

Lab 2- Image Warping

By PB17111585 张永停

一、实验内容

- 通过两种不同的算法实现图像变形，并对拉伸之后的图像产生的缝隙进行填充
- 进行内存检漏
- 默认图像的四个顶点固定不变的

二、算法描述

（一）IDW算法

- 找到满足 $f(p) = \sum w_i(p)f_i(p)$ 的局部近似函数
 1. $f_i(p_i) = q_i$
 2. 且 $\sum w_i(p) = 1$
- Shepard提出权重函数 $w_i(p) = \frac{\sigma_i(p)}{\sum_{j=1}^n \sigma_j(p)}$ ，其中 $\sigma_j(p) = d(p, p_j)^{-\mu}$ ， $d(p, p_j)$ 是 p 与 p_j 的距离， μ 是大于0的任意数
- 本次实验中，取局部近似函数为线性函数，考虑 $f_i(p) = x'_i + (p - p_i)T_i$ ，误差函数 $E(T_i) = \sum_{j=1, j \neq i}^n \sigma_i(p_j)(f_i(p_j) - x'_j)^2$ 。为使误差函数尽可能小，可对 T_i 求导数，得到线性方程组，从而通过解线性方程组，可以解得 T_i

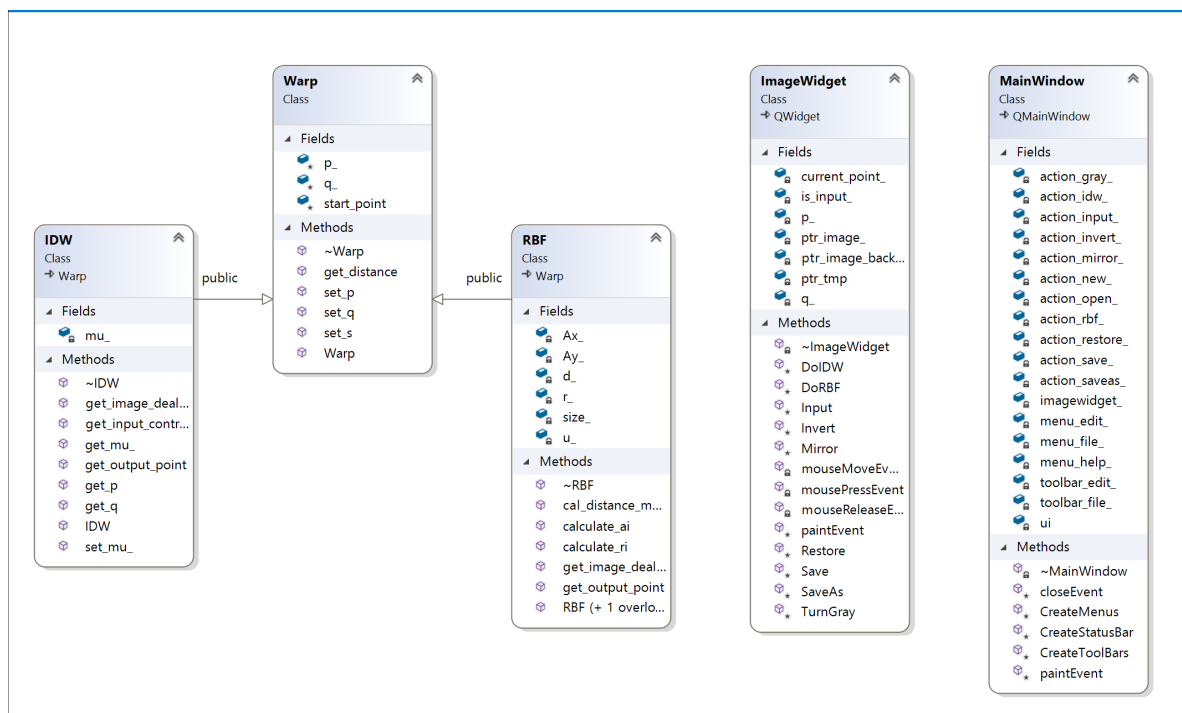
（二）RBF算法

- RBF给出的坐标变换公式为
$$f(p) = \sum_{i=1}^n \alpha_i g_i(d(p_i, p)) + Ap + B$$
，其中 g_i 为基函数
- 基函数取 $g_i(d) = (d^2 + r_i^2)^{\frac{1}{2}}$, $r_i = \min_{j \neq i} d(p_i, p_j)$
- 系数 α_i 可以通过方程组 $f(p_i) = q_i$ 解得

（三）白缝填充

- 由于变换过程存在拉伸，且像素点是离散的，故有些点可能并没有被别的点变换到。填充方法为在空洞周围 3×3 的矩阵区域中选择任意不为空的像素填补

三、代码框架



(一) Warp类

是虚类，主要被IDW和RBF所继承

1. IDW

使用IDW插值法

o public

`get_input_control_point_weight` 获取每个点对应控制点的权重表

`get_output_point` 获取输入点的变换点

`get_image_deal_with_IDW` 获取使用IDW算法处理过的图片，同时在该函数中进行白缝填充

`set_mu_` 设置参数

o private

`double mu_` 参数

2. RBF

使用Eigen库来解方程

o public

`cal_distance_martix` 计算距离矩阵

`calculate_ri` 计算p个数据点的R值

`calculate_ai()` 计算方程组的ax,ay

`get_image_deal_with_RBF` 获取使用RBF算法处理过的图片，同时在该函数中进行白缝填充

o private

`u_` 参数

`d_` 距离矩阵

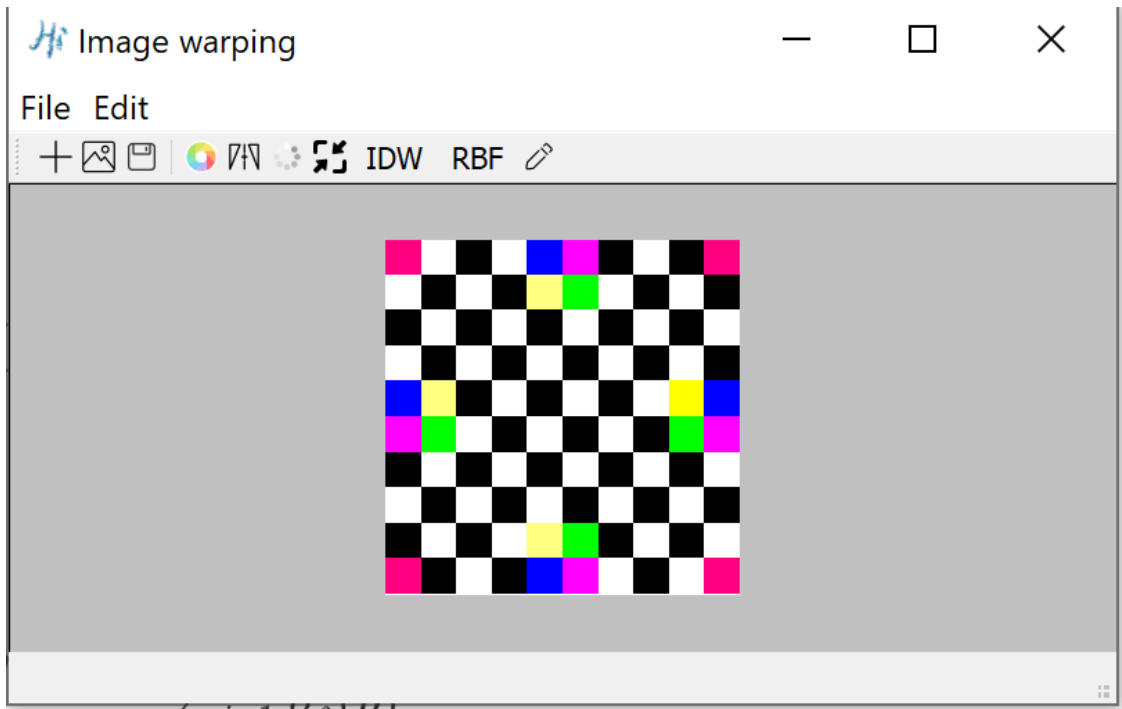
`r_` Ri数组

(二) ImageWidget类

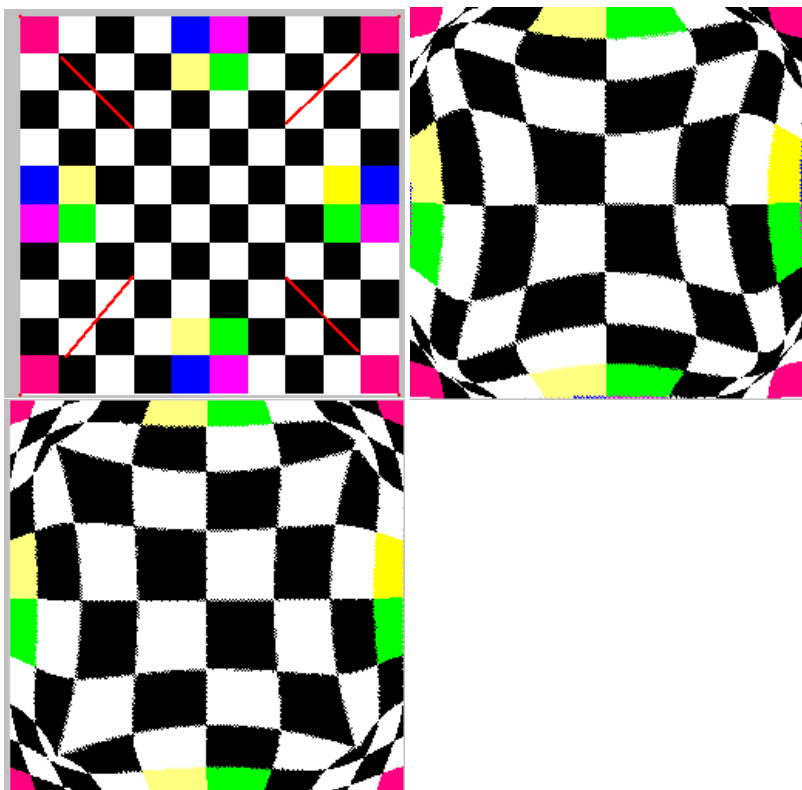
用于接收信号，是ui与算法的过渡层

四、实验结果

- 交互界面

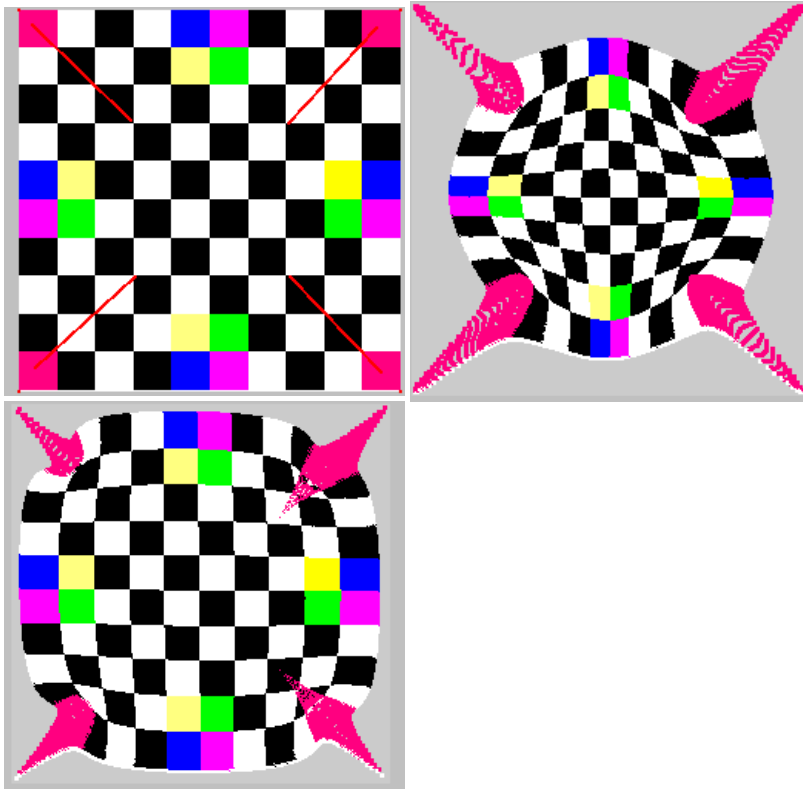


- 向外拉



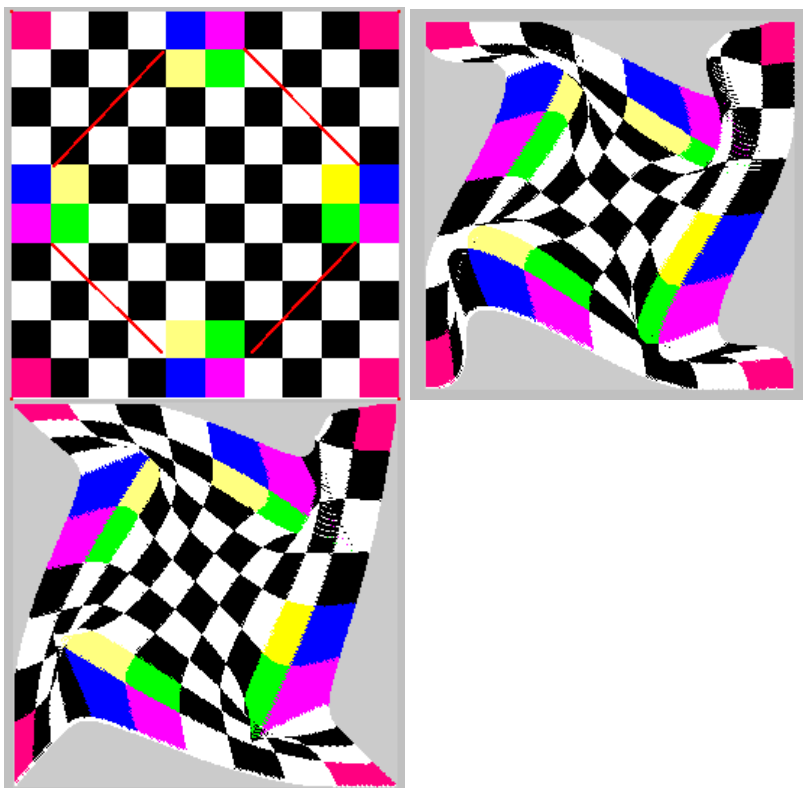
左: 拉伸示意, 中:IDW, 右:RBF

- 向内拉



左: 拉伸示意, 中:IDW, 右:RBF

- 旋转拉伸



左: 拉伸示意, 中:IDW, 右:RBF