

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



**THIẾT KẾ HAND EFFECT
(ENERGY BALL)**

Giáo viên hướng dẫn: Nguyễn Vĩnh Tiệp

Thành viên nhóm:

20521609 - Nguyễn Hoàng Minh

20521998 - Nguyễn Thiện Thuật

20521614 - Tạ Nhật Minh

20522010 - Nguyễn Minh Tiến

Sinh viên viết báo cáo:

20521998 - Nguyễn Thiện Thuật

TP. HỒ CHÍ MINH – 12/2022

LỜI MỞ ĐẦU

Trong những năm gần đây, các sản phẩm về những hiệu ứng hình ảnh có khả năng tương tác với con người được sử dụng trong nhiều mục đích giải trí khác nhau đã được thế giới sử dụng rộng rãi. Qua đó, việc sáng tạo và phát triển nhiều hiệu ứng hình ảnh hiện đại hơn, chân thật hơn và thông minh hơn luôn tạo ra những sản phẩm mới lạ, độc đáo, thú vị được người dùng đón nhận và hưởng ứng nhiệt tình. Đặc biệt trong thời đại mạng xã hội phát triển mạnh mẽ, mỗi người đều sở hữu riêng cho mình một chiếc điện thoại thông minh, những hiệu ứng hình ảnh luôn được cập nhật, thay đổi và được thêm vào nhiều nền tảng ứng dụng khác nhau. Kết hợp với nhu cầu giải trí được tăng cao thì việc tiếp cận đến các sản phẩm này trở nên dễ dàng hơn trước đây rất nhiều.

Với sự ra đời của Mediapipe trong python, thư viện giúp việc nhận dạng các bộ phận của cơ thể trở nên dễ dàng hơn. Nhằm hướng đến việc tương tác giữa con người với các hiệu ứng trở nên chân thực hơn, chúng tôi đã tận dụng khả năng nhận dạng khung xương ở tay người của thư viện kết hợp với các phương pháp xử lý ảnh khác nhau.

Vì vậy, chúng tôi đã tiến hành xây dựng các hiệu hình ảnh khác nhau và thực hiện các thao tác trên tay người. Trong bài này, chúng tôi thực hiện xây dựng hiệu ứng hình ảnh Energy Ball (Quả cầu năng lượng)

MỤC LỤC

1. GIỚI THIỆU	1
1.1. Đặt vấn đề.....	1
1.2. Mục tiêu và thách thức của đề tài.....	1
1.2.1. Mục tiêu.....	1
1.2.2. Thách thức	2
2. NỘI DUNG.....	2
2.1. Hướng tiếp cận	2
2.2. Giải pháp chi tiết	3
2.2.1. Mediapipe và các điểm trên tay.....	3
2.2.2. Transparent Image quả cầu với hình ảnh từ camera.....	4
2.2.3. Affine Transformations nhằm xoay quả cầu	5
2.2.4. Chuyển màu quả cầu	6
2.2.5. Effect nổ quả cầu bằng Image Blending	7
2.3. Kết quả và nhận xét	9
2.3.1. Kết quả đạt được.....	9
2.3.2. Nhận xét.....	9
3. KẾT LUẬN	9
4. HƯỚNG PHÁT TRIỂN.....	9

1. GIỚI THIỆU

1.1. Đặt vấn đề

Trong những năm gần đây, việc xuất hiện của Trí tuệ nhân tạo - Artificial Intelligence (AI) đã tạo ra một cuộc cách mạng công nghệ 4.0. AI xuất hiện ở khắp mọi nơi, mọi lĩnh vực và đã giúp cuộc sống hiện đại trở nên dễ dàng hơn.

Nhận diện chuyển động cử chỉ bàn tay với OpenCV – Hand Tracking sử dụng MediaPipe Hands cũng từ đó xuất hiện. Khả năng nhận biết hình dạng và chuyển động của bàn tay có thể là một thành phần quan trọng trong việc cải thiện trải nghiệm người dùng trên nhiều lĩnh vực và nền tảng công nghệ khác nhau. Bên cạnh đó khả năng nhận biết tay với thời gian thực là một nhiệm vụ khó khăn về thị giác máy tính (Computer Vision). Tuy nhiên với sự phát triển không ngừng của thế giới hiện nay, MediaPipe Hands đã xuất hiện và là giải pháp theo dõi ngón tay cũng như bàn tay có độ trung thực cao. Nó sử dụng công nghệ máy học (ML) để suy ra 21 đi ểm mốc 3D của bàn tay chỉ từ một khung hình duy nhất.

Với các ứng dụng của MediaPipe trong thị giác máy tính kết hợp với đó là nhu cầu được tiếp cận với những thứ mới mẻ mà AI mang lại của người dùng trên các nền tảng công nghệ hiện nay, ở bài này chúng tôi đề xuất áp dụng MediaPipe Hands kết hợp với nhiều phương pháp xử lý hình ảnh như Transparent Image, Affine Transformations, Image blending sẽ tạo ra những trải nghiệm mới mẻ, thú vị về hiệu ứng hình ảnh mà con người có khả năng tương tác với chúng như ở ngoài hiện thực.

1.2. Mục tiêu và thách thức của đề tài

1.2.1. Mục tiêu

Tạo ra các hiệu ứng hình ảnh mới mẻ, thú vị hứng đến người dùng trên các nền tảng công nghệ số bằng việc code python.

Hướng đến việc tương tác trực tiếp và tự nhiên giữa con người với các hiệu ứng.

Nghiên cứu và cải tiến các hiệu ứng hình ảnh được chân thực nhất, xử lý chính xác và nhanh nhất, hướng đến các hiệu ứng được diễn ra trong thời gian thực.

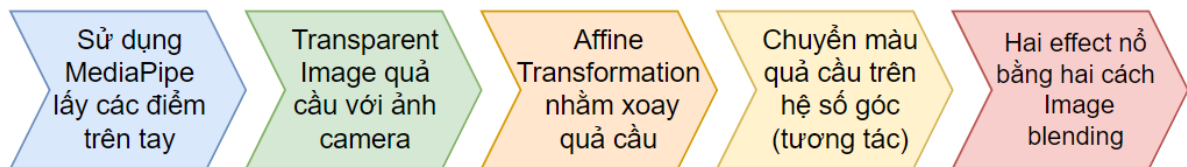
1.2.2. Thách thức

Vì nhằm hướng đến thời gian thực và xử lý chính xác nên việc tìm ra giải pháp giải quyết các vấn đề này đòi hỏi phải thử nghiệm với rất nhiều phương pháp. Bên cạnh đó là việc phải tạo ra nhiều ý tưởng để có một sản phẩm mà người dùng có thể tương tác với chúng. Nghiên cứu về các kỹ thuật lập trình để hiện thực hóa những ý tưởng cũng như chọn nhưng tham số phù hợp để hiệu ứng hoạt động chân thực và tự nhiên nhất.

2. NỘI DUNG

2.1. Hướng tiếp cận

Nhóm đã lên ý tưởng và nghiên cứu nhiều hiệu ứng hình ảnh khác nhau. Đặt biệt các hiệu ứng hình ảnh tận dụng khả năng của thư viện MediaPipe trong việc phát hiện các điểm trên ngón tay cũng như bàn tay. Trong bài báo cáo này, chúng tôi sẽ tập chung giải quyết vấn đề về cách để tạo ra một hiệu ứng Energy Ball (Quả cầu năng lượng). Hướng tiếp cận của bài toán được thực hiện theo trình tự như sau:



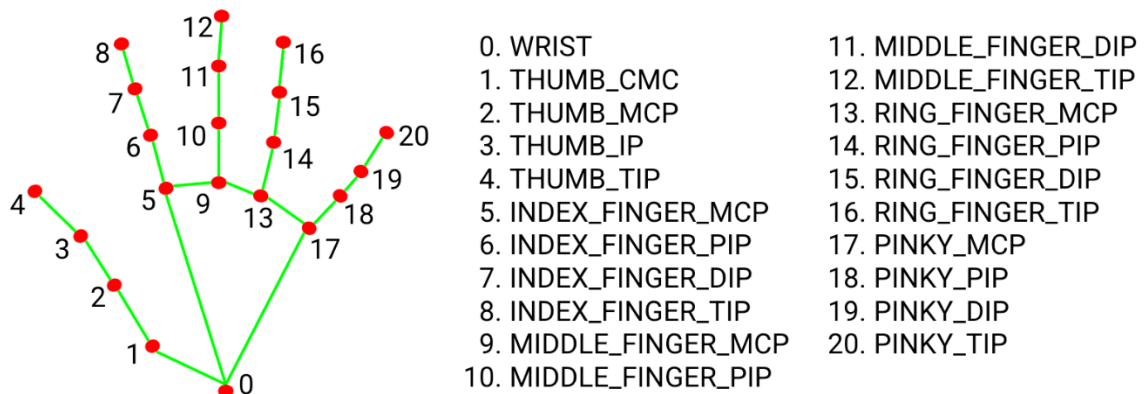
Hình 1. Hướng tiếp cận để tạo effect Energy Ball

Đầu tiên thư viện MediaPipe giúp lấy các điểm trên tay. Từ như điểm đó xác định vị trí và kích thước của quả cầu và thực hiện phương pháp Image blending cho quả cầu với ảnh từ camera. Sau khi có được quả cầu, chúng tôi tiến hành xoay quả cầu cho hiệu ứng được sinh động. Nhằm hướng đến việc tương tác với quả cầu được tạo ra, việc nghiên cứu cách chuyển màu trong hệ màu RGB được triển khai, kết hợp với đó là cách để tạo ra hệ số thay đổi màu với thao tác tay của người dùng. Cuối cùng là tạo ra hai effect hình ảnh làm nổ quả cầu khác nhau (nổ trắng màn hình và nổ ra màu sắc) với Image blending kết hợp với những điều kiện khác nhau.

2.2. Giải pháp chi tiết

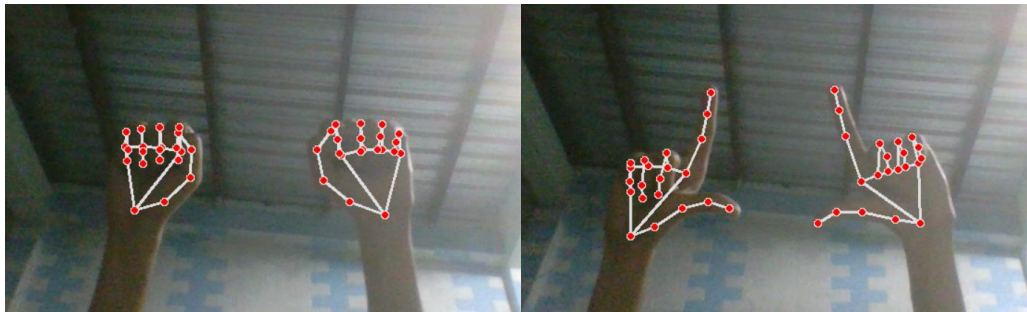
2.2.1. Mediapipe và các điểm trên tay

Nhằm hướng đến độ chính xác tốt nhất để lấy được các điểm trên tay, chúng tôi đã sử dụng hàm **multi_hand_landmarks** để lấy các điểm trên cả hai bàn tay như hình bên dưới.



Hình 2. Các điểm phát hiện được trên tay từ thư viện MediaPipe

Thao tác tay tạo quả cầu: Để tạo ra quả cầu cần phải cho cả 2 bàn tay xuất hiện trên màn hình camera. Tiếp đến nắm 2 bàn tay lại, sau đó bung ngón chỏ và ngón cái ở hai bàn tay ra như Hình 3.



Hình 3. Thao tác tay đúng để tạo quả cầu năng lượng

Để cho máy xác định được thao tác tay như trên chúng tôi sử dụng các điểm được xác định ở hình 2 và cài đặt các bước như sau:

- Tìm khoảng cách từ cổ tay (điểm 0) đến các đầu ngón tay (điểm 4, 8, 12, 16 và 20) và từ cổ tay đến gốc ngón trỏ (điểm 5).
- Tính $wrist_3tail$ = trung bình cộng khoảng cách từ cổ tay đến đầu ngón tay ngón giữa, nhẫn và út.

- Cài đặt điều kiện:

- Khoảng cách từ cổ tay đến đầu ngón trỏ $> \text{wrist_3tail}$.
- Khoảng cách từ cổ tay đến đầu ngón trỏ $>$ từ cổ tay đến gốc ngón trỏ (loại bỏ thao tác tay khi đầu ngón trỏ nằm trong lòng bàn tay).

Kích thước quả cầu: Đường kính của quả cầu được tính bằng trung bình cộng khoảng cách đầu ngón trỏ đến đầu ngón cái của hai tay nhân với hệ số (sản phẩm của chúng tôi hệ số bằng 1.25). Với việc cài đặt đường như vậy, người dùng hoàn toàn điều khiển được kích thước của quả cầu bằng thay đổi khoảng cách giữa hai đầu ngón trỏ và ngón cái.

Vị trí tạo quả cầu: Khi xác định thao tác tay đúng, máy tiến hành xác định vị trí trung tâm của quả cầu bằng cách xác định trung điểm từ đầu ngón trỏ và đầu ngón cái của bàn tay. Sau đó xác định trung điểm của quả cầu từ hai trung điểm ở hai tay đã được tính trước đó. Từ vị trí trung tâm của quả cầu kết hợp với kích thước của quả cầu ta tìm được vị trí đầu tiên (x_{\min} , y_{\min}).

2.2.2. *Transparent Image quả cầu với hình ảnh từ camera*

Sau khi xác định được vị trí và kích thước của quả cầu, chúng tôi có thể ghép ngay bằng những phương pháp đơn giản hơn. Tuy nhiên để quả cầu tạo cảm giác chân thực nhất, chúng tôi đã kết hợp chúng với nhau bằng phương pháp transparent Image. Trong phương pháp này, chúng tôi tận dụng hệ số alpha của hình ảnh trong việc pha trộn ảnh. Để Transparent Image quả cầu với hình ảnh từ camera ta tiến hành cài đặt như sau:

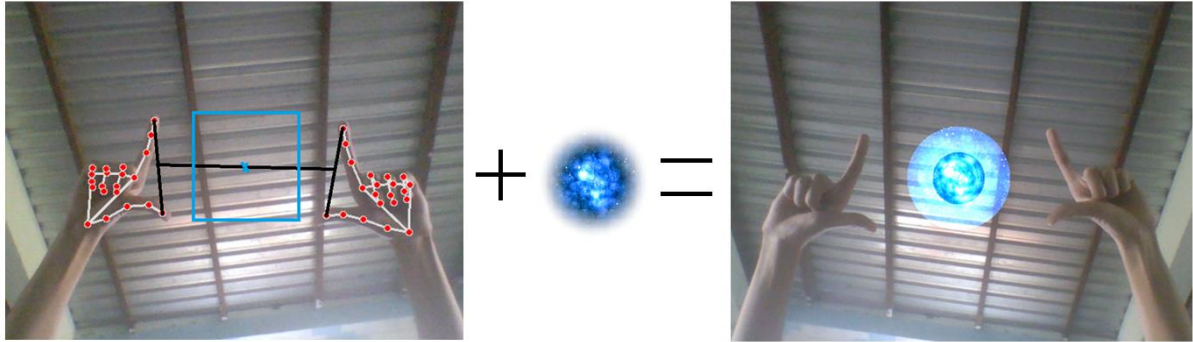
Chọn vùng ảnh từ camera để pha trộn quả cầu: Từ vị trí $[y, x]$ và kích thước (diameter) của quả cầu được xác định từ trước, đầu tiên tiến hành resize ảnh về đúng kích thước đã tính. Sau đó chọn vùng ảnh từ camera $[y:y + \text{diameter}; x:x + \text{diameter}]$

Tạo lớp mask từ lớp alpha của quả cầu: Những vị trí có hệ số alpha thấp thì phần background sẽ rõ hơn và ngược lại.

Tạo ảnh background: Ở vùng ảnh đã chọn trên camera, sử dụng hàm `bitwise_not()` trong OpenCV với lớp mask để những vị trí có hệ số alpha lớn thì phần background mờ đi và ngược lại.

Tạo ảnh foreground: Ở ảnh quả cầu, trái ngược với tạo ảnh background sử dụng hàm `bitwise_and()` để làm rõ quả cầu ở vị trí có hệ số alpha lớn và ngược lại.

Kết hợp hai ảnh foreground và background: Từ vị trí vùng ảnh đã chọn, tiến hành kết hợp hai ảnh foreground và background bằng hàm `add()` trong OpenCV.



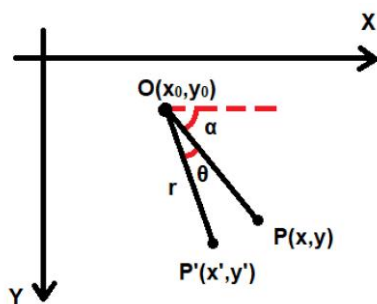
Hình 4. Minh họa cách ghép quả cầu với hình ảnh

2.2.3. Affine Transformations nhằm xoay quả cầu

Phép biến đổi Affine là phép biến đổi tuyến tính tọa độ điểm đặc trưng của đối tượng thành tập tương ứng các điểm mới để tạo ra các hiệu ứng cho từng đối tượng.

Sau khi thay quả cầu vào ảnh từ camera thành công, để hiệu ứng trở nên sinh động hơn, giống như phim ảnh hơn, chúng tôi tiến hành áp dụng phép biến đổi Affine nhằm xoay quả cầu theo 2 hướng ngược nhau với tốc độ phù hợp.

Giả sử chúng ta có một điểm tọa độ gốc $O(x_0, y_0)$ và một điểm $P(x, y)$ tạo với điểm O một góc α và khoảng cách r . Bây giờ chúng ta quay điểm P quanh gốc tọa độ O một góc θ theo chiều kim đồng hồ. Các tọa độ quay thu được như hình dưới đây:



Với: $x = x_0 + r \cos \alpha, y = y_0 + r \sin \alpha$

$$x' = x_0 + r \cos(\alpha + \theta)$$

$$= x \cos \theta - y \sin \theta + x_0(1 - \cos \theta) + y_0 \sin \theta$$

$$y' = y_0 + r \sin(\alpha + \theta)$$

$$= x \sin \theta + y \cos \theta + y_0(1 - \cos \theta) + x_0 \sin \theta$$

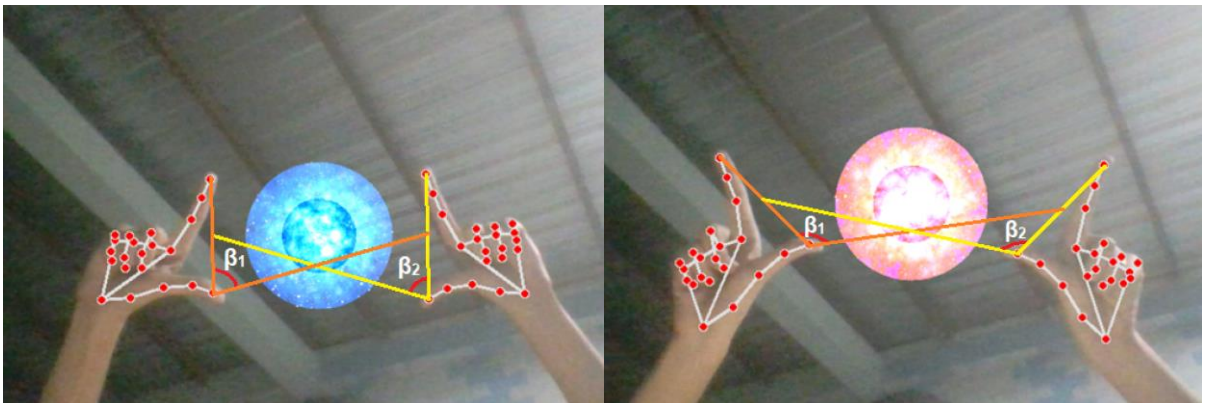
Trong thư viện OpenCV có hàm **getRotationMatrix2D()** hỗ trợ việc xoay ảnh được dễ dàng với các giá trị cần đưa vào lần lượt là vị trí gốc tọa độ O (tâm của quả cầu), góc theta và khoảng cách r (bán kính của quả cầu). Sau khi xử lý các giá trị qua hàm **getRotationMatrix2D()** ta được ma trận chuyển đổi M. Để có thể đảo chiều qua góc theta sẽ thay bằng $(360 - \text{theta})$.

Khi ma trận biến đổi M được tính toán, cần chuyển nó qua hàm **warpAffine()** để thực hiện phép biến đổi cho hình ảnh quả cầu. Ta lần lượt đưa vào các giá trị là hình ảnh quả cầu, ma trận biến đổi M và kích thước của hình ảnh quả cầu (đường kính x đường kính).

2.2.4. Chuyển màu quả cầu

Với việc hướng đến sự tương tác giữa người dùng và hiệu ứng, chúng tôi đã lên ý tưởng cho việc thay đổi màu quả cầu (xanh sang đỏ) thể hiện cho quả cầu năng lượng được tích tụ nhiều hơn.

Bước đầu tiên trong phương pháp này là xây dựng một hệ số tương tác (addpix). Hệ số này được tạo ra bằng cách tìm góc được tạo bởi 3 điểm (đầu ngón trỏ, đầu ngón cái và trung điểm của ngón trỏ và ngón cái ở tay khác) như hình 5.



Hình 5. Góc được tạo ra từ các điểm ở hai bàn tay

Khi xác định được 2 góc β_1, β_2 chúng tôi xây dựng hệ số tương tác như sau:

$$\text{addpix} = \left(\frac{|\cos(\beta_1)| + |\cos(\beta_2)|}{2} - 0.1 \right) \cdot 110$$

Với công thức được xây dựng như trên, $\text{addpix} \sim 0$ khi hai đoạn từ đầu ngón trỏ đến đầu ngón cái của hai bàn tay song song. Khi hai đoạn thẳng tạo góc càng lớn (phía trên hoặc phía dưới), addpix càng cao. Giá trị của addpix trong khoảng (0 – 110).

Sau khi xây dựng được hệ số addpix , chúng tôi tiến hành nghiên cứu chuyển từ màu xanh sang màu đỏ qua 3 kênh màu RGB. Sau nhiều lần thử nghiệm, để có thể chuyển đổi như thế cần giảm giá trị ở kênh màu Blue và tăng giá trị ở kênh màu Red. Từ đó, hiệu ứng sẽ chuyển dần từ xanh sang đỏ theo hệ số addpix .

Tiếp theo, để quả cầu có thể phóng ra năng lượng (effect nổ), chúng tôi đã xây dựng thêm thao tác chuyển đổi hình dạng với điều kiện như sau:

- Hệ số **$\text{addpix} > 90$**
- Xây dựng thêm biến đếm (count) kể từ lúc tạo quả cầu. Biến **$\text{count} = 50$**



Hình 6. Hình dạng biến đổi của quả cầu khi đủ điều kiện biến đổi

Khi thỏa mãn hai điều kiện quả cầu sẽ biến đổi như hình 6.

2.2.5. *Effect nổ quả cầu bằng Image Blending*

Đây là phần cuối cùng của hiệu ứng Energy Ball. Quả cầu năng lượng sau khi tích tụ và chuyển đổi hình dạng có thể phóng ra năng lượng bất cứ lúc nào. Để làm đa dạng sức mạnh của quả cầu, chúng tôi đã thiết kế hai thao tác để làm nổ quả cầu và tạo ra 2 effect khác nhau.

Đầu tiên là thao tác nắm hai tay lại quả cầu sẽ phát ra luồng ánh sáng trắng, giống hiệu ứng boom choáng nổ như hình 7.



Hình 7. Hiệu ứng nổ khi nắm hai tay

Thứ hai là thao tác bung hai bàn tay ra quả cầu sẽ phát nổ ra một luồng ánh sáng nhiều màu sắc như hình 8.



Hình 8. Hiệu ứng nổ khi bung hai tay

Để có được hai hiệu ứng như trên, chúng tôi đã tạo ra một video boom nổ trắng chỉ với hai màu trắng và đen.

Ở hiệu ứng đầu tiên, video và hình ảnh từ camera sẽ được kết với nhau ở kiểu dữ liệu int64 của numpy. Những vị trí có giá trị lớn hơn 255 sẽ được đổi thành 255.

Ở hiệu ứng thứ hai, video và hình ảnh từ camera sẽ được kết với nhau ở kiểu dữ liệu unit8. Khi đó những giá trị lớn 255 sẽ tự động chuyển đổi về số được chỉ lấy phần dư cho 255.

2.3. Kết quả và nhận xét

2.3.1. Kết quả đạt được

Chúng tôi đã tạo ra hiệu ứng hình ảnh về quả cầu năng lượng (Energy Ball). Hiệu ứng này con người có thể tương tác trực tiếp với quả bóng giống như ngoài thực tế. Hình ảnh tạo ra từ quả cầu khá chân thực và hoạt động khớp với thời gian thực.

2.3.2. Nhận xét

Đã thực hiện được đúng yêu cầu đặt ra từ ban đầu. Bên cạnh đó, còn mở rộng thêm khi ban đầu dự tính chỉ có một hiệu ứng nổ. Hiệu ứng khá đẹp mắt. Kết hợp nhiều phương pháp cho một sản phẩm. Quá trình nghiên cứu mất khá nhiều thời gian.

3. KẾT LUẬN

Effect Energy Ball là một hiệu ứng hình ảnh tái hiện lại những kỹ năng của nhiều nhân vật trong các bộ phim, các tựa game... Hiệu ứng đã mang lại cho người dùng một cảm giác khá giống các nhân vật trong các bộ phim. Hiệu ứng khá đẹp mắt, khá chân thực nhưng chỉ mới ở dạng 2D. Đây hứa hẹn sẽ là một hiệu ứng thú vị, thu hút người dùng trên các nền tảng công nghệ số hiện nay.

4. HƯỚNG PHÁT TRIỂN

Sản phẩm chỉ ở mức khá đẹp, khá chân thực và ở dạng 2D. Chúng tôi sẽ hướng đến một sản phẩm đẹp mắt nhất, chân thực nhất và có thể sẽ thể hiện quả cầu ở dạng 3D. Tạo thêm những cách tương tác với quả cầu và những hiệu ứng thú vị hơn.

TÀI LIỆU THAM KHẢO

- [1] Các bài tập trên course UIT
- [2] <https://google.github.io/mediapipe/solutions/hands.html>
- [3] https://docs.opencv.org/4.x/d0/d86/tutorial_py_image_arithmetics.html
- [4] <https://theailearner.com/tag/cv2-getrotationmatrix2d/>
- [5] <https://www.mathvn.com/2019/12/cong-thuc-tinh-goc-giua-hai-vecto-trong.html>
- [6] https://vi.wikipedia.org/wiki/M%C3%B4_h%C3%A0nh_m%C3%A0u_RGB