

Xử lý ảnh số và video số

# **Research of Seamless Cloning and its applications for natural-looking image integration.**

Thành viên nhóm:

20127631 - Thái Văn Thiên (\*)

19127146 - Lê Nguyễn Huy Hoàng

20127244 - Đào Trọng Nam

20127157 - Lê Nguyễn Chí Hiếu

# Nội dung

1. Introduction
2. Related Research Works
3. Methodology
4. Implementation and Testing
5. Conclusion and Future Directions

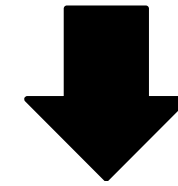
# Definition

- 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

# Definition



+



# Scientific Significance



- 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)

# Practical Applications

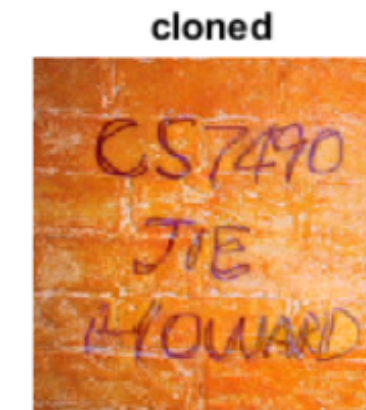
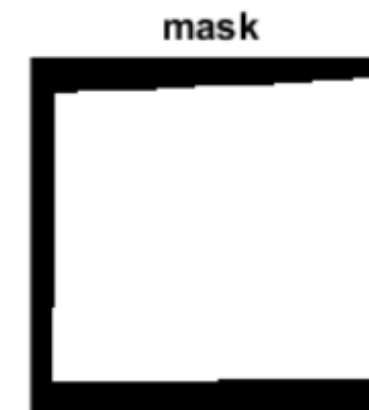
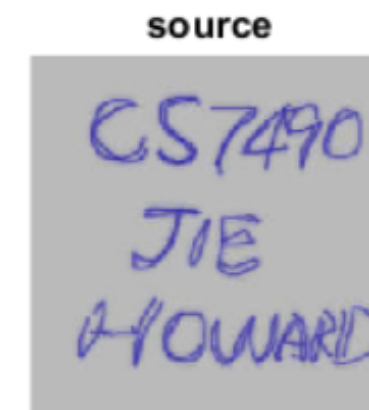
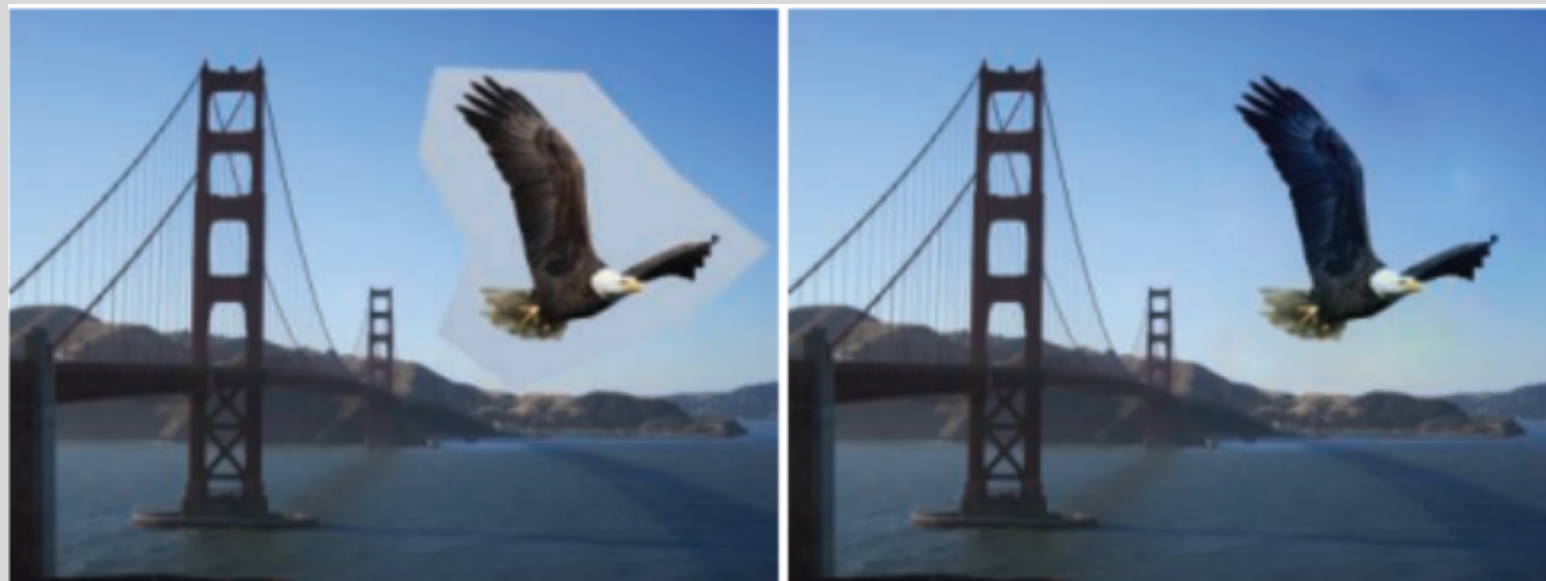
- **Nhiếp ảnh:** thêm hoặc bớt 1 hoặc nhiều đối tượng, cảnh vật vào ảnh nhưng không làm mất đi tính tự nhiên của nó
- **Thiết kế đồ họa:** tạo ra những hình ảnh và hoạt ảnh chân thực và thuyết phục (thiết kế quảng cáo,...)



# Problem Statement

Input: ảnh nguồn, ảnh đích.

Output: ảnh được tạo ra bằng cách hợp nhất các input một cách tự nhiên.





# Problem Statement

## 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.

# Contributions

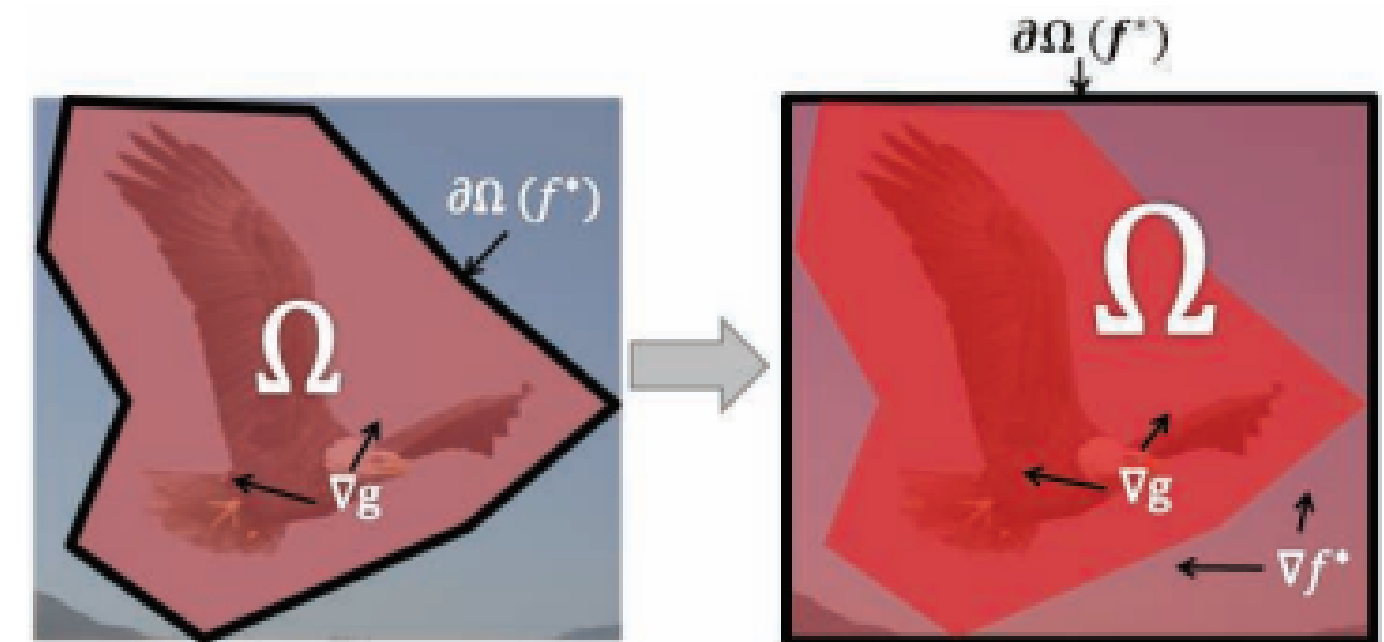
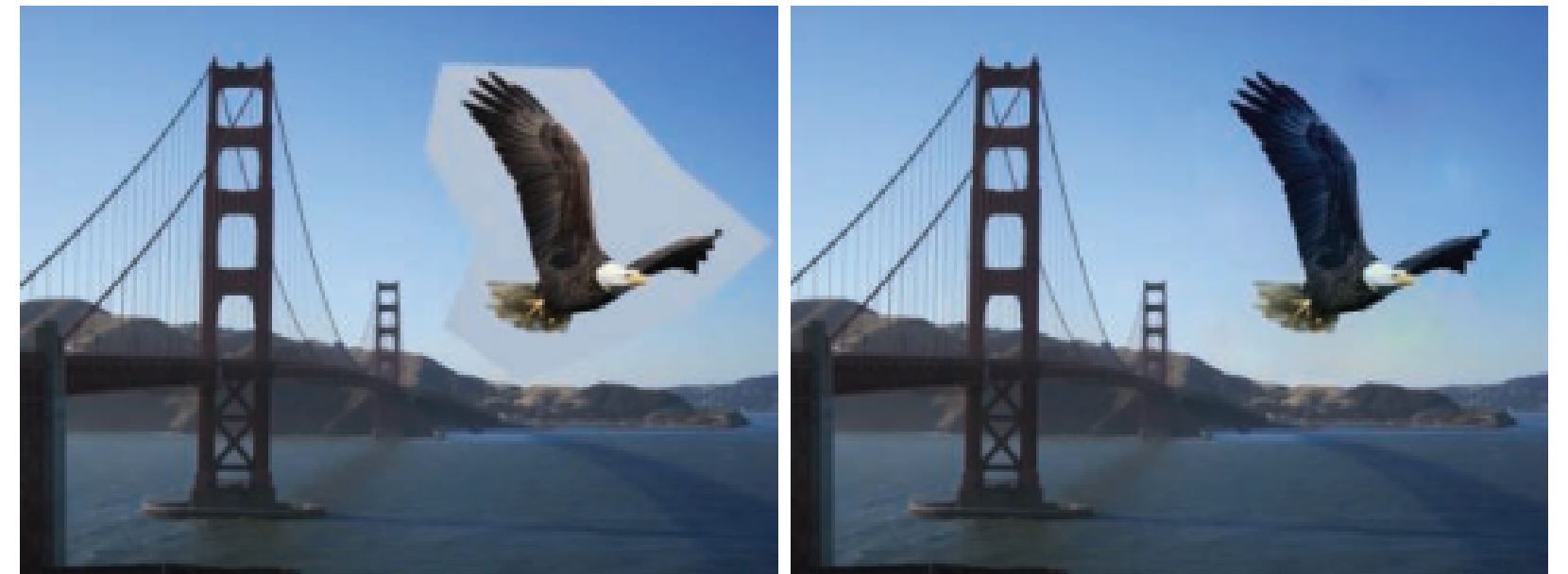
Cung cấp một cái nhìn tổng quan về Seamless Cloning, về ý nghĩa khoa học và một số ứng dụng.

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

# Related Research Works

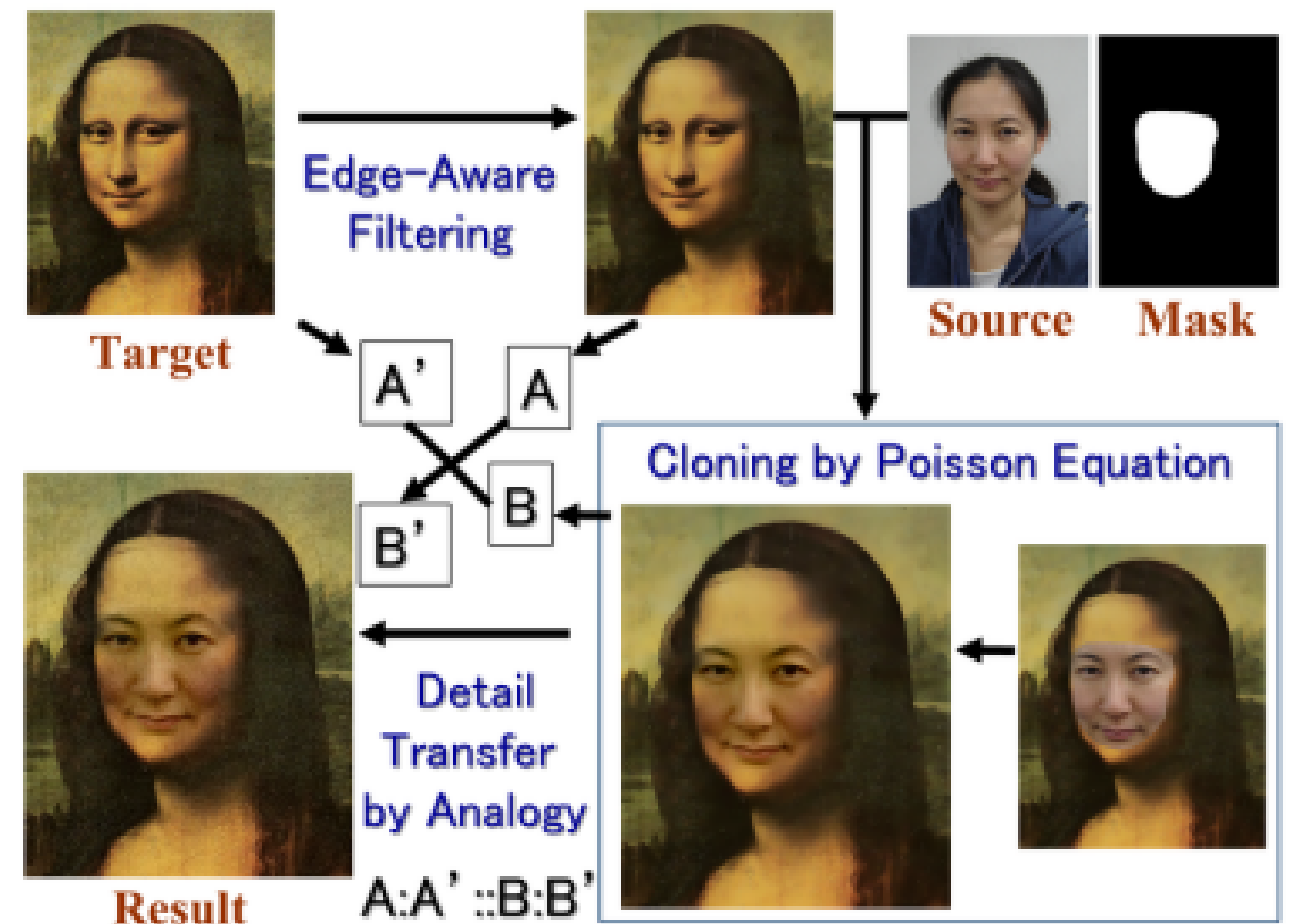
# Sparse Solvers for Poisson Seamless Cloning - Ben Humberston

- Áp dụng Poisson Seamless Cloning để làm mượt hình ảnh đích trước khi thực hiện cloning.
- 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.



# Poisson Image Analogy: Texture-Aware Seamless Cloning S. Yoshizawa and H. Yokota

- Áp dụng edge-aware filtering để làm mượt hình ảnh đích trước khi thực hiện cloning.
- Sau khi cloning, sử dụng một kỹ thuật tổng hợp texture để khôi phục lại chi tiết texture của hình ảnh đích.
- Kỹ thuật này có thể giúp tái tạo các chi tiết của ảnh đích mà Poisson-based interpolation có thể làm mất đi.



# Comparison table

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 lí 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.

# Methodology



# Methodology

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

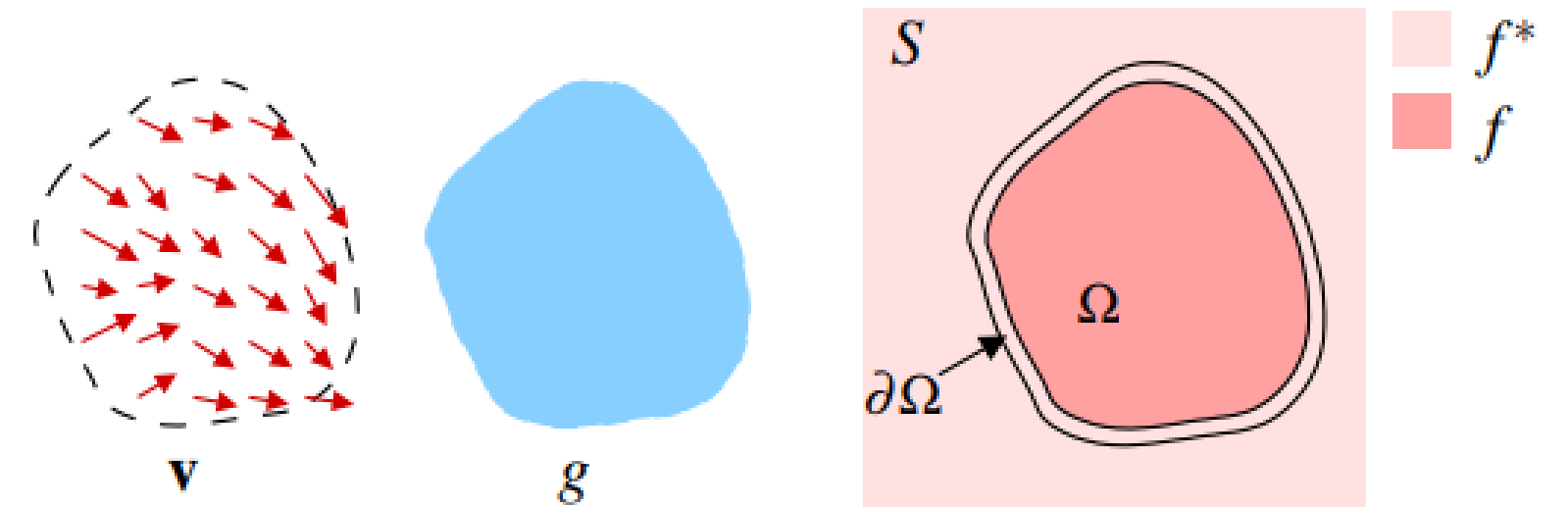


Figure 1: **Guided interpolation notations.** Unknown function  $f$  interpolates in domain  $\Omega$  the destination function  $f^*$ , under guidance of vector field  $\mathbf{v}$ , which might be or not the gradient field of a source function  $g$ .

# Methodology

**Chuẩn bị dữ liệu:** ảnh nguồn, ảnh mặt nạ và ảnh đích

**Áp dụng poisson image blending:** Tính toán gradients, laplacian, guidance field của ảnh nguồn và ảnh đích, sau đó giải phương trình Poisson để chuyển thông tin độ dốc từ ảnh nguồn sang ảnh đích.

**Áp dụng kết quả lên ảnh đích**

# Methodology

## Tổng quan

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

The simplest interpolant  $f$  of  $f^*$  over  $\Omega$  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  $\nabla \cdot = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$  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  $\Delta \cdot = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  is the Laplacian operator. Equation 2 is a Laplace equation with Dirichlet boundary conditions. For image

**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.

# Methodology

Để cải thiện kết quả nội suy, ta định nghĩa một **trường chỉ dẫn ( $\mathbf{v}$ )** và mở rộng bài toán như sau:

Trong đó,  $\mathbf{v}$  được định nghĩa bằng cách giữ lại các giá trị mạnh mẽ (lớn) hơn của gradient (cho từng kênh màu) giữa ảnh nguồn và ảnh mục tiêu.

masking out the textures, see Fig. 8.

However, the Poisson methodology allows non-conservative guidance fields to be used, which gives scope to more compelling effect. At each point of  $\Omega$ , we retain the stronger of the variations in  $f^*$  or in  $g$ , using the following guidance field:

$$\text{for all } \mathbf{x} \in \Omega, \mathbf{v}(\mathbf{x}) = \begin{cases} \nabla f^*(\mathbf{x}) & \text{if } |\nabla f^*(\mathbf{x})| > |\nabla g(\mathbf{x})|, \\ \nabla g(\mathbf{x}) & \text{otherwise.} \end{cases} \quad (12)$$

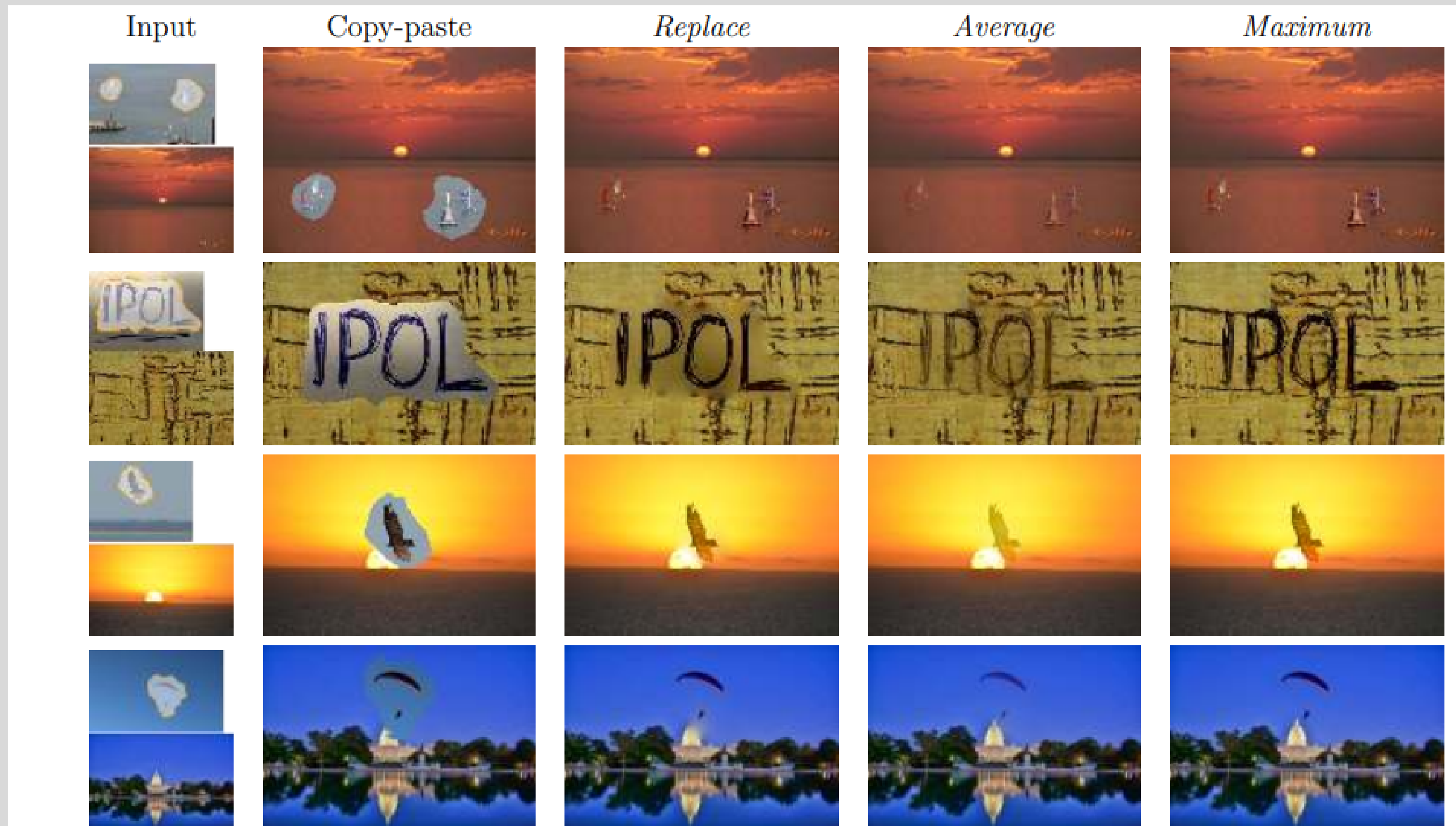
A *guidance field* is a vector field  $\mathbf{v}$  used in an extended version of the minimization problem (1) above:

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (3)$$

whose solution is the unique solution of the following Poisson equation with Dirichlet boundary conditions:

$$\Delta f = \text{div} \mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (4)$$

# Methodology



# Methodology

## Giải phương trình poisson

Sau khi xác định được bài toán và phương trình Poisson, giải phương trình ta sẽ thu được giá trị cường độ pixel mới tại các điểm trong vùng cần được clone (sao cho thoả mãn phương trình Poisson và điều kiện biên đã nêu trước đó).

Kết quả của quá trình là một hình ảnh mới mịn màng và trông tự nhiên, giữ nguyên cấu trúc của vùng clone và đảm bảo sự liên tục giữa biên và nội thất của vùng clone. Quá trình này giúp tạo ra hiệu ứng seamless cloning, nơi mà vùng clone tích hợp vào hình ảnh một cách mượt mà và không dễ nhận biết.

# Implementation and Testing



# Implementation and Testing

## Poisson image editing

## Chuẩn bị dữ liệu

```
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
```



# Implementation and Testing

## Tính toán gradient và guidance field (v)

```
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]
```

```
for i in range(num_pxls):
    # xây dựng ma trận khối laplacian
    # và trường hướng dẫn cho mảng
    if(mask[i] > 0.99):

        laplacian[i, i] = 4

    # xây dựng laplacian và tính toán gradient ảnh nguồn và
    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]
```

# Implementation and Testing

## 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
```

# Implementation and Testing

Kết quả



# Conclusion and Future Direction

# Conclusion and Future Direction

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.

# Future Direction:

1. Nghiên cứu làm thế nào để Seamless cloning có thể tích hợp vào các ứng dụng thời gian thực như chỉnh sửa video.
2. Khám phá cách tích hợp Seamless cloning để tăng tính thực tế của trải nghiệm AR và VR.
3. Nghiên cứu cách tích hợp Seamless cloning với trí tuệ nhân tạo để tối ưu hóa quy trình sáng tạo.
4. Hợp tác giữa nghiên cứu, nghệ sĩ và ngành công nghiệp để đưa ra những ứng dụng đa dạng.



# Reference

## Reference

[1] J. Matías Di Martino, Gabriele Facciolo, and Enric Meinhardt-Llopis, Poisson Image Editing, Image Processing On Line, 6 (2016), pp. 300–325.

(<https://doi.org/10.5201/ipol.2016.163>)

[2] J. Ben Humberston - University of British Columbia, Sparse Solvers for Poisson Seamless Cloning

([https://www.benhumberston.com/wp-content/uploads/2012/08/CPSC\\_517\\_BenHumberston\\_ProjectWriteup\\_compressed.pdf](https://www.benhumberston.com/wp-content/uploads/2012/08/CPSC_517_BenHumberston_ProjectWriteup_compressed.pdf))

[3] S. Yoshizawa and H. Yokota, Poisson Image Analogy: Texture-Aware Seamless Cloning

(<https://www2.riken.jp/briect/Yoshizawa/Papers/eg13Py.y.pdf>)

Cảm ơn mọi người đã lắng nghe