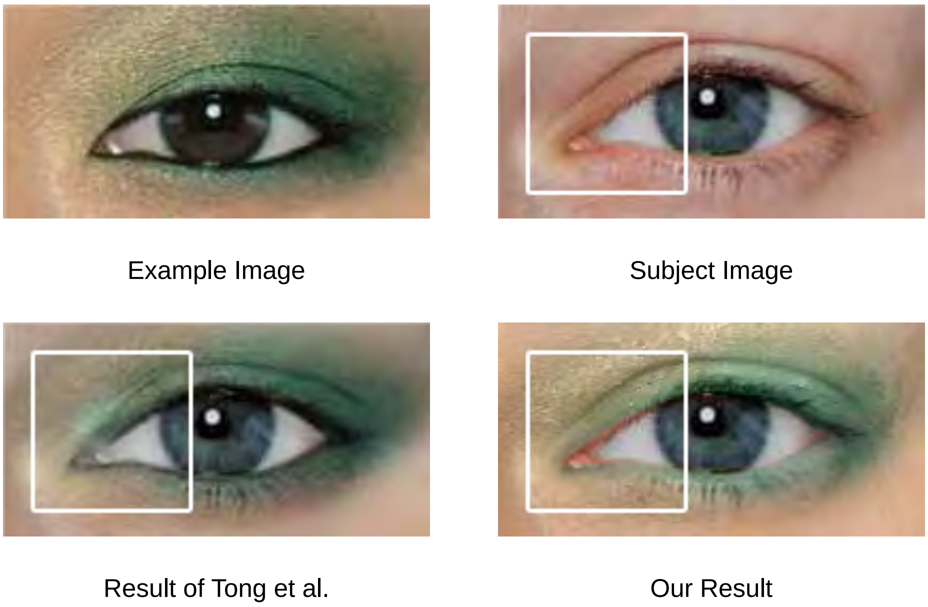
Example Image

Subject Image

Result of Tong et al.

Our Result

Figure 3.9: Comparison of our result with that of Tong *et al*. (lip close-up).



Example Image

Subject Image

Result of Tong et al.

Our Result

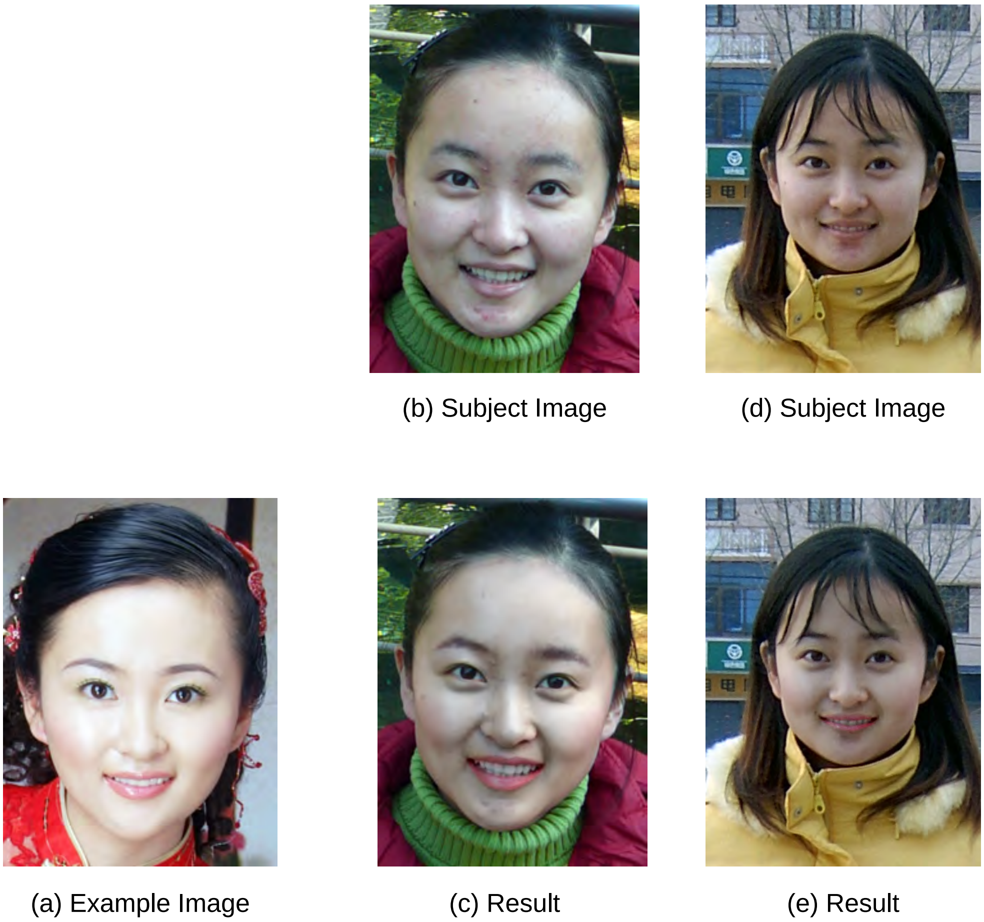
Figure 3.10: Comparison of our result with that of Tong *et al*. (eye close-up).

3.4.2 Photo retouching

Our method can be also used to retouch a face photo. This is almost the same as face makeup by example. We address it separately because this application itself is interesting and useful. Instead of photographs from others, users may also provide their own previously well-taken photograph, with or without makeup, as the example for retouching. We show an example in Figure 3.11. A photo (Figure 3.11(a)) with makeup taken in a professional studio is provided as the example. Two other photos of the same person (Figure 3.11(b) and (d)) taken by an amateur user are provided as the subject images. In the results (Figure 3.11(c) and (e)), the imperfection of the face skin was concealed and pleasant face skin was introduced successfully.

3.4.3 Makeup by portraiture

An interesting application is transferring makeup from a portraiture to a real face photo. This is similar to makeup transfer between real face photos. However, portraitures usually have non-realistic artifacts in the skin detail layer due to the drawing material, or aging and discoloration. These artifacts are not suitable for a face photo. For example, in Figure 3.12, the portraiture has some noise. Transferring it makes result unnatural (Figure 3.12(c)). In this case, we have to set $\delta_{\mathcal{E}} = 0$ and $\delta_{\mathcal{I}} = 1$ to restrain the noise. As a result, the result image becomes much better (see Figure 3.12(d)).



(b) Subject Image

(d) Subject Image

(a) Example Image

(c) Result

(e) Result

Figure 3.11: Examples of photo retouching. (a) The example image, taken in a professional studio, provides the desired retouching result. (b)(d) Photos of the same person in (a), taken by an amateur. (c)(e) The retouching results of (b) and (d), respectively.

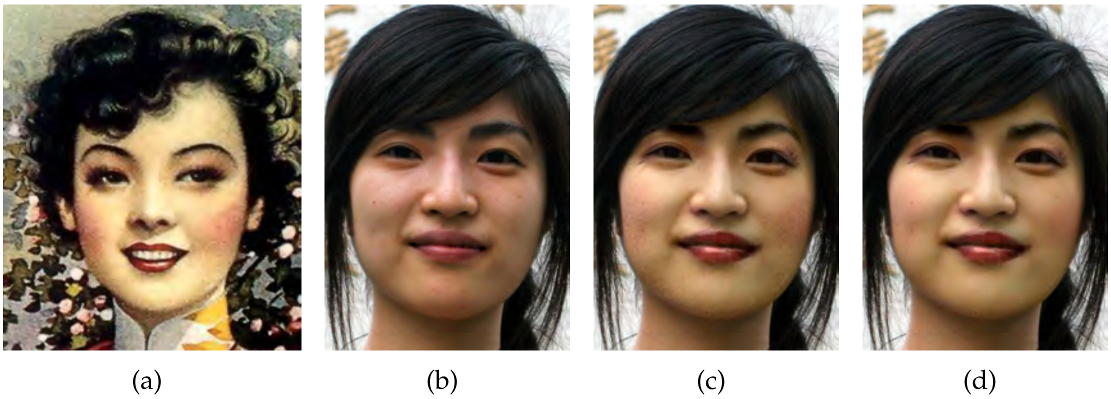


Figure 3.12: Makeup by portraiture. (a) An old portraiture scan, as the example image. (b) A photo, as the subject image. (c) The makeup result with skin detail form (a). (d) The makeup result with skin detail from (b).

3.4.4 Aging effects

Face makeup is not always to enhance one's facial appearance, but sometimes to achieve special purposes. In this subsection, we discuss how to apply our method to aging effects. Our method can remove facial texture and introduce new texture from another example face. This allows, as discussed in previous sections, removing small imperfections, like wrinkles, moles, *etc*. from one's face. In the opposite direction, we can also add wrinkles to a young face to make it look elder. To transfer wrinkles, we use an elder face as the makeup example $\mathcal{E}$ (Figure 3.13(a)). The color transfer coefficient γ is set to 0. The face detail transfer coefficient $\delta_{\mathcal{I}} = 0$ and $\delta_{\mathcal{E}} = 1$. Figure 3.13 shows an example of wrinkles transfer.

3.4.5 Beard transfer

Some males consider beard as an important component of facial appearance. Previewing different styles of beard is an interesting application. To transfer beard from an example image to a subject, we constrain the makeup process within the

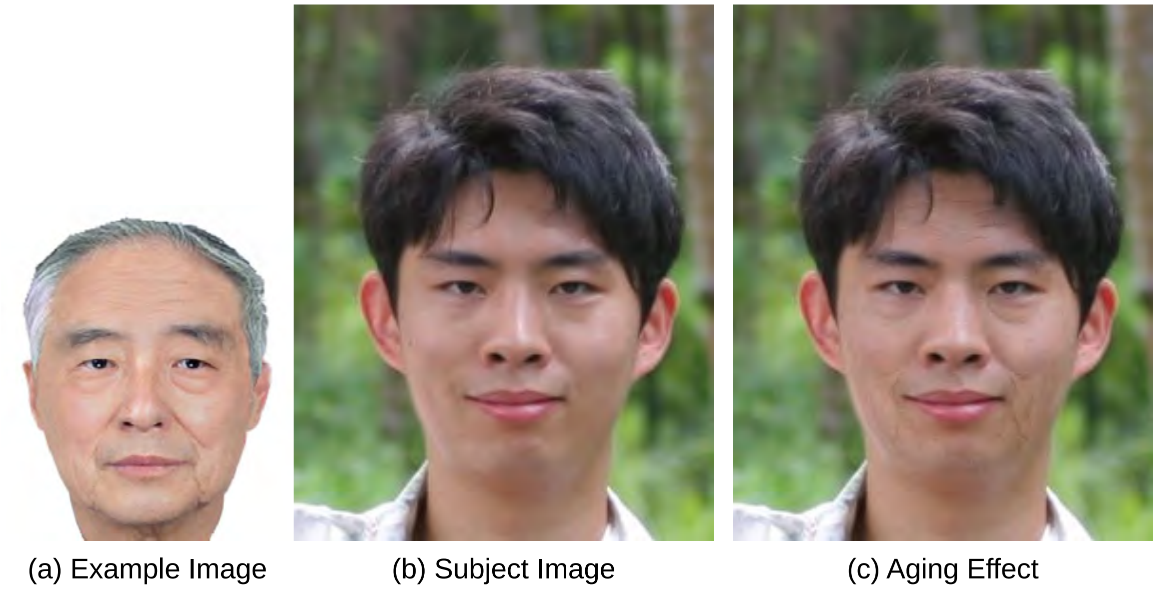


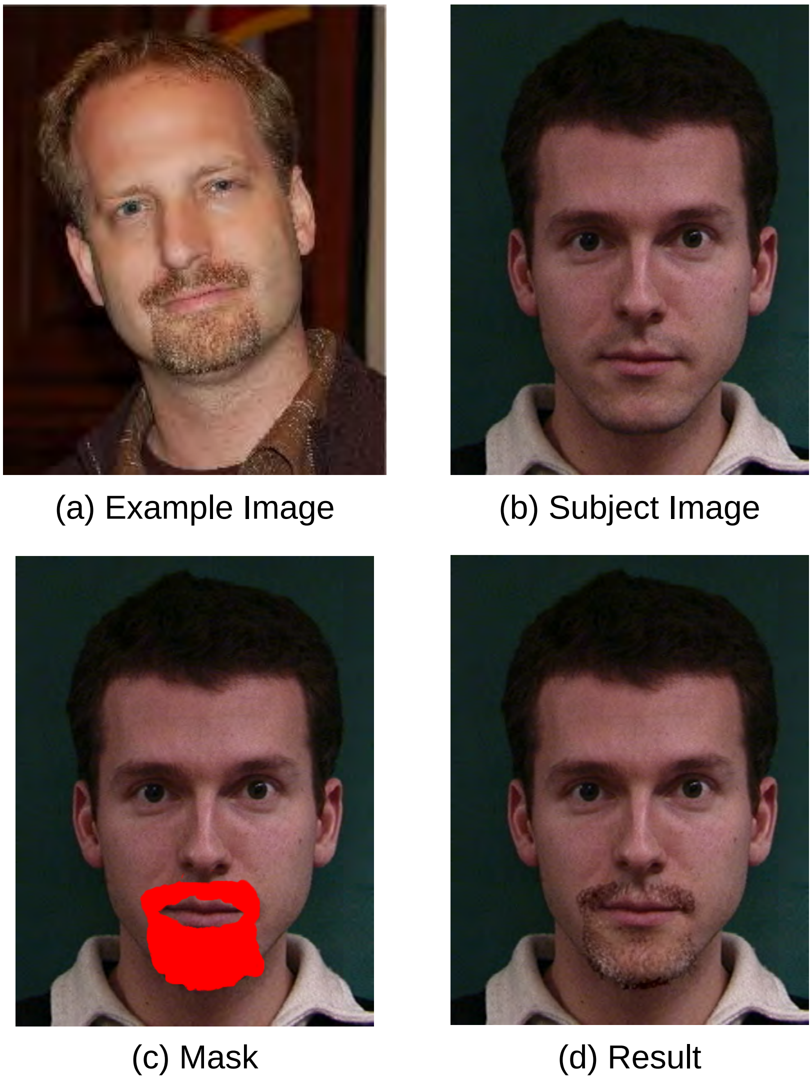
Figure 3.13: Aging effects. (a) An old man with obvious wrinkles. (b) A young guy. (c) The makeup result by transferring wrinkles from (a) to (b).

beard area. We used a pre-marked map to define beard region (Figure 3.14(c)). However, beard detection is not difficult since we have the feature points on face. Unlike aging effects, we still need the color makeup, and face detail transfer. An example is shown in Figure 3.14.

The running time for beauty makeup is about 3 - 5 seconds with matlab code on a Core 2 Duo 2.33GHz Computer for a 1000 × 1000 color image. For aging effects, it takes about 2 seconds.

3.5 Summary and Discussion

In this chapter, we have presented a method of creating makeup upon an image with another example image as the makeup style. The main idea is simple yet powerful. Promising results have demonstrated the effectiveness. One major



(a) Example Image

(b) Subject Image

(c) Mask

(d) Result

Figure 3.14: Beard transfer. (a) Example image providing beard. (b) Subject image without beard. (c) The makeup result by transferring beard from (a) to (b).

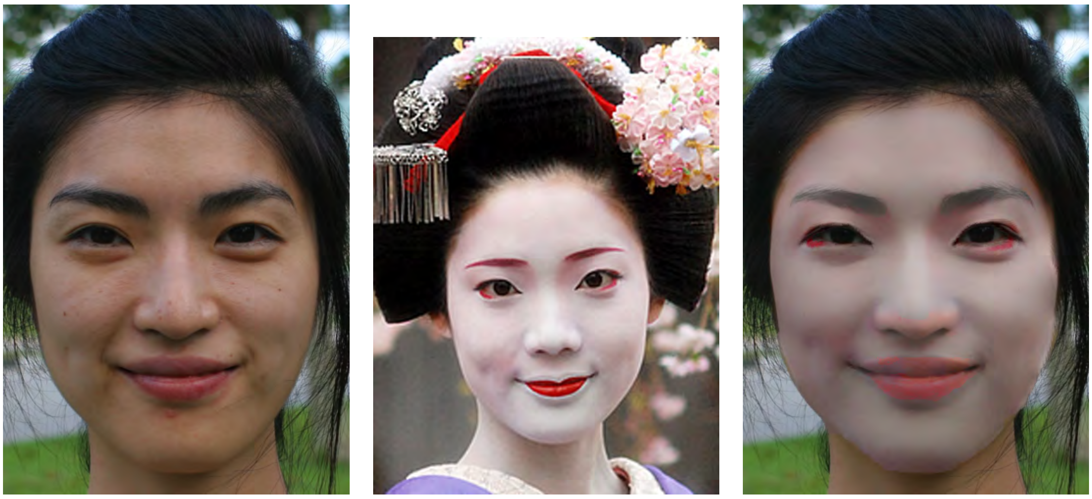
Table 3.2: Summary of different parameters for different types of makeup.

Type of makeup	**Recommended Parameters**	**Description**
Beauty makeup	$\delta_{\mathcal{I}} = 0, \delta_{\mathcal{E}} = 1, \gamma = 0.8$	Transferring all skin detail from example to subject. The original skin detail of subject is concealed and new skin details is introduced. Color makeup is almost copied.
Makeup by portraiture	$\delta_{\mathcal{I}} = 1, \delta_{\mathcal{E}} = 0, \gamma = 0.8$	There are noise or special texture in the skin detail of example image. Keep the skin detail of subject image.
Aging effects	$\delta_{\mathcal{I}} = 0, \delta_{\mathcal{E}} = 1, \gamma = 0$	Transfer the skin detail (wrinkles) only. No color transfer.
Beard transfer	Same as beauty makeup.	Only in beard region.

advantage of our method is that only one example image is required.

This method have several parameters to control the effects of makeup, namely $\delta_{\mathcal{I}}, \delta_{\mathcal{E}}, \gamma$. For beauty makeup, we have a set of recommended parameters. For other applications, we have shown results and related parameters. Here we summarize the parameters and the meaning in Table 3.2.

The proposed method has several assumptions. The active shape model we adopt assumes that the face is frontal and upright. Our system is currently tested only with both the subject and example images being nearly frontal. But we envision that it can work well on any pose as long as the pose difference is not large between the subject and example images. One future work is to extend this method to any pose. This method assumes the illumination in the example image is uniform, but it is not necessary to be the same as in subject image. If any shadow or specularity exists, they will be also transfered to subject image. Possible solution to this problem is shadow or specularity detection and removal.



(a) Subject image (b) Maiko makeup example (c) Makeup result

Figure 3.15: Maiko makeup example. The foundation is uniformly white, which is not transferred faithfully. In the result, white makeup appears gray and unnatural.

Limitation. Our method does not work well for black and white makeup. A typical example is maiko makeup, as shown in Figure 3.15. In our result, the white regions appears gray and unnatural. The white color is the foundation in physical makeup; but our method only transfers the texture introduced by foundation. The white color is interpreted as no color in LAB colorspace; the lightness of a white color is very important to human perception. But the lightness is not transfered in our method. Thus, the white color in our result appears gray. One possible solution to solve this problem is to add lightness transfer with user control on the amount of lightness transfer.

Currently, only one image is used as example. It would be interesting and practical to extend our current method to multiple example images. In this case, the consistency between different makeup styles may be an issue.

Since only skin detail and color are required from the example image, if we warped these information to a canonical face, the original face structure of the example image is not required any more. Doing so indeed helps to preserve the privacy of the makeup actors. Another future work is to collect more makeup examples and build up a makeup engine enabling users to browse different makeup styles with their own faces, as the previously introduced beauty salon case.

Expression in subject image does not affect much, although expressions in example may result in some artifacts. Current, our system can handle small expression in example image. However, if expression is large enough to cause deep dimples on cheeks or wrinkles, they will be indistinguishable from makeup effect. Part or full of these will be transferred to subject image. This is undesirable unless the user intend to do so, like what we have demonstrated in wrinkles transfer.

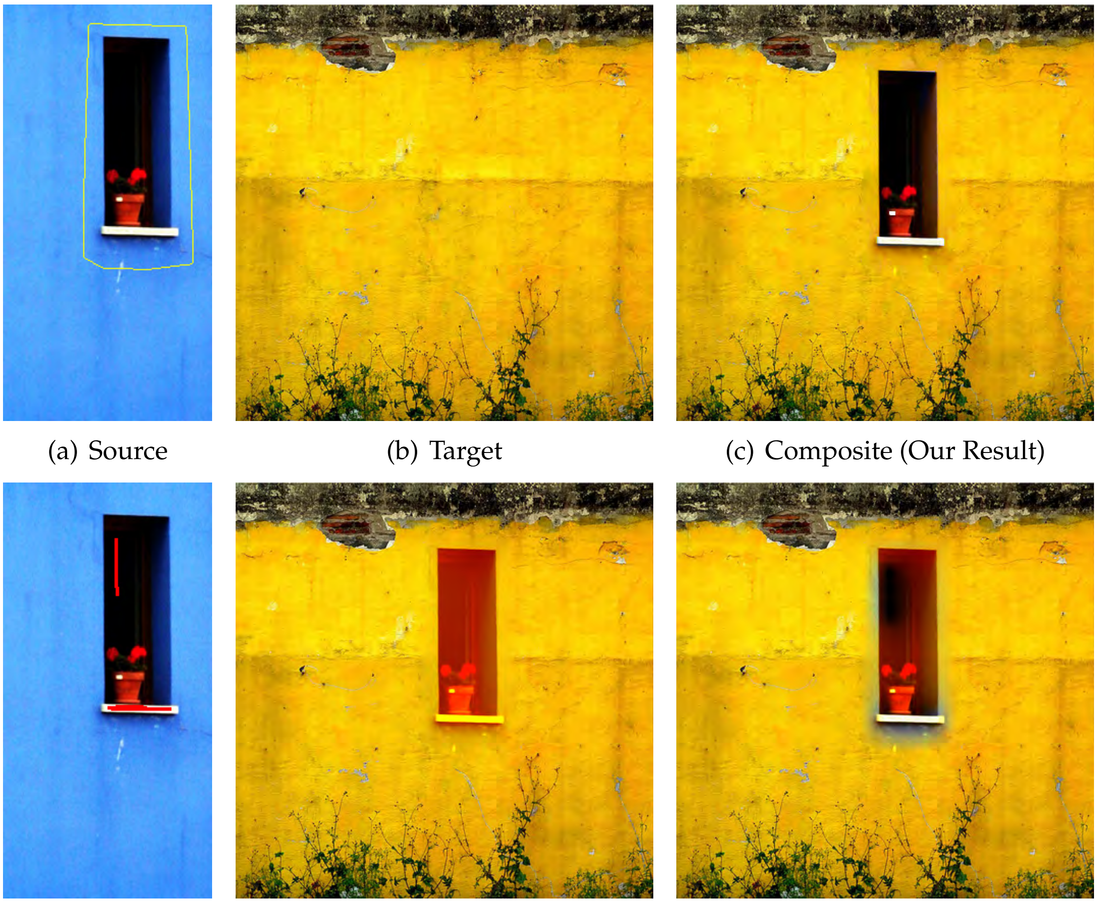
Chapter 4

Seamless Image Compositing

4.1 Overview

Digital image compositing is a process of blending a region of interest (ROI) from a source image onto a target image. An example of image compositing is shown in Figure 4.1. An ROI containing a window frame is selected by a user, as shown inside the yellow line in Figure 4.1(a). The ROI is then blended onto the target image (Figure 4.1(b)) to produce the final composite (Figure 4.1(c)).

Poisson image editing (PIE) [Pérez et al. 2003] is a method to seamlessly blend ROI onto the target image. In PIE, the gradient of ROI is pasted onto that of the target image. Then, the result image is reconstructed from the gradient domain by solving a Poisson equation. The main advantage of PIE is the seamless boundary around the pasted ROI in result images. To achieve this goal, the gradient inside ROI is to be kept unchanged to make the result visually similar to the source image. On the other hand, a hard constraint along the boundary is enforced to make the boundary of pasted ROI agree with the target image. PIE keeps the relative values


(a) Source (b) Target (c) Composite (Our Result)

(d) Markup (e) Composite (PIE) (f) Composite (PIE with Markup)

Figure 4.1: An example of seamless image compositing. (a) The source image. ROI is the region inside yellow line. (b) The target image. (c) The final composite of our proposed approach by pasting the window from (a) onto (b). (d) User markup (the red scribbles) used in our approach. (e) Composite of Poisson image editing. Note the color shift in the ROI. (f) Composite of Poisson image editing with color constraint by user markup in (d). Note the halo effects around the scribbles. See the text for details.

in ROI; however, the absolute values (the color) of ROI may shift in the process of blending. In some cases, the color shift of ROI is desirable (some examples demonstrated in [Pérez et al. 2003; Jia et al. 2006]) because this makes the tone of pasted ROI similar to the target image. However, the amount of the color change depends on the difference of boundaries of ROI in the source and that in target image, which is uncontrollable by users. Despite the case of tone matching, color shift is usually undesirable. As shown in Figure 4.1(e), PIE produced a large color shift in the result. The window and the flower in ROI was expected to be the same as in the source image; however, the whole ROI became reddish.

An improvement of PIE, proposed by Jia *et al*. [Jia et al. 2006], was trying to solve the problem of PIE producing unnatural blurring artifacts when the boundary of ROI does not meet the target image very well. In this chapter, we do not address this problem. Instead, we introduce a way of producing seamless image composite while preserving the color.

4.1.1 Related work

Image matting, such as [Chuang et al. 2001; Sun et al. 2004], and image segmentation techniques, such as [Li et al. 2004; Mortensen and Barrett 1995], are commonly used technologies to extract objects from an image. The extracted objects can be pasted onto a target image to produce a composite. These techniques are suitable for foreground and background being clearly separated. However, in many cases, the foreground interacts with the surroundings, without obvious boundaries, *e.g*. the shadow region below the window frame shown in Figure 4.1. These techniques are not suitable.

Table 4.1: Notation used in this chapter.

Notation	Meaning
Ω	Unknown region in the final composite.
$\partial\Omega$	The boundary of Ω.
C	The whole image of the final composite.
r	Pixel values in Ω. The unknowns to be solved.
s	Pixel values corresponding to r in the source image.
t	Pixel values of the target image. The actual useful part is $C - \Omega$.
M_1	User markup of foreground.
M_2	User markup of background.

Another work, interactive digital photomontage [Agarwala et al. 2004], aims to assemble images of the same or similar scenes together. Regions from different images are picked out by users and the seams among different regions are minimized with Graph-cut. The final composite keeps the color from each source images. However, if the source images are different from each other, it is difficult to find invisible seams to stitch the images together.

In this chapter, we introduce a method that takes the advantage of gradient-based editing to produce a seamless composite. We use weighted least squares in reconstructing the final composite from the gradient domain. With the boundary of ROI and the user markup being the hard constraints, the colors are faithfully preserved in specified regions. The idea of employing weighted least squares to enforce the similarity within a region can also be found in some previous works, such as colorization [Levin et al. 2004], tone adjustment [Lischinski et al. 2006] and edge-preserving image decomposition [Farbman et al. 2008].

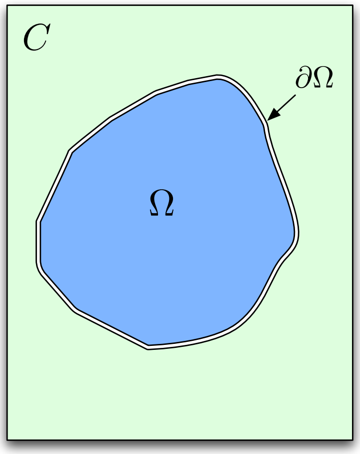


Figure 4.2: Illustration of notations.

4.2 Methodology

The goal of image compositing is to blend the region of interest (ROI) from the source onto the target image. In the final composite, only the region pasted from the source image is unknown, while the rest is directly from the target. As illustrated in Figure 4.2, we use Ω denote the unknown region in the final composite C, with $\partial\Omega$ being the boundary. The value of $(C - \Omega)$ is from the target image t. We use r denote the pixel values of Ω, which are the unknowns to be solved; s denotes the corresponding values of the source image. The notation used in this chapter is summarized and listed in Table 4.1.

4.2.1 Poisson image editing

In Poisson image editing [Pérez et al. 2003], the difference between gradient of r and that of s is minimized, while the value r on the boundary $\partial\Omega$ is enforced to be the same with t. Thus, r is the solution of the following minimization problem,

$$\min_{r} \int |\nabla r - \nabla s|^2 \quad \text{with} \quad r|_{\partial\Omega} = t|_{\partial\Omega}, \tag{4.1}$$

where ∇ is the gradient operator. The solution to Equation (4.1) is given by the following Poisson equation with Dirichlet boundary conditions,

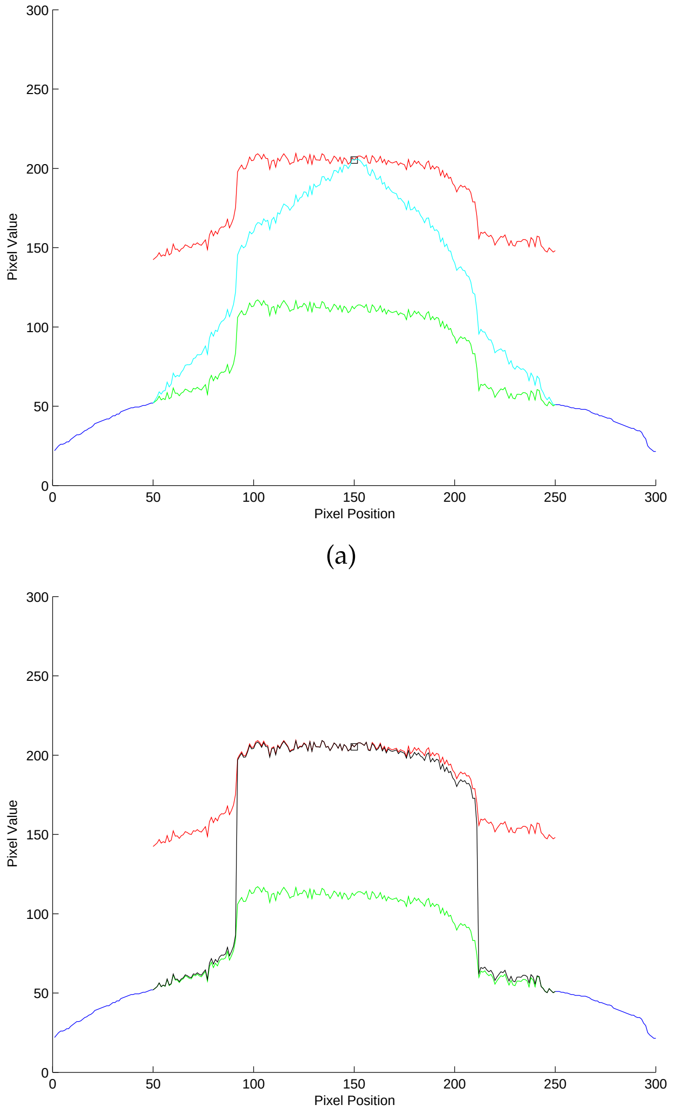
$$\triangle r = \triangle s \quad \text{with} \quad r|_{\partial\Omega} = t|_{\partial\Omega}, \tag{4.2}$$

where $\triangle$ is the Laplacian operator.

We use a 1D example illustrate this idea, as shown in Figure 4.3(a). The horizontal axis denotes the pixel position, while the vertical axis denotes the pixel value. The region of blue lines is $(C - \Omega)$, and the region of the green line is Ω. The red line is the source "signal", s. The middle part of s, which has larger value than the surroundings, indicates the object to be pasted. Usually, the ROI selected is slightly larger than the object inside. The result of PIE is shown as the green line, which fits two ends of the blue lines and keeps the gradient of s. It appears that PIE successfully blended s onto t without visible seams. However, one can easily find that the absolute values of reconstructed "signal" r shift quite a bit from the source s. In the image domain, this results in a color shift of the ROI in the process of blending, such as the example shown in Figure 4.1(e).

4.2.2 User markup constraints

To correct the color shift, an intuitive way is adding additional constraints: user markup indicating the foreground that should be the same as in the source image,



(a)

(b)

Figure 4.3: An illustration of Poisson image editing and the proposed color-preserving compositing in 1D. (a) Red: the source “signal”. Blue: the target “signal”. Green: result of PIE. Note the color shift from r down to s. Cyan: the result of PIE fixing the middle value of r to s and the boundary. This creates a halo effect. (b) Red, green, blue: same as in (a). Black: the result of our proposed approach. Note that the middle part of r is well preserved, while r is seamlessly connected to t.

s. If there are holes in the foreground, extra user markup should be involved indicating the holes (as background) as well. These constraints are added as hard constraints, *i.e.*

$$r|_{M_1} = s|_{M_1} \text{ and } r|_{M_2} = t|_{M_2}, \tag{4.3}$$

where M_1 and M_2 denote markup of foreground and background, respectively.

In the 1D example, a point in the middle of Ω is used as user markup of foreground. The result of Equation (4.2) with additional constraints of Equation (4.3) is shown as the cyan line in Figure 4.3(a). With the user markup as hard constraints, the error of the objective function usually becomes much larger. In case of using PIE, this error is evenly spread on each pixel position, resulting in a gradual change in r. The result "signal" r is more similar to the red line s when the position is nearer to the user markup, otherwise more similar to the green line (the result without user markup). In the image domain, the gradual change will undesirably result in a halo effect around the user markup.

4.2.3 Weighted least squares

The main idea to reconstruct r from the gradient domain is to minimize the error between ∇r and ∇s with hard constraints of $\partial\Omega$ and user markup. Instead of spreading the error evenly on each pixel in PIE, we add different weights to different pixel positions to control the desired similarity between ∇r and ∇s.

Ideally, the color of the regions from the foreground should be preserved during compositing; and color preserving should be restricted to these regions only. For a "region", we mean an area where the colors are similar. Within a region, small gradients in the source image should be more likely to be kept small in the result.

In other words, the larger the gradient of a pixel in s is, the more this pixel should share the error from the overall minimization error.

In 2D image domain, the solution of r can be formulated as minimization of the energy,

$$\mathcal{E} = \sum_{p \in \Omega} \Big(\alpha(p) \big(r_x(p) - s_x(p) \big)^2 + \beta(p) \big(r_y(p) - s_y(p) \big)^2 \Big) \\ \text{with} \quad r|_{\partial\Omega \cup M_2} = t|_{\partial\Omega \cup M_2} \text{ and } r|_{M_1} = s|_{M_1}, \tag{4.4}$$

where $\{.\}_x$ and $\{.\}_y$ denote the partial derivative of $\{.\}$ along x and y coordinates respectively, while $\alpha(.)$ and $\beta(.)$ are two weights controlling the similarity between the gradient of r and s. The hard constraints are the same as discussed before. The weights, α, β, are defined according to above analysis, *i.e.*

$$\alpha(p) = \frac{1}{|s_x(p)|^\gamma + \epsilon} \quad , \quad \beta(p) = \frac{1}{|s_y(p)|^\gamma + \epsilon} \quad , \tag{4.5}$$

where ϵ is a small constant preventing division by zero, and the exponent γ is the sensitivity of enforcing the similarity. A similar way of defining the two weights can be found in [Lischinski et al. 2006] and [Farbman et al. 2008].

The solution of Equation (4.4) is given by the following linear system,

$$\text{for each } p \in \Omega, \\ \sum_{q \in \mathcal{N}(p)} \frac{r(p) - r(q)}{|s(p) - s(q)|^\gamma + \epsilon} = \sum_{q \in \mathcal{N}(p)} \frac{s(p) - s(q)}{|s(p) - s(q)|^\gamma + \epsilon} \quad , \tag{4.6}$$

with hard constraints,

$$\begin{cases} r(p) = t(p) & p \in \partial\Omega \cup M_2 \\ r(p) = s(p) & p \in M_1 \end{cases} \quad , \tag{4.7}$$

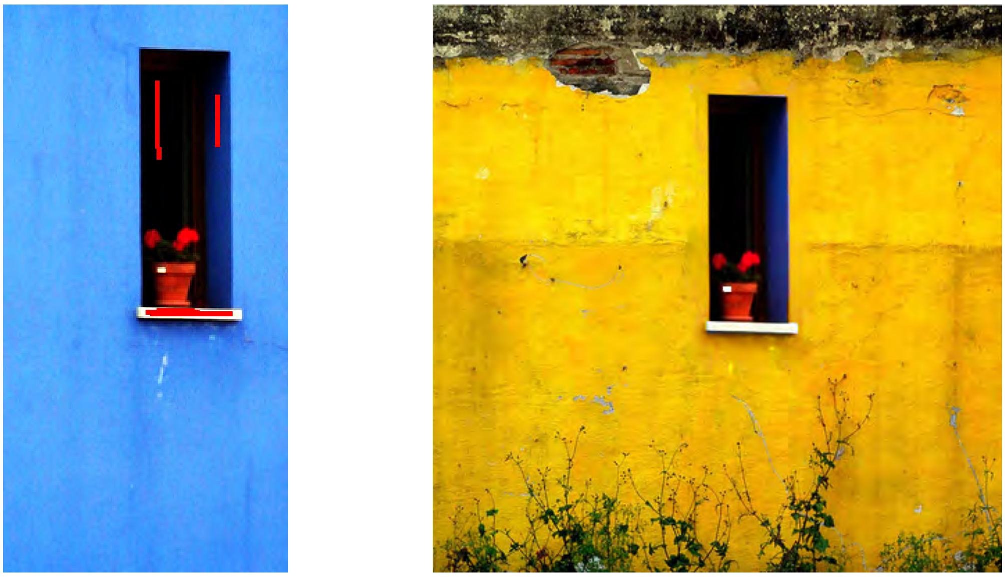
where $\mathcal{N}(p)$ denotes four neighbors of p. A detail derivation is given in Appendix B.3. By substituting the hard constraints (4.7) into (4.6), the sparse linear system can be solved efficiently. The unknowns are the region to be pasted. Thus the running time is only related to the size of the region. In our experiments, for a 500×500 region, the running time is 1-2 seconds with matlab code on a Core 2 Duo 2.33G computer.

In the 1D example previously discussed, we still use only the middle point (in fact, any point on the object works) as user markup. As shown in Figure 4.3(b), the resultant curve r (the black line) is obtained via Equation (4.6) by setting $\gamma = 4.0$. The middle part of r is kept almost the same to s, while two ends of r are seamlessly connected to t. As expected, the value of the region marked by user is preserved and no halo effects exist. A 2D example has been shown in Figure 4.1(c). The final composite was seamless and natural, while the color was well preserved.

4.3 Experiments and Results

Figure 4.5 shows an example of image compositing by blending a bear from the sea to a swimming pool. In the result of Poisson image editing, the pasted bear in the final composite appeared pale. By adding additional constraints of user markup on the body of the bear, the result of PIE produced a halo effect around the markup. In contrast, the final composite of our proposed approach preserved the look and feel of the bear in the source image. Also, the boundary of the pasted bear was seamless.

Another example is shown in Figure 4.6. The motorcyclist consists of a lot of different colors. Thus, a bit more user markup are required to cover each



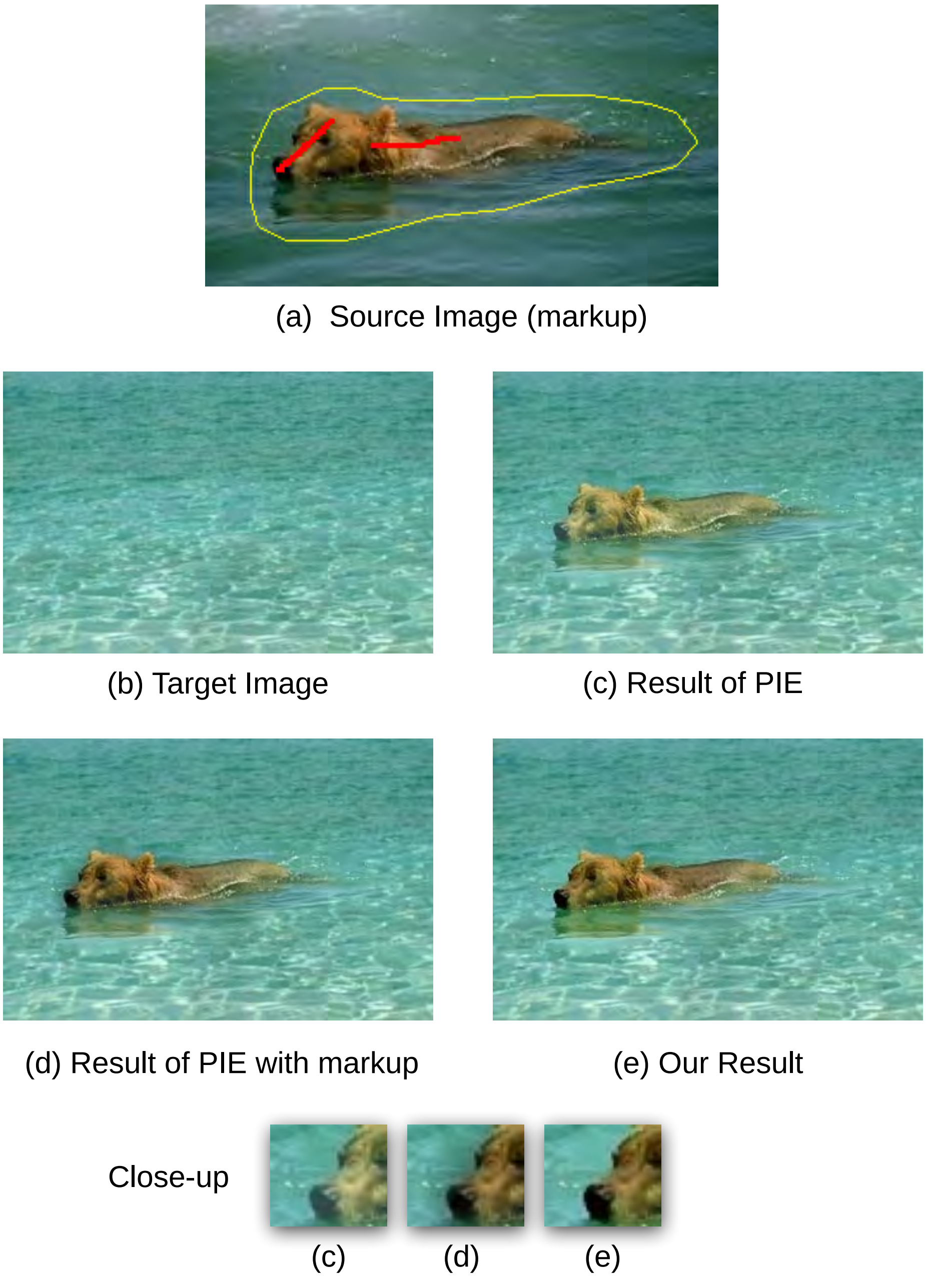
(a) New Markup

(b) Composite Result With New Markup

Figure 4.4: A new image composite result created with a different user markup. (a) User markup. One more stroke on the side of the window frame, compared to Figure 4.1(d). (b) Image composite with user markup of (a). Note that the color of the inside of the window frame has been changed to blue from yellow(Figure 4.1(c)).

component. As shown in Figure 4.6(a), nine strokes were marked on different parts, including one indicating the background where a hole exists between the motorcyclist's arms and the motorcycle. In the result of PIE, the motorcyclist became reddish. PIE with markup still produced obvious halo effect. The result of our proposed approach faithfully preserved the color of the motorcyclist.

With our proposed approach, a user can create different image composites by choosing different regions for color-preserving. We show this with the wall example in Figure 4.1. The source and target image are the same as before. The new user markup (Figure 4.4(a)) contained one more stroke on the side of the window frame. With the new markup, the side of the window frame was kept the same as the source image, with the color being blue, instead of the previous



(a) Source Image (markup)

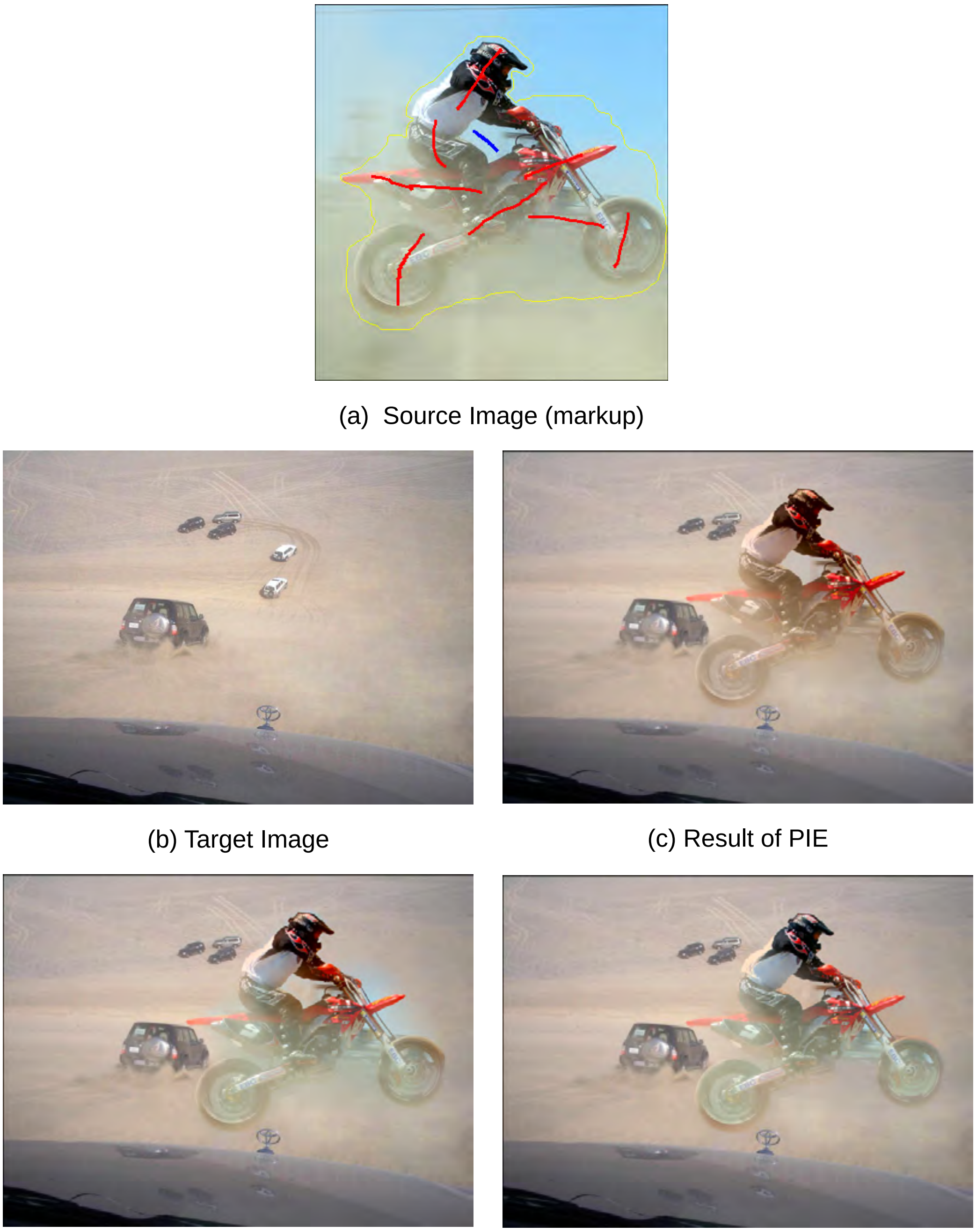
(b) Target Image

(c) Result of PIE

(d) Result of PIE with markup

(e) Our Result

Figure 4.5: Image compositing of a bear. (a) The source image with selected ROI (yellow boundary) and user markup (red strokes). (b) The target image. (c) Result of Poisson image editing. Note the color shift of the bear from original brown to pale. (d) Result of PIE with constraints by user markup. Note the halo effects around the head of the bear. (e) The result of our proposed approach without color shift or halo effect. The images are from [Pérez et al. 2003].



(a) Source Image (markup)

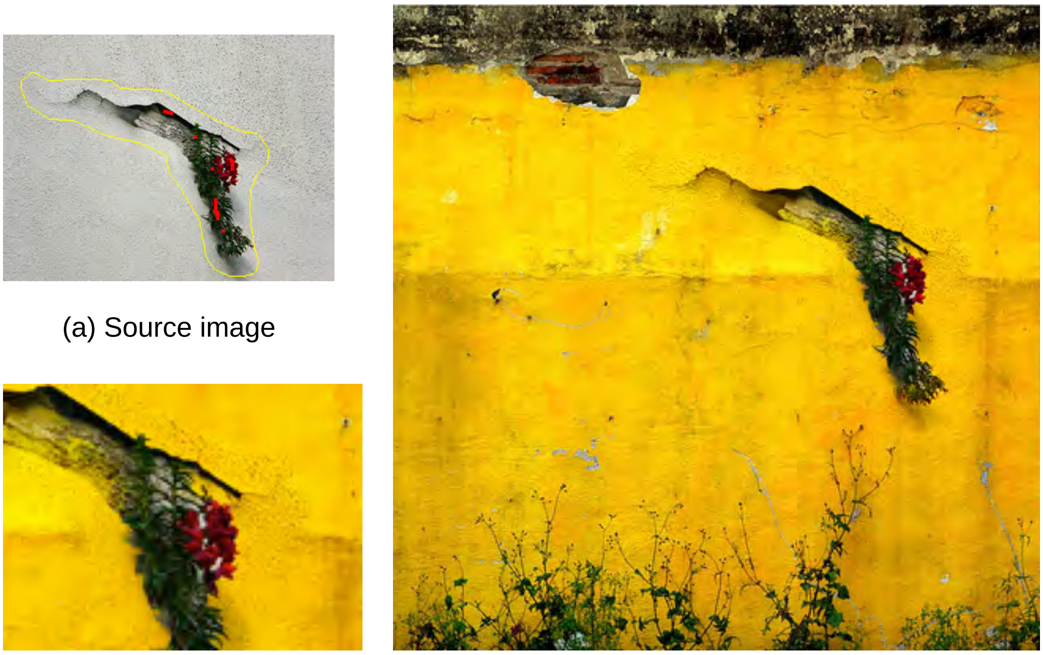
(b) Target Image

(c) Result of PIE

(d) Result of PIE with markup

(e) Our Result

Figure 4.6: Image compositing of a motorcyclist. (a) The source image with ROI and user markup of the foreground (red) and background (blue). (b) The target image. (c) Result of PIE. Note the color shift on the helmet. (d) Result of PIE with constraints by user markup. Note the halo effect around the foreground. (e) Result of our proposed approach.



(a) Source image

(c) Close-up of result

(b) Composite result

Figure 4.7: Limitation. Flower example. The target image is the same as in Figure 4.1(b). There are a lot of "holes" in the branches of the flowers (gray color). In the result, the color of "holes" appears the same as that in source image.

"yellow". Still, there is not any artifact produced in the final composite.

4.4 Summary and Discussion

In this chapter, we have presented an approach of creating seamless image composite with the color of specified region faithfully preserved. The main idea is simple yet powerful. By adding different weights to the minimization term, smaller gradient is more likely to be kept small. Thus the color within a region is kept similar in the final composite. By adding user markup, the color in specified regions is preserved in the final composite. With different markups, users have the flexibility in creating different image composites by choosing different regions for color-preserving. Experiment results have demonstrated the effectiveness of our approach.

If the object of interest is simple, *i.e.* its number of components is small, the user only need to draw one or two scribbles to make sure the color preserved of the object, such as in the wall (Figure 4.1) and bear (Figure 4.5) examples. If the object is complex, such as the motorcyclist (Figure 4.6), the user need to draw a lot of scribbles (red) on different parts of the object and scribbles (blue) for background as well. This might be tedious for some cases. An example is shown in Figure 4.7. There are a lot of small"holes" in the branches of the flowers. It is a tedious job for the user to mark all the "holes" with blue scribbles.

Chapter 5

Summary and Discussion

This thesis has presented three methods focusing on different purposes of photo editing, namely, over-exposure correction, face makeup by example, and seamless image compositing. Each of them is a self-contained work with a particular purpose of photo editing.

This chapter summarizes this thesis by giving a summary of key points for each of the three works introduced in previous chapters. Each of them has also provided a summary and discussion. The reader may also refer to them. Thereafter, future research directions are proposed at the end of this thesis.

5.1 Summary

Photo editing is a vast area with lots of existing or ongoing works. This thesis has picked three very common and useful problems. Two of them are very new, with very few existing works. Over-exposure correction has two similar existing works [Masood et al. 2009] [Zhang and Brainard 2004]; surprisingly, neither of them have solved the problem of full over-exposure (all three channels over-exposed, see Chapter 2 for details). Chapter 2 has introduced a method that could largely correct full over-exposure as well as partial over-exposure. Face makeup by example is an interesting work that provides an automatic tool to make over a face on a photo with another photo as makeup style example. Only one existing similar work [Tong et al. 2007] has addressed face makeup. Their method requires two photos, one "before", one "after" as the makeup example. This might not be applicable in practice. Another work, seamless image compositing, is based on an existing work, Poisson image editing (PIE) [Pérez et al. 2003].

In the method of over-exposure correction (Chapter 2), the gradient of well-exposed regions are slightly compressed so that the gradient of over-exposed regions could expend. De-saturated color is corrected by neighboring propagation and color confidence. With these two aspects, the results appear well-exposed and color looks natural. However, limitation exists; color propagation may render incorrect color in result when over-exposure is so large that the boundaries of objects are missing.

Face makeup by example (Chapter 3) is a method of creating face makeup on a face image with another image as the style example. The overall workflow is inspired by physical makeup; color and skin details of the subject image are modified while the face structure is preserved. Both images are decomposed into three layers: face structure layer, skin detail layer, and color layer. Information from each layer of the example image is transferred to corresponding layer in the subject image. One major advantage of this method is that only one example image is required. This renders face makeup very convenient and practical. Various makeup effects besides beauty makeup, such as makeup by portraiture, aging effects, beard transfer are easily accessible by our method with slightly different parameter settings.

Seamless image compositing (Chapter 4) provided a solution of creating image composite by seamlessly blending a region of interest from an image onto another while faithfully preserving the color of regions specified by user markup. The main idea is simple yet powerful, *i.e.* adding different weights to minimization terms so that smaller gradient is more likely to be kept small. Color of the same region is kept similar in the final composite. By adding user markup, the color in marked regions is preserved in the final composite. With different regions marked for color-preserving, this method provides users with the flexibility in creating different composites.

5.2 Future Research Directions

Although very promising results have been shown in previous chapters, these methods are far from perfect. Assumptions and/or limitations exist. Thus, one type of possible future research directions are to extend these works.

Over-exposure correction has a limitation that color may leak outside one object if the boundaries are missing due to serious over-exposure. One future direction is to solve this problem. Possible solution may be adding penalty if color propagation across a possible boundary. Also, additional user interaction may be involved to solve this problem.

Blooming effects usually come with over-exposure. Both the lightness and color of regions around the over-exposed region are affected. Our current method can correct somehow the color of the blooming effects, yet nothing for lightness.

One possible future work would be fixing the blooming effects in over-exposure correction.

In the method of face makeup by example, the active shape model we adopted assumes that the face is frontal and upright. Our system is currently tested only with the subject and example images being nearly frontal. But we envision that it can work well on any pose as long as the pose difference is not large between the subject and example images. One future work is to extend this method to across any pose.

Only one example image is used in face makeup currently. It would be interesting and practical to extend this method to multiple example images. The makeup could be combined from different examples. In this case, the consistency between different makeup styles should be considered.

Since only skin detail and color are required from the example image, if we warped them onto a canonical face, the original face structure of the example image is not required any more. Doing so is helpful in preserving the privacy of the makeup actors. Another possible future work is to collect more makeup examples and build up a makeup engine. The user could freely choose makeup styles from such an engine. Also, machine learning techniques may be applied to analyze the relation between face structure and makeup styles. Such knowledge is extremely helpful in suggesting makeup style for females.

In the work of seamless image compositing, since the user should add background markup to the holes of the object in ROI, the marking process could be time-consuming if the object consists of many holes. One future work could be automatically detecting the holes and then keeping these hole regions similar to the background in the final composite. One possible solution may be adding global color constraints to keep similar color being similar in the final composite.

Besides extension of the works presented in this thesis, rethinking and reformulating of these problem may also lead to future research directions. For example, if the details of over-exposed regions are completely lost, it is impossible to recover them in current proposed method. With the help of annotated database, the details could be borrowed from similar objects. Another example, our seamless image compositing method requires user's markup. It might also be possible if the problem is reformulated to automatically preserving the color of the objects and meanwhile blending seamlessly.

Photo editing is an interesting research area. It is full of fun with photographs. Just like photography accompanying people, research of photo editing will never stop.

Appendix A

The Euler-Lagrange Equation

The Euler-Lagrange equation was used in this thesis many times to solve different minimization problems. In this appendix, we briefly introduce the Euler-Lagrange equation. We first introduce the one dimensional Euler-Lagrange equation and then the 2 dimensional one. The 2D version could be easily adopted to solve the minimization problems defined in previous chapters, which will be discussed in Appendix B.

A.1 One Dimensional Euler-Lagrange Equation

Suppose a function $\mathcal{E}$ is defined as,

$$\mathcal{E}(u) = \int_{x_1}^{x_2} F(x, u, u')dx, \tag{A.1}$$

where $u = u(x)$ is a function of x and F is a function of x, u, and u'. According to the Variational Principle, function $u(x)$ that minimizes $\mathcal{E}$ must satisfy the Euler-

Lagrange equation

$$\frac{\partial F}{\partial u} - \frac{\mathrm{d}}{\mathrm{d}x}\frac{\partial F}{\partial u'} = 0, \tag{A.2}$$

which is a partial differential equation of u.

A.2 Two Dimensional Euler-Lagrange Equation

Similar as the 1D case, u is defined as

$$u = u(x, y). \tag{A.3}$$

The energy function $\mathcal{E}$ is defined as

$$\mathcal{E}(u) = \iint F\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) dx, \tag{A.4}$$

where F is a function of x, y, u, $\partial u/\partial x$, and $\partial u/\partial y$.

According to the Variational Principle [Fomin and Silverman 2000], function $u(x, y)$ that minimizes $\mathcal{E}$ must satisfy the following Euler-Lagrange equation

$$\frac{\partial F}{\partial u} - \frac{\partial}{\partial x}\frac{\partial F}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial F}{\partial_y} = 0, \tag{A.5}$$

where

$$u_x = \frac{\partial u}{\partial x}, \quad u_y = \frac{\partial u}{\partial y}, \tag{A.6}$$

The proof of Equation (A.2) and Equation (A.2) is not discussed here. The readers may refer to some references for the proof, such as [Fomin and Silverman 2000] and [Wikipedia 2010].

Appendix B

Solution to Minimization Problems

This appendix discusses how to solve the minimization problems defined in previous chapters. First, we discuss the solution to the minimization problem of lightness recovery in over-exposure correction. Then, the solution to the layer decomposition problem in face makeup by example is discussed. Last, the solution to the problem of seamless image compositing is presented.

B.1 Over-Exposure Correction

The corrected lightness $\tilde{L}$ could be solved by minimizing the energy $\mathcal{E}_L$ defined in Chapter 2. Now we give a detailed derivation of the solution.

The energy $\mathcal{E}_1$ can be rewritten in continuous domain as

$$\mathcal{E}_1 = \iint_{\Omega} \left| \nabla \tilde{L} - \mathbf{z}(\nabla L) \right|^2 \mathrm{d}x\mathrm{d}y, \tag{B.1}$$

where Ω is the domain of $\tilde{L}(x, y)$.

Similarly, $\mathcal{E}_2$ is rewritten as

$$\mathcal{E}_2 = \frac{1}{|\Omega|} \iint\limits_{\Omega} \mathcal{P} \left| \tilde{L} - L \right|^2 \mathrm{d}x\mathrm{d}y. \tag{B.2}$$

Thus, $\mathcal{E}_L$ becomes

$$\begin{aligned} \mathcal{E}_L &= \mathcal{E}_1 + \lambda \mathcal{E}_2 \\ &= \iint\limits_{\Omega} \left| \nabla \tilde{L} - \mathbf{z}(\nabla L) \right|^2 \mathrm{d}x\mathrm{d}y + \frac{\lambda}{|\Omega|} \iint\limits_{\Omega} \mathcal{P} \left| \tilde{L} - L \right|^2 \mathrm{d}x\mathrm{d}y \\ &= \iint\limits_{\Omega} \left(\left| \nabla \tilde{L} - \mathbf{z}(\nabla L) \right|^2 + \frac{\lambda \mathcal{P}}{|\Omega|} \left| \tilde{L} - L \right|^2 \right) \mathrm{d}x\mathrm{d}y \end{aligned} \tag{B.3}$$

We use F denote the middle part of Equation (B.3), *i.e.*

$$F(x, y, \tilde{L}, \frac{\partial \tilde{L}}{\partial x}, \frac{\partial \tilde{L}}{\partial y}) = \left| \nabla \tilde{L} - \mathbf{z}(\nabla L) \right|^2 + \frac{\lambda \mathcal{P}}{|\Omega|} \left| \tilde{L} - L \right|^2 . \tag{B.4}$$

Please note $\mathbf{z}(\nabla L)$, $\frac{\lambda \mathcal{P}}{|\Omega|}$, and L are all constants. We use

$$\mathbf{z} = \begin{pmatrix} \mathbf{z}_x \\ \mathbf{z}_y \end{pmatrix} \tag{B.5}$$

to represent $\mathbf{z}(\nabla L)$.

Function $\tilde{L}$ that minimizes $\mathcal{E}_L$ must satisfy the corresponding Euler-Lagrange equation,

$$\frac{\partial F}{\partial \tilde{L}} - \frac{\partial}{\partial x} \frac{\partial F}{\partial \tilde{L}_x} - \frac{\partial}{\partial y} \frac{\partial F}{\partial \tilde{L}_y} = 0, \tag{B.6}$$

where

$$\tilde{L}_x = \frac{\partial \tilde{L}}{\partial x}, \quad \tilde{L}_y = \frac{\partial \tilde{L}}{\partial y}. \tag{B.7}$$

Equation (B.6) can be simplified as

$$\begin{aligned}&\frac{\partial F}{\partial \tilde{L}} - \frac{\partial}{\partial x}\frac{\partial F}{\partial \tilde{L}_x} - \frac{\partial}{\partial y}\frac{\partial F}{\partial \tilde{L}_y}\\ =&2 \cdot \frac{\lambda \mathcal{P}}{|\Omega|}\left(\tilde{L} - L\right) - 2 \cdot \left(\frac{\partial^2 \tilde{L}}{\partial x^2} - \frac{\partial \mathbf{z}_x}{\partial x}\right) - 2 \cdot \left(\frac{\partial^2 \tilde{L}}{\partial y^2} - \frac{\partial \mathbf{z}_y}{\partial y}\right)\\ =&2 \cdot \left(\frac{\lambda \mathcal{P}}{|\Omega|}\left(\tilde{L} - L\right) - \Delta\tilde{L} + \mathrm{div}\mathbf{z}\right)\\ =&0,\end{aligned} \tag{B.8}$$

thus,

$$\frac{\lambda \mathcal{P}}{|\Omega|}\left(\tilde{L} - L\right) - \Delta\tilde{L} + \mathrm{div}\mathbf{z} = 0, \tag{B.9}$$

where Δ denote Laplacian operator, *i.e*.

$$\Delta\{.\} = \frac{\partial^2\{.\}}{\partial x^2} + \frac{\partial^2\{.\}}{\partial y^2} \tag{B.10}$$

and div**z** is the divergence of **z**, *i.e*.

$$\mathrm{div}\mathbf{z} = \frac{\partial \mathbf{z}_x}{\partial x} + \frac{\partial \mathbf{z}_y}{\partial y}. \tag{B.11}$$

Reorganizing Equation (B.9), we have

$$\frac{\lambda \mathcal{P}}{|\Omega|} \cdot \tilde{L} - \Delta\tilde{L} = \frac{\lambda \mathcal{P}}{|\Omega|} \cdot L - \mathrm{div}\mathbf{z}. \tag{B.12}$$

In Equation (B.12), the unknowns are $\tilde{L}(x, y)$. In discrete image domain, both the Laplacian Δ and div are linear operators. Thus, Equation (B.12) is a linear system. In this system, there is one equation for each pixel in Ω. And there are five unknowns in one equation. Thus this is a banded sparse linear system, which could be solved efficiently.

B.2 Layer Decomposition in Face Makeup

To decompose face structure layer s from the lightness channel of a face image, we defined an energy function E in Equation (3.1),

$$E(s) = \iint \Big(|s - l|^2 + \lambda H(\nabla s, \nabla l) \Big) \mathrm{d}x\mathrm{d}y. \tag{B.13}$$

We use F represent the middle part of above equation *i.e.*

$$F = |s - l|^2 + \lambda H(\nabla s, \nabla l). \tag{B.14}$$

E attains local minimum when s satisfies the following Euler-Lagrange equation

$$\frac{\partial F}{\partial s} - \frac{\partial}{\partial x}\frac{\partial F}{\partial s_x} - \frac{\partial}{\partial y}\frac{\partial F}{\partial s_y} = 0, \tag{B.15}$$

where

$$s_x = \frac{\partial s}{\partial x} \quad \text{and} \quad s_y = \frac{\partial s}{\partial y} \tag{B.16}$$

Simplifying Equation (B.15),

$$2 \cdot (s - l) - 2\lambda \cdot \frac{\partial}{\partial x}\left(\frac{\beta s_x}{|l_x|^\alpha + \epsilon}\right) - 2\lambda \cdot \frac{\partial}{\partial y}\left(\frac{\beta s_y}{|l_y|^\alpha + \epsilon}\right) = 0. \tag{B.17}$$

After reorganizing, we may get

$$s - \lambda\left(\frac{\partial}{\partial x}\left(\frac{\beta s_x}{|l_x|^\alpha + \epsilon}\right) + \frac{\partial}{\partial y}\left(\frac{\beta s_y}{|l_y|^\alpha + \epsilon}\right)\right) = l. \tag{B.18}$$

Thus, we have the solution of the minimization problem. The pixels of s are the unknowns. In discrete domain, using finite difference to approximate the derivatives, Equation (B.18) could be simplified as

$$\begin{aligned}&\text{For each } p,\\ &s(p) + \sum_{q\in\mathcal{N}(p)} \frac{\beta(p)(s(p)-s(q))}{|l(p)-l(q)|^{\alpha}+\epsilon} = l(p),\end{aligned} \tag{B.19}$$

where $\mathcal{N}(p)$ denotes the four neighbors of p. This is a linear system. For each p, there is a corresponding equation with only five unknowns. Similar to the solution to lightness recovery in over-exposure correction (Equation (B.12)), (B.19) is also a sparse banded linear system.

B.3 Image Compositing Problem

In Chapter 4, the compositing problem was formulated as a minimization of energy in Equation (4.4)

$$\mathcal{E} = \sum_{p\in\Omega} \Big(\alpha(p)\big(r_x(p) - s_x(p)\big)^2 + \beta(p)\big(r_y(p) - s_y(p)\big)^2\Big), \tag{B.20}$$

with a boundary condition. Define F as

$$F = \alpha(p)\big(r_x(p) - s_x(p)\big)^2 + \beta(p)\big(r_y(p) - s_y(p)\big)^2. \tag{B.21}$$

Function r that minimizes $\mathcal{E}$ must satisfy the Eular-Largrange equation,

$$\frac{\partial F}{\partial r} - \frac{\partial}{\partial x}\frac{\partial F}{\partial r_x} - \frac{\partial}{\partial y}\frac{\partial F}{\partial r_y} = 0, \tag{B.22}$$

which is,

$$0 - \frac{\partial}{\partial x}\left(2 \cdot \alpha(p)(r_x(p) - s_x(p))\right) - \frac{\partial}{\partial y}\left(2 \cdot \beta(p)(r_y(p) - s_y)\right) = 0 \tag{B.23}$$

After reorganizing, we have,

$$\frac{\partial}{\partial x}(\alpha(p)r_x(p)) + \frac{\partial}{\partial y}(\beta(p)r_y(p)) = \frac{\partial}{\partial x}(\alpha(p)s_x(p)) + \frac{\partial}{\partial y}(\beta(p)s_y(p)). \tag{B.24}$$

Each element of above equation is a partial derivative of weighted derivative. $\alpha(p)$ and $\beta(p)$ were defined according to s_x and s_y respectively. In image discrete domain, we use finite difference to approximate the derivatives. (B.24) can be easily simplified,

$$\sum_{q\in\mathcal{N}(p)} \frac{r(p) - r(q)}{|s(p) - s(q)|^{\gamma} + \epsilon} = \sum_{q\in\mathcal{N}(p)} \frac{s(p) - s(q)}{|s(p) - s(q)|^{\gamma} + \epsilon}. \tag{B.25}$$

This is again a sparse banded linear system with $r(p) : p \in \Omega$ being the unknowns.

Bibliography

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., AND AMD MICHAEL COHEN, D. S. 2004. Interactive digital photomontage. *ACM Transactions on Graphics*.

BOOKSTEIN, F. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 6, 567–585.

BRAND, M., AND PLETSCHER, P. 2008. A conditional random field for automatic photo editing. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

CHEN, H.-T., LIU, T.-L., AND CHANG, T.-L. 2005. Tone reproduction: A perspective from luminance-driven perceptual grouping. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

CHUANG, Y.-Y., CURLESS, B., SALESIN, D. H., AND SZELISKI, R. 2001. A bayesian approach to digital matting. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

DEBEVEC, P. E., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *Proceedings of ACM SIGGRAPH*, 369–378.

Eisemann, E., and Durand, F. 2004. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics 23*, 3, 673–678.

Farbman, Z., Fattal, R., Lischinski, D., and Szeliski, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics 27*, 3, 1–10.

Fattal, R., Lischinski, D., and Werman, M. 2002. Gradient domain high dynamic range compression. *ACM Transactions on Graphics 21*, 3, 249–256.

Fomin, S., and Silverman, R. 2000. *Calculus of variations*. Dover Books on Mathematics. Dover Publications.

Guo, D., and Sim, T. 2009. Color me right–seamless image compositing. In *Proceedings of 13th Intl. Conf. on Computer Analysis of Images and Patterns (CAIP)*.

Guo, D., and Sim, T. 2009. Digital face makeup by example. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

Guo, D., Cheng, Y., Zhuo, S., and Sim, T. 2010. Correcting over-exposure in photographs. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H. 2001. Image analogies. In *Proceedings of ACM SIGGRAPH*.

Jia, J., Sun, J., Tang, C.-K., and Shum, H.-Y. 2006. Drag-and-drop pasting. *ACM Transactions on Graphics 25*, 3, 631–637.

Lanitis, A., Taylor, C., and Cootes, T. 2002. Toward automatic simulation of aging effects on face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4, 442–455.

Levin, A., Lischinski, D., and Weiss, Y. 2004. Colorization using optimization. *ACM Transactions on Graphics 23*, 3, 689–694.

Leyvand, T., Cohen-Or, D., Dror, G., and Lischinski, D. 2008. Data-driven enhancement of facial attractiveness. *ACM Transactions on Graphics 27*, 3, 1–9.

Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. 2004. Lazy snapping. *ACM Transactions on Graphics 23*, 3, 303–308.

Lischinski, D., Farbman, Z., Uyttendaele, M., and Szeliski, R. 2006. Interactive local adjustment of tonal values. *ACM Transactions on Graphics 25*, 3, 646–653.

Liu, Z., Shan, Y., and Zhang, Z. 2001. Expressive expression mapping with ratio images. In *Proceedings of ACM SIGGRAPH*, ACM, New York, NY, USA, SIGGRAPH '01, 271–276.

Lukac, R., and Plataniotis, K. N., Eds. 2007. *Color Image Processing: Methods and Applications*. CRC Press.

Masood, S. Z., Zhu, J., and Tappen., M. F. 2009. Automatic correction of saturated regions in photographs using cross-channel correlation. In *Proc. Pacific Conference on Computer Graphics and Applications*.

Milborrow, S., and Nicolls, F. 2008. Locating facial features with an extended active shape model. In *Proceedings of European Conference on Computer Vision*.

Mitsunaga, T., and Nayar, S. K. 1999. Radiometric self calibration. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

Mortensen, E. N., and Barrett, W. A. 1995. Intelligent scissors for image composition. In *Proceedings of ACM SIGGRAPH*.

Nars, F. 2004. *Makeup your Mind*. PowerHouse Books.

Ojima, N., Yoshida, K., Osanai, O., and Akasaki, S. 1999. Image synthesis of cosmetic applied skin based on optical properties of foundation layers. In *Proceedings of International Congress of Imaging Science*, 467–468.

Pérez, P., Gangnet, M., and Blake, A. 2003. Poisson image editing. *ACM Transactions on Graphics 22*, 3, 313–318.

Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J. 2002. Photographic tone reproduction for digital images. *ACM Transactions on Graphics 21*, 3, 267–276.

Rempel, A. G., Trentacoste, M., Seetzen, H., Young, H. D., Heidrich, W., Whitehead, L., and Ward, G. 2007. Ldr2hdr: on-the-fly reverse tone mapping of legacy video and photographs. *ACM Trans. Graph. 26*, 3, 39.

Shan, Y., Liu, Z., and Zhang, Z. 2001. Image-based surface detail transfer. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

Sun, J., Jia, J., Tang, C.-K., and Shum, H.-Y. 2004. Poisson matting. *ACM Transactions on Graphics*.

Suo, J., Min, F., Zhu, S., Shan, S., and Chen, X. 2007. A multi-resolution dynamic model for face aging simulation. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

Taaz.com. http://www.taaz.com/.

Tomasi, C., and Manduchi, R. 1998. Bilateral filtering for gray and color images. In *Proceedings of IEEE International Conference on Computer Vision*.

TONG, W.-S., TANG, C.-K., BROWN, M. S., AND XU, Y.-Q. 2007. Example-based cosmetic transfer. In *Proc. Pacific Conference on Computer Graphics and Applications*.

TSUMURA, N., OJIMA, N., SATO, K., SHIRAISHI, M., SHIMIZU, H., NABESHIMA, H., AKAZAKI, S., HORI, K., AND MIYAKE, Y. 2003. Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin. *ACM Transactions on Graphics 22*, 3, 770–779.

WANG, L., WEI, L.-Y., ZHOU, K., GUO, B., AND SHUM, H.-Y. 2007. High dynamic range image hallucination. In *Eurographics Symposium on Rendering*.

WIKIPEDIA, 2010. Euler–lagrange equation (http://en.wikipedia.org/w/index .php?title=euler [Online; accessed 13-October-2010].

WOODLAND, A., AND LABROSSE, F. 2005. On the separation of luminance from colour in images. In *International Conference on Vision, Video and Graphics*, The Eurographics Association, Edinburgh, UK.

YAMAGUCHI, B. 2004. *Billy Yamaguchi Feng Shui Beauty*. Sourcebooks, Inc.

ZHANG, X., AND BRAINARD, D. H. 2004. Estimation of saturated pixel values in digital color imaging. *Journal of the Optical Society of America A 21*, 12, 2301–2310.

ZHANG, X., SIM, T., AND MIAO, X. 2008. Enhancing photographs with near infrared images. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.