

数字图像处理 实验报告

Part1 Image Fusion

一. 实验目标

用特定的算法将两幅或多幅图像综合成一幅新的图像。融合结果由于能利用两幅(或多幅) 图像在时空上的相关性及信息上的互补性, 并使得融合后得到的图像对场景有更全面、清晰的描述, 从而更有利于人眼的识别和机器的自动探测

二. 实验原理

泊松图像编辑

泊松图像编辑是泊松方程的一个重要应用, 首先提出该应用的是Poisson image editing. SIGGRAPH 2003,该文章对现在的图像编辑技术有着非常重要的影响, 随后的几年又出现了很多类似的图像编辑方法, 如 [Jiaya Jia et al. Drag and-drop pasting]于2006年提出了最优的融合边界用于改进泊松图像编辑的效果, [Zeev Farbman et al. coordinates for instant image cloning]在SIGGRAPH 2009中提出了使用Mean-Value coordinates用于计算基于梯度域的图像编辑, 该方法实现简单且运行速度快, 从而避免了求解复杂的泊松方程。

三. 实现思路

1. 背景图 T , 源图 S , 得到新图 N , 我们需要保持 S 中的区域 g 的梯度在 S 和 N 中保持一致, 因此可以得到方程组
2. 区域 g 从 S 挪到 N 中, 首先发生变化的便是 DEL_OMEGA 这一层的点, 因为 DEL_OMEGA 这一圈点的点附近有点的像素值发生了改变, 于是便会将这种变化层层传递, 使得图片无缝完美融合
3. 根据梯度场可求散度
4. $Ax=b$, 可利用库函数可解出 x

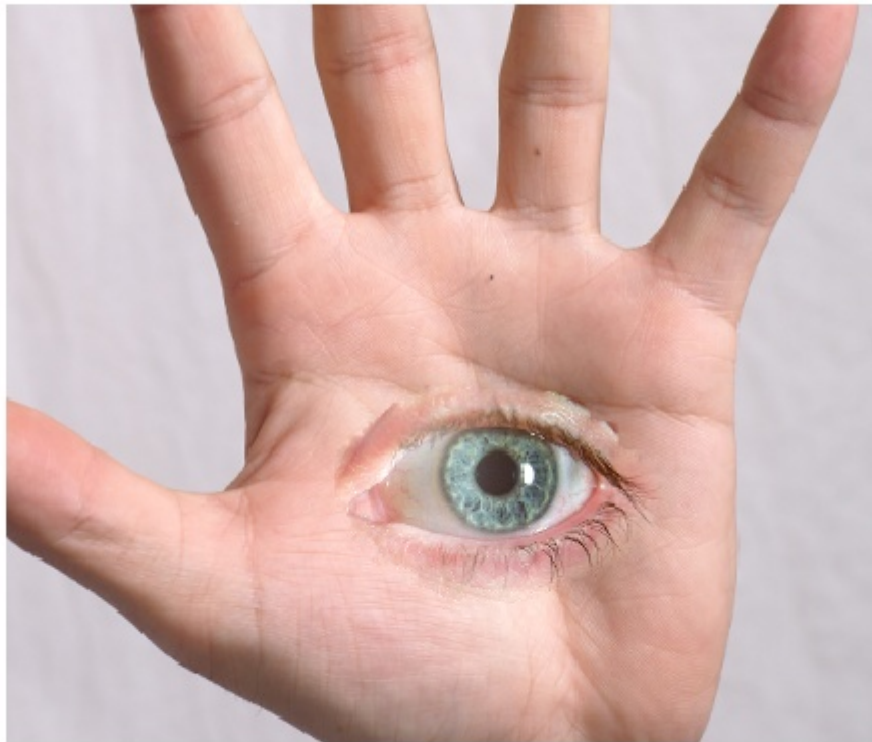
四. 实现细节

首先需要获得mask图像里白色部分的坐标, 然后在图 S 中找到对应的坐标的点, 然后根据像素值求出点的梯度, 然后, 对 DEL_OMEGA 的点进行维护, 利用周围像素值已经不一样的点维护矩阵 b , 然后根据相对位置维护矩阵 A , 然后调用库解出 x 即可

五. 代码说明

1. 修改 `src_img`, `mask_img`, `target_img`的绝对路径即可直接运行
2. `DEL_X`, `DEL_Y`常数为 g 在 N 中的 xy 偏移量, 可以通过修改这两个值修改 g 的位置

六. 效果图



Part 2 Image Morphing

一. 实验目标

实现人脸的过渡融合

二. 实验思路

1. 提取人脸关键点（调用Face++API）
2. 对两张图关键点进行三角剖分
3. 计算中间图特征点的坐标
4. 对两张图分别按照三角形求仿射矩阵
5. 遍历中间图的三角形，对其中的每个像素利用该三角形的仿射矩阵求出对应原图中的坐标，然后利用双线性插值获取像素值
6. 将两张中间图按照比例融合

三. 实现细节

1. 坐标！坐标！坐标！face++获取的坐标和像素的访问时候的坐标是反的，在调试过程中坐标的存储方式让我调了很久
2. 矩阵！一开始我求矩阵时两个矩阵的左右没有注意，于是在最终渲染图片的时候出现了明显的特征点错位的情况，我一直以为是矩阵求法有问题，但是其实是两个矩阵的相对位置放反了
3. 三角剖分获取的三角形的存储顺序是不一致的，不能用下标来访问，而应该利用点的下标来获取三角形，然后求仿射矩阵

四. 代码说明

1. 第一组图片中，需要使用第十行，十一行的路径，并且获取特征点应该使用269行和270行的代码，因为face++无法识别狮子，所以是手动添加特征点
2. 第二组图片中，使用13,14行的地址，并且添加15,16行的特征点字典。获取特征点使用266，257的代码
3. ALPHA为参数，手动调整可以获取不同比例的融合图片

五. 实验效果

见文件夹1,2，都以0.1为梯度

参考

<https://blog.csdn.net/u011534057/article/details/68922319>

课件

特征点字典：刘志强同学提供