

# 数值分析第一次大作业——图像变形 实验报告

陈策

自 72

2017011619

# 目录

目的需求分析 .....	3
图片表示 .....	3
扭曲变换 .....	3
旋转扭曲 .....	3
畸变矫正 .....	3
插值方法 .....	3
最近邻插值 .....	4
双线性插值 .....	4
双三次插值 .....	4
TPS 网格变形 .....	5
人脸预对齐 .....	5
TPS 变换 .....	5
方案设计 .....	6
开发环境 .....	6
依赖项 .....	6
程序设计 .....	6
前端 .....	7
后端 .....	7
项目效果 .....	7
扭曲变换 .....	7
旋转扭曲 .....	7
畸变矫正 .....	8
插值方法 .....	8
最近邻插值 .....	8
双线性插值 .....	9
双三次插值 .....	9
TPS 网格变形 .....	9
人脸未预对齐 <b>【正式版本已移除此方法】</b> .....	9
人脸预对齐 .....	10
误差分析 .....	10
观测误差 .....	10
舍入误差 .....	10
模型误差 .....	10
方法误差 .....	10
扭曲变换 .....	10
插值方法 .....	11
TPS 网格变形 .....	13
总结反思 .....	14
功能实现 .....	14
误差分析 .....	14
项目构建 .....	14

# 目的需求分析

本部分为大作业相关功能的需求分析。

## 图片表示

任意一张图片均可表示为一个三维矩阵[Width × Height × Channel]，其中Width为图片的宽度，Height为图片的高度，Channel为图片的通道数，对于二值图像与灰度图像，Channel = 1 (RGB8)，对于一般的彩色图像，Channel = 3 (RGB24) 或者Channel = 4 (RGB32)，对于图像的操作均可归结为**矩阵变换与函数映射**，由于此次作业为图像变形，其操作主要集中在矩阵的前两维，所以又可将其化简为普通的**函数映射**问题。

## 扭曲变换

扭曲变换为必做部分，其由旋转扭曲与畸变矫正两部分组成。

### 旋转扭曲

首先确定旋转中心 $O$ 、最大旋转角度 $\alpha_{\max}$ 扭曲半径 $R$ ，只需要对图像中的点进行旋转变换，即可得到旋转扭曲的图形效果，具体操作如下：

对于新图像中的任意点 $P$ ，其到旋转中心 $O$ 的距离是 $r_P$ ，则其绕旋转中心 $O$ 的旋转角度为

$$\alpha = \alpha_{\max} \times \frac{\max(R - r_P)}{R}$$

其在原图中的位置为 $P'$ ，则可得

$$P' = f(P) = P \cdot \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

### 畸变矫正

首先确定旋转中心 $O$ 与归一化伸缩因子 $\beta$ ，对全局的点进行畸变变换，只需要改变其在坐标系中相对中心点的距离，并且缩放比例与其离中心点的距离成正相关，对于结果超出原图范围的点，使其为黑色即可，具体操作如下：

对于新图像中的任意点 $P$ ，其到旋转中心 $O$ 的距离是 $r_P$ ，归一化伸缩因子为 $\beta$ ，其在原图中的位置为 $P'$ ，则可得

$$P' = f(P) = P \cdot \begin{bmatrix} \beta \cdot \frac{r_P^2}{\text{Scale}} & 0 \\ 0 & \beta \cdot \frac{r_P^2}{\text{Scale}} \end{bmatrix}$$

经过反复调试，当 $\text{Scale} = \text{Width}_{\text{原图像}} \times \text{Height}_{\text{原图像}}$ 时，归一化伸缩因子 $\beta$ 在 $[-1,1]$ 内表现最好。

## 插值方法

插值方法为必做部分，其由最近邻插值、双线性插值和双三次插值三部分组成。

## 最近邻插值

根据新图像尺寸确定插值缩放参数，将新图像坐标缩放到原图像上，取缩放后离其最近的点的颜色为新图像上的点的颜色即可实现最近邻插值，具体操作如下：

对于新图像中的任意点 $P$ 颜色为 $C_P$ ，其在原图中的位置为 $P'$ ，对应颜色为 $C_{P'}$ ，函数 $g$ 表示取对应点在原图中的颜色，则可得：

$$P' = P \cdot \begin{bmatrix} \frac{\text{Width}_{\text{原图像}}}{\text{Width}_{\text{新图像}}} & 0 \\ 0 & \frac{\text{Height}_{\text{原图像}}}{\text{Height}_{\text{新图像}}} \end{bmatrix}$$

$$C_P = f(C_{P'}) = g([P'])$$

## 双线性插值

根据新图像缩放到原图像上相邻的四个点，在 $x$ 方向和 $y$ 方向上分别进行线性插值，得到新图像目标点的颜色即可实现双线性插值，具体操作如下：

对于新图像中的任意点 $P$ 颜色为 $C_P$ ，其在原图中的位置为 $P'$ ，对应颜色为 $C_{P'}$ ，函数 $g$ 表示取对应点在原图中的颜色，则可得：

$$P' = P \cdot \begin{bmatrix} \frac{\text{Width}_{\text{原图像}}}{\text{Width}_{\text{新图像}}} & 0 \\ 0 & \frac{\text{Height}_{\text{原图像}}}{\text{Height}_{\text{新图像}}} \end{bmatrix}$$

$$[u \quad v] = P' - [P']$$

$$C = \begin{bmatrix} g([P']) + [0 \quad 0] & g([P']) + [0 \quad 1] \\ g([P']) + [1 \quad 0] & g([P']) + [1 \quad 1] \end{bmatrix}$$

$$C_P = f(C_{P'}) = [1 - u \quad u] \cdot C \cdot \begin{bmatrix} 1 - v \\ v \end{bmatrix}$$

## 双三次插值

根据新图像缩放到原图像上相邻的十六个点和参数 $a$ ，在 $x$ 方向和 $y$ 方向上分别进行三次插值，通过 $BiCubic$ 函数得到各点权重，加权求和得到新图像目标点的颜色即可实现双三次插值，具体操作如下：

对于新图像中的任意点 $P$ 颜色为 $C_P$ ，其在原图中的位置为 $P'$ ，对应颜色为 $C_{P'}$ ，函数 $g$ 表示取对应点在原图中的颜色，则可得：

$$P' = P \cdot \begin{bmatrix} \frac{\text{Width}_{\text{原图像}}}{\text{Width}_{\text{新图像}}} & 0 \\ 0 & \frac{\text{Height}_{\text{原图像}}}{\text{Height}_{\text{新图像}}} \end{bmatrix}$$

$$B(k) = \begin{cases} (a+2)|k|^3 - (a+3)|k|^2 + 1 & \text{for } |k| \leq 1 \\ a|k|^3 - 5a|k|^2 + 8a|k| - 4a & \text{for } 1 < |k| \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

$$[u \quad v] = P' - [P']$$

$$C = \begin{bmatrix} g([P'] + [-1 & -1]) & g([P'] + [-1 & 0]) & g([P'] + [-1 & 1]) & g([P'] + [-1 & 2]) \\ g([P'] + [0 & -1]) & g([P'] + [0 & 0]) & g([P'] + [0 & 1]) & g([P'] + [0 & 2]) \\ g([P'] + [1 & -1]) & g([P'] + [1 & 0]) & g([P'] + [1 & 1]) & g([P'] + [1 & 2]) \\ g([P'] + [2 & -1]) & g([P'] + [2 & 0]) & g([P'] + [2 & 1]) & g([P'] + [2 & 2]) \end{bmatrix}$$

$$C_p = f(C_{p'}) = [B(u+1) \quad B(u) \quad B(u-1) \quad B(u-2)] \cdot C \cdot \begin{bmatrix} B(v+1) \\ B(v) \\ B(v-1) \\ B(v-2) \end{bmatrix}$$

## TPS 网格变形

TPS 网格变形为选做部分。

## 人脸预对齐

在实际测例与真实场景中，不难发现，两张图片的人脸 68 个关键点位置与图片大小密切相关，如果在进行 TPS 变换之前不对人脸的关键点进行预对齐，轻则会导致图片内容过度扭曲或大小发生改变，重则将会导致 TPS 矩阵计算过程中矩阵不可逆，无法计算，由此可见人脸预对齐的重要性。

本项目中人脸预对齐的方式采用仿射变换与最小二乘法求解参数确定，通过仿射变换对坐标点进行拉伸、旋转、平移等操作，并对参数进行最小二乘法拟合，可以得到两张图片之间的对齐矩阵，形式如下：

$$[P' \quad 1]A = P$$

$$A = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$$

其中  $P'$  为人脸图像的关键点坐标， $P$  为源图像的关键点坐标  
使用最小二乘法拟合求解为

$$A^* = \left( \begin{bmatrix} P'_1 & \dots & P'_{68} \\ 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} P'_1 & 1 \\ \vdots & \vdots \\ P'_{68} & 1 \end{bmatrix} \right)^{-1} \left( \begin{bmatrix} P'_1 & \dots & P'_{68} \\ 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} P_1 \\ \vdots \\ P_{68} \end{bmatrix} \right)$$

对齐之后的人脸图像的关键点坐标

$$P'_A = [P' \quad 1]A^*$$

## TPS 变换

TPS 变换的目标是寻找一个通过所有控制点的光滑曲面  $f(x, y)$ ，使得能量函数  $I_f$  最小：

$$I_f = \iint_{R^2} \left( \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right) dx dy$$

其解为：

$$f(P) = \left( a_1 + a_x x + a_y y + \sum_{i=1}^n w_i U(|P_i - P|) \right)^T$$

对给定的控制点  $P_i = (x_i, y_i)$ ，上式参数有以下等式确定：

$$\begin{bmatrix} 0 & U(r_{12}) & U(r_{13}) & \cdots & U(r_{1n}) & 1 & x_1 & y_1 \\ U(r_{21}) & 0 & U(r_{23}) & \cdots & U(r_{2n}) & 1 & x_2 & y_2 \\ U(r_{31}) & U(r_{32}) & 0 & \cdots & U(r_{3n}) & 1 & x_3 & y_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ U(r_{n1}) & U(r_{n2}) & U(r_{n3}) & \cdots & 0 & 1 & x_n & y_n \\ 1 & 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ x_1 & x_2 & x_3 & \cdots & x_n & 0 & 0 & 0 \\ y_1 & y_2 & y_3 & \cdots & y_n & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \\ a_1 \\ a_x \\ a_y \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 \\ x'_2 & y'_2 \\ x'_3 & y'_3 \\ \vdots & \vdots \\ x'_n & y'_n \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

其中：

$$U(r) = \begin{cases} r^2 \log(r^2), r \neq 0 \\ 0, r = 0 \end{cases}$$

解出  $\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \\ a_1 \\ a_x \\ a_y \end{bmatrix}$ ，可得：

$$P = f(P) = [U(|P_1 - P|) \quad U(|P_2 - P|) \quad U(|P_3 - P|) \quad \cdots \quad U(|P_n - P|) \quad 1 \quad x \quad y] \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \\ a_1 \\ a_x \\ a_y \end{bmatrix}$$

## 方案设计

本部分包含大作业实现方案的设计。

### 开发环境

1. 开发平台：Windows 10 (1903)
2. 开发语言：C#
3. 界面框架：WPF
4. 开发 IDE：Visual Studio 2019
5. 运行环境：包含 .NetFramework 4.7.2 或以上版本的 Windows 平台

### 依赖项

除基本必须的 .NetFramework 4.7.2 内部库外，无任何其他依赖项，未使用 OpenCvSharp3 等第三方库。

### 程序设计

本项目采用前后端分离的设计思路，程序主体结构清晰，便于阅读与调试，关于本项目的详细说明及实现亮点，请参见技术报告。

## 前端

文件名	描述
App.xaml	程序界面文件，规定资源字典等信息
Button.xaml	资源字典，来自互联网，用于美化界面
Error.xaml	错误提示框
Info.xaml	信息提示框
MainWindow.xaml	程序主界面逻辑文件，详细逻辑请参见源代码
Success.xaml	成功提示框

## 后端

文件名	描述
ImageIO.cs	图像读取写入类，实现图像 IO 与编辑
ImageProcess.cs	图像处理类，包含所有的插值与变换算法的实现
FaceTransformation.cs	人脸变换类，包含 TPS 算法的实现
MyMatrix.cs	矩阵运算类，实现了本项目中必要的矩阵运算

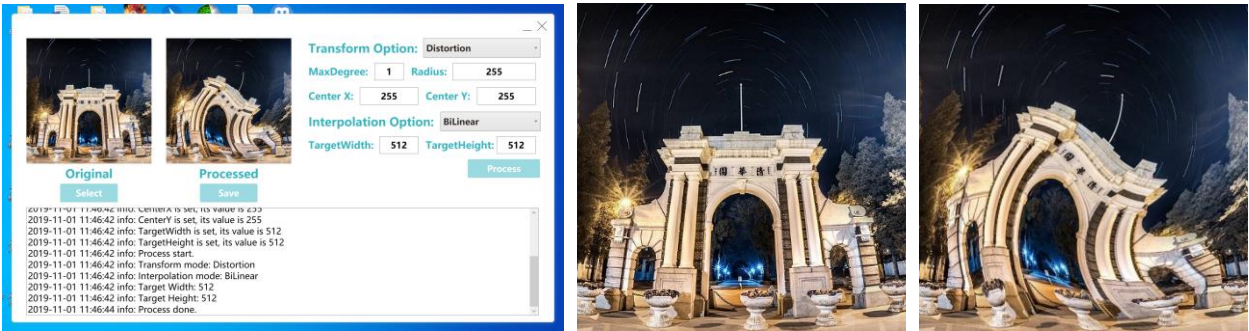
## 项目效果

本部分包含项目最终的实现效果，关于程序使用及参数详解，请参见技术报告。

### 扭曲变换

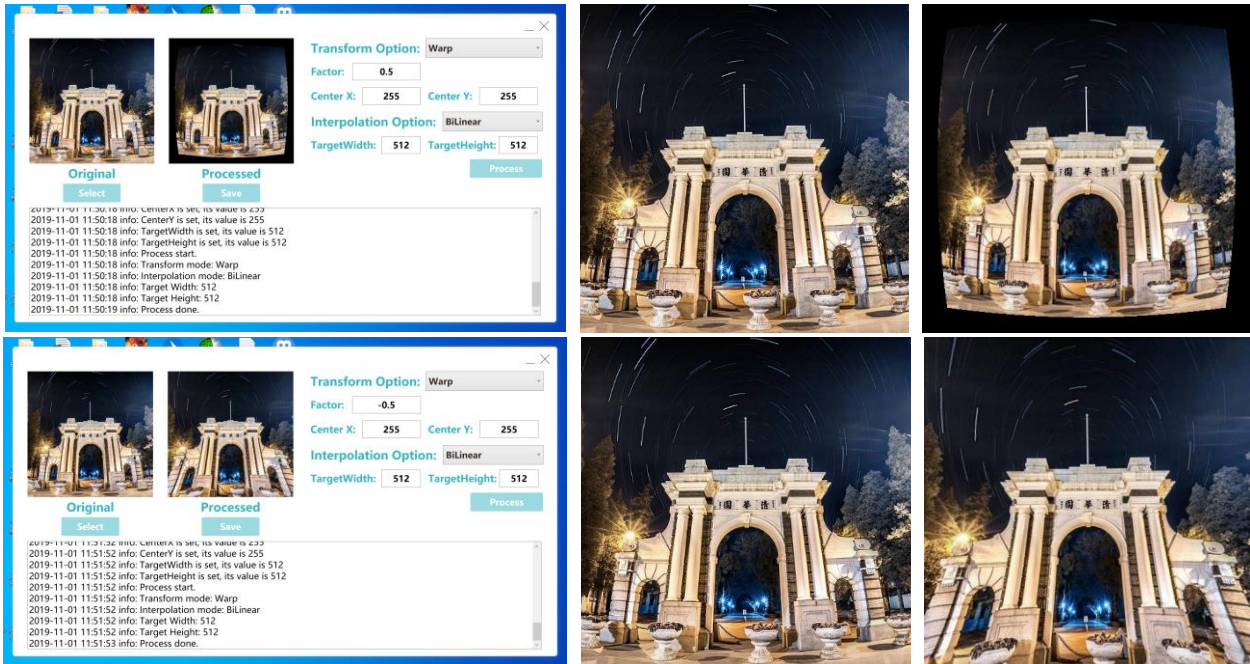
本部分包含扭曲变换部分的实现效果，插值方式采用统一的双线性插值，其他效果请运行程序。

### 旋转扭曲



(左：参数设置，中：原图片，右：处理结果)

# 畸变矫正

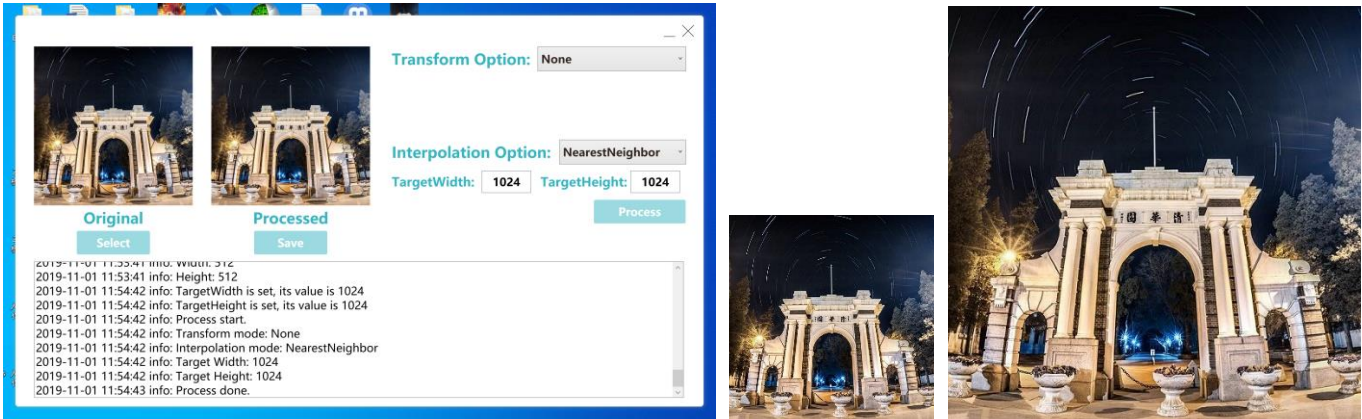


(左：参数设置，中：原图片，右：处理结果)

# 插值方法

本部分包含插值方法部分的实现效果，变换方式采用统一的无变换，其他效果请运行程序。

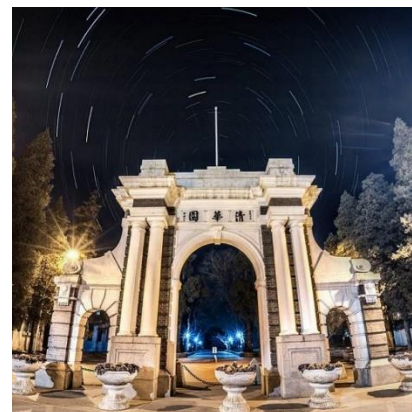
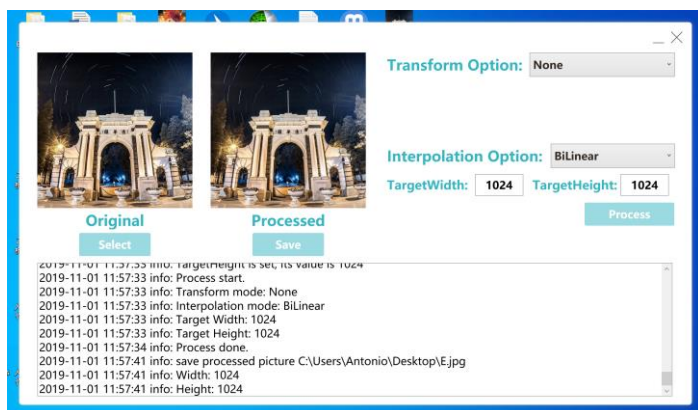
# 最近邻插值



(左：参数设置，中：原图片，右：处理结果)

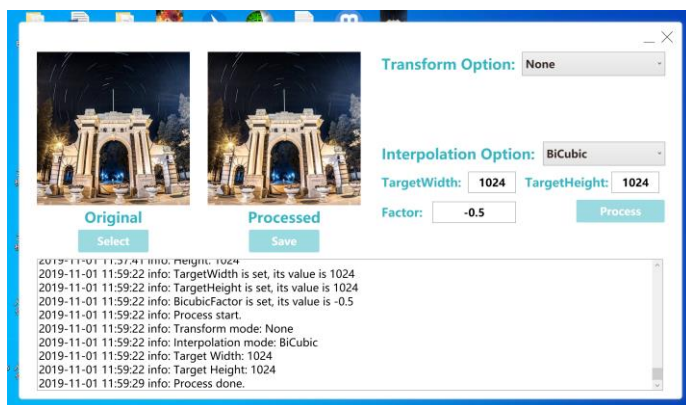


## 双线性插值



(左：参数设置，中：原图片，右：处理结果)

## 双三次插值

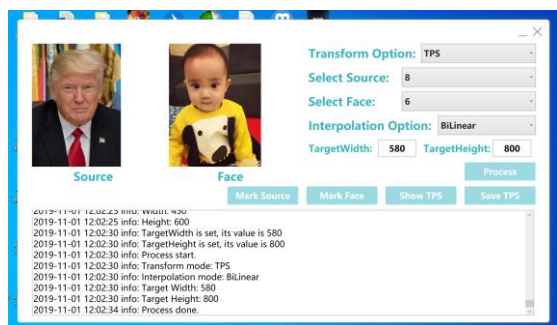


(左：参数设置，中：原图片，右：处理结果)

## TPS 网格变形

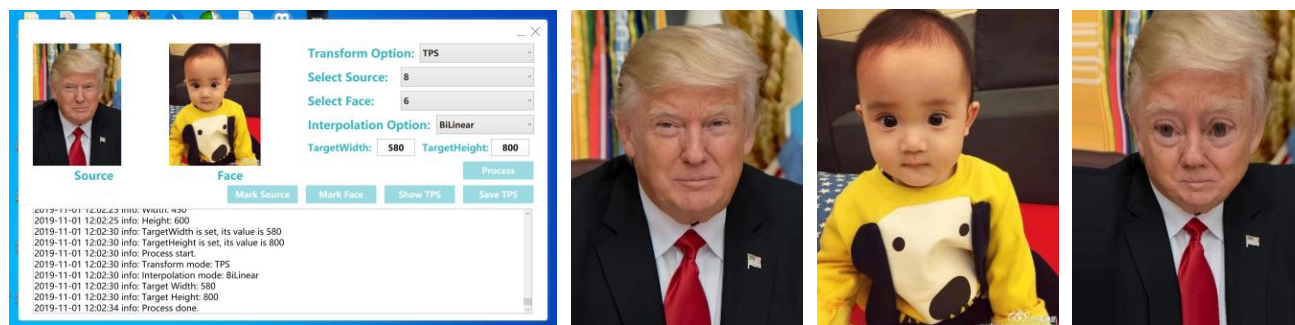
本部分包含 TPS 网格变形的实现效果，插值方式采用统一的双线性插值，其他效果请运行程序。

## 人脸未预对齐【正式版本已移除此方法】



(左 1：参数设置，左 2：源图片，右 2：人脸图片，右 1：处理结果)

## 人脸预对齐



(左 1: 参数设置, 左 2: 源图片, 右 2: 人脸图片, 右 1: 处理结果)

## 误差分析

本部分包含本项目处理过程中的误差分析。

### 观测误差

由于本项目处理的为数字图像，认为真实图像的像素点对应的颜色值为无限位数，由于采用 uint8 数据类型存储颜色值，数字图像的观测误差上限为 0.5。

### 舍入误差

本项目处理过程中变量均为 double 数据类型，其有效位数为 15，但所有的 double 变量在最后都会转换为 uint8 数据类型，此转换对舍入误差的影响远远大于计算机运算过程，故舍入误差上限为 0.5。

### 模型误差

由于本项目使用的为数字图像，图像转化为数字图像的模型误差无法计算，故可认为模型误差为 0。

### 方法误差

此部分为方法误差的详细分析，由于数字图像是真实图像的离散化结果，故以下分析过程使用差分代替求导。

### 扭曲变换

本部分为扭曲变换的方法误差分析。

### 旋转扭曲

考虑 $x$ 方向

$$|R_x(P)| \leq \max\left(\left|\Delta_x f(P)\right| \cdot |\Delta x|\right)$$

对于y方向同理

$$|R_y(P)| \leq \max\left(\left|\Delta_y f(P)\right| \cdot |\Delta y|\right)$$

由于 $|\Delta x| = |\Delta y| = 0.5$ , 故

$$|R(P)| \leq 0.5 \times \left(\max\left|\Delta_x f(P)\right| + \max\left|\Delta_y f(P)\right|\right) \leq 0.5 \times (1 + 1) \leq 1$$

## 畸变矫正

考虑x方向

$$|R_x(P)| \leq \max\left(\left|\Delta_x f(P)\right| \cdot |\Delta x|\right)$$

对于y方向同理

$$|R_y(P)| \leq \max\left(\left|\Delta_y f(P)\right| \cdot |\Delta y|\right)$$

由于 $|\Delta x| = |\Delta y| = 0.5$ , 故

$$\begin{aligned} |R(P)| &\leq 0.5 \times \left(\max\left|\Delta_x f(P)\right| + \max\left|\Delta_y f(P)\right|\right) \leq 0.5 \times \frac{\beta}{\text{Scale}} \cdot 2 \cdot \left(\text{Width}_{\text{原图像}}^2 + \text{Height}_{\text{原图像}}^2\right) \\ &\leq \frac{\beta}{\text{Scale}} \cdot \left(\text{Width}_{\text{原图像}}^2 + \text{Height}_{\text{原图像}}^2\right) \end{aligned}$$

当 $\text{Scale} = \text{Width}_{\text{原图像}} \times \text{Height}_{\text{原图像}}$ 时

$$|R(P)| \leq \beta \cdot \left(\frac{\text{Width}_{\text{原图像}}}{\text{Height}_{\text{原图像}}} + \frac{\text{Height}_{\text{原图像}}}{\text{Width}_{\text{原图像}}}\right)$$

因而当 $\text{Width}_{\text{原图像}} \approx \text{Height}_{\text{原图像}}$ 时, 此方法误差较小, 否则需要更换计算方法。

## 插值方法

本部分为插值方法的方法误差分析。

### 最近邻插值

考虑x方向

$$|R_x(P)| \leq \max\left(\left|\Delta_x f(P)\right| \cdot |\Delta x|\right)$$

对于y方向同理

$$|R_y(P)| \leq \max\left(\left|\Delta_y f(P)\right| \cdot |\Delta y|\right)$$

由于 $|\Delta x| = |\Delta y| = 0.5$ ，故

$$|R(P)| \leq 0.5 \times \left( \max |\Delta_x f(P)| + \max |\Delta_y f(P)| \right)$$

## 双线性插值

考虑 $x$ 方向

$$f(P') = v \cdot f\left(P' \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [P'] \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) + (1-v) \cdot f\left(P' \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + ([P'] + [1 \quad 1]) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$$

其中

$$f\left(P' \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [P'] \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = u \cdot ([P'] + [0 \quad 0]) + (1-u) \cdot f([P'] + [1 \quad 0])$$

$$f\left(P' \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + ([P'] + [1 \quad 1]) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = u \cdot ([P'] + [0 \quad 1]) + (1-u) \cdot f([P'] + [1 \quad 1])$$

分别求余项得：

$$\left| R_x\left(P \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [P] \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) \right| \leq \frac{1}{2!} \max |\Delta_x^2 f(P)| \cdot \max |u \cdot (1-u)| \leq 0.125 \times \max |\Delta_x^2 f(P)|$$

$$\left| R_x\left(P \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + ([P] + [1 \quad 1]) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) \right| \leq \frac{1}{2!} \max |\Delta_x^2 f(P)| \cdot \max |u \cdot (1-u)| \leq 0.125 \times \max |\Delta_x^2 f(P)|$$

从而可得：

$$\begin{aligned} |R_x(P)| &\leq 0.125 \times \left( v \cdot \left| R_x\left(P \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [P] \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) \right| + (1-v) \cdot \left| R_x\left(P \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + ([P] + [1 \quad 1]) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) \right| \right) \\ &\leq 0.125 \times \max |\Delta_x^2 f(P)| \end{aligned}$$

$y$ 方向同理得

$$|R_y(P)| \leq 0.125 \times \max |\Delta_y^2 f(P)|$$

故

$$|R(P)| \leq 0.125 \times \left( \max |\Delta_x^2 f(P)| + \max |\Delta_y^2 f(P)| \right)$$

## 双三次插值

考虑 $x$ 方向，推导方法与双线性插值类似，利用单方向的表达式逐项求误差之后相加，发现每一项误差上限均相同，在此直接给出推到结果，不再赘述推导方法：

$$|R_x(P)| \leq \frac{1}{4!} \max |\Delta_x^4 f(P)| \cdot \max |(u+1) \cdot u \cdot (u-1) \cdot (u-2)| \leq 0.0234375 \times \max |\Delta_x^4 f(P)|$$

$y$ 方向同理得

$$|R_y(P)| \leq 0.0234375 \times \max |\Delta_y^4 f(P)|$$

故

$$|R(P)| \leq 0.0234375 \times \left( \max |\Delta_x^4 f(P)| + \max |\Delta_y^4 f(P)| \right)$$

## TPS 网格变形

本部分为 TPS 网格变形的误差分析。

### 矩阵求逆

本项目中矩阵求逆使用的是高斯消元法，即： $AA^{-1} = I$

一般的，对于线性方程组 $Ax = b$ ，使用高斯消元法产生的误差与舍入误差之间的关系可用以下公式表述：

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\|A\|\|A^{-1}\| \left( \frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right)}{1 - \|A^{-1}\|\|\delta A\|}$$

对于非奇异 $n$ 阶方阵 $A$ ，称 $\|A\|\|A^{-1}\|$ 为矩阵 $A$ 的条件数，记为 $cond(A)$ 。

特别的， $cond(A)_2 = \sqrt{\frac{\max \lambda(A^T A)}{\min \lambda(A^T A)}}$ ，因而整个求逆过程的误差主要取决于 $cond(A)$ ，只要 $A$ 的奇异值数量级相当，那么这一误差将会较小。

注：

$$\|x\|_{\infty} = \max_i |x_i|, (l_{\infty} \text{范数})$$

$$\|x\|_1 = \sum_{i=1}^n |x_i|, (l_1 \text{范数})$$

$$\|x\|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}, (l_2 \text{范数})$$

$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, (l_p \text{范数})$$

### 人脸预对齐

人脸预对齐时使用了最小二乘法，在求解最小二乘法的过程中使用了高斯消元法的矩阵求逆，求解过程中认为 $A$ 的奇异值数量级相当，忽略求逆的误差。

求解误差为：

$$\Delta_1 = \delta([P' \quad 1]A^*) \leq \|A^*\| \|\delta[P' \quad 1]\|$$

此误差表示真实结果与所求结果之间的误差。

最小二乘法的误差为

$$\Delta_2 = \sum_{i=1}^n \|[P' \quad 1]A^* - P\|_2^2$$

此误差表示真实结果与源图片目标点之间的误差之和。

### TPS 变换

TPS 变换的求解过程与人脸预对齐相似，在求解过程中使用了高斯消元法的矩阵求逆，求解过程中认为 $A$ 的奇

异值数量级相当，忽略求逆的误差，因而系数误差都比较小，之后误差的来源主要是 TPS 径向基函数，然而 TPS 径向基函数求导为  $2 \cdot |r| \cdot (1 + \log r^2)$ ，其最大值取决于  $r$  的范围， $r$  的范围又取决于输入图片，从而无法直接给出 TPS 变换的误差上界。

## 总结反思

以下是本项目的反思与总结。

### 功能实现

回顾本项目，一共实现了 6 项功能：旋转扭曲、畸变矫正、TPS 变换、最近邻插值、双线性插值和双三次插值。实现这些功能的过程中，笔者对于图像这一文件的理解更加深入了，从其在计算机中的存储方式到对其的处理方式，笔者学习到了处理图片的基本技能，对于图片的相关操作的底层实现有了一定得了解，并且理解了一切的图像变换都是建立在一系列对图像的矩阵变换的基础之上，可见线性代数的知识对于处理图像问题的重要性，在实际处理过程中，笔者也了解到图像对齐在图像处理中的重要性，在 TPS 变换中，如果不加入图像对齐的操作，其处理结果将会无法达到预期效果，甚至无法求解，在现代化的图像处理软件中，图像对齐仍然是处理图像过程中十分重要的操作。在本项目之前，笔者也使用过一些其他工具，比如 MATLAB 和 Python 处理过数字图像，但由于这些工具已将基本功能封装好，笔者在本项目之前并不了解其底层实现，通过本项目，笔者锻炼了自己的代码能力也对图像相关操作的底层实现有了较为深入的了解，这对于笔者以后处理图像问题有极大的帮助。

### 误差分析

在本项目的误差分析过程中，笔者深刻体会到了计算机数据类型存储的舍入误差对实际计算的影响，例如高斯消元法，当需要求解的方程是一个病态方程组，其系数数量级差距巨大时，计算机利用高斯消元法求解将会不准确，这一点在方法误差部分已经分析过了，在分析此误差之前，笔者感到无从下手，在搜索了大量的资料之后，笔者才慢慢得出了方法误差分析过程中的公式，但是笔者所作的误差分析在几经思考之后发现仍有不准确的地方，有待进一步的求证与思考，可见误差分析并不是一件简单的事情，并且它是十分重要的。

### 项目构建

在完成本项目的过程中，笔者由于搜集资料数目过多，并且资料多来自互联网，在完成过程中并未及时记录自己查阅的网站的地址，所以无法列出完整的参考文献，笔者在此表示歉意，并以此为戒，以后做好参考文献的记录和管理，此外关于本项目程序的使用说明，请参见技术报告。