

《数字图像处理》课程报告

课题名称： 基于 L0 范数的图像平滑处理

课题负责人名(学号)： 何长鸿 2016141482154

同组成员名单（角色）： _____

指导教师： 李新胜

评阅成绩： _____

评阅意见： _____

课程名称：

学生姓名：

学生学号：

提交报告时间：2018 年 12 月 25 日

基于 L0 范数的图像平滑处理

计算机科学与技术

学生 何长鸿 指导老师 李新胜

【摘要】 图像平滑是一项重要的图像处理技术，在图像平滑的基础上可进行不同效果的边缘检测，且经过平滑的图像具有一定的视觉特效。本文通过 Python 语言，基于 Liu Xu 等人的学术论文，利用傅里叶变换、光学传递等理论，实现基于 L0 范数的图像平滑，具有较好的平滑和去噪效果，并对平滑后的图像进行边缘提取，可发现获得边缘更平滑。

关键词：图像平滑 L0 范数 傅里叶变换、边缘提取

0 [目录](#)

一、	背景介绍	4
二、	算法原理说明	4
三、	算法实现	5
(一)	数据说明	5
(二)	获取光学传递函数	5
(三)	图像傅里叶变换	6
(四)	迭代进行 L0 范数最优化	6
四、	实验结果及分析	7
(一)	其他图像测试（右图为本算法平滑后图像）	7
(二)	不同参数对比	8
(三)	在不同参数获得的图像上进行边缘提取	8
(四)	使用均值滤波后的边缘提取对比	9
五、	结论与展望	10
六、	参考文献	11

一、 背景介绍

图像平滑是计算摄影学一门基础重要的工具,其作用是拂去不重要的细节,保留较大的图像边缘,主要应用于边缘检测, JPEG 压缩图像人工伪迹去除,非真实绘制等领域。

图像平滑大体上可以分为两类:基于局部和基于全局方法,基于局部的方法像有名双边滤波,各向异性扩散,将图像分成一些局部块进行处理;全局方法比如全变分(Total Variation)和最小二乘滤波(Weighted Least Square),同时处理整幅图像,可以达到全局最优的目的。

以往的方法,拂去图像中去对图像细节部分也会对图像中大的边缘进行惩罚,这样也会导致图像中大的边缘减弱或丢失,因此徐立等人提出使用图像 L0 范数平滑,该滤波器是一种基于稀疏策略的全局平滑滤波器。

本文是对香港中文大学徐立等人所做的《Image Smoothing via L0 Gradient Minimization》的 Python 实现。该算法具有以下优点:

- 1) 通过去除小的非零梯度,抚平不重要的细节信息
- 2) 增强图像显著性边缘

由于该算法涉及到 L0 范数优化问题,计算过程比较复杂,原论文作者仅提供了 Matlab 程序代码,本文将通过论文及 Matlab 代码作为参考,使用 Python 实现该算法。

二、 算法原理说明

本文中, L0 范数指向量中非零元素的个数。本文先对图像向量计算梯度,由于初始梯度具有较多的噪点,利用傅里叶变换低通滤波去除噪点,在指定参数下完成图像梯度最小化平滑,从而达到尽可能保持图像主要边缘的情况下,使得图像平滑的目的。由于论文中有详细算法介绍,此处不再重复。

三、 算法实现

(一) 数据说明

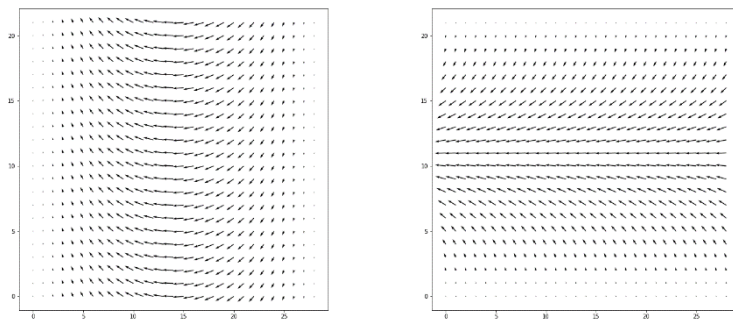
本文中算法实现部分使用例图如下：



图表 1 测试用图

(二) 获取光学传递函数

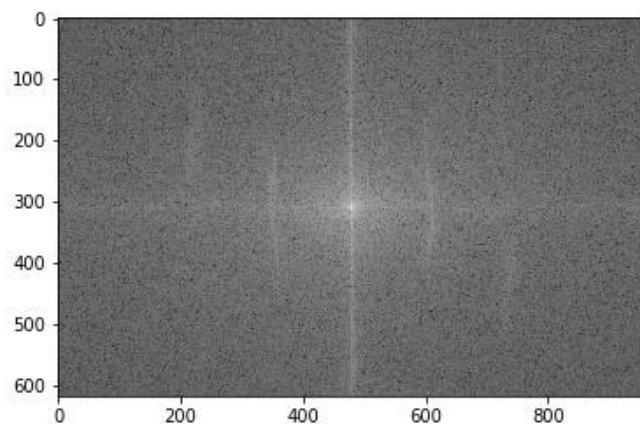
该步骤分别生成 X、Y 方向上的光学传递函数，用于频域空间图像梯度计算。光学函数为复数向量形式，分别以实部和虚部为向量分量，做出可视化模型图：



图表 2 X,Y 方向光学传递算子

(三) 图像傅里叶变换

分别对图像三个通道进行傅里叶二维变换。变换后的能量分布图如下:



图表 3 频域能量分布可视化

(四) 迭代进行 L0 范数最优化

L0 范数优化求值过程相对复杂, 并且速度较慢, 但可通过编写 CUDA 并行计算程序提高速度。文本附录中的代码为 CPU 串行执行代码。

当输入参数 $k=2$, $l=0.05$ 时, 得到如下平滑图像:



图表 4 平滑效果

可以看到, 该图像线条结构被明显简化, 结构复杂而颜色相似的部分被明显融合, 并且不同颜色的边界得到最大化保留, 区域之间没有颜色的融合。通

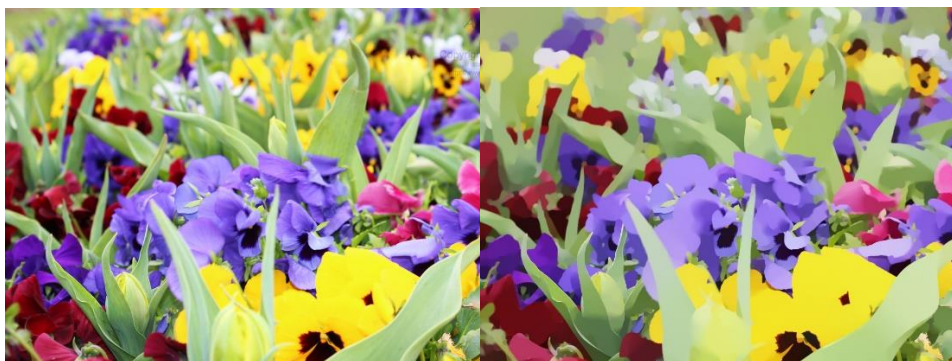
过调整 k, l 两者的比例，可以改变信息保留程度。

四、实验结果及分析

以下分别通过多个维度的对比来说明本文算法的优越性。

(一) 其他图像测试（右图为本算法平滑后图像）

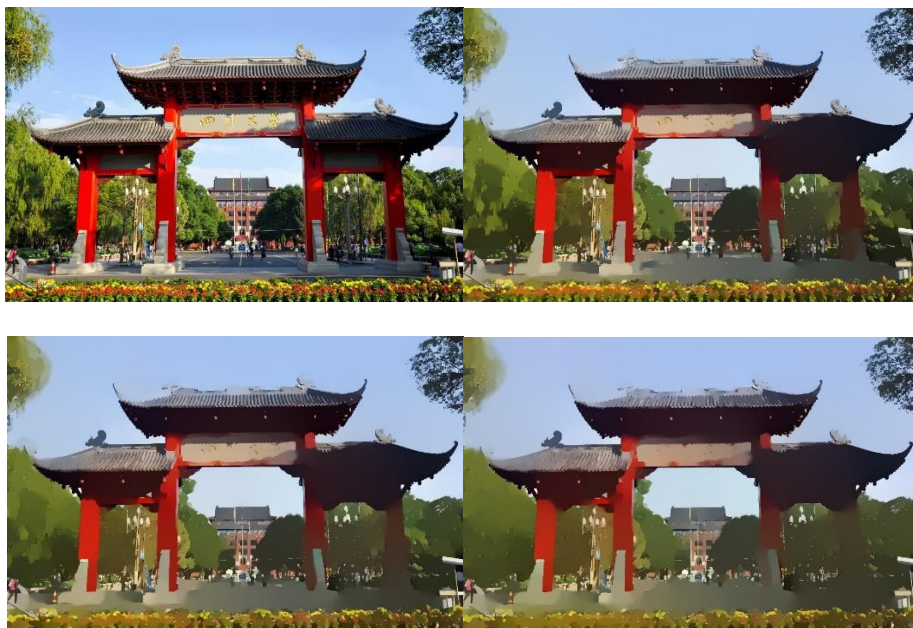
可以看到，平滑后的图像丢弃了颜色相近领域内的线条（边缘），有卡通水彩画的效果。



图表 5 不同图像使用 L0 范数平滑效果

(二) 不同参数对比

由于最终效果主要取决于参数 λ 和 κ 的比值, 因此设置 κ 为 2, 调节 λ 。从左上到右下四幅图的 λ 参数分别为(0.01, 0.05, 0.1, 0.15)。随着参数改变, 图像的颜色层次越简单, 从而达到平滑的目的。

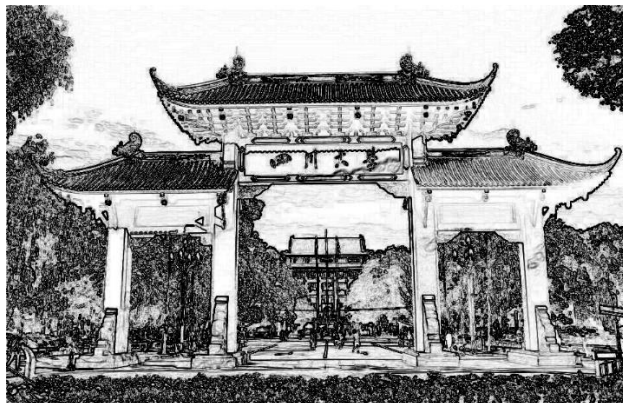


图表 6 的影响参数对平滑效果

(三) 在不同参数获得的图像上进行边缘提取

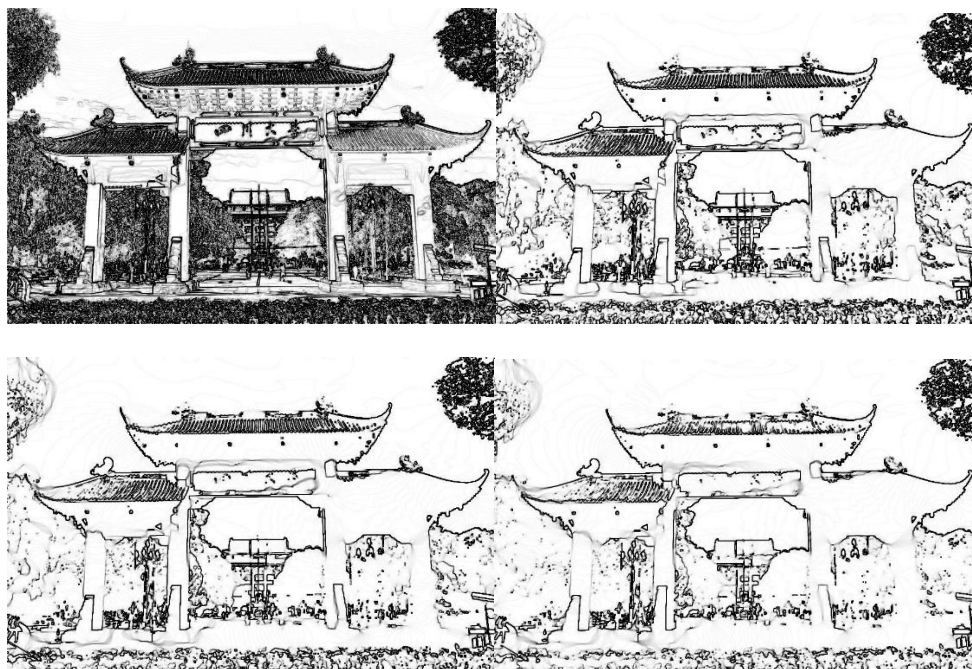
以下使用 Sobel 算子, 分别在原图和以上的四幅图中进行边缘提取:

- 1) 原图: 能够比较完整的得到图像信息, 但作为边缘提取存在较多复杂线条, 不够简洁。



图表 7 边缘提取（原图）

2) 不同平滑参数下的边缘提取

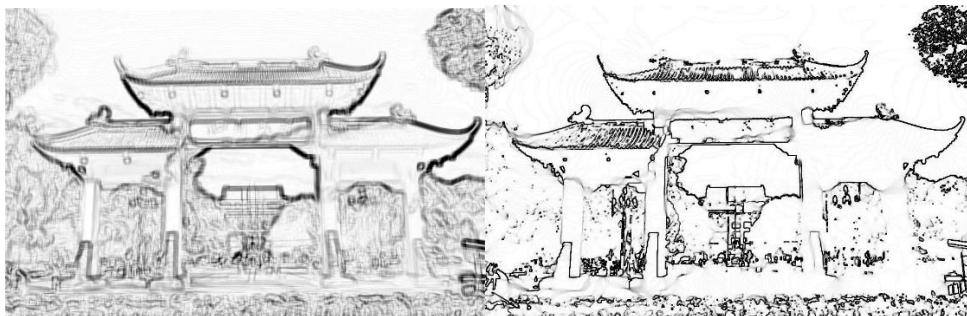


图表 8 不同 L0 平滑参数下的边缘提取

(四) 使用均值滤波后的边缘提取对比

从下图（左图为均值滤波，右图为本文方法）可以看到，通过均值滤波后

的边缘提取虽然也能获得结构简单的边缘，但整个图像相对暗淡，线条变粗，而本文方法提取的边缘更加干净整洁，线条更细腻清晰。



图表 9 均值滤波与 L0 范数平滑对比

五、 结论与展望

本文使用 Python 实现了基于 L0 范数的图像平滑算法，并且测试了不同参数下平滑的效果，以及通过均值滤波后的边缘提取与本文中算法的边缘提取，对比表现出了本算法的优势。

但此算法也存在一定的局限性，通过迭代对目标函数进行优化的计算量较大，速度较慢。在测试环境 3.5Ghz CPU 下，处理 1080x784 分辨率的图像速度为 15s/张。但理论上可以通过 GPU 并行计算提高速度。

六、 参考文献

- [1] Li Xu、Cewu Lu、Yi Xu、Jiaya Jia. Image Smoothing via L0 Gradient Minimization.
- [2] L0 范数图像平滑. <https://blog.csdn.net/bluecol/article/details/48750561>
- [3] 傅里叶变换的使用. <https://www.jb51.net/article/144187.htm>