数学建模
第 4 次实验
Poisson Blending

姓名 徐海阳　　学号 PB20000326　　院系 少年班学院

2023 年 5 月 16 日

摘　要

本文介绍了图像融合领域中的重要方法，Poisson Blending (泊松融合)。本文将依次介绍相关工作、问题分析、建模假设、符号说明以及实验结果，以全面阐述 Poisson Blending 算法的原理和性能。

1 前言

本文介绍了一种在图像处理领域中备受瞩目的算法，即 Poisson Blending（泊松融合）。自从 2003 年被发表 [1] 以来，该算法已经成为图像融合领域中的重要方法，并得到了许多基于其改进的研究。图像融合是指将不同图像的不同部分合并在一起，形成一幅新的图像。融合结果越自然，算法的质量就越高。

本文将依次介绍相关工作、问题分析、建模假设、符号说明以及实验结果，以全面阐述 Poisson Blending 算法的原理和性能。

2 相关工作及问题分析

在图像处理领域，图像融合是一个广泛研究的话题。在 Poisson Blending 算法之前，已经有一些相关的工作被提出和探索。

在融合领域中，最简单的算法之一是使用 alpha blending（透明度混合），即在融合边界处将两幅图像的透明度进行线性相加，生成新的图像。一个著名的例子是苹果橘子图，如下图1所示。

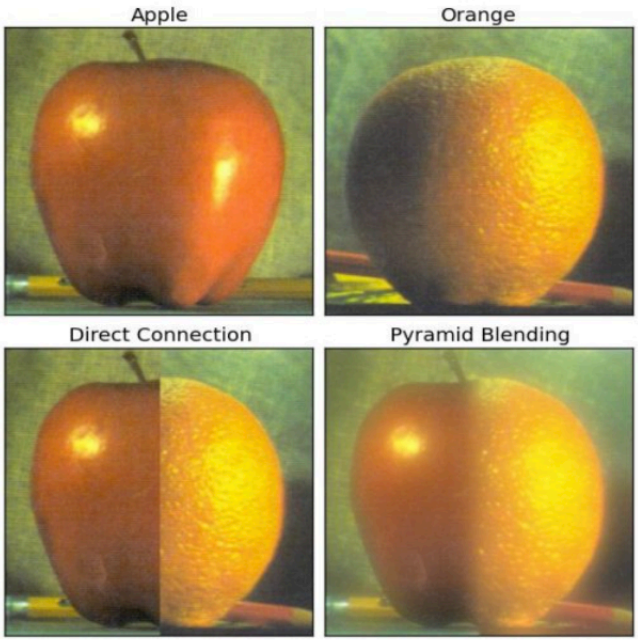

图 1: 苹果橘子图

然而，这种算法存在一个缺点：当两幅图像的颜色并不十分相近时，融合结果往往不理想。

为了改善融合效果，研究人员提出了拉普拉斯金字塔融合算法。该算法基于图像的高斯金字塔，通过逐层的 alpha blending 操作实现不同频率的融合。这种方法在一定程度上提高了融合的质量，但仍然存在边界模糊和颜色不匹配的问题。

Poisson Blending 算法是在此背景下提出的一种改进方法。它通过利用离散泊松方程来实现源图像与目标图像之间的精确融合。具体而言，通过将源图像的特定区域与目标图像的对应区域在边界上进行匹配，并将泊松方程应用于边界上的像素，使得融合结果在边界处具有一致性。这种方法在处理颜色不匹配和边界连续性方面取得了显著的改进。

此外，还有一些其他的图像融合算法被提出，如多分辨率融合、基于梯度域的融合等。这些方法各有优缺点，并在特定场景下具有一定的适用性。

综上所述，Poisson Blending 算法作为一种精确且能够保持边界一致性的图像融合方法，在图像处理领域引起了广泛关注。接下来的章节中，我们将详细介绍该算法的原理、建模假设以及实验结果，以验证其在不同场景下的性能和优越性。

3 建模假设

在使用 Poisson Blending 算法进行图像融合时，我们做出以下建模假设：

1. 源图像和目标图像已经经过对齐，即两幅图像的边界和结构已经对齐。

2. 融合区域已经被正确地标记，并且源图像的融合区域与目标图像的对应区域在边界上是一致的。

3. 假设图像的边界是封闭的，即融合区域的边界像素与图像内部的像素没有差别。

4 符号说明及算法原理

在本文中，我们使用以下符号来描述 Poisson Blending 算法：

- f：目标图像（foreground image）
- b：源图像（background image）
- Ω：融合区域（blending region）
- $\partial\Omega$：融合区域的边界（boundary of blending region）
- u：融合结果图像（blended image）
- ∇：梯度算子（gradient operator）
- Δ：拉普拉斯算子（Laplacian operator）

以下是 Poisson Blending 的数学原理及推导：

4.1 图像的梯度

什么是图像的梯度？我们可以把图像看成是一个离散的函数, 函数在某一点的梯度是它的导数, 那么根据导数的定义, 我们可以把图像在某一点的梯度定义为：下一个点的像素值-这个点的像素值, 即 $g(i) = f(i+1) - f(i)$ 。至于下一个像素点在那个位置, 要根据方向来定。例如行梯度就是右边的相邻点 $(g(x, y) = f(x+1, y) - f(x, y))$, 列梯度就是下方的相邻点 $(g(x, y) = f(x, y+1) - f(x, y))$ 。那么图像的梯度表示什么意思呢？很明显, 就是相邻两个像素的差异大小。

4.2 泊松方程

柏松方程的表示为

$$\Delta\varphi = f$$

它在物理上有很多应用, 例如表示电荷分布与电势的关系, 或者重力加速度和密度的关系等等。而泊松融合的思想并不是让两幅图叠加，而是让目标图像在融合部分“生长”出源图像。也就是说, 只提供原图像的斜率, 让目标图像根据自己图像的特点, 按照对应斜率生成融合部分。由于是按照自己的特点, 并没有添加外来的元素，生成的图片就更加自然。

4.3 泊松融合

在泊松融合中, 我们真正要处理的是这样一个函数

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

表达的意思是：如果要把图像 B 融合在图像 A 上, 那么表示融合的结果图像 C （也就是我们要求的图像）， f * 表示目标图像 A，V 表示源图像 B 的梯度，∇f 表示的一阶梯度（也就是图像的梯度），Ω 表示要融合的区域，dΩ 代表融合区域的边缘部分。那么这个式子可以解释为：在目标图像 A 边缘不变的情况下, 求融合部分的图像 C, 使得 C 在融合部分的梯度与源图像 B 在融合部分的梯度最为接近（也就是生长趋势最接近）。这里有人可能会有疑问: 如果梯度最接近的话, 直接复制过去不就好了么？我们要注意到, 这里有一个条件: C 的融合边缘与 A 的融合边缘相等。如果直接把 B 复制过去, 内部的梯度是与 B 相等了, 但是边缘就会有问题, 这个结果往往并不是上式的最小值。所以我们从边缘出发, 要求得一个生成图, 满足边缘不会突变, 而且融合部分的变化趋势几乎符合源图像的变化趋势, 这样的结果就是我们想要的。

4.4 解泊松方程

在弄清楚整个算法之后, 我们就可以开始求最优函数了。但如何求一个函数的最小值呢？这里, 我们继续考虑函数的梯度问题。

对于一个连续函数 f, 它的最大/小值点的梯度值一定为 0——这应该是众所周知。我们就以此作为切入点, 用梯度为 0 来作为最优函数的条件。对于这个函数来讲, 由于距离可以无限远, 但最小只能为 0 , 因此它拥有最小值, 而没有最大值, 所以梯度为 0 的点必定是最小值点。这个函数梯度为 0 的表示可以转化为我们刚刚讲到的泊松方程:

$$\Delta f = \operatorname{div} \mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

再解释一下这个等式：Δ 表示拉普拉斯算子，也就是二阶梯度（一阶梯度 ∇f 再求一次梯度）。div(Divergence, 也就是 ∇) 表示 v 的梯度, 也就是 ∇B 的梯度（B 是源图像）。可以理解为: 源图像 B 的二次梯度与新图像 C 的二次梯度相同。

有了这个等式, 我们就有了可以求解目标图像。但是由于图像是离散的, 我们需要把上面的等式变换一下, 符合图像的特点。首先根据梯度的定义, 我们先明确下二阶梯度的表示：

$$\begin{aligned}
&\Delta f = \operatorname{div}(\mathbf{v}) \\
&(f_{i+1,j} + f_{i-1,j} - 2f_{i,j}) + (f_{i,j+1} + f_{i,j-1} - 2f_{i,j}) = \operatorname{div}(\mathbf{v})_{i,j} = b_{i,j}
\end{aligned}$$

一个点上图像梯度的简单定义: 该像素与周围像素的差值的和 (对于不在边缘的像素点, 是与四个周围像素的差值)。因此，我们就可以把上方的等式转换为下面的差分方程 (总结了每一个 Ω 内部像素点的所有情况):

$$|N|H(x,y) - \sum_{(dx,dy)+(x,y)\in\Omega} H(x+dx, y+dy) - \sum_{((dx,dy)+(x,y)\in\partial\Omega} A(x+dx, y+dy) \tag{1}$$

$$= \sum_{(dx,dy)+(x,y)\in(\Omega\cup\partial\Omega)} (B(x+dx, y+dy) - B(x,y)) \tag{2}$$

如果 H 表示新的图像, 一个相邻像素是 (1) 边界像素, 那么它的值是固定的 (因为我们要保留边界值)；(2) 超出选区边界, 被排除。N 表示 $H(x,y)$ 邻居像素点数目（ $<= 4$, 如果此点位于图像边

界那么他可能只有 2 个或 3 个邻居像素), 而左边公式的最后一项则表示如果该像素点的邻居是选区边界的话, 那么就直接用源图像 A 中对应点的值。右边表示目标图像 B 在此点的二阶梯度。

如此一来，这里如果我们把求二阶梯度（也就是式子左边）看成一个 $n * n$ （n 表示要填充的像素总数）矩阵 A, 把未知区域的 H 看做是未知数 x ，把已知的源图像 B 的二阶梯度（也就是式子的右边）看做是一个 n 列矩阵 B ，那么问题的本质就是求解 $Ax = B$ 的问题，这是一个优化问题，可以利用各种数值算法求解。

5 结果

基于上述公式写代码，调用 OpenCV 库进行编写。

如图 2所示，是山路上的小猫和砖墙上的粉笔字。山路上的小猫还好，砖墙上的粉笔字明显有问题，发现字把墙壁给“模糊”了，也就是说插入的时候破坏了墙壁的纹理。

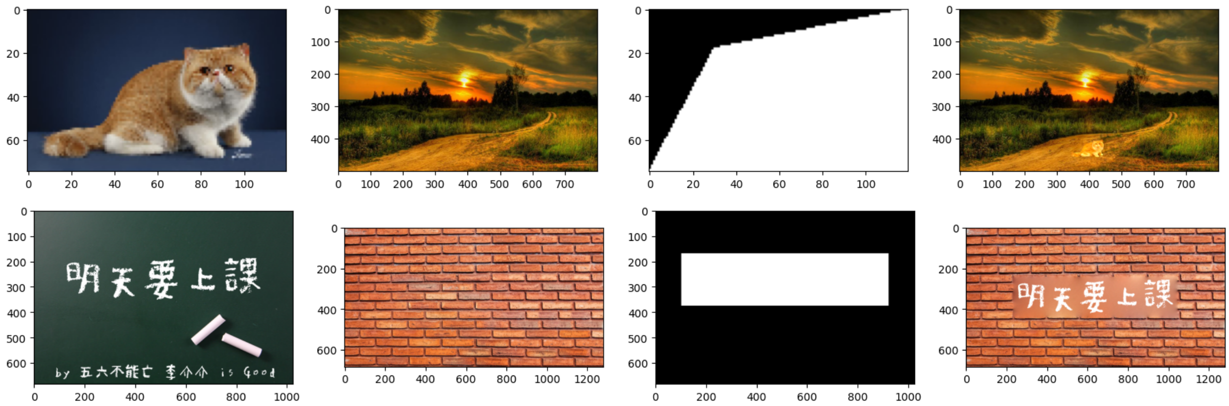

图 2: 山路小猫、砖墙粉笔字

这个时候怎么办呢？实际上，我们只要对上面公式的参数做小小的改动，就可以得到恢复纹理的结果。来想一下，纹理在图像的梯度层面表示了什么含义？纹理越多，对应的梯度图像中边缘也就越多，也就是说，图像中梯度不为 0 的地方也就越多。那么我们只需要对原式中的 V 做一个小小的改动，原先 V 表示为源图像的梯度图像，而现在我们把它表示为源图像梯度图像与目标图像梯度图像对应像素绝对值的最大值即可。这样一来，得到下图 3：

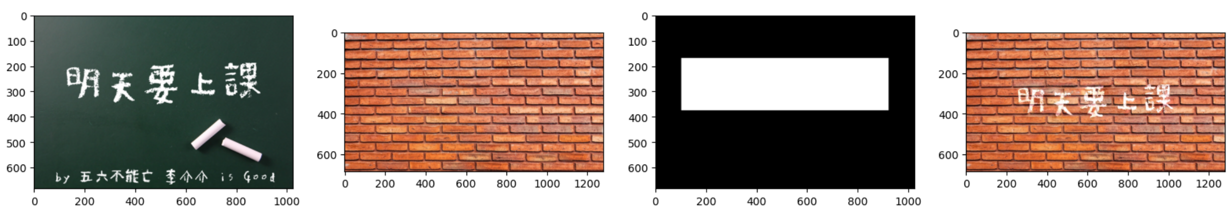

图 3: 砖墙粉笔字（混合 Poisson）

可以看到，纹理恢复得非常好。

以下是三个其他的例子 4：

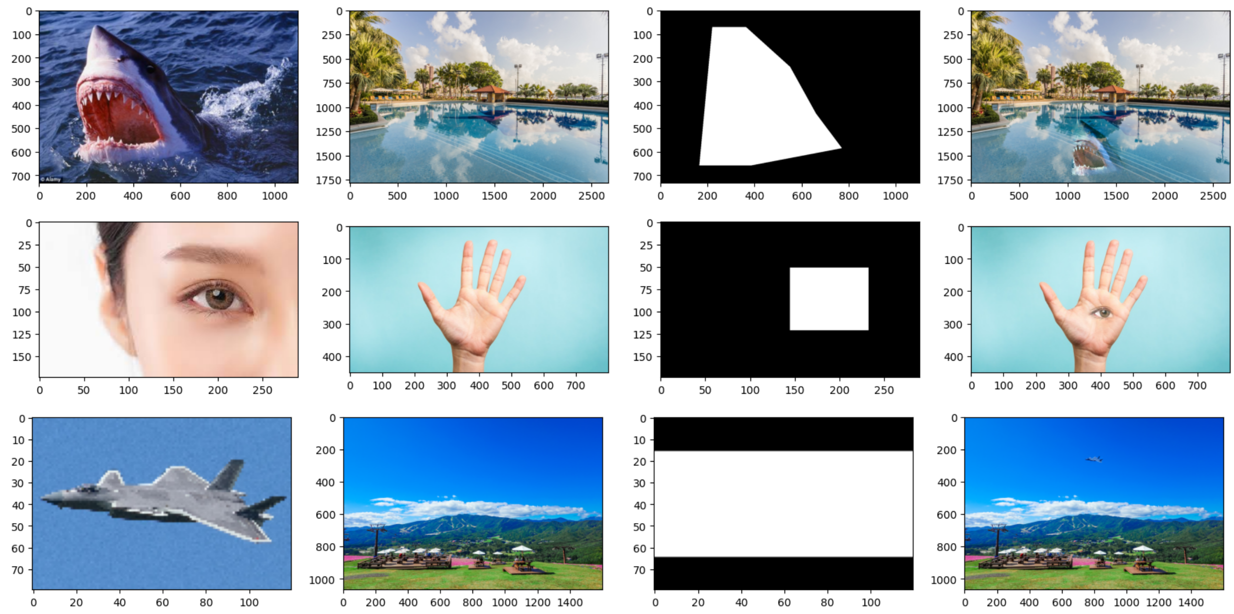

图 4: 游泳池鲨鱼、手上眼睛、山谷战斗机（混合 Poisson）

6 总结和一些想法

如果让两个图像在低频部分泊松融合，高频部分直接保留原图（高频代表细节、纹理，不包括颜色信息），这样的话，图像质量会不会更高呢?

上网搜索了一下，是可行的，质量确实会更高，符合我高低频特征，目前 PS 也是用的这种方法（不包括 learning-based 方法）。

参考文献

[1] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” in *ACM SIGGRAPH 2003 Papers*, pp. 313–318, 2003.