

# Matlab-1

高悟恒

2020-09-17

一. 算法: 设有给定插值条件: 起点  $x_i(p_i), i = 1, \dots, n$  和终点  $q_i, i = 1, \dots, n$ 。

1.IDW:

取形式为  $f(x) = \sum_{i=1}^n \omega_i f_i(x)$  的函数作为插值函数。

其中  $f_i(x) = q_i + D_i(x - p_i)$ , 根据论文这里选取  $D_i$  为线性变换。

因此  $D_i$  为一个二阶方阵, 记为  $(d_{i,kl}), k, l = 1, 2$ 。

由  $D$  导致的误差函数为:  $E_i(D) = \sum_{j=1, j \neq i}^n \omega_{ij} \|q_i + D(x_j - x_i) - q_j\|^2 = \sum_{j=1, j \neq i}^n \omega_{ij} ((d_{11}(p_{j,1} - p_{i,1}) + d_{12}(p_{j,2} - p_{i,2}) + q_{i,1} - q_{j,1})^2 + (d_{21}(p_{j,1} - p_{i,1}) + d_{22}(p_{j,2} - p_{i,2}) + q_{i,2} - q_{j,2})^2)$ 。

其中系数  $\omega_{ij}$  应当由  $x_i$  和  $x_j$  的距离决定, 一种简单有效的选择是  $\omega_{ij} = \sigma_i(x_j), \sigma_i(x) = \frac{1}{(d_i(x))^\mu}$ 。

求误差函数最小值是一个极小二乘问题, 只需对  $d_{kl}$  求偏导取零即可。

权重函数  $\omega_i$  需满足条件  $\omega_i(x) \geq 0, \sum_{i=1}^n \omega_i(x) = 1, \omega_i(x_j) = \delta_{ij}$ 。

论文中选取了  $\omega_i(x) = \frac{\sigma_i(x)}{\sum_{i=1}^n \sigma_i(x)}, \sigma_i(x) = \frac{1}{(d_i(x))^\mu}, \mu > 0$ , 这里为了便于计算通常选取  $\mu = 2$ 。

2.RBF:

取形式为  $T(x) = A(x) + R(x)$  的函数作为插值函数。

其中  $A(x) = Mx + b$  为仿射变换, 可以根据初始设定计算, 这里我们选取其为恒等变换。

而  $R(x)$  为径向基函数,  $R(x) = \sum_{i=1}^n a_i g(x - x_i)$ , 这里选取  $g(d) = (d^2 + r^2)^{\mu/2}, \mu = -1$ , 其中  $r_i = \min \{i \neq j\} d(x_i, x_j)$ 。

最后参数  $a_i$  可以由给定的插值条件求解方程组得到。

二. 实验结果:

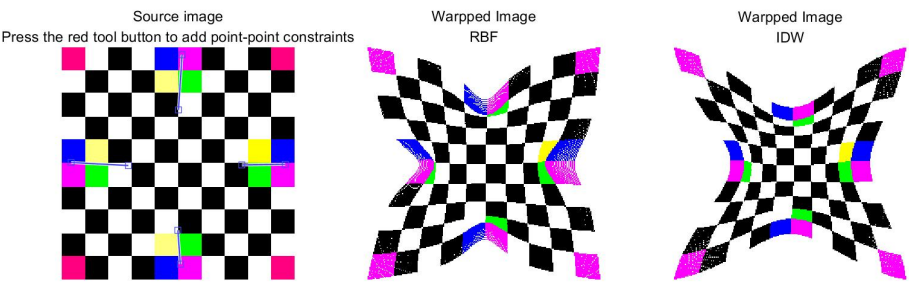


图 1: 变形效果（压缩）

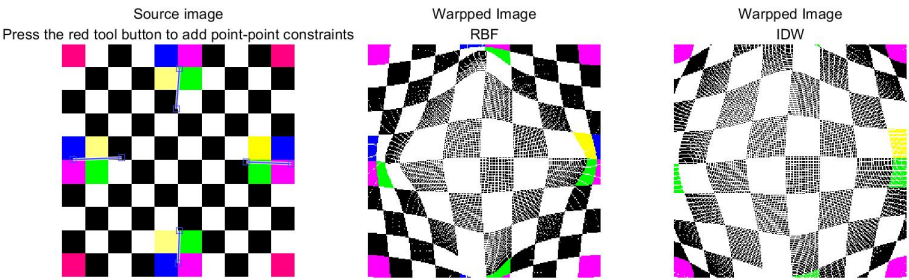


图 2: 变形效果（拉伸）

这里我们在编程中默认固定住四个角，点红色按钮添加约束条件，点蓝色按钮计算变形图片。程序实现了拖动约束条件的位置时，同时计算变形图片，但是不足之处在于计算时间较长，拖动时不够流畅。分别压缩或拉伸四条边的中点处，可以看到两种方法的计算效果都较好。

三. 白缝：这两种方法都是寻找一种连续的插值函数对图像进行变换，但是图像本身是离散的，这就导致我们只能计算原有图像上的点对应的点，而为了得到新的图像就需要对计算出来的点进行取整。又由于函数必然不是线性的，因此就会出现原来密集的点映射到相对较分散的区域的情况，这就产生了白缝。