数值分析与算法大作业 1

人脸变形

班级：自 66

姓名：夏卓凡

学号：2016011496

2018 年 11 月 27 日

目录

1 需求分析

1.1 问题形式化表述

本次大作业求解问题可以定义如下符号以进行形式化的表述：给定源人脸图像 $\mathbf{I}_{\text{src}}(x,y)$ 和导向人脸图像 $\mathbf{I}_{\text{guide}}(x,y)$，求解目标人脸图像 $\mathbf{I}_{\text{dst}}(x,y)$，使得在数量 (Quantitative) 上人脸关键点在空间上具有更加相似的模式 (Pattern)，质量 (Qualitative) 上看上去目标人脸与导向人脸更加一致，具有相同的“神韵”。

所处理人脸图像均保存为 $h \times w$ 的 4 通道张量，h 为图像高度，w 为图像宽度，通道数依次为 ARGB，各为 8 位无符号整数，组合为 32 位整数存储。程序中的 x、i 一般为 w 方向坐标，y、j 一般为 h 方向坐标。

特别地，为了简化问题，本次大作业中所有人脸图像在分析和处理时，**只考虑有且仅有一张人脸的情况**。

1.2 空间变换方法

为了解决上述问题，要从人脸关键点的移动出发，构造整个人脸图像的空间变形，最终改变人脸的形状，变换到相似的模式。根据本次大作业需求，我们需要实现 B 样条变换和 TPS 变换两种空间变换方法。

1.2.1 B 样条变换

B 样条变换源于 B 样条曲线，即曲线上的点的空间坐标由各个控制点确定，其控制关系满足

$$P_{k,n}(x,y;t) = \sum_{i=0}^{n} P_{i+k}(x,y) G_{i,n}(t) \tag{1.1}$$

其中 t 为参数曲线的参数，P_i 为控制点。如果原先的控制点在一条直线上，该变换实现了直线到曲线的变换。对应于平面到曲面的变换，可在 x 方向和 y 方向上分别参数化，即产生网格化的控制点阵，控制关系为

$$P_{k,l,n}(x,y;u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n} P_{i+k,j+l}(x,y) G_{i,n}(u) G_{j,n}(v) \tag{1.2}$$

其中 u、v 为参数，$P_{i,j}$ 为控制点，$G(\cdot)$ 为 B 样条基函数。

在实际使用中，需要将源图像网格化，即指定一个网格大小 s，可以确定整张图像的控制点。x 方向上控制点个数为 $C_x = \lceil \frac{w}{s} \rceil$，$y$ 方向上控制点个数为 $C_y = \lceil \frac{h}{s} \rceil$，于是控制点为 $P_{i,j}$，其中 $0 \le i \le C_x$，$0 \le j \le C_y$。这样就确定了任意一点的 $P(x,y)$ 参数 u 和 v，即

$$\begin{cases} u = \dfrac{x}{s} - \left\lfloor \dfrac{x}{s} \right\rfloor \\ v = \dfrac{y}{s} - \left\lfloor \dfrac{y}{s} \right\rfloor \end{cases} \tag{1.3}$$

对于本次大作业中的三次 B 样条逆变换，应取 $n = 3$ 的微分形式，即

$$\Delta P_{k,l}(x,y;u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} \Delta C_{i+k,j+l}(x,y) G_{i,n}(u) G_{j,n}(v) \tag{1.4}$$

其中控制点 C 邻域下标由下式确定

$$\begin{cases} k = \left\lfloor \dfrac{x}{s} \right\rfloor - 1 \\ l = \left\lfloor \dfrac{y}{s} \right\rfloor - 1 \end{cases} \tag{1.5}$$

3 次 B 样条基函数为

$$\begin{cases} G_{0,3}(t) = \dfrac{1}{6}(1-t)^3 \\ G_{1,3}(t) = \dfrac{1}{6}(3t^3 - 6t^2 + 4) \\ G_{2,3}(t) = \dfrac{1}{6}(-3t^3 + 3t^2 + 3t + 1) \\ G_{3,3}(t) = \dfrac{1}{6}t^3 \end{cases} \tag{1.6}$$

其中 $t \in [0,1]$。

可见 3 次 B 样条使用了 16 邻域信息，任意点位移是邻域内网格点位移的线性组合。本次大作业中主要使用这个公式近似 B 样条逆变换，取定 $s = 20$，一次变形多次衰减移动网格点，逼近较优的结果。

1.2.2 TPS 变换

TPS(Thin Plate Spline) 变换是另一种空间变换方法，从平面到曲面的变换根据控制点及最小化弯曲能量函数确定，即求取变换函数 f[1] 以最小化

$$I_f = \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) \mathrm{d}x\mathrm{d}y \tag{1.7}$$

事实上，大作业说明讲义上的这个公式是 TPS 能量函数中的一项，简单地说，这一项实现了弯曲这一非刚性变变换。而真正的 TPS 变换目标函数为

$$E = \sum_{i=1}^{K} ||P_{\text{guide}} - f(P_{\text{src}})||_2^2 + \mu I_f \tag{1.8}$$

其中 μ 为松弛参数，控制前一项仿射变换与后一项弯曲变换的比例以及抑制噪声。对 K 个控制点的情况，得到所求变换函数的表达式

$$f(P; \mathbf{w_0}, \mathbf{w_x w_y}, \mathbf{w_1}, \ldots, \mathbf{w_K}) = \mathbf{w_0} + \mathbf{w_x} P_x + \mathbf{w_y} P_y + \sum_{i=1}^{K} \mathbf{w_i} U(||P - C_i||_2) \tag{1.9}$$

其中 TPS 径向基函数 $U(r) = r^2 \ln r^2$，C_i 为 TPS 变换控制点。前面的线性部分是仿射变换，后面的径向基部分为弯曲变换，只要确定各个权重向量 $\mathbf{w}$ 的值即可确定变换函数。注意为了使 $f \in C^{(2)}(\mathbb{R}^2)$，权重向量满足约束 [2]

$$\sum_{i=0}^{K} \mathbf{w_i} = \mathbf{0} \quad \sum_{i=0}^{K} \mathbf{w_i} x_i = \mathbf{0} \quad \sum_{i=0}^{K} \mathbf{w_i} y_i = \mathbf{0} \tag{1.10}$$

其中 x_i、y_i 为第 i 个控制点 C_i 的坐标。

根据大作业说明讲义以及 [2] 中的最基本求解方法，应构造如下线性方程组进行求解

$$\begin{bmatrix} 0 & U(r_{12}) & U(r_{13}) & \cdots & U(r_{1K}) & 1 & x_1 & y_1 \\ U(r_{21}) & 0 & U(r_{23}) & \cdots & U(r_{2K}) & 1 & x_2 & y_2 \\ U(r_{31}) & U(r_{32}) & 0 & \cdots & U(r_{3K}) & 1 & x_3 & y_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ U(r_{1K}) & U(r_{2K}) & U(r_{3K}) & \cdots & 0 & 1 & x_K & y_K \\ 1 & 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ x_1 & x_2 & x_3 & \cdots & x_K & 0 & 0 & 0 \\ y_1 & y_2 & y_3 & \cdots & y_K & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w_1} \\ \mathbf{w_2} \\ \mathbf{w_3} \\ \vdots \\ \mathbf{w_K} \\ \mathbf{w_0} \\ \mathbf{w_x} \\ \mathbf{w_y} \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 \\ x'_2 & y'_2 \\ x'_3 & y'_3 \\ \vdots & \vdots \\ x'_K & y'_K \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{1.11}$$

其中 x'_i、y'_i 为变换目标点 T'_i 的坐标，本次大作业中主要使用逆变换公式，对人脸关键点，$K = 68$，并且控制点 C 设定为导向人脸图像关键点 P_{guide}，目标点 T 设定为源人脸图像关键点 P_{src}。

1.3 人脸预对齐方法

不难发现实际给出的测试样例和真实场景的人脸图像，它们人脸关键点的坐标与图片大小密切相关，如果不作处理地应用各种空间变换方法，会造成各种问题；轻则变换后图像内容扭曲过大或者黑边过多，重则使要计算的 TPS 变换矩阵不可逆 (Non-Invertable) 或者 B 样条变换无法工作，从而无法完成空间变换。

考虑到之前的实验室科研任务中，经常使用某些人脸特征点估计模型，如 MTCNN 等将待处理人脸图像的 5 个特征点估计出来，应用仿射变换对齐至标准脸型，实现人脸数据集的对齐；这里的人脸预对齐也采用类似处理，将待处理的人脸导向图中的关键点进行一次仿射变换，将其变换到与源人脸图像中的关键点，使得它们生成人脸结构的大小接近，姿态一致，位置靠近。

这一部分使用最小二乘法估计一个仿射变换，即求解一个变换矩阵 $\mathbf{A}^*$，最小化空间坐标的均方误差

$$L = \sum_{i=1}^{K} ||\mathbf{A}^* P_{i-\text{guide}} - P_{i-\text{src}}||_2^2 \tag{1.12}$$

其中 $\mathbf{A}$ 规定为 [3]

$$\mathbf{A} = \begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ t_x & t_y & 1 \end{bmatrix} \tag{1.13}$$

实际为求解 $\mathbf{A}^*$，做以下变形

$$[x \quad y \quad 1]\mathbf{A} = [x' \quad y' \quad 1] \tag{1.14}$$

其中 $[x \quad y \quad 1]$ 为导向图像中的关键点坐标，记为 $\mathbf{X}$，$[x' \quad y' \quad 1]$ 为人脸源图像中的关键点坐标，记为 $\mathbf{y}$，那么方程可以写为

$$\mathbf{XA} = \mathbf{y} \tag{1.15}$$

运用矩阵形式最小二乘法解欠定方程的技巧，求解以下方程即可确定所求的 $\mathbf{A}^*$

$$\mathbf{X}^\top \mathbf{X}\mathbf{A}^* = \mathbf{X}^\top \mathbf{y} \tag{1.16}$$

之后更新导向人脸图像的关键点坐标

$$\mathbf{X}' = \mathbf{X}\mathbf{A}^* \tag{1.17}$$

1.4 图像插值方法

在得到目标人脸图像 $\mathbf{I}_{\text{dst}}(x,y)$ 上各个点 (x,y) 与源人脸图像 $\mathbf{I}_{\text{src}}(x,y)$ 上各个点 (x',y') 的对应关系，由于所得源图像上的点 (x',y') 一般不是整数，所以需要采用各种插值技术确定 (x',y') 各个通道的像素值，也就是 (x,y) 各个通道的像素值。本次大作业中使用的插值方法有 3 种：最近邻插值、双线性插值和双三次插值。

1.4.1 最近邻插值

对源图像中的点 (x',y')，按照最近邻插值的规则，其值为 $f(x',y')=f([x'],[y'])$；即按照取整原则选取 (x',y') 四舍五入最近的点的像素值作为 (x',y') 的像素值，是一种 1 邻域插值。

1.4.2 双线性插值

对源图像中的点 (x',y')，令 $u=x'-\lfloor x'\rfloor$，$v=y'-\lfloor y'\rfloor$，$i=\lfloor x'\rfloor$，$j=\lfloor y'\rfloor$，则

$$f(x',y')=\begin{bmatrix}1-u & u\end{bmatrix}\begin{bmatrix}f(i,j) & f(i,j+1)\\ f(i+1,j) & f(i+1,j+1)\end{bmatrix}\begin{bmatrix}1-v\\ v\end{bmatrix} \tag{1.18}$$

这相当于使用 (x',y') 周围 4 个点的像素值按坐标线性变化的规律决定 (x',y') 的像素值，是一种 4 邻域插值。

1.4.3 双三次插值

对源图像中的点 (x',y')，令 $u=x'-\lfloor x'\rfloor$，$v=y'-\lfloor y'\rfloor$，$i=\lfloor x'\rfloor$，$j=\lfloor y'\rfloor$，则

$$\begin{aligned}f(x',y')=&\begin{bmatrix}S(u+1) & S(u) & S(u-1) & S(u-2)\end{bmatrix}\cdot\\ &\begin{bmatrix}f(i-1,j-1) & f(i-1,j) & f(i-1,j+1) & f(i-1,j+2)\\ f(i,j-1) & f(i,j) & f(i,j+1) & f(i,j+2)\\ f(i+1,j-1) & f(i+1,j) & f(i+1,j+1) & f(i+1,j+2)\\ f(i-1,j+2) & f(i+2,j) & f(i+2,j+1) & f(i+2,j+2)\end{bmatrix}\begin{bmatrix}S(v+1)\\ S(v)\\ S(v-1)\\ S(v-2)\end{bmatrix}\end{aligned} \tag{1.19}$$

其中 $S(t)$ 为三次插值基函数，定义为

$$S(t)=\begin{cases}1-2|t|^2+|t|^3 & |t|\le 1\\ 4-8|t|+5|t|^2-|t|^3 & 1<|t|<2\\ 0 & \text{其他}\end{cases} \tag{1.20}$$

这相当于使用 (x',y') 周围 16 个点的像素值按坐标三次基函数映射后的规律决定 (x',y') 的像素值，是一种 16 邻域插值。

2 方案设计

2.1 程序环境

使用 C#(.NET Framework 4.7.2) 语言，使用 Visual Studio 2017 进行开发，使用第三方库为 OpenCvSharp3(3.4.4.20181118) 和 DlibDotNet(19.15.0.20181008)。在生成后需要将库的相关依赖的 DLL 放在应用程序根目录下，在 Windows 10 下进行构建与测试。

2.2 依赖项说明

使用 C# 核心语言进行全部必做功能的开发，包括各种空间变换方法、预对齐方法和插值方法，其中用到了 OpenCvSharp3 的相关数据结构如 `Point2d` 和 `Vec2d` 用来提高程序的可读性。为了使程序能够处理一般的人脸图像，使用了 DlibDotNet 的人脸关键点检测器 [4]，在读入任意人脸图像后进行人脸关键点检测，从而使变形方法具有更广泛的使用。

2.3 系统设计

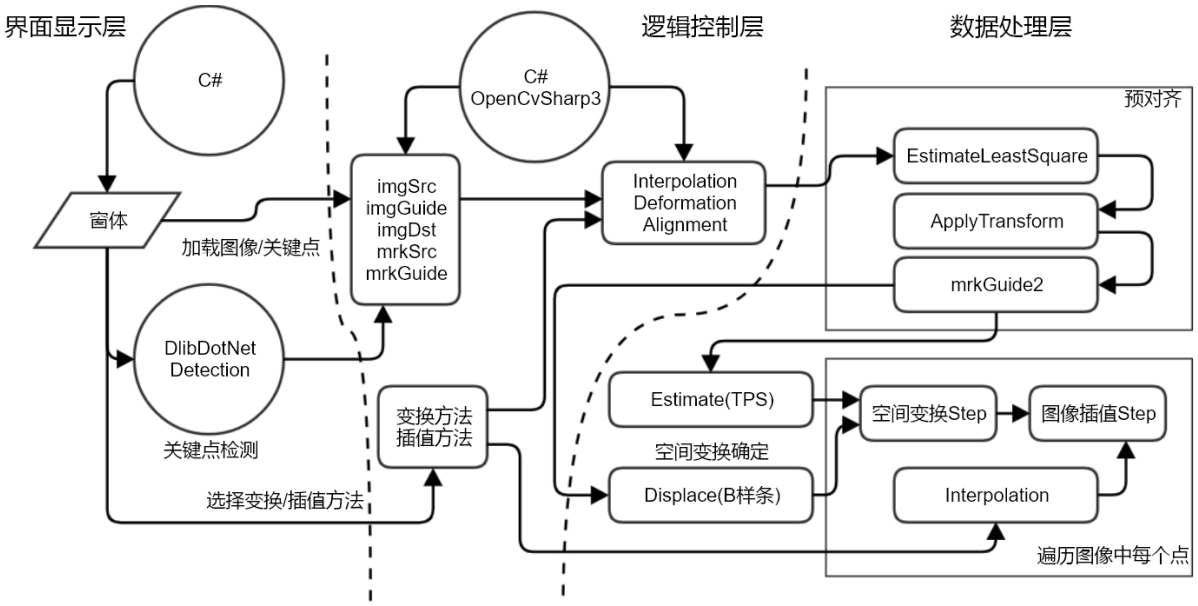

图 1: 系统设计图

系统设计如上图所示，首先在界面显示层上用户选择 9 张测试样例图片（有关键点）或者加载真实图片（无关键点），之后可以选择加载关键点或者使用内置的关键点检测（DlibDotNet）完成关键点检测。然后逻辑控制层根据选定的空间变换方法和图像插值方法，将源图像、目标图像和对应的关键点数据送给数据处理层，首先完成预对齐，之后遍历图像中每个点，根据空间变换确定目标图像在源图像的对应点，再执行图像插值确定像素值，完成目标图像的构建。

这里有一些实现上的技巧，比如 C# 内置的 `Bitmap` 类的 `GetPixel` 和 `SetPixel` 函数执行效率过低，改用 `BitmapData` 类配合原始指针实现直接内存访问，可以高效地完成插值这类大规模遍历任务。还有考虑到 `Bitmap` 的内存数据存在 4 字节对齐问题，直接使用 32 位的 ARGB 像素值并打包为 `Int32` 处理，而不是 24 位的 RGB 像素值，这样做避免了指针换行时的对齐的问题，开发起来一致性更高。

2.4 实验结果与分析

以下给出一部分实验结果并作简要的分析，以证实方案的正确性和合理性。

2.4.1 验证大作业说明讲义中的样例 1

首先验证特朗普（Donald John Trump）和吉诺比利（Emanuel David Ginóbili）的变换，使用带预对齐的 TPS 变换以及双线性插值。

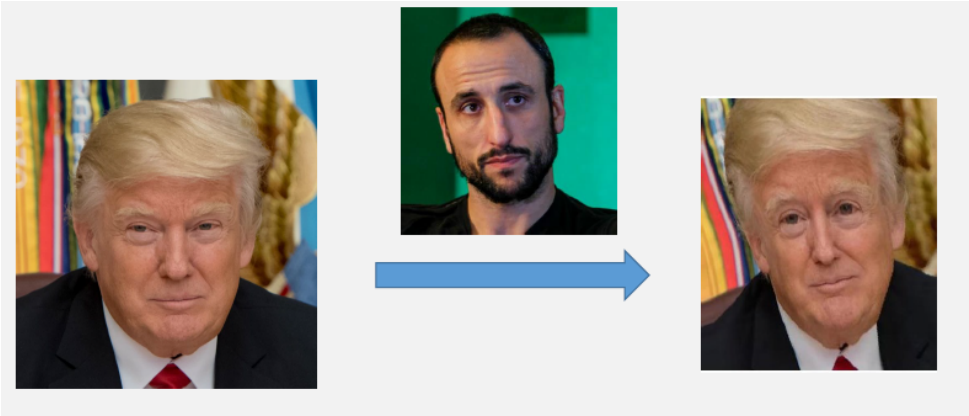

图 2: 测试样例 1

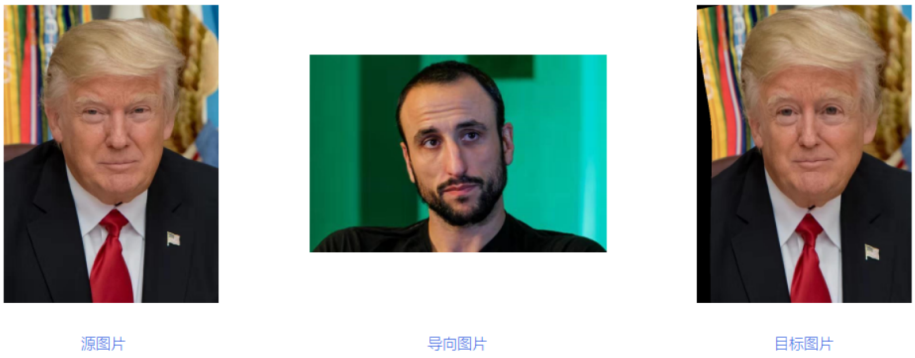

图 3: 测试样例 1-结果

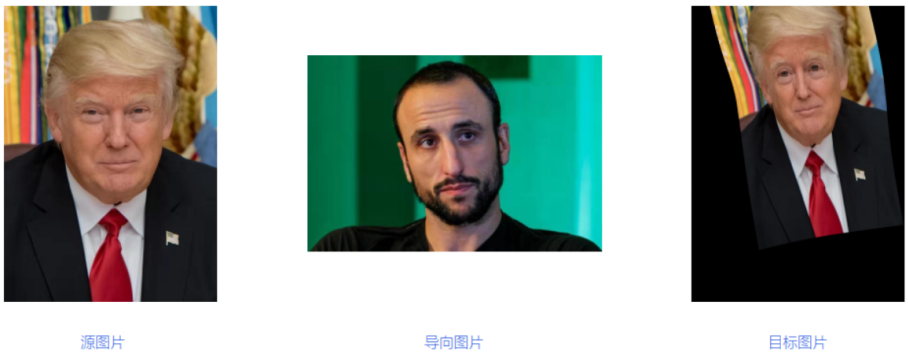

图 4: 测试样例 1-结果-无预对齐

根据视觉效果，可以推测讲义中的结果应为 TPS 变换在无预对齐条件下生成的。所以可以看出，本方案是正确的。关于预对齐，可以看出预对齐使得人脸无关的空间变化更少，体现在人脸的位置基本不变，姿态更加一致，生成图像的黑边更少。

2.4.2 验证大作业说明讲义中的样例 2

接下来验证特朗普（Donald John Trump）和安德森（Anderson Chao）的变换，使用带预对齐的 TPS 变换以及双线性插值。

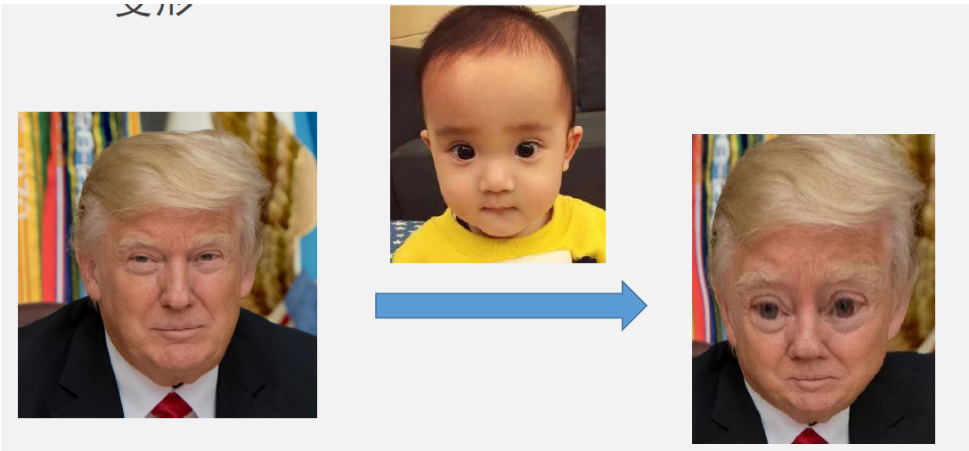

图 5: 测试样例 2

图 6: 测试样例 2-结果

图 7: 测试样例 2-结果-无预对齐

这一结果同样证实了方案的有效性和预对齐的必要性，其中预对齐能有效将目标图像和原图像的尺度（scale）保持一致，只做人脸模式的变化而与导向人脸图像具体的图片大小无关。

2.4.3 插值方法的比较

这一部分我们将比较最近邻插值、双线性插值和双三次插值的性能和效果。测试用例为讲义中的样例 1，整体图像的效果如下图所示。

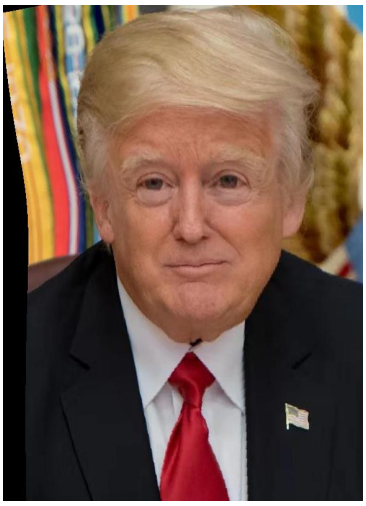

图 8: 最近邻插值

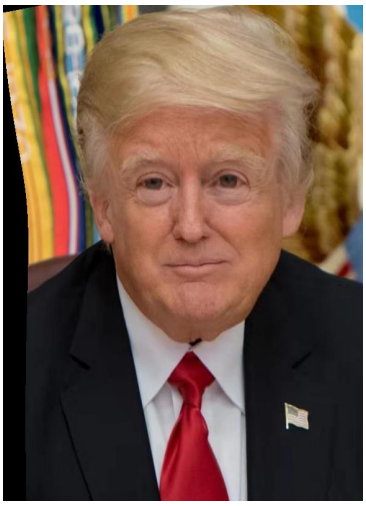

图 9: 双线性插值

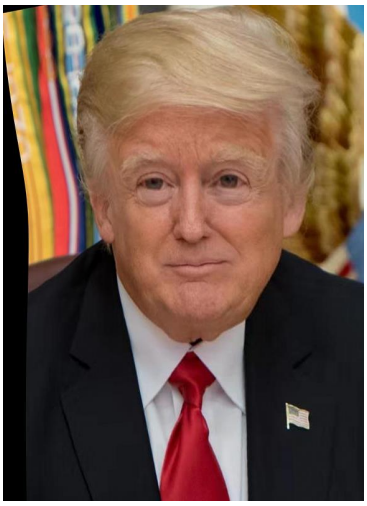

图 10: 双三次插值

下图为插值结果的局部图片，从左至右依次为最近邻插值、双线性插值和双三次插值。

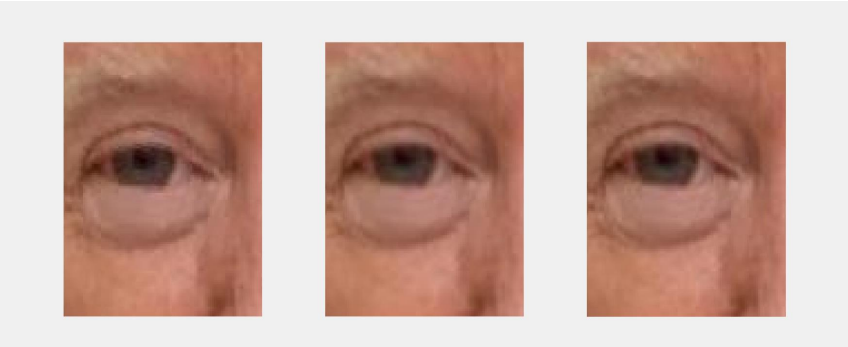

图 11: 眼部局部结果对比

通过这组视觉结果对比，在局部上，三种插值方法的图像清晰程度为双三次插值 > 双线性插值 > 最近邻插值，但对于原本分辨率足够高的图像来说，全局看上去三种方法没有太大差别。运行速度为双三次插值 < 双线性插值 < 最近邻插值，本程序在运行某些非常大的图片的处理时会有一些较为明显的延迟。

2.4.4 验证一般人脸图像的变换

图 12: 一般人脸图像测试

为了避免不必要的麻烦，我选取了自己在 2018 年 6 月 10 日系三口合影中的照片作为源人脸图像，导向人脸图像为安德森（Anderson Chao），使用 B 样条变换和双三次插值，得到如图所示的效果。值得注意的是，B 样条变换是局部的变换，只能尽可能满足特征点的最大可能移动，而不像 TPS 变换可以在全局上改变图形，可以看出我的眼睛被放大，嘴被压扁，这符合安德森图片中的面部模式。可见对于一般人脸图像，本方案仍能正常工作。

2.5 应用程序说明

图 13: 应用程序界面

程序界面非常简洁明快，通过左右按钮可以在 9 张测试样例中快速切换；加载图片、关键点可以加载外部的图片和对应的关键点，检测关键点用于一般人脸图像，保存关键点可以保存检测出的关键点；保存图片用于保存生成的目标图像，下方的选框用于调节变换方法和插值方法。

3 误差分析

由于空间变换方面的误差对最终结果的影响没有十分直接的关系，而 TPS、B 样条变换本身的变换方程过于复杂且不直接，以下涉及到空间变换的误差分析都是尽力而为，很遗憾无法给出明确的误差估计。

3.1 观测误差、舍入误差与模型误差

所谓观测误差是指测量值与真实值之间的误差，由于所处理对象为数字图像，数字图像是真实的图像经采样量化得到的。如果认为真实图像的像素值具有无限位有效数字，那么对于使用 `UInt8` 类型存储的数字图像，其观测误差为 0.5。

舍入误差为运算步骤中累积的存储误差，由于中间步骤中的 `double` 类型有 15 位有效数字，但在最后一步要转 `UInt8` 类型，中间舍入误差可以忽略，其舍入误差上界为 0.5。

模型误差是将对象数学化带来的误差，这涉及到图像使用数字进行表示的误差，这个误差无法分析，而本次大作业中直接处理数字图像，可认为不存在模型误差。

3.2 空间变换方法的方法误差

3.2.1 预对齐方法

预对齐方法是最小化各对关键点的空间位置之间的均方误差之和，由此操作导致的图像像素之间的误差无法估计；事实上包括 B 样条变换、TPS 变换，空间变换方法的空间误差对最后图像插值的像素值误差没有明确而直接的关系，这些误差的分析侧重于空间位置的误差。

求解预对齐仿射变换使用了最小二乘法，求解方程组使用的是高斯消元法。对形如

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.1}$$

的线性方程组，使用高斯消元法求解，其对相对误差误差的放大上界 [5] 为

$$\frac{||\delta\mathbf{x}||}{||\mathbf{x}||} \leq \mathrm{Cond}(\mathbf{A}) \left(\frac{||\delta\mathbf{A}||}{||\mathbf{A}||} + \frac{||\delta\mathbf{b}||}{||\mathbf{b}||} \right) \tag{3.2}$$

这个放大误差取决于 $\mathbf{A}$ 的条件数，即只要导向人脸图像的关键点 $\mathbf{X}$ 构成的矩阵 $\mathbf{X}^\top\mathbf{X}$ 的最大最小奇异值之比不太大，即各奇异值的量级相当，这一项基本能控制在舍入误差的量级上，不会成为主要误差。实际上求解的变换矩阵 $\mathbf{A}^*$ 再作用到变换点上，这一项误差实际为

$$\epsilon(\mathbf{A}^*\mathbf{X}) \leq ||\mathbf{A}^{*\top}||||\epsilon(\mathbf{X})|| \tag{3.3}$$

取决于 $\mathbf{A}^*$ 的范数，记为 ϵ_1。

另一个误差则是最小化的均方误差 (1.12)，则为 $\mathbf{A}^*$ 下的 L^*，记为 ϵ_2。二者的关系为，ϵ_1 为所求 $\mathbf{A}^*\mathbf{X}$ 与应求 $\mathbf{A}^*\mathbf{X}$ 的误差，ϵ_2 为 L2 范数意义下的应求 $\mathbf{A}^*\mathbf{X}$ 变换点与目标点的空间坐标误差之和。

3.2.2 B 样条变换方法

由于 B 样条变换中，控制点位移要先转化为网格点位移，再通过 B 样条基函数转化为图像上点的位移，这一过程十分复杂。而且由于逆变换中的取整导致无法求出 B 样条变换的逆变换方程，仅能近似地依据变换方式分析空间坐标变换误差。

B 样条变换的具体过程是将对应关键点位移按基函数赋给 16 邻域内网格点，再将网格点的位移按 B 样条基函数组合给目标图像中的点。反变换采用简单近似方法，即

$$(x', y') = (x, y) - H(x, y) \tag{3.4}$$

其中 H 为 B 样条正变换，定义为 (1.4)。由于取整以及网格点间距等因素无法一次移动到较优的距离，使用衰减迭代进行逼近，具体为

$$\Delta C_i(x, y) = \sum_{r=0}^{m} \gamma^r (P_{i-\mathrm{guide}}(x, y) - P_{i-\mathrm{src}}(x, y)) \tag{3.5}$$

实际中取最大迭代次数 $m = 5$，衰减率 $\gamma = 0.1$，达到了还算能看的结果。

接下来分析误差的传递，控制点的位移误差为

$$\epsilon(C) = \sum_{i=1}^{K} \sum_{r=0}^{m} \gamma^r \epsilon(P_{i-\mathrm{guide}}(x, y) - P_{i-\mathrm{src}}(x, y)) \tag{3.6}$$

其中 $\epsilon(P_{i-\mathrm{guide}}(x, y) - P_{i-\mathrm{src}}(x, y)) \leq 4 \times 0.5$，放大为 4 个 (x, y) 的舍入误差之和。由于网格点与任意点对应关系复杂，且无法闭式表达，故不继续分析。

3.2.3 TPS 变换方法

TPS 变换与预对齐类似，均以高斯消元法为核心，关注条件数；由于预对齐后，径向基函数的 r 都能处在合理的区间，所以方程 (1.11) 左边的矩阵条件数不大，实际情况也验证了这一点，所以舍入误差不会被放大，系数的误差都非常小。

其余的误差涉及到 TPS 基函数，但不能保证基函数的导数有界，之后的误差也就无法继续分析了。

3.3 图像插值方法误差

由于本方案处理的是 4 通道图像，但各通道相互独立而且误差分析一致，为了方便说明分析的原理和简化符号，接下来只分析一个通道的插值误差，结果可以类推至其他通道。

3.3.1 最近邻插值方法

首先考察 x 方向的误差，由于数字图像为离散值，使用差分代替导数，我们有

$$|R_x(x,y)| \leq \max|\Delta_x f(x,y)||\Delta x| \tag{3.7}$$

同理

$$|R_y(x,y)| \leq \max|\Delta_y f(x,y)||\Delta y| \tag{3.8}$$

由于数字图像的各阶差分必然有界，记 $M_{x1} = \max|\Delta_x f(x,y)|$，$M_{y1} = \max|\Delta_y f(x,y)|$，并代入 $|\Delta x| = |\Delta y| = \frac{1}{2}$，得到最终结果

$$|R(x,y)| \leq \frac{M_{x1} + M_{y1}}{2} \tag{3.9}$$

3.3.2 双线性插值方法

将双线性表达式 (1.18) 展开，可以认为先在 x 方向插值，再在 y 方向插值，我们有

$$f(x',y') = vf(x',j) + (1-v)f(x',j+1) \tag{3.10}$$

其中

$$f(x',j) = u(i,j) + (1-u)f(i+1,j) \tag{3.11}$$

$$f(x',j+14) = uf(i,j+1) + (1-u)f(i+1,j+1) \tag{3.12}$$

分别求余项误差估计，如对 (3.11) 分析，则

$$|R_x(x,j)| \leq \frac{1}{2!}\max|\Delta_x^2 f(x,y)| \max_{0\leq u\leq 1}|u(1-u)| = \frac{1}{8}\max|\Delta_x^2 f(x,y)| \tag{3.13}$$

同理

$$|R_x(x,j+1)| \leq \frac{1}{2!}\max|\Delta_x^2 f(x,y)| \max_{0\leq u\leq 1}|u(1-u)| = \frac{1}{8}\max|\Delta_x^2 f(x,y)| \tag{3.14}$$

代入 (3.10)，我们有

$$|R_x(x,y)| \leq \frac{1}{8}v\max|\Delta_x^2 f(x,y)| + \frac{1}{8}(1-v)\max|\Delta_x^2 f(x,y)| = \frac{1}{8}\max|\Delta_x^2 f(x,y)| \tag{3.15}$$

同理

$$|R_y(x,y)| \leq \frac{1}{8}\max|\Delta_y^2 f(x,y)| \tag{3.16}$$

记 $M_{x2}=\max|\Delta_x^2 f(x,y)|$，$M_{y2}=\max|\Delta_y^2 f(x,y)|$，得到最终结果

$$|R(x,y)|\leq\frac{M_{x2}+M_{y2}}{8} \tag{3.17}$$

3.3.3 双三次插值方法

与双线性插值方法类似，分析两个方向，我们有

$$\begin{aligned}|R_x(x,j-1+l)|&\leq\frac{1}{4!}\max|\Delta_x^4 f(x,y)|\max_{0\leq u\leq 1}|(u+1)u(u-1)(u-2)|\\&\leq\frac{3}{128}\max|\Delta_x^4 f(x,y)|\end{aligned} \tag{3.18}$$

$$|R_x(x,y)|\leq\frac{3}{128}\max|\Delta_x^4 f(x,y)| \tag{3.19}$$

于是

$$|R_y(x,y)|\leq\frac{3}{128}\max|\Delta_y^4 f(x,y)| \tag{3.20}$$

记 $M_{x4}=\max|\Delta_x^4 f(x,y)|$，$M_{y4}=\max|\Delta_y^4 f(x,y)|$，得到最终结果

$$|R(x,y)|\leq\frac{3(M_{x4}+M_{y4})}{128} \tag{3.21}$$

4 总结与讨论

4.1 B 样条变换与 TPS 变换

TPS 变换是一种全局变换，它将整个图片所在平面弯曲，再将此投影在水平平面上，类似将整张纸弯曲后再拍到桌子上。TPS 可以将整个图片的所有区域扭曲，以实现最大程度的人脸关键点匹配。然而 B 样条受限于网格点以及 B 样条基函数，这使得它是一种局部变换，B 样条的效果类似将图片放在一层橡皮膜上，根据网格点的移动对膜上的每个网格作向上弯曲（膨胀）或者向下弯曲（收缩），再将这层膜拍到桌子上。B 样条只能将人脸区域内的部分网格作扭曲，产生的形变在整体效果上不如 TPS。

关于如何提升 B 样条的表现，我与我的室友们（林嘉成，2016011498；岑哲鹏，2016011516）进行了讨论；他们提出了使用梯度下降法实现最小化各个关键点位置差的 MSE（L2）或者 MAE（L1），并取得了不错的效果。但我认为这样做缺乏理论依据，究竟如何衡量一个变形的好坏，或者说如何通过关键点衡量两组人脸的形状相似程度才是最适合人脸变形问题，仍然值得考虑。所以我回到了问题的出发点，将关键点位移直接按权重赋值给网格点位移会导致位移不足，导致人脸形变效果很差。我采用了多次位移的方法补偿这些不足的位移，也在一定程度上解决了这个问题。

4.2 预对齐的必要性

我在与其他同学交流时，他们指出应当先做 TPS 变换，再在其基础上做 B 样条变换。我发现实际上第一步的 TPS 变换与预对齐起到了类似的作用，但是如果用 TPS 代替一个预对齐的话，会出现测试样例部分讨论的问题——人脸平移，姿态及尺度不一致，这个时候再在更小尺度上应用 B 样条变换会遇到更大的困难——虽然 TPS 变换已经足够完成大部分的变形任务。

事实上，分析 TPS 变换的函数表达式，可以看出它实际上是仿射变换（线性项）和弯曲变换（非线性基函数项）的组合，如果没有初步的仿射变换，那么会出现 TPS 的将源图叠加一个与导向图相关的仿射变换，以期待将图像的大小、旋转、平移等属性也变换成导向图，而不仅仅是人脸关键点所描述的脸部模式。所以，预对齐的意义在于让 TPS 变换专注于人脸的变形，即人脸模式的变换，而不是图像其他属性的变换。

另一点值得注意的是，如果不做预对齐，B 样条变换将无法正常工作，这也说明了将两组关键点一致化的必要性。

4.3 总结与反思

本次大作业实际上是我第一次使用非数据科学语言且不借助 OpenCV 的算法编写图像处理程序，从底层数据处理到顶层的界面都是一次宝贵的机会。通过这次大作业，我对图像，特别是人脸的空间形状变换有了更深刻的认识，也对之前在实验室项目中处理人脸数据集中最常用的方法——仿射变换有了进一步的理解。关于插值，除了课上的 1 维插值，我现在也手动推导了 2 维插值的误差分析，收获很大。除此之外，手动实现了方程求解也是让我的编程能力有所提高。

本次大作业仍然留下了一些问题，比如如何分析空间变换的误差对最终图片误差的影响，如何衡量人脸变形效果的好坏，如何评估人脸的相似程度。在尝试解决这些问题的同时，我也加深了对人脸关键点的意义的理解，对人脸图像的本质有了新的思考。但是这些问题仍然有待我进一步思考，或许今后我能一一解决。

最后感谢助教批准我的两周延期申请，能让我有机会在繁忙的科研任务与举办电子设计大赛之余，认真打磨程序，撰写报告，完成一个让自己满意的大作业。

参考文献

[1] Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

[2] Gianluca Donato and Serge J Belongie. *Approximation methods for thin plate spline mappings and principal warps*. Citeseer, 2003.

[3] 蝈蝈俊. affine transformation matrix 仿射变换矩阵与 opengl. https://www.cnblogs.com/ghj1976/p/5199086.html. 2016-02-18.

[4] takuya takeuchi. Face landmark detection. https://github.com/takuya-takeuchi/DlibDotNet/tree/master/examples/FaceLandmarkDetection. 2018-07-15.

[5] 李庆扬. 数值分析. 清华大学出版社有限公司, 2001.