Homework 2 Image Warping

华南理工大学 曾亚军

一　实验目的

- 阅读论文并实现 IDW 方法和 RBF 方法。
- 学习 Eigen 库的使用。
- 处理白缝问题。

二　算法的基本过程

IDW 方法和 RBF 方法这两个方法目的都是构造一个插值函数，满足给定控制点的插值需求。给定 n 对控制点 $(\mathbf{p}_i, \mathbf{q_i})$，$\mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^2$，$i = 1, \ldots, n$，得到一个函数 $\mathbf{f} : \mathbb{R}^2 \to \mathbb{R}^2$，满足插值条件，即 $\mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i, i = 1, \ldots, n$。

核心的不同在于构建的插值函数的表达形式。

2.1 IDW 方法

IDW 方法求解局部插值函数 $\mathbf{f}_i(\mathbf{p}) : \mathbb{R}^2 \to \mathbb{R}^2$ 满足 $f_i(\mathbf{p}_i) = \mathbf{q}_i$，具体为：

$$\mathbf{f}_i(\mathbf{p}) = \mathbf{q}_i + \mathbf{D}_i(\mathbf{p} - \mathbf{q}_i)$$

其中 $\mathbf{D}_i : \mathbb{R}^2 \to \mathbb{R}^2$，满足 $\mathbf{D}_i(\mathbf{0}) = \mathbf{0}$。本文中选取 $\mathbf{D}_i$ 为线性变换。

求解的插值函数形式为：

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x})\mathbf{f}_i(\mathbf{x})$$

其中 $w_i : \mathbb{R}^2 \to \mathbb{R}$，为 inverse distance 函数的归一化形式：

$$w_i(\mathbf{x}) = \frac{\sigma_i(\mathbf{x})}{\sum_{j=1}^{n} \sigma_j(\mathbf{x})}$$

$$\sigma_i(\mathbf{x}) = \frac{1}{\|\mathbf{x} - \mathbf{p}_i\|^{\mu}}$$

线性变换 $\mathbf{D}_i$ 可以通过最小化如下能量得到

$$\begin{aligned} E_i(\mathbf{D}_i) &= \sum_{j=1, j\neq i}^{n} \sigma_{ij} \left\| \mathbf{q}_i + \begin{pmatrix} d_{i,11} & d_{i,12} \\ d_{i,21} & d_{i,22} \end{pmatrix} (\mathbf{p}_j - \mathbf{p}_i) - \mathbf{q}_j \right\|^2 \\ &= \sum_{j=1, j\neq i}^{n} \sigma_{ij}((d_{i,11}(p_{j,1} - p_{i,1}) + d_{i,12}(p_{j,2} - p_{i,2}) + q_{i,1} - q_{j,1})^2 + \\ &\quad (d_{i,21}(p_{j,1} - p_{i,1}) + d_{i,22}(p_{j,2} - p_{i,2}) + q_{i,2} - q_{j,2})^2) \end{aligned}$$

通过求导取得极值点满足如下方程：

$$\frac{\partial E_i(\mathbf{D}_i)}{\partial d_{i,11}} = \sum_{j=1,j\neq i}^{n} 2\sigma_{ij}((d_{i,11}(p_{j,1}-p_{i,1})+d_{i,12}(p_{j,2}-p_{i,2})+q_{i,1}-q_{j,1})(p_{j,1}-p_{i,1}) = 0$$

$$\frac{\partial E_i(\mathbf{D}_i)}{\partial d_{i,12}} = \sum_{j=1,j\neq i}^{n} 2\sigma_{ij}((d_{i,11}(p_{j,1}-p_{i,1})+d_{i,12}(p_{j,2}-p_{i,2})+q_{i,1}-q_{j,1})(p_{j,2}-p_{i,2}) = 0$$

$$\frac{\partial E_i(\mathbf{D}_i)}{\partial d_{i,21}} = \sum_{j=1,j\neq i}^{n} 2\sigma_{ij}((d_{i,21}(p_{j,1}-p_{i,1})+d_{i,22}(p_{j,2}-p_{i,2})+q_{i,2}-q_{j,2})(p_{j,1}-p_{i,1}) = 0$$

$$\frac{\partial E_i(\mathbf{D}_i)}{\partial d_{i,22}} = \sum_{j=1,j\neq i}^{n} 2\sigma_{ij}((d_{i,21}(p_{j,1}-p_{i,1})+d_{i,22}(p_{j,2}-p_{i,2})+q_{i,2}-q_{j,2})(p_{j,2}-p_{i,2}) = 0$$

进一步化简可以得到方程：

$$d_{i,11}\sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,1}-p_{i,1})^2 + d_{i,12}\sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,2}-p_{i,2})(p_{j,1}-p_{i,1}) = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(q_{i,1}-q_{j,1})(p_{j,1}-p_{i,1})$$

$$d_{i,11}\sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,1}-p_{i,1})(p_{j,2}-p_{i,2}) + d_{i,12}\sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,2}-p_{i,2})^2 = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(q_{i,1}-q_{j,1})(p_{j,2}-p_{i,2})$$

$$d_{i,21}\sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,1}-p_{i,1})^2 + d_{i,22}\sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,2}-p_{i,2})(p_{j,1}-p_{i,1}) = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(q_{i,2}-q_{j,2})(p_{j,1}-p_{i,1})$$

$$d_{i,21}\sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,1}-p_{i,1})(p_{j,2}-p_{i,2}) + d_{i,22}\sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,2}-p_{i,2})^2 = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(q_{i,2}-q_{j,2})(p_{j,2}-p_{i,2})$$

令

$$A_{i,11} = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,1}-p_{i,1})^2, \qquad A_{i,12} = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,2}-p_{i,2})(p_{j,1}-p_{i,1})$$

$$A_{i,21} = \sum_{j=1,j\neq i}^{n}\sigma_{ij}p_{j,1}-p_{i,1})(p_{j,2}-p_{i,2}), \qquad A_{i,22} = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(p_{j,2}-p_{i,2})(p_{j,2}-p_{i,2})^2$$

$$b_{i,1}^{(1)} = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(q_{i,1}-q_{j,1})(p_{j,1}-p_{i,1}), \quad b_{i,2}^{(1)} = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(q_{i,1}-q_{j,1})(p_{j,2}-p_{i,2})$$

$$b_{i,1}^{(2)} = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(q_{i,2}-q_{j,2})(p_{j,1}-p_{i,1}), \quad b_{i,2}^{(2)} = \sum_{j=1,j\neq i}^{n}\sigma_{ij}(q_{i,2}-q_{j,2})(p_{j,2}-p_{i,2})$$

等价于求解两组 2 维线性方程组：

$$\begin{cases} A_{i,11}d_{i,11} + A_{i,12}d_{i,12} = b_{i,1}^{(1)} \\ A_{i,21}d_{i,11} + A_{i,22}d_{i,12} = b_{i,2}^{(1)} \end{cases}$$

$$\begin{cases} A_{i,11}d_{i,21} + A_{i,12}d_{i,22} = b_{i,1}^{(2)} \\ A_{i,21}d_{i,21} + A_{i,22}d_{i,22} = b_{i,2}^{(2)} \end{cases}$$

IDW 方法的实现过程即对每个锚点求解两个 2 维线性方程组

2.2 RBF 方法

给定 n 对控制点 $(\mathbf{p}_i, \mathbf{q_i})$，$\mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^2$，$i = 1, \ldots, n$，构建插值函数：

$$\boldsymbol{f}(\boldsymbol{p}) = \sum_{i=1}^{n}\boldsymbol{\alpha}_i R(\|\mathbf{p}-\mathbf{p}_i\|) + A\mathbf{p} + \mathbf{b}$$

其中权重系数 $\boldsymbol{\alpha}_i \in \mathbb{R}^2$，$A \in \mathbb{R}^{2\times 2}$，$\mathbf{b} \in \mathbb{R}^2$，径向基函数 $R(d) = (d^2 + r^2)^{\mu/2}$ 。

要求满足插值条件

$$\mathbf{f}(\mathbf{p}_j) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i R(\|\mathbf{p}_j - \mathbf{p}_i\|) + A\mathbf{p}_j + \mathbf{b} = \mathbf{q}_j, \quad j = 1, \ldots, n$$

RBF 方法的核心想法在于将锚点的变换分解成径向基变换和仿射变换。最终得到的插值函数用每个点的径向基变换和仿射变化的线性组合构成。方程未知数个数维 2(n+3) 个。因而需对每维进行补充 3 个约束。本作业中选取的自由约束根据锚点的个数选取。最终的结果即求解两个 (每个未知数都是 2 维)(n+3) 维的线性方程组。

三　类设计

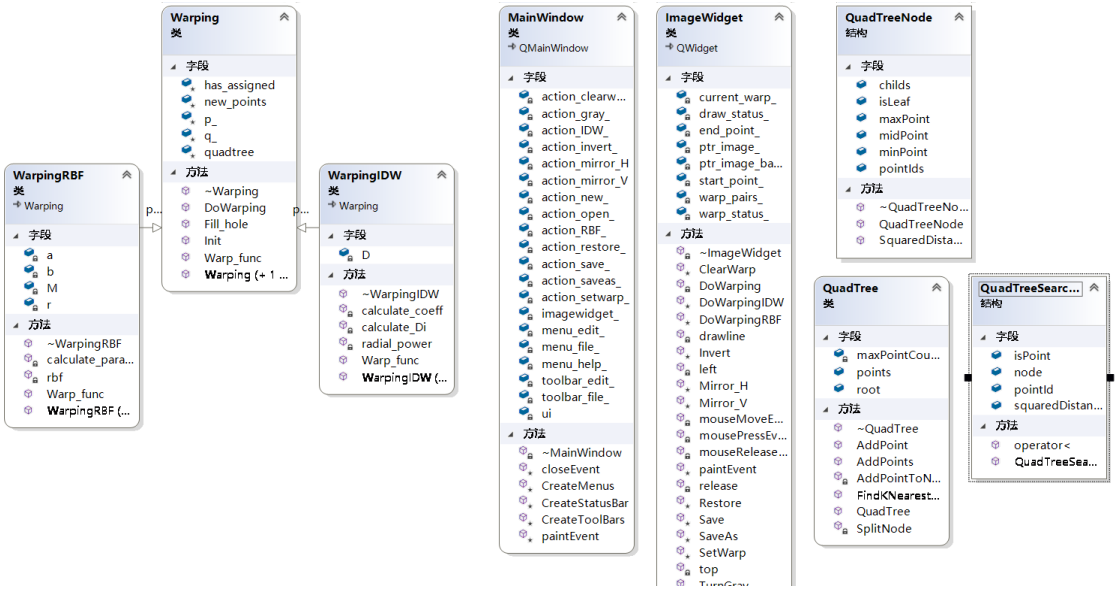


图 1: 类图

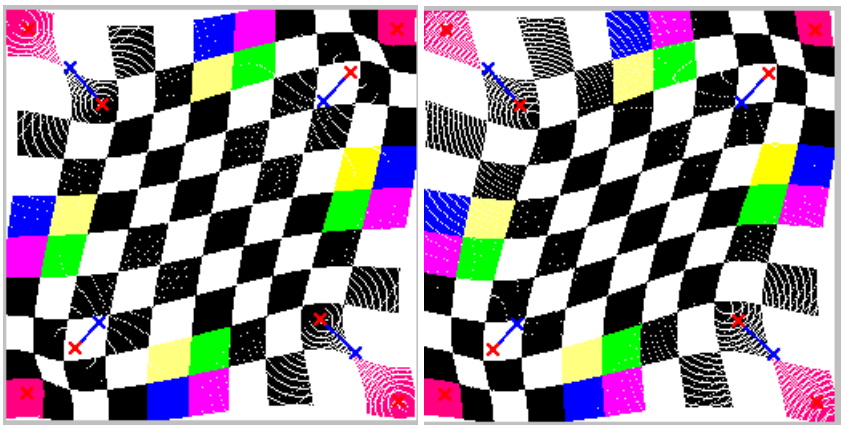
本次作业主要是实现两方面的功能，锚点的用户交互，以及图像变形。而图像变形 (Image Warping) 又分成了两种方法，IDW 法和 RBF 法。因而在本作业中，我构建一个基类 *Warping* 类，在 *Warping* 类中完成了两个方法公有的部分：锚点信息的接受和处理，插值函数等。派生类 *WarpingIDW* 和 *WarpingRBF* 继承自基类 *Warping*，并覆写了插值函数的内容。

由于要实现最近邻算法，框架中的 ANN 我无法运行，因而我在本作业中撰写了一个平面点的四叉树 (QuadTree) 剖分的方法。类图中 QuadTreeNode 为四叉树节点，包括了平面矩形的最小点，最大点，以及中点。QuadTree 是 QuadTreeNode 的树结构，包含最近邻搜索的方法。而 QuadTreeSearchNode 是四叉树寻找最近邻的搜索节点，用于加快搜索速度。

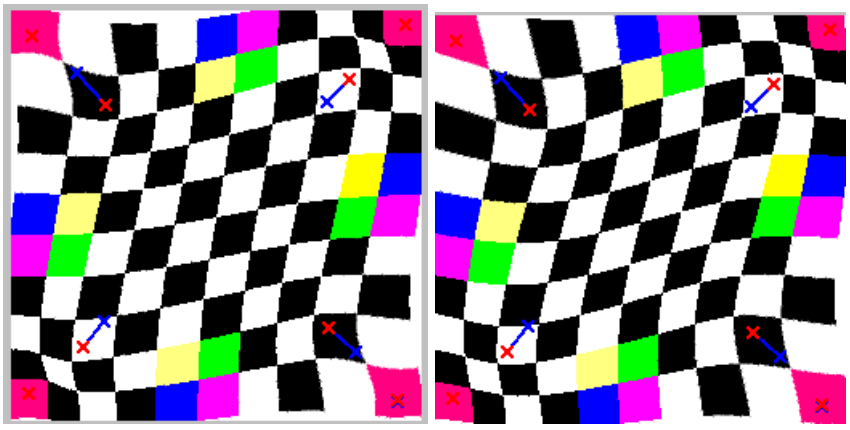
四　实现结果

4.1　补洞算法

在图像变形过程中，由于图像的拉伸和放缩，会导致存在像素点分布不均，出现白色缝隙的问题。对此，在本作业中，我采取了寻找最近邻点进行填补的方式。我一开始运行作业框架时 ANN 部分出错。因而我在 CMakeLists 中删掉了 ANN 部分的内容，使其能正常运行。但由于要实现最近邻算法，我在本作业中撰写了一个平面点的四叉树剖分的算法。其具体效果如下：补洞效果良好。且补洞速度很快。。



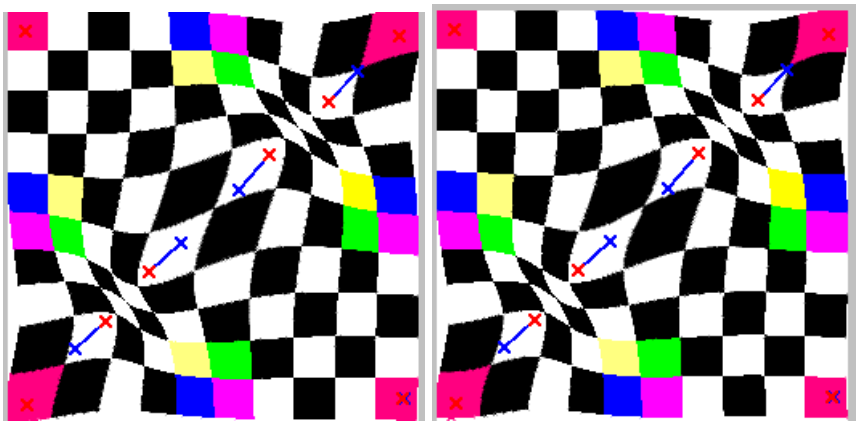
(a) IDW 法　　(b) RBF 法



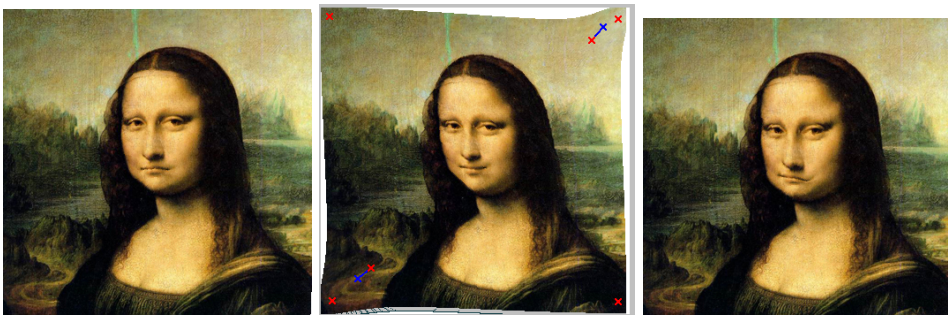
(c) IDW 法 + 补洞　　(d) RBF 法 + 补洞

4.2　方法对比

两种方法都是构成一个插值函数，但具体的形式并不一样。这里展示其图像变形的一些结果。就结果而言，两者的变形结果肉眼看不出太大区别。但从速度上，RBF 法的速度明显要 IDW 要快很多。但从算法本身上看，IDW 法是去求解 n 个 2 维线性方程组，且每个方程的求解互不影响，便于实现并行运算。而 RBF 法是求解 2 个 n+3 维的线性方程组。从求解线性方程组的时间复杂性性而言，RBF 法应当是 $O(n^3)$ 时间复杂性的，而 IDW 法应当是 $O(n)$ 时间复杂性。但 IDW 法求解每个系数需要全局的信息。因而在锚点数目比较少的时候，RBF 法要更快，而锚点很多的情况下，IDW 法可能会更适用。



(e) IDW 法　　(f) RBF 法



(g)　　(h)　　(i)

五　实现难点

- 对 IDW 法，需要耐心地推导公式结果。
- 本作业中最近邻算法是自己用四叉树剖分实现的，设计结构和实现四叉树的最近邻花了一定的时间。
- 对 RBF 法，锚点数目不同决定了仿射变换的形式。

六　收获和感悟

其实插值和逼近算是我大学期间国创接触得比较多的一件事情，径向基函数也接触得不少，包括这次使用的 IDW 用的径向基函数 inverse distance 函数和 RBF 方法用的径向基函数之前也接触过（虽然之前会将这两种径向基函数称为 radial power 和 MQ 方法）。在之前接触的项目研究里面，我了解到的基本上只是这些径向基函数的一些性质，区别，且着重在曲面重建板块。而这次的作业会让我发现，自己在图形学领域学过和了解过的知识，其实可以应用在很多领域很多功能。这也使得我对计算机图形学的相关内容更具好奇心和热情。我认为这一点是非常重要的。

参考文献

Ruprecht D, Muller H. Image warping with scattered data interpolation. IEEE Computer Graphics and Applications, 1995, 15(2): 37-43.

Arad N, Reisfeld D. Image warping using few anchor points and radial functions. Computer graphics forum. Edinburgh, UK: Blackwell Science Ltd, 1995, 14(1): 35-46.