数值分析

大作业一

图像扭曲变形

学　　号　　2017011589

姓　　名　　吾尔开西

班　　级　　自 76

目录

一、需求分析

1、必做任务

必做任务要求用最近邻、双线性、双三次插值方法来实现旋转扭曲和畸变扭曲两种图像扭曲结果。

用户可以任意选择三种插值方式中的任意一种，也可以任意选择对图像进行哪种扭曲，且能很方便地设置扭曲参数。

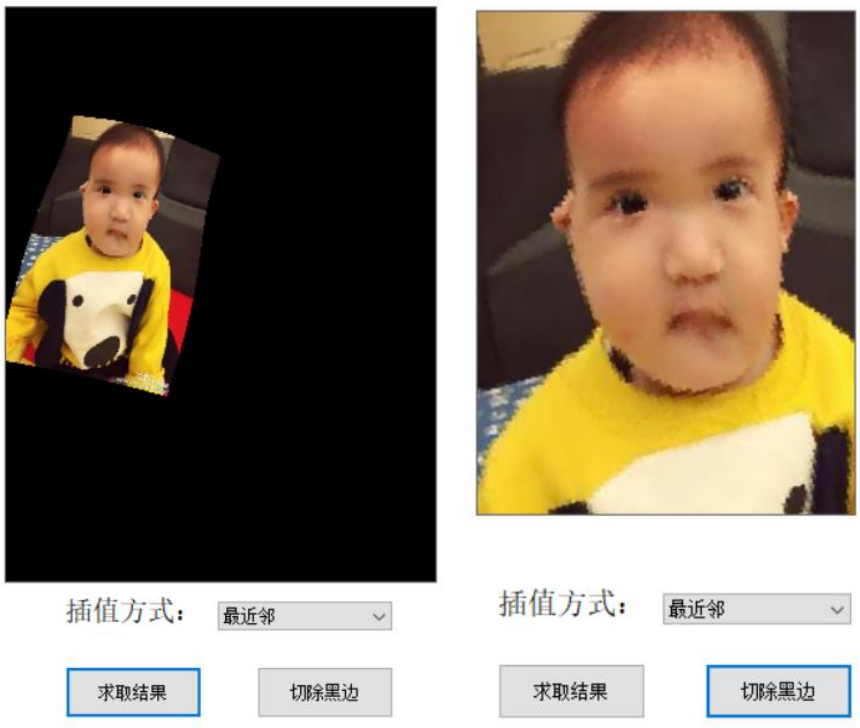
2、选做任务

助教提供了 9 张图片，每张图片都有 68 个特征点，这些特征点的坐标保存在 txt 文件里。

任务要求对于任意两张图片，使用 TPS 薄板样条插值方法，求出一个从目标图像坐标点到原图像坐标点的映射，使得经过这个映射后原图像中的脸型变换为目标图像中的脸型。

用户可以任意选择 9 张图片中的两张图片作为原图片和目标图片，并使用三种插值方法中的任意一种来得到变换后的图像。

变换结果可能会出现较多黑边，有效结果面积过小。为了改善用户体验，我增加了切除黑边的功能，将变换结果旋转到正方向并拉伸到合适大小，再切除图片黑边，效果如下。



二、扭曲变形方式

必做任务要求用旋转扭曲和畸变扭曲两种方式对原图进行变形，本项目用目标图到原图的计算方式来实现。

设扭曲目标图为$AIM_{i'j'}$，原图像为S_{ij}，对于目标图中的每一点坐标(i',j')，通过扭曲函数$f(i',j')$得到对应原图的坐标点(i,j)，将$AIM(i',j')$的 RGB 值设为$S(i,j)$的 RGB 值。对遍历目标图中所有的点都遍历使用上述方法，便能得到目标图像。

然而，扭曲函数$(i,j)=f(i',j')$得到的原图坐标可能不是整数坐标，所以需要使用插值的方法求得$S(i,j)$，本节介绍扭曲变形的方法。

1、旋转扭曲

旋转扭曲将图片中心半径 R 以内的部分进行旋转变形，越靠近中心的部分变形程度越大，即旋转角度越大。用户可以选择旋转半径 R 和旋转最大角度a_{max}。

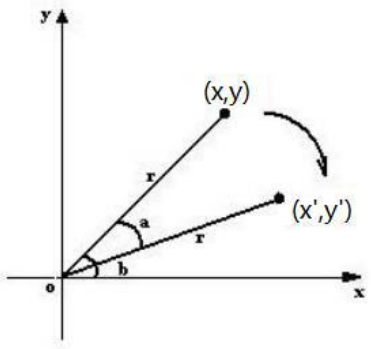
设扭曲目标图为$AIM_{i'j'}$，原图像为S_{ij}，原图像高度为 h，宽度为 w。对于目标图中的一点坐标(i',j')，先计算它到中心的距离$D=\sqrt{\left(i'-\frac{h}{2}\right)^2+\left(j'-\frac{w}{2}\right)^2}$，若$\mathrm{D}>\mathrm{R}$，则对于原图坐标$(i,j)=(i',j')$，否则，计算旋转角度：

$$\alpha=\alpha_{max}\times\frac{R-D}{R}$$

对应原图坐标(i,j)：

$$\begin{cases} i=(i'-\frac{h}{2})cos\alpha-(j'-\frac{w}{2})sin\alpha+\frac{h}{2} \\ j=(i'-\frac{h}{2})sin\alpha+(j'-\frac{w}{2})cos\alpha+\frac{w}{2} \end{cases}$$

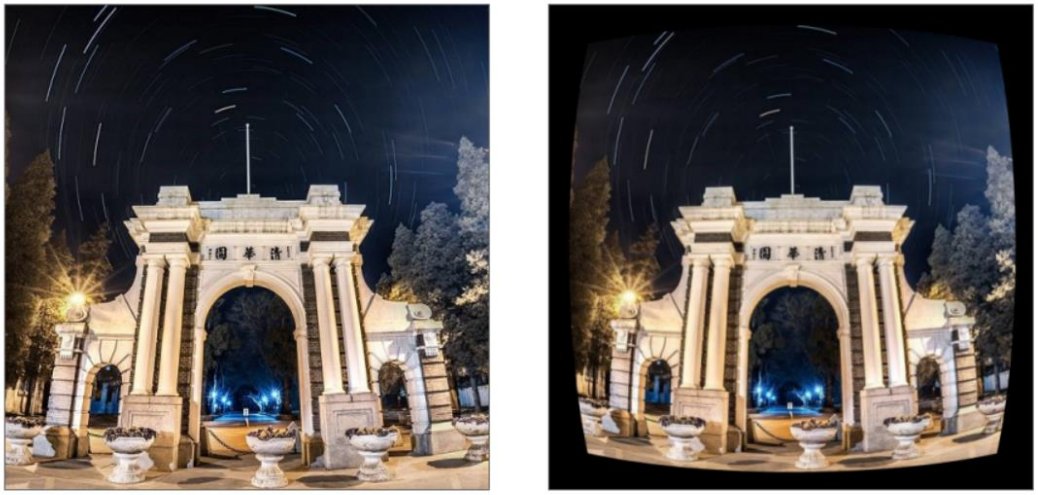
再用插值的方法将原图中坐标(i,j)处的 RGB 值赋值给目标图(i',j')。



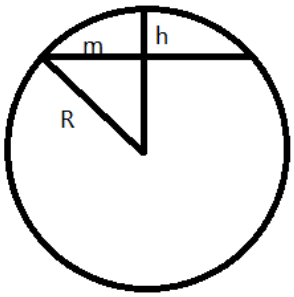
2、畸变扭曲

畸变扭曲是使图片拱起或凹陷，越靠近中间区域的部分变形程度越高，就像是将图片放在一个球的上表面或下表面再从球的正上方看下去。使图片拱起（）

（1）桶形畸变



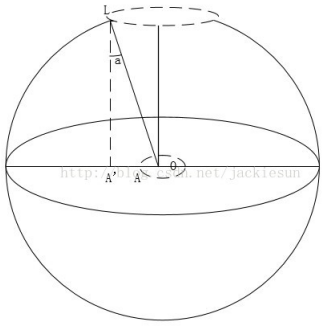
用户需要输入球形畸变的高度 H，设原图像高度为 h，宽度为 w，$m=\sqrt{\left(\frac{h}{2}\right)^2+\left(\frac{w}{2}\right)^2}$，则畸变球的半径$R=\frac{H^2+m^2}{2|H|}$。



设扭曲目标图为$AIM_{i'j'}$，原图像为S_{ij}。对于目标图中的一点坐标(i',j')，先计算它到中心的距离$D=\sqrt{\left(i'-\frac{h}{2}\right)^2+\left(j'-\frac{w}{2}\right)^2}$。

设$x'=i'-\frac{h}{2},y'=j'-\frac{w}{2}$，$arc=R\times\arcsin\left(\frac{D}{R}\right)$为目标点坐标在球面上与中心点连线所形成弧线的长度 x 坐标，则$\frac{arc}{D}\times x'$便是原图对应 x 坐标。

$$\begin{cases}x=\dfrac{R}{D}\times\arcsin\left(\dfrac{D}{R}\right)\times x'\\ y=\dfrac{R}{D}\times\arcsin\left(\dfrac{D}{R}\right)\times y'\end{cases}$$



原图坐标$i=x+\frac{h}{2}$，$\mathrm{j}=\mathrm{y}+\frac{\mathrm{w}}{2}$。再用插值的方法将原图中坐标$(i,j)$处的 RGB 值赋值给目标图$(i',j')$。

（2）枕形畸变

枕形畸变的算法大体与桶形畸变相同，只是坐标计算部分，需要将原图坐标和目标图坐标位置颠倒，所以计算公式变为：

$$\begin{cases} x = \dfrac{D}{R \times \arcsin\left(\dfrac{D}{R}\right)} \times x' \\ y = \dfrac{D}{R \times \arcsin\left(\dfrac{D}{R}\right)} \times y' \end{cases}$$

三、插值

设扭曲目标图为$AIM_{i'j'}$，原图像为S_{ij}。用扭曲函数$(i,j) = f(i',j')$得到的原图坐标(i,j)可能不是整数坐标，所以需要使用插值的方法求得$S(i,j)$，本节介绍插值的方法。

设原图位于坐标(i,j)处的 RGB 值为$S(i,j)$，表示一个三元向量，满足向量的加减性质。

1、最近邻插值

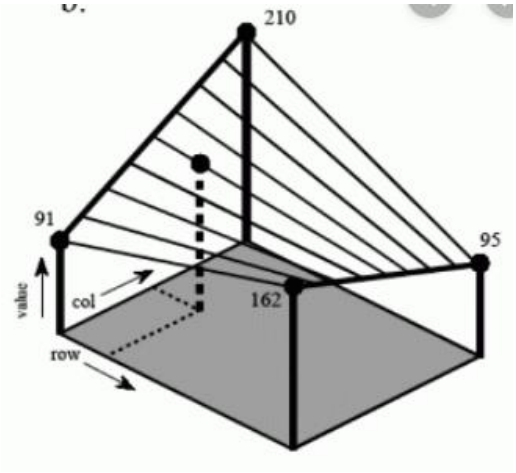
最近邻插值的公式比较简单：

$$S(i,j) = S(round(i), round(j))$$

其中，$round(x)$函数表示最靠近浮点数 x 的整数值。

2、双线性插值

双线性插值，顾明思义，是分别在 x 方向和 y 方向进行两次插值，得到最终插值结果的方法。



（图片来自维基百科）

对于原图坐标(i,j)，先得到离它最近的四个整数点坐标$(i_1,j_1),(i_1,j_2),(i_2,j_1),(i_2,j_2)$以及他们的 RGB 值。其中:

$$i_1 = floor(i), j_1 = floor(j)$$

$$i_2 = i_1 + 1, j_2 = j_1 + 1$$

首先在 x 方向进行插值，

$$S(i,j_1) = \frac{i_2 - i}{i_2 - i_1} S(i_1,j_1) + \frac{i - i_1}{i_2 - i_1} S(i_2,j_1)$$

$$S(i,j_2) = \frac{i_2 - i}{i_2 - i_1} S(i_1,j_2) + \frac{i - i_1}{i_2 - i_1} S(i_2,j_2)$$

在用得到的两个点在 y 方向进行插值，即可得到结果

$$S(i,j) = \frac{j_2 - j}{j_2 - j_1} S(i,j_1) + \frac{j - j_1}{j_2 - j_1} S(i,j_2)$$

3、双三次插值

双三次插值原理与双线性插值类似，但涉及更多的点（16 个点），插值系数使用特定函数来计算，自变量是到所求点的横纵坐标距离。

设$i_1 = floor(i),\ j_1 = floor(j);\ \ dx = i - i_1,\ dy = j - j_1$

用到的点包括:

$$(i_1 + m, j_1 + n) \quad m = -1,0,1,2;\ \ n = -1,0,1,2$$

插值函数:

$$S(i,j) = \sum_{m=-1}^{2} \sum_{n=-1}^{2} S(i_1 + m, j_1 + n) R(m - dx) R(n - dy)$$

其中，$R(x)$为 B 样条曲线函数

$$R(x) = \begin{cases} \frac{2}{3} + \frac{1}{2}|x|^3 - (x)^2, & 0 \le |x| \le 1 \\ \frac{1}{6}(2 - |x|)^3, & 1 \le |x| \le 2 \end{cases}$$

运算结果时还需要做正则化，除以运算过程中所乘系数的和。

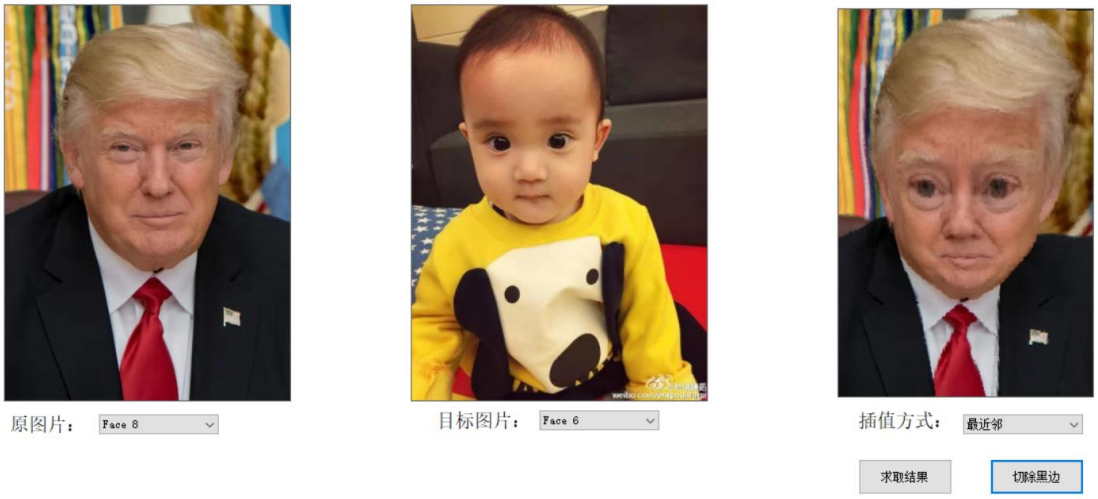
四、人脸变形

任务要求从提供的 9 张图片中，任意选出两张作为原脸型图和目标脸型图，使用 TPS 薄板样条插值方法，求出一个从目标脸型图坐标点到原脸型图坐标点的映射，并用这个映射求出结果图坐标对应原脸型图坐标的位置，结果图中原图像脸型变换为目标图像中的脸型。

用户可以任意选择 9 张图片中的两张图片作为原图片和目标图片，并使用三种插值方法中的任意一种来得到变换后的图像。

基本变换完成后，结果有一个问题，因为原图和目标图中脸的位置可能不在图片同一区域，甚至可能相差很大。这样变换得到的结果中，除脸部以外的部分会有较大的变形，为了避免这种的位置变形，本项目首先将目标脸型图的特征点平移到原脸型图特征点附近位置，再做上述步骤中的计算。

此外，变形结果中会有一定的黑边，本项目也增加了切除黑边的功能。



1、TPS 薄板样条插值

TPS 薄板样条插值是一种插值算法，在本项目中可以把矫正问题看作是一个二维插值问题，已知 n 个特征点对（n=64,目标脸型图中点的坐标）(x_i, y_j)和它们对应的函数值（原图脸型对应点的坐标）$f(x_i, y_i) \in D^2, (i = 1,2,\dots,n)$，求一个插值函数，并用这个插值函数求得目标图中所有点对应的原图中点的坐标，求出的原图中点坐标可能不是整数，所以还需要用插值方法求得这个点的 RGB 值。

关键是如何求得这个插值函数，在 TPS 算法中(Bookstein & intelligence, 1989)，插值函数形式为：

$$f(x,y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} w_i U(|(x_i, y_i) - (x,y)|)$$

其中，$U(t) = t^2 \log t$。a_0, a_1, a_2, w_i均为参数，维度均为2×1，满足$\sum_{i=1}^{n} w_i = \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} w_i y_i = 0.$

将上面的条件和插值条件写成矩阵形式：

$$\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} V \\ 0 \end{bmatrix}$$

其中，$K \in D^{n\times n},\ K_{ij} = U(\left|(x_i, y_i) - \left(x_j, y_j\right)\right|;\ P \in D^{3\times n},\ P_i = [1\ \ x_i\ \ y_i];\ w \in$

$D^{n\times 2};\ a = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix};\ V \in D^{n\times 2},\ V_i = f(x_i,\ y_i)$

设L $= \begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix}$，则$\begin{bmatrix} w \\ a \end{bmatrix} = L^{-1}\begin{bmatrix} V \\ 0 \end{bmatrix}$。

2、求取参数矩阵 W

求取参数$\begin{bmatrix} w \\ a \end{bmatrix}$的过程其实也是一个解 n+3 元线性方程组的过程。

该项目使用

3、使用 W 求取结果图

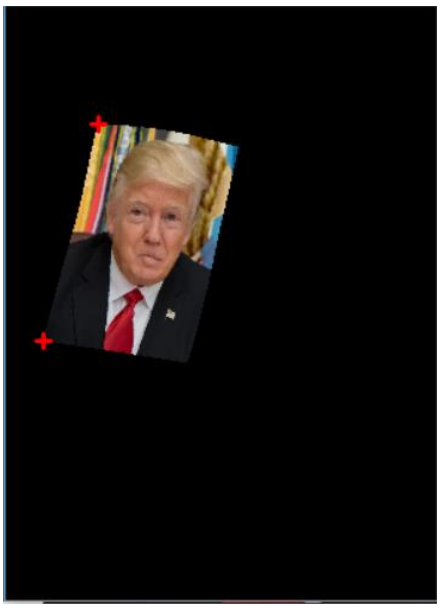
求得参数矩阵 W 后，就可以使用 TPS 公式求取结果图中每个坐标点(x, y)对应原脸型图中的坐标

$$f(x, y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} w_i U(|(x_i, y_i) - (x, y)|)$$

$f(x, y)$可能不是整数值，所以需要使用前面介绍的三种插值方法之一来求得原图在$f(x, y)$坐标点处的 RGB 值，并将它赋给结果图(x, y)处。

4、切除黑边

由于目标图和原图中脸型方向可能不同，所以变换结果相较原图，可能会有较大角度的旋转，如图所示。如果直接在这样的结果上进行水平纵向的裁剪，那么图像有效部分将有较大程度的损失，所以需要先将结果图片旋转到正方向。

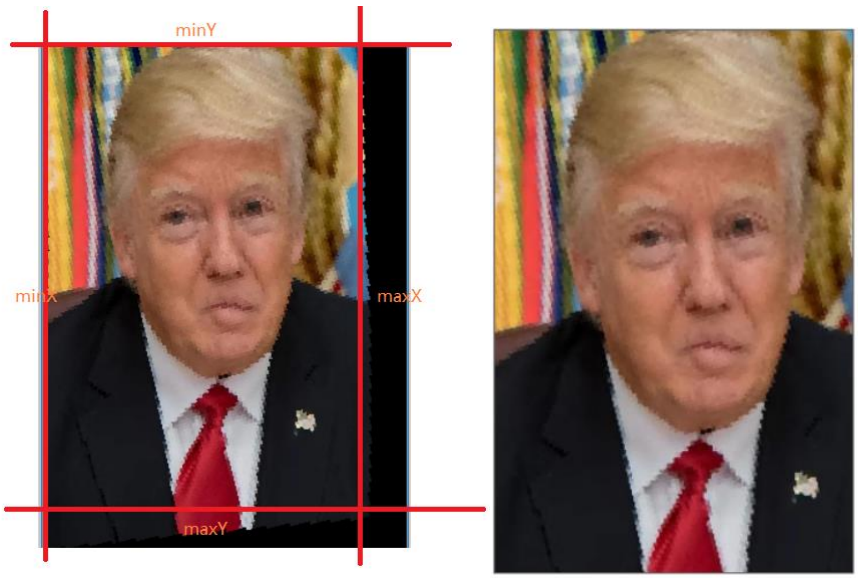


为了将结果图片旋转到正方向，需要找到图片的左上角和左下角，这两个点可以在 TPS 变换过程中进行记录，结果如上图中红色十字标注。利用这两个点就能算出图片当前的倾斜角α，再用旋转公式得到目标图中每一个点(i', j')对应原图的坐标点(i, j)

$$\begin{cases} i = (i' - \frac{h}{2})cos\alpha - (j' - \frac{w}{2})sin\alpha + \frac{h}{2} \\ j = (i' - \frac{h}{2})sin\alpha + (j' - \frac{w}{2})cos\alpha + \frac{w}{2} \end{cases}$$

将旋转后的坐标进行伸缩和平移，就能让图片占据框架的主要部分了，如图所示。

最后，遍历检测出图像没有黑边的最小范围，并对图像进行裁剪就可以了。



五、误差分析

1、舍入误差

（1）计算过程中使用 double 型变量，其精度约为10^{-16}，因此单次计算舍入误差$R_1 \leq 5 \times 10^{-17}$，总的舍入误差会随计算次数累积，计算过程中循环次数最多在n^2量级（100^2量级），所以舍入误差大概在10^{-13}量级。

（2）在进行插值运算时，运算结果的 RGB 值需要离散化为 0~255 的整数，舍入误差$R_2 \leq 0.5$

2、方法误差

本项目用到的模块化算法有：三种插值方法、两种扭曲方法、TPS 薄板样条插值等，其中图像扭曲方法和 TPS 算法在生成结果图像时都用到了三种插值方法，也就是说最后的方法误差都落在三种插值方法上。

（1）最近邻插值

插值公式：

$$S(i,j) = S(round(i), round(j))$$

误差：

$$R = \Delta S \leq \max\left(\left|\frac{\partial S}{\partial i}\right|\right) \cdot |\Delta i| + \max\left(\left|\frac{\partial S}{\partial j}\right|\right) \cdot |\Delta j|$$

$$= \frac{1}{2}\left(\max\left(\left|\frac{\partial S}{\partial i}\right|\right) + \max\left(\left|\frac{\partial S}{\partial j}\right|\right)\right)$$

其中，$\max\left(\left|\frac{\partial S}{\partial i}\right|\right)$和$\max\left(\left|\frac{\partial S}{\partial j}\right|\right)$大小与具体图像有关，但可以保证$\max\left(\left|\frac{\partial S}{\partial i}\right|\right) \leq 255$

（2）双线性插值

插值公式:

$$S(i,j_1)=\frac{i_2-i}{i_2-i_1}S(i_1,j_1)+\frac{i-i_1}{i_2-i_1}S(i_2,j_1)$$

$$S(i,j_2)=\frac{i_2-i}{i_2-i_1}S(i_1,j_2)+\frac{i-i_1}{i_2-i_1}S(i_2,j_2)$$

$$S(i,j)=\frac{j_2-j}{j_2-j_1}S(i,j_1)+\frac{j-j_1}{j_2-j_1}S(i,j_2)$$

双线性插值是在 i 方向和 j 方向分别进行一维一阶（n=1）插值，先计算一维误差:

$$R_i \leq \frac{\max|S''(i)|}{2}|w_{n+1}(i)| = \frac{1}{2}\cdot\max\left|\frac{\partial^2 S}{\partial i^2}\right|\cdot|(i-i_1)(i-i_2)| \leq \frac{1}{8}\max\left|\frac{\partial^2 S}{\partial i^2}\right|$$

$$R_j \leq \frac{1}{8}\max\left|\frac{\partial^2 S}{\partial j^2}\right|$$

$$R \leq \left|R_i+R_j\right| \leq \frac{1}{8}(\max\left|\frac{\partial^2 S}{\partial j^2}\right|+\max\left|\frac{\partial^2 S}{\partial i^2}\right|)$$

（3）双三次插值

双三次插值可以视为在 i 方向和 j 方向分别进行一维三阶（n=3）插值，先计算一维误差:

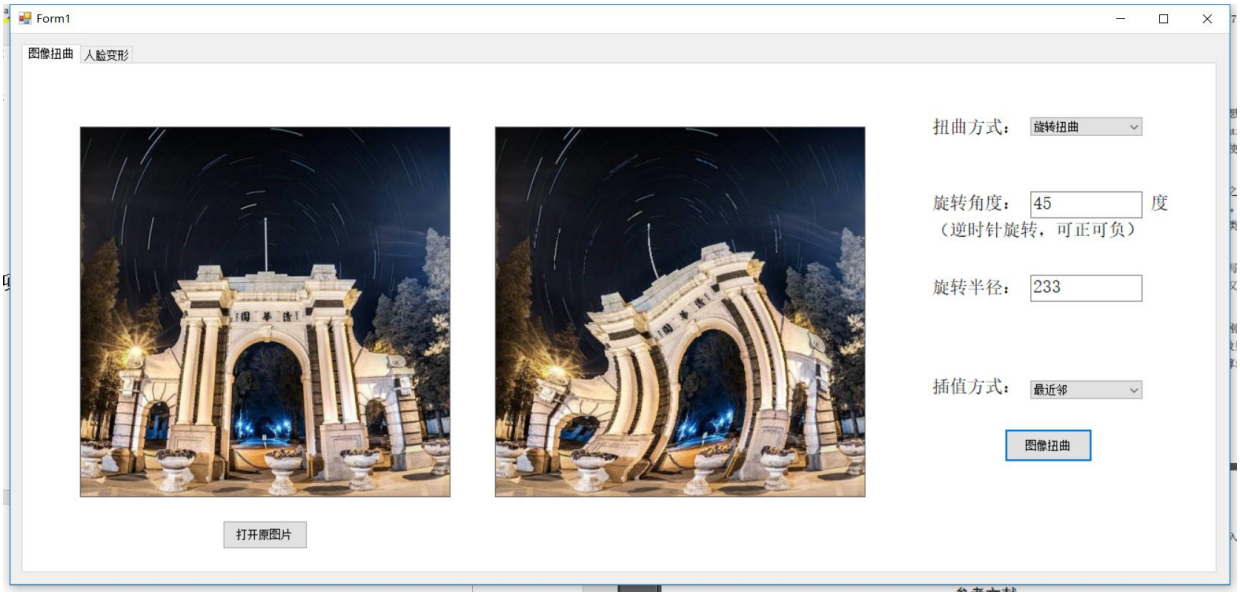
$$R_i \leq \frac{\max\left|S^{(4)}(i)\right|}{4!}|w_{n+1}(i)| = \frac{1}{24}\cdot\max\left|\frac{\partial^4 S}{\partial i^4}\right|\cdot|(i-i_1)(i-i_2)(i-i_3)(i-i_4)|$$
$$\leq \frac{5}{384}\max\left|\frac{\partial^4 S}{\partial i^4}\right|$$

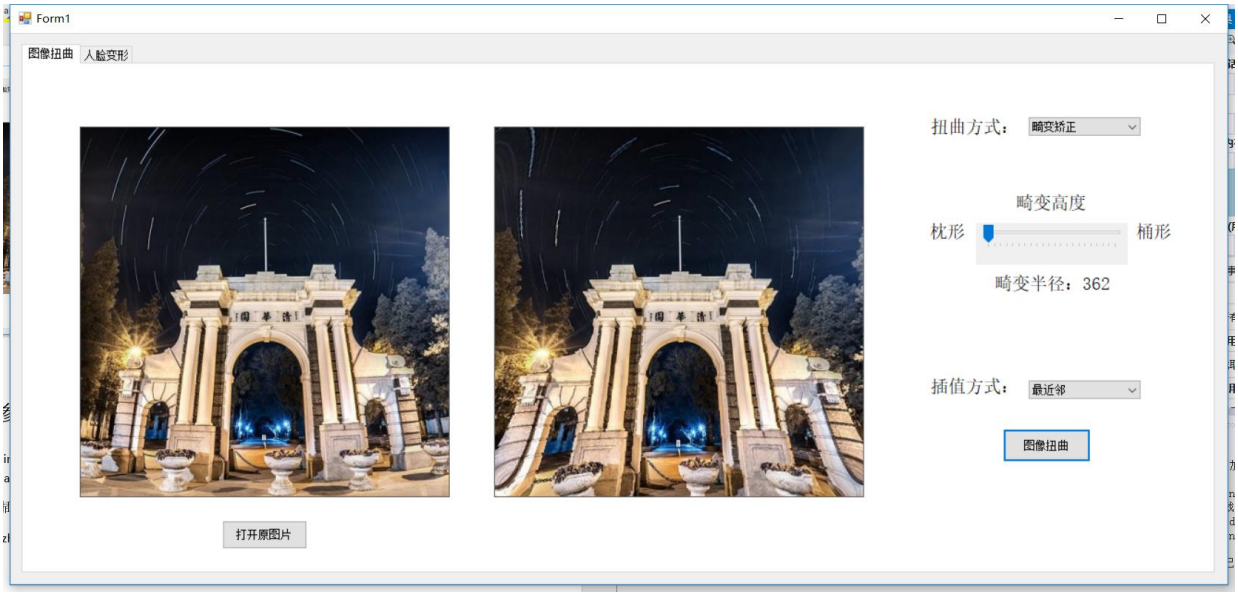
$$R_j \leq \frac{5}{384}\max\left|\frac{\partial^4 S}{\partial j^4}\right|$$

$$R \leq \left|R_i+R_j\right| \leq \frac{5}{384}(\max\left|\frac{\partial^4 S}{\partial j^4}\right|+\max\left|\frac{\partial^4 S}{\partial i^4}\right|)$$

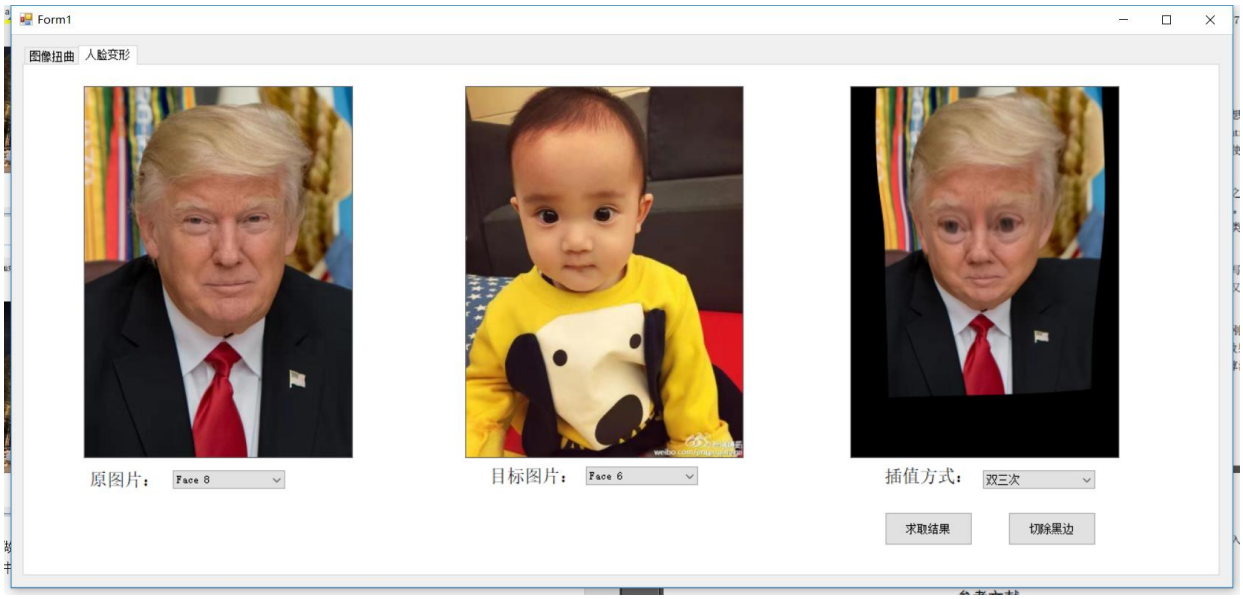
六、项目结果

程序界面如图所示，用两个标签页分别展示必做任务和选做任务的界面。必做任务的标签页可以选择扭曲方式、插值方式、输入扭曲参数，扭曲参数的输入部件会根据扭曲方式的选择而切换。

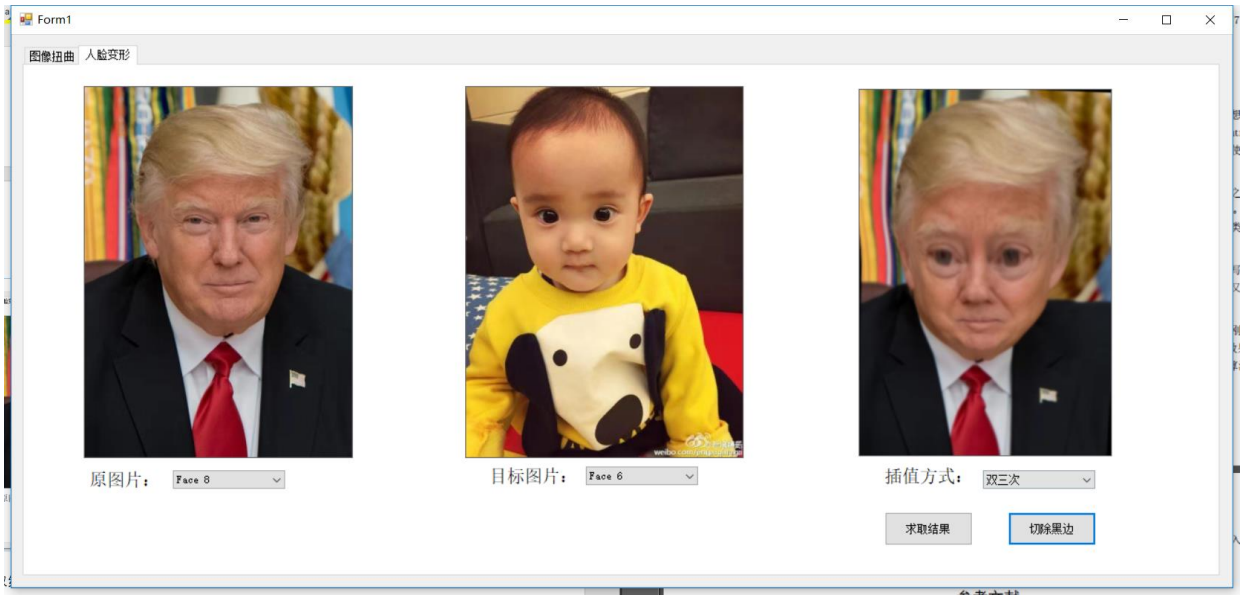




选做任务的界面中，可以任意选择原脸型图片和目标脸型图片，可以选择三种插值方式中的一种。



求取结果后可以按“切除黑边”按钮切除图片黑边。



具体变换的结果已在前面贴出，不再重复。

七、总结

通过完成这次大作业，我提高了运用理论知识解决实际问题的能力，比如插值的方法；同时也学到了很多新知识，比如 TPS 薄板样条插值，使用稳定的高斯消元法求解多元线性方程等。

在进行算法的实现时，对算法的深入理解和清晰的思路是非常重要的。以 TPS 算法为例，首先求解参数矩阵 W，求解过程是以目标脸型图片中的特征点映射到原脸型

图片中的特征点，与之对应的，求解结果图片也是将结果图片中的每个坐标点带入 TPS 函数求解原图片中对应的位置。

此外，这次大作业也让我意识到代码模块化和标准化的重要性，从一开始我就将代码实现分成了几个类的实现，减少代码复用，所以调试起来很方便，增加新功能也很便捷。

总之，通过这次大作业我增长了很多新知识，加深了对理论知识的理解，也加强了我动手解决实际问题的能力。

八、参考资料

Bookstein, F. L. J. I. T. o. p. a., & intelligence, m. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *11*(6), 567-585.

双线性插值<维基百科，自由的百科全书>

https://zh.wikipedia.org/wiki/%E5%8F%8C%E7%BA%BF%E6%80%A7%E6%8F%92%E5%80%BC