

图像扭曲

● 简述

图像扭曲是一种比较常见得图像变化，本作业使用 IDW 和 RBF 两种方法来实现图像扭曲。

● 原理

IDW(Inverse Distance Weighted)是一种常用而简便的空间插值方法，它以插值点与样本点间的距离为权重进行加权平均，离插值点越近的样本点赋予的权重越大。

RBF(Radial basis function)径向基函数是一个取值仅仅依赖于离原点距离的实值函数。RBF 网络能够逼近任意的非线性函数，可以处理系统内的难以解析的规律性，具有良好的泛化能力，并有很快的学习收敛速度，已成功应用于非线性函数逼近、时间序列分析、数据分类、模式识别、信息处理、图像处理、系统建模、控制和故障诊断等。

● IDW

为了达到变换效果我们需要下面的函数：

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

方程输入一个点的坐标 p 得到了移动之后该点的坐标。其中 w 函数计算权重， f_i 计算出的为平移操作后的坐标。

为了计算权重我们使用了以下函数：

$$\begin{cases} w_i(\vec{p}) = \frac{\sigma_i(\vec{p})}{\sum_{j=1}^n \sigma_j(\vec{p})} \\ \sigma_i(\vec{p}) = \frac{1}{d(\vec{p}, \vec{p}_i)^\mu} \end{cases}$$

其中 d 函数得到的是两点之间的距离， μ 常数可以根据条件自行设置， p 为输入点， p_i 为控制点对的起始点，当两者相同时有：

$$w_i(\vec{p}_i) = 1$$

同时为了让点产生平移我们需要下面的函数：

$$f_i(\vec{p}) = \vec{q}_i + T * (\vec{p} - \vec{p}_i)$$

方程中 p 为输入点 q_i 为控制点的终止点， p_i 为控制点的起始点， T 为一个矩阵，为了方便计算在实现中我们可以取：

$$T = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

● RBF

同上，我们首先需要一个转换函数，输入一个点，然后经由函数计算得到移动之后该的坐标，函数如下：

$$f(\vec{p}) = \sum_{i=1}^n \vec{\alpha}_i * (d(\vec{p}, \vec{p}_i)^2 + r_i^2)^{\frac{\mu}{2}} + \vec{p}$$

其中 d 函数得到的是两点之间的距离， μ 常数可以根据条件自行设置，p 为输入点， p_i 为控制点对的起始点。

对于 r 我们可以通过下面的方法得到：

$$r_i = \min_{i \neq j} d(\vec{p}_j, \vec{p}_i)$$

α_i 是一组向量，通过在求解上式的过程中得到的一个临时矩阵做高斯矩阵变换然后每一个控制点对向量做一个权值运算得到。

● 分析

在试验中我们发现图片中会出现白点或者白线，分析原因，我认为有两点。

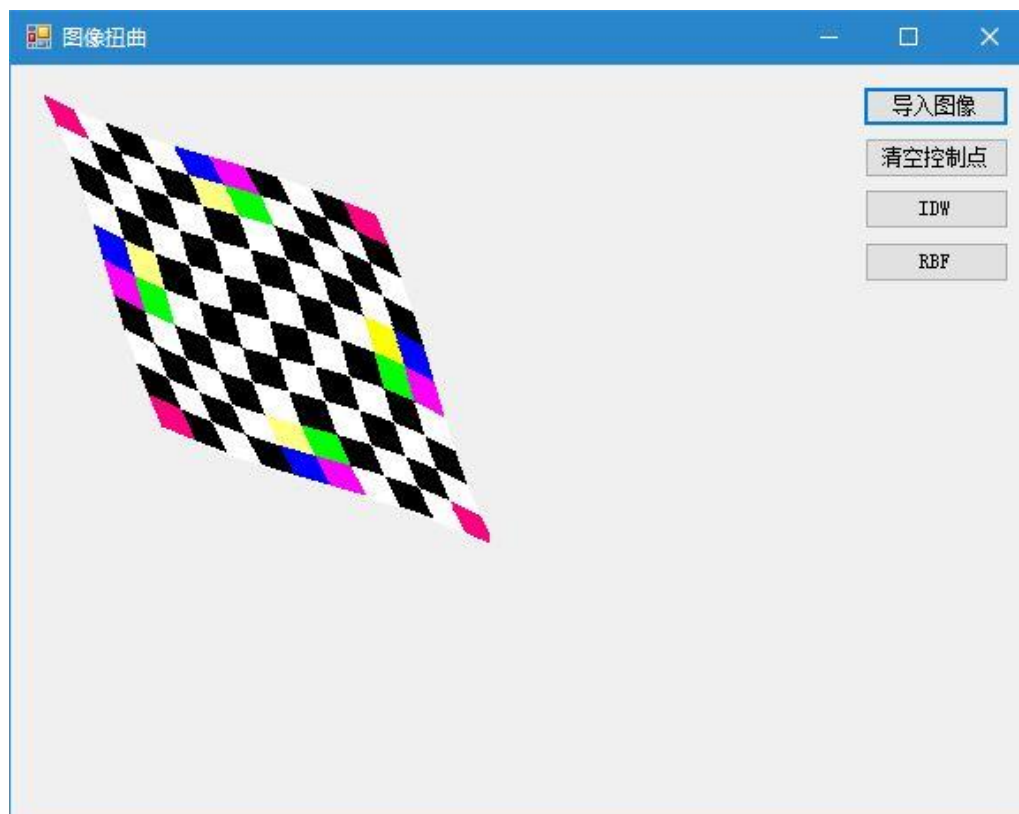
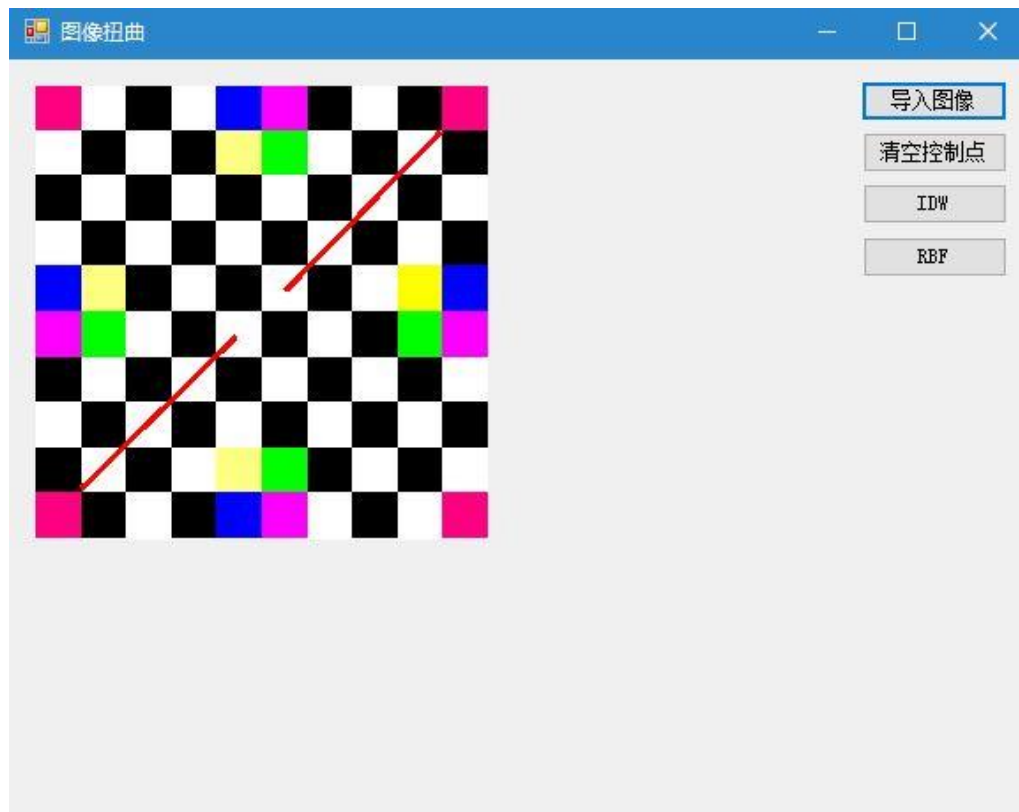
第一点是精度问题，我们在计算中使用了 double 类型，而图片中像素点的坐标只能为整数，这可能会造成部分像素信息偏离导致白点和白线的产生。

第二点是像素信息丢失或重叠，在计算的过程中有些像素点移动到了图片的范围之外，导致图片的部分像素信息丢失，另外还存在某些像素点在计算后移动到了同一个位置，这也可能导致白线或者白点的产生。

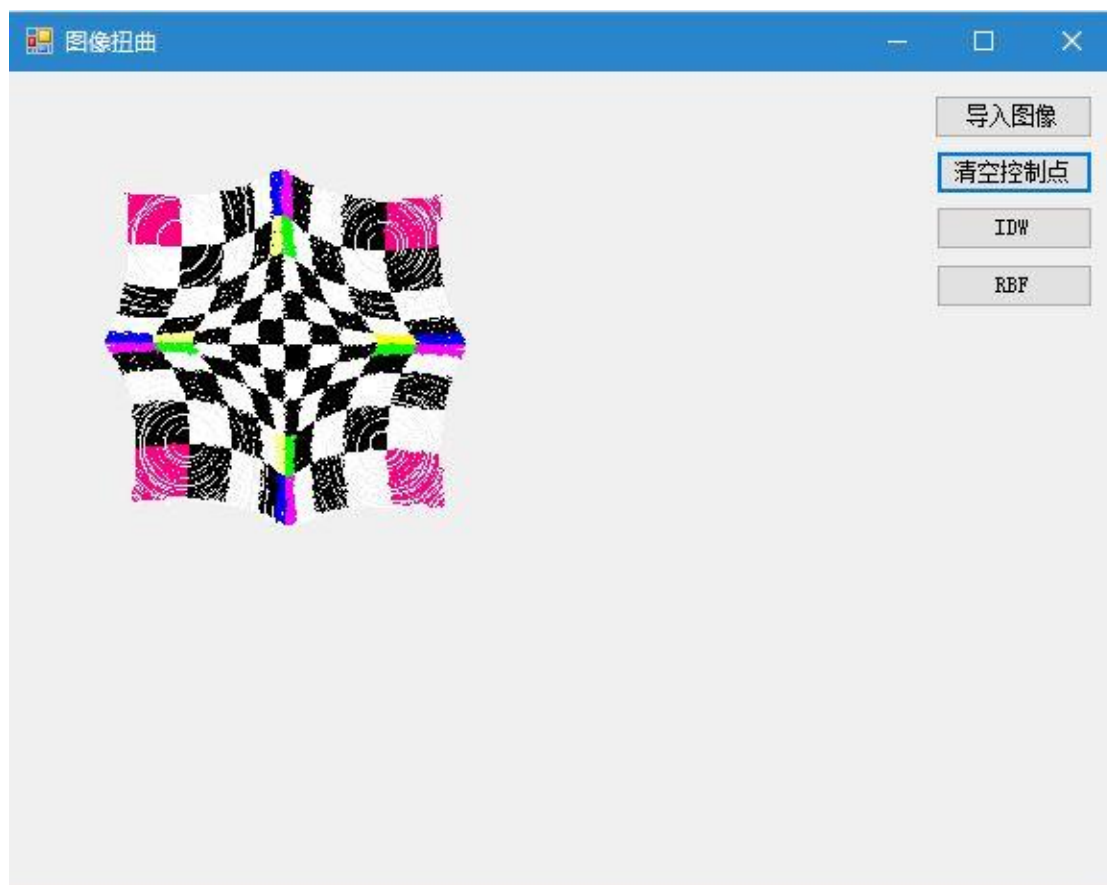
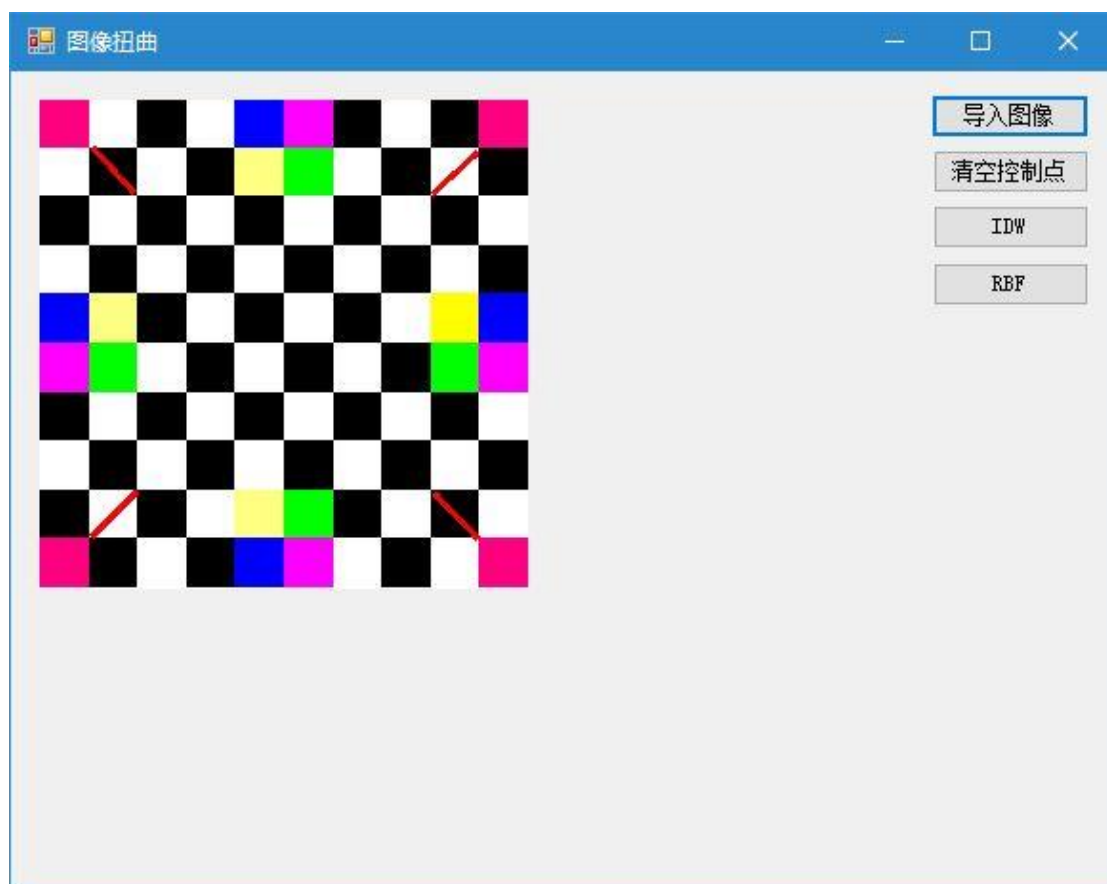
为了解决这个问题，我们可以设计一个补插值的算法，根据空白像素点周围的像素信息来补充这个像素点丢失的信息。

● 实验结果

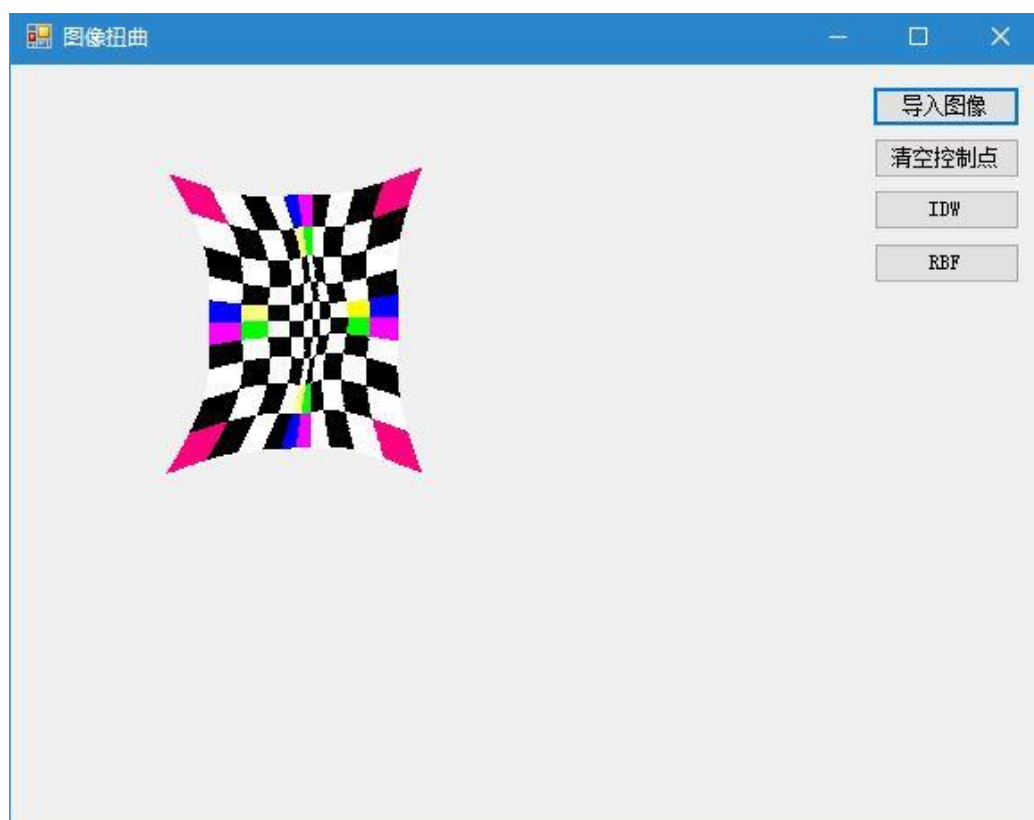
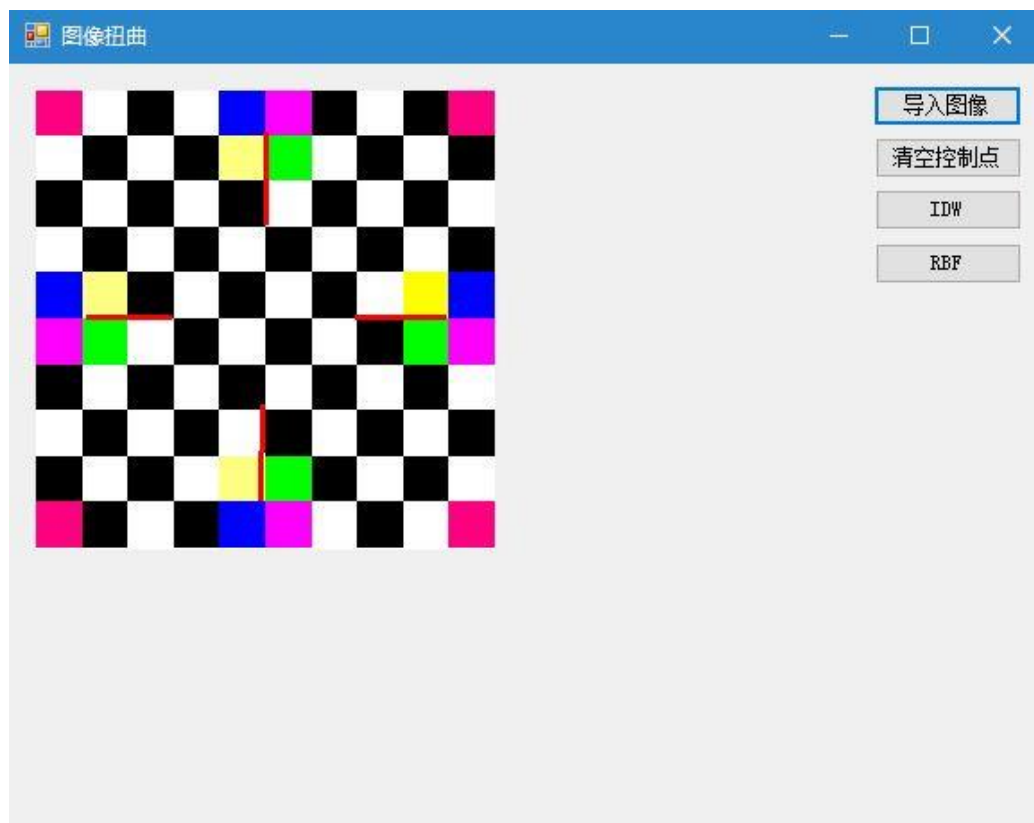
RBF:



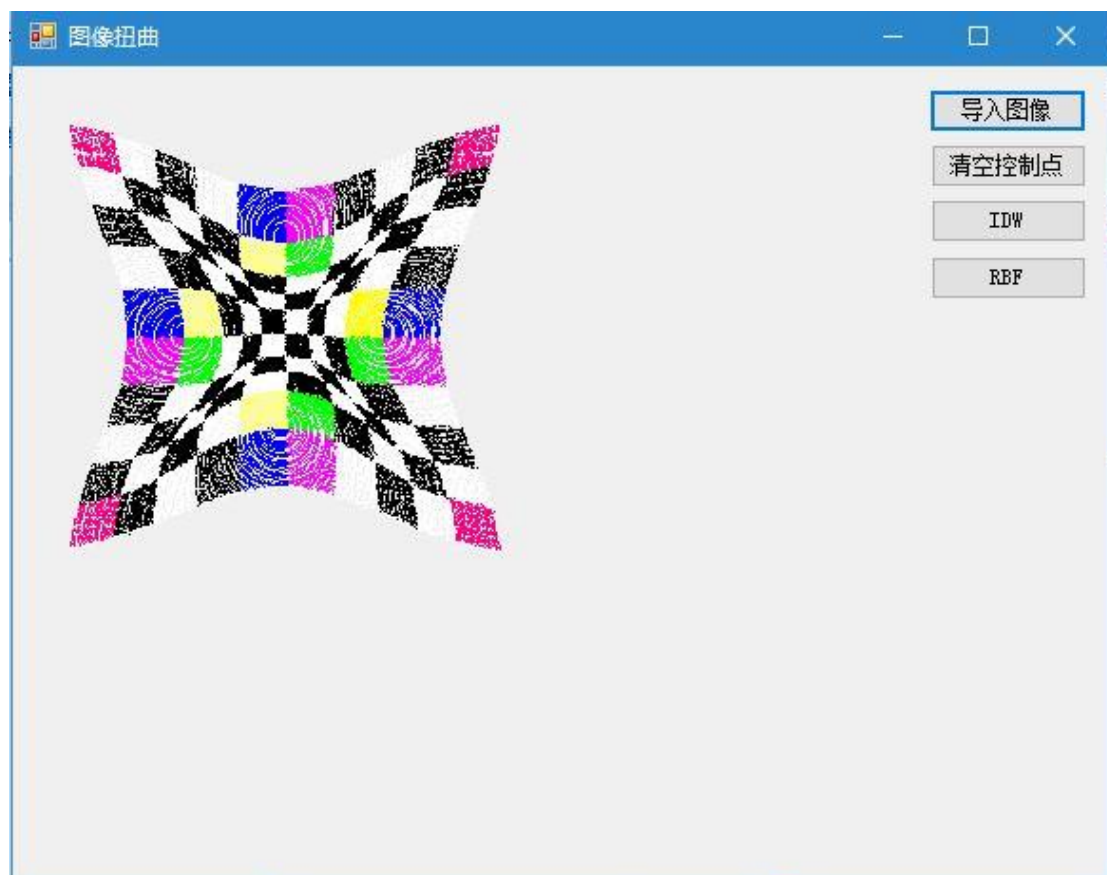
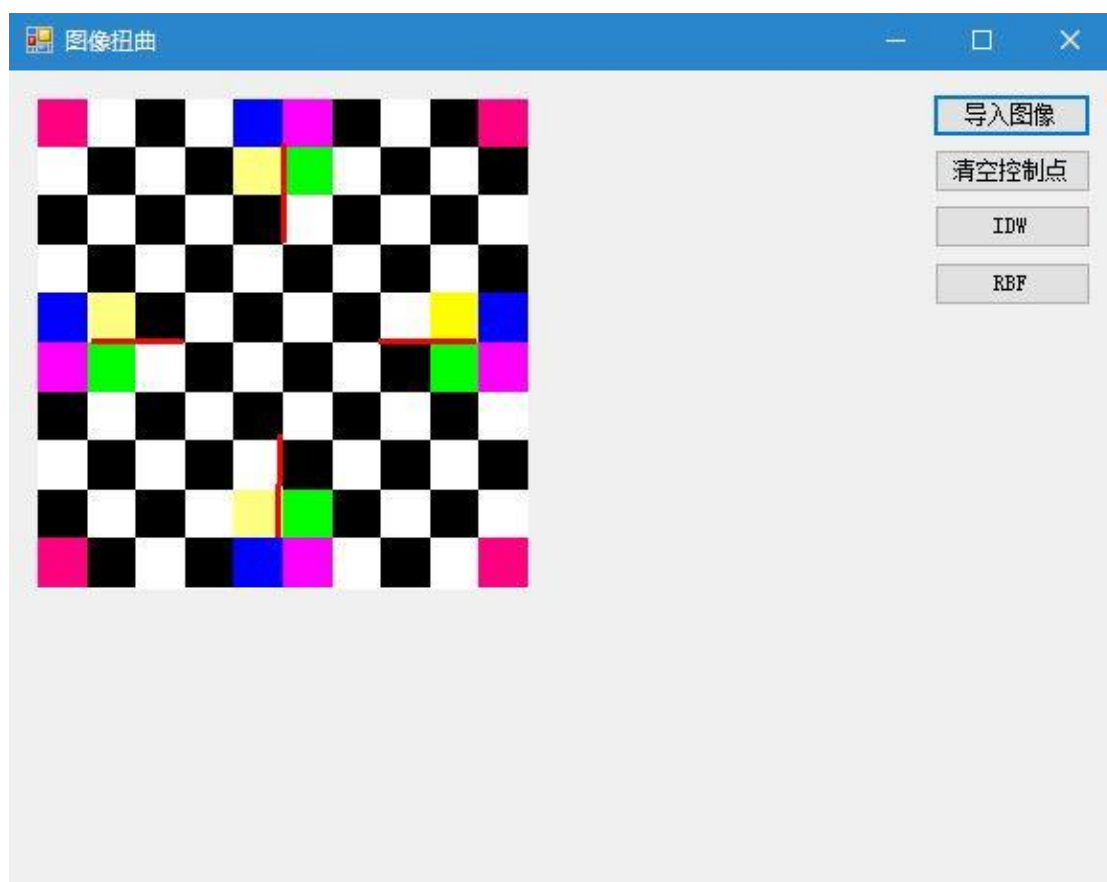
IDW:



RBF



IDW:



RBF:

