

一、问题描述

- 阅读论文,实现Image warping的两种方法
 - IDW(Inverse distance-weighted)方法
 - RBF(Radial basis functions)方法
- 使用QImage类四个接口函数width(), height(), pixel(), setPixel()

二、算法描述

1.反距离加权插值(IDW)算法

- 通过给定的一组控制点对, 计算控制点间的位移矢量, 实现图像每一个像素点的位移。
- 反距离加权插值(IDW)通过计算图片上每一个像素点和一组控制点对间的逼近关系以及权重关系, 求得像素点变化之后的位置。
- 求解像素点变化的函数f(p)如下:

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

其中 $w_i(p)$ 是权重函数

$$w_i(p_i) = 1, \sum_{i=1}^n w_i(p) = 1 \text{ 且 } w_i(p) \geq 0 \quad i = 1, \dots, n$$

$$w_i(p) = \frac{\sigma_i(p)}{\sum_{j=1}^n \sigma_j(p)} \text{ 且 } \sigma_i(p) = \frac{1}{d(p, p_i)^\mu}$$

$f_i(p)$ 为逼近函数

$$f_i(p) = q_i + T_i(p - p_i)$$

- T是一个2*2矩阵, 可以取近似值1。将控制点数据代入可以求出每一个局部插值函数。
- 这里d (p, pi) 为p和 p_i 的距离, 指数 μ 可以取>0的数, 这里测试取2最优。通过公式求取权重函数, 求和体现多组控制点共同产生影响。
- IDW 插值法是一种全局插值法,即全部样本点都参与某一待估点的估算之中,所以它的计算复杂度为O(nN),其中n 为样本点个数,而 N 需要变换的点的总个数。

2.径向基函数(RBF)算法

- RBF 方法已经被验证,在离散数据的多元插值问题上是一个很有效的工具。该方法要求我们找到满足以下形式的插值函数:

$$f(p) = \sum_{i=1}^n \alpha_i f_i(d(p, p_i)) + P_m(p) \quad (2)$$

- $\alpha_i \in \mathbb{R}^2$ 是系数向量, $P_m : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ 是一个m次的多项式函数。当m=1时, 反而有比高次多项式更好的表现。

- $f_i(x)$ 是关于正实数 $d(p, p_i)$ (距离)的函数。
- RBF 插值法是一种全局插值法,即全部样本点都参与某一待估点的估算之中,所以它的计算复杂度为 $O(nN)$,其中 n 为样本点个数,而 N 需要变换的点的总个数。

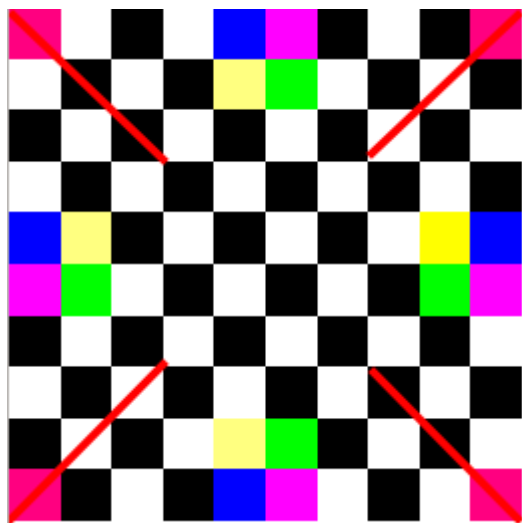
3.白缝问题

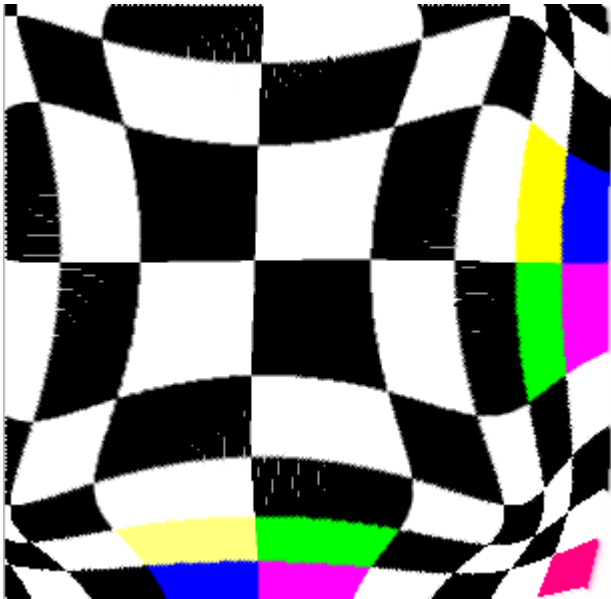
- 我的解决方式是新建一个bool类型的二维数组，用于存储新生成的图像上的点的映射情况。初始值全部设为 false，如果某一个点被映射到了，那么将这个点的位置所对应的数组中的元素设为true。
- 使用数组中的值来遍历判断图像中的每一个点是否是没有被映射到的点，没有被映射到的点正是白缝问题的根源。这里分成两种情况解决：
 1. 未映射点周围八个点有映射点，那么只取映射点的颜色的平均值，并将这个颜色赋给未映射点。
 2. 未映射点周围八个点没有映射点，那么取周围八个点的颜色的平均值，并将这个颜色赋给未映射点。

三、结果

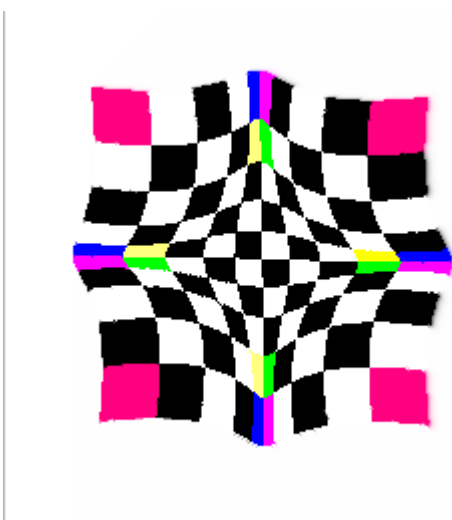
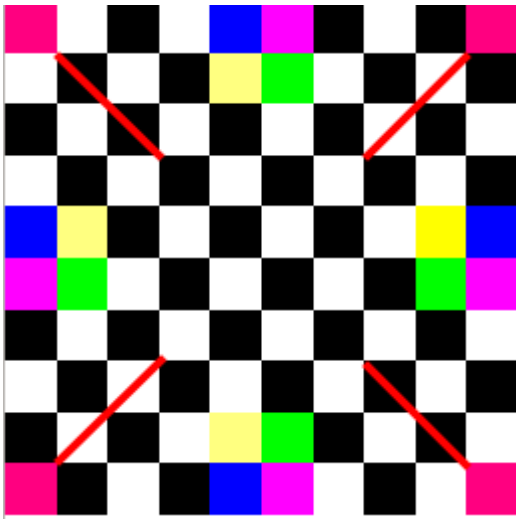
1.IDW

四角拉伸：

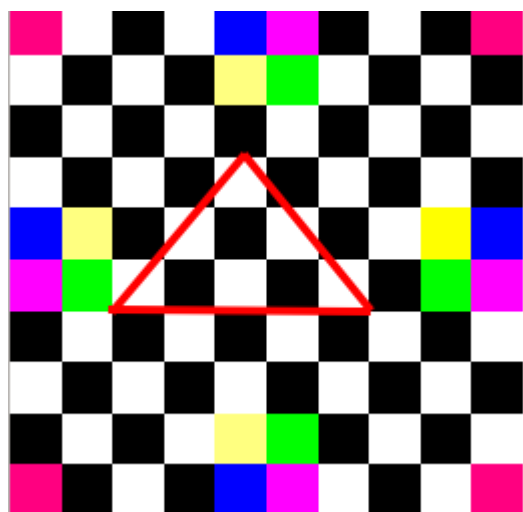




四角压缩：

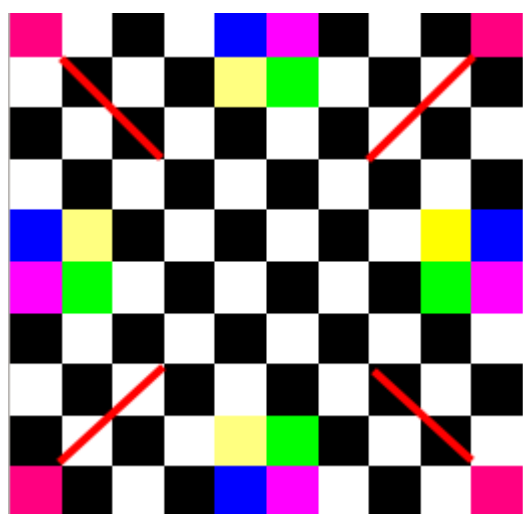


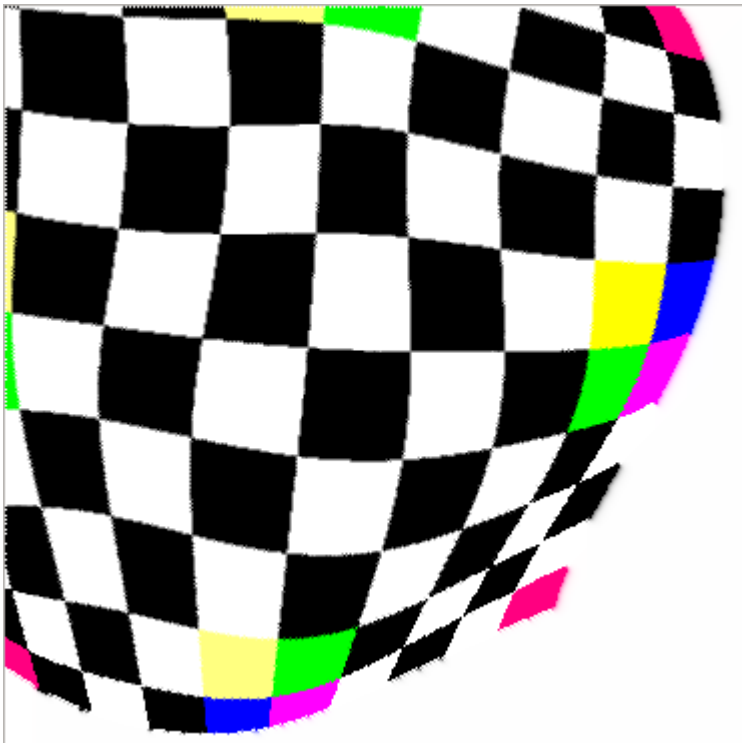
三角形：



2.RBF

四角拉伸：

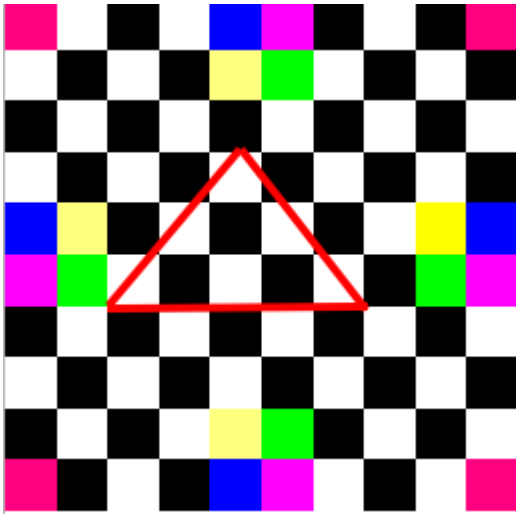




四角压缩：



三角形：



四、总结

通过这次作业，我深刻地体会到了精妙的图形变换离不开复杂数学算法的支持。并且我意识到如果之后碰见类似白缝问题这样的难以解决的问题，可以在有初步的想法之后实现一个雏形，然后根据雏形不断调整，最终可以调整出比较理想的结果。