



计算机图形学

实验三：Poisson Image Editing

姓名： 王子卓

学号： 71115115

东南大学计算机科学与工程学院、软件学院

School of Computer Science & Engineering

College of Software Engineering

Southeast University

二〇壹柒年拾壹月

一、问题描述

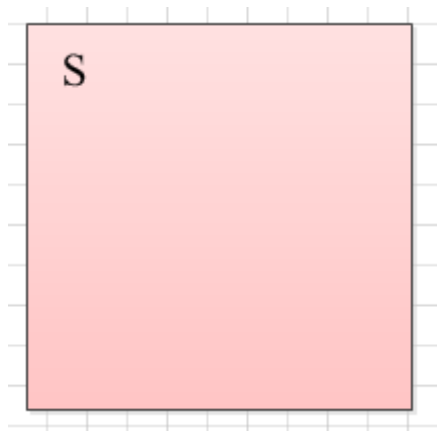
- 实现Poisson Image Editing算法的无缝融合应用(大型稀疏方程组求解)

二、算法描述

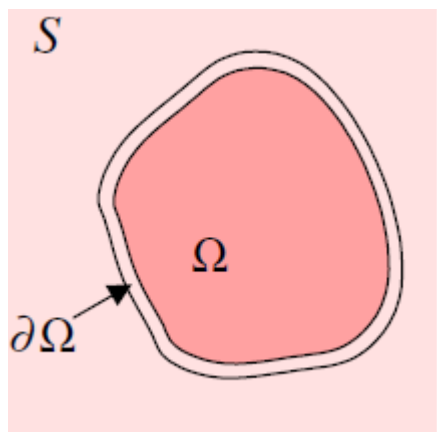
现在假设我们有图像 g ，如下图所示：



还有一张背景图片 S ：

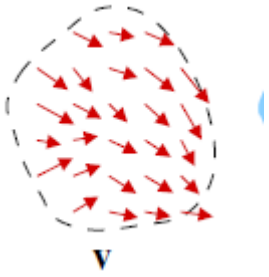


现在我们希望把图片 g 融合粘贴到 S 中，且实现自然融合的效果：



1.计算图像 g 的梯度场

通过差分的方法，可以求得图像 g 的梯度场 v ：

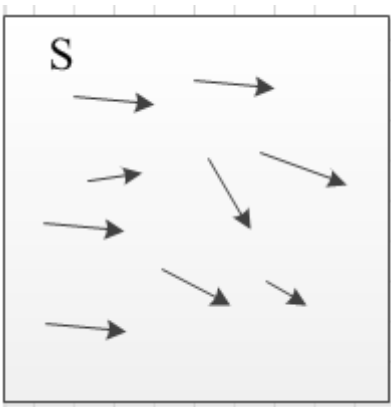


OpenCV接口为：

```
computeGradientX(patch,patchGradientX);// 计算ROI区域转换复制到 destination 一样大小的 patch 图片 x 方向梯度  
computeGradientY(patch,patchGradientY);// 计算 y 方向梯度
```

2. 计算背景图片的梯度场

变量destination为背景图像。这样就得到了背景图片的梯度场(destinationGradientX,destinationGradientY)，如下图：



```
computeGradientX(destination,destinationGradientX);// 计算背景图像的 x 方向梯度  
computeGradientY(destination,destinationGradientY);// 计算背景图像 y 方向的梯度
```

3. 计算融合图像的梯度场

计算完了以后，我们就把图像g的梯度场覆盖到S的梯度场上，先对destinationGradientX、destinationGradientY做mask操作：

```
arrayProduct(destinationGradientX,binaryMaskFloatInverted, destinationGradientX);  
arrayProduct(destinationGradientY,binaryMaskFloatInverted, destinationGradientY);
```

然后把图像g的梯度场覆盖到S的梯度场上：

```
Mat laplacianX = Mat(destination.size(),CV_32FC3);
Mat laplacianY = Mat(destination.size(),CV_32FC3);
//因为前面已经对destinationGradientX 做了固定区域的mask，patchGradientX 做了修改区域的mask
laplacianX = destinationGradientX + patchGradientX;// 求解整张图片新的梯度场
laplacianY = destinationGradientY + patchGradientY;
```

把背景图片的 Ω 区域的梯度场直接替换为g的梯度场v就可以了。

4.求解融合图像的散度

通过步骤3，我们可以得到每个像素点的梯度值，也就是待重建图像的梯度场，因此接着我们需要对梯度求偏导，从而获得散度。

```
computeLaplacianX(laplacianX,laplacianX);// 求解梯度的散度 也就是拉普拉坐标
computeLaplacianY(laplacianY,laplacianY);
```

5.求解系数矩阵

第4步我们已经把散度计算完毕，前面的泊松重建方程， $Ax=b$ ，b便是散度，因此接着我们需要只要构建系数矩阵，还有约束方程就完成了。

1	2	3
4	5	6
7	8	9

如果一幅图像，除了边界像素点之外，比如上面3*3图像的边界像素点1、2、3、4、6、7、8、9，其它像素点的散度(上图中的像素5)已知。那么可以列出泊松方程：

$$[V(2) + V(4) + V(6) + V(8)] - 4 * V(5) = div(5) \quad (1)$$

然后如果一幅图像的边界像素点的像素值已知，那么就可以求解泊松方程了，假设约束点的值为u。以上面3*3的图像为例，最后系数矩阵A的构造为：

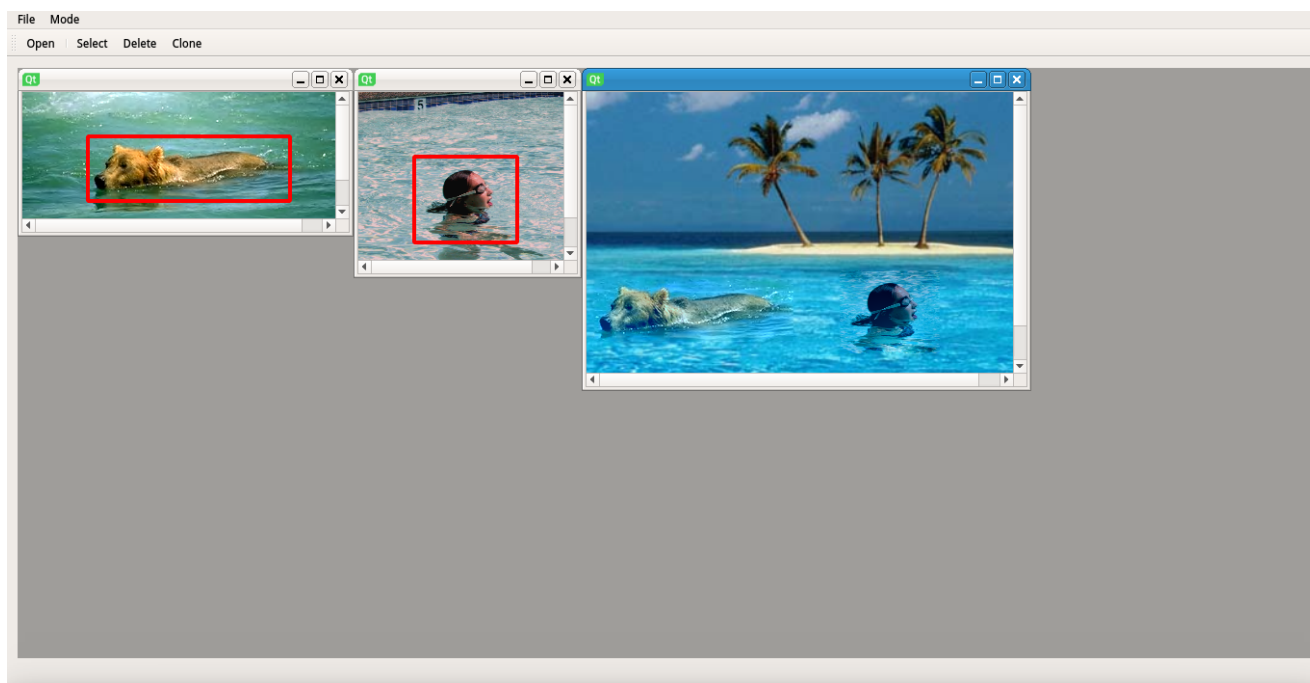
$$A = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & 1 & & 1 & -4 & 1 & & 1 & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix}$$

然后最后列出Ax=b的结果为：

$$\begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & 1 & & 1 & -4 & 1 & & 1 & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \cdot \begin{bmatrix} R_1 & G_1 & B_1 \\ R_2 & G_2 & B_2 \\ R_3 & G_3 & B_3 \\ R_4 & G_4 & B_4 \\ R_5 & G_5 & B_5 \\ R_6 & G_6 & B_6 \\ R_7 & G_7 & B_7 \\ R_8 & G_8 & B_8 \\ R_9 & G_9 & B_9 \end{bmatrix} = \begin{bmatrix} uR_1 & uG_1 & uB_1 \\ uR_2 & uG_2 & uB_2 \\ uR_3 & uG_3 & uB_3 \\ uR_4 & uG_4 & uB_4 \\ \text{div}R_5 & \text{div}G_5 & \text{div}B_5 \\ uR_6 & uG_6 & uB_6 \\ uR_7 & uG_7 & uB_7 \\ uR_8 & uG_8 & uB_8 \\ uR_9 & uG_9 & uB_9 \end{bmatrix}$$

这样分别求解三个通道的方程，我们就可以获得每个点的像素R,G,B值了。

三、结果



四、总结

通过这次作业，我感觉到任何计算机图形学的精妙效果都离不开数学知识的支持，算法是支持一切特效的基础。想要做出令人惊叹的图形学特效，需要有良好的数学知识。