**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**Môn: Nhập môn Thị giác máy tính**

**Chủ đề: Hiệu ứng trên dáng người trong video**



**Lớp: CS231.N21**

**Trần Thành 20521924**

**Trần Hoàng Anh 20521079**

**Lê Thành Hiếu 19521498**

**Phòng Lai Bảo Minh 20522217**

**Nguyễn Ích Tài 18521362**

[**Lời mở đầu** 3](#_Toc138214457)

[**Danh sách thành viên** 4](#_Toc138214458)

[**I. Hiệu ứng Darth Vader** 4](#_Toc138214459)

[**I. 1. Ý tưởng cho hiệu ứng** 4](#_Toc138214460)

[**I. 2. Nội dung và quy trình thực hiện** 4](#_Toc138214461)

[**I. 3. Xử lí trigger list** 5](#_Toc138214462)

[**I. 4. Xử lí hiệu ứng** 7](#_Toc138214463)

[**I. 4. 1. Hiệu ứng xuất hiện và biến mất** 7](#_Toc138214464)

[**I. 4. 2. Cố định vật thể vào các landmark** 8](#_Toc138214465)

[**I. 4. 3. Các điều khác** 9](#_Toc138214466)

[**I. 5. Tổng hợp xử lí** 9](#_Toc138214467)

[**II. ABC** 11](#_Toc138214468)

[**III. ABC** 11](#_Toc138214469)

[**IV. ABC** 11](#_Toc138214470)

[**V. Hiệu ứng cổ trang** 11](#_Toc138214471)

[**V. 1. Ý tưởng cho hiệu ứng** 11](#_Toc138214472)

[**V.2. Nội dung** 11](#_Toc138214473)

[**V.3. Quy trình thực hiện** 11](#_Toc138214474)

[**V.4. Xác định vị trí các điểm** 13](#_Toc138214475)

[**V.5. Mặt nạ phân mảnh** 13](#_Toc138214476)

[**V.6. Vấn đề khi trộn ảnh (blending)** 14](#_Toc138214477)

[**V.6.1 Ảnh RGBA** 15](#_Toc138214478)

[**V.7. Xử lý di chuyển** 16](#_Toc138214479)

[**V.8. Tạo ảnh động** 16](#_Toc138214480)

[**V.9. Xử lý ngắt quãng** 17](#_Toc138214481)

[**VI. Tài liệu tham khảo** 18](#_Toc138214482)

Nội dung

# **Lời mở đầu**

Abc

# **Danh sách thành viên**

Những thành viên trong nhóm bao gồm 5 người. Mỗi hiệu ứng do mỗi thành viên đóng góp.

|  |  |  |
| --- | --- | --- |
| **Thông tin thành viên** | **MSSV** | **Hiệu ứng xây dựng** |
| 1. Trần Hoàng Anh | 20521079 | I. Hiệu ứng Darth Vader |
| 2. |  |  |
| 3. |  |  |
| 4.Nguyễn Ích Tài | 18521362 | IV.Tách viền trong video thực |
| 5. Trần Thành | 20521924 | V. Hiệu ứng cổ trang |

# **I. Hiệu ứng Darth Vader**

## **I. 1. Ý tưởng cho hiệu ứng**

Với niềm yêu thích đối với bộ phim Star Wars, đặc biệt là với nhân vật Darth Vader, và niềm yêu thích được hóa thân vào nhân vật. Tuy nhiên, việc chỉnh sửa video bằng các phần mềm chuyên dụng đôi lúc cũng sẽ tốn nhiều thời gian và công sức. Nên em sử dụng kiến thức và hiểu biết về lập trình của mình để có thể trực tiếp thực hiện niềm yêu thích này.

## **I. 2. Nội dung và quy trình thực hiện**

- Xây dựng một hiệu ứng xử lí trên video, chất lượng video đầu vào 1920x1080, video màu RGB, 30fps.

- Nội dung xử lí tập trung vào hành động (lấy kiếm, chạm tay, giậm chân) của nhân vật trong video, để tạo ra các hiệu ứng (xuất hiện kiếm, mặt nạ và xóa bỏ hiệu ứng).

- Sử dụng thư viện mediapipe để hỗ trợ phát hiện dáng người thông qua pose landmark detection.

- Chi tiết các công đoạn, bao gồm:

+ Xử lí trên động tác tay, chân của nhân vật trong video để rút trích được các frame kích hoạt hiệu ứng (trigger list).

+ Xử lí các trigger list.

+ Xử lí tạo hiệu ứng động dựa trên các trigger list đã được xử lí.

- Video kết quả có thể linh hoạt thay đổi, có thể cùng những thông số như video đầu vào: video màu RGB, chất lượng 1020x1080, 30fps.

Sau đây, chúng ta cùng đi qua chi tiết các vẫn đề và cách thức xử lí các vấn đề trong quá trình thực hiện.

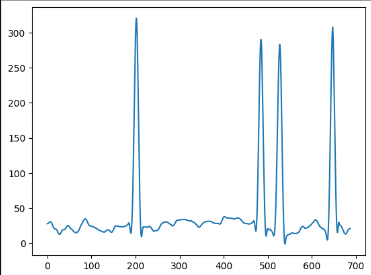
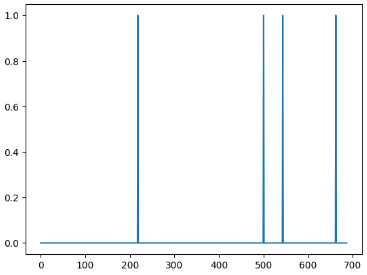
## **I. 3. Xử lí trigger list**

Việc xử lí riêng biệt tác vụ xác định thời gian thực hiện các hiệu ứng, ta có thể khiến video có hiệu ứng được diễn ra một cách mượt mà hơn, không bị đứt gãy, tạo cảm giác khó chịu cho người xem.

Để thực hiện quá trình này, ta đọc video đầu vào dưới dạng từng frame, sau đó từng frame sẽ được đưa qua Mediapipe để rút trích ra được các thông tin về landmark, và sẽ được tinh chỉnh và chuẩn hóa về kích thước video, rút trích được list chứa 33 vị trí landmark của từng frame trong video.

Từ những thông tin được trích xuất bên trên, duyệt qua từng frame và chạy qua các hàm trigger hiệu ứng (như isClose dùng để kiểm tra xem 2 landmark có đủ gần nhau hay không, có trả về 1, 0 cho ngược lại). Từ đó tạo ra được các trigger list riêng cho từng hành động như:

* lấy kiếm: khi landmark tay và bên trên vai đủ gần
* chạm tay: khi landmark bàn tay trái và phải đủ gần
* giậm chân: sử dụng hàm get\_trigger\_stomping, để xử lí dữ liệu độ chênh lệch theo chiều dọc giữa 2 bàn chân trái và phải. Với các tham số là min\_delta (độ chênh lệch ít nhất khi bắt đầu giậm chân xuống trong 1 window xử lí), và window\_size (giới hạn số lượng frame xử lí để được tính là một lần giậm chân, mang ý nghĩa là tốc độ giậm chân)

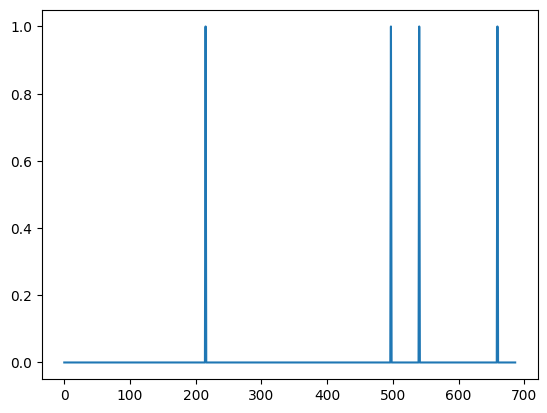
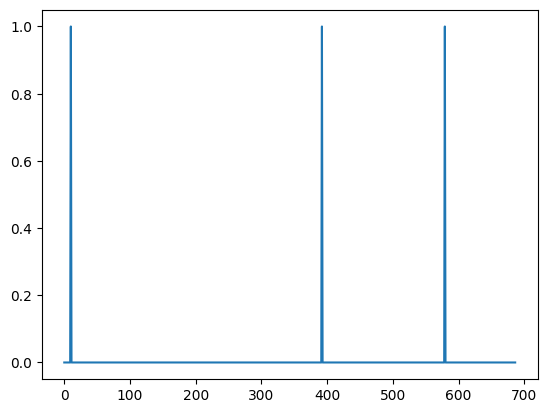
 

Chênh lệch theo Sau khi xử lí qua hàm

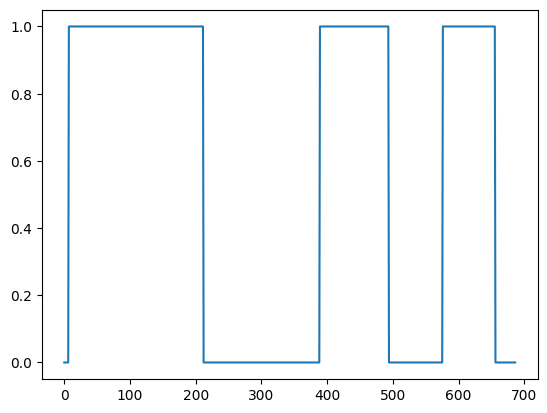
chiều dọc giữa 2 chân get\_trigger\_stomping

Khi trích xuất được 3 loại trigger list là trigger\_grab (lấy kiếm), trigger\_cross (chạm tay), và trigger\_stomp (chậm chân). Thì ta tiến hành xử lí cho từng cặp trigger\_cross – trigger\_stomp và trigger\_grab – trigger\_stomp. Mục đích của việc này nhằm xác định được chính xác khi nào hiệu ứng sẽ được xử lí, và khi nào không:

* Với cặp trigger\_cross – trigger\_stomp: Thực hiện việc gán 1 liên tục kể từ khi trigger\_cross xuất hiện 1 (chạm tay), gán 0 liên tục khi trigger\_stomp xuất hiện 1 (giậm chân để hủy bỏ hiệu ứng).

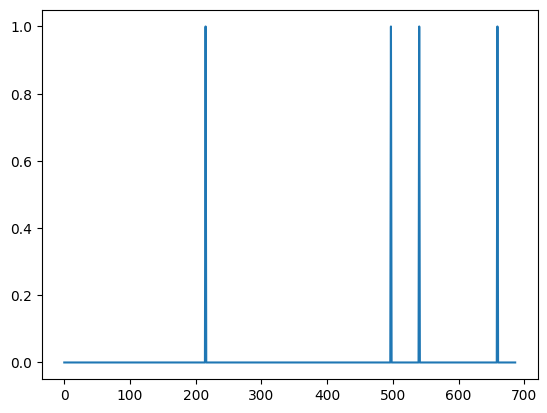
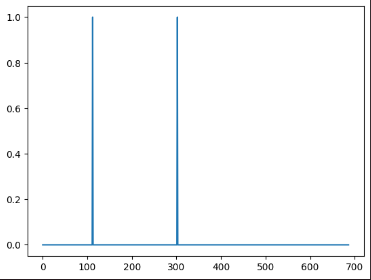


Trigger\_cross Trigger\_stomp

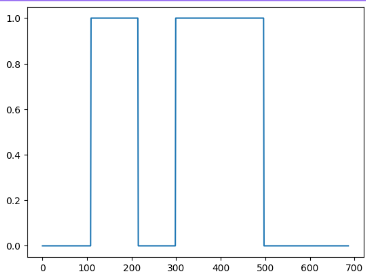


Kết hợp trigger\_cross và trigger\_stomp

* Tương tự cho cặp trigger\_grab và trigger\_stomp.



Trigger\_grab Trigger\_stomp



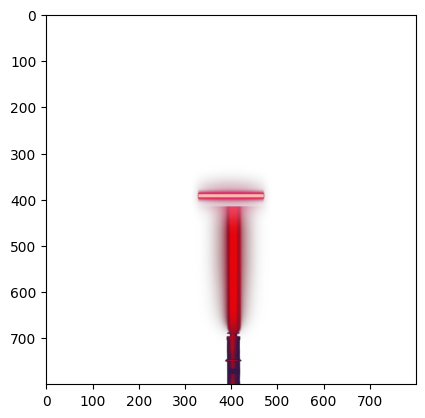
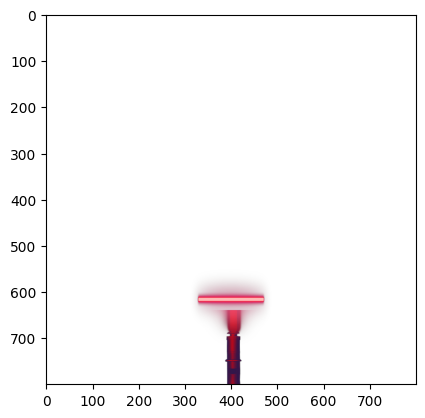
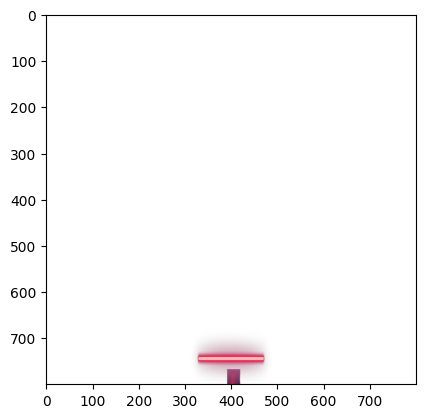
Kết hợp trigger\_grab và trigger\_stomp

## **I. 4. Xử lí hiệu ứng**

## **I. 4. 1. Hiệu ứng xuất hiện và biến mất**

Với đầu vào là ảnh của kiếm và mặt nạ, tiến hành chia thành **n** phần (tốc đọ hiệu ứng xuất hiện), và cho từng phần xuất hiện lần lượt cho đến khi ảnh được xuất hiện đầy đủ. Và để tăng thêm phần bắt mắt của hiệu ứng thì một đường thẳng neon được thêm vào ngay phía trên của phần mới được xuất hiện.

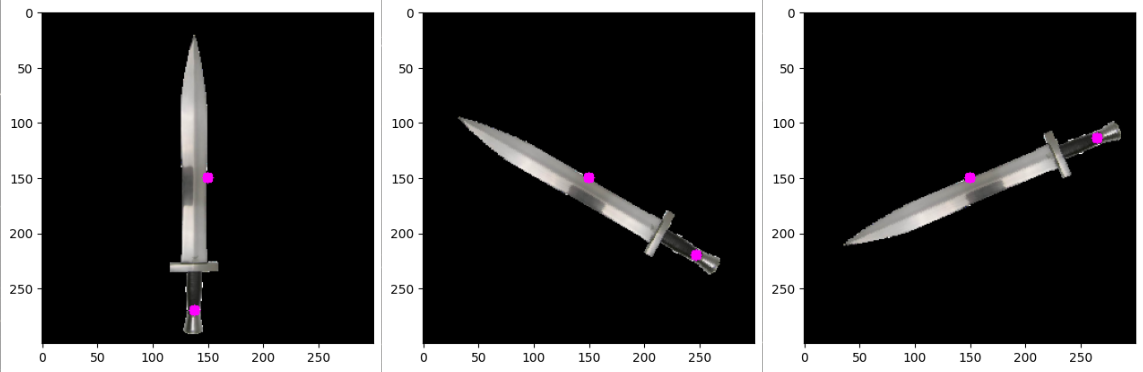
Tương tự cho hiêu ứng biến mất, ta cũng tiến hành chia thành **n** phần, và cho từng phần đó biến mất dần theo từng frame, và cuối cùng là chèn thêm một đường thẳng neon và để thêm độ bắt mắt.



## **I. 4. 2. Cố định vật thể vào các landmark**

Việc cố định các vật thể như kiếm và mặt nạ vào landmark sẽ được chia làm 2 kiểu xử lí đó là cố định theo tâm vật thể, hoặc cố định theo một điểm cụ thể của vật thể (ví dụ như tay cầm của kiếm). Vì khi di chuyển tay, vật thể như kiếm cũng sẽ được xoay theo một góc nào đó được xác định (của nhóm là góc tạo bởi cánh tay với 2 điểm là khuỷu tay và cổ tay, với trục hoành).

* Cố định theo tâm: Xác định tâm của vật thể bằng cách lấy trung điểm của chiều cao và rộng của ảnh và xoay ảnh theo một góc cụ thể nào đó. Sau đó tiến hành chèn vật thể vào ảnh chính dựa vào 2 điểm là landmark (trung điểm 2 mắt) và tâm vật thể, bằng hàm insert\_object mà nhóm tự định nghĩa.
* Cố định theo điểm cụ thể: Xác định điểm tay cầm (x1, x2) khi vật nằm thẳng đứng, và trọng tâm của nó (c1, c2), khi có được góc xoay, tiến hành thông qua hàm get\_rotate\_image\_and\_point để xác định được ảnh sau khi xoay điểm tay cầm sau khi xoay (x2, y2). Và chèn vật thể vào ảnh chính dựa vào 2 điểm là landmark và điểm (x2, y2).



(x1, y1) = (138, 270) (x2, y2) = (247, 220) (x2, y2) = (265, 114)

(c1, c1) = (150, 150) Angle = 60 Angle = 113

## **I. 4. 3. Các điều khác**

Ngoài các thành phần chính như trên, còn có các thành phần khác cũng không kém phần quan trọng trong việc bổ trợ cho hiệu ứng được mượt mà hơn:

* Để giúp kiếm và mặt nạ được fit với cơ thể hơn, thì nhóm cũng tiến thành trích xuất tự động độ dài của mặt (2.5 lần khoảng cách 2 tai) cho kích thước ảnh mặt nạ, độ dài của cẳng tay (khoảng cách khuỷu tay và cổ tay) cho độ dài của kiếm (3.5 lần độ dài cẳng tay) cho từng frame. Ngoài ra, áp dụng Savgol-Golay filter để giảm nhiễu giúp hiệu ứng mượt mà hơn.
* Trích xuất trước các góc độ xoay của từng hiệu ứng cho từng frame.

## **I. 5. Tổng hợp xử lí**

Duyệt qua từng frame trong video, ta phân thành các trạng thái:

* Chèn hiệu ứng:

+ Xuất hiện.

+ Biến mất.

+ Tĩnh.

* Không chèn hiệu ứng

Để xác định được các trạng thái, nhóm đưa ra các thông số sau:

* isSword, isMask:

+ Được cập nhật liên tục thông qua trigger\_grab và trigger\_cross. Với 1 là True và 0 là False.

+Lúc này thì trạng thái Xuất hiện của riêng từng vật thể sẽ được kích hoạt thông qua trạng thái Chèn hiệu ứng.

+Sau khi hiệu ứng Xuất hiện đã chèn đủ số lượng frame thì sẽ tiến đến trạng thái Tĩnh của Chèn hiệu ứng.

* isStomp, out\_waiting:

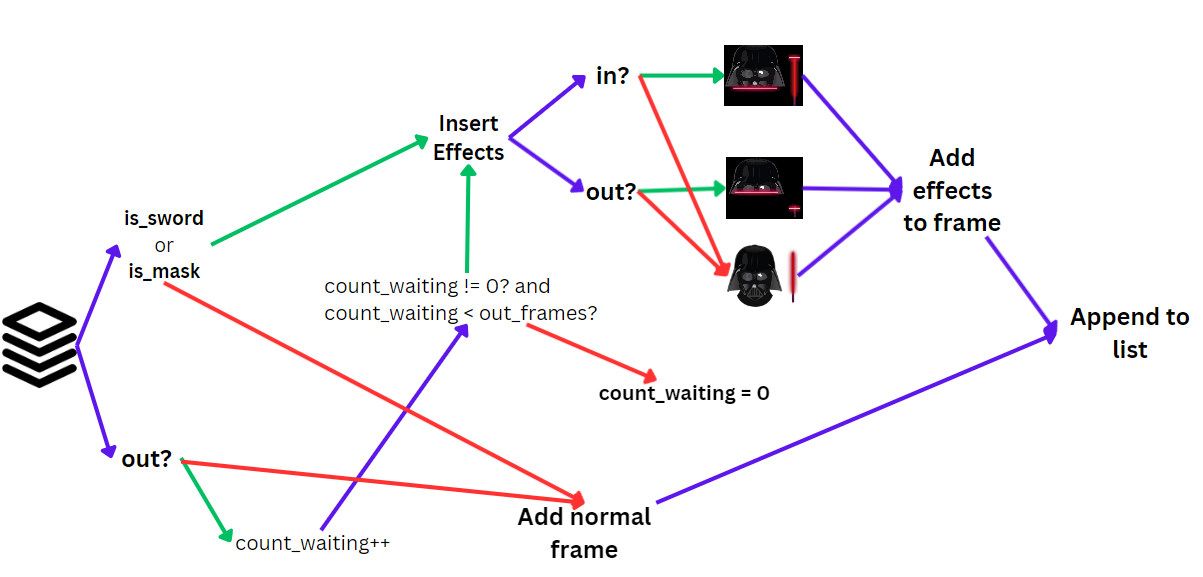
+ isStomp = True khi duyệt trong trigger\_stomp xác định được 1

+ Khi isStomp = True, thì trạng thái Biến mất được kích hoạt, và out\_waiting sẽ tăng dần đến khi đủ số lượng frame của hiệu ứng biến mất thì isStomp = False và out\_waiting = 0.

+ Khi trạng thái Biến mất được kích hoạt, thông qua trạng thái Chèn hiệu ứng chèn vào các frame hiệu ứng biến mất.

Việc phải chia thành nhiều trạng thái bởi vì khi hiệu ứng kết thúc thì isSword và isMask lúc đó sẽ đều bằng 0, lúc này khó có thể chèn hiệu ứng biến mất một cách dễ dàng được. Từ đó mới phải có thông số out\_waiting để giữ trạng thái Chèn hiệu ứng tiếp tục ở lại đến khi hiệu ứng biến mất kết thúc.

Với trạng thái Không chèn hiệu ứng thì đơn giản chỉ là khi không có trạng thái Chèn hiệu ứng thì tiến hành chèn frame của video gốc vào.



Sơ đồ tổng quan việc tổng hợp xử lí. Với đường xanh dương là thực hiện, xanh lá là nếu đúng, đỏ là nếu sai.

Và cuối cùng, sau khi xử lí xong, ta sẽ thu được 1 list bao gồm các frame đã được xử lí. Có thể xuất list này ra bằng các định dạng như video, gif, ...

# **II. ABC**

# **III. ABC**

# **IV. Tô màu biên cạnh ảnh trong video thời gian thực**

1.Ý tưởng :

Các hiệu ứng ánh sáng thường được ứng dụng nhiều trong các Mv âm nhạc chủ đề rap và rock.Với việc vận dụng linh hoạt các hiệu ứng chuyển cảnh, thay đổi màu sắc, các hiệu ứng tô màu biên cạnh với ánh sáng neon... tạo nên sự ấn tượng thị giác cho các MV ca nhạc đình đám hiện nay.Xuất phát từ cảm hứng khi xem các video âm nhạc trên, cùng với muốn vận dụng các kiến thức được học về xử lý ảnh cho video thời gian thực, nên em đã thực hiện đề tài này.

2.Nội dung

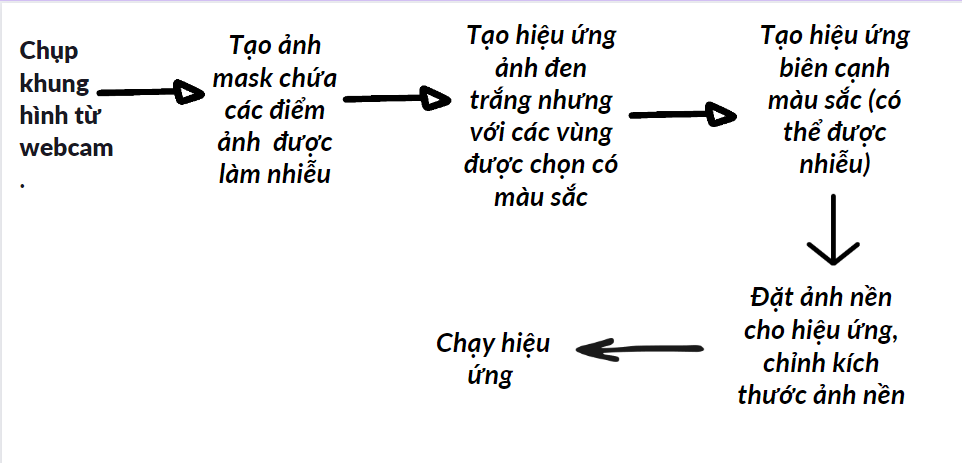
**A. Chuẩn bị dữ liệu**

* Để thực hiện đề tài này, em chuẩn bị sử dụng hàm gọi videocam trong CV2 để có video thực chứa người và đồ vật cần tô màu.
* Tiếp theo , em tìm kiếm các ảnh nền màu chứa màu dùng để tô viền cho video.
* Và tìm kiếm ,xây dựng các hàm cần thiết để thực hiện đề tài.



Một số ảnh nền dùng để chạy hiệu ứng cho đề tài

**B.Quy trình thực hiện**



B1.Chụp khung hình từ webcam

Tạo một class được đặt tên là CamEffect. Class này có chức năng chụp một khung hình từ video stream.

Đầu tiên, trong hàm khởi tạo **init**, class sẽ khởi tạo một đối tượng VideoCapture với tham số 0 để mở camera. Nếu không mở được camera thì sẽ raise ra một Exception.

Hàm capture\_frame sẽ đọc một khung hình từ video stream và lật ngược hình ảnh theo chiều dọc bằng hàm flip của OpenCV.

B2: Xây dựng hàm tách cạnh và tô màu biên cạnh cho video

* Để thực hiện được tô màu cho biên cạnh, ý tưởng của em là xây dựng hàm tách ra các cạnh của khung hình được trả về của class CamEffect, sử dụng các kỹ thuật được học như chuyển đổi ảnh sang không gian Grayscale, phương pháp Canny.
* Sau đó tạo một ảnh mask chứa các cạnh đã được tách, chuyển đổi ảnh mask về ảnh đen trắng, với màu trắng là màu các cạnh.Thao tác này được thực hiện bằng cách sử dụng hai hàm tạo ảnh mask và một hàm black-white. Cụ thể:
* Ở hàm tạo ảnh mask,tạo một hiệu ứng nhiễu (fizz) trên các điểm ảnh được chỉ định bởi một mặt nạ (mask). Hàm này tạo ra một mặt nạ mới với các điểm ảnh đã được nhiễu và gán giá trị 255 cho các điểm đó.
* Hàm black-white này sẽ chuyển đổi ảnh đầu vào sang không gian màu HSV và tạo ra một mask mới với các điểm có giá trị của kênh V (Value) trong khoảng [100, 255]
* Sau khi có hàm black-white để chuyển ảnh sang đen trắng,và gọi hàm tạo mask để tạo ra các mask mới với các điểm có giá trị kênh trong khoảng từ [100,255] ,ta sẽ thực hiện tô màu mask mới này bằng cách blending các viền với màu được lấy từ ảnh nền đã được chuẩn bị.
* Để tạo sự linh hoạt cho hiệu ứng trên, em đã thực hiện xây dựng 2 cách tô màu cho ảnh mask.Với biến choice được gọi, khi choice bằng 1 thì ta thực hiện tách cạnh và tạo ảnh mask, và tô màu cho các cạnh đó.Với choice=2 thì ta làm ngược lại, tô màu cho các vùng không phải cạnh.

**C. Các vấn đề khi thực hiện**

- Hiệu ứng được xây dựng còn đơn giản, chưa có tạo ra hiệu ứng thị giác cao như mong đợi. Kết quả tô màu chỉ đạt hiệu ứng tốt khi không gian mà chạy webcam có ánh sáng đủ mạnh, nếu không thì ảnh trả về bị tô đen rất nhiều.Hướng khắc phục của em có thể xây dựng thêm hàm tăng độ sáng đến một ngưỡng quy định cần thiết để việc tách đạt kết quả.

- Hiệu ứng chưa đạt được như ý tưởng mong muốn, chưa đủ linh hoạt, chỉ có thể bị động chạy hiệu ứng từ đầu đến lúc tắt code.Hướng đi em đang hướng đến chính là đưa hiệu ứng này vào việc vận dụng hand segmentation, khi mà khẩu hình tay đạt một hình thái nào đó mới kêu gọi tách cạnh của video.Tuy nhiên em chưa giải quyết được vấn đề vận dụng hand segmentation nên chỉ mới dừng đến hiệu ứng hiện tại.

**D.Tài liệu tham khảo**

https://minhng.info/tutorials/xu-ly-anh-opencv-hien-thuc-canny-edge.html( hàm tách cạnh)

https://minhng.info/tutorials/xu-ly-anh-opencv-hien-thuc-canny-edge.html( xử lý với video thời gian thực) https://gist.github.com/hungneox/4814bbd49e5fbcf67f809be7a96cbfe9( tạo ảnh mask)

* Các hiệu ứng về ánh sáng thường được sử dụng nhiều trong các mv âm nhạc phong cách rap và edm.
* Vận dụng các kiến thức được học như tách viền, blending ảnh, thay đổi kích thước ảnh ,... để tạo nên hiệu ứng với video thời gian thực.
* Các hiệu ứng về ánh sáng thường được sử dụng nhiều trong các mv âm nhạc phong cách rap và edm.
* Vận dụng các kiến thức được học như tách viền, blending ảnh, thay đổi kích thước ảnh ,... để tạo nên hiệu ứng với video thời gian thực.

# **V. Hiệu ứng cổ trang**

## **V. 1. Ý tưởng cho hiệu ứng**

Sự phát triển của mạng xã hội đồng thời làm cho các yếu tố cổ trang hóa từ trang phục, âm nhạc, điện ảnh, … càng trở nên phổ biến và được ưu chuộng. Bên cạnh đó, niềm yêu thích cổ trang của bản thân đã thúc đẩy em đến với đề tài này.

Mong rằng sự cố gắng trong việc xây dựng hiệu ứng cổ trang có thể lan tỏa được niềm cảm hứng yêu thích cổ trang đến cho những người khác.

## **V.2. Nội dung**

- Xây dựng một hiệu ứng xử lí trên video, chất lượng VGA (640x480), video màu RGB, 60fps.

- Nội dung xử lí tập trung vào xử lí hành động của nhân vật trong video.

- Sử dụng thư viện mediapipe để hỗ trợ phát hiện dáng người thông qua pose landmark detection.

- Chi tiết hiệu ứng, bao gồm:

+ Xử lí trên động tác tay của nhân vật trong video

+ Xử lí thay nền (background)

+ Xử lí trộn ảnh

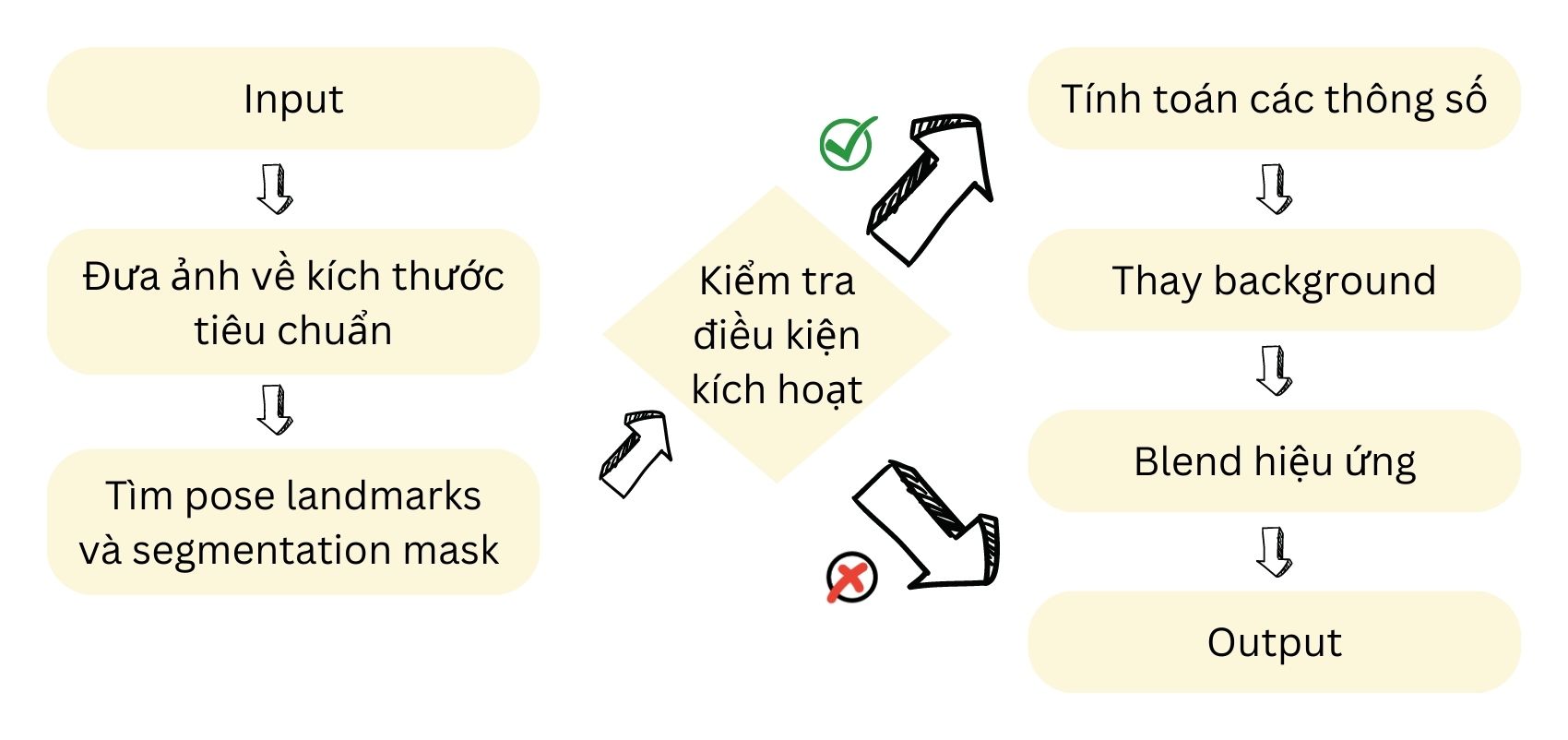
+ Xử lí tạo hiệu ứng động

- Video kết quả có cùng những thông số như video đầu vào: video màu RGB, chất lượng VGA (640x480), 60fps.

Sau đây, chúng ta cùng đi qua chi tiết các vẫn đề và cách thức xử lí các vấn đề trong quá trình thực hiện.

## **V.3. Quy trình thực hiện**

Để thực hiện quá trình xử lí các yêu cầu bên trên, ta thực hiện đọc video đầu vào dưới dạng frame, sau đó từng frame sẽ được đưa qua sơ đồ bên dưới để xử lí một cách hiệu quả.



*Hình. Quy trình xử lí sơ bộ của hiệu ứng*

Khung hình (frame) từ input sẽ được chuẩn hóa về khích thước đã quy định trước đó.

Sau đó sử dụng thư viện mediapipe để thực hiện phát hiện (detecition) các điểm dáng người của nhân vật trong ảnh (pose landmark detection).

Sau khi có được tọa độ các điểm cần thiết, ta sẽ kiểm tra điều kiện kích hoạt hiệu ứng, cụ thể với hiệu ứng mà em đã xây dựng thì đó là bàn tay (bàn tay so với góc nhìn của nhân vật trong ảnh) giơ cao hơn vai và bàn tay giơ cao hơn đầu nhân vật (thông qua điểm cổ tay và một số trung điểm).

Nếu không thỏa mãn điều kiện kích hoạt thì khung hình này không cần xử lí, nó cũng chính là ảnh đầu ra (output).

Nếu thỏa mã điệu kiện kích hoạt, ta sẽ đi xử lí theo trình tự:

+ Tính toán các thông số: vị trí các điểm cần (trộn ảnh), tỉ lệ người và tỉ lệ ảnh của nhân vật sao cho phù hợp.

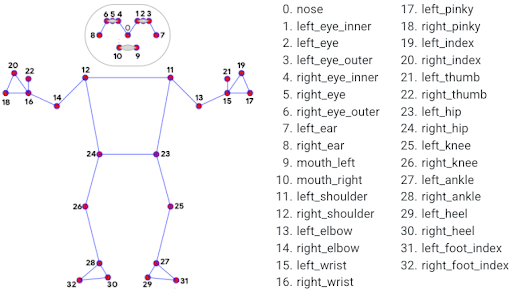
+ Thay đổi ảnh nền

+ Thực hiện các phép trộn ảnh với thông số đã tính

Chi tiết của quá trình xử lí sẽ được em trình bày cụ thể bên dưới.

## **V.4. Xác định vị trí các điểm**

Xác định vị trí các điểm tức là tính toán để tìm ra tọa độ và tỉ lệ cần thiết cho việc xử lí. Quá trình này sẽ diễn ra sau khi ta đã sử dụng thư viện để phát hiện (detection) các điểm dáng người.



*Hình. Các điểm dáng người mà thư viện mediapipe hỗ trợ phát hiện*

Tọa độ các điểm cần tính bao gồm:

+ Trung điểm giữa hai mắt, trung điểm giữa hai vai, để tính các vị trí trung điểm này, ta chỉ đơn thuần tính trung bình cộng giữa các điểm liên quan lại với nhau.

+ Tâm của lòng bàn tay: vị trí này là cực kì cần thiết để xác định trung tâm của hiệu ứng chính, tính toán bằng công thức trung bình cộng của ba điểm gồm cổ tay (wrist), ngón út (pinky) và ngón trỏ (index).

+ Khoảng cách giữa vai và hai mắt: khoảng cách này giúp đo đạc tỉ lệ người trong khung ảnh, giúp tỉ lệ hiệu ứng được áp vào cho phù hợp. Tính bằng công thức độ đo khoảng cách Euclidean giữa hai điểm trung điểm mắt và trung điểm vai.

## **V.5. Mặt nạ phân mảnh**

Mặt nạ phân mảnh (segmentation mask) là ma trận xác định vị trí những điểm ảnh nào (pixel) có chứa nhân vật chính hay người trong ảnh.

Mặt nạ phân mảnh có nằm trong kết quả đi kèm trả về của thư viện mediapipe. Kết quả trả về là một ma trận có kích thước bằng với kích thước khung hình đưa vào. Mỗi vị trí trong ma trận chứa một con số nguyên 0 – 255, với 0 biểu thị không chứa người và 255 là có người tại ví trí điểm ảnh tương ứng.

Vấn đề của việc xử lí trên mặt nạ phân mảnh là do những giá trị thuộc khoảng (0, 255) là những giá trị cần xử lí, ở đây em tìm được hai phương pháp cụ thể để xứ lí như sau: xử lí với ngưỡng (thresh), và dùng cách dilate hoặc erode.

Cân nhắc hai phương pháp trên:

- Sử dụng ngưỡng:

+ Không có tính linh hoạt với những khung hình khác nhau

+ Đáp ứng được yêu cầu

- Sử dụng dilate, erode :

+ Giảm chiều ma trận

+ Không thay đổi quá nhiều so với trước khi chưa sử dụng



*Hình. Mặt nạ phân mảnh gốc, sau khi dilate, sau khi erode theo thứ tự từ trái sang*

Kết quả không quá thay đổi nhiều sau khi dùng và nhược điểm giảm chiều ma trận mặt nạ phân mảnh đã khiến em không lựa chọn cách xử lí này.

Cuối cùng, sau khi cân nhắc, thì sử dụng ngưỡng được em đánh giá là phù hợp hơn với bài toán của mình.

## **V.6. Vấn đề khi trộn ảnh (blending)**

Vấn đề khi trộn ảnh là việc ảnh bị lem ( ảnh có thể bị tối đi hoặc sáng hơn ở những vị trí khồn cần thiết) làm cho ảnh sau khi xử lí không được chân thực.

Để giải quyết vấn đề trên, em sử dụng ảnh hiệu ứng là ảnh RGBA.

### **V.6.1 Ảnh RGBA**

Ảnh RGB thực chất là ảnh sử dụng thang màu RGB (Red Green Blue) đỏ lục lam, tuy nhiên có thêm kênh A (Alpha) biểu thị cho độ mờ.

Giá trị của kênh Alpha là một số thực từ 0 – 1, với 0 biểu thị cho điểm ảnh đó trong suốt và 1 là điểm ảnh tương ứng đục màu (hiển thị với giá trị gốc).



*Hình. Ảnh RGBA*

Ảnh hiệu ứng được sử dụng là ảnh RGBA với giá trị alpha = 0 tại những vị trí nền của hiệu ứng, và khác 0 với những vị trí điểm ảnh cần giữ.

Như vậy, quá trình trộn ảnh sẽ là trộn giữa ảnh gốc và ảnh hiệu ứng có sử dụng mặt nạ phân mảnh (segmentation mask) và kênh alpha:

+ Xác định điểm ảnh có cần trộn ảnh với mặt nạ mask

+ Xác định tỉ lệ hòa trộn trên ảnh hiệu ứng với kênh alpha.

Kết quả là ảnh sau khi hòa trộn sẽ giữ được các đặc tính mà ta mong muốn.



*Hình. Ảnh vòng sáng không hề làm ảnh nền (background) bị lem*

## **V.7. Xử lý di chuyển**

Xử lí di chuyển là xử lí các hiệu ứng để các hiệu ứng có thể di chuyển và có tỉ lệ phù hợp với tỉ lệ của nhân vật trong khung hình.

Một nhân vật sẽ bao gồm 2 nhóm di chuyển chính trong khung hình:

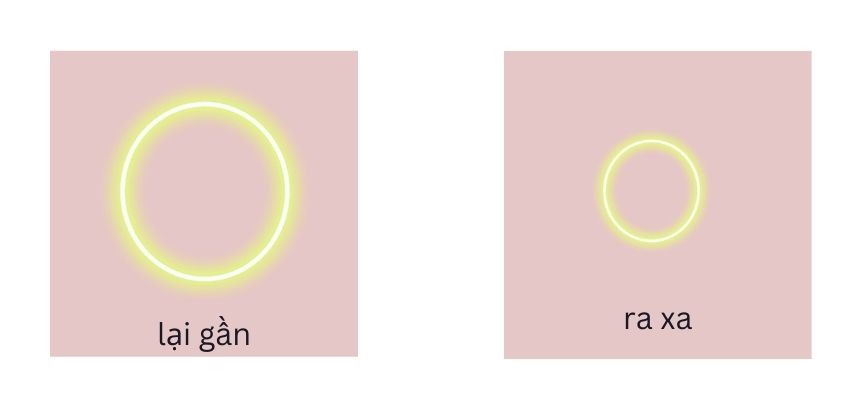
+ Di chuyển sang trái, sang phải

+ Di chuyển lại gần, ra xa góc nhìn của camera.

Đối với mỗi nhóm, sẽ có các phương pháp xử lí khác nhau.

+ Di chuyển sang trái, sang phải: xử lí bằng cách thực hiện chọn một vị trí của nhân vật, thực hiện trộn ảnh với vị trí đó là trung tâm, cụ thể ở đây, bài toán này em sử dụng trung điểm mắt và tâm lòng bàn tay.

+ Di chuyển lại gần, ra xa: xử lí bằng cách tính tỉ lệ cơ thể nhân vật trong điểm ảnh, sau đó thay đổi kích thước cho phù hợp.



*Hình. Cùng vòng sáng, nhưng nhân vật lại gần nên to hơn khi nhân nhân vật ra xa*

## **V.8. Tạo ảnh động**

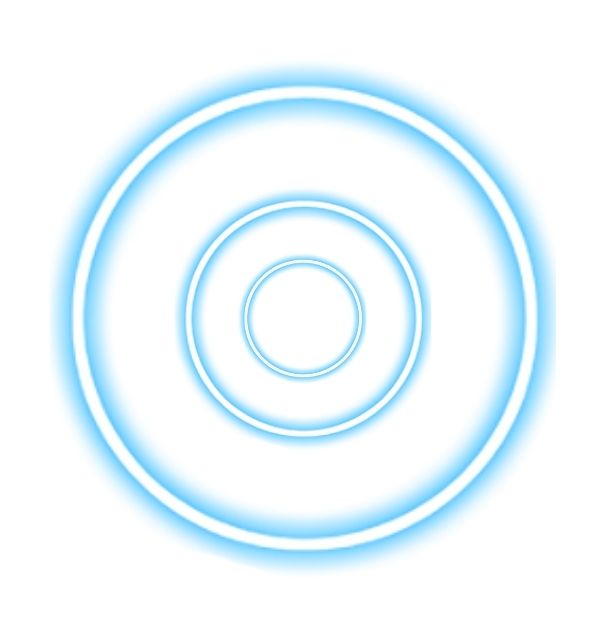
Tạo ảnh động trong bài toán này là thay đổi kích thước của vòng sáng trên tay nhân vật liên tục cho phù hợp. Từ đó, tạo cảm giác tỏa sáng liên tục.

Có hai loại tỉ lệ em sử dụng:

+ Thay đổi từ 25% - 100% so với ảnh hiệu ứng gốc

+ Thay đổi từ 10% - 100% so với ảnh hiệu ứng gốc.

Quá trình này chỉ đơn thuần là thay đổi kích thước ảnh (resize) theo những tỉ lệ đã chỉ định từ trước so với khung hình gốc.

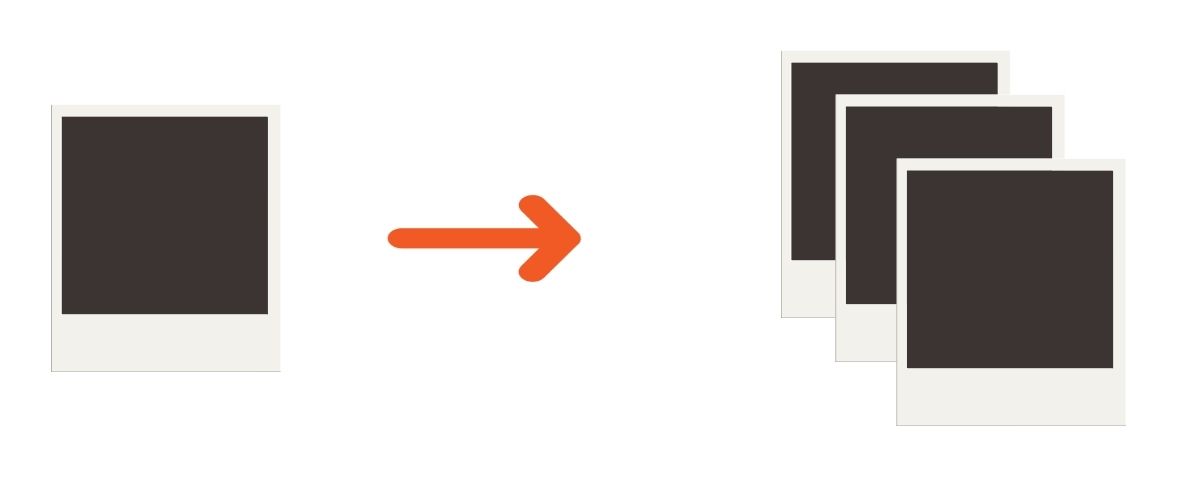


*Hình. Minh họa cho việc thay đổi kích thước liên tục so với ảnh hiệu ứng gốc.*

## **V.9. Xử lý ngắt quãng**

Xử lí ngắt quãng là giảm bớt độ ngắt quãng hiệu ứng cho video vì việc phát hiện các khung hình liên tục của thư viện mediapipe và di chuyển của nhân vật trong video có thể làm cho hiệu ứng trong video bị ngắt quãng.

Trong bài toán này, em dùng cách xử lí trên một nhóm frames ảnh được đọc vào từ video, cụ thể là 3. Mặc dù phương pháp này không được tốt, nhưng nó có giúp giảm bớt phần nào độ ngắt quãng cho video, cụ thể là với video demo cho hiệu ứng này.



*Hình. Đọc 3 frames, xử lí trên 1 frame đầu và tận dụng kết quả đó cho 2 frames còn lại*

Để thực sự xử lí tốt ngắt quãng hiệu ứng của video thì cần tìm ra một phương pháp tốt hơn, chẳng hạn như xử lí toàn bộ các vị trí bắt đầu và kết thúc hiệu ứng một lúc, thông qua việc đọc hết tất cả những frames của một video, rồi mới bắt đầu thực hiện xử lí.

Đây cũng là một hướng phát triển tốt nhằm cải thiện hiệu quả cho hiệu ứng của em sau này.

## **VI. Tài liệu tham khảo**

<https://mediapipe-studio.webapps.google.com/demo/pose_landmarker>

<https://picsart.com>

<https://www.wikipedia.org>

<https://opencv.org>