

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
MÔN HỌC: NHẬP MÔN THỊ GIÁC MÁY TÍNH

Đề tài:

High Lightning Effect

Lớp: CS231.N11

Giảng viên hướng dẫn: TS. Nguyễn Vinh Tiệp

Sinh viên thực hiện: Nguyễn Minh Tiến

Nhóm sinh viên:

Nguyễn Minh Tiến	20522010
Tạ Nhật Minh	20521614
Nguyễn Hoàng Minh	20521609
Nguyễn Thiện Thuật	20521998

Thành phố Hồ Chí Minh, ngày 26 tháng 1 năm 2023

MỤC LỤC

Table of Contents

<i>I. Tổng quan bài toán.....</i>	<i>3</i>
1. Giới thiệu	3
2. Thử thách và mục tiêu đồ án	4
2.1. Mục tiêu	4
2.2. Thử thách	4
<i>II. Cách tiếp cận bài toán.....</i>	<i>5</i>
1. Hướng tiếp cận.....	5
2. Xử lý chi tiết.....	6
2.1. Xác định các điểm trên bàn tay	6
2.2. Biến đổi Afin.....	7
2.3. Transparent Image.....	13
2.4. Các effect đã thực hiện	14
2.5 Phương pháp thực hiện chung	18
2.6 Source code python và demo	20
<i>III. Kết quả thực nghiệm.....</i>	<i>20</i>
1. Danh sách các effect.....	20
2. Đánh giá mức độ hoàn thiện của các effect	21
<i>IV. Kết luận.....</i>	<i>21</i>
1. Nhận xét	21
1.1. Kết quả đạt được	21
1.2. Hạn chế.....	21
2. Hướng phát triển	22
<i>V. Tài liệu tham khảo.....</i>	<i>22</i>

I. Tổng quan bài toán

1. Giới thiệu

Trong thời đại phát triển của công nghệ thì các trang mạng xã hội đang trở nên phổ biến hơn bao giờ hết. Tiktok, Facebook, Instagram và nhiều trang khác đang trở thành nguồn giải trí hàng ngày cho người dùng. Tuy nhiên, để tăng hứng thú và cải thiện chất lượng hình ảnh thì các hiệu ứng cũng dần được ra đời nhiều hơn, đồng thời các hiệu ứng này cũng là một trong những kỹ thuật phổ biến được sử dụng trong các ngành điện ảnh, trang mạng xã hội và các game thực tế ảo. Ở bài báo cáo này, tôi sẽ giới thiệu những kỹ thuật được sử dụng để tạo ra các hiệu ứng ánh sáng tối thượng (High Lightning Effect) trên bàn tay giúp cho hình ảnh trở nên sắc nét hơn và tăng độ hứng thú cho người xem.

Trong bài báo cáo này, tôi sẽ sử dụng hands được lấy ra từ thư viện Mediapipe để theo dõi các điểm đặc trưng của tay trong thời gian thực và sử dụng chúng để kiểm soát việc áp dụng các hiệu ứng trên tay như thêm quả cầu ảo hoặc tạo ra khung cảnh giật sét. Ngoài ra, chúng tôi còn sử dụng công nghệ như OpenCV để phát hiện các điểm đặc trưng của tay, lấy phần foreground của ảnh và GIF để áp dụng các hiệu ứng trên tay. Đồ án này cho thấy được machine learning và xử lý hình ảnh có thể được sử dụng để theo dõi cử chỉ và các hiệu ứng có trên tay để từ đó có thể có nhiều ứng dụng trong lĩnh vực như thực tế ảo, trò chơi và giao tiếp người máy hoặc làm nhiều đoạn phim với hình ảnh hay hơn, kỹ xảo hơn.

2. Thử thách và mục tiêu đề án

2.1. Mục tiêu

Tìm hiểu về hướng tiếp cận và các thuật toán để xử lý bài toán “High Lightning Effect” bằng cách sử dụng thư viện Mediapipe để trích xuất các điểm trên bàn tay và sử dụng thư viện OpenCV để tạo hiệu ứng từ việc xóa nền hình ảnh và GIF, cuối cùng là áp dụng các hiệu ứng làm được lên bàn tay. Bên cạnh đó, mục tiêu cụ thể hơn là xây dựng một hệ thống hoặc ứng dụng cho phép người dùng tạo hiệu ứng trên bàn tay của họ dựa trên các điểm trích xuất được để từ đó phục vụ cho nhu cầu của người dùng.

2.2. Thử thách

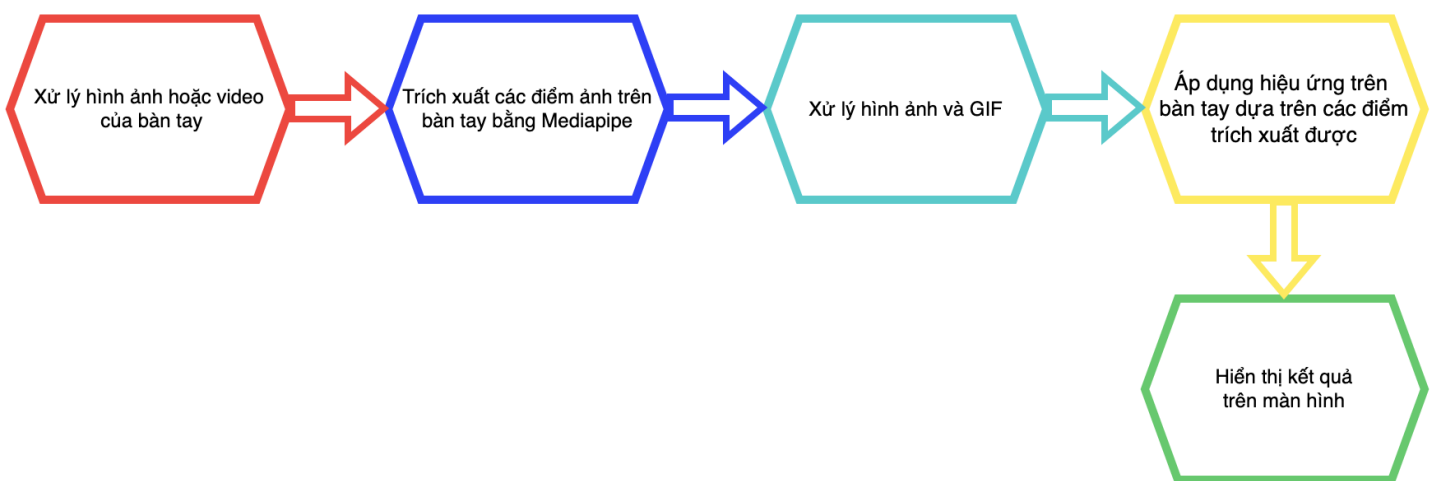
Bên cạnh những mặt tích cực mà bài toán hiệu ứng ánh sáng tối thượng đã mang lại thì cũng gặp không ít khó khăn thử thách. Đầu tiên là độ chính xác của việc trích xuất điểm trên bàn tay, việc trích xuất điểm có thể khó khăn vì bàn tay có thể có nhiều hình dạng khác nhau và có thể di chuyển hoặc xoay trong quá trình ghi hình qua camera. Tiếp đến, tính tương thích của hệ thống với nhiều loại hình ảnh và GIF khác nhau gây ảnh hưởng đến hiệu suất và tốc độ xử lý của hệ thống, tính minh bạch và cách sử dụng của hệ thống đối với người dùng. Cuối cùng, các tham số của bài toán phải được thực hiện nhiều lần để chọn ra các giá trị phù hợp cho từng thao tác tay tạo ra hiệu ứng.

Ngoài ra còn các yếu tố khách quan như môi trường, độ sáng, khung cảnh xung quanh cũng làm ảnh hưởng không nhỏ đến kết quả của bài toán.

II. Cách tiếp cận bài toán

1. Hướng tiếp cận

Đối với bài toán tạo hiệu ứng ánh sáng tối thượng từ việc trích xuất các điểm ảnh trên bàn tay có nhiều phương án tiếp cận khác nhau bao gồm từ phức tạp như sử dụng các phương pháp học sâu để nhận dạng thao tác với ảnh (deep learning) cho đến cơ bản như ứng dụng các kỹ thuật xử lý trên ảnh. Trong báo cáo này, nhóm đã xây dựng nhiều thuật toán tạo ra các hiệu ứng khác nhau thông qua thư viện hỗ trợ sẵn trên Python là OpenCV. Hướng tiếp cận các kỹ thuật sử dụng theo trình tự sau:



Hình 1. Trình tự các kỹ thuật xử lý

Để bắt đầu, nhóm chúng em cần lấy hình ảnh của tay người dùng sử dụng một camera hoặc thiết bị quét và xử lý chúng để loại bỏ nhiễu và tăng độ chính xác. Tiếp theo, việc cần làm là sử dụng thư viện Mediapipe để nhận dạng bàn tay và tìm ra các điểm trên tay của người dùng. Sau khi có điểm trên tay người dùng, ta có thể sử dụng chúng kết hợp với các ảnh hoặc GIF đã được xóa nền để tạo ra các hiệu ứng ánh sáng tối thượng nằm trên bàn tay hay các hiệu ứng giật sét. Cuối cùng, ta cần kiểm tra kết quả và

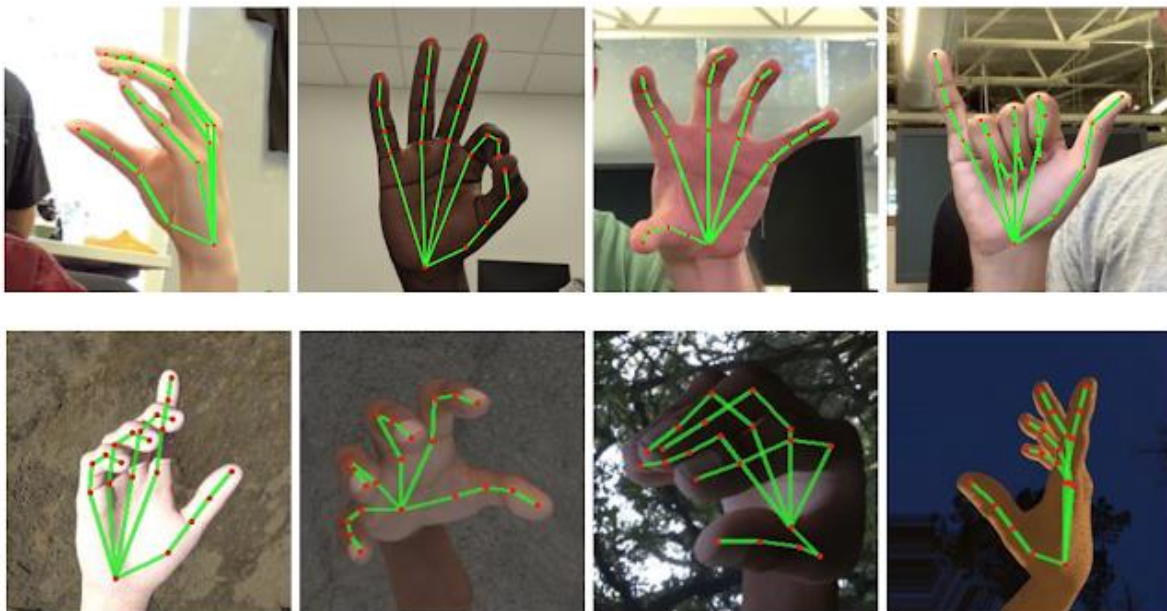
điều chỉnh kỹ thuật nếu cần thiết để đạt được kết quả mong muốn và hiển thị ra trên màn hình.

2. Xử lý chi tiết

2.1. Xác định các điểm trên bàn tay

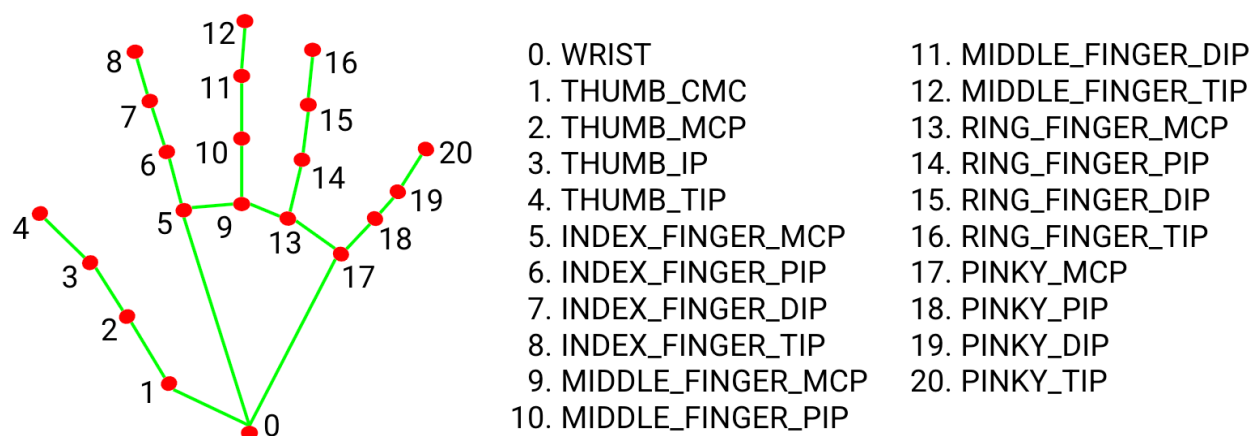
MediaPipe là một nền tảng mã nguồn mở cho việc phát hiện và chuyển đổi đặc trưng, cung cấp các thành phần có sẵn để xử lý dữ liệu video, hình ảnh và âm thanh. MediaPipe Handpose là một gói thành phần của MediaPipe cho phép phát hiện và trích xuất các điểm có trên bàn tay từ dữ liệu video hoặc hình ảnh.

Phát hiện bàn tay trong MediaPipe Handpose được thực hiện bằng mạng neural tự động, sử dụng mô hình chuyển đổi đặc trưng (CPM) để tìm và xác định vị trí của bàn tay trong hình ảnh. Sau khi phát hiện được vị trí của bàn tay, một mô hình đặc trưng cấp cao sẽ được sử dụng để trích xuất các điểm có trên bàn tay.



Trích xuất các điểm có trên bàn tay trong MediaPipe Handpose được thực hiện bằng một mô hình chuyên đổi đặc trưng (CPM) được huấn luyện trên tập dữ liệu các bàn tay 2D đã được sắp xếp. Mô hình này được sử dụng để tìm các điểm có trên bàn tay, bao gồm các điểm trên ngón tay, các điểm trên các khu vực khác của bàn tay, như đầu và cuối của các ngón tay, các điểm trên đốt sừng và các điểm trên các khu vực khác của bàn tay.

Trích xuất tay và các điểm trên bàn tay là một trong những kỹ thuật mà thị giác máy tính đang được sử dụng rộng rãi trong các ứng dụng nhận dạng về cử chỉ tay và các thao tác thực tế ảo. Thư viện Mediapipe là một trong những thư viện phổ biến được sử dụng nhiều trong các lĩnh vực này. Mediapipe đã sử dụng học máy để xác định 21 điểm trên tay, bao gồm các ngón tay và các đốt sừng.



MediaPipe Handpose cung cấp các chức năng phát hiện và trích xuất điểm có trên bàn tay để hỗ trợ các ứng dụng như điều khiển thiết bị bằng bàn tay, tương tác người-máy và nhiều ứng dụng khác.

2.2. Biến đổi Afin

Biến đổi Afin (Affine Transformation) là một loại biến đổi hình học được sử dụng để chuyển đổi vị trí, kích thước, góc quay và chiều dọc của một hình ảnh hoặc đối tượng.

Nó không tham gia trực tiếp trong quá trình phát hiện và trích xuất các điểm có trên bàn tay hoặc các hiệu ứng được sử dụng. Tuy nhiên, việc sử dụng biến đổi Affine trước khi phát hiện và trích xuất các điểm có trên bàn tay có thể giúp tăng tính chính xác của hệ thống, đặc biệt là trong trường hợp hình ảnh có kích thước, góc quay hoặc chiều dọc khác nhau.

Phép biến đổi Afin (Affine transformation) T ánh xạ tập P(x, y) sang tập Q(x', y'):

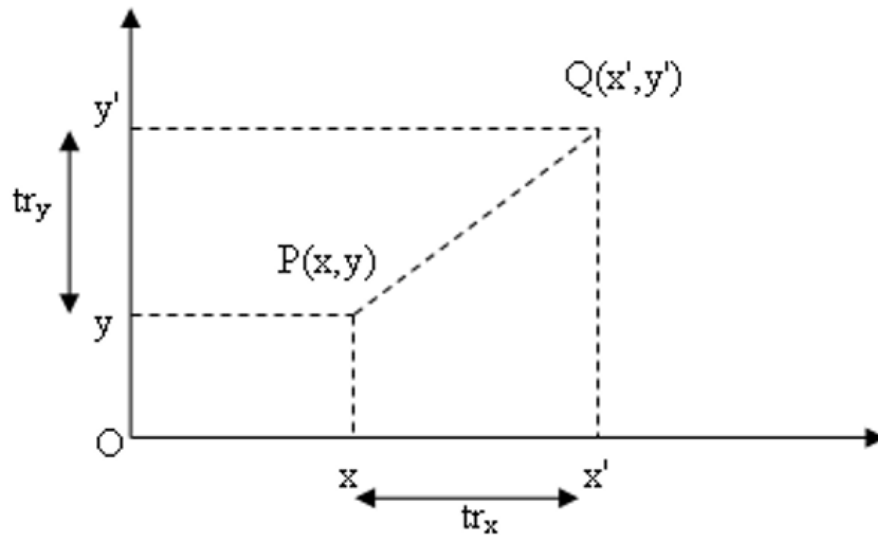
với a, b, c, d, e, f là các hệ số
$$\begin{cases} x' = ax + by + c \\ y' = dx + ey + f \end{cases}$$

Biểu diễn ma trận:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Leftrightarrow \mathbf{Q} = \mathbf{T} \cdot \mathbf{P}$$

Các phép biến đổi cơ sở:

a) Phép tịnh tiến (translation): là một trong những loại biến đổi afin được sử dụng để di chuyển một đối tượng trong một không gian hai hoặc ba chiều. Nó được thực hiện bằng cách cộng hoặc trừ một vector cho từng tọa độ của đối tượng. Ví dụ, nếu muốn di chuyển một đối tượng sang phải 10 đơn vị và 5 đơn vị lên trên, ta sẽ cộng một vector (10, 5) vào từng tọa độ của đối tượng. Trong ma trận affine thì phép tịnh tiến được thể hiện bằng cách cộng một ma trận 2x3 vào ma trận afin ban đầu.



Gọi $tr = (tr_x, tr_y)$ là vector tịnh tiến từ điểm P đến điểm Q thì:

$$\begin{cases} x' = x + tr_x \\ y' = y + tr_y \end{cases}$$

Ma trận biến đổi của phép tịnh tiến.

$$T(tr_x, tr_y) = \begin{pmatrix} 1 & 0 & tr_x \\ 0 & 1 & tr_y \\ 0 & 0 & 1 \end{pmatrix}$$

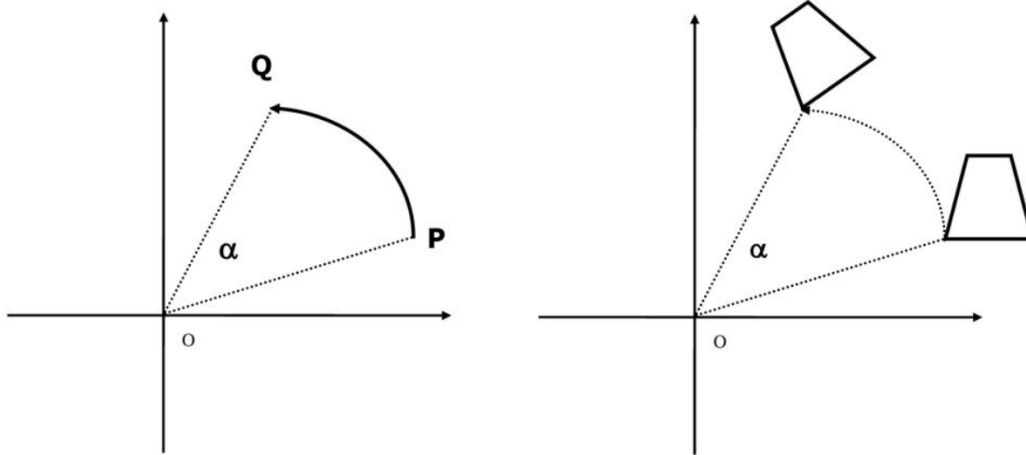
b) Phép quay (rotation): là một phép biến đổi hình học có tác dụng xoay một đối tượng quanh trục hoặc một điểm cố định. Trong xử lý hình ảnh, phép quay có thể được sử dụng để chỉnh sửa hình ảnh hoặc tìm kiếm một đối tượng trong hình ảnh. Phép quay có thể được thực hiện bằng cách sử dụng ma trận biến đổi hoặc bằng cách sử dụng thuật toán interpolation.

Phép quay trong Mediapipe có thể thực hiện bằng cách sử dụng các node quay và phóng to/thu nhỏ trong pipeline. Các node này có thể được sử dụng để quay hình ảnh hoặc để tìm kiếm một đối tượng trong hình ảnh. Bạn có thể sử dụng các node này để giải quyết vấn đề của phép quay trong bài toán xử lý phát hiện bàn tay và điểm trên bàn tay.

$$\begin{cases} x' = \cos\alpha x - \sin\alpha y \\ y' = \sin\alpha x + \cos\alpha y \end{cases} \Leftrightarrow T(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Với s_x và s_y được gọi là hệ số co giãn theo trục Ox và trục Oy ($s_x, s_y \leq 1$).

Phép quay một góc α quanh gốc tọa độ



c) Phép biến đổi tỉ lệ (scaling): là một dạng ma trận biến đổi hình học, được sử dụng để thay đổi kích thước của một đối tượng hình học. Nó có thể sử dụng để thay đổi kích thước theo chiều rộng, chiều cao hoặc cả hai.

Phép biến đổi tỉ lệ 2D có dạng sau:

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ 1 \end{bmatrix}$$

Scaling Matrix
(Homogeneous Coordinates Representation)

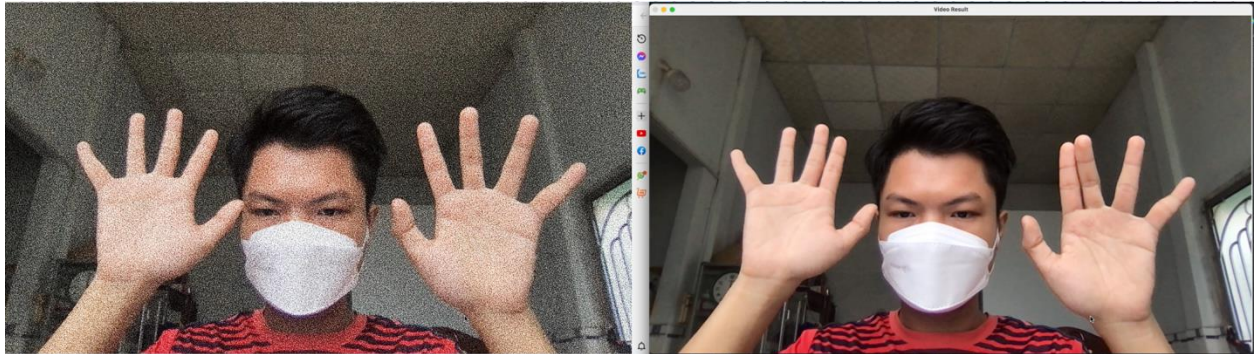
Trong đó, s_x là hệ số co giãn theo chiều rộng, s_y là hệ số co giãn theo chiều cao. Nếu s_x và s_y đều là 1 thì không có biến đổi nào xảy ra. Còn nếu s_x hoặc s_y lớn hơn 1 thì đối tượng sẽ bị phóng to, nếu s_x hoặc s_y nhỏ hơn 1 thì đối tượng sẽ bị thu nhỏ.

d) Ngoài ra, trong việc phát hiện và trích xuất điểm trên tay, có một số kỹ thuật tối ưu được sử dụng để tăng tốc độ và chính xác của thuật toán. Một số trong số chúng bao gồm:

- **Lọc nhiễu (Noise Filtering):** Sử dụng các thuật toán lọc như median filter, Gaussian filter để loại bỏ nhiễu từ hình ảnh, giúp tăng chính xác của kết quả. Sau đây là công thức toán học để lọc nhiễu:

$$K = \frac{1}{\text{Kenel}_{width} \cdot \text{Kenel}_{hieght}} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{height,width} \end{vmatrix}$$

$$outputPixel(x,y) = \sum_{xy} image(x + k_{col}, y + k_{col})K(k_{row}, k_{col})$$



- **Tối ưu hóa (Optimization):** Sử dụng các thuật toán tối ưu như gradient descent, conjugate gradient descent để tìm giá trị tối ưu của các tham số của thuật toán.
- **Tối ưu hóa độc lập (Independent Optimization):** Sử dụng các thuật toán tối ưu hoá độc lập như Nelder-Mead, Powell hoặc BFGS để tìm giá trị tối ưu của các tham số của thuật toán.
- **Tối ưu hoá theo luồng (Flow optimization):** Sử dụng các thuật toán tối ưu hoá theo luồng như L-BFGS, Rprop hoặc Adam để tìm giá trị tối ưu của các tham số của thuật toán.
- **Ứng dụng các kết quả:** sau khi trích xuất được các điểm trên tay, các kết quả được sử dụng cho các ứng dụng khác nhau như nhận dạng cử chỉ tay, thực tế ảo và thực tế tăng cường.

2.3. Transparent Image

Trong thị giác máy tính, hình ảnh trong suốt (Transparent Image) có thể được xem là một tấm hình có một phần hoặc toàn bộ đã được xóa hoặc "trong suốt" và có thể được sử dụng để chèn hiệu ứng của bất kỳ tấm hình khác vào.

Trong suốt (Transparency) có thể xem là một "phần mở rộng" của một tấm hình. Nó chỉ được áp dụng trong một số loại tập tin như GIF, PNG, TIFF... bằng cách sử dụng màu trong suốt hoặc sử dụng kênh alpha (Alpha Compositing).

Trong thị giác máy tính, tính năng trong suốt (Transparency) có thể được áp dụng trên cả hình ảnh raster (digital image) và vector image. Tuy nhiên, trong bài báo cáo này, chúng ta sẽ chỉ tập trung vào việc sử dụng trong suốt trên hình ảnh raster, cụ thể hơn là hình ảnh kỹ thuật số. Mỗi pixel trong hình ảnh được biểu diễn trong một hệ màu, sẽ có một giá trị được gọi là "giá trị trong suốt" và nó không phải là một màu trong thực tế. Nếu bộ giải mã màu gặp được pixel với giá trị này, nó sẽ được hiển thị dưới dạng màu nền của phần màn hình mà hình ảnh được đặt trên. Để dễ hiểu, hãy tưởng tượng rằng một hình ảnh trong suốt là một tấm hình với tất cả các pixel đều có giá trị trong suốt, nó sẽ trông giống như một tấm gương và cho phép chúng ta thấy các pixel của nền mà hình ảnh được đặt trên.

Trong thị giác máy tính, có nhiều cách khác nhau để biểu diễn màu sắc trên một tấm hình. Tuy nhiên, để tạo ra một hình ảnh trong suốt, ta có thể sử dụng kênh màu RGB kết hợp với kênh Alpha (RGBA), với giá trị alpha nằm trong khoảng từ 0.0 (hoàn toàn trong suốt) đến 1.0 (không trong suốt hoàn toàn). Đây là một trong nhiều cách để tạo ra một hình ảnh trong suốt. Với mỗi hệ màu khác nhau, ta có thể thêm một kênh màu Alpha hoặc đặt giá trị cho pixel để xác định những pixel nào sẽ trở thành pixel trong suốt.

Ví dụ: Một ví dụ cụ thể của hình ảnh trong suốt đối với trường hợp trên là một hình ảnh của một logo của một công ty được đặt trên một nền màu trắng. Trong trường hợp này, logo được lưu trữ trong định dạng PNG với kênh Alpha được sử dụng để xác định những pixel nào sẽ là trong suốt, tức là những pixel nào sẽ cho phép nền trắng để

hiển thị qua logo. Khi hình ảnh được xuất ra và đặt trên một nền khác, chỉ có logo sẽ được hiển thị mà không có bất kỳ màu nền nào xuất hiện trong nó.

2.4. Các effect đã thực hiện

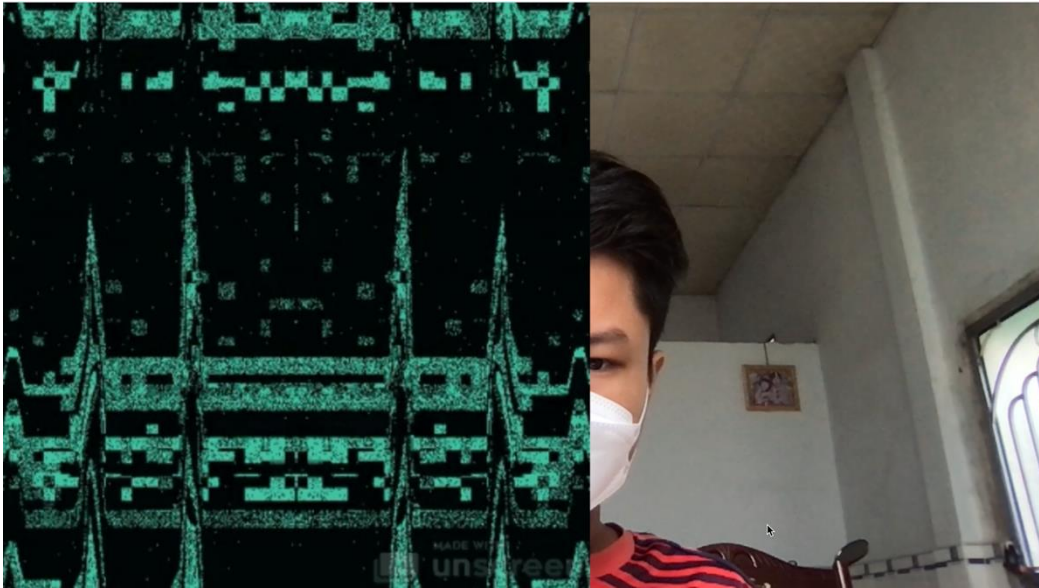
Tạo quả cầu vũ trụ & phá vỡ màn hình:

Trong bài báo cáo này, tôi sẽ dùng hình ảnh vũ trụ đã được xử lý để tạo ra hiệu ứng chính. Tiếp đến là việc sử dụng hàm `warpAffine` và `getRotationMatrix2D` với mục đích xoay hình ảnh vũ trụ nhằm tạo ra hiệu ứng quả cầu sét. Để đảm bảo hiệu ứng này xuất hiện tại vị trí thích hợp, tôi đã tính toán khoảng cách giữa đầu ngón trỏ và đầu ngón giữa và điều chỉnh vị trí hiệu ứng tương ứng. Kết quả cuối cùng là một bức ảnh vũ trụ đẹp mắt với hiệu ứng quả cầu sét được xếp chính xác giữa hai ngón tay

Dùng hình ảnh vũ trụ sau khi đã được xử lý để làm hiệu ứng chính, sau đó sử dụng hàm `warpAffine` và `getRotationMatrix2D` để làm xoay hình ảnh vũ trụ kết hợp với hiệu ứng quả cầu sét được xử lý tương tự. Tiếp đến ta sẽ tính khoảng cách giữa đầu ngón trỏ (`wrist_tip`) và đầu ngón giữa (`midle_tip`) sau đó điều chỉnh vị trí quả cầu hiệu ứng này xuất hiện ở giữa 2 ngón tay ấy.



Cuối cùng là nếu quả cầu vũ trụ tăng đến một kích thước nhất định thì nó sẽ xảy ra hiệu ứng phát nổ và làm hỏng màn hình.



Tích tụ năng lượng:

Để thi triển các kỹ năng thì việc cần làm đầu tiên đó là tích tụ năng lượng bằng cách vận sức.

Trong bài toán này, tôi đã sử dụng hai đầu ngón tay cái (thumb_tip) và ngón út (pinky_tip) để triển khai hiệu ứng tích tụ năng lượng. Thông qua việc sử dụng thuật toán, chúng tôi đã thiết lập điều kiện cho phép hiệu ứng này xuất hiện khi hai đầu ngón tay chạm nhau hoặc cách nhau một khoảng nhỏ. Để tạo ra hiệu ứng tương tác và hấp dẫn hơn, tôi đã sử dụng kết hợp của ba ảnh GIF đã được xử lý. Mỗi lần vận sức được tăng thêm 3 điểm năng lượng mỗi giây cho đến khi tích đầy 250 điểm. Lưu ý rằng khi năng lượng đã được tích đầy, không thể vận sức thêm nữa.



Giật sét:

Trong quá trình thực hiện thao tác này, mỗi giây sẽ trừ đi 1 điểm năng lượng. Để vận hành, tôi đã sử dụng hai đốt ngón giữa bao gồm đốt thứ nhất (middle_mcp) và đốt thứ hai (middle_pip) gần như trùng nhau. Khi năng lượng đã hết, sẽ cần tích thêm năng lượng để tiếp tục triển khai kỹ năng.



Tung chuông & Super Saiyan Blue:

Tung chuông & Super Saiyan Blue là kỹ năng cuối cùng và đồng thời cũng là kỹ năng phức tạp nhất.

- Trường hợp chỉ có một tay dùng chiêu:

Trong quá trình thực hiện kỹ năng, tôi đã sử dụng một phép tính toán chia lấy tỉ lệ để điều chỉnh độ rộng xòe ra của hai bàn tay từ đó tạo ra hiệu ứng chèn vào. Quả cầu sẽ xoay một lúc và sau đó phóng ra tia sáng. Mỗi giây, năng lượng sẽ bị trừ đi một đơn vị. Nếu năng lượng hết, kỹ năng sẽ không thể tiếp tục triển khai.



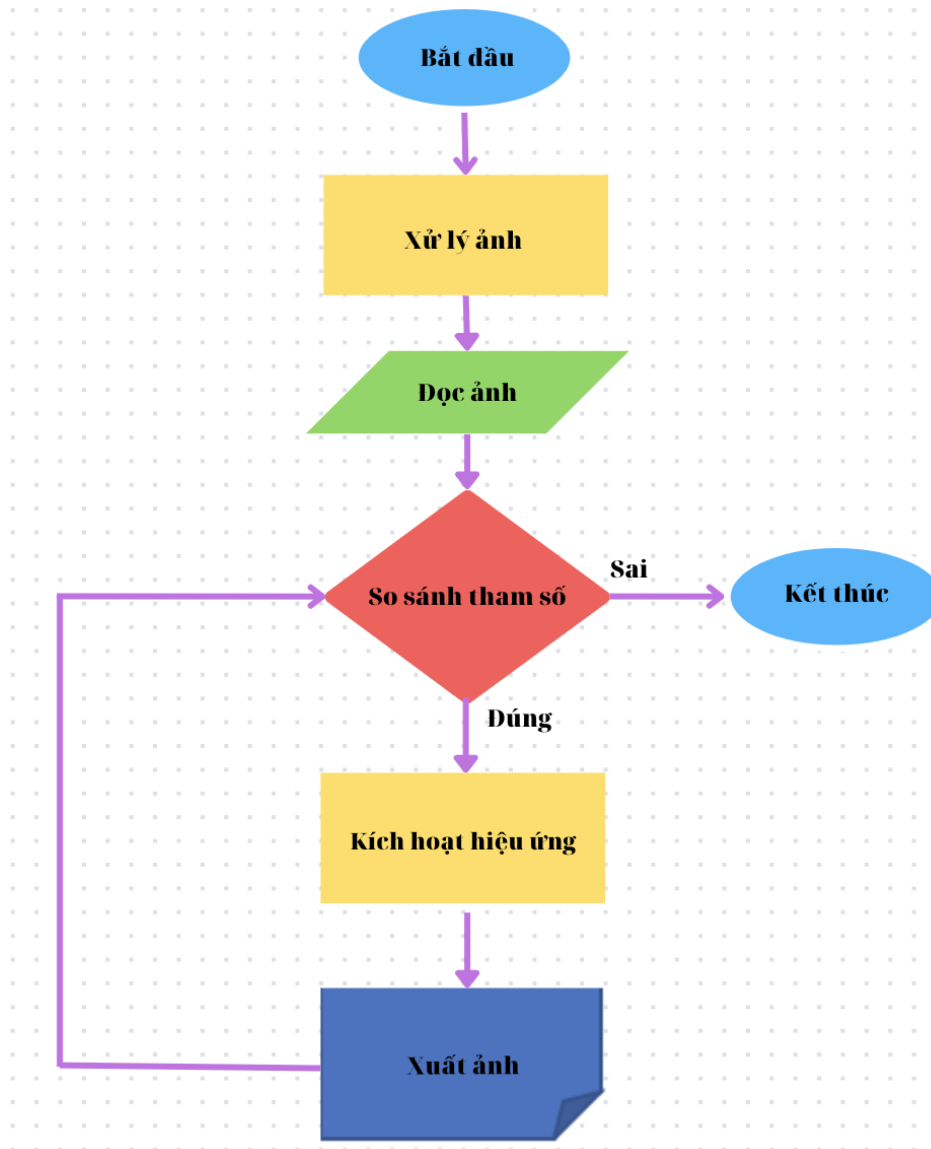
- Trường hợp có cả hai bàn tay:

Trong quá trình thực hiện kỹ năng, tôi đã sử dụng biến đếm bàn tay để kiểm tra số lượng bàn tay đang được sử dụng. Nếu biến đếm bàn tay bằng 2 nó có nghĩa là tôi đang thi triển kỹ năng trên cả hai bàn tay, quả cầu sẽ được tạo ra nằm ở giữa lòng bàn tay và bắn ra tia sáng. Ngoài ra, tôi còn thêm hiệu ứng Super Saiyan Blue để tạo ra sự bùng nổ đẹp mắt hơn. Mỗi giây, năng lượng sẽ bị trừ đi một đơn vị. Nếu hết năng lượng, kỹ năng sẽ không thể tiếp tục triển khai.



2.5 Phương pháp thực hiện chung

Bài toán trích xuất điểm trên tay là một bài toán Computer Vision được sử dụng để tìm các điểm đặc trưng trên tay. Điểm đặc trưng này có thể bao gồm các điểm như ngón tay, đỉnh ngón tay, giữa ngón tay và các điểm khác.



Các bước thực hiện phương pháp chung cho bài toán trích xuất điểm trên tay bao gồm:

Chuẩn bị dữ liệu: Lấy một tập hợp các ảnh hoặc GIF của tay với background đã được xóa. Có thể sử dụng các thuật toán như GrabCut hoặc thuật toán đơn giản hơn như chuyển đổi sang định dạng binary để loại bỏ background.

Xử lý ảnh: Sử dụng thư viện xử lý ảnh để chuyển đổi các ảnh hoặc GIF thành định dạng grayscale hoặc binary. Điều này có thể giúp giảm nhiễu và tăng tính liên tục của dữ liệu.

Trích xuất điểm: Sử dụng các thuật toán trích xuất điểm như Harris Corner Detection, SIFT hoặc SURF để tìm các điểm đặc trưng trên tay. Các thuật toán này sẽ tìm các điểm có sự thay đổi lớn trong đặc trưng của ảnh hoặc GIF, chẳng hạn như giữa ngón tay và đỉnh ngón tay. Các thuật toán này có thể yêu cầu một số tham số để điều chỉnh để đạt được kết quả tốt nhất.

Sử dụng Mediapipe: Sử dụng hàm transparent của thư viện Mediapipe để tạo ra hiệu ứng trên bàn tay, ví dụ như tạo ra một hình ảnh cận cảnh các điểm trên tay hoặc tạo ra một video chạy theo điểm trên tay.

2.6 Source code python và demo

- Link Google Drive cho Source Code python và Video Demo cho High Lightning Effect:
<https://drive.google.com/drive/folders/11b9xGJ918g27zgKMR68wIWiuSRo6Mi2M?usp=sharing>

III. Kết quả thực nghiệm

1. Danh sách các effect

STT	Tên Effect
1	Tạo quả cầu vũ trụ & phá vỡ màn hình
2	Tích tụ năng lượng
3	Giật sét
4	Tung chưởng & Super Saiyan Blue

2. Đánh giá mức độ hoàn thiện của các effect

STT	Tên Effect	Mức độ hoàn thành	Đánh giá
1	Tạo quả cầu vũ trụ & phá vỡ màn hình	100%	Tốt
2	Tích tụ năng lượng	100%	Tốt
3	Giật sét	100%	Tốt
4	Tung chuông & Super Saiyan Blue	100%	Tốt

IV. Kết luận

1. Nhận xét

1.1. Kết quả đạt được

Nhóm đã tìm hiểu và hoàn thành được những yêu cầu, quy định đã được đặt ra từ ban đầu của bài toán trích xuất điểm trên tay, tìm được các điểm chính xác trên tay người, cho phép phát triển các ứng dụng tiếp theo là sử dụng các điểm này để tạo ra các hiệu ứng trên bàn tay hoặc tiến hành phân tích thông tin.

1.2. Hạn chế

Bên cạnh những thành tựu đạt được thì nhóm gặp không ít khó khăn:

- Khó khăn trong việc loại bỏ nền: việc loại bỏ nền có thể khó khăn khi tay người có màu tương tự hoặc có nhiều đối tượng khác nhau trong hình ảnh.
- Độ chính xác của tìm kiếm điểm: độ chính xác của tìm kiếm điểm có thể bị giảm khi tay người có vết thương, có điều kiện quá tối hoặc ánh sáng khác nhau.

- Việc thiết lập các tham số để kích hoạt hiệu ứng có độ chính xác chưa cao do phải thử lặp đi lặp lại nhiều lần để lựa chọn cho phù hợp.

2. Hướng phát triển

Hướng phát triển cho bài toán trích xuất điểm trên tay có thể bao gồm:

Sử dụng các mô hình deep learning mới: Sử dụng các mô hình deep learning mới như YOLOv4 hoặc EfficientDet để tăng độ chính xác và tốc độ trong việc tìm kiếm điểm trên tay.

Sử dụng các thuật toán để xử lý tay người có vết thương: Sử dụng các thuật toán như biến đổi Fourier hoặc thuật toán deep learning để xử lý tay người có vết thương, giúp tăng độ chính xác trong việc trích xuất điểm trên tay.

Sử dụng các thuật toán để xử lý sự biến đổi do ánh sáng và góc quay: Sử dụng các thuật toán như histogram equalization hoặc thuật toán deep learning để xử lý sự biến đổi do ánh sáng và góc quay trong hình ảnh, giúp tăng độ chính xác trong việc trích xuất điểm trên tay.

Thiết kế ra các ứng dụng, nền tảng phát triển về các vấn đề trích xuất bàn tay và các hiệu ứng đi kèm để phục vụ nhu cầu của con người hoặc các vấn đề về kỹ xảo phim ảnh, các virtual game sống động,...

V. Tài liệu tham khảo

[1] Kulin Patel, “Hand-Tracking,” 18 October 2021. [Online]. Available:

<https://github.com/kulin-patel/Hand-Tracking>. [Accessed 25 January 2023].

[2] Mishu Dhar Chando, “Doctor_Strange_Magical_Circle,” 16 July 2022. [Online].

Available: https://github.com/Chando0185/Doctor_Strange_Magical_Circle. [Accessed 25 January 2023].

[3]“ Python OpenCV – Affine Transformation,” 18 January 2023. [Online]. Available: <https://www.geeksforgeeks.org/python-opencv-affine-transformation/>. [Accessed 25 January 2023].

[4]“ Affine Transformations,” [Online]. Available: https://docs.opencv.org/3.4/d4/d61/tutorial_warp_affine.html. [Accessed 25 January 2023].