

作业 2 图像变换与点列的曲线拟合

晏瑞然 87 PB19000196

2021 年 4 月 6 日

摘要

图像变换与曲线拟合一直是数学建模与计算机图形学中的两个重要领域本文对图像变换与平面的点列曲线拟合进行了深入研究。对图像变换，本文实现了 IDW 及 RBF 方法，对比展示了两种算法的效果；而对于曲线拟合，本文比较了不同维度多项式及不同参数化的方法来实现的拟合效果，并对其进行了说明。

关键词：图像变换；曲线拟合；IDW 方法；RBF 方法；多项式拟合；参数化

一、前言

1.1 问题概述

图像变换：图像变换一直是图像学研究中的经典议题，即给定某种要求或限制，将一张图修改为另一张满足要求的图，如将蒙娜丽莎的微笑改为蒙娜丽莎的“愤怒”，在现代的许多图像处理的软件中该技术已得到了广泛应用。

曲线拟合：对于任意形状的曲线，我们可以用一组多项式来对该种曲线进行拟合，这种方式能让我们得到某种未知图形的具体表达式，在如今该技术有着广泛应用，工业中我们可以得到设备外壳的轮廓，军事中我们可以用它得到机翼等重要军事组件的模型，十分重要。

本文通过具体方法来对以上两种算法进行了复现，并分析他们的效果进行对比。

1.2 具体问题描述

图像变换：给定名画蒙娜丽莎的微笑，通过图像变换改变蒙娜丽莎的表情。

曲线拟合：通过鼠标任意选取一组点，对其进行不同维度及不同参数化方法的曲线拟合，并将拟合结果可视化。

二、相关工作

见文献^[1]，文献^[2]，文献^[3]。

三、问题分析

3.1 图像变换

所谓图像变换就是一个二维空间到二维空间的映射，我们只需要找到一个映射 f ，使得其满足控制点之间的关系，并让图像其他像素点平滑的改变即可。该问题可以简化成如下问题：

输入：给定 n 对控制点 (p_i, q_i) ； $p_i, q_i \in R^2$ ， p_i 为控制起点， q_i 为控制终点； R^2 表示图像所有点的集合，一般为一个二维矩阵；

输出：一个至少连续的控制函数 $f: R^2 \rightarrow R^2$ ，满足 $f(p_i) = q_i$ ；首先的想法便是读取图像上每一点，分别对每一个点进行与控制点相同的变换，但这样做最后只能得到一个平移的结果，所以还要求出每对控制点函数变换的权重。即根据某种算法可以求出图像上某种控制函数的权重 ω_i ，再用 ω_i 与控制函数作点积即可得到变换后的坐标。通过后面内容可知，若该控制函数是变换点到控制起点的距离，该算法即为 IDW(Inverse distance-weighted interpolation) 算法；若控制函数是控制点之间的距离的函数，该算法即为 RBF (Radial basis functions) 算法。

3.2 曲线拟合

对于曲线拟合现在普遍采用插值法，例如给定函数 $f(x)$ 在区间 $[a, b]$ 中互异的 n 个点的值 $f(x_i), i = 1, 2, \dots, n$ 基于这个列表数据，寻找某一个函数 $p(x)$ 去逼近 $f(x)$ 。若要求 $p(x)$ 在 x_i 处与 $f(x_i)$ 相等，就称这样的逼近问题为插值问题。而当型值点太多时，构造插值函数使其通过所有的型值点相当困难的。客观上看，由于过多的型值点也会有误差，也没有必要寻找一个插值函数通过所有的型值点。往往选择一个次数较低的函数，在某种意义上最佳逼近这些型值点即可。可以使用最小二乘法进行逼近。

但是值得注意的是，同一组数据点，即使采用同样的插值法，若数据点的参数化不同，将可能获得不同的插值曲线。人们希望：对数据点的参数化，应尽可能反映被插 (逼) 曲线或设计员欲用数据点所构造的曲线的性质。对数据点实行参数化有 4 种方法^[3]，分别为均匀参数化法、弦长参数化法、中心参数化法、Foley-Nielsen 参数化法。

四、符号说明

表 1: 符号说明

符号	说明
m	均匀参数化
d	弦长参数化
c	中心参数化
f	Foley-Nielsen 参数化
p_i	第 i 个已知点
$ \Delta p_i $	p_i 与 p_{i+1} 两点的欧式距离
u_i	第 i 个点对应的参数
K_i	Forly 参数化中对应的修正系数
ω	回归得到的多项式系数
(p_i, q_i)	控制点元组, p_i 为控制起点, q_i 为控制终点
P_i	控制起点的集合
R^2	图像矩阵中点坐标的集合
f_i	控制函数
ω_i	第 i 个控制函数的权重

五、数学模型

5.1 图像变换

1.IDW 模型:

找到函数 $h(p) = \sum_{i=1}^n \omega_i(p) f_i(p) + p$ 其中 $p \in R^2$, 且 $f_i(p)$ 满足 $f_i(p_i) = q_i$; ω_i 满足条件 $\omega_i(p_i) = 1, \sum_{i=1}^n \omega_i(x) (x \in R^2 \setminus P_i)$ 我们可以使用以下简单权值函数:

$$\omega_i(x) = \frac{\sigma_i(x)}{\sum_{j=1}^n \sigma_j(x)} (x \in R^2 \setminus P_i)$$

其中 $\sigma_j(x) = \frac{1}{d(x, p_j)^u}$, $d(x, p_j)$ 为 x 与 p_j 的距离, u 为大于 0 的任意常数。

2.RBF 模型:

通过如下函数:

$$h(p) = \sum_{i=1}^n (d(x, p_i)^2 + r_i^2)^{u/2} * \omega_i(p) + p$$

得到 $p \in R^2$ 的转换。其中 ω_i 为一个二维向量, 分别代表第 i 个控制点的 x 方向与 y 方向的偏移权重; u 为一个大于 0 常数; r_i 满足条件: $r_i = \min_{i \neq j} d(p_i, p_j), p_i \text{ and } p_j \in P_i$

$\omega_j^i, i = 1, 2 \dots n, j = x, y$ 可以通过求解以下方程组得到:

$$\Phi \Omega_j = D_j, j = x, y$$

其中 $\Phi[i][j] = (d(p_i, p_j)^2 + r_j^2)^{u/2}; i, j = 1, 2 \dots n$; Ω_j 为 $\omega_j^i, i = 1, 2 \dots n, j = x, y$ 组成的列向量, D_j 为控制点 x 或 y 方向的偏移量组成的列向量。

5.2 曲线拟合

采用最常用的最小二乘法进行逼近: 假设给定一组型值点 $Q_i(x_i, y_i) i = 1, 2 \dots n$, 要求构造一个 $m(m < n - 1)$ 次多项式函数 $f(x)$ 逼近这些型值点, 衡量逼近程度的代价函数是取各点偏差的平方和, 令该值最小:

$$\min E = \sum_{i=1}^N \|y_i - f(x_i)\|^2$$

而因为 $f(x)$ 为 m 次多项式函数, 我们可以算出每个 $x_i^j (i = 1, 2 \dots n; j = 1, 2 \dots m)$, 就可以把它转换为多元的线性回归问题, 可以很容易的得到最小二乘的解。

下面来介绍 4 种不同的参数化方法:

1. 均匀参数化法
2. 弦长参数化法
3. 中心参数化法
4. Foley-Nielsen 参数化法

1) 均匀参数化法

使每个节点区间长度 (用向前差分表示), $\Delta i = u_{i+1} - u_i = \text{正常数} (i = 0, 1, \dots, n)$, 即各参数点等距分布, 为处理方便起见, 常取成整数序列:

$$u_i = i, i = 0, 1, \dots, n$$

这种参数化法仅适合于数据点多边形各边 (或称弦) 相等的场合。否则, 在多边形相邻段弦长相差悬殊的情况下, 生成插值曲线后较长的那段曲线显得较扁平, 弦长较短的那段曲线则膨得厉害, 甚至出现尖点或打圈自交的情况。

出现上述问题, 从物理角度可解释如下: 把参数 u 看做时间, 质点 P 随着时间变化在空间扫出一条依次经过给定位置 (即数据点) 的曲线 $p(u)$, 采用均匀参数化就意味着在任意顺序两邻点间花费同样多的时间, 而不管它们间的距离如何。如果汽车沿着这样一条插值曲线行驶时, 数据点成了站点, 当两相邻站点间距离大时, 就必须高速前进。如果接着两相邻站点间距离很小时, 由于不能把速度突然减下来, 就会发生过冲。

2) 弦长参数化法

$$\begin{cases} u_0 = 0 \\ u_i = u_{i-1} + |\Delta p_{i-1}|, i = 1, 2, \dots, n \end{cases} \quad (1)$$

这种参数化法如实反映了数据点按弦长的分布情况, 一直被认为是最佳参数化法。它克服了数据点按弦长分布不均匀时采用均匀参数化所出现的问题。在多数情况下能获得较满意的结果。

当数据点取得足够密。且插值法具有收敛性质（即加密数据点使插值曲线收敛到被插曲线的性质）时，将生成近似弧长参数化的插值曲线。但在工程实践中，都不希望这样的麻烦。人们希望能用尽可能少但又足以表示形状的数据点，方便地生成所要求的曲线。

3) 中心参数化法

$$\begin{cases} u_0 = 0 \\ u_i = u_{i-1} + |\Delta p_{i-1}|^{\frac{1}{2}}, i = 1, 2, \dots, n \end{cases} \quad (2)$$

这是美国波音公司的李 (Lee, 1989) 提出的。他看到积累弦长参数化法并不总能保证生成光顺的插值曲线、认为问题在于未考虑数据点相邻弦线的折拐情况。这种参数化方法对一些不规则的曲率变化较大的曲线曲面具有较好的逼近效果。

4) Foley-Nielsen 参数化法

$$\begin{cases} u_0 = 0 \\ u_i = u_{i-1} + K_i |\Delta p_{i-1}|, i = 1, 2, \dots, n \\ K_i = 1 + \frac{3}{2} \left(\frac{|\Delta p_{i-2}| \theta_{i-1}}{|\Delta p_{i-2}| + |\Delta p_{i-1}|} + \frac{|\Delta p_i| \theta_i}{|\Delta p_{i-1}| + |\Delta p_i|} \right) \\ \theta_i = \min(\pi - \angle p_{i-1} p_i p_{i+1}, \frac{\pi}{2}) \\ |\Delta p_n| = |\Delta p_{-1}| = 0 \\ \theta_n = \theta_0 = 0 \end{cases} \quad (3)$$

其中，弦长修正系数 $K_i > 1$ 。从公式可知，与前后邻弦长 $|\Delta p_{i-2}|$ 及 $|\Delta p_i|$ 相比，若 $|\Delta p_{i-1}|$ 越小，且与前后邻弦边夹角的外角 θ_{i-1} 和 θ_i (不超过 $\frac{\pi}{2}$ 时) 越大，则修正系数 K_i 就越大。因而 $K_i |\Delta p_{i-1}|$ 也就越大，这样就对因该曲线段绝对曲率偏大，与实际弧长相比，实际弦长偏短的情况起到了修正作用。修正弦长就较接近于实际弧长。

六、算法实现

6.1 IDW 算法

- 1) 根据给出的 n 对控制点，可以求出 n 个控制起点到控制终点的线性函数组，该函数即为上面公式中 $f_i(p)$ 组成的函数组。
- 2) 根据模型中的数学公式得到每点的权重。
- 3) 将控制函数与权值点积后对每个点进行映射，得到输出，注意该过程中会出现小数而真正计算机中图片的矩阵存储坐标都是整数，所以还要进行取整操作。
- 4) 将输出的点赋予映射前的点的颜色即可得到图片的变换。
- 5) 色彩填充，原因见结果说明。

6.2 RBF 算法

- 1) 计算权重方程组的系数矩阵 Φ 及控制点 x, y 方向的偏移列向量 D_x, D_y 。
- 2) 求解线性方程 $\Phi \Omega_j = D_j, j = x, y$ 。
- 3) 通过公式 $h(p) = \sum_{i=1}^n (d(x, p_i)^2 + r_i^2)^{u/2} * \omega_i(p) + p$ 输出变换后的坐标。
- 4) 将输出的点赋予映射前的点的颜色即可得到图片的变换。
- 5) 色彩填充。

6.3 曲线拟合

- 1) 通过鼠标响应在窗口中选 n 个点，每选一点将该点的横纵坐标值加入列向量 X, Y 。
- 2) 通过不同参数化法得到每个点对应的参数。
- 3) 选定回归的维度，设多项式回归中最大次数 $m-1$ ，以 u_i^j 为第 i 行第 j 列元素组成矩阵 $U_{n \times m}$ 。
- 4) 通过最小二乘法得到需求系数矩阵 $\omega_X = (U^T U)^{-1} U^T X, \omega_Y = (U^T U)^{-1} U^T Y$ ，并得到多项式 $P_X(u), P_Y(u)$ 。
- 5) 画出由不同参数化方法及所设定的不同多项式次数得到的 $x = P_X(u), y = P_Y(u)$ 代表的参数方程的图像。

七、结果

7.1 结果说明

曲线拟合：

首先我们通过不同次数的多项式进行了点列拟合，如图 1 所示，这些图皆由弦长参数化法得到。

对于某些特定点列来说，比如大致曲线只有 2 个拐点，那么拟合效果都还比较不错 (图 1.a)。但当点列复杂起来 (图 1.b, 图 1.c)，低次曲线的拟合效果就不尽如人意了，也就是会造成欠拟合的效果；同样若次数过高就会造成过拟合现象，如图 1.b 所示，代表 9 次多项式回归的曲线，很明显可以看出该图中曲线在许多地方过分扭曲，显然不是我们想要看到的，6 次曲线就表现的很好，能够很好的拟合这组点列。

通过不同维度的多项式拟合，我们给每条曲线赋予一个颜色，若点列震荡十分剧烈，就可以得到很好看的曲线涂鸦，如图 2 所示，当然还可以加入更高维的多项式曲线，图片色彩缤纷，这一条条曲线无不展示着数学之美！

接下来展示的是通过 6 次多项式拟合的不同参数化法得到的拟合曲线，其中，红、紫、绿、黄线分别为均值参数化法、弦长参数化法、中心参数化法、Forly-Neilsen 参数化得到的曲线，对比图如图 3 所示。

最后我们用 10 次多项式进行了不同参数化法的对比，见图 4。可以看出，均值参数化及弦长参数化都会造成剧烈的震荡，中心参数化比上面两种稍微好一点，而 Forly 参数化法能很不错的减少过拟合带来的影响。所以对于产生过拟合后的效果，不同参数化法对结果的影响就很大了，表现较差的参数化法会带来严重的过拟合，表现好的能很好的减少过拟合带来的影响。

图像变换：

首先通过棋盘格进行测试，如图 5、图 6 所示，其中原图中的红点为控制起点，绿点为控制终点，可以看出对于相同的控制点组，运用不同的算法及不同的参数效果皆不同，但不管哪种算法 u 取值越大，收缩及旋转的效果越显著。

再进行蒙娜丽莎图片结果展示，见图 7。可以看到得到的图中有很多的白点，这是因为有上述方法得到的映射并不是一个满射，会有点映射到别处而没有点映射到该处造成空缺，所以需要填色，而填色的方法便是找到该点周围的点将此点填入。填色后效果如图 8。当然，通过这两种算法还能使蒙娜丽莎大笑(图 9)。

7.2 结果展示

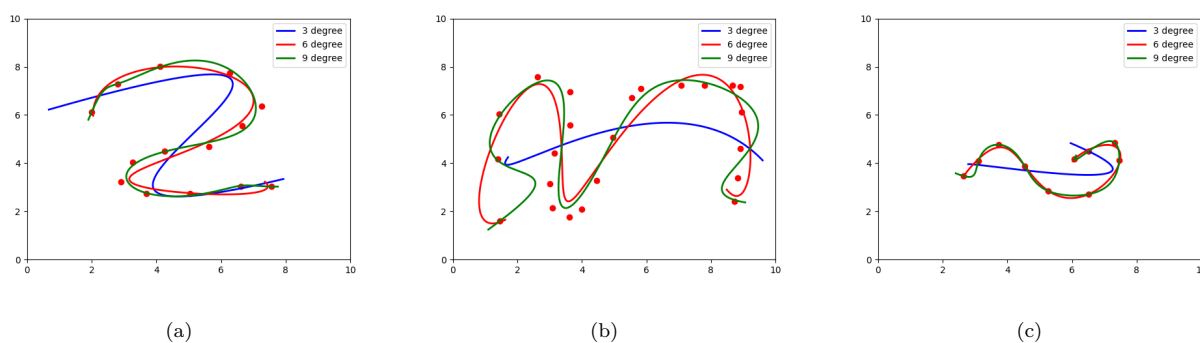


图 1: 不同点列的 3、6、9 次多项式拟合

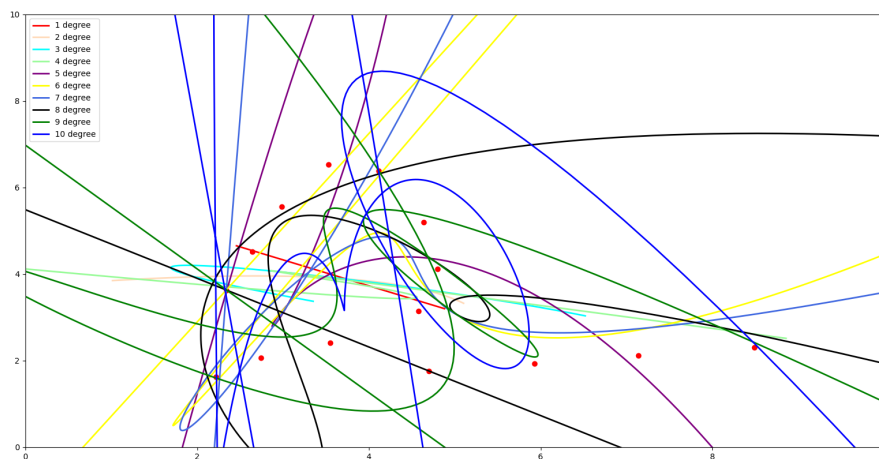


图 2: 通过不同维度的多项式拟合得到的涂鸦

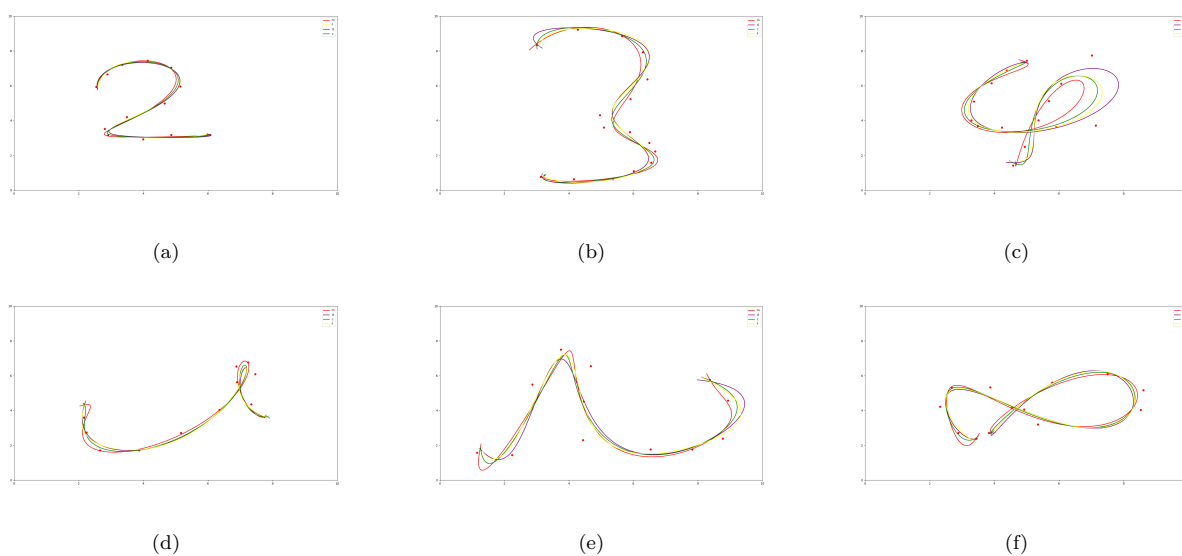


图 3: 不同参数化法得到的拟合曲线

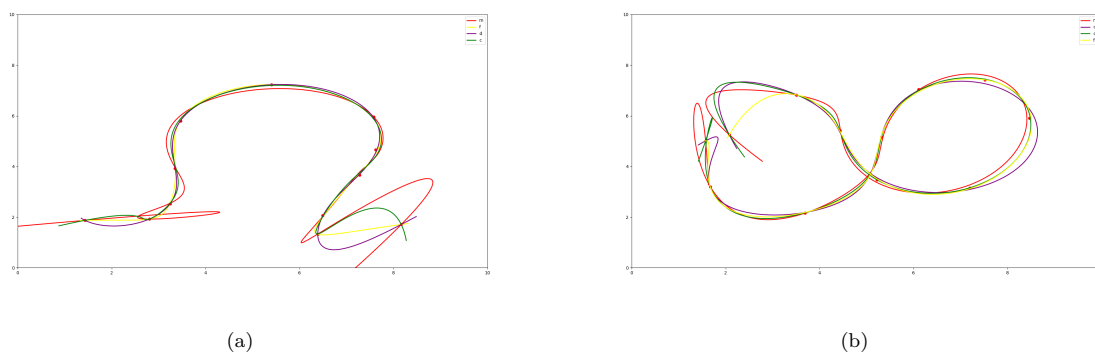


图 4: 不同点列的 3、6、9 次多项式拟合

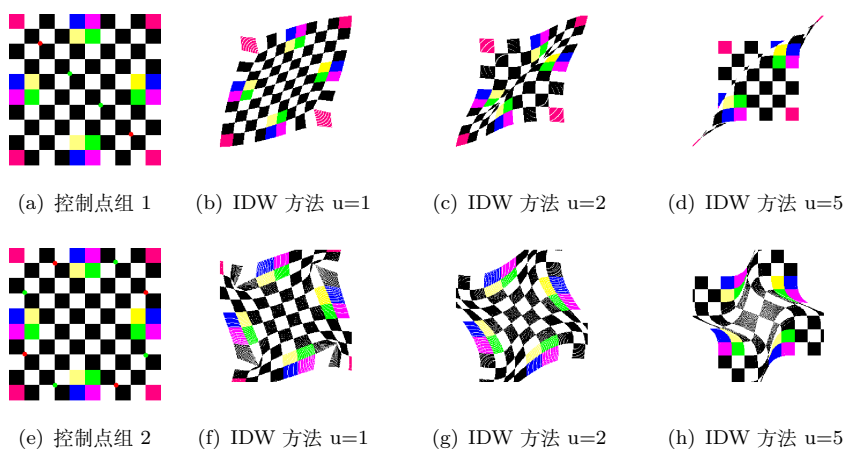
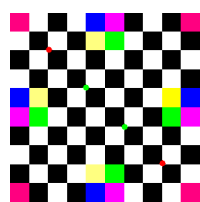


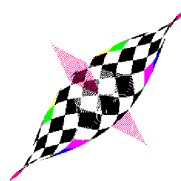
图 5: 不同控制点及不同 u 取值的 IDW 方法比较



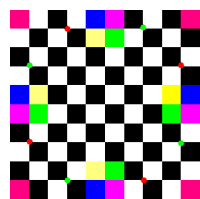
(a) 控制点组 1



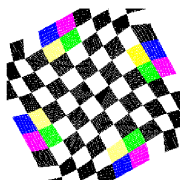
(b) RBF 方法 $u=1$



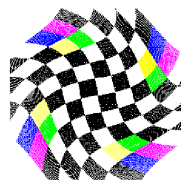
(c) RBF 方法 $u=5$



(d) 控制点组 2

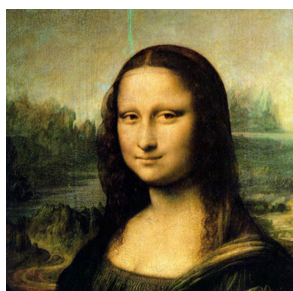


(e) RBF 方法 $u=1$

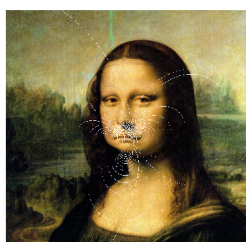


(f) RBF 方法 $u=5$

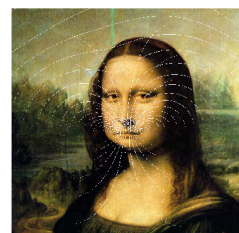
图 6: 不同控制点及不同 u 取值的 RBF 方法比较



(a) 原图

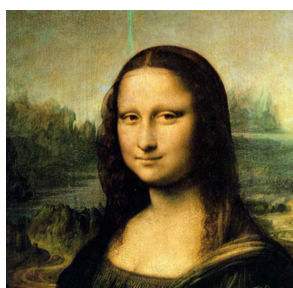


(b) IDW 方法 ($u=2$)

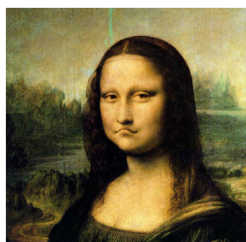


(c) RBF 方法 ($u=1$)

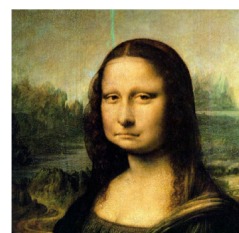
图 7: “蒙娜丽莎的苦脸”



(a) 原图

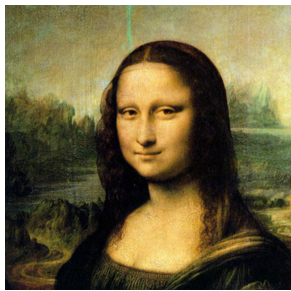


(b) IDW 方法 ($u=2$)

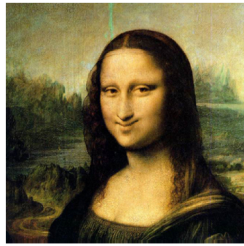


(c) RBF 方法 ($u=1$)

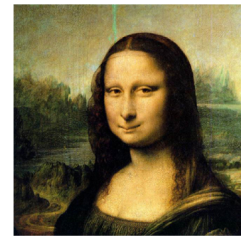
图 8: “蒙娜丽莎的苦脸”



(a) 原图



(b) IDW 方法 ($u=2$)



(c) RBF 方法 ($u=1$)

图 9: “蒙娜丽莎的大笑”

八、问题与优化

- 1) IDW 算法与 RBF 算法中得到的映射会映射成一对小数坐标，而计算机的图片存储的坐标都是整数，所以计算过程中要进行取整操作。
- 2) 填色过程中可以取该点周围 8 个点的平均值这样能使图片更加平滑。
- 3) 进行 Foley-Nielsen 参数化时容易出现边界值越界的问题，所以要在程序中做好边界值的处理，对 $i = 0, i = n$ 两处进行特别处理。
- 4) 高次多项式回归会造成过拟合，可以通过正则化的方式来对高次系数进行惩罚，得到更好的拟合效果。

九、代码及程序说明

具体代码见压缩包中的三个 py 文件。

其中 degreeCompare.py 文件为比较不同维度多项式回归结果的程序代码，默认使用弦长参数化法，可在程序中自主调节。选点后，按 1~9 中任意数即可得到该数维度的多项式回归的结果，按其他按键会按十次多项式回归得到曲线。

parameterizationCompare.py 为比较不同参数化方法的程序代码，可改变全局变量 exp 设置回归多项式次数，默认为十次。鼠标选点后按'm','d','c','f' 分别代表进行均匀参数化、弦长参数化、中心参数化、福利参数化，默认使用福利参数化。

imageWrapping.py 为图形变换代码，运行后鼠标按下点为控制起点，松开点为控制终点，控制点选好后按'i' 为使用 IDW 方法，其他键为使用 RBF 方法。可以改变全局变量改变 u 的值进行测试，也可以改变 imageName 全局变量用不同的图进行测试

参考文献

- [1] D. Ruprecht and H. Muller, “Image warping with scattered data interpolation,” *IEEE Computer Graphics and Applications*, vol. 15, no. 2, pp. 37–43, 1995.
- [2] N. Arad and D. Reisfeld, “Image warping using few anchor points and radial functions,” *Computer graphics forum*, vol. 14, no. 1, pp. 35–46, 1995.
- [3] 施法中, “计算机辅助几何设计与非均匀有理 b 样条,” 北京: 高等教育出版社, pp. 223–240 and 334–348, 2001.