

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN



# PROJECT REPORT

## ĐỀ TÀI: BABY FACE GENERATOR

❖ **Họ và tên các thành viên:**

1/ Dương Đăng Khoa      MSSV: 1512251

2/ Đặng Thành Phát      MSSV: 1512390

❖ **Thông tin giảng viên:**

GVLT: Trần Thái Sơn

GVTH: Võ Hoài Việt

*Ngày 10 tháng 06 năm 2018*

# MỤC LỤC

<b>1</b>	<b>Thông tin nhóm .....</b>	<b>3</b>
<b>2</b>	<b>Phát biểu bài toán sơ lược.....</b>	<b>3</b>
<b>3</b>	<b>Giải pháp đề xuất.....</b>	<b>4</b>
<b>4</b>	<b>Kế hoạch phát triển .....</b>	<b>5</b>
<b>5</b>	<b>Bảng phân công công việc .....</b>	<b>6</b>
<b>6</b>	<b>Báo cáo chi tiết triển khai đề án .....</b>	<b>7</b>
<b>7</b>	<b>Hướng dẫn thực thi chương trình.....</b>	<b>11</b>
<b>8</b>	<b>Kết quả khi chạy chương trình.....</b>	<b>12</b>
<b>9</b>	<b>Tham khảo .....</b>	<b>13</b>

# 1

## Thông tin nhóm

MSSV	Họ Tên	Email	Điện thoại
1512251	Dương Đăng Khoa	khoadangduong63@gmail.com	0961604198
1512390	Đặng Thành Phát	dtphat.itus@gmail.com	0964427065

# 2

## Phát biểu bài toán sơ lược

- *Bài toán Phát sinh khuôn mặt trẻ con từ hai ảnh bố mẹ.*
  - + Input: Người dùng nhập vào 2 ảnh của cha và của mẹ, trên mỗi bức ảnh chỉ có duy nhất một khuôn mặt.
  - + Output: Chương trình sẽ phát sinh ra khuôn mặt người con có nét giống với khuôn mặt của cha và mẹ

# 3

## Giải pháp đề xuất

### 3.1 Phần mềm

Danh sách các chức năng phần mềm:

- Nhập ảnh bố từ tập ảnh Father
- Nhập ảnh mẹ từ tập ảnh Mother
- Nhập hệ số alpha để biến đổi (hệ số  $\alpha * 100\%$  là tỉ lệ phần trăm giống cha, và  $(1 - \alpha) * 100\%$  là tỉ lệ phần trăm giống mẹ với  $0 < \alpha < 1$ )
- Xuất ảnh con (còn nhỏ)
- Xuất ảnh con lúc trưởng thành

### 3.2 Phần cứng

- Máy tính hoạt động trên môi trường hoạt động: Windows
- Thư viện nguồn mở kèm theo: OpenCV, Dlib. Link download thư viện:
  - + OpenCV: [https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.4.1/opencv-3.4.1-vc14\\_vc15.exe/download](https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.4.1/opencv-3.4.1-vc14_vc15.exe/download)
  - + Dlib: <https://sourceforge.net/projects/dclib/files/dlib/v19.7/>

# 4

## Kế hoạch phát triển

### 4.1 *Phân tích yêu cầu*

- Tìm hiểu thư viện OpenCV, dlib để thực hiện các chức năng phát hiện và lai ghép khuôn mặt.
- Tìm hiểu các điểm phát sinh sau khi phát hiện khuôn mặt.
- Tìm hiểu thuật toán lai ghép giữa 2 khuôn mặt.

### 4.2 *Cài đặt (implement) phần mềm*

- Phân chia công việc và xây dựng phần mềm theo các thuật toán và các hàm thư viện đã tìm hiểu.

### 4.3 *Kiểm thử phần mềm*

- Thu thập tập ảnh bố mẹ trên Internet và tiến hành kiểm thử.

# 5

## Bảng phân công công việc

STT	Giai đoạn	MSSV	Họ Tên	Chi tiết	Thời hạn
1	Tìm hiểu bài toán và các thuật toán, hàm thư viện liên quan	1512251	Dương Đăng Khoa	Tìm hiểu các hàm trong thư viện	2 tuần
		1512390	Đặng Thành Phát	OpenCV, dlib, đọc bài báo Face Morphing	2 tuần
2	Viết proposal	1512251	Dương Đăng Khoa	Viết Project proposal	1 ngày
3	Viết Prototype lần 1	1512251	Dương Đăng Khoa	Viết Prototype	2 ngày
		1512390	Đặng Thành Phát	Kiểm tra, bổ sung chỉnh sửa Prototype	2 ngày
4	Cài đặt chương trình	1512251	Dương Đăng Khoa	Cài đặt và nghiên cứu hàm morphing	4-5 tuần
		1512390	Đặng Thành Phát	Cài đặt và nghiên cứu hàm phát sinh điểm ảnh và tam giác	4-5 tuần
5	Kiểm thử phần mềm	1512251	Dương Đăng Khoa	Sưu tầm tập ảnh bố từ f1-f25 và tập ảnh mẹ m1-m25 và kiểm tra phát sinh ảnh con	1 tuần
		1512390	Đặng Thành Phát	Sưu tầm tập ảnh bố từ f27-f50 và m26-m50 và kiểm tra phát sinh ảnh con	1 tuần

6	Viết Prototype lần 2	1512251	Dương Đăng Khoa	Viết Prototype	2 ngày
		1512390	Đặng Thành Phát	Kiểm tra, bổ sung chỉnh sửa Prototype	2 ngày
7	Nộp sản phẩm và viết báo cáo	1512251	Dương Đăng Khoa	Viết báo cáo chi tiết, kiểm tra tổng thể và nộp sản phẩm	2 tuần
		1512390	Đặng Thành Phát	Viết báo cáo chi tiết và kiểm tra, bổ sung, chỉnh sửa	2 tuần

## 6 Báo cáo chi tiết triển khai đồ án

Như đã nêu trên, bài toán phát sinh khuôn mặt trẻ em sẽ thực hiện bằng cách tìm ra điểm đặc trưng của 2 khuôn mặt bố mẹ và sau đó pha trộn lại với nhau và theo tỉ lệ của khuôn mặt 1 đưa trẻ có sẵn ta sẽ thu được hình ảnh khuôn mặt trẻ em theo yêu cầu. Ta phân thành 2 bài toán nhỏ là Image Morphing và biến khuôn mặt theo tỉ lệ trẻ em.

Các công đoạn biến đổi: Trong đó Stage 1, 2, 3 sẽ giải quyết bài toán 1 – Image Morphing, Stage 4 sẽ giải quyết bài toán 2 – Biến đổi khuôn mặt theo tỉ lệ trẻ em.

- + Stage 1: Tìm điểm tương ứng bằng cách sử dụng tính năng phát hiện khuôn mặt
- + Stage 2: Phát sinh thêm tập dữ liệu Delaunay Triangulation
- + Stage 3: Tính toán vị trí các bộ phận khuôn mặt và pha trộn màu ảnh
- + Stage 4: Biến đổi ảnh theo tỉ lệ của khuôn mặt đưa trẻ

### ❖ Bài toán 1: Image Morphing

Ý tưởng đằng sau Image Morphing khá đơn giản. Với hai hình ảnh I và J, chúng ta muốn tạo một ảnh giữa M bằng cách pha trộn hình ảnh I và J. Sự pha trộn hình ảnh I và J được điều khiển bởi tham số alpha là giữa 0 và 1 ( $0 \leq \alpha \leq 1$ ). Khi alpha là 0, ảnh M sẽ trông giống như mẹ, và khi alpha là 1, M trông giống hệt như bố. Ta sẽ sử dụng phương trình sau áp dụng lên mỗi điểm ảnh (x, y) của M:

$$M(x, y) = (1 - \alpha) I(x, y) + \alpha J(x, y)$$

Tuy nhiên, bằng cách sử dụng phương trình trên để tạo ra một sự pha trộn giữa hình ảnh của Bộ trưởng Hillary Clinton và Thượng nghị sĩ Ted Cruz với thiết lập là 0,5, ta sẽ nhận được kết quả như sau đây.



Vì vậy, để biến hình ảnh I thành hình ảnh J, trước tiên chúng ta cần thiết lập tương ứng pixel giữa hai hình ảnh. Nói cách khác, đối với mỗi điểm ảnh  $(x_i, y_i)$  trong ảnh I, chúng ta cần tìm pixel tương ứng  $(x_j, y_j)$  trong ảnh J. Giả sử chúng ta đã tìm thấy những sự tương ứng một cách kỳ diệu, chúng ta có thể trộn hình ảnh theo hai bước. Đầu tiên, chúng ta cần tính toán vị trí  $(x_m, y_m)$  của pixel trong hình ảnh biến đổi.

Và sau đây là từng bước biến đổi:

### Stage 1: Tìm điểm tương ứng bằng cách sử dụng tính năng phát hiện khuôn mặt

#### 1.1/ Phát hiện 68 điểm đặc trưng:

- Cách thực hiện:

+ B1: Kiểm tra ảnh đầu vào có nhiều hơn một khuôn mặt hay không, nếu nhiều hơn thì trả về -1 (báo lỗi), ngược lại tiếp tục chương trình

+ B2: Lưu 68 điểm đặc trưng vào một vector

- Mục đích: Chuẩn bị 68 điểm đặc trưng khuôn mặt của mỗi bức ảnh

- Hàm thực hiện: `int getFeaturePoints(std::vector<Point2i> &PointList, string ImgName)`



## 1.2/ Phát sinh thêm điểm dữ liệu

- Cách thực hiện:

+ B1: Gán  $\Delta$  = Tọa độ trục tung của điểm thứ 36 - Tọa độ trục tung của điểm thứ 18

+ B2: Thực hiện lặp từ điểm thứ 16 đến 27 để phát sinh thêm các điểm theo công thức:

Hoành độ của điểm mới = hoành độ của điểm đang xét

Tung độ của điểm mới = tung độ của điểm đang xét -  $\Delta$

- Mục đích: Phát sinh thêm 12 điểm dữ liệu và những điểm dữ liệu được thêm vào là những điểm thuộc vùng trăn của khuôn mặt

- Hàm thực hiện: `void getMorePoints(std::vector<Point2i> &PointList)`

## Stage 2: Delaunay Triangulation

### 2.1/ Khái niệm Delaunay Triangulation:

- Cho 1 tập hợp các điểm trong một mặt phẳng, chúng ta có thể vẽ tam giác để phân chia mặt phẳng thành các hình tam giác với các điểm là đỉnh. Một tập hợp các điểm có thể tạo được nhiều hình tam giác nhưng Delaunay Triangulation là tốt nhất vì nó có thuộc tính tốt đẹp. Trong 1 tam giác Delaunay, hình tam giác được chọn sao cho không có điểm nào của tam giác khác nằm trong đường tròn ngoại tiếp của tam giác đang xét.

### 2.2/ Mục đích sử dụng Delaunay Triangulation:

- Một thuộc tính thú vị của tam giác Delaunay là nó không ưu tiên hình tam giác “gầy” (tức là hình tam giác với một góc lớn).

- Thuật toán Delaunay Triangulation được tạo thành từ nhiều giải pháp khác nhau nhưng giải pháp tạo thành tam giác có các cạnh cách đều 2 đỉnh tương ứng

=> Mục đích chia ảnh thành lưới tam giác để chuẩn bị cho các phép biến đổi sau

### 2.3/ Cách lưu trữ

- Là một tập tin text với mỗi dòng là 1 bộ ba đỉnh của tam giác gồm 132 dòng.

- Tên file lưu trữ Delaunay Triangulation trong project: Triangulation.txt

## Stage 3: Tính toán vị trí các bộ phận khuôn mặt và pha trộn màu ảnh

### 3.1/ Tìm vị trí các điểm đặc trưng trong Morphed Image

- Cách thực hiện:

+ B1: Từ các bước trên ta đã có 2 bộ dữ liệu điểm đặc trưng của ảnh bố và mẹ. Tính tọa độ các điểm đặc trưng cho ảnh Morphed Image từ tập dữ liệu của ảnh bố và ảnh mẹ theo công thức:

$$x_m = (1 - \alpha) * x_i + \alpha * x_j$$

$$y_m = (1 - \alpha) * y_i + \alpha * y_j$$

+ B2: Sau khi kết thúc ta được tập 80 điểm đặc trưng cho ảnh Morphed Image

- Hàm thực hiện:

```
for (int i = 0; i < FatherPoint.size(); i++)
{
    float X, Y;
    X = Alpha * FatherPoint[i].x + (1 - Alpha) * MotherPoint[i].x;
    Y = Alpha * FatherPoint[i].y + (1 - Alpha) * MotherPoint[i].y;
    Points.push_back(Point2f(X, Y));
}
```

### 3.2/ Tính toán tọa độ điểm bằng phép biến đổi Affine:

- Hiện tại chúng ta đã có 3 bộ dữ liệu đặc trưng tương ứng với ảnh cha, ảnh mẹ và ảnh Morphed Image

- Chúng ta cũng biết được các tam giác được tạo thành từ các bộ dữ liệu đặc trưng.

- Chọn một hình tam giác trong hình ảnh bố và hình tam giác tương ứng trong Morphed Image và tính toán biến đổi affine ánh xạ ba góc của tam giác trong hình bố đến ba góc của tam giác tương ứng trong hình Morphed Image. Trong OpenCV, điều này có thể được thực hiện bằng cách sử dụng `getAffineTransform`, tính toán biến đổi cho toàn bộ 132 tam giác

- Biến đổi tương tự với ảnh mẹ và ảnh Morphed Image.

### 3.3/ Biến đổi các vùng tam giác bằng Warp triangles:

- Đối với mỗi tam giác trong ảnh bố, sử dụng biến đổi affine được tính toán trong bước trước để chuyển đổi tất cả các pixel bên trong hình tam giác thành Morphed Image. Lặp lại điều này cho tất cả các hình tam giác trong ảnh bố, để có được một ma trận ảnh biến đổi tất cả pixel của ảnh bố và Morphed Image. Tương tự, có được phiên bản bị biến dạng cho hình ảnh mẹ. Trong OpenCV, điều này đạt được bằng cách sử dụng hàm `warpAffine`.

### 3.4/ Pha trộn màu cho ảnh Morphed Image:

- Trong các công đoạn trước, chúng ta nhận được 2 ma trận bị biến dạng của ảnh bố và ảnh mẹ

- Hai hình ảnh này có thể được pha trộn bằng cách sử dụng phương trình dưới đây:

$$M(x_m, y_m) = (1 - \alpha) I(x_i, y_i) + \alpha J(x_j, y_j)$$

- Ta thu được một hình ảnh có điểm đặc trưng của bố và mẹ

## ❖ Bài toán 2: Biến đổi ảnh theo tỉ lệ khuôn mặt trẻ em

### Stage 4: Phát sinh ảnh con

#### 4.1/ Tạo mask và loại bỏ những phần dư thừa

- Tạo một ma trận mặt nạ Mask. Cách tạo:

+ B1: Tạo 1 vector lưu tọa độ các điểm đặc trưng của ảnh Morphed Image

+ B2: Dùng hàm fillConvexPoly để tạo một ma trận ảnh chứa đa giác được tạo thành từ vector của B1. Ta được ma trận Mask

- Loại bỏ những phần dư thừa bằng cách nhân ma trận Mask với ma trận hình ảnh đặc trưng của bố và mẹ

#### 4.2/ Tạo ảnh kết quả:

- Từ vector lưu tọa độ các điểm đặc trưng của ảnh Baby được đưa vào làm kích thước chuẩn, tạo một khung nhờ hàm boundingRect để chép ảnh con vào ta được ảnh kết quả là ảnh có đặc trưng của bố và mẹ theo tỉ lệ khuôn mặt trẻ con.

# 7 Hướng dẫn thực thi chương trình

**Bước 1:** Mở Release và chạy chương trình FinalProject.exe

**Bước 2:** Nhập tên ảnh của bố

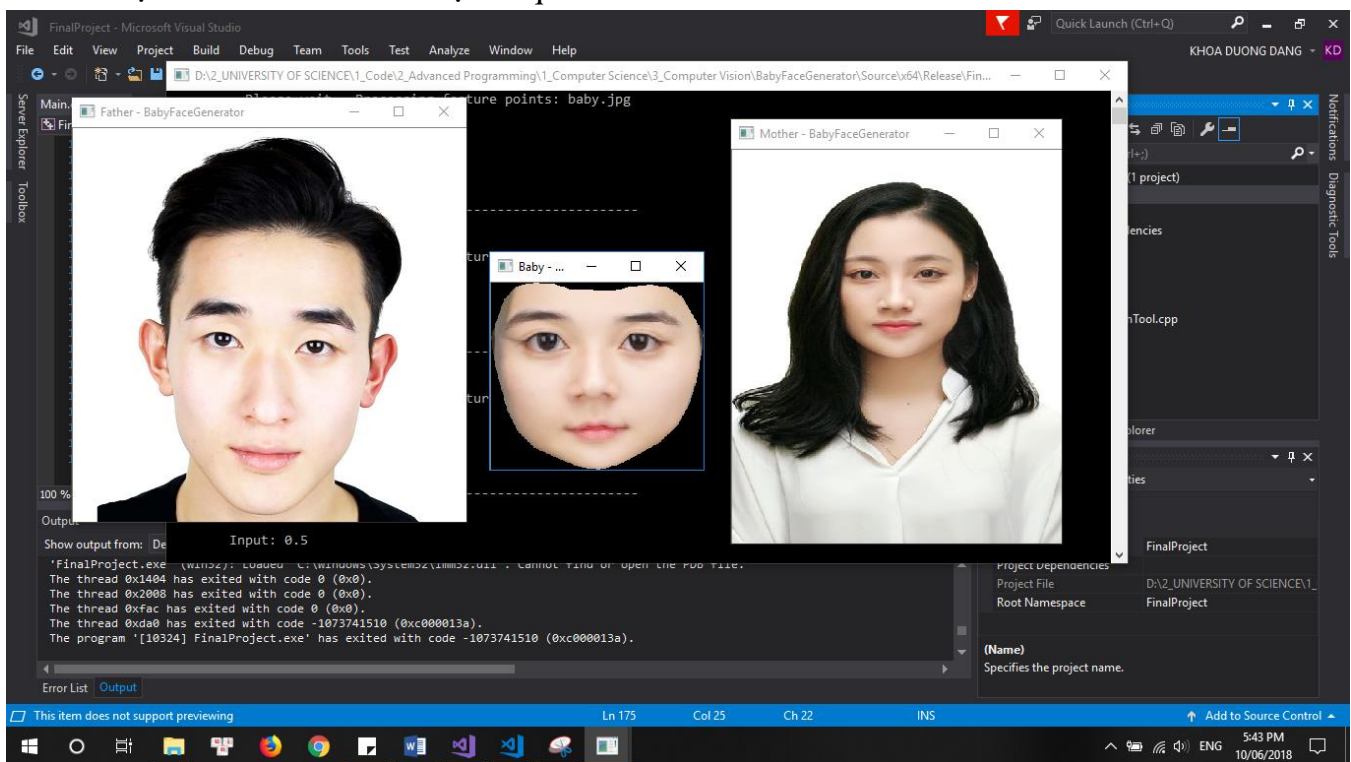
**Bước 3:** Nhập tên ảnh của mẹ

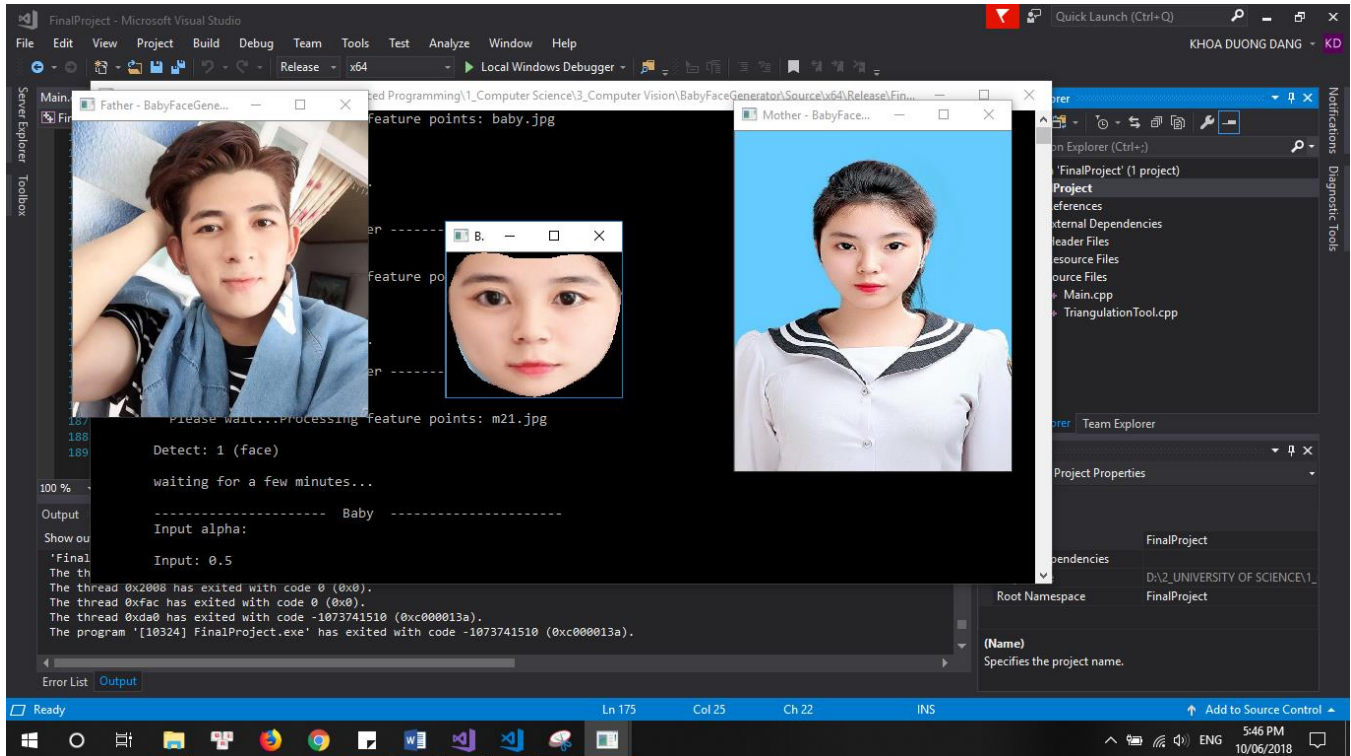
**Bước 4:** Nhập hệ số alpha (alpha càng nhỏ càng giống mẹ, càng lớn càng giống bố)

# 8

## Kết quả khi chạy chương trình

- Tất cả dữ liệu ảnh của bố được lưu ở tập tin Father, dữ liệu ảnh của mẹ lưu ở tập tin Mother bên trong Project.
- Tất cả kết quả hình ảnh sau khi chạy thử chương trình đã được chép vào tập tin Baby bên trong Project.
- Một số hình ảnh minh họa kết quả:





## 9 Tham khảo

- [1] <https://www.learnopencv.com/face-morph-using-opencv-cpp-python/>
- [2] <https://ccrma.stanford.edu/~jacoblui/368Report/>
- [3] <http://vision.gel.ulaval.ca/~jflalonde/cours/4105/h16/tps/results/tp3/JIZHA16/index.html>
- [4] <https://opencv.org/>
- [5] <http://dlib.net/>
- [6] <https://cmake.org/>