CS270 Digital Image Processing Project: Multi-resolution Blending

Hongyang Lin, Bingnan Li, Xinzhou Cong

I. Gaussian and Laplacian Pyramid.

Gaussian Pyramid: The Gaussian pyramid is a hierarchical image representation formed by applying a series of Gaussian filters and subsequent downsampling operations. It is constructed by recursively convolving the original image with Gaussian filters, followed by downsampling to reduce the image resolution. The Gaussian filter serves to smooth the image and remove high-frequency components, such as noise and fine details. Mathematically, the construction of the Gaussian pyramid can be described as follows:

- Given an original image I_0, the first level of the pyramid is obtained by convolving I_0 with a Gaussian kernel, resulting in I_1.
- To create the subsequent levels, the previous level image, I_n, is convolved with a Gaussian kernel to obtain I_{n+1}.
- After each convolution, the image is downsampled by discarding every alternate row and column, yielding a reduced-resolution image for the next pyramid level.

The Gaussian pyramid provides a multiscale representation of the image, where each level captures different levels of detail. The higher levels correspond to the coarser-scale information, while the lower levels preserve finer details. This pyramid structure is useful for various applications, such as image compression, image blending, and image resizing.

Laplacian Pyramid: The Laplacian pyramid complements the Gaussian pyramid by representing the details or high-frequency components that are discarded during the construction of the Gaussian pyramid. It is constructed based on the difference between the Gaussian pyramid levels and the upsampled versions of the subsequent levels. The construction of the Laplacian pyramid can be described as follows:

- Given a Gaussian pyramid with levels I_0, I_1, I_2, ..., I_n, the Laplacian pyramid is created by computing the difference between each level and the upsampled version of the next level.
- Starting from the highest level, the Laplacian image Ln is obtained by subtracting the upsampled and Gaussian-filtered version of I_{n+1} from I_n.
- This process is repeated for each level, resulting in a Laplacian pyramid L_0, L_1, L_2, ..., L_{n-1}.

Each level of the Laplacian pyramid represents the details or residuals that are necessary for reconstructing the original image. The higher levels contain coarser details, while the lower levels contain finer details. The Laplacian pyramid can be used for tasks such as image reconstruction, image enhancement, and image blending.

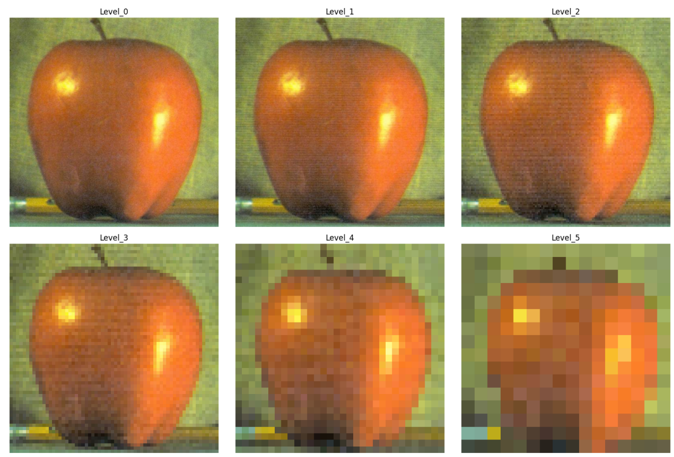

Fig. 1. The Gaussian Pyramid of image `apple.png`

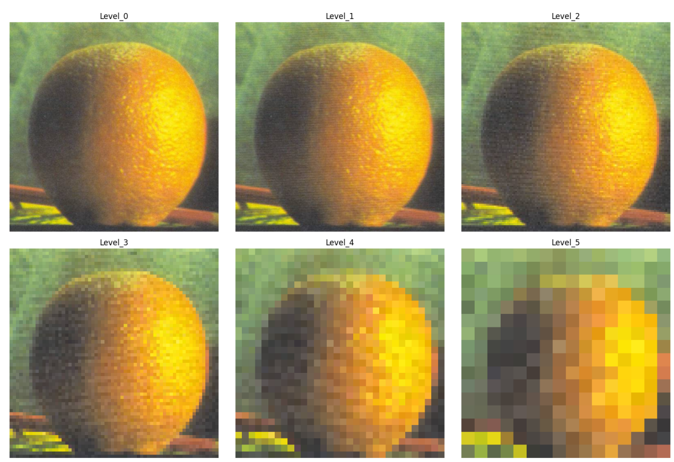

Fig. 2. The Gaussian Pyramid of image `orange.png`

Here we show the results of the Gaussian Pyramid in Fig. 1 and Fig. 2

II. Multi-Resolution Blending

The process of multi-resolution blending involves the following steps:

- Construct Gaussian and Laplacian pyramids for the source and destination images, as described in section.1.
- Blend the corresponding levels of the Laplacian pyramids: Starting from the top level of the Laplacian pyramids, blend the corresponding levels of the source and destination Laplacian pyramids together using the provided mask.

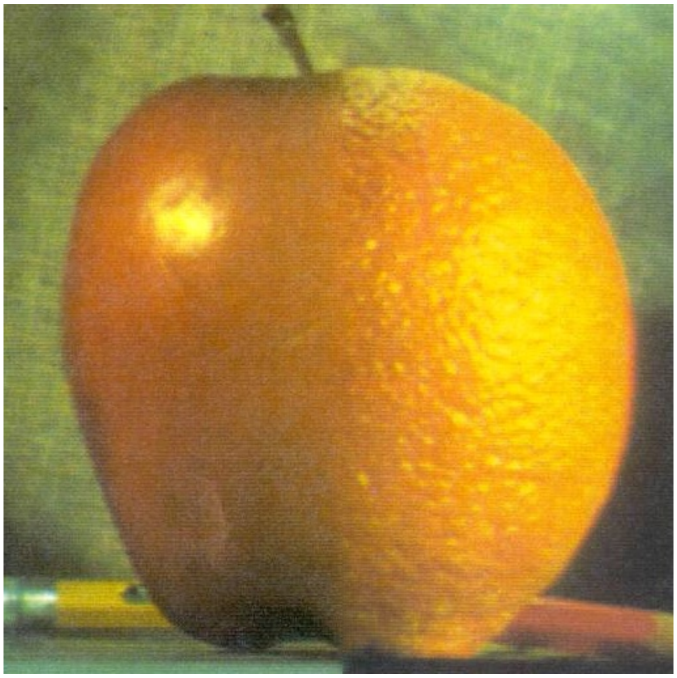

Fig. 3. The multi-resolution blending of image `orange.png` and `apple.png`.

- Reconstruct the blended image from the blended Laplacian pyramid: Starting from the top level of the blended Laplacian pyramid, upsample the blended image and add it to the upsampled version of the next level of the Laplacian pyramid. This step is repeated for each level until the original image resolution is reached.The reconstruction process combines the blended details from the Laplacian pyramid with the low-frequency components from the Gaussian pyramid to create a smooth and seamless final blended image.

Multi-resolution blending achieves a smooth transition between the source and destination images at different scales. This technique ensures that the blended image appears visually consistent and avoids noticeable artifacts or abrupt changes in pixel values.

Here we show the result of Multi-resolution blending in Fig 3.

III. Deblurring

In this section, we will demonstrate the principle we followed during deblurring and how we determine the type of blur.

Point Spread Function restoration: Point Spread Function(PSF) is a class of cylinder lens with circular aperture that has the form as follows:

$$h(x,y) = \begin{cases} \dfrac{1}{\pi R^2} & x^2+y^2 \leq R^2 \\ 0 & otherwise \end{cases} \tag{1}$$

where $h(x,y)$ is the PSF in sptial domain. By [3], the Fourier Transform(FT) of PSF $H(u,v)$ has the property that "the amplitude spectrum of $H(u,v)$ is is circularly symmetric and is characterized by periodic circles at radius r, along which Fourier magnitude takes value zero" and

$$2\pi Rr = 3.83, 7.02, 10.2, 13.2, 16.5\cdots \tag{2}$$

where $r = \sqrt{u^2+v^2}$ and R is the radius of the defocus blur radius in Eq. 1. Thus, once we determine the value of r from the FT of blurred image, we can reconstruct the PSF kernel.

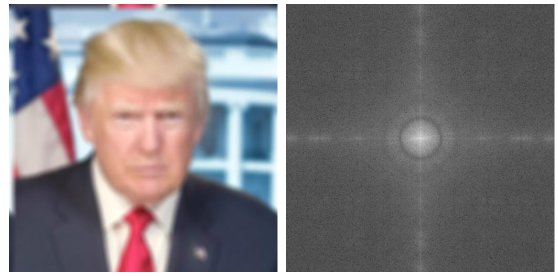

Fig. 4. The blurred image and the log-magnitude spectrum of `Trump` in frequency domain

method	PCPE/C	PCPE	manual
r	36.858	**39.358**	39.5

TABLE I
Comparison between PCPE and PCPE without compensation (PCPE/C) and manually estimated r

In our project, we aim to restore an image of Donald Trump blurred by PSF kernel whose image in spatial domain and log-magnitude spectrum in frequency domain are shown in Fig. 4. By performing Canny edge detection [1] on the log-spectrum, we can get a coarse contour of the inner-most circle shown in Fig. 5. However, we encountered two main difficulties in our experiments:

- The Canny Detector utilizes Gaussian kernel to blur the image, but Gaussian blur leads to "contour shrinkage", which means the edge detected after Gaussian blur will shrink several pixels, leading to measurement error.
- The edge detected by Canny with resolution 512×512 is in disconnected shape and not in a standard-circle way, which reduces the robustness of estimation.

To address those two problems, we proposed **Pyramid Compensate Parameter Estimation(PCPE)** method instead of using First White Method proposed by [2]. To be detail, we processed the image as follows:

1. Add Gaussian blur to the log-frequency-magnitude spectrum in order to reduce noise.
2. Perform Canny Edge Detection to the blurred spectrum.
3. Take the maximum radius around the circle edge and multiply it with the downsampling factor.
4. Downsample the clear spectrum by factor 2 and repeat step 1-3.
5. Take average of radius of all resolution and add $\frac{k}{2}$ where k is the kernel size of the Gaussian kernel to compensate the "contour shrinkage" effect.

Tab. I showed the comparison between Pyramid Compensate Parameter Estimation(PCPE), PCPE without compensa-

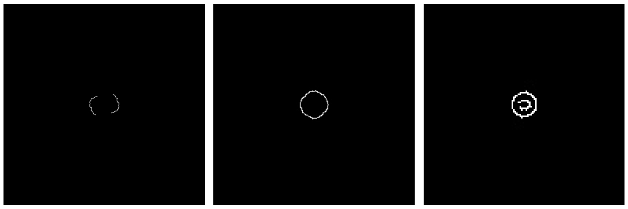

Fig. 5. The contour detected by Canny Edge Detector at resolution 512, 256 and 128.

tion and manually estimated result. The results shows that compensation helps the estimation much more close to the manually estimated results and get a better restoration result (shown in Fig. 6).

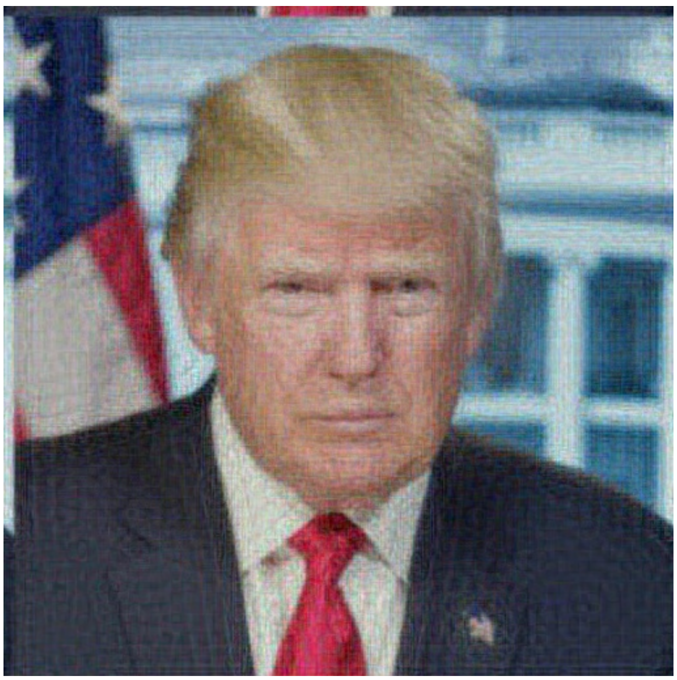

Fig. 6. The restoration of Trump

Motion Blur restoration: Motion Blur(MB) has the following mathematical representation in spatial domain as follows:

$$g(x,y) = \int_0^T f(x - x_0(t), y - y_0(t))\ dt \tag{3}$$

where T is the exposure time and $x_0(t), y_0(t)$ are $\frac{at}{T}, \frac{bt}{T}$ respectively. Hence, the Fourier Transform of Motion Blur kernel ($H(u,v)$) is:

$$\begin{aligned} G(u,v) &= F(u,v) \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\ dt \\ H(u,v) &= \frac{G(u,v)}{F(u,v)} \\ &= \frac{T}{\pi(au+bv)} \sin(\pi(au+bv)) e^{-j\pi(au+bv)} \\ &= T sinc(\pi(au+bv)) e^{-j\pi(au+bv)} \end{aligned} \tag{4}$$

By setting $H(u,v) = 0$, we get the following relation between a and b

$$au + bv = k,\ \ k = \pm1, \pm2, \cdots \tag{5}$$

which means the $H(u,v)$ takes a shape of a set of parallel lines along which $H(u,v)$ takes value zero.

In our project, we aim to restore an image of John Biden with unknown extent of motion blur and the corresponding image and its log-frequence-magnitude spectrum is shown in Fig. 7. However we encountered almost the same problems in previous PSF restoration and again we utilized PCPE to address those problems with the following precedure:

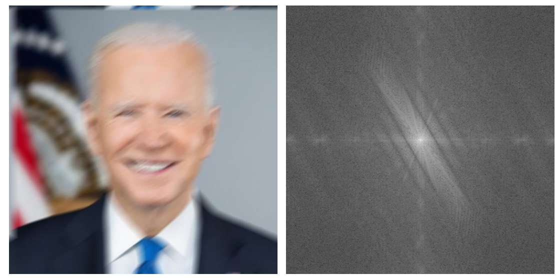

Fig. 7. The blurred image and the log-magnitude spectrum of `Biden` in frequency domain

1. Add Gasussian blur to the log-frequency-magnitude spectrum.
2. Perform Canny Edge Detection to the blurred spectrum.
3. Perform Hough Transform to determine the ρ and θ of two inner-most lines.
4. Calculate the distance between two inner-most lines, then add the distance with $\frac{k}{2}$ where k is the size of Gaussian blur kernel and multiply it with downsample factor.
5. Downsample the clear spectrum by factor 2 and repeat step 1-4.
6. Take average of distance and θ estimated from different resolution and get the final distance d and the final θ.

Once we estimated distance d and θ, we can calcuate a and b by the following formulas:

$$a = -\frac{2\cos(\theta)}{d} \tag{6}$$

$$b = -\frac{2\sin(\theta)}{d} \tag{7}$$

The mutil-resolution contour detected by Canny detector is shown in Fig. 8 and the comparison between different estimation method are shown in Tab. II. The results further proved that our proposed PCPE method well-estimated a and b and the restoration result is shown in Fig. 9.

method	PCPE/C	PCPE	manual
a	-0.02464	**-0.02157**	-0.02100
b	0.04210	**0.03686**	0.03574

TABLE II
COMPARISON BETWEEN PCPE AND PCPE WITHOUT COMPENSATION (PCPE/C) AND MANUALLY ESTIMATION a AND b

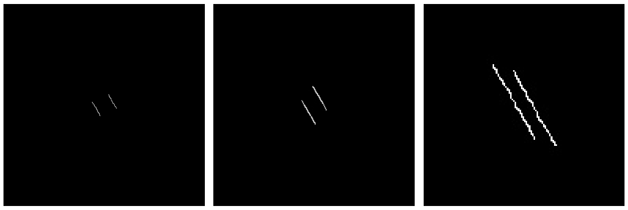

Fig. 8. The contour detected by Canny Edge Detector at resolution 512,256 and 128.

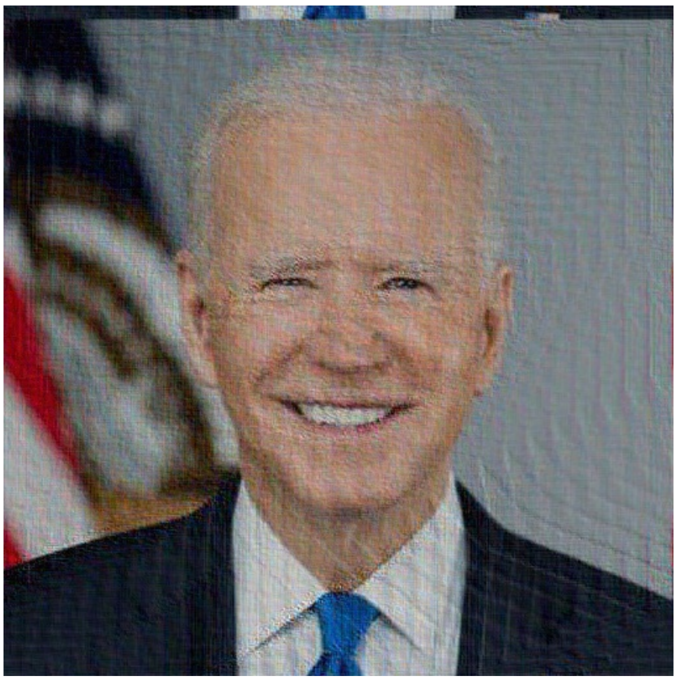

Fig. 9. The restoration of Biden

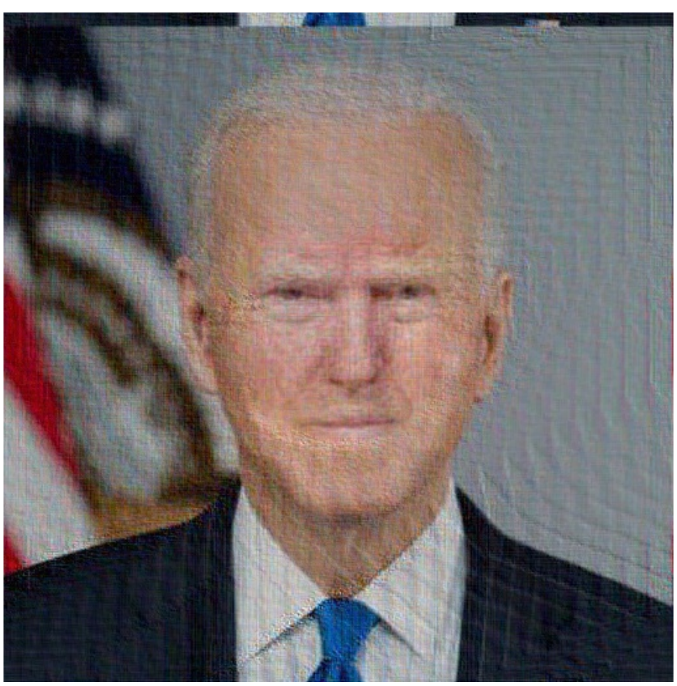

Fig. 10. The binding result of Biden and Trump

Image Binding: Once we get the restored images of Biden and Trump, we leverage the Gaussian and Laplacian Pyramid and the given mask to bind two image and the result is shown in Fig. 10.

References

[1] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[2] Ruchi Gajjar, Aditi Pathak, and Tanish Zaveri. Defocus blur parameter estimation technique. *International Journal of Electronics and Communication Engineering and Technology*, 7(4):85–90, 2016.

[3] Ruchi Gajjar and Tanish Zaveri. Defocus blur parameter estimation using polynomial expression and signature based methods. In *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 71–75. IEEE, 2017.