

图像处理基础

（作业1）



学 院（系）： 电子信息与电气工程学部

专 业： 计算机科学与技术

班 级： 电计1704

学 生 姓 名： 谢玉宁

学 号： 201792260

完 成 日 期： 2020年3月17日

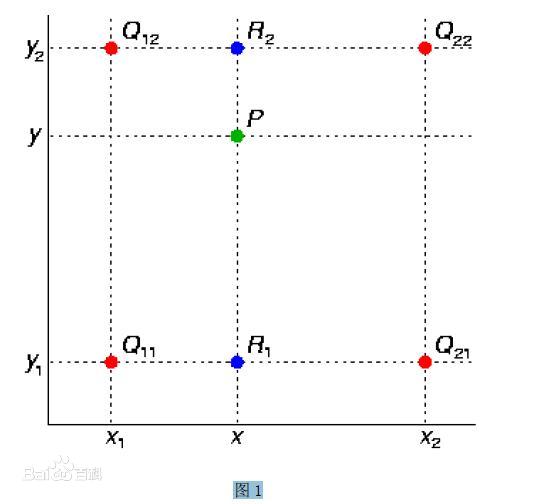
实验目标：实现双线性插值（bilinear interpolation）

实现原理:

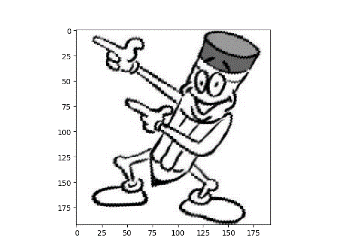
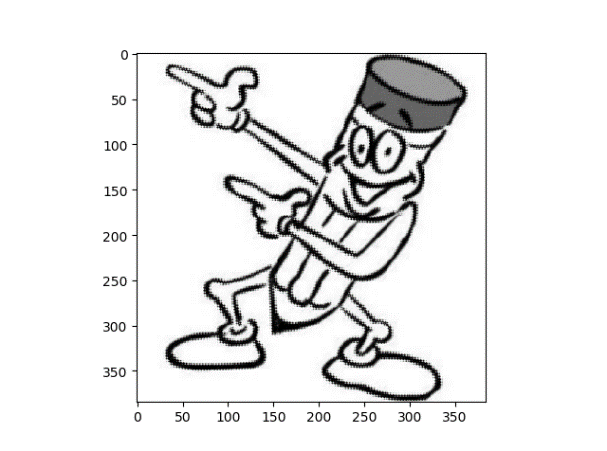
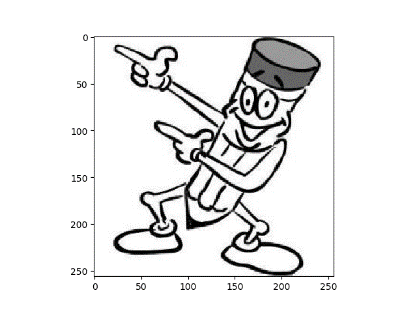
如图，已知Q12，Q22，Q11，Q21，首先在x轴方向上，对R1和R2两个点进行插值，这个很简单，然后根据R1和R2对P点进行插值，这就是所谓的双线性插值。在数学上，双线性插值是有两个变量的插值函数的线性插值扩展，其核心思想是在两个方向分别进行一次线性插值。

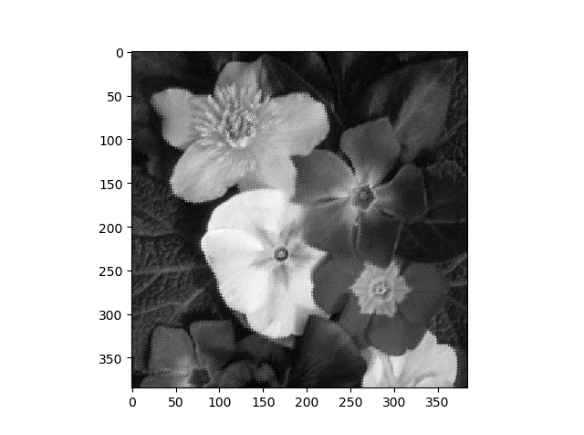
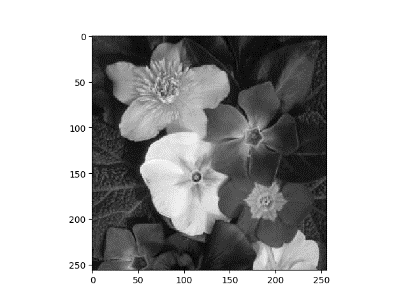
第一步：X方向的线性插值，在Q12,Q22中插入蓝色点R2，Q11，Q21中插入蓝色点R1；

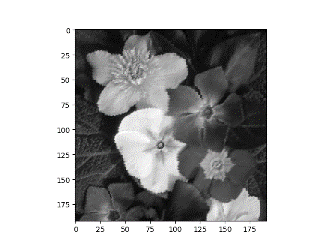
第二步 ：Y方向的线性插值 ,通过第一步计算出的R1与R2在y方向上插值计算出P点。



输入与输出图像(matplotlib):







结果分析:

双线性插值法总体上能够较好地保持原图片的灰度值信息，但是在边缘等细节处会产生失真，而在平滑的地方效果较好。

代码（Python）：

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.image as plimg

#im1\_path = r'C:\Users\Lenovo\Desktop\CARTOON.jpg'

im1\_path = r'C:\Users\Lenovo\Desktop\flowergray.jpg'

im1 = plimg.imread(im1\_path)

print(im1.shape)

%matplotlib qt5

plt.imshow(im1,cmap='gray')

def interplotation(coord,derive\_size,origin\_img):

delta\_x = (origin\_img.shape[0]-1)/(derive\_size[0]-1)

delta\_y = (origin\_img.shape[1]-1)/(derive\_size[1]-1)

x = coord[0]\*delta\_x

y = coord[1]\*delta\_y

gray1 = 0

gray2 = 0

gray\_level = 0

x1 = int(np.floor(x))

x2 = int(np.ceil(x))

y1 = int(np.floor(y))

y2 = int(np.ceil(y))

if(x1==x2):

gray1 = origin\_img[x1][y1]

gray2 = origin\_img[x1][y2]

else:

gray1 = (x-x1)\*origin\_img[x1][y2]+(x2-x)\*origin\_img[x1][y1]

gray2 = (x-x1)\*origin\_img[x2][y2]+(x2-x)\*origin\_img[x2][y1]

if(y1==y2):

gray\_level = gray1

else:

gray\_level = (y-y1)\*gray2+(y2-y)\*gray1

return gray\_level

def bilinear(k,origin):

derive = np.zeros((int(k\*origin.shape[0]),int(k\*origin.shape[1])),dtype = int)

for i in range(derive.shape[0]):

for j in range(derive.shape[1]):

derive[i,j] = interplotation((i,j),derive.shape,origin)

return derive

def imresize(original\_image,original\_size,target\_size):

derive = np.zeros((int(target\_size[0]),int(target\_size[1])),dtype = int)

for i in range(derive.shape[0]):

for j in range(derive.shape[1]):

derive[i,j] = interplotation((i,j),derive.shape,original\_image)

return derive

k = 1.5

#%matplotlib qt5

plt.imshow(imresize(im1,im1.shape,[int(k\*i) for i in im1.shape]),cmap = 'gray')

k=0.75

plt.imshow(imresize(im1,im1.shape,[int(k\*i) for i in im1.shape]),cmap = 'gray')