

ImageWarping

SA21010060 周俊亦

Abstract

图像变形是插值算法的重要应用之一。本报告复现了两种经典的插值算法: IDW(Inverse distance-weighted) 和 RBF(Radial basis functions).

1 算法原理

1.1 概述

将图像看成二次函数, 就像在三维平面上波澜起伏的山峰一般, 可以对图像进行非常有趣的操作, 例如图像变形就是应用之一。借助傅里叶变换¹的思想, 通过离散点拟合, 可以将图像分解成一系列基函数 (其中高斯基函数理论上可以拟合任意复杂的函数)。通过这些基函数, 我们可以利用插值算法将图像重新“塑形”。

- A function (height field) over 2D domain

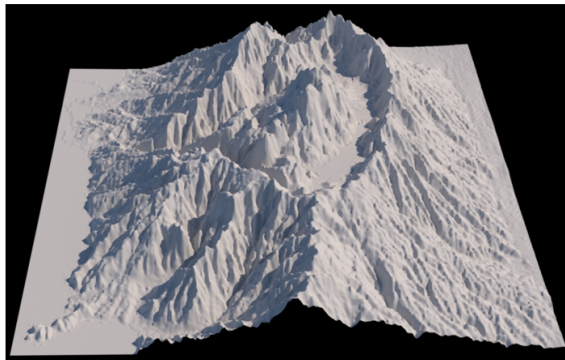


图 1: 另一角度的灰度图

1.2 原理

算法的输入输出描述为:

Input: 给定 n 对控制点对 $(\mathbf{p}_i, \mathbf{q}_i)$, 其中 $\mathbf{p}_i(x_{p_i}, y_{p_i}), \mathbf{q}_i(x_{q_i}, y_{q_i}), \mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^2, i =$

¹写作业时的胡思乱想: 傅里叶变换真是神奇而又伟大的变换。万物皆可分解, 万物本就同源。旋转的 \sin 和 \cos , 就像精密的齿轮啮合, 滚动走出正弦轨迹, 再复杂的事物, 都逃不过拟合, 看似复杂的命运, 背后是不是也有一根牵引的线呢?

$1, 2, \dots, n$.

Output: (An at-least-continues function) $\mathbf{f}: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, 使得 $\mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i, i = 1, 2, \dots, n$

1.3 IDW 算法

设

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

其中, $f_i(\mathbf{p})$ 称为 \mathbf{p} 的局部近似函数, 也即基函数, ω_i 为基函数权重.

$f_i(\mathbf{p})$ (\mathbf{p} 为平面上任一点) 满足:

$$f_i(\mathbf{p}_i) = \mathbf{q}_i$$

ω_i 满足:

$$\begin{cases} \sum_{i=1}^n \omega_i(\mathbf{p}) = 1 & \forall \mathbf{p} \in \mathbb{R}^2 \\ \omega_i(\mathbf{p}_i) = 1 \\ \omega_i(\mathbf{p}) \geq 0, i = 1, 2, \dots, n \end{cases}$$

取:

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

其中 (μ 可取不同值)

$$\sigma_i(\mathbf{p}) = \frac{1}{d(\mathbf{p}, \mathbf{p}_i)^\mu}$$

可以猜测, 当 μ 越大时, 给定配对点的影响区域将越大.

取线性基函数 $f_i(\mathbf{p})$:

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

误差函数

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

求 $\mathbf{T}_i, i = 1, 2, \dots, n$ 使误差函数最小.

1.4 RBF 算法

RBF 算法给出坐标变换

$$f(\mathbf{p}) = \sum_{i=1}^n \alpha_i g_i(d(\mathbf{p}, \mathbf{p}_i)) + M(\mathbf{p}) + b$$

径向基函数 g_i (μ 可取不同值) 取

$$g_i(d) = (d^2 + r_i^2)^{\mu/2}, \quad r_i = \min_{j \neq i} \{d(p_i, p_j)\}$$

系数 $\alpha_i, i = 1, 2, \dots, n$ 可由

$$\sum_{j=1}^n \alpha_j g_i(d(\mathbf{p}_i, \mathbf{p}_j)) = \mathbf{q}_i - (M\mathbf{p}_i + b) \quad (i = 1, 2, \dots, n)$$

解得. 这里取

$$M(\mathbf{p}) + b = \mathbf{p}$$

为恒等变换.

2 实验分析

实验使用 Qt5 库 进行算法验证, 执行 Cmake, 使用 MSVC2017 构建生成。

2.1 程序界面

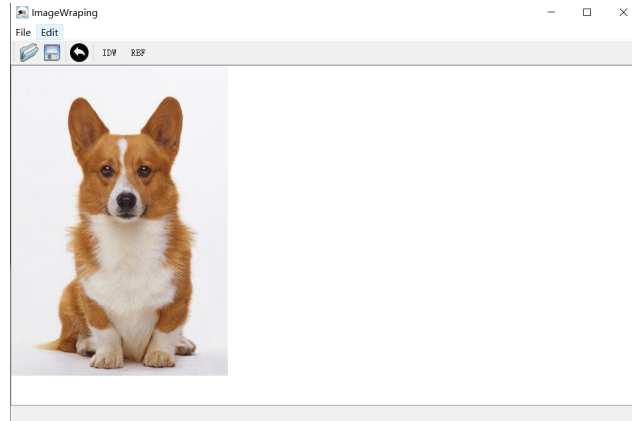


图 2: 程序界面

2.2 四边固定，四角内拉

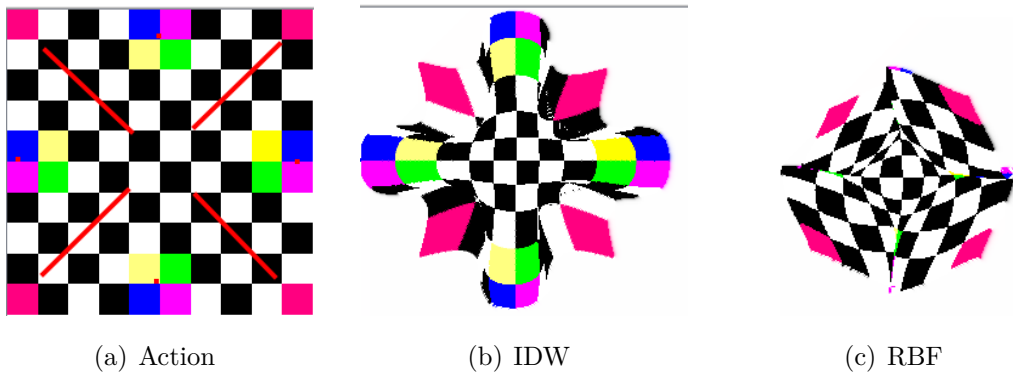


图 3: 四边固定，四角内拉

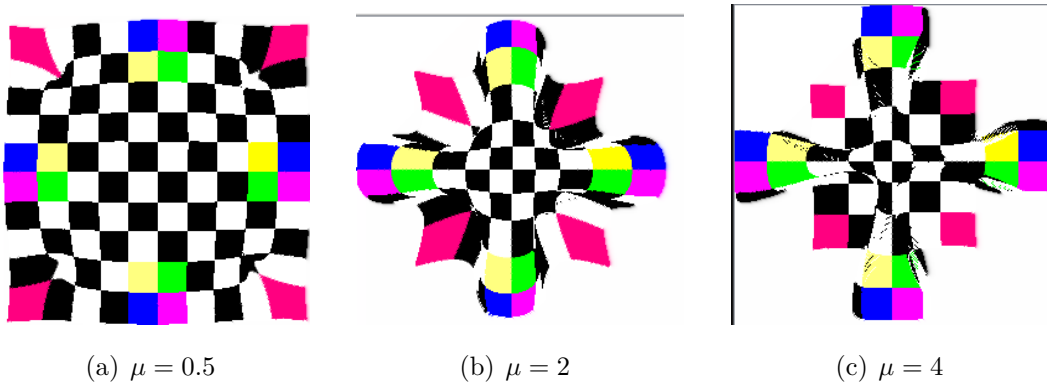


图 4: IDW 不同 μ

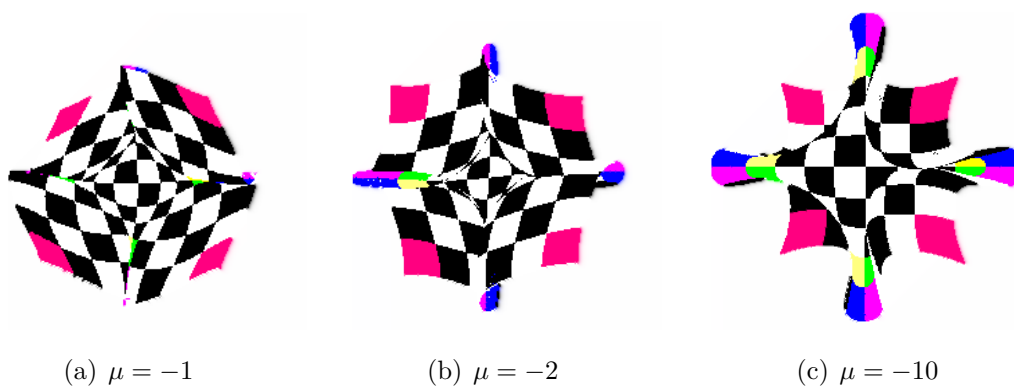


图 5: RBF 不同 μ

可以见到，在 IDW 算法中， μ 度量了给定配对点对其他点的影响力，当 μ 越大，控制点能影响到的范围就越大；在 RBF 算法中同样如此。实验结果印证了理论分析。

2.3 四角固定，四边内拉

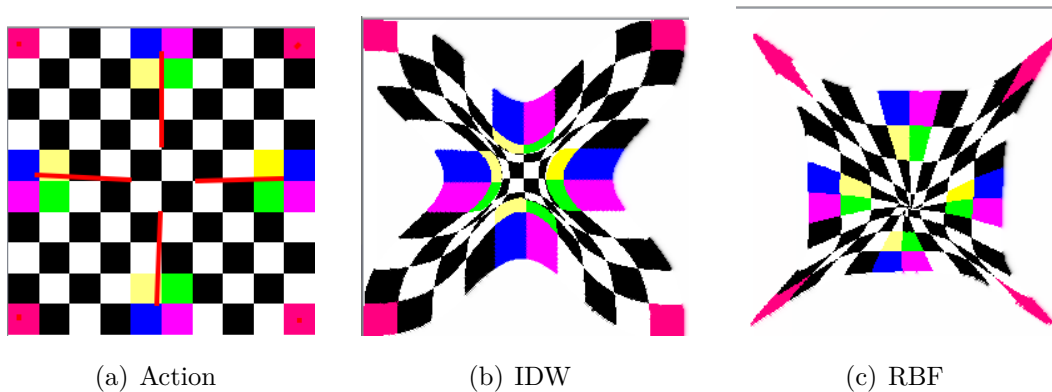
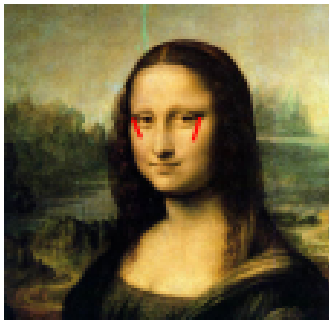
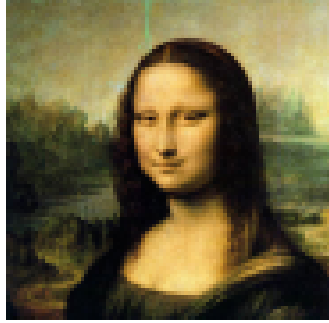


图 6: 四角固定，四边内拉

2.4 Interesting Images



(a) Action

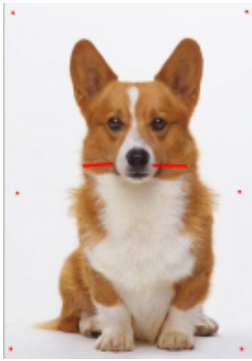


(b) source

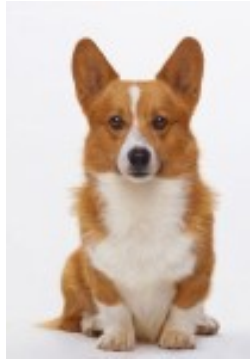


(c) result

图 7: Ex1.



(a) Action



(b) source



(c) result

图 8: Ex2.

3 实验总结

本次实验复现了图像变形的两种经典算法，直观验证了算法中参数对实际应用的作用，并学习了 Qt 的使用。

附睿客网代码链接：<https://rec.ustc.edu.cn/share/5a0b54b0-2f47-11ec-90e7-f14835723dfc>

参考文献

- [1] Image Warping Using few Anchor Points and Radial Functions:Nur Arad and Daniel Reisfeld.Computer Graphics Forum,1995.
- [2] Image warping with scattered data interpolation:D. Ruprecht and H. Muller, IEEE Computer Graphics and Applications, vol. 15, no. 2, pp. 37-43, March 1995, doi: 10.1109/38.365004.