

# 计算机视觉 Ex5 实验文档

梁俊华 16340129 数据科学与计算机学院

## 一、实验要求

根据 Image Morphing 的方法完成中间 11 帧的差值,得到一个 Image Morphing 的动画视频。

## 二、实验原理

本实验的 Morphing 的过程包括下面四个步骤:

(1) 根据预先输入的网格顶点,生成 source 图和 target 图的三角形网格.

在上述步骤中,网格三角形的顶点通过在图像中寻找后,手动输入,生成相应的三角形网格,每个三角形网格中都含有当前图像的某些具体特征信息,通过源三角形网格中的信息到目标三角形网格中信息的均匀过度,就能够实现图像的均匀化。

(2) 根据输入的 frame 的值,生成中间 frame 值数量的中间帧的三角形网格.

frame 的值规定了中间帧的数目,在题目中规定为 11。然后通过对 source 图像和 target 图像插值得到中间帧三角形的顶点,生成中间值的公式如下所示:

$$mid = \begin{cases} mid\_x = source\_x + (currentFrame + 1) / (frame + 1) * (target\_x - source\_x) \\ mid\_y = source\_y + (currentFrame + 1) / (frame + 1) * (target\_y - source\_y) \end{cases}$$

得到所以相应三角形的顶点坐标以后,就能够组成中间帧三角形集合。

(3) 对于每一个中间帧,对于每一个点,检索出其所在的三角形,然后通过相应的线性变换,得到该点对应的 source 图中的点和 target 图中点,然后对 RGB 的值取相应的比例作为中间帧的 RGB 的值。

对于中间帧的每一个点,需要检索出其所在的三角形网格,本实验中检索点 P 是否在三角形中的算法为重心判断法,假设三角形的三个顶点分别为 A, B, C, 则通过向量运算能够得到向量 AB, AC, BC, 再计算出下面的值:

$$u = \frac{\overrightarrow{AB} \cdot \overrightarrow{AB} * \overrightarrow{AP} \cdot \overrightarrow{AC} - \overrightarrow{AC} \cdot \overrightarrow{AB} * \overrightarrow{AP} \cdot \overrightarrow{AB}}{\overrightarrow{AC} \cdot \overrightarrow{AC} * \overrightarrow{AB} \cdot \overrightarrow{AB} - \overrightarrow{AC} \cdot \overrightarrow{AB} * \overrightarrow{AB} \cdot \overrightarrow{AC}}$$

$$v = \frac{\overrightarrow{AC} \cdot \overrightarrow{AC} * \overrightarrow{AP} \cdot \overrightarrow{AB} - \overrightarrow{AC} \cdot \overrightarrow{AB} * \overrightarrow{AP} \cdot \overrightarrow{AC}}{\overrightarrow{AC} \cdot \overrightarrow{AC} * \overrightarrow{AB} \cdot \overrightarrow{AB} - \overrightarrow{AC} \cdot \overrightarrow{AB} * \overrightarrow{AB} \cdot \overrightarrow{AC}}$$

若计算结果满足：

$$0 \leq u \leq 1 \text{ and } 0 \leq v \leq 1 \text{ and } u + v \leq 1$$

则可以判定点 P 在三角形 ABC 内。

在判定点 P 在三角形 ABC 内部后，需要通过点 P 在当前帧的位置，通过相应线性变换后，在 source 图和 target 图中的对应坐标，此处转换的方法为采用线性变换的方法进行转换，相应的知识为仿射变换。假设当前帧中的点为  $x$ ，转换后的点为  $y$ ，若线性变换矩阵为  $A$ ，则相应的仿射变换为  $y = Ax + b$ ，则为进行相应的矩阵变换，需要写成如下形式：

$$\begin{bmatrix} y \\ 1 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}$$

$$\text{若 } x = \begin{bmatrix} x_k \\ y_k \end{bmatrix}, y = \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix}, \text{ 则上述矩阵可以写成 } \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ 1 \end{bmatrix}, \text{ 即}$$

可以转换成求一个 6 个未知数，6 个方程的方程组，即可得到相应的线性变换矩阵。

在求出相应的线性变换的矩阵后，就能够求出当前帧中点 P 在原图像和目标图像中对应的点，就能够算出当前点的 RGB 值，具体的计算公式如下所示：

$$middle\_frame(R,G,B) = \left(1 - \frac{frameNum+1}{frame+1}\right) * SOURCE(R,G,B) + \frac{frameNum+1}{frame+1} * TARGET(R,G,B)$$

计算后，最终能够得到 frame 张过程图像。

(4) 将生成的所以图片支撑.gif 图片。

将所以变换过程的图像和原图像和目标图像，在 PS 中生成.gif 图像。

### 三、测试说明

本实验的测试环境为 Windows10 操作系统，使用 MinGW64 进行命令行编译，实验本身已经生成.exe 文件，若具体编译，则编译指令如下所示：

```
D:\大三上\2016级\计算机视觉\HW5
λ g++ Morphing.h -std=c++11 -O2 -lgdi32 -c

D:\大三上\2016级\计算机视觉\HW5
λ g++ Morphing.cpp -std=c++11 -O2 -lgdi32 -o main

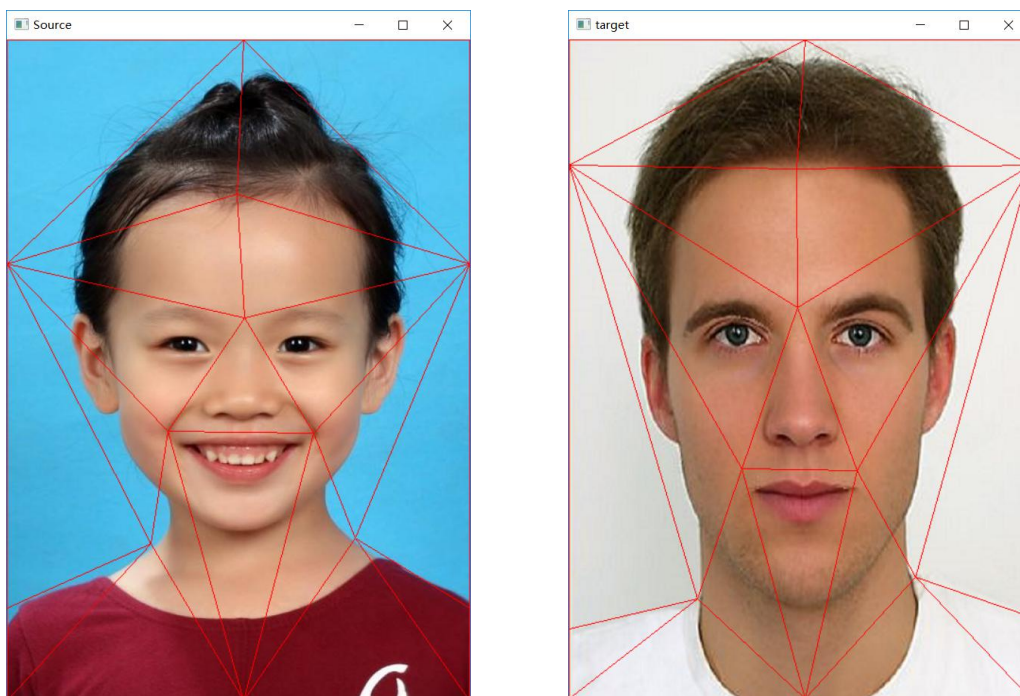
D:\大三上\2016级\计算机视觉\HW5
λ main.exe ./Test/test1.bmp ./Test/test2.bmp 11 < in.txt > out.txt
```

其中 Test 文件中存储的是原图像和目标图像，11 表示的是 frame 的值，in.txt 文件中存储的是原图像的三角形网格点和目标图像中的三角形网格点，还有相应三角形对应的网格点坐标，out.txt 则是用于输出。

执行 main.exe 需要实现输入上述 5 个参数，分别代表原图像的路径，目标图像的路径，帧数，输入网格点数据文件，输出文件。

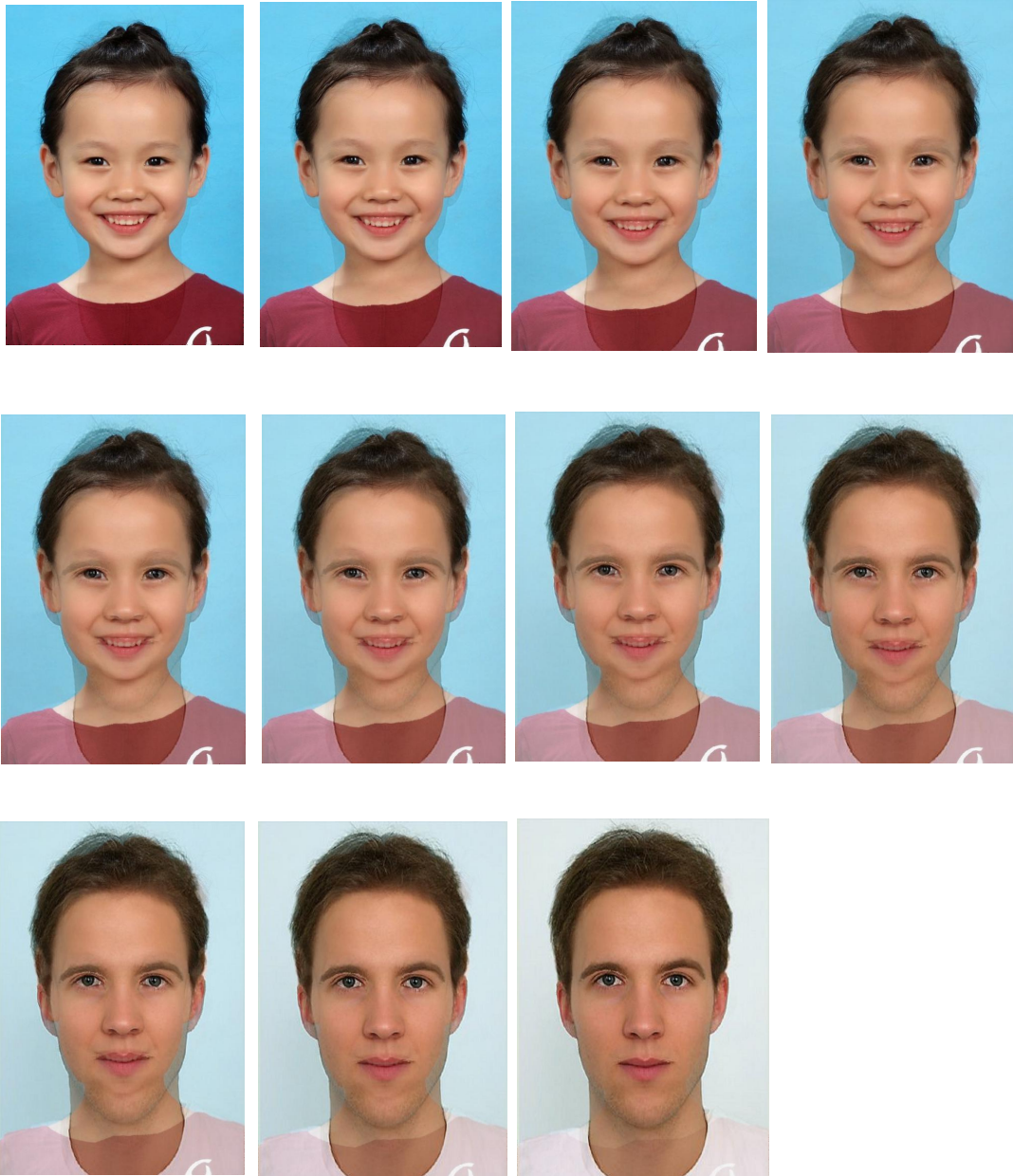
## 四、实验结果

本实验中，通过对图像的观察，划分的网格如下图所示：



对应网格点的数据均存储在 in.txt 文件中

执行相应的程序以后，就能够生成中间的 11 帧图像，具体如下所示：



将上述 11 张图片生成.gif 即能得到最终实验结果,实验结果为Morphing\_result.gif.

## 五、实验分析

本实验的结果还是能够得到比较完整的变化过程,但是有一个非常麻烦的地方,就是要手动输入三角形网格的顶点来生成相应的网格,显得非常冗余,也是一种比较笨的方法。因此可以在获取特征的方法上进行一些改进,例如采用某种自动生成三角形网格的算法,或者是特征线的变化算法,都能够有效改进这一缺点。其次,直接用线性变换的话,会设计比较多的矩阵变换,在速度上没有什么

优势，因此可以考虑用双线性插值等方法，但是代码会显得更为复杂。