EIE4512 Final Project 2023-High/Low Frequency Skin Processing

Anonymous Submission
Paper ID: 6

Abstract

In commercial photography the requirement of skin processing is strict, because it not only requires skin smoothness, but also requires skin texture, which makes it hard to process the skin by a simple approach. Existing skin processing algorithm utilizes low-pass filter to conduct skin smoothing, and existing skin processing software applies the algorithm that behaves like a low-pass filter to conduct skin smoothing. However, in both circumstances, the skin loses its texture. Therefore, applying merely the low-pass filter to the skin fails to accomplish the requirement. My method utilizes High/Low Frequency Skin Processing to process both the low frequency component and the high frequency component. The low frequency component is smoothed while the high frequency component is enhanced. The processed low and high frequency components are combined together as the final result. The experiment result shows that the proposed method is capable of accomplishing the skin processing requirement for commercial photography.

1 Introduction

Nowadays the skin processing software is widely used to beautify the skin appearance of portrait images. The skin processing software provides users with an efficient and user-friendly way to beautify the image for social media sharing.

In commercial photography, ordinary skin beautification does not satisfy the requirement, where both skin smoothness and skin texture are expected. Although the processed image of the software appears smoother, the skin texture is usually lost. It is because the algorithm behaves like a low-pass filter in the frequency domain, eliminating the high frequency component that represents the skin texture in the spatial domain. Therefore, the image processed by the rudimental skin processing function of the software is not satisfactory. Accordingly, a more advanced algorithm is required for commercial photography.

In this paper, I propose the High/Low Frequency Skin Processing algorithm. The algorithm consists of four key steps. Firstly, the original image is transformed into the frequency domain and the low frequency component and high frequency component are separated. Secondly the low frequency component is smoothed using the low-pass filter in the frequency domain. Thirdly the high frequency component is enhanced using the high-pass filter in the frequency domain. Finally, the processed low frequency component and the processed high frequency component are transformed into the spatial domain and combined as the ultimate processed image.

As the face beautification software applies a sophisticated algorithm that detects and excludes the non-skin area, the result generated by the software looks more natural compared with the result generated by the low-pass filter. To improve the drawback of the face beautification software, I add the processed high frequency component of the original image to the image processed by the software. The result appears more natural than the proposed High/Low Frequency Skin Processing algorithm.

1.1 Related Work

Filtering in the frequency domain is a classic method to process the image, including transforming the image from the spatial domain to the frequency domain by the 2-D discrete Fourier transform[2, 4, 6], image smoothing using frequency domain filters[3, 4, 9], image sharpening using frequency domain filters[3, 4, 9], and transforming the image from the frequency domain to the spatial domain by the 2-D discrete inverse Fourier transform[4, 6]. However, applying a single classic method may not generate satisfactory result as the commercial photography expected. The low-pass filter in the frequency domain may generates an over smoothed result, and the high-pass filter in the frequency domain may generates an over sharpened result.

In commercial photography, professional photographers apply frequency separation to process the image. The low-frequency component represents tone and color and the high frequency component represents the texture and hair[1, 4, 5, 10]. They apply the technique of separating the image into typically two layers and process them independently[1, 5, 7, 8]. However, it is only a general idea and the algorithm is not specific and unique. The actual algorithm may vary according to different implementations.

2 The Proposed Algorithm

2.1 Frequency domain decomposition

Firstly, the input image is padded to prepare it for Fourier transform operations, which is denoted by $P = 2M, Q = 2N$, where M, N denotes the height and width of the original image, and P, Q denotes the height and width of the padded image. Secondly, for each color channel, its Fourier transform is computed, according to the following equation:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M+vy/N)} \tag{1}$$

and the zero frequency component is shifted to the center. Thirdly, a Gaussian Lowpass Filter and a Gaussian Highpass Filter are applied to each color channel's frequency domain to obtain low-frequency component and high-frequency component, according to the following equation:

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} \tag{2}$$

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2} \tag{3}$$

where is D_0 the cutoff frequency, and

$$D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2\right]^{1/2} \tag{4}$$

2.2 Low-frequency component processing

For each color channel, apply a Gaussian Lowpass Filter with Equation (2). Inverse Fourier Transform is applied to obtain the component in the spatial domain

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M+vy/N)} \tag{5}$$

2.3 High-frequency component processing

2.3.1 Gaussian Highpass Filters

For each color channel, apply a Gaussian Highpass Filter with Equation (3). Inverse Fourier Transform is applied to obtain the component in the spatial domain with Equation (5).

2.3.2 The Laplacian in the Frequency Domain

For each color channel, apply the Laplacian

$$H(u,v) = -4\pi^2 \left[(u - P/2)^2 + (v - Q/2)^2\right] \tag{6}$$

$$= -4\pi^2 D^2(u,v) \tag{7}$$

Then, the Laplacian image is obtained as:

$$\nabla^2 f(x,y) = \mathfrak{F}^{-1}\left\{H(u,v)F(u,v)\right\} \tag{8}$$

Enhancement is achieved using the equation:

$$g(x,y) = f(x,y) + c\nabla^2 f(x,y) \tag{9}$$

2.3.3 Intersection of the Gaussian Highpass Filter and the Laplacian

The intersection of the processed high-frequency component in the spatial domain, denoted by $g_1(x,y)$, and the Laplacian in the spatial domain, denoted by $g_2(x,y)$, is computed using the equation:

$$g(x,y) = \min\{g_1(x,y), g_2(x,y)\} \tag{10}$$

2.4 Composition of processed results

Using the equation:

$$g(x,y) = g_1(x,y) + g_2(x,y) \tag{11}$$

the processed low-frequency component, denoted by $g_1(x,y)$, and the processed high-frequency component, denoted by $g_2(x,y)$, are merged into one image, denoted by $g(x,y)$. The final image is denoted by result1.

The processed low-frequency component, denoted by $g_1(x,y)$, and the intersection of the processed high-frequency component and the Laplacian, denoted by $g_2(x,y)$, are merged into one image, denoted by $g(x,y)$. The final image is denoted by result2.

The image processed by Meitu, denoted by $g_1(x,y)$, and the processed high-frequency component, denoted by $g_2(x,y)$, are merged into one image, denoted by $g(x,y)$. The final image is denoted by result3.

3 Experiments

3.1 Evaluation Dataset

The evaluation dataset contains portrait images of different skin. The images are collected on the Internet from https://www.freepik.com, and they are the portrait photography close-up images. Shot by the professional camera, the original images have clear skin texture. For run-time efficiency, all the images are resized to a smaller resolution with width 800 pixels. The height of the resized image varies according to the height-and-width ratio. A 4:5 image has height 1000 pixels, and a 2:3 image has height 1200 pixels.

3.2 Evaluation Metric

PSNR(Peak signal-to-noise ratio) and SSIM(Structural Similarity) are computed to evaluate the processed image quality compared to the original image.

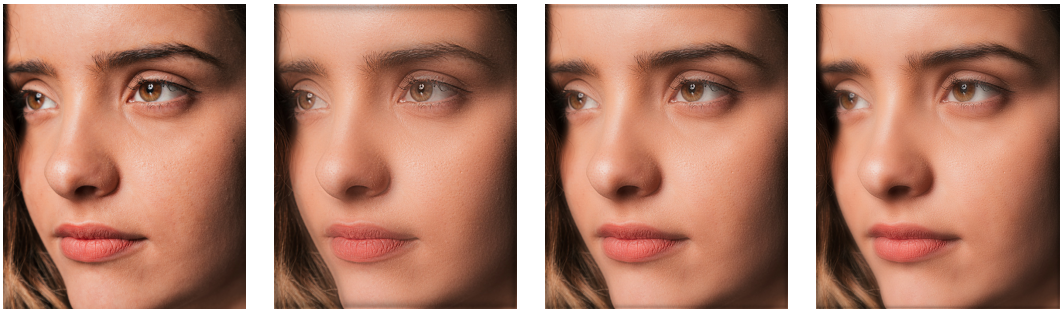
3.3 Implementation Details

To explore which value of D_0 has better effect in section 2.1, different D_0 are investigated, as is shown in Figure 1. (a) original image. (b) $D_0 = 20$. (c) $D_0 = 60$. (d) $D_0 = 100$. $D_0 = 50$ in both

section 2.2 and section 2.3. PSNR and SSIM are measured, as is shown in Table 1. It can be seen that the processed image with $D_0 = 60$ in section 2.1 perceives the best among the three and obtains the best PSNR and SSIM. To explore which value of D_0 has better effect in section 2.2 and section 2.3, different D_0 are investigated, as is shown in Figure 2. (a) original image. (b) $D_0 = 10$. (c) $D_0 = 50$. (d) $D_0 = 90$. $D_0 = 60$ in section 2.1. PSNR and SSIM are measured, as is shown in Table 2. It can be seen that the image with $D_0 = 50$ in section 2.2 and section 2.3 perceives the best and has good PSNR and SSIM. Therefore, $D_0 = 60$ in section 2.1 and $D_0 = 50$ in section 2.2 and section 2.3 are chosen to process different images.

In section 2.3.2, $c = -1$.

For result3, in section 2.1 $D_0 = 200$ because I intend to decrease the area that does not belong to skin texture. In section 2.3, $D_0 = 90$ because I intend to make the skin texture finer.

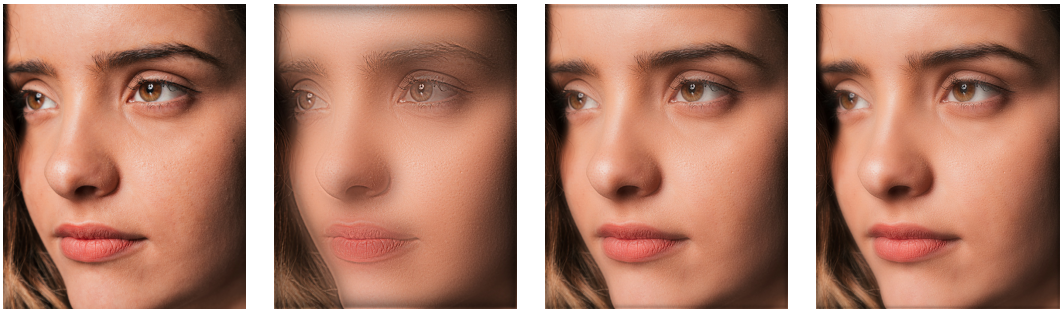


(a) (b) (c) (d)

Figure 1: Comparison between the results with different D_0 in section 2.1

Table 1: Comparison between the results with different D_0 in section 2.1

Comparison pairs			
original image	processed image	PSNR	SSIM
	$D_0 = 20$ in section 2.1, $D_0 = 50$ in both section 2.2 and section 2.3	26.8257	0.9749
	$D_0 = 60$ in section 2.1, $D_0 = 50$ in both section 2.2 and section 2.3	31.7759	0.9881
	$D_0 = 100$ in section 2.1, $D_0 = 50$ in both section 2.2 and section 2.3	31.4275	0.9872



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

Figure 2: Comparison between the results with different D_0 in section 2.2 and section 2.3

3.4 Comparison between the original image and the processed image

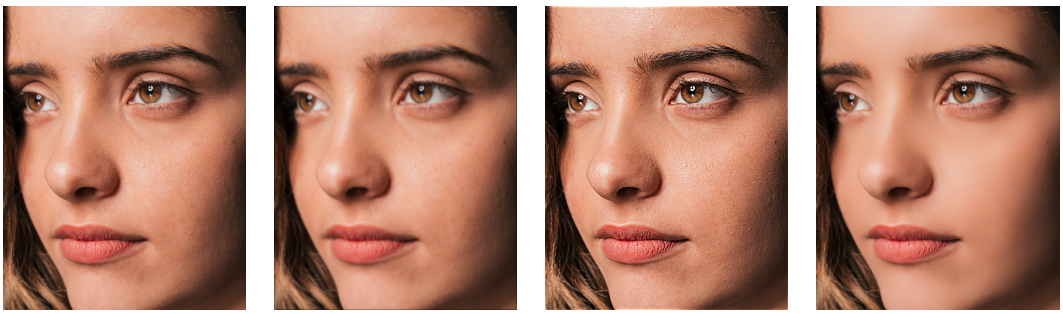
The comparison between the original image and the processed image is shown in Figure 3. (a) original image. (b) image processed with only low-pass filter. (c) image processed with only high-pass filter. (d) image directly processed by Meitu. The PSNR and SSIM between the original image and the

Table 2: Comparison between the results with different D_0 in section 2.2 and section 2.3

Comparison pairs			
original image	processed image	PSNR	SSIM
	$D_0 = 60$ in section 2.1, $D_0 = 10$ in both section 2.2 and section 2.3	21.9124	0.9474
	$D_0 = 60$ in section 2.1, $D_0 = 50$ in both section 2.2 and section 2.3	31.7759	0.9881
	$D_0 = 60$ in section 2.1, $D_0 = 90$ in both section 2.2 and section 2.3	32.5404	0.9892

processed image is shown in Table 3. Due to the limitation of pages, only the first image is displayed for this part.

The comparison between the original image and the processed image is shown in Figure 4, Figure 5, and Figure 6. (a) original image. (b) result1. (c) result2. (d) result3. The PSNR and SSIM between the original image and the processed image is shown in Table 4, Table 5, and Table 6.

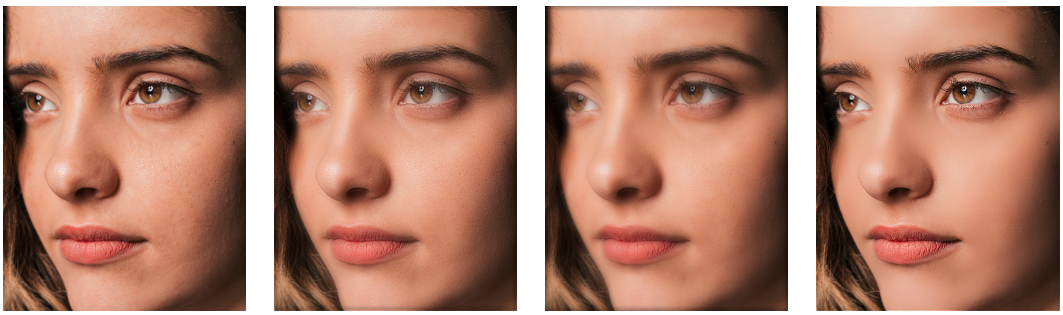


(a) (b) (c) (d)

Figure 3: Comparison between the results of the first input image

Table 3: Comparison between the results of the first input image

Comparison pairs			
original image	processed image	PSNR	SSIM
	result1	31.5404	0.9824
	result2	30.3018	0.9796
	result3	34.2340	0.9925

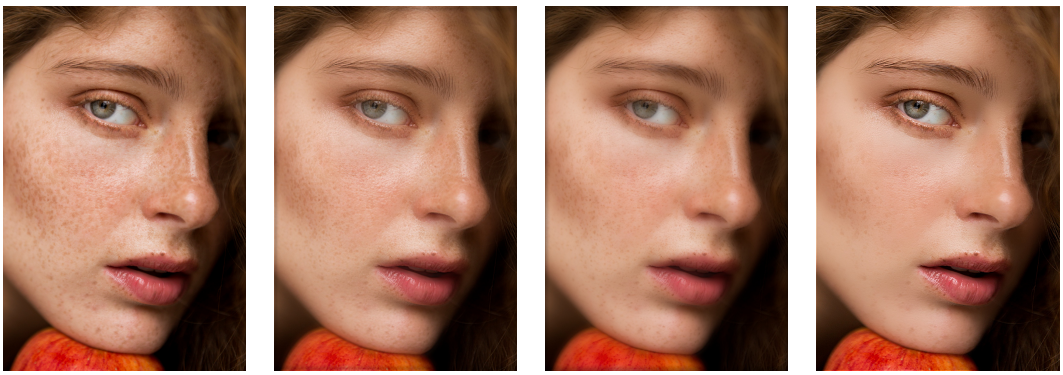


(a) (b) (c) (d)

Figure 4: Comparison between the results of the first input image

Table 4: Comparison between the results of the first input image

Comparison pairs			
original image	processed image	PSNR	SSIM
	result1	31.7759	0.9881
	result2	28.3729	0.9775
	result3	33.6097	0.9906

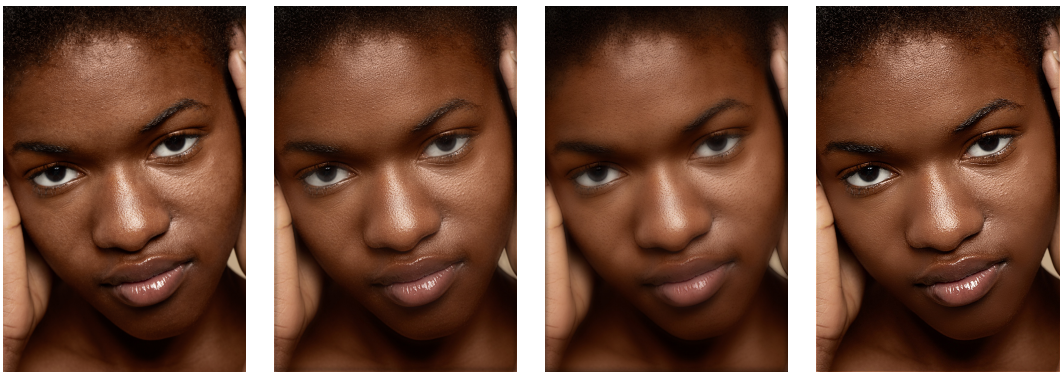


(a) (b) (c) (d)

Figure 5: Comparison between the results of the second input image

Table 5: Comparison between the results of the second input image

Comparison pairs			
original image	processed image	PSNR	SSIM
	result1	33.4588	0.9877
	result2	30.3733	0.9811
	result3	34.4460	0.9917



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

Figure 6: Comparison between the results of the third input image

4 Conclusion

The image combining the processed low-frequency component and processed high frequency component has better performance than the image combining the processed low-frequency component and the intersection.

Table 6: Comparison between the results of the third input image

Comparison pairs			
original image	processed image	PSNR	SSIM
	result1	33.1890	0.9825
	result2	28.0923	0.9582
	result3	31.9852	0.9782

The Meitu Software API makes the skin smoother, preserves the detail of the hair and the mouth, but loses the skin texture.

High/Low Frequency Skin Processing makes the skin smoother, preserves the detail of the hair and the mouth, and preserves the skin texture.

The combination of the Meitu Software API and the high frequency component generally makes the image looks more natural while preserving the detail of the hair, the mouth and the skin texture.

The skin of black people may not achieve the expected result because of the high contrast between the skin and the highlight area and their strong skin texture.

References

[1] K. Ames. *Faces: The Complete Guide to Retouching Portraits with Photoshop*. Wiley Publishing, 2012.

[2] K. De and V. Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64:149–158, 2013.

[3] M. Fritsche, S. Gu, and R. Timofte. Frequency separation for real-world super-resolution. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3599–3608. IEEE, 2019.

[4] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2008.

[5] S. Kelby. *Professional portrait retouching techniques for photographers using photoshop*. Pearson Education, 2011.

[6] B. V. Kumar, M. Savvides, K. Venkataramani, and C. Xie. Spatial frequency domain image processing for biometric recognition. In *Proceedings. International Conference on Image Processing*, volume 1, pages I–I. IEEE, 2002.

[7] C. Lee, M. T. Schramm, M. Boutin, and J. P. Allebach. An algorithm for automatic skin smoothing in digital portraits. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3149–3152. IEEE, 2009.

[8] X. Lu, X. Chang, X. Xie, J.-F. Hu, and W.-S. Zheng. Facial skin beautification via sparse representation over learned layer dictionary. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2534–2539. IEEE, 2016.

[9] V. M. Radulescu and C. A. Maican. Algorithm for image processing using a frequency separation method. In *2022 23rd International Carpathian Control Conference (ICCC)*, pages 181–185. IEEE, 2022.

[10] S. Velusamy, R. Parihar, R. Kini, and A. Rege. Fabsoften: face beautification via dynamic skin smoothing, guided feathering, and texture restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 530–531, 2020.