关于 Image warping 的实验报告

SA20001911 王克淳

1. Inverse distance weight(IDW) 方法

1.1 算法说明

利用给定的控制点的矢量位移，构造插值函数，最终实现每一个像素点的位移变化。反距离加权方法得到的插值函数与位置距离有关，可以理解为未知点到给定点的距离越近，则影响越大，反之距离越近，则影响越小。

1.2 具体算法

插值函数的具体构造方法如下：

插值函数具有以下形式：

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

其中表示权值的函数需要满足下列条件：

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

特别的，在实现过程中，可以取以下形式，d 表示两点之间的距离。

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

那么，在二维图像的情况下，还需要确定函数 fi 的具体形式，在论文中给出了如下一个常用的结果：

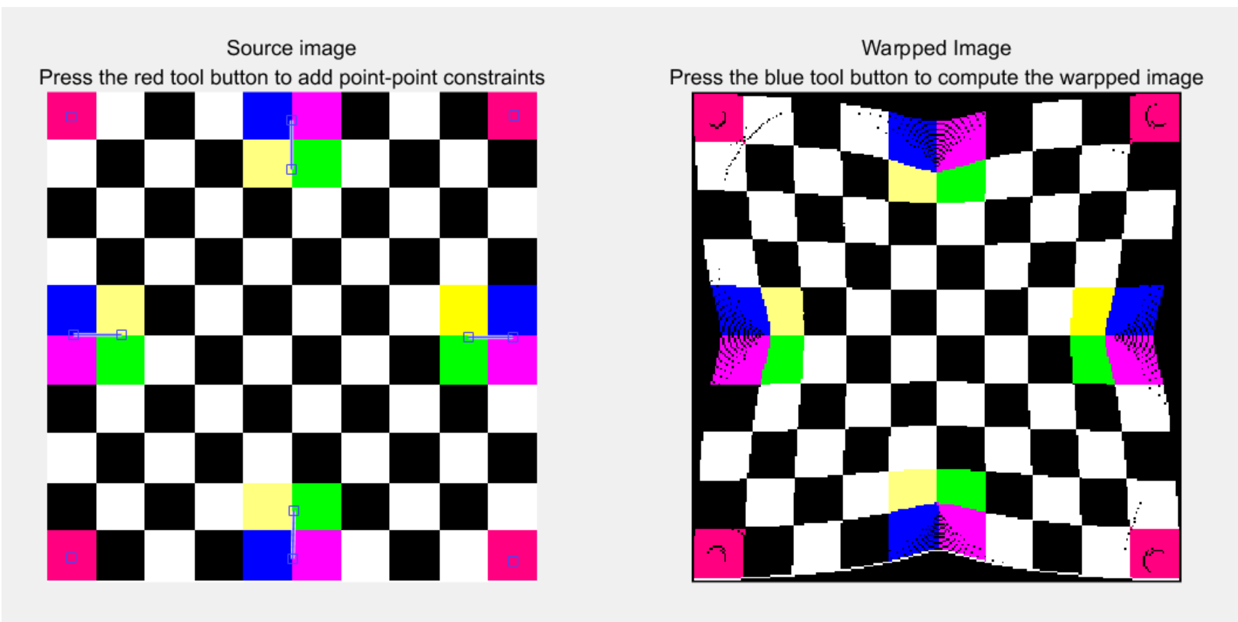
$$\mathrm{f_i}(\mathbf{p})=\mathbf{q_i}+\mathbf{T_i}(\mathbf{p}-\mathbf{p_i})$$

其中，**q** 代表位置矢量的终点的位置。2 阶矩阵 T 则可由误差函数求出，使得 T 满足误差最小。一般情况下，论文指出，待定 T 时，实际上对矩阵进行 SVD 分解后是一个最小二乘法的极值问题，误差函数如下：

$$E_i(\mathbf{T}) = \sum_{j=1, j\neq i}^{n} \sigma_i(\mathbf{p}_j) \cdot \left\| \mathbf{q}_i + \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} (\mathbf{p}_j - \mathbf{p}_i) - \mathbf{q}_j \right\|^2$$

1.3 运行结果

在取定μ=1 的情况下，函数的输入输出结果如下：



1.4 结果分析

可以看出，这里出现了部分像素丢失的问题，最主要的原因应该是在图像的拉伸过程中，部分像素点出现了丢失，可能需要采取别的一些方法来补全像素点。

关于算法的时间复杂度，由于对全部像素点进行了一次计算复杂度应该为 O(nN)，这里由于给定的固定点的数量 n<<N，所以不用考虑跟 n 有关的时间复杂度。

2. RBF 方法

2.1 算法说明

该方法要找到满足以下形式的插值函数:

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

其中α是一个二维的系数组，$\mathbf{P_m}$是一个 m 次的多项式函数，文章建议直接用一次函数进行计算效果就很好。

f 是关于两点之间距离的函数，有几种形式，论文中给出了常用的一种形式，如下:

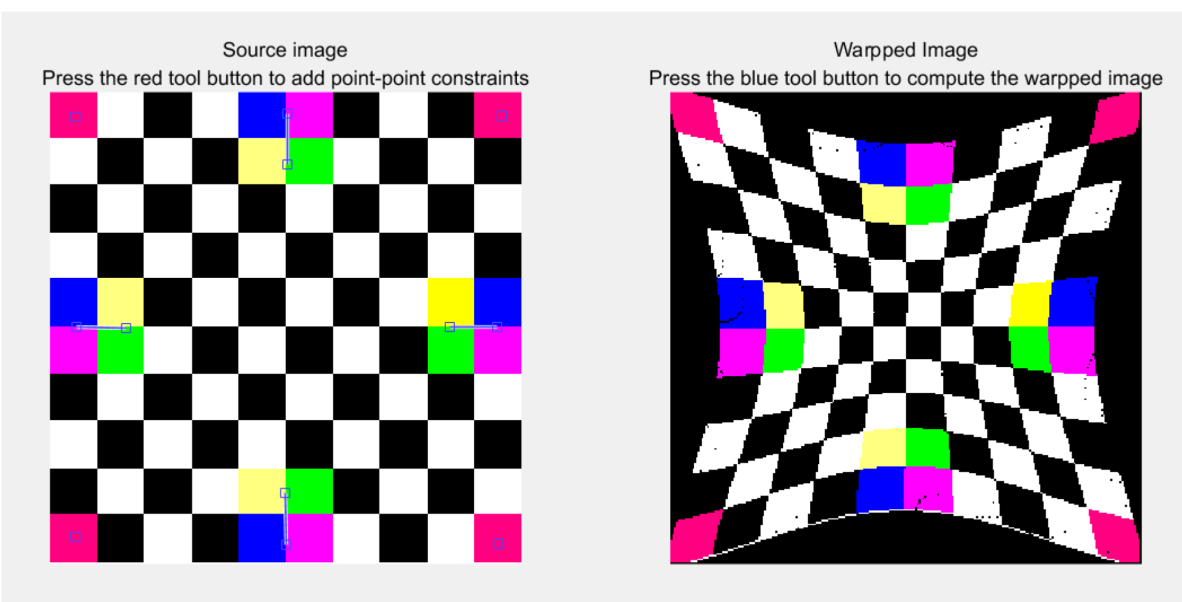
$$f_i(p)=(d^2+r^2)^{\mu}$$

d 是两点之间的距离，每个r_i是距离第 i 个点最近的另一个点与它之间的距离。

而α的确定则需要待定系数法，求解线性方程组来完成。

2.2 算法结果

以下是取μ=1 的情况下，函数的输入输出结果如下:



2.3 算法分析

可以看出，这里出现了部分像素丢失的问题，最主要的原因应该是在图像的拉伸过程中，部分像素点出现了丢失，可能需要采取别的一些方法来补全像素点。

关于算法的时间复杂度，由于对全部像素点进行了一次计算复杂度应该为 O(nN)，还要再加上求解系数组的时间复杂度 O(n^3)，但是这里由于给定的固定点的数量 n<<N，所以不用考虑跟 n 有关的时间复杂度。