

数字图像处理

实验报告

一、完成内容

我的代码全部通过 c++ 完成，使用了 opencv、Qt、dlib 库，具体完成了以下 4 部分内容：

1.Point Processing

我完成的点处理的 5 种处理效果，包括：Brightness、Darkness、Contrast、Gamma、Histogram Equalization。

2.Image Fusion

通过拉普拉斯卷积核计算散度度，并通过最小化散度误差完成图像融合，后面会介绍具体算法细节。

3.Face Morphing

通过调用 dlib 库提取特征点，再通过 Delaunay 三角剖分方法进行三角剖分，最后 warp 实现 face morphing 的效果。

4.GUI

我实现了一个方便操作的 GUI，能方便得进行输入，并展示上述三个完成的内容。通过 GUI 能进行点处理部分的实时处理及显示。图像融合部分可以编辑融合图像的位置。face morphing 部分同样可以编辑位置，并且能展示 morphing 过程的视频。

二、算法细节

1.Point Processing

实现方法与上课讲的一样，不再赘述。该代码部分在 ProcessItem.cpp 中实现，相关函数如下：

```
void brightness(Mat simg);      //变亮
void darkness(Mat simg);       //变暗
void contrast(Mat simg);       //增加对比度
void gamma(Mat simg);          //gamma 变换
void makeHistogram();          //histogram 计算
void equalization(Mat simg);    //均衡化
```

2.Image Fusion

图像融合部分部分代码在 FusionResult.cpp 中实现。实现方法是先采用了拉普拉斯卷积核计算前景图散度。

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

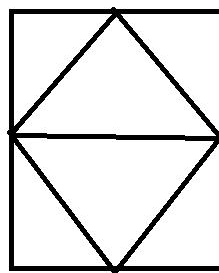
拉普拉斯卷积核

再对前景图 RGB 三个通道分别列方程，对每个像素，若该像素为 mask 的边缘，则必须等于背景图对应位置颜色。若不为边缘则改点散度应与前景图散度相等。因此对每个通道，有 mask 像素个数个方程，并且有 mask 像素个数个未知数，通过高斯消元的方法进行方程求解，即可得到 mask 部分每个像素的颜色，将其放在背景相应位置即可完成融合。

3.Face Mophing

因为我可以对图像进行位移后再进行 face morphing，故首先将两张图像非公共部分裁去，得到两张一样大小的图像。再通过调用 dlib 获得两张图像的 68 对特征点坐标，另外增加四角及四边的中点共 76 对特征点。

再对图像 1 进行三角剖分，我使用的是 Delaunay 三角剖分算法，具体如下。首先通过额外增加的 8 个点构造默认的 6 个三角形（如下图）。

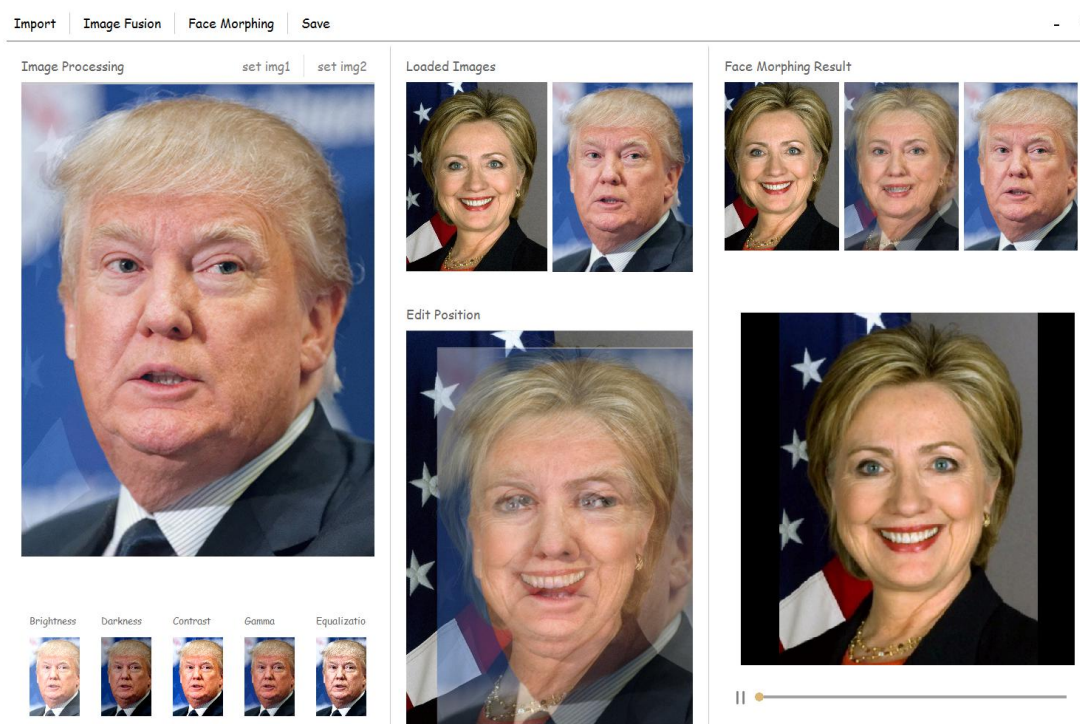


8 个默认点三角剖分

再依次取其它特征点，判断其在哪个三角形内部，删去改三角形并增加 3 个该点与原三角形连边形成的三角形。为了避免出现角度特别小的三角形，在每次新增加三角形时，判断其它与之公边的三角形，计算修改他们组成的四边形的对角线形成的新的三角形对的最小边。若比原先大，则改他们组成的四边形的对角线，从而替换两个三角形。

完成三角剖分后，对于原图每个像素，根据其所属的三角形及在三角形中的位置计算 warp 后的位置，即可得到 warp 后的图像。两张图像 warp 到特征点位置都相同，再通过不同透明度进行融合，即可得到结果。

三、结果展示



上图所示即为 GUI 界面，点击左上角的 **Import** 可以导入图片，然后可以对图片进行简单编辑，即为点处理部分。点下方缩略图，即可进行编辑，下图即为点击均衡化后的结果，可以看到编辑后还能继续实时进行编辑。



通过点击 **set img1** 或 **set img2**，即可将编辑完成的图像放入 **loaded image** 的左边或右边。设置两张待处理图片后，即可在 **edit position** 部分通过拖动对这两张图片的相对位置进行编辑。用该功能可以方便地将图片拖到想要位置，如下图将眼镜图片拖到 **trump** 眼睛处，效果如下

Loaded Images



Edit Position



最后点击上方的 Image Fusion 或 Face Morphing 即可生成结果。这里注意，使用 Image Fusion 时必须保证

Image Fusion Result



Face Morphing 生成效果如下，上面为实验要求的三张渐变图，下方可以播放渐变视频。



由于视频不能在报告中展示，我将更多的生成结果存放在 demos 文件夹下。