

Project 3: Gradient Domain Editing

Nguyễn Trang Sỹ Lâm

Ngày 30 tháng 10 năm 2024

Môn học: Xử lý Ảnh số và Thị giác Máy tính
Giảng viên hướng dẫn: Võ Thanh Hùng

1 Giới thiệu

Gradient của hình ảnh có thể được chỉnh sửa trực tiếp để thực hiện nhiều tác vụ hữu ích, được gọi là **gradient-based image processing** hoặc **Poisson editing**. Các tác vụ phổ biến như **seamless cloning** (sao chép liền mạch), **contrast enhancement** (tăng cường độ tương phản), **texture flattening** (làm phẳng bề mặt kết cấu), và **seamless tiling** (ghép ô liền mạch) đều có thể được thực hiện một cách đơn giản và hiệu quả bằng cách kết hợp hoặc thay đổi các gradient của hình ảnh.

Trong bài toán này, chúng ta sẽ tập trung vào kỹ thuật **Poisson Blending**, một phương pháp chỉnh sửa hình ảnh dựa trên gradient. Chúng ta sẽ thực hiện **Poisson Blending** để chèn đối tượng từ ảnh nguồn vào ảnh đích một cách tự nhiên và liền mạch. Kỹ thuật này dựa trên việc giải phương trình Poisson với các điều kiện biên nhằm đảm bảo rằng sự chuyển tiếp giữa đối tượng và nền là mượt mà cả về mặt màu sắc lẫn độ sáng. Phương trình Poisson giúp đảm bảo rằng gradient của ảnh trong vùng được chèn sẽ tương đồng với ảnh nguồn, trong khi các giá trị pixel tại biên khớp với ảnh đích.

2 Công thức toán

2.1 Poisson Blending

Poisson Blending nhằm mục đích chèn liền mạch một đối tượng từ ảnh nguồn S vào ảnh đích D bằng cách giải phương trình Poisson. Phương pháp này đảm bảo rằng sự chuyển tiếp giữa vùng đối tượng và nền sẽ mượt mà, không xuất hiện biên sắc nét hoặc thay đổi bất thường về màu sắc.

Kỹ thuật Poisson Blending dựa trên việc giải phương trình Poisson để tìm hàm $f(x, y)$, là ảnh kết quả trong vùng Ω , thỏa mãn:

$$\Delta f = \Delta S \quad \text{trong} \quad \Omega \quad (1)$$

$$f = D \quad \text{trên} \quad \partial\Omega \quad (2)$$

Trong đó:

- Δ là toán tử Laplace, tính tổng các đạo hàm riêng bậc hai của hàm f :

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- f là hàm cường độ sáng cần tìm trong vùng Ω (kết quả sau khi chèn đối tượng).
- Ω là vùng mặt nạ nơi đối tượng từ ảnh nguồn S sẽ được chèn vào.
- $\partial\Omega$ là biên của vùng Ω .
- S là ảnh nguồn chứa đối tượng cần chèn.
- D là ảnh đích.

2.2 Giãn nở Hình thái học (Morphological Dilation)

Giãn nở là một phép toán hình thái học sử dụng một phần tử cấu trúc (structuring element) để mở rộng các vùng trắng (giá trị pixel cao) trong ảnh nhị phân.

Cho ảnh nhị phân I và phần tử cấu trúc S , phép giãn nở được định nghĩa như sau:

$$I \oplus S = \{z \mid (S)_z \cap I \neq \emptyset\} \quad (3)$$

Trong đó:

- $(S)_z$ là phần tử cấu trúc S được dịch chuyển bởi vectơ z .
- $I \oplus S$ là kết quả của phép giãn nở.

Mục tiêu của phép giãn nở trong ngữ cảnh này là mở rộng vùng mặt nạ của đối tượng để bao gồm cả các pixel biên, giúp quá trình pha trộn trở nên mượt mà hơn.

3 Hiện thực

Trong phần này, chúng ta sẽ hiện thực các bước chính của thuật toán bằng Python sử dụng OpenCV và NumPy.

3.1 Poisson Blending

```
def poisson_blend(source, target, mask, offset=(0, 0)):
    x_offset, y_offset = offset
    center_point = (x_offset + source.shape[1] // 2, y_offset +
source.shape[0] // 2)

    blended = cv2.seamlessClone(
        source, target, mask, center_point, cv2.MIXED_CLONE
    )

    return blended
```

3.2 Thêm Padding cho Mặt nạ bằng Giãn nở Hình thái học

```
import cv2
import numpy as np

def add_mask_padding(mask, padding=10):
    kernel = np.ones((padding, padding), np.uint8)
    padded_mask = cv2.dilate(mask, kernel, iterations=1)
    return padded_mask
```

3.3 Quy trình Tổng thể

```
# Read the background image and the object image (with alpha channel)
background = cv2.imread('image/background.jpg')
object_img = cv2.imread('image/object.png', cv2.IMREAD_UNCHANGED)

# Resize the object image if necessary
object_img = cv2.resize(object_img, (200, 267))

# Split the object image into color channels and the alpha channel
```

```

b, g, r, alpha = cv2.split(object_img)

# Merge the color channels back into an RGB image
object_rgb = cv2.merge([b, g, r])

# Create a mask from the alpha channel
mask = np.where(alpha > 0, 255, 0).astype(np.uint8)

# Add padding to the mask
padded_mask = add_mask_padding(mask, padding=15)

# Set the position to place the object on the background
offset = (200, 450)

# Perform Poisson blending
result = poisson_blend(object_rgb, background,
padded_mask, offset=offset)

# Save the result
cv2.imwrite('image/blended_result.jpg', result)

```

4 Kết quả và Thảo luận

4.1 Kết quả

Sau khi thực hiện các bước trên, chúng ta thu được ảnh kết quả với đối tượng được chèn một cách tự nhiên vào ảnh nền.



Hình 1: Ảnh nguồn.



Hình 2: Ảnh đích.



Hình 3: Ảnh kết quả sau khi thực hiện Poisson Blending.



Hình 4: Ảnh kết quả sau khi thực hiện Poisson Blending với padding = 0.



Hình 5: Ảnh kết quả sau khi thực hiện Poisson Blending dùng NORMAL_CLONE thay vì MIXED_CLONE.

4.2 Các Phương Pháp Cloning trong Poisson Blending

Normal Cloning (NORMAL_CLONE): Trong phương pháp Normal Cloning, kết cấu (gradient) của ảnh nguồn được giữ nguyên trong vùng được sao chép. Điều này có nghĩa là tất cả các chi tiết về kết cấu, màu sắc và ánh sáng từ ảnh nguồn sẽ được duy trì hoàn toàn trong khu vực chèn vào ảnh đích. Phương pháp này phù hợp khi đối tượng giữ nguyên đặc điểm của nó, bất kể nền như thế nào. Tuy nhiên, nếu sự khác biệt về ánh sáng giữa ảnh nguồn và ảnh đích quá lớn, phương pháp này có thể gây ra hiện tượng biên sắc nét.

Gradient Mixing (MIXED_CLONE): Phương pháp Mixed Cloning sử dụng tham số để kết hợp gradient từ cả ảnh nguồn và ảnh đích. Điều này đặc biệt hữu ích khi nền ảnh có nhiều chi tiết hoặc ánh sáng phức tạp. Thay vì chỉ giữ lại kết cấu của ảnh nguồn, Mixed Cloning lựa chọn các gradient chiếm ưu thế giữa ảnh nguồn và ảnh đích để tạo ra một sự pha trộn tự nhiên

hơn. Tuy nhiên, vì phương pháp này chỉ giữ lại gradient mạnh nhất, nó có thể không tạo ra các vùng mịn màng, đặc biệt nếu gradient của ảnh nguồn và ảnh đích rất khác nhau.

4.3 Hiệu quả của Giãn nở Hình thái học

Việc thêm padding vào mặt nạ bằng phép giãn nở hình thái học giúp mở rộng vùng pha trộn. Điều này đảm bảo rằng các gradient được kết hợp một cách chính xác ở biên của đối tượng, tránh hiện tượng "ghosting" hoặc các đường biên không mong muốn giữa đối tượng và nền.

5 Kết luận

Chúng ta đã thành công trong việc thực hiện kỹ thuật Poisson Blending để chèn một đối tượng vào ảnh nền một cách tự nhiên. Việc áp dụng giãn nở hình thái học cho mặt nạ và giải phương trình Poisson giúp đảm bảo sự chuyển tiếp mượt mà giữa đối tượng và nền, tránh các hiện tượng không mong muốn như ghosting.

6 Code đầy đủ

Mã nguồn đầy đủ có thể truy cập tại GitHub: [ComputerVisionAssignment](#).