

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



**BÁO CÁO ĐỒ ÁN**  
**NHẬP MÔN THỊ GIÁC MÁY TÍNH**

**Đề tài Nhóm: Hand effects**

**Đề tài Cá nhân: Firework hand effect**

**Giảng viên hướng dẫn:** TS. NGUYỄN VINH TIỆP  
**Sinh viên thực hiện:** NGUYỄN HOÀNG MINH 20521609  
**Lớp:** CS231.N11

**Nhóm sinh viên:**  
NGUYỄN HOÀNG MINH 20521609  
TẠ NHẬT MINH 20521614  
NGUYỄN THIỆN THUẬT 20521998  
NGUYỄN MINH TIẾN 20522010

# MỤC LỤC

<b>Phần I. Giới thiệu bài toán .....</b>	<b>2</b>
1. Tổng quan bài toán Hand effects .....	2
2. Bài toán Firework hand effect .....	2
2.1. Mô tả bài toán .....	2
2.2. Xác định yêu cầu .....	2
2.3. Mục tiêu cần đạt được .....	2
<b>Phần II. Phương pháp tiếp cận bài toán .....</b>	<b>3</b>
1. Ý tưởng .....	3
2. Hướng giải quyết .....	3
3. Thách thức của đề tài .....	3
<b>Phần III. Giải quyết bài toán .....</b>	<b>4</b>
1. Thư viện MediaPipe Hands .....	4
2. Transparent image .....	4
3. Giải pháp .....	5
<b>Phần IV. Cài đặt và thử nghiệm kết quả (demo) .....</b>	<b>6</b>
<b>Phần V. Kết luận .....</b>	<b>7</b>
<b>Phần VI. Tài liệu tham khảo .....</b>	<b>7</b>

## Phần I. Giới thiệu bài toán

### 1. Tổng quan bài toán Hand effects

Ngày nay, con người đã ứng dụng rất nhiều bài toán về nhận biết hình dạng và chuyển động của bàn tay trong thời gian thực nhằm tạo ra những sản phẩm và các tương tác trên nền tảng công nghệ trong nhiều lĩnh vực, mục đích nhằm làm tăng trải nghiệm của người dùng và giúp con người mô phỏng thực tế lên không gian ảo tốt hơn. Ví dụ như có thể dùng cho việc hiểu ngôn ngữ ký hiệu, điều khiển công nghệ thông qua cử chỉ tay, các bài toán về thực tế tăng cường (AR),...

Bài toán hand effect và các bài toán khác về face effect hay pose effect cũng được ứng dụng rất nhiều trong các filter trên các mạng xã hội như Facebook, Instagram, TikTok,... nhằm tạo ra những tương tác thú vị hơn cho người dùng trên các ứng dụng của họ. Hand effect sử dụng các thuật toán hand-tracking nhằm xác định các điểm trên tay của người dùng thông qua webcam để từ đó có thể thêm nhiều effect khác nhau giúp họ có thể tương tác và chơi được với chúng.

Lấy cảm hứng và ý tưởng từ các filter này, nhóm đã chọn đề tài về hand effect để xây dựng nhiều filter khác nhau từ chúng.

### 2. Bài toán Firework hand effect

#### 2.1. Mô tả bài toán

Bài toán firework hand effect là một hiệu ứng được đề xuất xây dựng theo đề tài hand effect với mong muốn tạo được một filter bắn pháo hoa từ các tương tác tay của người dùng thông qua webcam.

Hiệu ứng sử dụng hand-tracking nhằm xác định vị trí của bàn tay từ webcam theo thời gian thực làm cơ sở để lập trình, giúp các bàn tay có thể thao tác được với các hiệu ứng trên màn hình như cầm pháo, đốt pháo và cho nổ pháo.

#### 2.2. Xác định yêu cầu

Hiệu ứng cần đảm bảo các yêu cầu sau nhằm mô tả đúng quá trình bắn pháo hoa đồng thời phải phù hợp với yêu cầu của môn học:

- Phải có 2 dụng cụ pháo và quẹt để đốt pháo.
- Trong quá trình đốt, quẹt phải đốt vào dây của pháo và năng lượng sẽ giảm khi nguồn nhiệt ở xa.
- Khi cho nổ pháo, pháo phải đủ năng lượng và đảm bảo tay đã buông quả pháo.
- Thực hiện xử lý ảnh không thông qua ngữ nghĩa của ảnh.
- Sử dụng các kiến thức xử lý ảnh đã được học trong môn học (blending,...).

#### 2.3. Mục tiêu cần đạt được

- Xây dựng được hiệu ứng bắn pháo hoa (firework effect) sử dụng bàn tay thời gian thực thông qua ảnh từ webcam.
- Ứng dụng được các kiến thức về xử lý ảnh cơ bản đã được học để làm đề tài.

## Phần II. Phương pháp tiếp cận bài toán

### 1. Ý tưởng

Đề tài chia bài toán lớn Firework hand effect thành 3 bài toán để giải quyết. Mỗi bài toán này sẽ xử lý từng phần nhỏ hơn của một quy trình bắn pháo hoa:

**Bài toán 1 – Xử lý cầm pháo và quẹt ở hai bàn tay:** phần này sẽ xử lý phần hiệu ứng 2 tay xuất hiện pháo và quẹt.

**Bài toán 2 – Xử lý đốt pháo:** phần này sẽ xử lý đưa pháo lại gần quẹt để đốt pháo và tích tụ năng lượng cho pháo.

**Bài toán 3 – Xử lý nổ pháo:** phần này sẽ xử lý việc nổ của quả pháo hoa.

### 2. Hướng giải quyết

- Xác định các điểm trên bàn tay và tính tỉ lệ các khoảng cách giữa chúng.
- Đặt điều kiện các tỉ lệ nằm trong khoảng nhất định để phù hợp với thể tay của hiệu ứng (cầm, đốt,...).

- Sử dụng transparent image để chèn các effect lên trên ảnh của webcam.

**Bài toán 1: Ta xác định các tỉ lệ của ngón tay đến cổ tay ứng với mỗi vật**

- Tay giờ nắm đấm sẽ quy định là quả pháo: khoảng cách từ các ngón tay tới cổ tay sẽ thấp nên tỉ lệ cũng sẽ thấp.
- Tay giờ giống hình nút like sẽ quy định là cái quẹt: ngón cái giờ cao nên khoảng cách từ các ngón tay tới cổ tay sẽ cao hơn và tỉ lệ cũng sẽ cao.

**Bài toán 2: Ta xác định tỉ lệ của khoảng cách giữa 2 vật ở 2 tay khác nhau**

- Hai tay để càng gần thì tỉ lệ khoảng cách sẽ càng thấp.
- Tay quẹt để dưới pháo thì tọa độ bàn tay của quẹt sẽ cao hơn pháo trên ảnh.
- Sử dụng biến count\_energy để làm thời gian tích tụ năng lượng.

**Bài toán 3: Ta xác định tỉ lệ của ngón tay cầm pháo đến cổ tay**

- Tỉ lệ này sẽ lớn nếu ta thực hiện buông quả pháo.

### 3. Thách thức của đề tài

**Bài toán 1:**

- Nhận dạng được cả 2 bàn tay đang giờ vật gì (tay trái cầm quẹt tay phải cầm pháo, tay trái cầm pháo tay phải cầm quẹt đều phải xử lý được).
- Xác định được tay nào là tay pháo, tay nào là tay quẹt (cho máy tính nhận dạng được 2 vật đó là 2 vật riêng biệt từ 2 bàn tay giống nhau).

**Bài toán 2:**

- Thỏa mãn đã xuất hiện đủ cả 2 dụng cụ pháo và quẹt.
- Thỏa mãn pháo nằm trên quẹt và quẹt phải đưa sát vào pháo.
- Quẹt để xa pháo thì không tích tụ năng lượng và nếu có năng lượng được đốt từ trước thì năng lượng đó sẽ giảm.

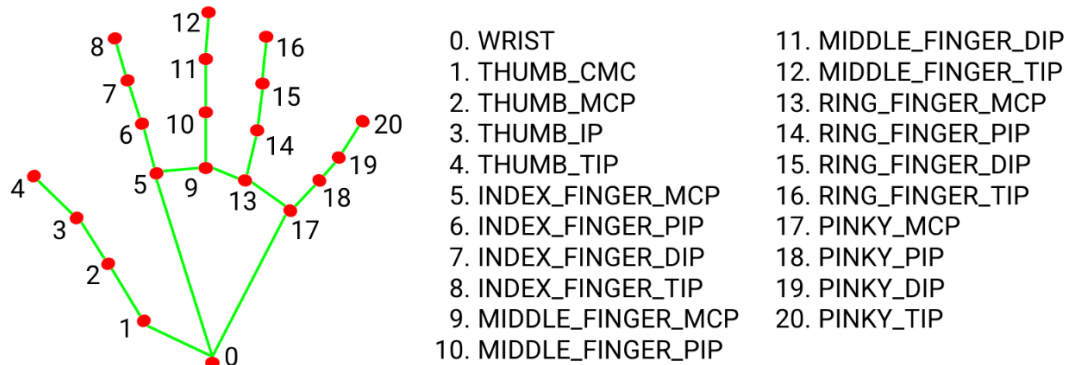
**Bài toán 3:**

- Tìm được phương pháp xử lý ảnh để chèn effect từ file mp4.

## Phần III. Giải quyết bài toán

### 1. Thư viện MediaPipe Hands

MediaPipe Hands là một giải pháp được GOOGLE đề xuất giúp theo dõi ngón tay từ bàn tay đạt độ chính xác cao thông qua ảnh. Nó sử dụng công nghệ máy học (Machine Learning) để suy ra 21 điểm 3D của một bàn tay chỉ từ một khung hình duy nhất. Trong khi các phương pháp tiên tiến hiện nay chủ yếu dựa vào sức mạnh của môi trường máy tính để suy luận, thì giải pháp của họ đạt được hiệu suất thời gian thực trên cả điện thoại di động và thậm chí đã xử lý được cho nhiều bàn tay.



Hình 1. 21 hand landmarks

### 2. Transparent image

Transparent image là một tấm ảnh được xóa nền một (hoặc toàn bộ) phần và phần đó sẽ hoàn toàn “trong suốt”. Vì vậy, transparent image có thể dùng để chèn vào tấm ảnh khác.

Transparency chỉ được sử dụng trong một số loại file (ví dụ: GIF, PNG,...) bằng cách sử dụng một Transparent color hoặc sử dụng Alpha channel (Alpha blending). Transparency có thể được sử dụng trên Raster image (hay digital image) và Vector image.

Chúng ta chỉ đề cập về Transparency trên Raster image, hay cụ thể hơn là Digital image. Với mỗi pixel được biểu diễn trong một hệ màu, sẽ có một giá trị được định nghĩa là “giá trị trong suốt” và nó không phải là một màu trong thực tế. Nếu bộ giải mã màu sắc gặp pixel mang giá trị “trong suốt” này, pixel đó sẽ được thể hiện bằng màu của tấm ảnh được transparent image chèn vào.



Hình 2. Transparent image của quẹt với phần background đã được xóa

### 3. Giải pháp

#### Các thư viện sử dụng

- Sử dụng thư viện MediaPipe Hands để xác định các điểm của ngón tay thông qua ảnh đầu vào của webcam.
- Sử dụng Transparent Image để chèn vào ảnh để làm các hiệu ứng.
- Ngoài ra còn sử dụng thư viện OpenCV và numpy để thực hiện các xử lý với ảnh.

#### Phương pháp xử lý ảnh transparent image

Mã giả:

---

##### Algorithm 4: Xử lý chèn transparent image vào ảnh webcam

---

```
def transparent(targetImg, x, y):
    newFrame = img.copy()
    h, w, _ = targetImg.shape
    roi = newFrame[y:y + h, x:x + w]
    img2gray = cv2.cvtColor(targetImg, cv2.COLOR_BGR2GRAY)
    _, mask = cv2.threshold(img2gray, 0, 255, cv2.THRESH_BINARY)
    img1_bg = cv2.bitwise_and(roi.copy(), roi.copy(), mask = cv2.bitwise_not(mask))
    img2_fg = cv2.bitwise_and(targetImg, targetImg, mask = mask)
    newFrame[y:y + h, x:x + w] = cv2.add(img1_bg, img2_fg)
    return newFrame
```

\*Xác định vùng của ảnh gốc có ảnh cần chèn\*

\*BG (ảnh webcam)\*

\*FG (pháo, quẹt)\*

\*Kết hợp background và foreground\*

---

#### Bài toán 1 – Xử lý cầm pháo và quẹt ở hai bàn tay

Mã giả: (Tên các biến tương ứng với các điểm trong Hình 1)

---

##### Algorithm 1: Xử lý hiệu ứng tay cầm pháo và quẹt

---

```
hands = mediapipe.hands.process(image)
for hand in hands:
    mcp_index_dis = calculate_distance(wrist, index_mcp)
    mcp_middle_dis = calculate_distance(wrist, middle_mcp)
    mcp_ring_dis = calculate_distance(wrist, ring_mcp)
    mcp_pinky_dis = calculate_distance(wrist, pinky_mcp)
    distance_avg = (mcp_index_dis + mcp_middle_dis + mcp_ring_dis + mcp_pinky_dis) / 4
    tip_thumb_index_dis = calculate_distance(index_mcp, thumb_tip)
    ratio_dis = tip_thumb_index_dis / distance_avg
    if ratio_dis >= 0.75:
        Vẽ "quẹt"
    if ratio_dis < 0.5:
        Vẽ "pháo"
```

\*Tìm các điểm của bàn tay từ ảnh\*

\*Tính khoảng cách các điểm mcp tới wrist (trừ thumb)\*

\*Tính trung bình khoảng cách của 4 ngón\*

\*Tính khoảng cách điểm 5 tới điểm 4\*

\*Tính tỉ lệ\*

\*Vẽ quẹt và pháo vào webcam bằng phương pháp xử lý ảnh transparent image\*

---

**Bài toán 2 – Xử lý đốt pháo**

Mã giả: (Tên các biến tương ứng với các điểm trong Hình 1)

**Algorithm 2: Xử lý hiệu ứng đốt cháy khi quẹt để sát pháo**


---

```

if count_hand = 2 & thumb_tip_quet ≠ 0 & index_mcp_quet ≠ 0 &
pinky_mcp_phao ≠ 0 & wrist_phao ≠ 0:                                *Điều kiện xuất hiện cả pháo và quẹt*
    if y_thumb_tip_quet > y_pinky_mcp_phao & y_index_mcp_quet >
y_wrist_phao:                                                         *Điều kiện pháo nằm trên quẹt*
        if ((x_thumb_tip_quet + x_thumb_tip_quet)/2 > x_pinky_mcp_phao
& (x_thumb_tip_quet + x_thumb_tip_quet)/2 < x_wrist_phao) or
((x_thumb_tip_quet + x_thumb_tip_quet)/2 < x_pinky_mcp_phao
and (x_thumb_tip_quet + x_thumb_tip_quet)/2 > x_wrist_phao):
            X = diameter_phao + diameter_quet                                *Xử lý quẹt để sát pháo*
            Y = abs(y_thumb_tip_quet - y_pinky_mcp_phao) +
abs(y_index_mcp_quet - y_wrist_phao)
            if (0.25 < Y/X < 0.85):
                count_energy += 1
                Vẽ “flare”                                                *Vẽ flare vào webcam bằng phương pháp xử lý ảnh transparent image*

    elif count_energy > 1: count_energy = count_energy – 1
if count_energy >= 40: phao_no = True                                *Xử lý quẹt để xa pháo*

```

---

**Bài toán 3 – Xử lý nổ pháo**

Mã giả: (Tên các biến tương ứng với các điểm trong Hình 1)

**Algorithm 3: Xử lý hiệu ứng nổ pháo khi đủ năng lượng**


---

```

def sum2img(img1, img2):                                              *Hàm cộng 2 ảnh*
    img = np.int64(img1) + np.int64(img2)
    img[img>255] = 255
    img = np.uint8(img)
    return img
if (phao_no == True) & (tip_index_dis_phao > 150):                  *Điều kiện pháo đủ năng lượng và tay pháo đã buông*
    count_energy = 0
    ret_, firework = vid.read()                                        *Xử lý file mp4 (file hiệu ứng pháo hoa) chèn vào webcam*
    img = sum2img(img, firework)

```

---

**Phần IV. Cài đặt và thử nghiệm kết quả (demo)**

Hình 3. Effect dụng cụ



Hình 4. Effect đốt pháo



Hình 5. Effect nổ pháo

- Video hướng dẫn demo effect được đính kèm chung với file báo cáo

## Phần V. Kết luận

### **Kết quả đạt được**

- Xây dựng được hiệu ứng bắn pháo hoa (firework effect) sử dụng bàn tay thời gian thực thông qua ảnh từ webcam.
- Ứng dụng được các kiến thức về xử lý ảnh cơ bản đã được học để làm đề tài.

### **Hạn chế**

- Thuật toán chưa được tối ưu và chưa được đóng gói (function, module hóa) hiệu quả.
- Giải pháp phải xử lý thông qua nhiều biến trung gian.

### **Tổng kết**

Trong đồ án này, đề tài đã đưa ra được phương pháp tạo firework hand effect thông qua xử lý ảnh và hand-tracking, giải pháp được đưa ra cho kết quả khá thụ vị và đã đạt được yêu cầu mục tiêu ban đầu. Từ những kết quả này, đề tài có thể được tối ưu và phát triển thêm để có thể đẹp hơn và hoạt động hiệu quả hơn nữa trong tương lai.

Đối với các hiệu ứng khác thuộc đề tài Hand effect, nhóm cũng đã đưa ra được nhiều effect và giải pháp khác nhau để xây dựng effect từ hand-tracking: Energy ball effect, God of thunder effect, High Lightning Effect. Các hiệu ứng này đã giúp cho nhóm sử dụng được các kỹ thuật xử lý ảnh để ứng dụng giải quyết vấn đề của đề tài.

## Phần VI. Tài liệu tham khảo

- [1] <https://google.github.io/mediapipe/solutions/hands.html> (last access: 30/12/2022)
- [2] <https://www.analyticsvidhya.com/blog/2022/03/hand-landmarks-detection-on-an-image-using-mediapipe/> (last access: 23/12/2022)
- [3] [https://en.wikipedia.org/wiki/Transparency\\_\(graphic\)](https://en.wikipedia.org/wiki/Transparency_(graphic)) (last access: 23/12/2022)
- [4] <https://stackoverflow.com/questions/40895785/using-opencv-to-overlay-transparent-image-onto-another-image> (last access: 27/12/2022)
- [5] <https://www.youtube.com/watch?v=LffHViW9mYA&t=1719s> (last access: 29/12/2022)
- [6] <https://www.youtube.com/watch?v=v7bw6ThEEqo> (last access: 29/12/2022)
- [7] <https://www.youtube.com/watch?v=QER11-ckFr8&t=2293s> (last access: 29/12/2022)