

USTC_CG HW2 Image Warping

张继耀,PB20000204

2023 年 6 月 16 日

目录

1	问题介绍	1
1.1	主要目的	1
2	算法设计	2
2.1	程序框架	2
2.2	IDW算法	2
2.3	RBF算法	3
2.4	填补白缝	3
3	结果展示	4
3.1	程序界面	4
3.2	功能讲解	4
3.3	实验结果	4
3.3.1	向外拉	5
3.3.2	向内拉	6
3.3.3	旋转变换	7
3.3.4	其他例子	7
4	总结与讨论	7

1 问题介绍

1.1 主要目的

- 阅读论文，实现IDW、RBF算法
- 进行内存泄露检测
- 填补可能产生的白缝

2 算法设计

2.1 程序框架

下面我们首先给出整个项目的类视图

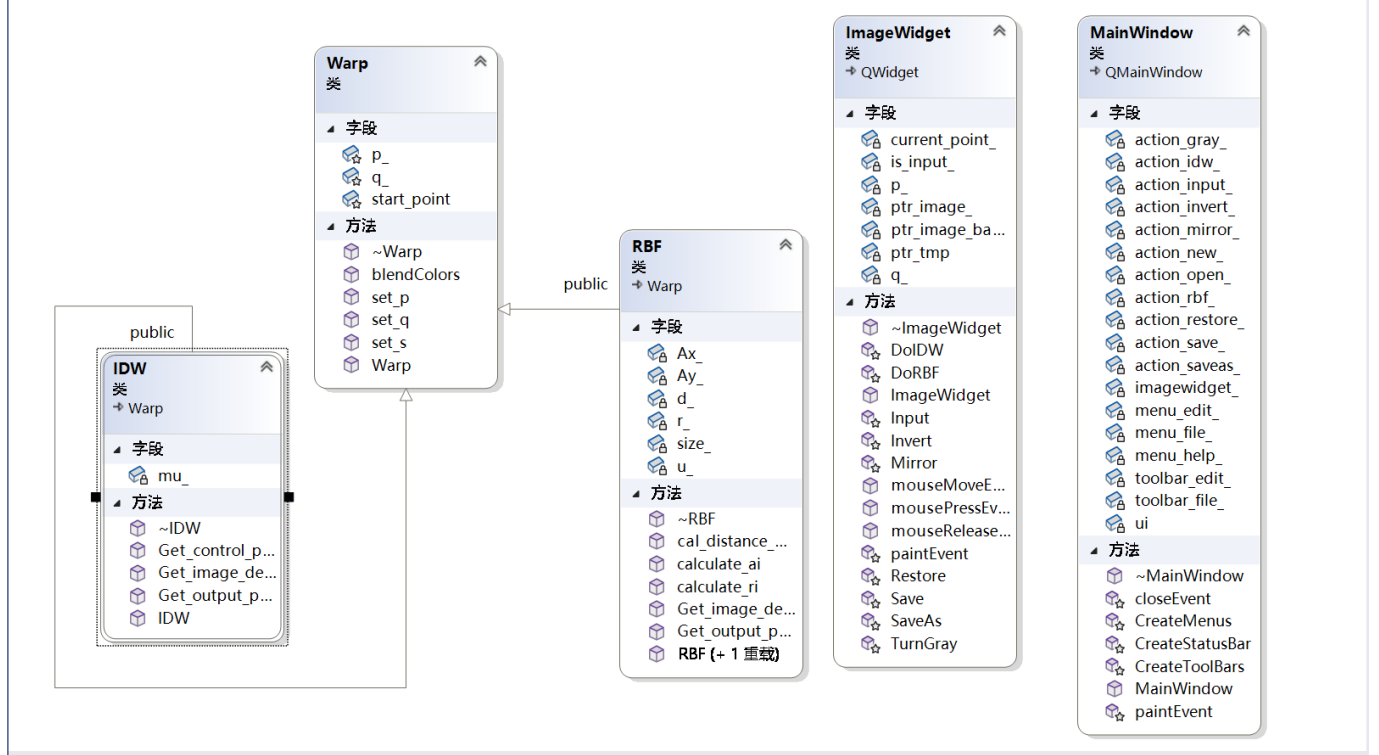


图 1: 项目的类视图

两种方法分别用一个类来封装，并且这两个类都继承于父类warp。

接下来我们描述一下需要实现的算法:

输入: n 对控制点 $(p_i, q_i), p_i = (x_i, y_i), q_i = (x'_i, y'_i) \in \mathbb{R}^2$, 其中 $i = 1, \dots, n$. 其中 p_i, q_i 分别为起点和终点。

目标: 找到连续映射 $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, 满足 $f(p_i) = q_i, i = 1, \dots, n$.

2.2 IDW算法

IDW算法的思路就是找到一个局部近似函数，保证起始点可以变换到终点。

对原图中任一点P，变换后的点Q由下述的表达式给出

$$f(p) = \sum_{i=1}^n w_i(p) f_i(p)$$

其中 $w_i(p)$ 是权重函数，满足方程：

$$\begin{cases} w_i(p_i) = 1 \\ \sum_{i=1}^n w_i(p) = 1, \forall p \in \mathbb{R}^2 \\ w_i(p) \geq 0, i = 1, \dots, n \end{cases}$$

DONALD SHEPARD提出了可以取权重函数为

$$w_i(p) = \frac{\sigma_i(p)}{\sum_{j=1}^n \sigma_j(p)}$$

其中 $\sigma_j(p) = d(p, p_j)^{-\mu}$ ，在本实验中取 $\mu = 2$ 。

取局部近似函数 f_i 为线性函数，考虑 $f_i(p) = q_i + (p - p_i)T_i$ ，并满足 $T_i : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ， $T_i(0) = 0$ 。

通过解如下的最小二乘问题来确定 T_i ：

$$\min \sum_{j=1, j \neq i}^n \sigma_i(p_j) \|T_i(p_j - p_i) + q_i - q_j\|^2$$

2.3 RBF算法

RBF算法是另一种思路的变换函数，它的取值仅依赖于点到标记点之间的距离。函数定义为如下形式：

$$f(p) = A(p) + T(p)$$

$$T(p) = \sum_{i=1}^n \alpha_i g_i(d(p_i, p))$$

其中

$$g_i(d) = (d^2 + r^2)^{\frac{\mu}{2}}$$

μ 可以取1或-1，本次实验默认取-1。 r 为 $\min_{j \neq i} d(p_i, p_j)$ ，系数 α_i 可以通过解方程组 $f(p_i) = q_i$ 解得。

至此，需要处理的问题本质上已经转化为了解线性方程组。

2.4 填补白缝

由于在对图像进行变形的过程中存在对图像的拉伸，而处理的图形像素点是离散的，算法中构造的映射 f 不一定是一个满射，因此可能会出现无法填充到的点，也就是白缝。

在实验中用周围非白缝的点来填充出现的白缝，即搜索 3×3 内最近的点来填充。一开始我还打算取颜色RGB的均值来填充，后来经过对比发现并没有上面的这个方法好，于是作罢了。

3 结果展示

3.1 程序界面

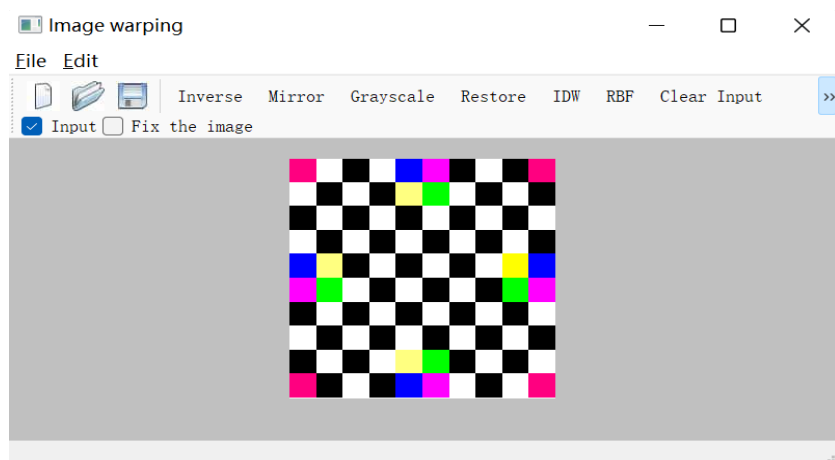


图 2: 程序界面

程序界面如上所示

3.2 功能讲解

- 最左边的图标是打开、保存图片
- Restore可以恢复图片原始的样子，Inverse,Mirror,Grayscale可以直接修改图片像素，具体可以看程序的说明。
- IDW，RBF在给定输入点的情况下会对图像做IDW，RBF变换
- 交互时，勾选Input，即可开始绘制输入点，蓝色代表起点，绿色代表终点。若要清除输入点，请点击Clear Input。取消勾选Input可以不显示输入的点，但已有的点还会保存(请注意)。
- 勾选Fix the image时，再用IDW、RBF时则会填补产生的白缝。使用时可以先不勾选，点击IDW/RBF，在Restore加载到原图，勾选上Fix在进行一次IDW/RBF。

3.3 实验结果

注意：下面为了效果明显，都是固定四个边界点不动的

3.3.1 向外拉

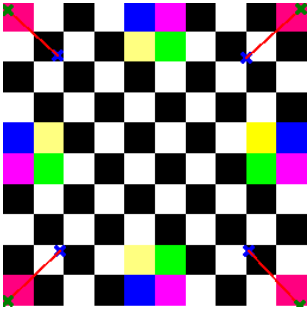


图 3: 控制点

是否填充 使用方法	是	否
	是	否
IDW		
RBF		

表 1: 向外拉的效果

3.3.2 向内拉

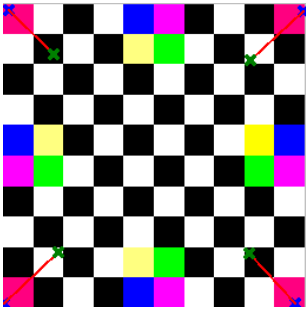


图 4: 控制点

是否填充 使用方法	是	否
	是	否
IDW		
RBF		

表 2: 向内拉的效果

3.3.3 旋转变换

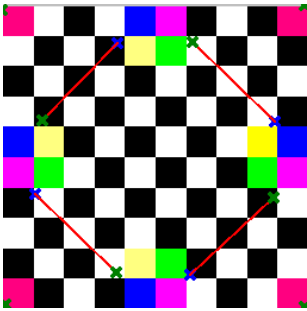


图 5: 控制点

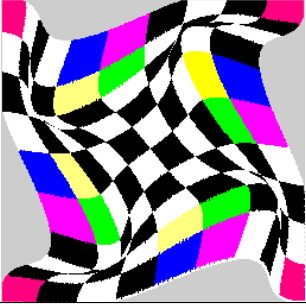
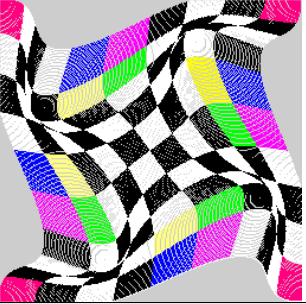

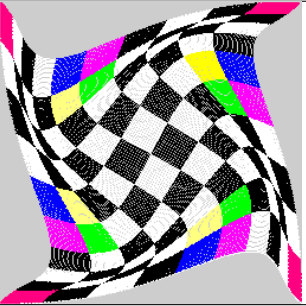
是否填充 使用方法		是	否
IDW			
RBF			

表 3: 旋转的效果

3.3.4 其他例子

下面是一个处理蒙娜丽莎图像的测试数据(图6、图7):

4 总结与讨论

- 关于RBF和IDW的对比，直观上的感受是，RBF在收缩或拉伸时会更“好”一些，也就是会更加明显一点。比如同样拉伸，感觉RBF拉的距离会更远一点。

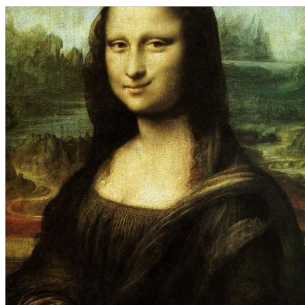


图 6: 微笑

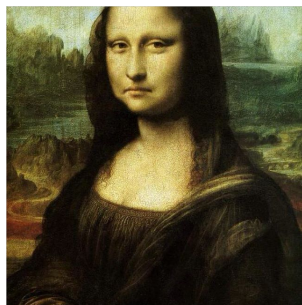


图 7: 悲伤

- 在性能上，能感受到IDW的性能是要差于RBF的。RBF运行速度快，复杂度也更低。不过时间原因，没有详细分析了。