

Poisson Image Editing

1 算法

1.1 Poisson Seamless cloning

给定两个图像 f^* 和 g ，Poisson Seamless cloning是一种把 g 的选定区域和谐地复制到 f^* 的指定区域的算法，其基本想法是在 f^* 的指定区域上求一个函数 f ，使得它在边界与 f^* 相同的条件下梯度和 g 的梯度相差最小。求导，问题转化为求解poisson方程：

$$\begin{cases} \Delta f = \Delta g & \text{in } \Omega \\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \end{cases}$$

用二阶中心差分对上述方程离散化（硬约束），得到线性方程组：

$$4f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}$$

其中： $v_{pq} = g_p - g_q$;

Ω : 指定区域;

p : Ω 内任一像素点;

N_p : p 的四个邻居像素点的集合;

值得注意的是，这个方程组的系数矩阵正定对称，可以用Cholesky分解。且系数矩阵与 $f^*|_{\partial\Omega}$ 无关，可以预分解。

Mixing gradients

只是将梯度改为两个图像的局部最大梯度。

$$v_{pq} = \begin{cases} f_p^* - f_q^* & \text{if } |f_p^* - f_q^*| > |g_p - g_q| \\ g_p - g_q & \text{otherwise} \end{cases}$$

1.2 扫描线算法

用于确定多边形区域的内部、外部、边界，实现参考https://blog.csdn.net/xiaowei_cqu/article/details/7712451

按照此网页的算法，对于斜率绝对值小于1的边界线可能会漏掉一些边界点，稍微注意一下即可。

此外，我稍作了一点修改，因为不想对vector反复插入，我没有得到AET，而是直接从NET计算与每条扫描线的交点并得到这条扫描线上的内部/边界/外部点信息。

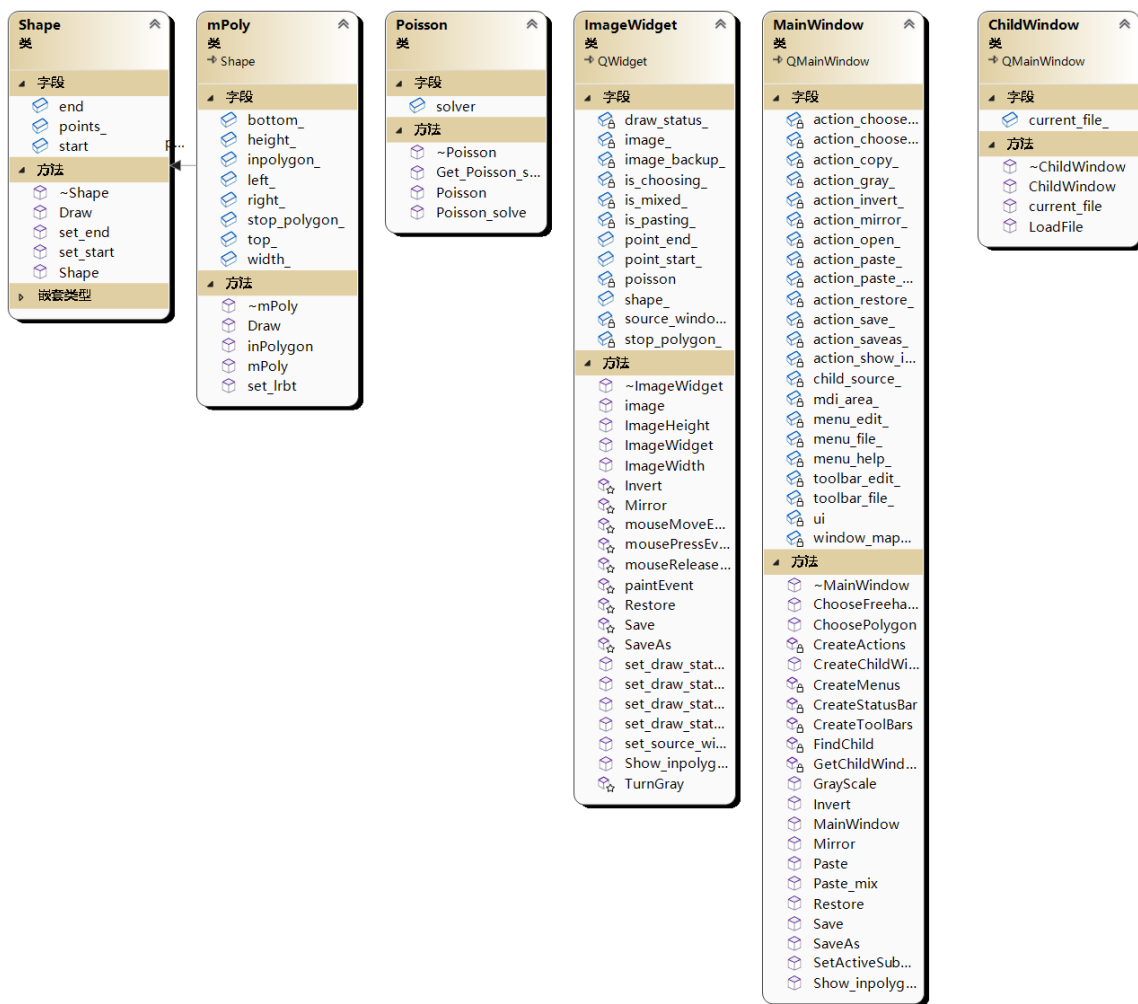
值得一提的是，上面的网页给出的算法通过在判断与扫描线的交点时把每条边视为下闭上开，并忽略水平边，巧妙的解决了所有奇异情况。

2 功能介绍

PolygonChoose, FrehandChoose: 画出要复制的区域。Polygon单击鼠标右键以结束，freehand松开鼠标结束，结束后自动计算系数矩阵并完成预分解。

Paste, Paste(mixed): 粘贴。

3 类图

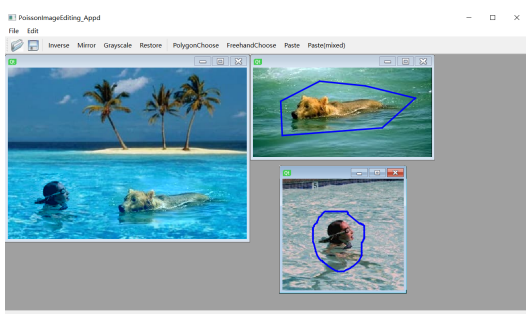


4 运行结果

原图像:



融合结果:



mix与非mix方法的比较



(a) Paste



(b) Paste(mixed)

扫描线算法结果：

