Assignment 4

Mu-Ruei Tseng

March 9, 2024

1 Overview

In this assignment, we perform gradient-domain image blending to reduce the color mismatch to seamlessly blend the masked areas of the source image into a target photograph. There are two approaches: the Poisson blending approach, and the mixing gradient approach. A more detailed explanation will be discussed in the following sections.

2 Main Task - Poisson Blending

2.1 Poisson Blending

The core idea of Poisson image blending is that we want the gradient of the composite image inside the mask Ω to look as close as possible to the source image's gradient. We can define the loss function as:

$$\min_f \iint_\Omega ||\nabla f(x,y) - \nabla S(x,y)||^2 dxdy, \text{ subject to } f(x,y)|_{\partial\Omega} = T(x,y)|_{\partial\Omega} \tag{1}$$

where f is the composite image,S is the source image and T is the target image. We can rewrite our loss function as:

$$J(f) = \iint_\Omega L(x,y,f(x,y),\frac{\partial f}{\partial x},\frac{\partial f}{\partial y})dxdy \tag{2}$$

Since we want to minimize $J(f)$ w.r.t. f, we can apply the Euler-Lagrange equation to solve it:

$$\frac{\partial}{\partial x}(\frac{\partial L}{\partial f_x}) + \frac{\partial}{\partial y}(\frac{\partial L}{\partial f_y}) - \frac{\partial L}{\partial f} = 0 \tag{3}$$

where $L = ||\nabla f(x,y) - \nabla S(x,y)||^2 = (\frac{\partial f}{\partial x} - \frac{\partial S}{\partial x})^2 + (\frac{\partial f}{\partial y} - \frac{\partial S}{\partial y})^2$. Since L does not directly depend on f but on its derivatives, we have $\frac{\partial L}{\partial f} = 0$. Now we need to solve the function:

$$\frac{\partial}{\partial x}(\frac{\partial L}{\partial f_x}) + \frac{\partial}{\partial y}(\frac{\partial L}{\partial f_y}) = 0 \tag{4}$$

$$\frac{\partial}{\partial x}(2(\frac{\partial f}{\partial x} - \frac{\partial S}{\partial x})) + \frac{\partial}{\partial y}(2(\frac{\partial f}{\partial y} - \frac{\partial S}{\partial y})) = 0 \tag{5}$$

$$2(\frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 S}{\partial x^2}) + 2(\frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 S}{\partial y^2}) = 0 \tag{6}$$

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \Rightarrow \Delta f = \Delta S \tag{7}$$

Therefore, we have the solution for our loss function:

$$\begin{cases} \Delta f = \Delta S, & \text{if in } \Omega \\ f(x,y) = T(x,y), & \text{otherwise} \end{cases} \tag{8}$$

We can derive:

$$\Delta f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \text{ and} \tag{9}$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{1}{\epsilon}(\frac{f(x+1,y) - f(x,y)}{\epsilon} - \frac{f(x,y) - f(x-1,y)}{\epsilon}) \tag{10}$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{1}{\epsilon}(\frac{f(x,y+1) - f(x,y)}{\epsilon} - \frac{f(x,y) - f(x,y-1)}{\epsilon}) \tag{11}$$

with $\epsilon = 1$. Finally, we can see that:

$$\Delta f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4 \times f(x,y) \tag{12}$$

and

$$\begin{aligned} 4 \times f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) = \\ 4 \times S(x,y) - S(x+1,y) - S(x-1,y) - S(x,y+1) - S(x,y-1) \end{aligned} \tag{13}$$

Therefore, we can see that the solution is similar to applying the Laplacian matrix on point $f(x,y)$ for all points inside Ω. Now we can rewrite our solution further in matrix forms and solve for image $f \in \mathbb{R}^{H \times W}$. Let $X \in R^N$ as the flattening of the image f where $N = H \times W$. The solution is to solve the linear system:

$$AX = b \tag{14}$$

where $A \in R^{N \times N}$ and $b \in R^N$.

Matrix Construction To construct A and b, we first let A equal the identity matrix of size $N \times N$ and b equal the flatten target image T. This ensures our solution satisfies the second condition of the equation, i.e., $f(x,y) = T(x,y)$.

Next, we modify the rows that represent the pixel in Ω to satisfy our first condition. We denote $(x,y) \in \Omega$, $\langle A[i], x \rangle = b[i]$ where $i = y \times W + x$ and we have:

$$\begin{cases} A[i,i+1] = A[i,i-1] = A[i,i-W] = A[i,i+W] = -1 \\ A[i,i] = 4 \end{cases} \tag{15}$$

and

$$b[i] = 4 \times S[x,y] - S[x+1,y] - S[x-1,y] - S[x,y+1] - S[x,y-1] \tag{16}$$

We can observe that A is a sparse matrix, as each row of A contains either a single value (if x, y are not within Ω) or five values (if x, y are within Ω). To construct the matrix, I identify the rows and columns in A that require modification, along with their corresponding values, and utilize *scipy.sparse.csr_matrix* to initialize the sparse array. Subsequently, *spsolve* can be employed to solve the linear system $AX = b$.

Boundary Cases To address boundary cases where $(x, y) \in \Omega$ is located on the image boundary, I have padded the source and target images by one pixel on all sides, duplicating the values of the original boundary pixels. Similarly, the mask is padded with zeros. This approach eliminates instances of $(x, y) \in \Omega$ being at the image boundary. Given the one-pixel padding on all sides, it is necessary to select only the original region in the final composite image.

Color Image Since a color image comprises three channels, we process each channel independently to determine the corresponding X for each. Specifically, we have:

$$AX^c = b^c \tag{17}$$

where X^c represents the intensity of channel c, and b^c is derived from the pixel values in the target (T^c) and source (S^c) images at channel c.

Finally, after obtaining the blended image, we should clip the values so that all the pixels are within 0 and 1.

2.2 Results

Here, we present the outcomes of implementing our Poisson Blending technique. This technique notably enhances the blending quality compared to naive blending. However, in Example 6 (Fig. 6) and Example 7 (Fig. 7), the results appear blurry inside the mask. To address this, we aim to improve the blending quality by employing the mixing gradient approach. I also include one of my examples (Fig. 8).

Figure 1: Poisson Blending: Example 1. Offset: (210, 10)

Figure 2: Poisson Blending: Example 2. Offset: (10, 28)

Figure 3: Poisson Blending: Example 3. Offset: (140, 80)

Figure 4: Poisson Blending: Example 4. Offset: (-40, 90)

Figure 5: Poisson Blending: Example 5. Offset: (60, 100)

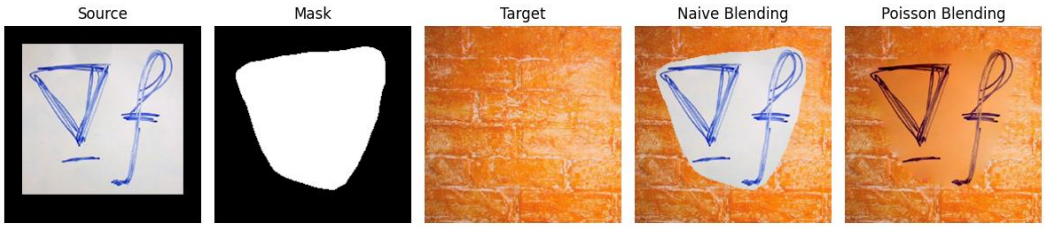

Figure 6: Poisson Blending: Example 6. Offset: (20, 20)

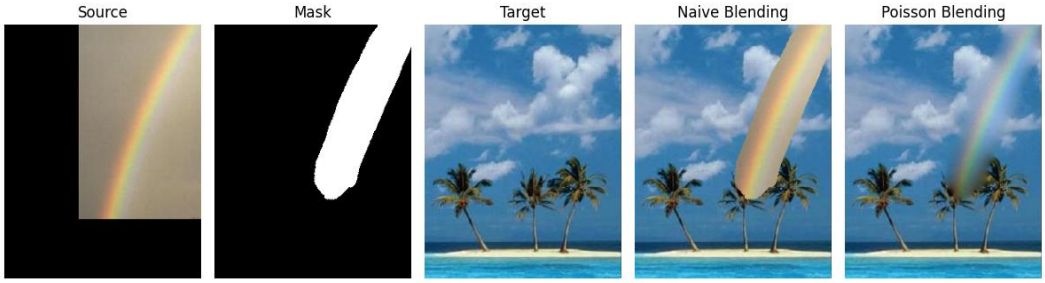

Figure 7: Poisson Blending: Example 7. Offset: (-28, 88)

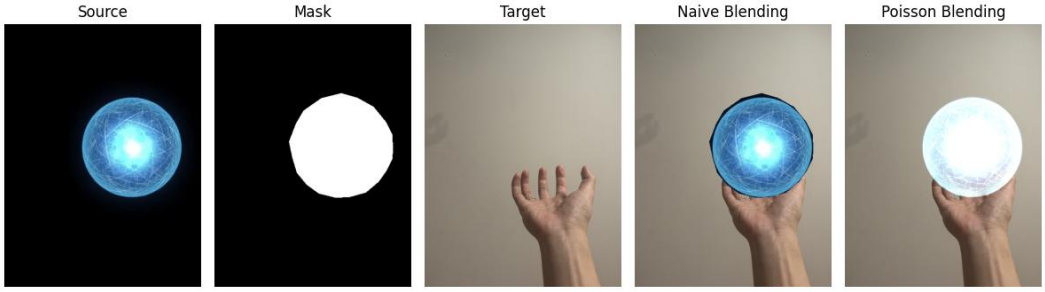

Figure 8: Poisson Blending: My Own Example. Offset: (70, -70)

3 Extra Credit - Mixing Gradient

In this section, we introduce the implementation of the mixing gradient method, aimed at enhancing the outcomes for cases such as examples 6 and 7. Distinguishing the mixing gradient approach from Poisson blending, the key variation lies in our handling of gradients for each pixel $(x, y) \in \Omega$. Rather than exclusively utilizing the source image's gradient, we conduct a comparison between the gradients of both the target and source images at each (x, y). If the gradient magnitude of the target image $T(x, y)$ is greater than that of the source image $S(x, y)$, the target image's gradient is then employed. We can rewrite it into a more general form:

$$\Delta f(x,y) = \nabla \cdot \begin{bmatrix} I_x(x,y) \\ I_y(x,y) \end{bmatrix} \tag{18}$$

where

$$\nabla I(x,y) = \begin{cases} \nabla T(x,y) \text{ , if } ||\nabla T(x,y)|| > ||\nabla S(x,y)|| \\ \nabla S(x,y) \text{ , otherwise} \end{cases} \tag{19}$$

$$\begin{aligned} &4 \times f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) = \\ &(I(x,y) - I(x+1,y)) + (I(x,y) - I(x-1,y)) + (I(x,y) - I(x,y+1)) + (I(x,y) - I(x,y-1)) \end{aligned} \tag{20}$$

Since the source and target are RGB images, we calculate the gradient per channel. Take $(I(x, y) - I(x + 1, y))$ for an example:

$$I(x,y) - I(x+1,y) = \begin{cases} T(x,y) - T(x+1,y) \text{ , if } |T(x,y) - T(x+1,y)| > |S(x,y) - S(x+1,y)| \\ S(x,y) - S(x+1,y) \text{ , otherwise} \end{cases} \tag{21}$$

We follow the same approach to solve the linear system $AX = b$. The sparse matrix A is the same as in Poisson Blending, but for the mixing gradient, we modify b according to the equation we discussed above.

3.1 Results

Here, we present the results of employing the mixing gradient approach to improve the blending performance for Examples 6 and 7, as shown in Fig. 9 and Fig. 10. By utilizing the mixing gradient method, we preserve the strong gradients present in the target image. This approach ensures that the textures and edges from the target are seamlessly integrated into the blended result, enhancing the visual fidelity and overall coherence of the composite image. The resulting blend appears more natural and reduces the blurriness observed in the mask regions of the initial Poisson Blending technique. I also include the result comparison in my own example (Fig. 11). We can see that by using the mixing gradient approach, we can preserve the hand in the original target image.

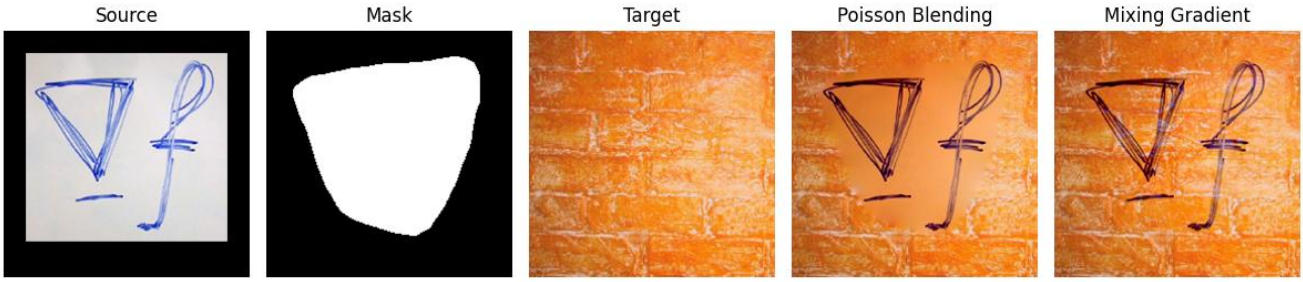

Figure 9: Mixing Gradient: Example 6. Offset: (20, 20)

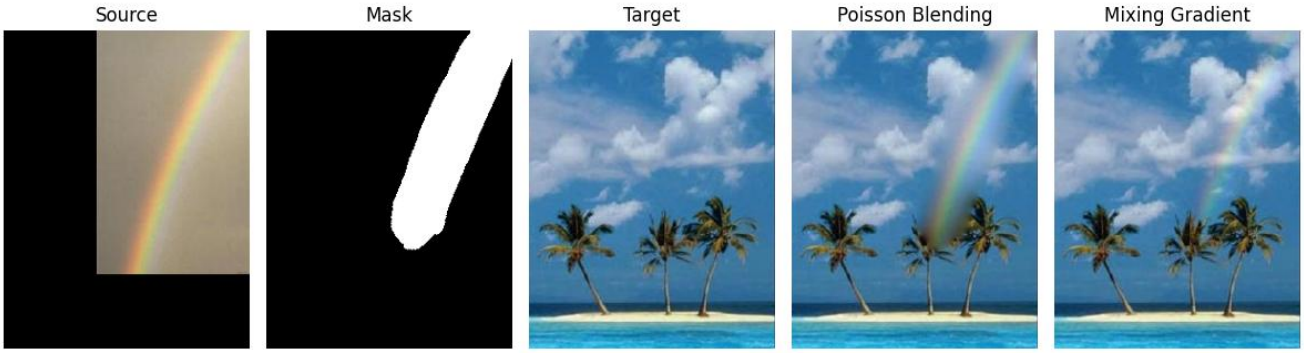

Figure 10: Mixing Gradient: Example 7. Offset: (-28, 88)

Figure 11: Mixing Gradient: My Own Example. Offset: (70, -70)