Poisson Image Editing

陈柯

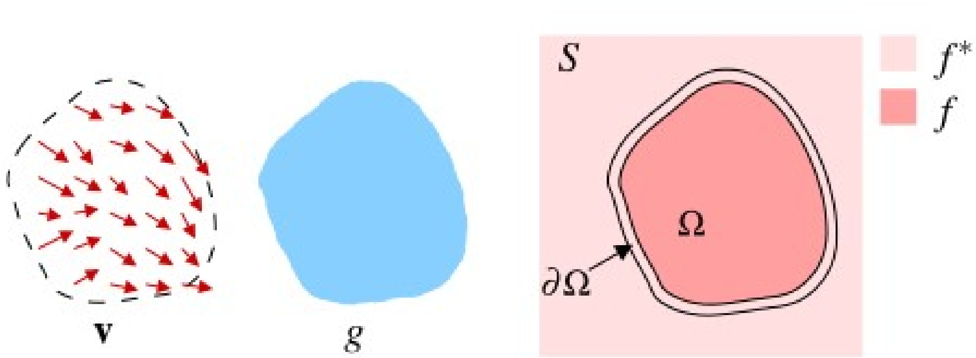
2022 年 8 月 8 日

1 目的

1. 实现 Siggraph 2003 论文“Poisson Image Editing”的算法
2. 实现多边形扫描转换算法。
3. 学习使用图像库 OpenCV。

2 算法原理

2.1 Poisson solution to guided interpolation



如图，S 为 $\mathbb{R}^2$ 中的一个闭集，是我们研究的图像的定义域。Ω 为 S 的一个闭子集，它有边界 $\partial\Omega$。f^* 是已知的定义在 $S\backslash\Omega^\circ$ 上的标量函数，f 是未知的定义在 Ω° 上的标量函数。$\mathbf{v}$ 是已知的定义在 Ω 上的向量场。

f^* 在 Ω 上最简单的插值 f 就是膜插值 (membrane interpolant)，即极小化问题：

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

其解为

$$\Delta f = 0 \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{2}$$

这个方法会带来模糊等问题。我们将 (1) 改进为

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{3}$$

其唯一解为

$$\Delta f = \mathrm{div}\mathbf{v} \ \text{ over } \ \Omega, \ \text{ with } \ f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{4}$$

为了实际的操作，我们将 (3) 和 (4) 改为离散版本。S 和 Ω 是无限离散像素网格上定义的有限点集。对于每个 S 上的像素 p，N_p 为 p 在 S 中的上下左右 4 个相邻像素，符号 $\langle p, q\rangle$ 代表 $q \in N_p$。此时则有 $\partial\Omega = \{p \in S \backslash \Omega : N_p \cap \Omega \neq \emptyset\}$。令 f_p 为 f 在 p 处的值。我们的任务就是去计算 $f|_\Omega = \{f_p, p \in \Omega\}$

将 (3) 离散化，可得

$$\min_{f|_\Omega} \sum_{\langle p,q\rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2, \ \text{with } f_p = f_p^*, \ \text{for all } p \in \partial\Omega \tag{5}$$

其中 $v_{pq} = \mathbf{v}(\frac{p+q}{2}) \cdot \overrightarrow{pq}$。其解为

$$\text{for all } p \in \partial\Omega, \ |N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq} \tag{6}$$

2.2 Seamless cloning

2.2.1 Importing gradients

一个基本的选择就是选取 $\mathbf{v} = \nabla g$，此时有 $v_{pq} = g_p - g_q$。

2.2.2 Mixing gradients

上述的方法不会在 Ω 中保留目标图像 f^* 中的任何痕迹。然而在某些情况下，我们需要将 f^* 和 g 的属性结合起来，例如在一个有纹理或杂乱的背景上添加有孔的物体，或部分透明的物体。

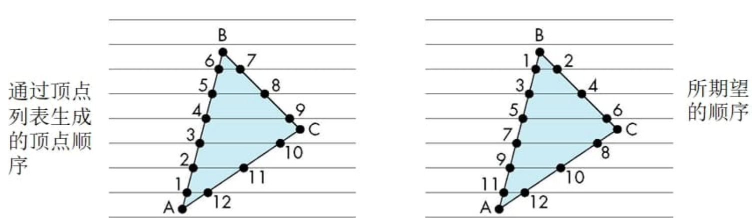
Possion 方法可以允许 $\mathbf{v}$ 为非保守梯度场，这将带来更加引人注目的效果。在 Ω 的每个点处，我们保留 f^* 和 g 中变化更强烈的信息，即取

$$\text{for all } x \in \Omega, \ \mathbf{v}(\mathbf{x}) = \begin{cases} \nabla f^*(\mathbf{x}) & \text{if } |\nabla f^*(\mathbf{x})| > |\nabla g(\mathbf{x})| \\ \nabla g(\mathbf{x}) & \text{otherwise} \end{cases} \tag{7}$$

2.3 多边形的扫描转换算法

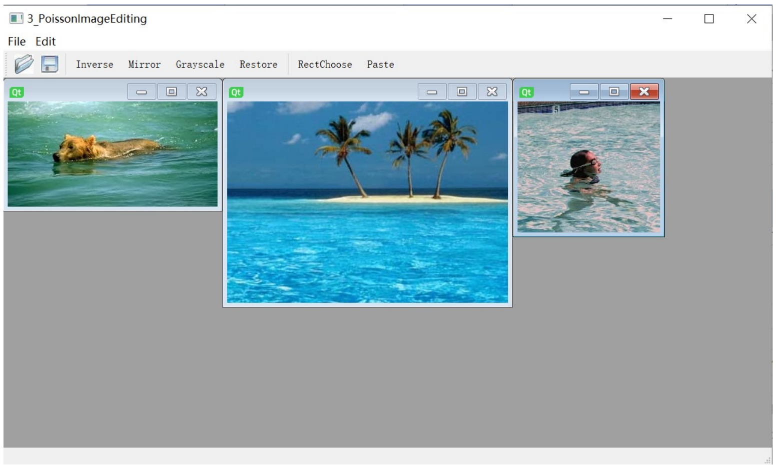
多边形的扫描转换算法是多边形区域光栅化（求解一个平面多边形区域的内部像素）的经典算法，可在任何一本计算机图形学的课本上都能找到，网上也有不少详细介绍资料。

算法的基本思想是：通过维持一个特别的数据结构（结构中保存扫描线与多边形的交点）进行填充。

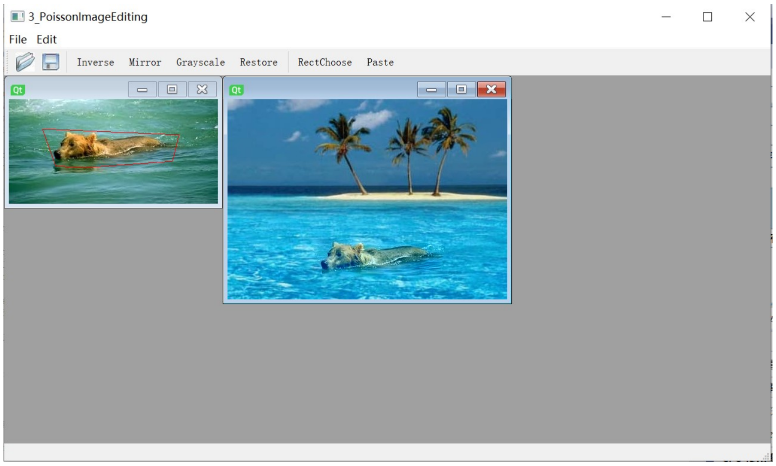


3 实现结果

3.1 UI 设计



3.2 possion 方法



4 不足

没有采用插值的方法来补缝。

5 参考文献

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

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