

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

____o0o____



XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Project 6. Research of Seamless Cloning and its applications for natural-looking image integration.

GIẢNG VIÊN:

Lý Quốc Ngọc

MỤC LỤC

I. THÔNG TIN NHÓM.....	2
II. BÁO CÁO ĐỒ ÁN.....	3
Chương 1: Giới thiệu.....	3
Cung cấp một cái nhìn tổng quan về Seamless Cloning, về ý nghĩa khoa học và một số ứng dụng.....	4
Từ đó tìm ra giải pháp tốt phù hợp để nghiên cứu sâu hơn.....	4
Chương 2: Các công trình nghiên cứu liên quan.....	4
Chương 3: Phương pháp.....	6
Chương 4: Cài đặt và thử nghiệm.....	6
Chương 5: Kết luận và hướng phát triển.....	8

I. THÔNG TIN NHÓM

Nhóm: 6

Nhóm trưởng: Thái Văn Thiên

MSSV	Họ và Tên
20127631	Thái Văn Thiên
20127244	Đào Trọng Nam
20127157	Lê Nguyễn Chí Hiếu
19127146	Lê Nguyễn Huy Hoàng

II. BÁO CÁO ĐỒ ÁN

Chương 1: Giới thiệu

Định nghĩa:

- Seamless Cloning là một kỹ thuật được sử dụng để kết hợp hai hình ảnh theo cách mà mắt người không nhận thấy được.
- Có thể trộn hai hình ảnh lại với nhau một cách tự nhiên trông như một hình ảnh duy nhất
- Loại bỏ các đối tượng không mong muốn khỏi ảnh hoặc kết hợp nhiều ảnh thành một ảnh ghép

1.1. Ý nghĩa về khoa học của chủ đề

- Tạo dữ liệu đa dạng cho học sâu và big data bằng cách tích hợp đối tượng vào nhiều ngữ cảnh khác nhau.
- Cung cấp ảnh tăng cường chất lượng và tự nhiên, giúp cải thiện hiệu suất của các mô hình và ứng dụng trong lĩnh vực xử lý hình ảnh. (loại bỏ các đối tượng không mong muốn khỏi hình ảnh)

1.2. Ý nghĩa về ứng dụng của chủ đề

- **Nhiếp ảnh:** thêm hoặc bớt 1 hoặc nhiều đối tượng, cảnh vật vào ảnh nhưng không làm mất đi tính tự nhiên của nó
- **Thiết kế đồ họa:** tạo ra những hình ảnh và hoạt ảnh chân thực và thuyết phục (thiết kế quảng cáo,...)

1.3. Phát biểu bài toán

- Mục tiêu là chụp hai hình ảnh màu được chia tỷ lệ, được gọi là nguồn và đích, đồng thời sao chép nguồn vào mục tiêu sao cho ranh giới của nguồn trên mục tiêu xuất hiện liền mạch mà không làm hỏng nguồn.

- Người dùng sẽ chọn ranh giới của ảnh nguồn và ảnh đích, do đó chọn một tập hợp pixel riêng biệt từ nguồn và một vị trí riêng biệt trong ảnh đích.

- Khi ánh xạ nguồn vào mục tiêu, dự án sẽ cố gắng làm phẳng các ranh giới, loại bỏ/thay đổi bất kỳ sự khác biệt chói tai nào được tìm thấy trên các đường viền của hình ảnh nguồn bằng cách sửa đổi các pixel của hình ảnh nguồn.

Input: ảnh gốc, ảnh chứa điểm đích đến, ảnh làm nền hay gọi là mặt nạ

Output: ảnh được tạo ra bằng cách ghép ảnh gốc vào ảnh đích một cách tự nhiên, mượt mà.

Framework chung:

Thuật toán Poisson Seamless Cloning là một khung để kết hợp liền mạch hai hình ảnh với nhau bằng thuật toán dựa trên phương trình Poisson. Phương trình Poisson là một phương trình toán học mô tả cách một đại lượng vật lý, chẳng hạn như điện thế hoặc nhiệt độ, thay đổi như thế nào trong một không gian nhất định. Thuật toán Poisson Seamless Cloning sử dụng phương trình Poisson để tính toán các giá trị độ dốc ở ranh giới của hình ảnh nguồn và đích, sau đó sử dụng các giá trị này để trộn các hình ảnh lại với nhau theo cách liền mạch và trông tự nhiên.

Các bước của giải thuật:

Chuẩn Bị Đầu Vào:

Nhận hình ảnh đầu vào với đối tượng cần tích hợp và ngữ cảnh.

Xác định vùng đối tượng và vùng mục tiêu.

Phân Tích Hình Ảnh:

Xác định đặc trưng của đối tượng và ngữ cảnh.

Tạo mặt nạ cho vùng đối tượng và vùng ngữ cảnh.

Tích Hợp Đối Tượng:

Tính toán gradient cho cả đối tượng và ngữ cảnh.

Sử dụng gradient để tạo sự liên tục và tự nhiên.

1.4. Đóng góp

Cung cấp một cái nhìn tổng quan về Seamless Cloning, về ý nghĩa khoa học và một số ứng dụng.

Từ đó tìm ra giải pháp tốt phù hợp để nghiên cứu sâu hơn.

Chương 2: Các công trình nghiên cứu liên quan

Công trình	
Sparse Solvers for Poisson Seamless Cloning-Ben Humberston	<p>Bài toán là làm thế nào người ta có thể hợp nhất một số phần của hình ảnh vào một hình ảnh khác mà không nhìn thấy đường nối. Theo "đường nối", đó là sự gián đoạn ở biên giới của một khu vực clone nơi một người quan sát con người có thể nhận ra một cách dễ dàng.</p> <p>Áp dụng Poisson Seamless Cloning để làm mượt hình ảnh đích trước khi thực hiện cloning.</p> <p>Phương pháp này sử dụng các giải thuật giải hệ phương trình tuyến tính thưa để giải quyết bài toán. Các giải thuật này bao gồm Poisson blending, Laplacian pyramid blending,... và một số giải thuật khác.</p>
Poisson Image Analogy: Texture-Aware Seamless Cloning S. Yoshizawa and H. Yokota	<p>Áp dụng edge-aware filtering để làm mượt hình ảnh đích trước khi thực hiện cloning.</p> <p>Sau khi cloning, sử dụng một kỹ thuật tổng hợp texture để khôi phục lại chi tiết texture của hình ảnh đích.</p> <p>Kỹ thuật này có thể giúp tái tạo các chi tiết của ảnh đích mà Poisson-based interpolation có thể làm mất đi.</p>

- Lập bảng so sánh các giải pháp dựa trên một số tiêu chí tự chọn

Công trình	Phương pháp	Hiệu suất	Ưu điểm	Khuyết điểm
Sparse Solvers for Poisson Seamless Cloning - Ben Humberston	Dùng phương trình Poisson	Tốt	Hiệu suất cao, tính chính xác cao, phù hợp với xử lý ảnh quy mô lớn.	Giải phương trình Poisson có thể đòi hỏi nhiều tài nguyên tính toán, đặc biệt là trên ảnh có độ phân giải cao.
Poisson Image Analogy: Texture-Aware Seamless Cloning S. Yoshizawa and H. Yokota	Làm mịn ảnh bằng phương pháp Gauss. Áp dụng Poisson tại điểm pixel cần ghép và xử lý biên cạnh bằng phương pháp Dirichlet	Tốt	Độ chính xác cao, biên cạnh ít bị mờ, lỗi, pixel phần bị ghép khá hài hoà	Mặc dù hiệu suất tính toán là lợi ích, nhưng có thể khó để duy trì tính tự nhiên và seamless như khi sử dụng phương trình Poisson.

Chương 3: Phương pháp

Dùng kỹ thuật **Poisson Image Editing**: thực hiện các chỉnh sửa hình ảnh một cách mượt mà và tự nhiên bằng cách áp dụng phương trình Poisson.

The simplest interpolant f of f^* over Ω is the membrane interpolant defined as the solution of the minimization problem:

$$\min_f \iint_{\Omega} |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (1)$$

where $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ is the gradient operator. The minimizer must satisfy the associated Euler-Lagrange equation

$$\Delta f = 0 \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (2)$$

where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator. Equation 2 is a Laplace equation with Dirichlet boundary conditions. For image

Mục tiêu của **phương trình (1)** là tìm một hàm f sao cho độ dốc của nó bên trong Ω là **nhỏ nhất** (ảnh sẽ mượt mà).

Phương trình (2) là phương trình Laplace với điều kiện biên Dirichlet, đảm bảo giá trị của hàm f trên biên $\partial\Omega$ giữ nguyên giá trị của f^* .

Điều kiện ràng buộc f phải giữ nguyên giá trị trên biên $\partial\Omega$, tức là f tại biên phải bằng giá trị của f^* tại biên.

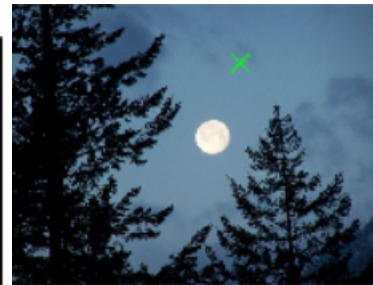
Chương 4: Cài đặt và thử nghiệm

B1: Thư viện

```
import numpy as np
import scipy.sparse as sps
import scipy.sparse.linalg as linalg
import cv2
import matplotlib.pyplot as plt
```

B2: Đọc ảnh:

```
image_data = {}
source = cv2.imread("../Data/source.png") # nguồn
mask = cv2.imread("../Data/mask.png") # mặt nạ
target = cv2.imread("../Data/target.png") # ảnh đích
```



B3: Tiền xử lý (cắt phần dư,...)

B4: Tính toán gradient và guidance field (v)

```
for i in range(num_pxls):
    # xây dựng ma trận khối laplacian
    # và guidance field
    if(mask[i] > 0.99):
        laplacian[i, i] = 4
        # xây dựng laplacian và tính toán gradient ảnh nguồn và đích
        if(i - Ws > 0):
            laplacian[i, i-Ws] = -1
            Np_up_s = source[i] - source[i-Ws]
            Np_up_t = target_subimg[i] - target_subimg[i-Ws]
        else:
            Np_up_s = source[i]
            Np_up_t = target_subimg[i]

        if(i % Ws != 0):
            laplacian[i, i-1] = -1
            Np_left_s = source[i] - source[i-1]
            Np_left_t = target_subimg[i] - target_subimg[i-1]
        else:
            Np_left_s = source[i]
            Np_left_t = target_subimg[i]
```

```
if(i + Ws < num_pxls):
    laplacian[i, i+Ws] = -1
    Np_down_s = source[i] - source[i+Ws]
    Np_down_t = target_subimg[i] - target_subimg[i+Ws]
else:
    Np_down_s = source[i]
    Np_down_t = target_subimg[i]

if(i % Ws != Ws-1):
    laplacian[i, i+1] = -1
    Np_right_s = source[i] - source[i+1]
    Np_right_t = target_subimg[i] - target_subimg[i+1]
else:
    Np_right_s = source[i]
    Np_right_t = target_subimg[i]

guidance_field[i] = (_compare(Np_up_s, Np_up_t) +
                     _compare(Np_left_s, Np_left_t) +
                     _compare(Np_down_s, Np_down_t) +
                     _compare(Np_right_s, Np_right_t))

else:
    # nếu điểm nằm bên ngoài mảng, sao chép hàm target
    laplacian[i, i] = 1
    guidance_field[i] = target_subimg[i]

return [laplacian, guidance_field]
```

B5: Giải phương trình Poisson

```
# Giải Phương trình Poisson
image_solution = np.empty_like(data['source'])
for i in range(3):
    image_solution[:, :, i] = linlsq_solver(equation_param[i][0], equation_param[i][1], data['source'].shape)

image_solution = stitch_images(image_solution, image['target'], ch_data['dims'])

return image_solution
```

Chương 5: Kết luận và hướng phát triển

Kết luận:

Seamless Cloning là một công nghệ mạnh mẽ trong xử lý hình ảnh, mang lại khả năng tích hợp hình ảnh một cách tự nhiên. Nghiên cứu của chúng ta đã chiếu sáng về vai trò quan trọng của nó trong thiết kế đồ họa, sản xuất phim và thực tế ảo.

Hướng phát triển:

- Nghiên cứu làm thế nào để Seamless cloning có thể tích hợp vào các ứng dụng thời gian thực như chỉnh sửa video.
- Khám phá cách tích hợp Seamless cloning để tăng tính thực tế của trải nghiệm AR và VR.
- Nghiên cứu cách tích hợp Seamless cloning với trí tuệ nhân tạo để tối ưu hóa quy trình sáng tạo.

- Hợp tác giữa nghiên cứu, nghệ sĩ và ngành công nghiệp để đưa ra những ứng dụng đa dạng.