# Poisson Editing

陈雨竹 PB19000160

2022 年 3 月 15 日

## 1 问题描述

在 MATLAB 上实现基于泊松方程的图像融合，并实现实时用户交互，得到类似图 1的界面。

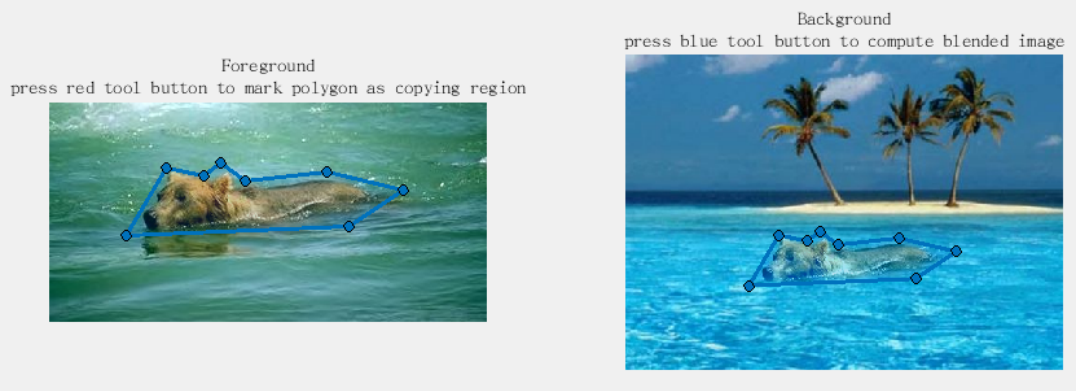


图 1: Poisson Editing Result

## 2 实现方法

编程环境为 Ubuntu 20.04.3 LTS MATLAB R2021a (9.10.0.1602886)。

本文使用两种基于泊松方程的方法实现图像融合，其核心问题为生成一个颜色与背景图片融合，轮廓又包含前景图片的图像。其中使得颜色和背景融合的方法是使得融合边界采用背景图像的像素，使得轮廓包含前景图片的方法是使得融合内部的梯度尽可能靠近前景图像。

若将图像看做一个函数 $f:(x,y)\to(R,G,B)$，并设前景图像为 $f_f$，背景图像为 $f_b$，目标生成的图像为 $f$，融合区域为 $\Omega$，则梯度尽可能接近前景图像可被描述为

$$\min_f \int_\Omega ||\nabla f - \nabla f_f||^2 dx$$

融合边界采用背景图像的像素可被描述为

$$f|_{\partial\Omega} = f_b|_{\partial\Omega}$$

从而化为了一个优化问题。

设 $g = f - f_f$，优化问题化为

$$\min_g \int_\Omega ||\nabla g||^2 dx$$
$$s.t. \quad g|_{\partial\Omega} = (f_b - f_f)|_{\partial\Omega}$$

由 Euler-Lagrange 方程，这样的 $g$ 需要满足泊松方程 $\nabla^2 g = 0$，这可以离散化为每一点的像素等于周围四点的像素平均。从而可以分为以下三步进行图像处理:

- 计算背景图像和前景图像的差在边界的取值
- 根据边界条件，解离散化的泊松方程
- 把方程的解加上前景图像，嵌入背景图像对应位置

在具体实现过程中，单纯的使用上述方法由于计算复杂度大，无法实现实时用户交互，其中计算过程中主要的时间消耗是列方程和解方程的过程。由于不融合的地方像素值不变，所以可以只对融合区域的外接矩形列方程，相比之下这样可以使得方程规模降低，而且在拖动目标融合位置时方程的系数是不变的。

由于方程的系数不变，且可化为少量带状矩阵的和，生成矩阵时可以避免使用循环从而加速。由于拖动目标位置时方程系数矩阵不变，可以在选中区域时生成，这样可以避免每次都要列方程。为了加速解方程的过程，本文在生成方程稀疏矩阵之后对其进行 LU 分解，经过测试，如表 1，这样确实可以起到加速的效果。在 1中，这样做可以实现实时用户交互。

| | 直接解 | LU 分解 | LU 方法解 |
|---|---|---|---|
| 时间 (s) | 0.045158 | 0.094595 | 0.007478 |

表 1: 时间消耗对比

使用以上方法，计算速度有所提升，已经可以实现实时用户交互。

## 3 实验结果

使用上述方法，本文的一个实现结果如图 1，符合预想的融合结果。此外，这种方法可以进行换脸，如图 2为把普京的脸换到徐峥上。

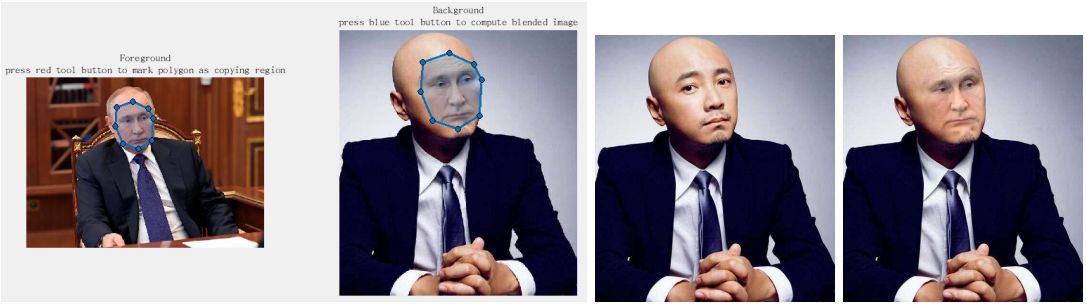


图 2: 换脸

然而在融合区域的背景图像也有轮廓时，使用上述融合方法会导致背景图像轮廓丧失，如图3中，狗的轮廓完全覆盖住了树的轮廓导致树的轮廓消失了。


图 3: $k = 0$ Result

一个解决方案是引入比例系数 $k$，使得目标图像梯度尽可能接近 $f_f + kf_b$，其中 $k = 0$ 时为原方法。当 $k = 1$ 时，处理的结果如图 4，可以看出树的轮廓就得到了保持。

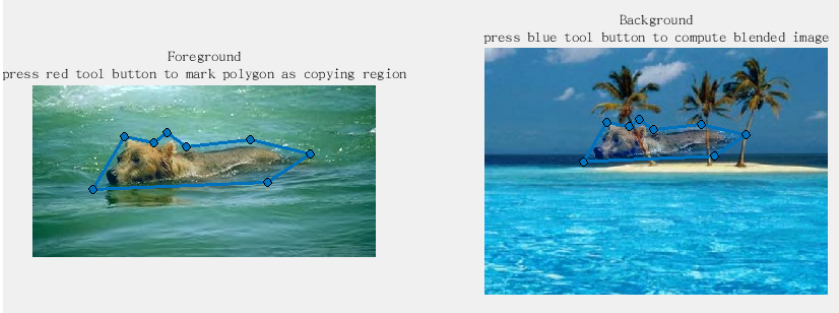


图 4: $k = 1$ Result

这样做在其它一些情况会有较好的效果，如把文字融合到有轮廓的背景上，如图 5为把白纸的字融合到木板上，可以看出融合部分仍有木板的痕迹。

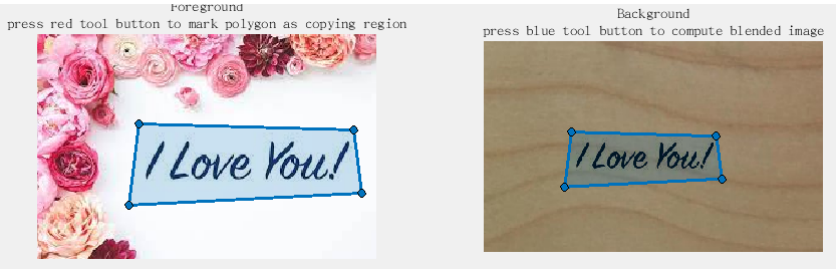


图 5: 文字融合

# 4 参考文献

[1]Patrick Pérez, Michel Gangnet, Andrew Blake. Poisson image editing. Siggraph 2003.