Reproduction of Possion Image Edition

陈芳达 121090029

May 2022

1　论文研究

本次问题的解决方法主要来源于 P. P´erez, M. Gangnet, and A. Blake. (2003) 的 Poisson image editing.

1.1　离散泊松解

论文中提到的基本理念是将像素 (r, g, b) 看做关于位置 (x, y) 的二元函数，并认为凸显图像特征的是这个函数的梯度（我的理解是，像素之间的相对距离指示了图片是什么东西，而非颜色本身）。本文直接采用与原论文相同的记号，将 R、G、B 通道分开计算，记前景图像的函数为 $g : \Omega \to R$，变换后的前景为 $f : \Omega \to R$，后景为 $f^* : S \to R$，边界为 $\partial\Omega$，那么最终的目的是使原图像与变换后的图像梯度尽可能一致，并且边界要与背景相同，即

$$\min_f \iint_\Omega |\nabla f - \nabla g|^2 \text{ with } f\big|_{\partial\Omega} = f^*\big|_{\partial\Omega} \tag{1}$$

这个方程的解是

$$\Delta(f - g) = \nabla \cdot (\nabla f - \nabla g) = 0 \text{ with } f\big|_{\partial\Omega} = f^*\big|_{\partial\Omega} \tag{2}$$

抛开证明不谈，其中的数学思想是较易理解的。我的直观理解是，若场 $\nabla f - \nabla g$ 的散度不为 0，则曲面 $f - g$ 中必出现“凸点”或“凹点”，这比起更平滑的曲面，其梯度的平方和无疑是更大的。虽然还不太能说明为什么它能达到最小值，但令梯度为 0 无疑是一种不错的做法。

接下来就是把上述连续函数的性质应用到离散函数上。若仅从上述数学过程来看，那么合理的流程就是

1. 对于原前景图 g 边界内的每一个位置 p，用 N_p 表示与 p 相邻的点，计算 guidance field g 的散度
$$\nabla \mathbf{v} = \Delta g = \sum_{\mathbf{x} \in N_p} (g(\mathbf{x}) - g(p)) \tag{3}$$
2. 对于相同位置的 f，
$$\Delta f = -4f(p) + \sum_{\mathbf{x} \in (N_p \cap \partial\Omega)} f^*(\mathbf{x}) + \sum_{\mathbf{x} \in N_p / \partial\Omega} f(\mathbf{x}) \tag{4}$$
3. 最后列出 $\Delta f = \Delta g$ 的方程即可。

1.2　梯度混合

论文中还提到了梯度混合的方法，可以让后景的纹理也显示在前景，以免在前景边缘颜色平滑时产生模糊带。核心观念是，前景和后景谁的纹路更明显（梯度更大）就用谁的。按

照原论文，这时 $\nabla\mathbf{v}$ 的计算公式改变为

$$\nabla\cdot\mathbf{v}\big|_p=\sum_{\mathbf{x}\in N_p}\begin{cases}g(\mathbf{x})-g(p),\ |g(\mathbf{x})-g(p)|>|f^*(\mathbf{x})-f(p)|\\ f^*(\mathbf{x})-f^*(p),\ |g(\mathbf{x})-g(p)|<|f^*(\mathbf{x})-f(p)|\end{cases}\tag{5}$$

这就带来了问题。根据这个公式，只是把上下左右四个方向上替换成差较大的那一组，但这单个方向并不能表示这个点整体的梯度。如果要比较梯度大小的话，应该先计算

$$|\nabla g|^2=(g_{left}-g_{right})^2+(g_{up}-g_{down})^2\tag{6}$$

再根据此值选择使用 f^* 的梯度还是 g 的梯度来计算散度，即

$$\nabla\cdot\mathbf{v}=\begin{cases}\Delta g,\ |\nabla g|^2>|\nabla f|^2\\ \Delta f,\ |\nabla g|^2<|\nabla f|^2\end{cases}\tag{7}$$

我也试过这样算，最后得到的图片虽然主体部分保持得更加完整，但部分区域过亮。原因未知。有可能是因为过度使用更大的梯度造成颜色值趋向于 255。

(a) 使用 (3)（原论文）所示方法

(b) 使用 (4) 所示方法

(c) 使用 (3)（原论文）所示方法

(d) 使用 (4) 所示方法

图 1: 使用不同梯度大小判定方法的的梯度混合

2 python 实现

根据论文按部就班地完成就行了。latex 是在不适合编辑代码，故用 jupyter 代替。详见笔记本。

3 讨论

根据原论文，若使用普通的泊松图像编辑，则可能出现模糊边缘；若使用梯度混合，可能导致前景图的主题部分被染上后景的纹理。所以也许可以通过前景的颜色和边界平均颜色的色差来判断该使用谁的梯度，如

$$\mathbf{v} = \nabla g \cdot \frac{|g - f^*|}{3 \times 256} + \nabla f^* \cdot \left(1 - \frac{|g - f^*|}{3 \times 256}\right) \tag{8}$$

不过实际尝试过程中并没有得到很好的效果。

此外，不知为什么使用 (5) 会比 (7) 效果更好。

4 感想

从开始看论文到初步写完代码历时 6 小时。起初以为复刻研究是很按部就班的工作，但真正做完后还是发现了很多新问题，收获许多。

最令我惊讶的是，这个项目带给我关于“图像”的全新认知。

我最开始认识“图像”是高中时对 CNN 感兴趣时研究手写数字识别的算法。当时，我看到算法将 28*28 的图片当做 784*1 的向量进行计算，起初感到困惑，但随即一想，将像素按顺序重新排布并没有损失信息，把单纯地看成像素的序列也无可厚非。随着研究的深入，我心理又产生了一个疑问：如果假设图像与数字的对应关系是线性的，这是否太违背常理。之后我又见识了卷积神经网络，初识将“每个像素与它周围的像素的关系”作为信息来提取的算法。但当时我依旧不理解：使用只考察 9 个像素的卷积核岂不是以点概面、以偏概全？为什么将点阵分组提取数据能达到理想的结果？这是我一直以来的疑惑。

带着这个疑惑来看本次的 project，我发现这种“把图像看做关于坐标的二元函数”的观点回答了这个问题。决定图像本质的并不是像素本身，而在于每个像素与别的像素是何种关联。也就是说，图片的梯度场就是一种包含大量信息的研究对象。此时我对图像的认知已不再是简单的“像素序列”。

此外，对于刚上完 MAT1002 的我，这次 project 也刷新了我对多元微积分应用的认知。起初我以为多元微积分和向量场等仅仅是物理中引力、电磁学的工具，现在才发现其在计算机科学领域也有应用。这也给了我继续学习数学的动力。

期待下一个 project！