

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
NHẬP MÔN THỊ GIÁC MÁY TÍNH
Đề tài: Xóa đối tượng sử dụng Seam Carving

Thành viên	Mã số sinh viên
Nguyễn Đức Anh Phúc	20520276
Huỳnh Viết Tuấn Kiệt	20521494
Phan Hoàng Ân	16520017

Lớp	CS231.M21
Giảng viên hướng dẫn	TS. Nguyễn Vinh Tiệp

ĐỒ ÁN CUỐI KÌ – NHẬP MÔN THỊ GIÁC MÁY TÍNH

Tên đề tài (tiếng Việt): Xóa đối tượng sử dụng Seam Carving

Tên đề tài (tiếng Anh): Removing Object Using Seam Carving

Lớp: Nhập môn thị giác máy tính – CS231

Mã lớp: CS.231.M21

Giảng viên hướng dẫn: TS. Nguyễn Vinh Tiệp

Thành viên nhóm:

- ✓ Nguyễn Đức Anh Phúc – 20520276
- ✓ Huỳnh Viết Tuấn Kiệt – 20521494
- ✓ Phan Hoàng Ân – 16520017

Ngày bắt đầu: 29/05/2022

Ngày kết thúc: 03/07/2022

MỤC LỤC

<u>CHƯƠNG 01. TÓM TẮT</u>	5
<u>CHƯƠNG 02. GIỚI THIỆU</u>	6
<u>CHƯƠNG 03. XÓA ĐỐI TƯỢNG SỬ DỤNG SEAM CARVING</u>	7
03.01 THUẬT TOÁN SEAM CARVING	7
03.02 MINH HỌA BÀI TOÁN	8
03.03 MÔ HÌNH HÓA BÀI TOÁN	9
03.03.01 INPUT	9
03.03.02 OUTPUT	9
03.03.03 CONSTRAINT	9
<u>CHƯƠNG 04. PHƯƠNG PHÁP TIẾP CẬN BÀI TOÁN</u>	10
04.01 PHƯƠNG PHÁP TIẾP CẬN BÀI TOÁN	10
04.01.01 LỰA CHỌN ĐỐI TƯỢNG	10
04.01.02 LẤY MASK ĐỐI TƯỢNG	10
04.01.03 XÓA ĐỐI TƯỢNG	10
04.01.04 KHÔI PHỤC KÍCH THƯỚC HÌNH ẢNH	11
04.02 CÂY PHƯƠNG PHÁP TIẾP CẬN BÀI TOÁN	12
04.03 MÔ HÌNH PHƯƠNG PHÁP TIẾP CẬN BÀI TOÁN	12
<u>CHƯƠNG 05. PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN</u>	13
05.01 LỰA CHỌN ĐỐI TƯỢNG	13
05.02 LẤY MASK ĐỐI TƯỢNG	15
05.03 XÓA ĐỐI TƯỢNG	17
05.03.01 SINH ẢNH NĂNG LƯỢNG	17
05.03.01.1 CÔNG THỨC	17
05.03.01.2 HƯỚNG TIẾP CẬN	18
05.03.02 TÍNH SỐ LƯỢNG ĐƯỜNG SEAM	18
05.03.03 THIẾT LẬP ĐƯỜNG SEAM	20
05.03.03.1 XÂY DỰNG BẢN ĐỒ SEAM	20

05.03.03.2 CHỌN ĐƯỜNG SEAM CÓ MỨC NĂNG LƯỢNG NHỎ NHẤT	21
05.03.04 XÓA ĐƯỜNG SEAM	23
05.03.05 XÓA ĐỐI TƯỢNG KHỎI HÌNH ẢNH	25
05.04 KHÔI PHỤC KÍCH THƯỚC HÌNH ẢNH	27
05.04.01 LƯU LẠI CÁC ĐƯỜNG SEAM	27
05.04.02 CHÈN ĐƯỜNG SEAM	28
<u>CHƯƠNG 06. CÀI ĐẶT VÀ THỰC NGHIỆM</u>	<u>30</u>
<u>CHƯƠNG 07. DEMO</u>	<u>33</u>
<u>CHƯƠNG 08. KẾT LUẬN</u>	<u>34</u>
08.01 ƯU ĐIỂM	34
08.02 HẠN CHẾ	34
<u>CHƯƠNG 09. HƯỚNG PHÁT TRIỂN</u>	<u>35</u>
<u>CHƯƠNG 10. TÀI LIỆU THAM KHẢO</u>	<u>35</u>
<u>CHƯƠNG 11. BẢNG PHÂN CÔNG CÔNG VIỆC</u>	<u>36</u>
<u>CHƯƠNG 12. LỜI CẢM ƠN</u>	<u>36</u>

CHƯƠNG 01. TÓM TẮT

Các nội dung trong báo cáo gồm có:

- **0:** Tóm tắt toàn bộ nội dung báo cáo
- **CHƯƠNG 02:** Giới thiệu hoàn cảnh của bài toán xóa đối tượng sử dụng Seam Carving, giới thiệu bài toán.
- **0:** Mô tả các thao tác quan trọng với Seam Carving, sơ lược cách thức hoạt động trong thao tác xóa đối tượng của Seam Carving, minh họa cho bài toán và mô hình hóa bài toán.
- **CHƯƠNG 04:** Mô tả phương pháp tiếp cận bài toán, cây phương pháp tiếp cận và mô hình tổng thể phương pháp tiếp cận.
- **CHƯƠNG 05:** Chi tiết phương pháp giải quyết bài toán
- **CHƯƠNG 06:** Các hàm cài đặt thực nghiệm và mô tả chi tiết
- **0:** Link video quá trình thực hiện và file GIF
- **CHƯƠNG 08:** Kết luận, các ưu điểm và hạn chế của bài toán
- **CHƯƠNG 09:** Các hướng phát triển đề xuất cho bài toán
- **CHƯƠNG 10:** Các nguồn tài liệu và link tham khảo
- **CHƯƠNG 11:** Bảng phân công công việc nhóm
- **CHƯƠNG 12:** Lời cảm ơn

CHƯƠNG 02. GIỚI THIỆU

SEAM CARVING

Sự đa dạng và tính linh hoạt của các thiết bị hiển thị ngày nay đặt ra những nhu cầu mới về phương tiện kỹ thuật số. Việc hỗ trợ thay đổi động của bố cục trang, văn bản để thích ứng với các nội dung trang web và bố cục khác nhau của các thiết bị khác nhau là vấn đề mang tính cần thiết. Hình ảnh – biểu diễn phổ biến và quan trọng nhất của phương tiện kỹ thuật số thường vẫn có kích thước cứng và không thể biến dạng tự động để phù hợp với các bố cục khác nhau. Ngoài ra, để vừa với kích thước ipad, điện thoại di động hoặc để in ấn trên một khổ giấy hay độ phân giải nhất định, kích thước của hình ảnh cần phải thay đổi.

Cropping là giải pháp đơn giản để thay đổi kích thước hình ảnh, tuy nhiên nó bị hạn chế vì chỉ có thể loại bỏ các pixel khỏi cạnh biên hình ảnh. Do đó, tiêu chuẩn thay đổi kích thước hình ảnh hiệu quả có thể đạt được bằng cách xem xét nội dung hình ảnh thay vì các ràng buộc hình học.

Dựa trên tiêu chuẩn thay đổi kích thước hình ảnh trên, Sahi Avidan và Ariel Shamir đề xuất một thuật toán hình ảnh đơn giản, gọi là **Seam Carving** – thuật toán thay đổi kích thước hình ảnh có nhận thức nội dung.

CHƯƠNG 03. XÓA ĐỐI TƯỢNG SỬ DỤNG SEAM CARVING

03.01 Thuật toán Seam Carving

Khác với Cropping, thuật toán đề xuất có thể thay đổi kích thước hình ảnh bằng cách xóa hoặc chèn pixel một cách có trật tự vào các phần khác nhau của hình ảnh.

Seam Carving sử dụng một hàm năng lượng (**Energy function**) để xác định mức năng lượng của các pixels, mức năng lượng này ứng với tầm quan trọng của các pixels đó trong hình ảnh gốc. Đường Seam là một đường kết nối các pixel năng lượng thấp bằng qua hình ảnh từ trên xuống dưới hoặc từ trái sang phải. Bằng cách loại bỏ các đường Seam đó, chúng ta có thể giảm kích thước hình ảnh hoặc chèn liên tiếp các đường Seam để có được hình ảnh có kích thước lớn hơn theo cả 2 hướng. Để giảm kích thước hình ảnh, lựa chọn xóa các đường Seam thỏa mãn sau khi xóa vẫn có thể giữ được cấu trúc của hình ảnh và hình ảnh không trở nên quá bất thường. Tương tự, để phóng to hình ảnh, các đường Seam được lựa chọn phải có thứ tự chèn đảm bảo sự cân bằng giữa nội dung hình ảnh gốc và các pixels được chèn.

Xóa đối tượng sử dụng Seam Carving lấy ý tưởng dựa trên việc xóa các đường Seam có mức năng lượng thấp với việc thiết lập các pixel biểu diễn đối tượng cần xóa một mức năng lượng cực thấp để đảm bảo các đường Seam bị xóa sẽ đi qua đối tượng.

03.02 Minh họa bài toán

Ảnh gốc



Xác định đối tượng cần xóa



Thiết lập các đường Seam cần xóa



Hình ảnh sau khi xóa đối tượng



03.03 Mô hình hóa bài toán

03.03.01 Input

Một tấm hình



Hình mặt nạ đối tượng cần xóa



Hình mặt nạ đối tượng cần bảo vệ



03.03.02 Output

Ảnh sau khi xóa đối tượng



03.03.03 Constraint

Hình ảnh đầu vào và ảnh kết quả của bài toán phải có kích thước bằng nhau.

Nội dung hình ảnh còn lại sau khi xóa đối tượng không quá bất thường, tức là các đối tượng còn lại trong hình ảnh giữ nguyên bố cục ban đầu, không bóp méo hay bị nhòe

CHƯƠNG 04. PHƯƠNG PHÁP TIẾP CẬN BÀI TOÁN

Quy trình xây dựng phương pháp tiếp cận bài toán bao gồm 4 bước chính:

- Bước 1: Lựa chọn đối tượng (*)
- Bước 2: Lấy mask đối tượng
- Bước 3: Xóa đối tượng
- Bước 4: Khôi phục lại kích thước hình ảnh

(*) Cho phép điều khiển tương tác, quá trình lựa chọn được thực hiện bởi người dùng

04.01 Phương pháp tiếp cận bài toán

04.01.01 Lựa chọn đối tượng

Thao tác lựa chọn đối tượng cho phép lựa chọn **đối tượng cần xóa** – đối tượng sẽ hoàn toàn mất đi ở hình ảnh đầu ra và **đối tượng cần bảo vệ** – đối tượng sẽ được giữ lại ở hình ảnh đầu ra nhưng không đảm bảo toàn vẹn cấu trúc

04.01.02 Lấy mask đối tượng

Lấy mask đối tượng cần xóa và đối tượng cần bảo vệ đã lựa chọn ở bước 1 với ảnh mask được biểu diễn ở dạng **binary** – hai màu trắng và đen, trong đó đối tượng được lựa chọn sẽ biểu diễn màu trắng, phần background và phần còn lại biểu diễn màu đen.

04.01.03 Xóa đối tượng

Áp dụng cơ chế vận hành của thuật toán Seam Carving:

- Sinh ảnh năng lượng xác định các pixels chứa ít hay nhiều thông tin quan trọng (mức năng lượng của pixels)
 - + Đầu ra là một ma trận năng lượng của các pixels
- Tính toán số đường Seam cần xóa
 - + Số lượng đường Seam cần xóa để xóa hoàn toàn đối tượng khỏi hình ảnh ban đầu

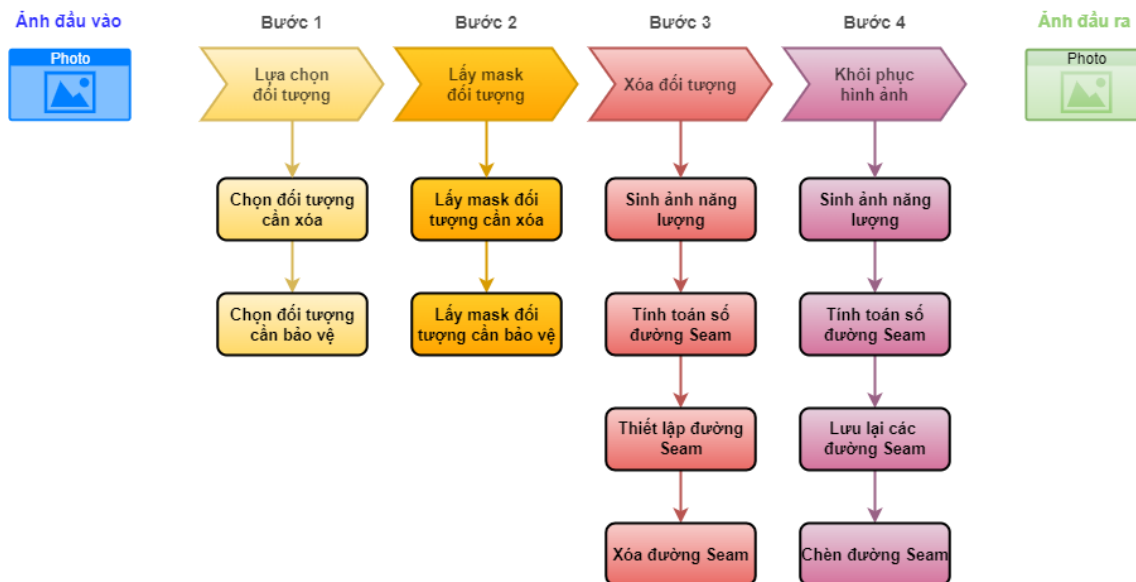
- Thiết lập các đường Seam có tổng năng lượng nhỏ nhất
 - + Đầu ra là đường Seam có tổng năng lượng nhỏ nhất (chứa ít thông tin quan trọng nhất)
- Xóa đường Seam đã được thiết lập khỏi hình ảnh theo chiều đã thiết lập
 - + Nếu xóa theo chiều dọc, các pixels bên phải đường Seam bị xóa sẽ đẩy qua bên trái
 - + Nếu xóa theo chiều ngang, các pixels bên dưới đường Seam bị xóa sẽ đẩy lên bên trên
 - + Đầu ra là hình ảnh với kích thước nhỏ hơn hình ảnh gốc

04.01.04 Khôi phục kích thước hình ảnh

Áp dụng cơ chế vận hành của thuật toán Seam Carving tương tự như ở bước **04.01.03**, nhưng thay vì xóa các đường Seam, ta chèn các đường Seam để khôi phục kích thước của hình ảnh.

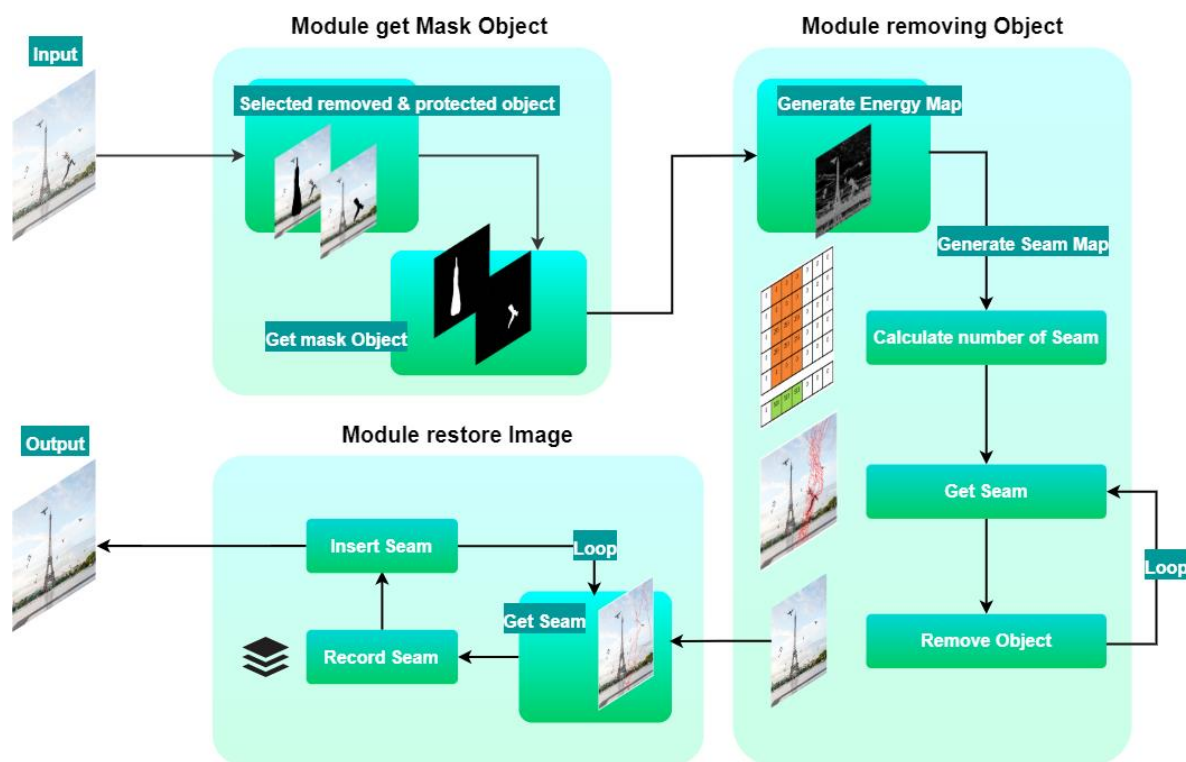
- Sinh ảnh năng lượng từ hình ảnh đã xóa đối tượng
 - + Đầu ra là một ma trận năng lượng của các pixels
- Tính toán số đường Seam cần khôi phục
 - + Số lượng đường Seam cần khôi phục bằng với số lượng đường Seam đã xóa ở **04.01.03**
- Lưu lại các đường Seam cần khôi phục
 - + Đầu ra là một danh sách chứa các đường Seam cần khôi phục
- Chèn các đường Seam trong danh sách lưu trữ vào hình ảnh đã xóa đối tượng
 - + Hình ảnh xóa đối tượng với kích thước bằng kích thước ảnh gốc

04.02 Cây phương pháp tiếp cận bài toán



Hình 4.2. Cây phương pháp tiếp cận bài toán

04.03 Mô hình phương pháp tiếp cận bài toán



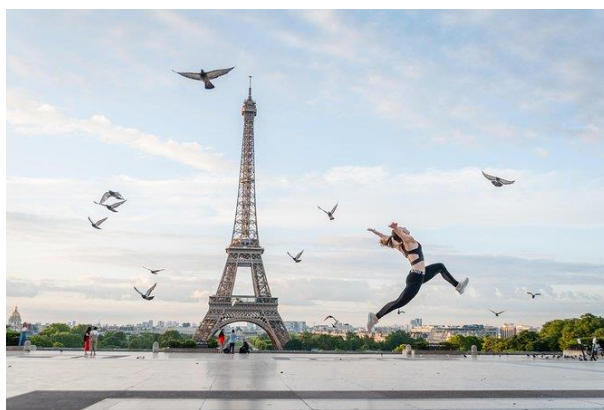
Hình 4.3. Mô hình phương pháp tiếp cận bài toán

CHƯƠNG 05. PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN

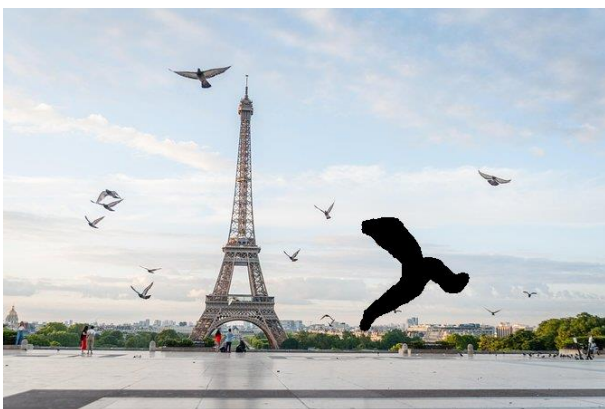
05.01 Lựa chọn đối tượng

Mục tiêu	Xác định được đối tượng cần xóa và đối tượng cần bảo vệ
Input	Hình ảnh đầu vào
Output	Hình ảnh ban đầu với vùng đánh dấu đối tượng cần xóa và đối tượng cần bảo vệ

Không có bất kì tham số nào trong quá trình loại bỏ hoặc chèn đường Seam. Tuy nhiên, để cho phép điều khiển tương tác, chúng tôi cung cấp giao diện người dùng dựa trên các nét vẽ nguệch ngoạc bằng chuột để thêm trọng số vào năng lượng của các pixel trên hình ảnh và điều chỉnh các kết quả có thể xảy ra mong muốn.



Hình 5.1.1. Hình ảnh ban đầu



Hình 5.1.2. Hình ảnh chọn đối tượng cần xóa (vùng màu đen)



Hình 5.1.3. Hình ảnh chọn đối tượng cần bảo vệ (vùng màu đen)

Xác định đối tượng cần xóa, bảo vệ và thực hiện các nét vẽ nguệch ngoạc bao phủ hoàn toàn phạm vi đối tượng và những vùng cần xóa, bảo vệ tương ứng

Lưu ý: Đối tượng cần xóa và đối tượng cần bảo vệ được vẽ trên 2 hình ảnh xuất hiện tuần tự (không vẽ 2 loại đối tượng trên cùng 1 lần xuất hiện hình ảnh)

Một số keycap bắt sự kiện bàn phím để vẽ chính xác và tiện hơn:

- Nút đại diện cho “*increase*”: Tăng kích thước nét vẽ
- Nút đại diện cho “*decrease*”: Giảm kích thước nét vẽ
- Nút : Tăng, giảm nhanh kích thước nét vẽ với tập giá trị {2, 50, 100}
- Nút : Xác nhận vẽ xong và kết thúc



Hình 5.1.4. Minh họa các loại nét vẽ có thể được sử dụng cho phù hợp với kích thước đối tượng trong quá trình lựa chọn đối tượng

05.02 Lấy mask đối tượng

Mục tiêu	Lấy ảnh mask
Input	Hình ảnh đầu vào
Output	Hình ảnh ban đầu với vùng đánh dấu đối tượng cần xóa và đối tượng cần bảo vệ

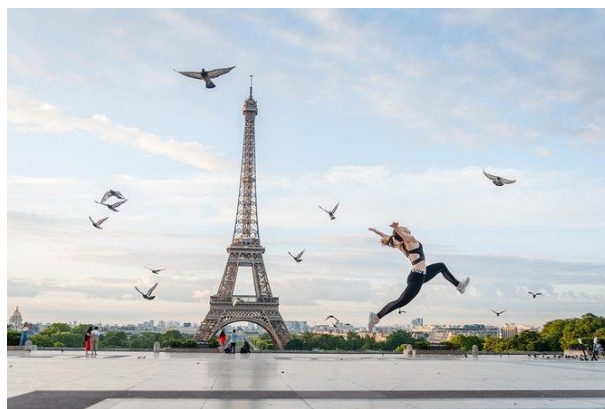
Mask đối tượng được lấy sử dụng cách tiếp cận **Brute force (vét cạn)**

```
for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        if img[i, j].all() == 0: img[i, j] = 255
        else: img[i, j] = 0
return img
```

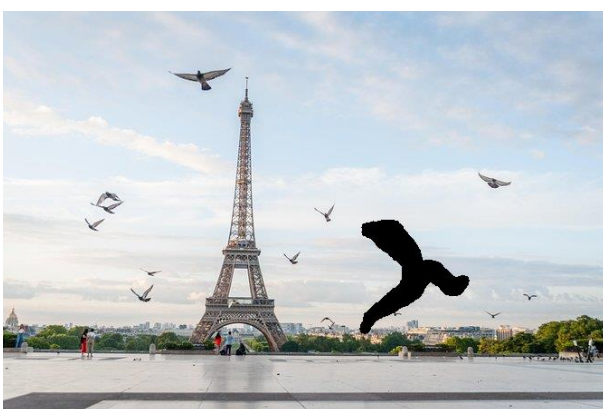
Hình 5.2.1. Thuật toán vét cạn tạo hình ảnh mask trắng đen

- Hình ảnh được vẽ ở bước lựa chọn đối tượng (mục **05.01**) tiến hành 2 vòng lặp duyệt qua n dòng và m cột của ma trận pixels.
- Các pixels mang giá trị (0,0,0) sẽ được chuyển thành (255,255,255) trong ảnh mask, các pixels còn lại chuyển thành (0,0,0). Do đó, ảnh mask sẽ mang 2 giá trị trắng đen với vùng màu trắng là vùng được chọn, phần còn lại màu đen là background và các đối tượng khác.

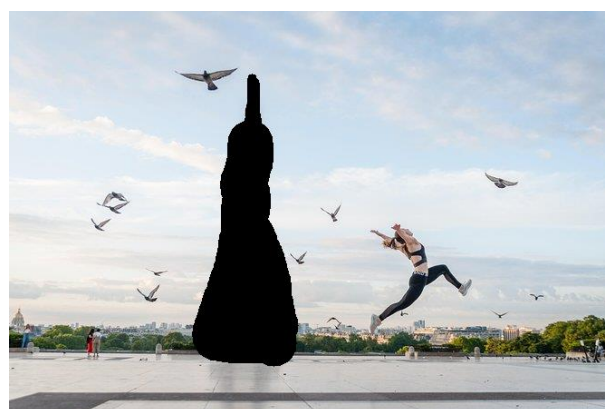
Đối tượng cần xóa và đối tượng cần bảo vệ được vẽ trên 2 hình ảnh riêng biệt nên ảnh mask của 2 đối tượng cũng riêng biệt



Hình 5.2.2. Hình ảnh ban đầu



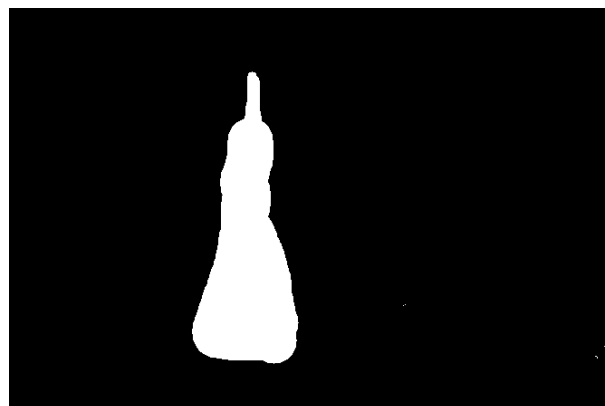
Hình 5.2.3. Hình ảnh chọn đối tượng cần xóa (vùng màu đen)



Hình 5.2.4. Hình ảnh chọn đối tượng cần bảo vệ (vùng màu đen)



Hình 5.2.5. Ảnh mask đối tượng cần xóa



Hình 5.2.6. Ảnh mask đối tượng cần bảo vệ

05.03 Xóa đối tượng

05.03.01 Sinh ảnh năng lượng

Mục tiêu	Tạo bản đồ năng lượng của hình ảnh
Input	Hình ảnh đầu vào ($h \times w \times d$)
Output	Bản đồ năng lượng của hình ảnh dạng ma trận năng lượng các pixel ($h \times w$)

Ảnh năng lượng là ảnh thể hiện những pixel mang thông tin quan trọng hay ít quan trọng, thông tin đó gọi là năng lượng của pixel tại mỗi điểm. Việc xác định chính xác năng lượng của pixel giúp xác định chính xác các pixel cần xóa (chứa ít thông tin), đảm bảo được tính bình thường của ảnh đầu ra.

05.03.01.1 Công thức

- Năng lượng tại mỗi điểm được xác định theo công thức:

$$e_{i,j}(I) = \sqrt{\left(\frac{\partial}{\partial x} I\right)^2 + \left(\frac{\partial}{\partial y} I\right)^2}$$

Trong đó

- + I : Hình ảnh (Image)
 - + $e_{i,j}(I)$: Năng lượng tại điểm pixel $I(i, j)$
 - + $\frac{\partial}{\partial x} I = I(i+1, j) - I(i-1, j)$: Độ biến thiên các pixel lân cận theo trục dọc
 - + $\frac{\partial}{\partial y} I = I(i, j-1) - I(i, j+1)$: Độ biến thiên các pixel lân cận theo trục ngang
- Minh họa

2	4	1
6	2	3
0	7	8

$$e_{2,2} = \sqrt{(4-7)^2 + (6-3)^2} = 3\sqrt{2}$$

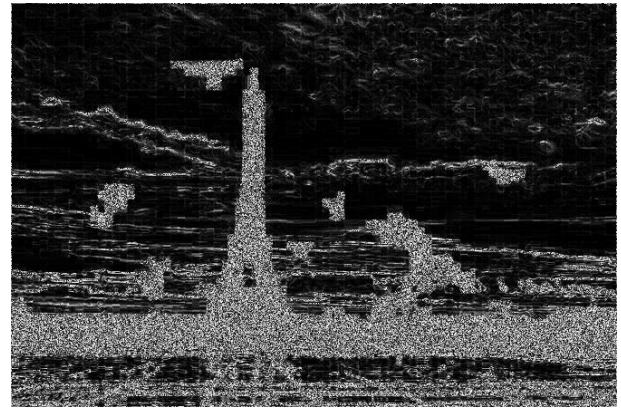
	$3\sqrt{2}$	

05.03.01.2 Hướng tiếp cận

Các nghiên cứu Seam Carving chỉ ra có nhiều cách sinh ảnh năng lượng khác nhau. Thuật toán phát hiện cạnh **Canny** có giai đoạn làm mịn ảnh để khử nhiễu¹. Do đó, các đối tượng chính trong ảnh sẽ có mức năng lượng cao hơn so với các đối tượng phụ.



Hình 5.3.1. Ảnh gốc



Hình 5.3.2. Bản đồ năng lượng

```
def genEnergyMap(img):  
    Gx = ndi.convolve1d(img, np.array([1, 0, -1]), axis=1, mode='wrap')  
    Gy = ndi.convolve1d(img, np.array([1, 0, -1]), axis=0, mode='wrap')  
    energyMap = np.sum(Gx**2, axis=2) + np.sum(Gy**2, axis=2)  
    return energyMap
```

Hình 5.3.3. Thuật toán sinh ảnh năng lượng

05.03.02 Tính số lượng đường Seam

Mục tiêu	Tính toán số lượng đường Seam vừa đủ để xóa đối tượng khỏi hình ảnh
Input	Ảnh mask chứa vùng đối tượng cần xóa
Output	Một giá trị nguyên n thể hiện số đường Seam cần xóa

¹ Tham khảo từ báo cáo liên quan

Ý tưởng: Từ ảnh mask đối tượng cần xóa đã có ở bước **05.02**, duyệt qua hình ảnh từ trái sang phải và từ trên xuống dưới, tính tổng theo các cột có trong hình ảnh, số lượng cột có tổng khác 0 là giá trị n cần tìm.

```
tmp = np.sum(maskDelete.T,axis = 1)
lenObject = len(np.where(tmp!=0)[0])
```

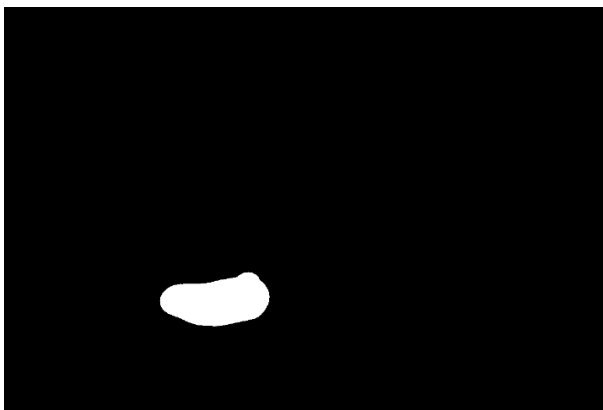
maskDelete: Ảnh mask đối tượng cần xóa

axis=1: tính tổng theo cột

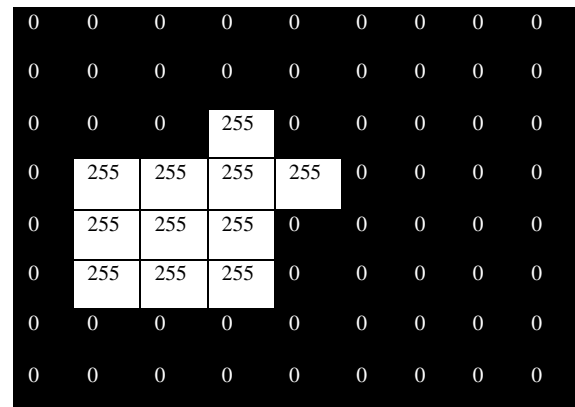
np.where: Kiểm tra điều kiện

Hình 5.3.4. Thuật toán tính số đường Seam cần xóa

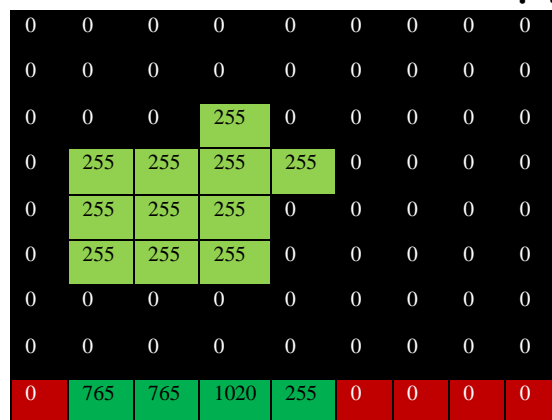
Ảnh mask mang 2 giá trị (0 và 255), với đối tượng thể hiện giá trị 255 và background thể hiện giá trị 0. Do đó, các cột có tổng khác 0 là các cột có chứa giá trị 255 (có chứa thành phần của đối tượng)



Hình 5.3.5. Ảnh mask đối tượng cần xóa



Hình 5.3.6. Ảnh mask đối tượng cần xóa dạng ma trận các pixel



Hình 5.3.7. Tính tổng các cột

Dựa vào hình 5.3.7, số lượng đường Seam cần xóa là 4, vì có 4 giá trị khác 0

05.03.03 Thiết lập đường Seam

Mục tiêu	Chọn ra được đường Seam chứa ít thông tin quan trọng nhất trong hình ảnh
Input	Bản đồ năng lượng ($h \times w$)
Ouput	Đường Seam có mức năng lượng nhỏ nhất trong hình ảnh

05.03.03.1 Xây dựng bản đồ Seam

Bản đồ Seam được xây dựng từ bản đồ năng lượng đã được thiết lập ở 05.03.01 bằng cách áp dụng kỹ thuật quy hoạch động.

Ý tưởng: Trên bản đồ Seam, giá trị tại một điểm ảnh thuộc hàng i bằng giá trị tại điểm ảnh đó trên Energy map cộng với giá trị liên kề nhỏ nhất nằm ở hàng $i - 1$.

Công thức xây dựng bản đồ Seam tại điểm ảnh i, j :

$$SMap_{i,j} = EMap_{i,j} + \min \begin{pmatrix} SMap_{i-1,j-1} \\ SMap_{i-1,j} \\ SMap_{i-1,j+1} \end{pmatrix}$$

- $SMap$: Seam map
- $EMap$: Energy map
- *Lưu ý: Công thức trên chỉ ra rằng giá trị ở hàng đầu tiên của $EMap$ sẽ là giá trị hàng đầu tiên của $SMap$ (vì hàng đầu tiên thì không có hàng ở trên để tính toán. Từ các hàng tiếp theo sẽ áp dụng công thức trên*

10	5	4	12
26	23	6	3
2	11	8	5
43	22	1	3

(a)

10	5	4	12
31			

(b)

10	5	4	12
31	27		

(c)

10	5	4	12
31	27	10	7
29	21	15	12
64	37	13	15

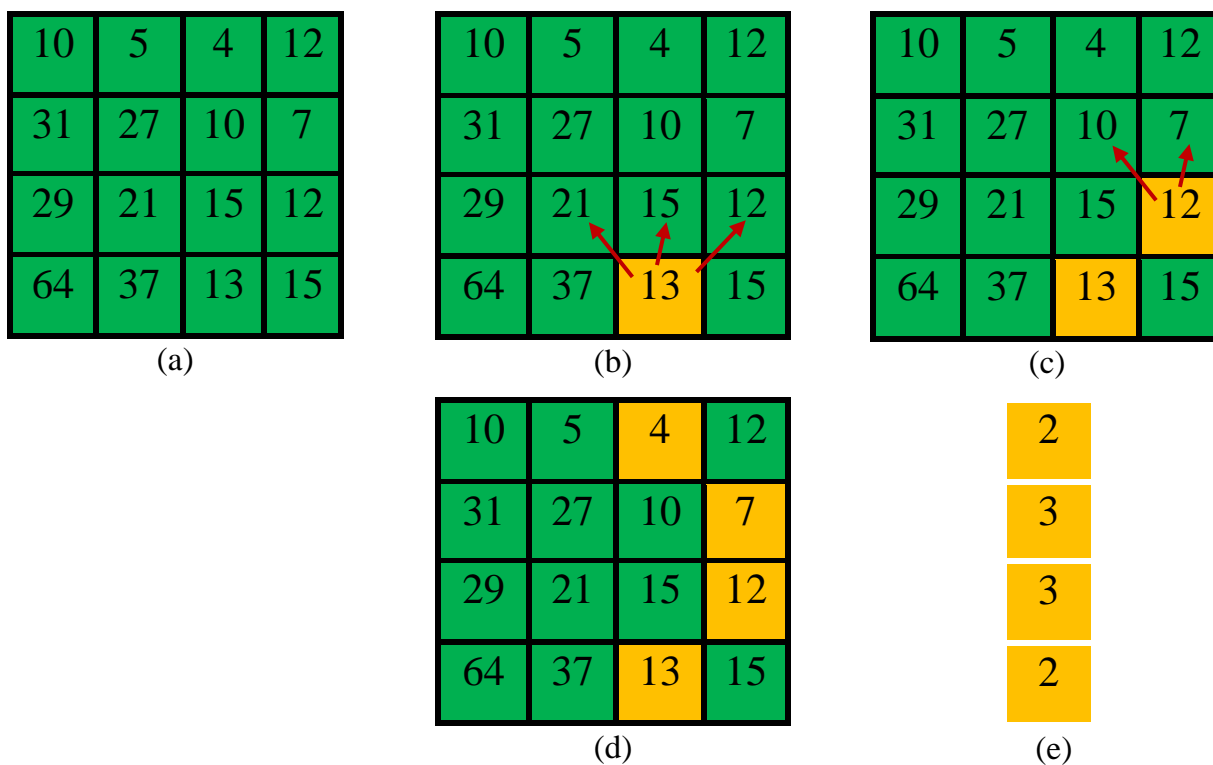
(d)

Hình 5.3.8. Minh họa thao tác tính toán tại một điểm trên Seam Map

- (a) Bản đồ năng lượng
- (b) Giá trị liên kề nằm trên ô $I(2,2)$
- (c) Chọn giá trị liên kề nằm trên nhỏ nhất và cập nhật giá trị tại ô $I(2,2)$
- (d) Bản đồ Seam

05.03.03.2 Chọn đường Seam có mức năng lượng nhỏ nhất

Từ bản đồ Seam được xây dựng ở mục **05.03.03.1**, để chọn được đường Seam chứa ít thông tin quan trọng nhất (có mức năng lượng thấp nhất), tiến hành truy xuất ngược từ dưới lên. Tại điểm ảnh thuộc hàng i , lựa chọn điểm ảnh có giá trị liên kề nhỏ nhất nằm ở hàng $i - 1$.



Hình 5.3.9. Minh họa thao tác tìm đường Seam có năng lượng nhỏ nhất

- (a) Bản đồ Seam
- (b) Bước đầu tiên, chọn giá trị nhỏ nhất hàng cuối cùng và tìm giá trị nhỏ nhất nằm liền kề ở hàng trên
- (c) Thực hiện tương tự cho đến khi đạt hàng đầu tiên
- (d) Hoàn thành tìm kiếm đường Seam nhỏ nhất
- (e) Ánh xạ kết quả với bản đồ năng lượng để tìm đường Seam tương ứng

Lưu ý quan trọng: Kết quả trả về của đường Seam là các **index** của các Pixel nằm trên đường Seam đó, không phải giá trị của pixel



Hình 5.3.10. Demo 100 đường Seam được lựa chọn trong hình ảnh

Nhận xét: Thuật toán lựa chọn các đường Seam tương đối tốt, các đường Seam không đi qua các đối tượng chính trong hình ảnh (tòa tháp và con người)

05.03.04 Xóa đường Seam

Mục tiêu	Xóa đường Seam đã chọn khỏi hình ảnh
Input	<p>Ảnh đầu vào $h \times w \times d$</p> <p>Mảng một chiều chứa các index của các pixel nằm trên đường Seam (h)</p>
Output	Hình ảnh sau khi xóa đường Seam ($h \times (w - 1) \times d$)

Ý tưởng:

- Cho ảnh ban đầu kích thước $h \times w \times d$.
- Khởi tạo ảnh kết quả với kích thước $h \times (w - 1) \times d$, ảnh sau khi xóa đường Seam sẽ nhỏ hơn ảnh gốc 1 pixels theo chiều ngang.
- Áp đường Seam cần xóa lên hình ảnh gốc, các pixel nằm bên trái đường Seam sẽ giữ nguyên giá trị trong ảnh kết quả, các giá trị nằm bên phải đường Seam sẽ dịch sang trái 1 pixel.

Thao tác xóa đường Seam

```
result[i, :seam[i], :] = img[i, :seam[i], :]  
if seam[i] + 1 <= w - 1:  
    result[i, seam[i]:, :] = img[i, seam[i] + 1:, :]
```

result: Ảnh kết quả

Hình 5.3.11. Thuật toán thực hiện thao tác xóa đường Seam

Mình họa thuật toán xóa đường Seam

11	3	21	2	4	2	2	23	5	32
2	0	23	3	7	32	21	2	23	3
4	57	4	1	88	1	41	3	21	23
2	32	2	42	9	23	11	5	4	2
4	4	1	3	2	21	5	4	12	3
4	1	44	1	34	23	15	3	31	23
12	34	2	22	53	1	15	6	23	2
32	3	5	22	22	31	1	54	13	32
13	17	2	41	11	3	5	36	1	3
4	4	5	21	23	43	35	5	31	2

(a)

0
1
2
2
1
2
3
4
5
5

(b)

11	3	21	2	4	2	2	23	5	32
2	0	23	3	7	32	21	2	23	3
4	57	4	1	88	1	41	3	21	23
2	32	2	42	9	23	11	5	4	2
4	4	1	3	2	21	5	4	12	3
4	1	44	1	34	23	15	3	31	23
12	34	2	22	53	1	15	6	23	2
32	3	5	22	22	31	1	54	13	32
13	17	2	41	11	3	5	36	1	3
4	4	5	21	23	43	35	5	31	2

(c)

3	21	2	4	2	2	23	5	32	
2	23	3	7	32	21	2	23	3	
4	57	1	88	1	41	3	21	23	
2	32	42	9	23	11	5	4	2	
4	1	3	2	21	5	4	12	3	
4	1	1	34	23	15	3	31	23	
12	34	2	53	1	15	6	23	2	
32	3	5	22	31	1	54	13	32	
13	17	2	41	11	5	36	1	3	
4	4	5	21	23	35	5	31	2	

(d)

Hình 5.3.12. Minh họa thao tác xóa đường Seam

- (a) Hình ảnh ban đầu
- (b) Mảng một chiều chứa các index của các pixel nằm trên đường Seam cần xóa
- (c) Áp đường Seam lên hình ảnh
- (d) Xóa đường Seam

05.03.05 Xóa đối tượng khỏi hình ảnh

Mục tiêu	Xóa hoàn toàn đối tượng được lựa chọn khỏi hình ảnh
Input	Ảnh kích thước $h \times w \times d$
Ouput	Hình ảnh sau khi xóa đối tượng ($h \times w' \times d$)

Kết hợp những bài toán nhỏ đã được mô tả trên, phương pháp xóa đối tượng khỏi hình ảnh được thực hiện với các bước:

- **Bước 1:** Lấy mask đối tượng cần xóa và bảo vệ
- **Bước 2:** Tính số lượng đường Seam cần xóa
- **Bước 3:** Sinh ảnh năng lượng
- **Bước 4:** Thiết lập và tính toán đường Seam cần xóa (đường Seam có mức năng lượng nhỏ nhất)
- **Bước 5:** Xóa đường Seam được thiết lập ở bước 4
- **Bước 6:** Lặp lại bước 3 cho tới khi xóa đủ số lượng đường Seam cần thiết

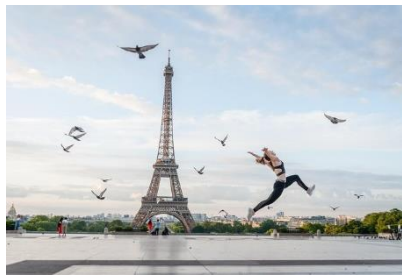
Bước 4 là bước quan trọng nhất, thuật toán Seam Carving ưu tiên xóa các đường Seam có mức năng lượng nhỏ nhất, do đó, không thể đảm bảo rằng các đường Seam đó luôn đi qua đối tượng cần xóa. Để xóa hoàn toàn đối tượng khỏi hình ảnh, các đường Seam được chọn ở mỗi lần lặp phải đi qua vùng đối tượng cần xóa. Và để đảm bảo điều này, thiết lập mức năng lượng của các pixel trong vùng đối tượng cần xóa là cực kì nhỏ. Tương tự với đối tượng cần bảo vệ, để đảm bảo đối tượng cần bảo vệ toàn vẹn bố cục sau khi xóa đối tượng khác, chỉ cần thiết lập mức năng lượng ở các pixel thuộc vùng đối tượng cần bảo vệ cực kì lớn để đảm bảo đường Seam ở mỗi vòng lặp không đi qua đối tượng cần bảo vệ. Tóm lại, phương pháp ràng buộc để xóa và bảo vệ đối tượng tương tự nhưng ngược nhau. Ý tưởng thực hiện như sau:

- Sinh ảnh năng lượng.
- Áp ảnh mask đối tượng cần xóa và bảo vệ lên ảnh năng lượng.

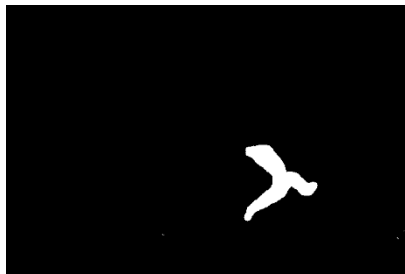
- Tại vị trí có mask đối tượng cần xóa (các pixel mang giá trị 255), thiết lập các pixel ở vị trí tương ứng trên ảnh năng lượng thành $-\mathbf{inf}$ (với $-\mathbf{inf}$ là một giá trị cực kì nhỏ).
- Tại vị trí có mask đối tượng cần bảo vệ (các pixel mang giá trị 255), thiết lập các pixel ở vị trí tương ứng trên ảnh năng lượng thành \mathbf{inf} (với \mathbf{inf} là một giá trị cực kì lớn).

Với phương pháp trên, các pixel tại vị trí đối tượng cần xóa là cực kì nhỏ, đảm bảo đường Seam sẽ đi qua các pixel đó. Tương tự, các pixel tại vị trí đối tượng cần bảo vệ là cực kì lớn, đảm bảo các đường Seam sẽ không đi qua các pixel đó. Tuy nhiên, trong trường hợp kích thước hình ảnh co lại còn đúng kích thước của đối tượng cần bảo vệ mà vẫn chưa xóa đủ số lượng đường Seam cần xóa thì dĩ nhiên các đường Seam còn lại sẽ đi qua đối tượng cần bảo vệ.

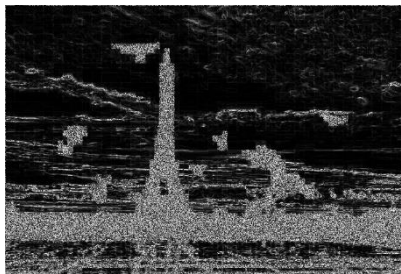
Minh họa cho phương pháp thiết lập mức năng lượng cực đại và cực tiểu:



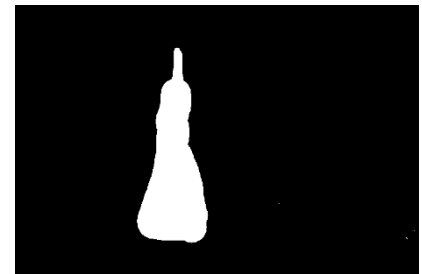
(a)



(b)



(c)



(d)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	255	0	0	0
0	0	0	0	255	255	0	0
0	0	0	255	255	0	0	0
0	0	0	0	0	0	0	0

(e)

3	3	2	6	3	4	2	
2	5	221	11	22	21	13	
4	6	206	21	65	5	11	
11	157	192	31	31	24	28	
9	221	90	12	31	57	64	
12	241	203	13	90	98	67	
22	132	225	99	222	98	78	
44	64	148	31	41	54	63	

(f)

0	0	0	0	0	0	0	0
0	0	255	0	0	0	0	0
0	0	255	0	0	0	0	0
0	255	255	0	0	0	0	0
0	255	255	0	0	0	0	0
0	255	255	0	0	0	0	0
0	0	255	0	0	0	0	0
0	0	0	0	0	0	0	0

(g)

3	3	2	6	3	4	2
2	5	<i>inf</i>	11	22	21	13
4	6	<i>inf</i>	21	65	5	11
11	<i>inf</i>	<i>inf</i>	31	31	24	28
9	<i>inf</i>	<i>inf</i>	12	<i>-inf</i>	57	64
12	<i>inf</i>	<i>inf</i>	13	<i>-inf</i>	<i>-inf</i>	67
22	132	<i>inf</i>	<i>-inf</i>	<i>-inf</i>	98	78
44	64	148	31	41	54	63

(h)

Hình 5.3.13. Thiết lập mức năng lượng cực đại và cực tiểu

- (a) Ảnh ban đầu
- (b) Mask đối tượng cần xóa
- (c) Bản đồ năng lượng
- (d) Mask đối tượng cần bảo vệ
- (e) Mask đối tượng cần xóa dạng ma trận pixels
- (f) Bản đồ năng lượng dạng ma trận pixels
- (g) Mask đối tượng cần bảo vệ dạng ma trận pixels
- (h) Thiết lập mức năng lượng cực đại và cực tiểu trên ảnh năng lượng

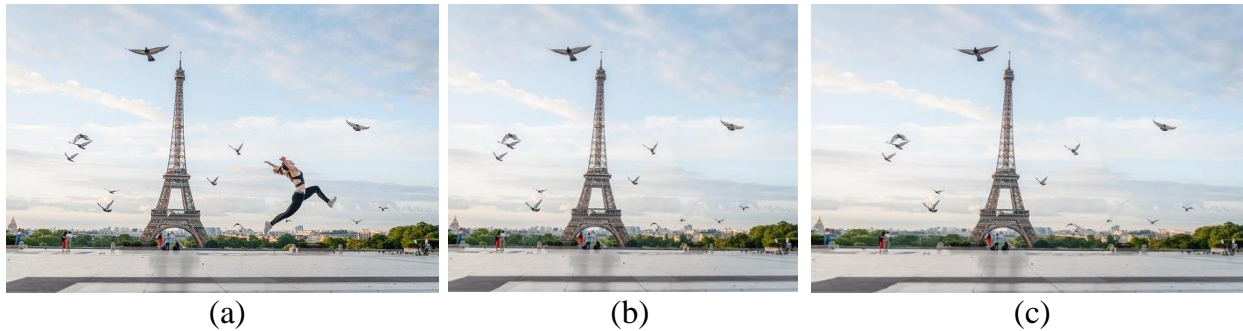
inf là giá trị cực lớn và *-inf* là giá trị cực nhỏ (Source code thiết lập lần lượt là $1e8$ và $-10e8$)

05.04 Khôi phục kích thước hình ảnh

05.04.01 Lưu lại các đường Seam

Mục tiêu	Lưu lại các đường Seam để chèn vào hình ảnh
Input	Ảnh kích thước $h \times w' \times d$ (ảnh đã xóa đối tượng)
Output	Tập hợp các đường Seam dùng khôi phục hình ảnh, mỗi đường Seam là một vector có độ dài h
Constraint	Kích thước ảnh còn lại sau khi xóa đối tượng không dưới 50% kích thước ảnh gốc

Sau khi xóa đối tượng, kích thước hình ảnh còn lại sau khi xóa nhỏ hơn kích thước hình ảnh gốc n pixels theo chiều ngang (với n là số đường Seam đã xóa). Tiến hành xóa thêm n đường Seam trên ảnh đã xóa hình ảnh và thực hiện thao tác lưu trữ các đường Seam đã xóa. Sau khi có n đường Seam được lưu trữ, chèn n đường Seam đó vào hình ảnh đã xóa đối tượng, ta được hình ảnh mới bằng với kích thước hình ảnh gốc.



Hình 5.4.1. Các giai đoạn biến thể của hình ảnh

- (a) Ảnh đầu vào
- (b) Ảnh sau khi xóa đối tượng
- (c) Ảnh sau khi khôi phục

05.04.02 Chèn đường Seam

Mục tiêu	Chèn đường Seam vào hình ảnh
Input	Ảnh kích thước $h \times w \times d$ Mảng một chiều chứa các index của các pixel nằm trên đường Seam (h)
Output	Hình ảnh sau khi chèn đường Seam ($h \times (w + 1) \times d$)

Ý tưởng:

- Cho ảnh ban đầu kích thước $h \times w$
- Khởi tạo ảnh kết quả với kích thước $h \times (w + 1) \times d$, ảnh sau khi chèn đường Seam sẽ khôi phục 1 pixels hình ảnh theo chiều ngang
- Áp đường Seam cần chèn lên hình ảnh, các pixel nằm bên trái và bên phải đường Seam sẽ giữ nguyên giá trị trong ảnh kết quả, các giá trị nằm trên đường Seam sẽ là giá trị trung bình của của pixel bên trái và pixel bên phải nó.

Thao tác chèn đường Seam

```
tmp[i, :seam[i], :] = img[i, :seam[i], :]  
tmp[i, seam[i]+1:, :] = img[i, seam[i]:, :]  
if seam[i] == 0:  
    tmp[i, seam[i], :] = img[i, seam[i]+1, :]  
elif seam[i] == w-1:  
    tmp[i, seam[i], :] = img[i, seam[i]-1, :]  
else:  
    tmp[i, seam[i], :] = (img[i, seam[i]-1, :] + img[i, seam[i]+1, :]) / 2
```

tmp: Ảnh kết quả

Hình 5.3.12. Thuật toán thực hiện thao tác chèn đường Seam

Minh họa thuật toán chèn đường Seam

11	3	21	2	4	2	2	23	5
2	0	23	3	7	32	21	2	23
4	57	4	1	88	1	41	3	21
2	32	2	42	9	23	11	5	4
4	4	1	3	2	21	5	4	12
4	1	44	1	34	23	15	3	31
12	34	2	22	53	1	15	6	23
32	3	5	22	22	31	1	54	13
13	17	2	41	11	3	5	36	1
4	4	5	21	23	43	35	5	31

0
1
2
2
1
2
3
4
5
5

(a)

(b)

11	11	3	21	2	4	2	2	23	5
2	12.5	23	3	7	32	21	2	23	3
4	57	29	1	88	1	41	3	21	23
2	32	21	42	9	23	11	5	4	2
4	2.5	1	3	2	21	5	4	12	3
4	1	2	1	34	23	15	3	31	23
12	34	2	27.5	53	1	15	6	23	2
32	3	5	22	26.5	31	1	54	13	32
13	17	2	41	11	8	5	36	1	3
4	4	5	21	23	29	35	5	31	2

(c)

Hình 5.3.13. Minh họa thao tác chèn đường Seam

- (a) Hình ảnh ban đầu
- (b) Mảng một chiều chứa các index của các pixel nằm trên đường Seam cần chèn
- (c) Chèn đường Seam vào hình ảnh

CHƯƠNG 06. CÀI ĐẶT VÀ THỰC NGHIỆM







Các hàm cài đặt thực nghiệm			
STT	Tên hàm	Giá trị trả về	Mô tả
1	<code>genEnergyMap(img)</code>	Bản đồ năng lượng	Xây dựng bản đồ năng lượng từ hình ảnh đầu vào
2	<code>genSeamMap(energyMap)</code>	Bản đồ Seam	Xây dựng bản đồ Seam từ bản đồ năng lượng áp dụng kỹ thuật quy hoạch động
3	<code>getSeamLine(energyMap)</code>	Đường Seam	Từ bản đồ năng lượng, tạo bản đồ Seam, xây dựng đường Seam có mức năng lượng nhỏ nhất bằng cách truy xuất từ dưới lên trong bản đồ Seam
4	<code>removeSeamLineImg(img, seam)</code>	Hình ảnh xóa 1 đường Seam	Áp đường Seam lên hình ảnh, các giá trị nằm bên trái đường Seam giữ nguyên vị trí, các giá trị nằm bên phải đẩy qua trái 1-pixel, kích thước hình ảnh sau khi xóa nhỏ hơn ảnh ban đầu 1-pixel theo chiều ngang
5	<code>removeSeamLineMask(mask, seam)</code>	Mask xóa 1 đường Seam	Áp đường Seam lên ảnh mask, các giá trị nằm bên trái đường Seam giữ nguyên vị trí, các giá trị nằm bên phải đẩy qua trái 1-pixel, kích thước ảnh mask sau

			khi xóa nhỏ hơn ảnh mask ban đầu 1-pixel theo chiều ngang
6	<code>removeObjectfromMask(img, objectMaskProtect, objectMaskDelete)</code>	2 giá trị trả về: Hình ảnh sau khi xóa đối tượng và mask bảo vệ đối tượng (hỗ trợ cho thao tác khôi phục ảnh)	Tính số lượng đường Seam cần xóa (n), tạo vòng lặp n lần, mỗi bước lặp áp ảnh mask cho đối tượng cần xóa và bảo vệ lên hình ảnh, thiết lập giá trị cực đại và cực tiểu tương ứng với mask trên bản đồ năng lượng, xây dựng bản đồ Seam từ bản đồ năng lượng đó và từng bước xóa các đường Seam. Lặp cho đến khi đủ số lượng đường Seam cần xóa.
7	<code>insertSeam(img, seam)</code>	Hình ảnh chèn 1 đường Seam	Áp đường Seam lên hình ảnh, các giá trị nằm bên trái và bên phải đường Seam giữ nguyên vị trí, các giá trị nằm trên đường Seam được tính bằng trung bình cộng của 2-pixel bên cạnh, kích thước hình ảnh sau khi chèn lớn hơn ảnh ban đầu 1-pixel theo chiều ngang
10	<code>enlargeImage(img, size, maskInsert)</code>	Hình ảnh kết quả sau khi đã xóa đối tượng và khôi phục ảnh	Từ hình ảnh đã xóa đối tượng, tiếp tục xóa thêm n đường Seam và lưu trữ các đường Seam cần xóa vào 1 buffer. Chèn các đường Seam trogn buffer vào hình ảnh đã xóa đối tượng, được hình ảnh kết quả đã xóa đối tượng và kích thước được khôi phục ban đầu
	<code>shiftSeam(bufferSeam, seam)</code>	Buffer chứa các đường Seam	Cập nhật các vị trí các index trong đường Seam vì khi chèn

			đường Seam theo thứ tự sẽ xảy ra lệch index
12	<code>visualizeSeam(img, seam, color=COLOR)</code>	Hình ảnh trực quan 1 đường Seam	Hiển thị màu cho các pixels nằm trên đường Seam
11	<code>drawPractice(img, num=100)</code>	Hình ảnh hiển thị các đường seam	Hiển thị màu có các pixels nằm trên các đường Seam. Đối số <code>num</code> thể hiện số đường Seam được trực quan
13	<code>produceVideo(videoPath, originalShape)</code>	Ảnh GIF	Ghi lại quá trình xóa và khôi phục hình ảnh

CHƯƠNG 07. KẾT QUẢ VÀ DEMO

- Link Demo thực nghiệm: [Removing Objects using Seam Carving.mp4 - Google Drive](#)
- File GIF demo: [Thư mục - Google Drive](#)
- Source Code: [HiImKing1509/Removing-Objects-using-Seam-Carving \(github.com\)](https://github.com/HiImKing1509/Removing-Objects-using-Seam-Carving)

Input	Kích thước	Output	Time
	669 × 446		Time Remove: 8.78s Time Revert: 10.8s
	612 × 409		Time Remove: 15.36s Time Revert: 19.28s
	757 × 506		Time Remove: 15.77s Time Revert: 21.9s

CHƯƠNG 08. KẾT LUẬN

08.01 Ưu điểm

Thao tác lựa chọn đối tượng cho phép tương tác người dùng, sử dụng chuột như một cây bút vẽ dùng tô vùng đối tượng cần chọn và có thể điều chỉnh kích thước bút phù hợp với kích thước đối tượng. Điều này mang lại sự dễ dàng sử dụng cho người dùng.

Trong các trường hợp hình ảnh đầu vào và cách chọn đối tượng xóa, đối tượng bảo vệ thỏa mãn ràng buộc, thuật toán Seam Carving giúp xóa đối tượng tương đối chính xác và quá trình khôi phục hình ảnh cũng không làm bất thường hình ảnh quá mức.

08.02 Hạn chế

Còn nhiều hạn chế trong phương pháp tiếp cận:

- Thao tác lựa chọn đối tượng bằng chuột cho phép tô vùng đối tượng. Tuy nhiên, nếu tô nhầm hoặc lệch con trỏ chuột, không có thao tác quay lại hành động trước đó, và chấp nhận vùng đối tượng bị lệch hoặc chạy lại chương trình.
- Cách lấy mask đối tượng sử dụng 2 vòng lặp **for** vẫn chưa tối ưu
- Nếu đối tượng cần xóa có bề ngang quá rộng, việc xóa đối tượng tốn khá nhiều thời gian
- Nếu đối tượng cần xóa có kích thước chiều cao nhỏ hơn chiều rộng, việc xóa đối tượng theo chiều cao (các đường Seam nằm ngang đi từ trái sang phải) sẽ giúp cải thiện đáng kể thời gian, nhưng phần cài đặt chưa hỗ trợ thực hiện xóa các đường Seam theo chiều cao.
- Ở thao tác khôi phục hình ảnh, nếu kích thước đối tượng cần xóa vượt quá 50% kích thước ảnh gốc theo chiều ngang, bài toán sẽ không thực hiện thành công
- Cách lấy mask như đã thực hiện trên có xảy ra vấn đề: Bút tô có giá trị (0,0,0) màu đen và tại đó giá trị sẽ đặt thành (255,255,255) màu trắng. Tuy nhiên, giả sử trong hình ảnh gốc có sẵn một vài pixels có giá trị (0,0,0) thì vô tình làm kích thước hình ảnh tăng thêm 1 vài pixels, do đó sau khi xóa hoàn toàn đối tượng, hình ảnh sẽ có xu hướng xóa thêm 1 vài đường Seam nữa, và điều này là không cần thiết.

CHƯƠNG 09. HƯỚNG PHÁT TRIỂN

Một trong những hướng phát triển được hướng tới đầu tiên là cho phép xóa đối tượng theo chiều cao (xóa các đường Seam từ trái sang phải) để giảm thiểu thời gian tính toán trong một vài trường hợp đặc biệt.

Tối ưu hóa trải nghiệm người dùng trong thao tác lấy mask đối tượng.

Thực hiện các bước điều chỉnh để có thuật toán thực hiện chính xác và đạt hiệu quả cao hơn.

CHƯƠNG 10. TÀI LIỆU THAM KHẢO

GitHub - vivianhylee/seam-carving: Seam Carving for Content Aware Image Resizing. (2022). Retrieved 3 June 2022, from <https://github.com/vivianhylee/seam-carving>

Seam carving - Wikipedia. (2022). Retrieved 3 June 2022, from https://en.wikipedia.org/wiki/Seam_carving

Avidan, S., & Shamir, A. (2007). Seam carving for content-aware image resizing. In ACM SIGGRAPH 2007 papers (pp. 10-es).

Seam Carving Algorithm. (2022). Retrieved 3 June 2022, from <https://vnoi.info/wiki/cs/imageprocessing/Seam-Carving.md>

CHƯƠNG 11. BẢNG PHÂN CÔNG CÔNG VIỆC

Tên	Nguyễn Đức Anh Phúc	Huỳnh Viết Tuấn Kiệt	Phan Hoàng Ân
MSSV	20520276	20521494	16520017
Mức độ tích cực	★★★★★	★★★★★	★★★★★
Phân chia công việc	✓		
Đọc hiểu bài toán	✓	✓	✓
Tìm hiểu bài toán	✓	✓	✓
Tìm kiếm tài liệu	✓	✓	✓
Xây dựng ý tưởng	✓	✓	✓
Cài đặt chương trình	✓	✓	✓
Kiểm thử chương trình	✓	✓	✓
Viết báo cáo	✓	✓	
Thuyết trình	✓	✓	✓
Đánh giá hoàn thành	100%	100%	90%

CHƯƠNG 12. LỜI CẢM ƠN

Cảm ơn trường Đại học Công Nghệ Thông tin đã tạo điều kiện đăng kí môn học Nhập môn Thị giác máy tính (CS231). Cảm ơn TS. Nguyễn Vinh Tiệp đã tận tình hướng dẫn thực hiện đồ án môn học.