

## 第二章 图像采集

学号 18071220 姓名 毛千辰

### 一、理论作业

#### 1. 《数字图像处理（第三版）》，冈萨雷斯著，P59，2.10

- 2.10 高清晰度电视(HDTV)使用 1080 条水平电视线隔行扫描来产生图像(每隔一行在显像管表面画一条线，每两场形成一帧，每场用时 1/60 秒)。图像的宽高比是 16 : 9。在水平行数不变的情况下，求图像的垂直分辨率。一家公司已经设计了一种图像获取系统，该系统由 HDTV 图像生成数字图像。在该系统中，每条(水平)电视行的分辨率与图像的宽高比成正比，彩色图像的每个像素都有 24 比特的灰度分辨率，红色、绿色、蓝色图像各 8 比特。这三幅原色图像形成彩色图像。存储 90 分钟的一部 HDTV 电影需要多少比特？

像素：1080\*16/9=1920

每帧用时：2\*1/60=1/30 则 1s 有 30 张图像

存储空间：90\*60\*1920\*1080\*30\*24=8062156800000bit

#### 2. 《数字图像处理（第三版）》，冈萨雷斯著，P59，2.11

- 2.11 考虑两个图像子集  $S_1$  和  $S_2$ ，如图 1 所示。对于  $V = \{1\}$ ，确定这两个子集是 (a) 4 邻接的，(b) 8 邻接的，还是 (c)  $m$  邻接的？

	$S_1$					$S_2$				
0	0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	1	0	0	0	1
1	0	0	1	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0
0	0	1	1	1	0	0	1	1	1	1

以 (5, 4) 和 (6, 3) 两个欧式距离最接近的为观察对象：

- (a)  $S_1$  和  $S_2$  不是 4 邻接的
- (b) 由于上面两点为 8 邻接，所以  $S_1$  和  $S_2$  是 8 邻接的。
- (c) 由 a、b，得  $S_1$  和  $S_2$  是  $m$  邻接的。

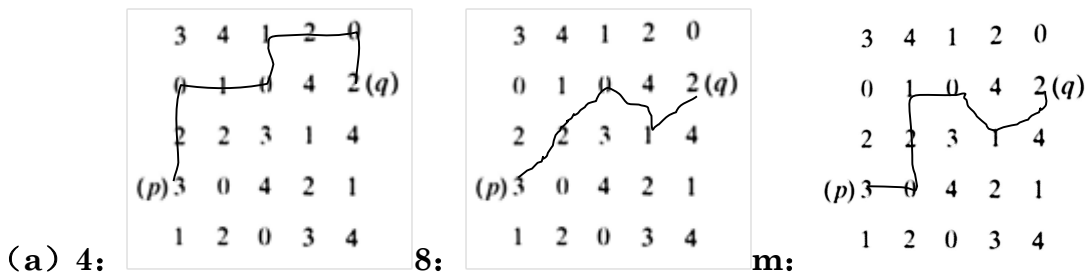
3. 《数字图像处理（第三版）》，冈萨雷斯著，P60，2.15

2.15 考虑右图的图像分割。

★(a) 令  $V = \{0, 1, 2\}$ ，计算  $p$  和  $q$  间 4、8 和  $m$  通路的最短长度。如果在这两点间不存在一个特殊通路，试解释原因。

(b) 令  $V = \{2, 3, 4\}$ ，重复问题(a)。

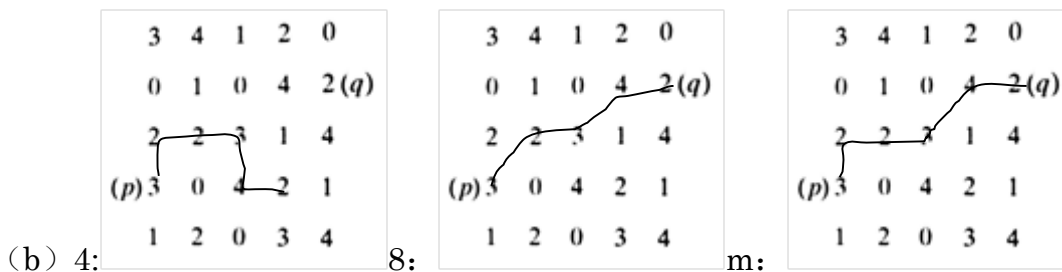
	3	4	1	2	0
	0	1	0	4	2(q)
	2	2	3	1	4
(p)	3	0	4	2	1
	1	2	0	3	4



长度为: 8

长度为: 4

长度为: 6



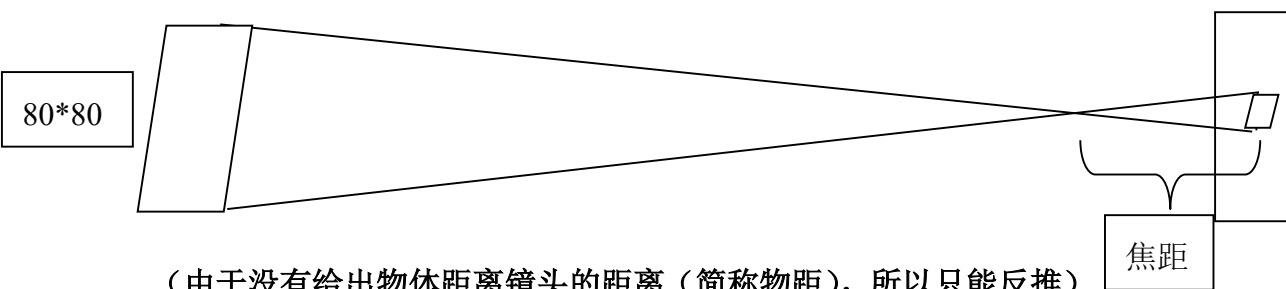
不存在

长度为: 4

长度为: 5

4. 《数字图像处理（第三版）》，冈萨雷斯著，P61，2.27

**2.27** 一家工厂生产半透明的微型聚合物方形产品。要求进行 100% 的视觉检测来严格控制质量，工厂管理者发现人工检测越来越贵，于是检测改为半自动方式。在每个检测位置，机械装置把聚合物方形产品放在一盏灯的上方，灯位于光学系统的下方，该光学系统产生方形产品的放大图像，图像完全落在大小为  $80\text{ mm} \times 80\text{ mm}$  的观察屏上。缺陷以暗斑形式出现，检测者的工作是观察屏幕。若屏幕上一个样品中出现了多个直径大于等于  $0.8\text{ mm}$  的黑斑，则丢弃该样品。管理者相信，如果能找到一种全自动的处理方法，利润将提高 50%。她还相信项目的成功是她升迁为法人的阶梯。经过大量调研后，管理者确定了解决这一问题的方法，即用 CCD 电视摄像机去观察每个检测屏幕，并把摄像机的输出送到能够检测斑点的图像处理系统中，测量它们的直径，并触发以前由检测员操作的接受和丢弃按钮。她可以找到一个系统，只要最小的缺陷在数字图像中至少为  $2 \times 2$  像素的区域发生，就可以完成该工作。假设管理者聘任你来帮助她确定摄像机和镜头系统的参数，但要求你使用现有的元件。对于镜头，假设该约束条件意味着任何  $25\text{ mm}$ 、 $35\text{ mm}$  直至  $200\text{ mm}$  焦距的整数倍。对于摄像机，这意味着分辨率为  $512 \times 512$  像素、 $1024 \times 1024$  像素或  $2048 \times 2048$  像素。这些摄像机中的各个成像元素是大小为  $8\text{ }\mu\text{m} \times 8\text{ }\mu\text{m}$  的方形，成像元素的间距为  $2\text{ }\mu\text{m}$ 。对于该应用，摄像机要远贵于镜头，因此该问题应以选择镜头为基础，尽可能使用最低分辨率的摄像机来解决。作为一名顾问，请你提供一份书面建议，对你的结论给出合理且详细的分析。使用习题 2.5 中建议的相同成像几何。



（由于没有给出物体距离镜头的距离（简称物距），所以只能反推）

### 习题 2.5 的成像模型

若要“最小的缺陷在数字图像中至少为  $2 \times 2$  像素区域发生”，也就是每个缺陷至少在图像中以  $2 \times 2$  像素的形式描绘出来的。由于以选择镜头为基础，不妨先考虑分辨率为  $512 \times 512$  像素的摄像机。则成像区的一条边长为  $512 \times (8+2) = 5120\text{ }\mu\text{m} = 5.12\text{ mm}$ 。 $80/5.12 = \text{物距}/\text{焦距} = 15.625$ ，即物距  $= 15.625 \times \text{焦距}$  时即可清晰成像。由于最小缺陷（不妨假定为正方形，其他形状只需要直径大于 2 像素即可）的边长为  $2 \times 10 = 20\text{ }\mu\text{m} = 0.02\text{ mm}$ ， $0.8\text{ mm}/\text{缺陷边长} = 15.625$ ，缺陷边长为  $51.2\text{ }\mu\text{m} > 20\text{ }\mu\text{m}$ ，所以  $512 \times 512$  的相机可以完成该任务，只需要物距保持在  $390.625 \sim 3125\text{ mm}$  即可。

## 二、编程作业

编程语言：

- Matlab（推荐） 或
- Python（可能需要使用 OpenCV）

**实现以下操作：**

（群文件中的 Lena 图，或自选）

1. 基本操作：读取图像、显示图像、加噪声
2. 几何变换：缩放、旋转、左右翻转、上下翻转

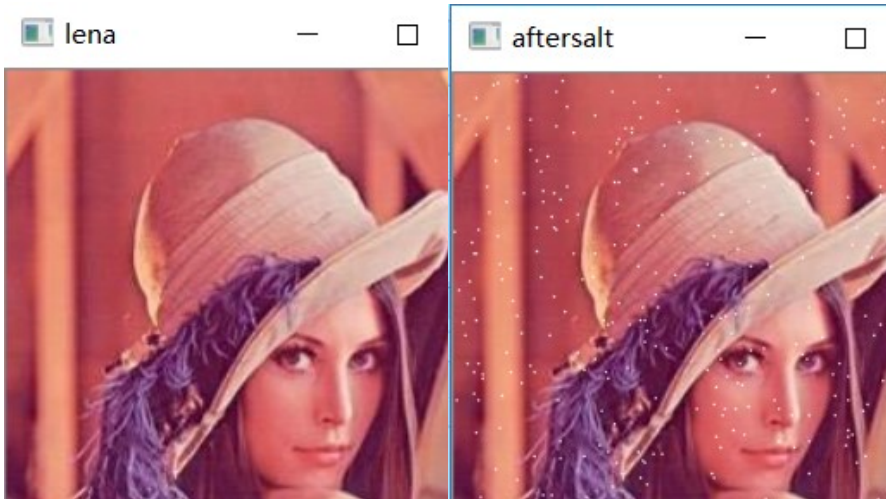
说明：

- (1) 方式一：调用 Matlab 自带函数（可以尝试使用不同的参数，了解其中的差异）
- (2) 方式二：基于原理，自行实现，进行计算（可选）
- (3) 1,2 中，各自所得图像，按照子图样式显示(*subplot*)，并标注(*title*)

1.

```
import cv2
img=cv2.imread('lena.jpg')
cv2.namedWindow('lena')
cv2.imshow('lena',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
import numpy as np
def salt(img,n):
    for k in range(n):
        i=int(np.random.random()*img.shape[1])
        j=int(np.random.random()*img.shape[1])
        img[i,j,:]=255

salt(img,500)
cv2.imshow('aftersalt',img)
cv2.namedWindow('aftersalt')
cv2.waitKey(0)
cv2.destroyAllWindows()
```



结果分析: `imshow` 和 `imread` 不必多说。对于噪声, 这里用的是简单的 `np` 随机数生成的噪声, 此外还可以实现高斯噪声。总体实现比较简单不多赘述。

2.

```
import cv2
img2=cv2.imread('lena.jpg')
img2=cv2.resize(img2,(512,512))
cv2.namedWindow('resize')
cv2.imshow('resize',img2)
cv2.waitKey(0)
cv2.destroyAllWindows()

rows,cols,chls=img2.shape
MAT=cv2.getRotationMatrix2D((cols / 2, rows / 2), 45, 0.8)
dst = cv2.warpAffine(img2, MAT, (2 * cols, 2 * rows))

cv2.imshow('45', dst)

cv2.waitKey(0)
cv2.destroyAllWindows()

ximg=cv2.flip(img2,1)
yimg=cv2.flip(img2,0)
```

```
cv2.imshow('x', ximg)
cv2.imshow('y', yimg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



结果分析：opencv 库实现这些功能还是比较简单的（有现成的 api），若要手动实现，其实就是应用 numpy 将图片各个像素点的位置调换一下即可。

**作业文档要求：**

1. 内容：

(1) 理论作业：推荐直接在 word 内编辑，也可以手写拍照（处理后）贴到 word 里。

(2) 编程作业：按照各个部分，依次提供代码、实验结果图和结果分析

2. 要求：内容完备（包含计算过程），结构清晰、排版美观

**作业提交要求：**

- 1) 截止时间：**2019 年 9 月 23 日 20:00**
- 2) 提交 **PDF** 文档（附件）
  - 文档命名为：《数字图像处理》课程作业 1（学号 姓名）
- 3) 发送邮件到：gaofeihifly@163.com
  - 邮件标题为：《数字图像处理》课程作业 1（学号 姓名）