

[Python图像处理] 二十三.傅里叶变换之高通滤波和低通滤波

原创 Eastmount 2019-04-28 20:35:03 22598 收藏 116

编辑 版权

分类专栏: Python图像处理及图像识别 文章标签: Python图像处理 傅里叶变换 高通滤波 低通滤波



Python图像处理及图像识别

¥9.90

本专栏主要结合Python语言讲述图像处理相关的知识,从二值图像、灰度图像到RGB图像基础知识,再到常见的图像处理算法,包括:灰度算法、图像锐化、图像分割等知识,最后会结合深度学习和机器...



Eastmount

该系列文章是讲解Python OpenCV图像处理知识,前期主要讲解图像入门、OpenCV基础用法,中期讲解图像处理的各种算法,包括图像锐化算子、图像增强技术、图像分割等,后期结合深度学习研究图像识别、图像分类应用。希望文章对您有所帮助,如果有不足之处,还请海涵~

该系列在github所有源代码: <https://github.com/eastmoutyxz/ImageProcessing-Python>

PS: 请求帮忙点个Star,哈哈,第一次使用Github,以后会分享更多代码,一起加油。

同时推荐作者的C++图像系列知识:

[数字图像处理] 一.MFC详解显示BMP格式图片

[数字图像处理] 二.MFC单文档分割窗口显示图片

[数字图像处理] 三.MFC实现图像灰度、采样和量化功能详解

[数字图像处理] 四.MFC对话框绘制灰度直方图

[数字图像处理] 五.MFC图像点运算之灰度线性变化、灰度非线性变化、阈值化和均衡化处理详解

[数字图像处理] 六.MFC空间几何变换之图像平移、镜像、旋转、缩放详解

[数字图像处理] 七.MFC图像增强之图像普通平滑、高斯平滑、Laplacian、Sobel、Prewitt锐化详解

前文参考:

[Python图像处理] 一.图像处理基础知识及OpenCV入门函数

[Python图像处理] 二.OpenCV+Numpy库读取与修改像素

[Python图像处理] 三.获取图像属性、兴趣ROI区域及通道处理

[Python图像处理] 四.图像平滑之均值滤波、方框滤波、高斯滤波及中值滤波

[Python图像处理] 五.图像融合、加法运算及图像类型转换

[Python图像处理] 六.图像缩放、图像旋转、图像翻转与图像平移

[Python图像处理] 七.图像阈值化处理及算法对比

[Python图像处理] 八.图像腐蚀与图像膨胀

[Python图像处理] 九.形态学之图像开运算、闭运算、梯度运算

[Python图像处理] 十.形态学之图像顶帽运算和黑帽运算

[Python图像处理] 十一.灰度直方图概念及OpenCV绘制直方图

[Python图像处理] 十二.图像几何变换之图像仿射变换、图像透视变换和图像校正

[Python图像处理] 十三.基于灰度三维图的图像顶帽运算和黑帽运算

[Python图像处理] 十四.基于OpenCV和像素处理的图像灰度化处理

[Python图像处理] 十五.图像的灰度线性变换

[Python图像处理] 十六.图像的灰度非线性变换之对数变换、伽马变换

[Python图像处理] 十七.图像锐化与边缘检测之Roberts算子、Prewitt算子、Sobel算子和Laplacian算子

[Python图像处理] 十八.图像锐化与边缘检测之Schar算子、Canny算子和LOG算子

[Python图像处理] 十九.图像分割之基于K-Means聚类的区域分割

[Python图像处理] 二十.图像量化处理和采样处理及局部马赛克特效

[Python图像处理] 二十一.图像金字塔之图像向下取样和向上取样

[Python图像处理] 二十二.Python图像傅里叶变换原理及实现

前面一篇文章我讲解了傅里叶变换,它将时间域上的信号转变为频率域上的信号,用来进行图像除噪、图像增强等处理。本文将继续补充基于傅里叶变换的高通滤波和低通滤波。基础性文章,希望你有所帮助。同时,该部分知识均为杨秀璋查阅资料撰写,转载请署名CSDN+杨秀璋及原地址出处,谢谢!!

1.高通滤波

2.低通滤波

3.本章小结

PS：文章参考自己以前系列图像处理文章及OpenCV库函数，同时参考如下文献：

《数字图像处理》（第3版），冈萨雷斯著，阮秋琦译，电子工业出版社，2013年.

《数字图像处理学》（第3版），阮秋琦，电子工业出版社，2008年，北京.

《OpenCV3编程入门》，毛星云，冷雪飞，电子工业出版社，2015，北京.

[百度百科-傅里叶变换](#)

网易云课堂-高登教育 Python+OpenCV图像处理

一.高通滤波

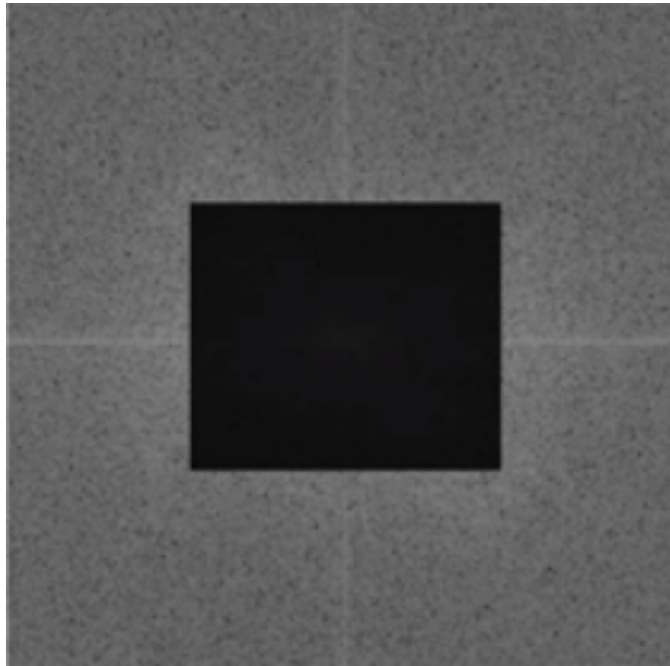
傅里叶变换的目的并不是为了观察图像的频率分布（至少不是最终目的），更多情况下是为了对频率进行过滤，通过修改频率以达到图像增强、图像去噪、边缘检测、特征提取、压缩加密等目的。

过滤的方法一般有三种：低通（Low-pass）、高通（High-pass）、带通（Band-pass）。所谓低通就是保留图像中的低频成分，过滤高频成分，可以把过滤器想象成一张渔网，想要低通滤波器，就是将高频区域的信号全部拉黑，而低频区域全部保留。例如，在一幅大草原的图像中，低频对应着广袤且颜色趋于一致的草原，表示图像变换缓慢的灰度分量；高频对应着草原图像中的老虎等边缘信息，表示图像变换较快的灰度分量，由于灰度尖锐过度造成

高通滤波器是指通过高频的滤波器，衰减低频而通过高频，常用于增强尖锐的细节，但会导致图像的对比度会降低。该滤波器将检测图像的某个区域，根据像素与周围像素的差值来提升像素的亮度。图展示了“Lena”图对应的频谱图像，其中心区域为低频部分。



接着通过高通滤波器覆盖掉中心低频部分，将255两点变换为0，同时保留高频部分，其处理过程如下图所示。



```
rows, cols = img.shape
crow, ccol = int(rows/2), int(cols/2)
fshift[crow-30:crow+30, ccol-30:ccol+30] = 0
```

通过高通滤波器将提取图像的边缘轮廓，生成如下图所示图像。



原始图像

高通滤波图像

<http://www.it-ebooks.info>

```
# -*- coding: utf-8 -*-
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

#读取图像
img = cv.imread('Lena.png', 0)

#傅里叶变换
f = np.fft.fft2(img)
fshift = np.fft.fftshift(f)

#设置高通滤波器
rows, cols = img.shape
crow, ccol = int(rows/2), int(cols/2)
```

```
fshift[crow-30:crow+30, ccol-30:ccol+30] = 0

#傅里叶逆变换
ishift = np.fft.ifftshift(fshift)
iimg = np.fft.ifft2(ishift)
iimg = np.abs(iimg)

#显示原始图像和高通滤波处理图像
plt.subplot(121), plt.imshow(img, 'gray'), plt.title('Original Image')
plt.axis('off')
plt.subplot(122), plt.imshow(iimg, 'gray'), plt.title('Result Image')
plt.axis('off')
plt.show()
```

输出结果如下图所示，第一幅图为原始“Lena”图，第二幅图为高通滤波器提取的边缘轮廓图像。它通过傅里叶变换转换为频谱图像，再将中心的低频部分设置为0，再通过傅里叶逆变换转换为最终输出图像“Result Image”。



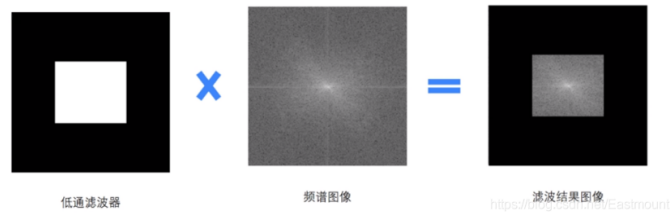
二.低通滤波

低通滤波器是指通过低频的滤波器，衰减高频而通过低频，常用于模糊图像。低通滤波器与高通滤波器相反，当一个像素与周围像素的插值小于一个特定值时，平滑该像素的亮度，常用于去噪和模糊化处理。如PS软件中的高斯模糊，就是常见的模糊滤波器之一，属于削弱高频信号的低通滤波器。

下图展示了“Lena”图对应的频谱图像，其中心区域为低频部分。如果构造低通滤波器，则将频谱图像中心低频部分保留，其他部分替换为黑色0，其处理过程如图所示，最终得到的效果图为模糊图像。



那么，如何构造该滤波图像呢？如下图所示，滤波图像是通过低通滤波器和频谱图像形成。其中低通滤波器中心区域为白色255，其他区域为黑色0。



低通滤波器主要通过矩阵设置构造，其核心代码如下：

```
rows, cols = img.shape
crow, ccol = int(rows/2), int(cols/2)
mask = np.zeros((rows, cols, 2), np.uint8)
mask[crow-30:crow+30, ccol-30:ccol+30] = 1
```

通过低通滤波器将模糊图像的完整代码如下所示：

```
# -*- coding: utf-8 -*-
import cv2
import numpy as np
from matplotlib import pyplot as plt

# 读取图像
img = cv2.imread('lena.bmp', 0)

# 傅里叶变换
dft = cv2.dft(np.float32(img), flags = cv2.DFT_COMPLEX_OUTPUT)
fshift = np.fft.fftshift(dft)

# 设置低通滤波器
rows, cols = img.shape
crow, ccol = int(rows/2), int(cols/2) # 中心位置
mask = np.zeros((rows, cols, 2), np.uint8)
mask[crow-30:crow+30, ccol-30:ccol+30] = 1

# 掩膜图像和频谱图像乘积
f = fshift * mask
print f.shape, fshift.shape, mask.shape

# 傅里叶逆变换
ishift = np.fft.ifftshift(f)
iimg = cv2.idft(ishift)
res = cv2.magnitude(iimg[:, :, 0], iimg[:, :, 1])

# 显示原始图像和低通滤波处理图像
plt.subplot(121), plt.imshow(img, 'gray'), plt.title('Original Image')
plt.axis('off')
plt.subplot(122), plt.imshow(res, 'gray'), plt.title('Result Image')
plt.axis('off')
plt.show()
```

输出结果如图所示，第一幅图为原始“Lena”图，第二幅图为低通滤波器模糊处理后的图像。

Original Image



Result Image



<https://blog.csdn.net/Eastmount>

三.总结

讲到这里，傅里叶变换的图像增强、图像去噪、边缘检测、特征提取已介绍，前面的文章详细讲解了图像平滑和图像锐化各种算子，希望读者能进行相关对比。下一篇文章，作者将分享几个Python图像处理的特效处理，包括素描、黄昏、灯光、浮雕等效果，希望读者喜欢。

时也，命也。

英语低分数线一分，些许遗憾，但不气馁，更加努力。雄关漫道真如铁，而今迈过从头越，从头越。苍山如海，残阳如血。感谢一路陪伴的人和自己。

无论成败，那段拼搏的日子都很美。结果只会让我更加努力，学好英语。下半年沉下心来好好做科研写文章，西藏之行，课程分享。同时，明天的博士考试加油，虽然裸泳，但也加油！还有春季招考开始准备。

最后补充马刺小石匠精神，当一切都看起来无济于事的时候，我去看一个石匠敲石头。他一连敲了100次，石头仍然纹丝不动。但他敲第101次的时候，石头裂为两半。可我知道，让石头裂开的不是那最后一击，而是前面的一百次敲击的结果。人生路漫漫，不可能一路一帆风顺，暂时的不顺只是磨练自己的必经之路，夜最深的时候也是距黎明最近的时刻，经历过漫漫长夜的打磨，你自身会更加强大。

最后希望这篇基础性文章对您有所帮助，如果有错误或不足之处，请海涵！



Eastmount 今天考完最后一科，这半年报考的博士基本结束，五月份还有最后一个学校。都是离家近的，横跨贵州、云南、四川、重庆、湖南、湖北、福建等，有的太忙错过时间，有点可惜。但不论结果，这段经历我真是享受，回头看看，过程很苦，岁月很美。谢谢女神这一年的陪伴。同时，我希望我的学生也拿出这种冲劲，找到属于自己的工作和未来。最后，真的好好休息下了，新的日子扬帆起航



<https://blog.csdn.net/Eastmount>

2020年8月18新开的“娜璋AI安全之家”，主要围绕Python大数据分析、网络空间安全、人工智能、Web渗透及攻防技术进行讲解，同时分享CCF、SCI、南核北核论文的算法实现。娜璋之家会更加系统，并重构作者的所有文章，从零讲解Python和安全，写了近十年文章，真心想把自己所学所感所做分享出来，还请各位多多指教，真诚邀请您的关注！谢谢。

