

# 数值分析第一次大作业

## 项目报告

姓名： 陈华玉

学号： 2017011518

## 目录

1	需求分析与方案设计.....	1
2	插值方法原理.....	1
2.1	最临近插值.....	1
2.2	双线性插值.....	1
2.3	双三次插值.....	2
3	图像变形.....	2
3.1	旋转畸变.....	2
3.2	球形畸变.....	3
4	TPS网格变换.....	3
5	误差分析.....	4
5.1	颜色插值误差分析.....	4
5.1.1	最临近插值.....	5
5.1.2	双线性插值.....	5
5.1.3	双三次插值.....	5
5.1.4	比较.....	6
5.2	图像变形产生的误差.....	6
5.3	颜色整型存储表示产生的舍入误差.....	6
6	总结.....	7
7	参考文献.....	7

# 1 需求分析与方案设计

本项目主要功能有两个，一是人脸图像进行扭曲变形，二是对输入图像做旋转扭曲或者球形畸变变换。为了实现这些功能，我们需要完成的核心模块也是两部分，一是点到点扭曲变形的映射函数（将在3,4部分分别介绍）、二则是完成图像插值的插值算法（在2部分中介绍）。此外在5中也将会给出本项目的误差分析。

## 2 插值方法原理

插值的在本项目中目的是，给定坐标（i，j），当i，j均为浮点数时，通过插值方法与图片像素值，估计出其对应的颜色像素f（i，j）。本项目实现三种二维插值方式，分别为最临近、双线性与双三次插值，分别对应一到三次的多项式拟合。

### 2.1 最临近插值

最临近插值想法非常简单，就是认为（i，j）的颜色像素应当和与其最近的坐标点颜色相同，因而近似处理为相同颜色。不再赘述。

### 2.2 双线性插值

双线性插值是在一位线性插值函数基础上的拓展，基本思想为在左右两个有值临近点的像素值之间线性连线，并类似加权平均地取出像素值。其具体计算公式如图1。

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

Figure 1 双线性插值公式

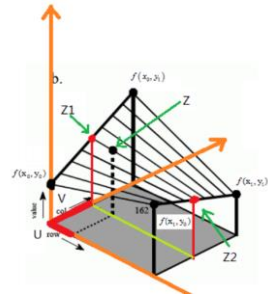


Figure 2 插值示意图

## 2.3 双三次插值

双三次插值与双线性插值类似，但是这里我们要求的平滑不再仅仅限于“一阶平滑”而是要求“一阶导数平滑”。为了达到这一目的，我们以给定插值点为中心求取 $4 \times 4 = 16$ 个点为参考点而不是仅仅4个点。然后在横纵坐标上分别拟合三次插值函数，并求出插值点像素。

但是由于严格的三次插值需要进行非线性运算，为了计算的效率，[1]提出可以将三次插值公式简化为线性的近似公式。

令：

$$u = x' - \lfloor x' \rfloor, \quad v = y' - \lfloor y' \rfloor, \quad i = \lfloor x' \rfloor, \quad j = \lfloor y' \rfloor$$

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

其中  $S(t)$  就是我们的近似函数，定义为：

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

这样便可以完成我们的双三次插值映射。

## 3 图像变形

### 3.1 旋转畸变

旋转畸变想法很简单，以图像原点为中心，像素旋转一定角度  $\alpha$ ，且离中心越近旋转角度越大，线性递减。

因为公式可以表示为：

$$\begin{aligned} x &= x' \cdot \cos \alpha - y' \cdot \sin \alpha \\ y &= x' \cdot \sin \alpha + y' \cdot \cos \alpha \end{aligned}$$

其中  $\alpha$  可以表示为：

$$\alpha = \alpha_0 \left( 1 - \frac{r}{r_0} \right)$$

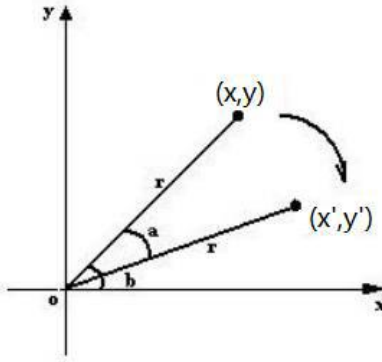


Figure 3 旋转畸变示意图

### 3.2 球形畸变

球形畸变模拟相机的畸变模式，数学原理是二维图像投影到球面上后展开的效果。

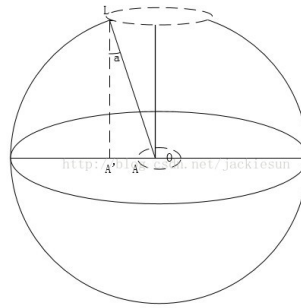


Figure 4 畸变原理示意图

其计算公式为：

$$x = x' \frac{r_0}{r} \arcsin\left(\frac{r}{r_0}\right)$$

$$y = y' \frac{r_0}{r} \arcsin\left(\frac{r}{r_0}\right)$$

其中 $r_0$ 为可以调整的参数，代表畸变球的半径。

## 4 TPS网格变换

TPS网格变形（薄板样条插值）的最基本思想就是求取任意数量二维点到二维点映射的插值函数。我们可以把这样一个求取最优插值函数的任务描述为求取最优插值函数使得损失最小。损失函数保证了插值函数的平滑性，而损失函数 $I_f$ 定义为：

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

为了求解这一插值函数，我们首先定义基函数：

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

我们定义对于输入坐标  $(x, y)$ ，通过映射函数  $f$ ，将其映射为目标点  $f(x, y)$ 。  
具体来讲，有：

$$f(x, y) = [f_x(x, y), f_y(x, y)]^T = \mathbf{a}_1 + \mathbf{a}_x x + \mathbf{a}_y y + \sum_{i=1}^n \mathbf{w}_i U(\|P_i - (x, y)\|)$$

其中  $\mathbf{a}_x$ ,  $\mathbf{a}_y$ ,  $\mathbf{a}_1$ ,  $\mathbf{w}$  为待求参数，有目标点与原点坐标对联立求出。

给定  $n$  个控制点  $P_1 = (x_1, y_1), \dots, P_n = (x_n, y_n)$ ，记：

$$K = \begin{bmatrix} 0 & U(r_{12}) & \cdots & U(r_{1n}) \\ U(r_{21}) & 0 & \cdots & U(r_{2n}) \\ \cdots & \cdots & \cdots & \cdots \\ U(r_{n1}) & U(r_{n2}) & \cdots & 0 \end{bmatrix} \quad P = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \cdots & \cdots & \cdots \\ 1 & x_n & y_n \end{bmatrix} \quad L = \begin{bmatrix} K & P \\ P^T & \mathbf{0} \end{bmatrix}$$

同时给定  $n$  个目标点  $\tilde{P}_1 = (x'_1, y'_1), \dots, \tilde{P}_n = (x'_n, y'_n)$ ，记：

$$V = \begin{bmatrix} x'_1 & x'_2 & \cdots & x'_n \\ y'_1 & y'_2 & \cdots & y'_n \end{bmatrix}, \quad Y = \begin{pmatrix} V & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{pmatrix}^T$$

则带求取参数则为方程组  $L[\mathbf{w}_1, \dots, \mathbf{w}_n, \mathbf{a}_1, \mathbf{a}_x, \mathbf{a}_y]^T = Y$  的解。

## 5 误差分析

我们将变形产生的误差分为三类，分别为：

- 由差值函数估计插值点颜色产生的误差
- 由图像变形时计算变形后像素坐标产生的误差
- 像素由整型数据表示产生的舍入误差

下面我们逐一对着三类误差进行分析。

### 5.1 颜色插值误差分析

### 5.1.1 最临近插值

我们设原图片的像素值颜色映射函数为 $I(x, y)$ ，则由于最临近插值采用直接四舍五入的方法，我们有 $|\Delta_x| \leq 1$ ，且 $|\Delta_y| \leq 0.5$ 。

$$\text{因而我们有 } |\Delta I(x, y)| \leq \frac{1}{2} [\max \left( \left| \frac{\partial I}{\partial x} \right| \right) + \max \left( \left| \frac{\partial I}{\partial y} \right| \right)]$$

### 5.1.2 双线性插值

为了求解二维情况下的线性插值误差，我们可以先考虑一维的情况然后考虑维度叠加。这里我们考虑使用插值余项公式，这条公式的使用前提是我们假定原映射函数是一阶连续的，这显然是符合自然规律与人类认知的，因而可以被用作基本假设。

设  $u = x - [x]$ ,  $v = y - [y]$ ，由插值余项公式，则有

$$|\Delta I(x, y)|_x \leq \frac{1}{2} * \max \left( \left| \frac{\partial^2 I}{\partial x^2} \right| \right) * |(u-1)u| \leq \frac{1}{8} * \max \left( \left| \frac{\partial^2 I}{\partial x^2} \right| \right)$$

$$|\Delta I(x, y)|_y \leq \frac{1}{2} * \max \left( \left| \frac{\partial^2 I}{\partial y^2} \right| \right) * |(v-1)v| \leq \frac{1}{8} * \max \left( \left| \frac{\partial^2 I}{\partial y^2} \right| \right)$$

两个维度叠加则有：

$$|\Delta I(x, y)| \leq \frac{1}{8} * [\max \left( \left| \frac{\partial^2 I}{\partial x^2} \right| \right) + \max \left( \left| \frac{\partial^2 I}{\partial y^2} \right| \right)]$$

### 5.1.3 双三次插值

与双线性插值类似，我们先对一位的情况进行分析：

由教材46页推导公式，我们有在一维条件下

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

因而得到：

$$|\Delta I(x,y)|_x \leq \frac{3}{128} * \max \left( \left| \frac{\partial^4 I}{\partial x^4} \right| \right)$$

$$|\Delta I(x,y)|_y \leq \frac{3}{128} * \max \left( \left| \frac{\partial^4 I}{\partial y^4} \right| \right)$$

因为维度合成我们有：

$$|\Delta I(x,y)| \leq \frac{3}{128} * [\max \left( \left| \frac{\partial^4 I}{\partial x^4} \right| \right) + \max \left( \left| \frac{\partial^4 I}{\partial y^4} \right| \right)]$$

#### 5.1.4 比较

首先再直观上，由于线性与双三次插值加入了“估计”颜色的方法，因为在再实际运行中其真实度远高于最临近插值方法。而在图片较小时，线性插值与双三次插值肉眼分析效果差别并不大，但是在再放大模式下双三次插值效果明显优于双线性插值。

我们将一张（256\*256）图片降维到（64\*64）再重新还原比较不同插值算法的可见差异。



（从左至右依次为原图、采样图像，最邻近插值，双线性插值与双三次插值）

理论分析中如果认为不同阶数高阶导数值上限相同，则可以发现插值次数越高，误差相对也会越小。

## 5.2 图像变形产生的误差

由于再计算变形函数时，我们的数值全部由浮点数表示，考虑进去计算次数，总体误差在 $10^{-8}$ 量级左右，这个误差相比于其他部分过小因而可以忽略不计。

## 5.3 颜色整型存储表示产生的舍入误差

由于数字图像中颜色均为  $0 \sim 255$  的整数值，因而对像素点  $I(i, j)$  我们有舍入误差：

$$|\Delta I(x,y)| \leq 0.5$$



## 6 总结

本次数值分析大作业我们完成了三种不同的图像插值算法与三种不同的图像变形函数，并将他们有机地组合在一起完成了图像变形与人脸畸变两项任务，并在此基础上设计了图形化界面与可执行程序。本项目亮点有：

- 良好的程序封装结构，将颜色插值与图像变换任务有效解耦合出来，并设计了良好的调用架构。
- 简洁易用的界面，实现了点击图像更改，滑动条调参等功能，帮助用户快速上手。

## 7 参考文献

[1] R. Keys (1981). "Cubic convolution interpolation for digital image processing". IEEE Transactions on Acoustics, Speech, and Signal Processing. 29 (6): 1153 - 1160.

[2]李庆扬、王能超、易大义《数值分析(第5版)》p168.