

数字图像处理实验一

实验指导

1. 写在最初的话

如果需要用Python取代Matlab的大部分功能，Numpy,Scipy,Matplotlib这三个类库就够了，除此以外，图像处理还有PIL和pyopencv，小波变换有pywt，机器学习相关的有pebl,pymvpa，对于SVM的libsvm也有Python接口，还有许多其他科学计算的类库。

打算用Python编写后面四个实验的你，或许早已掌握Python语言的基本用法。如果还没有接触过，那也不用担心，作为一种解释性语言，你都已经把有点基于Fortran语法略显怪的matlab都会了，根本不需要担心优雅的Python，找一本好的书，半天时间足够了。推荐Magnus Lie Hetland的Python基础教程，更为详细一点的可以看，Learning Python。

对于独立的编写接下来的四个实验，或许你还会需要张若愚写的Python科学计算，里面详细介绍了Python如何用做科学计算特别是信号处理领域，对应本实验，也就是Scipy,Numpy,Matplotlib，尤其是Matplotlib，如果不搞清楚它的那些container的层次，还是有点恼人的。

在以后的日子里，你可能需要根据概念定义编写直方图均衡、直方图规定化，Sobel锐化，Laplacian锐化，Roberts锐化等函数,对此，你准备好了吗？上课一定要认真听讲哦！

带上冈萨雷斯的第三版一块出发吧！

2. 实验目的

通过上机实验的手段巩固课堂上所学的关于图像反转、幂次变换和分段线性变换有关的知识，感受不同的灰度变换方法对最终图像效果的影响。

3. 实验内容

3.1. 方法技术介绍

灰度变换函数是图像增强技术中最简单的一类，处理前后的像素值分别用 r 和 s 定义，于是我们可以如下表达式来表示原像素值 r 映射到值 s 的变换：

$$s = T(r) \quad (1)$$

3.1.1. 图像反转

灰度级范围为 $[0, L-1]$ 的图像反转表达式为：

$$s = L - 1 - r \quad (2)$$

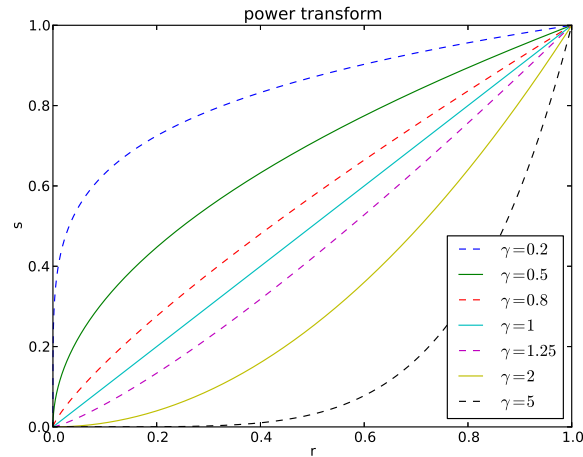
采用上式可以产生图像反转的对等图像，这种处理尤其适合用于增强嵌入于图像暗色区域的白色或灰色细节，特别是当黑色面积占主导地位时。

3.1.2. 幂次变换

幂次变换的基本形式为：

$$s = cr^\gamma \quad (3)$$

其中 c 和 γ 为正常数。从下图可以看出，当 $\gamma < 1$ ，幂次变换能够将窄带暗值映射成宽带输出值， $\gamma > 1$ ，则正好相反。应注意的是，当 $c = \gamma = 1$ 时，幂次变换将简化为正比运算。



3.1.3. 分段线性变换

与前面两个函数相比，分段线性变换的主要优势在于形式可以任意合成，其中分段线性变换还可以细分为对比拉伸、灰度切割、位图切割等方法。

3.1.3.a. 对比拉伸

对比拉伸的思想是提高低对比度图像处理是灰度级的动态范围。低对比度的成因有照明不足、成像传感器动态范围太小，甚至在图像获取过程中透镜光圈设置错误引起。

3.1.3.b. 灰度切割

在图像处理中提高特定灰度范围的亮度通常是很有必要的,这样的灰度切割可以让感兴趣的图像内容更加明显。

3.2. 实验步骤

3.2.1. 步骤1

将实验用图：rice.png读入内存，以便后续处理。

3.2.2. 步骤2

用如下的灰度变换函数对rice.png图像进行处理，在同一个窗口中画出处理前后的图像，并保存。

$$s = \begin{cases} 0.3r, & r < 0.35 \\ 0.105 + 2.6333(r - 0.35), & 0.35 \leq r \leq 0.65 \\ 1 + 0.3(r - 1), & r > 0.65. \end{cases} \quad (4)$$

3.2.3. 步骤3

用如下灰度变换函数对rice.png进行处理，在同一个窗口中画出处理前后的图像，并保存。

$$s = \begin{cases} 15.9744r^5, & r \leq 0.5 \\ (r - 0.5)^{0.2} + 0.12, & r > 0.5 \end{cases} \quad (5)$$

3.2.4. 步骤4

分别用 $s = r^{0.6}$ $s = r^{0.4}$ $s = r^{0.3}$ 对kids.tif图像进行处理，在同一个窗口中画出处理前后的图像，并保存。

3.2.5. 步骤5

对circuit.jpg图像实施反变换，在同一个窗口中画出处理前后的图像，并保存。

3.2.6. 步骤6

对rice.jpg图像实施灰度切片。具体要求为：当 $0.2 \leq r \leq 0.4$ 时，将 r 置为 0.6，当 r 位于其他区间时，保持其灰度值不变。在同一个窗口中画出处理前后的图像，并保存。

3.2.7. 步骤7

利用灰度变换对Picture.jpg做增强处理，突出图中人物，改善整个图像过于灰暗的背景。给出图像处理前后的直方图，写出所采用的拉伸表达式。在同一个窗口中画出处理前后的图像，并保存。

4. 实验报告要求

提交原图像和各种变换函数的曲线，以及按照各种变换函数处理后的图像。

5. Tips

稍许隐形福利。

先行约定采用如下方式导入相关模块：

```
import Image
import numpy as np
import matplotlib.cm as cm
import matplotlib.pyplot as plt
```

5.1. 读入图片

可以使用matplotlib下的imread函数，也可以使用Image的Open，再将图片对象转化成numpy数组，以便于后面的数学操作。convert是将图片转换成指定的mode。简单的例子如下：

```
rice_0 = np.asarray(Image.open('rice.png').convert('L'))/float(255)
```

5.2. 绘图

matplotlib是一个面向对象的绘图库，如果不搞清楚那些containers，真的很难受。建议参考《Python科学计算》，里面讲述的很清晰。先放上一个示例，仅供参考。

```
fig = plt.figure() # create a window to put images
ax1 = fig.add_subplot(221) # add subimage to fig
# default imshow mode is pseudo-color ,set gray
```

```
ax1.imshow(rice_0,cmap=cm.gray,norm=pl.Normalize(vmin=0,vmax=255))
ax1.set_title('input image') # set current subimage's name
ax1.set_xticks([])
ax1.set_yticks([])
fig.suptitle('exp_1_2') # set title of whole image
fig.savefig('exp_1_2.jpg') # save image
```