**数字图像处理第二次作业**

浮焕然 PB22061345

问题：

1. 马赫带和同时对比度反映了什么共同问题

人类视觉感知的主观亮度并不是物体表面照度的简单函数，而是受到物体与周围环境亮度相对关系的影响。

1. 列举几个视觉错觉的例子

环绕错觉：当一个人在旋转的房间里站立时，起初会感到自己在向相反方向旋转。当房间停止旋转后，他会感到自己在向与房间旋转方向相同的方向旋转。

三维视觉错觉：一些静态图像通过特殊的绘制方式营造出三维效果，让人感觉图像中有深度和立体感，但实际上它只是平面上的图案，这种错觉利用了人们对光影和透视的感知。

1. 计算5\*5领域各像素到中心像素的欧氏距离，街区距离和棋盘距离

表 1.欧氏距离

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 2 |  |  |
|  |  | 1 |  |  |
| 2 | 1 | 0 | 1 | 2 |
|  |  | 1 |  |  |
|  |  | 2 |  |  |

表 2.街区距离

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 2 |  |  |
| 3 | 2 | 1 |  |  |
| 2 | 1 | 0 | 1 | 2 |
|  |  | 1 |  |  |
|  |  | 2 |  |  |

表 3.棋盘距离

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 2 |  |  |
|  |  | 1 |  |  |
| 2 | 1 | 0 | 1 | 2 |
|  |  | 1 |  |  |
|  |  | 2 |  |  |

编程：

1. 编写一个程序，打开灰度图像lena.bmp，读出以（200，200）为左上角的10\*10区域的像素值。

图 1.以200,200为左上角的10\*10区域像素

1. 编写一个程序，打开灰度图像lena.bmp，将前256行像素设为255，打印处理后的图像。

图 2.前256行像素为255

1. 任意修改灰度的lena.bmp 的彩色映像表，写出你的修改方法，给出修改后图像打印显示（彩色打印）。

修改方法：

red = i

green = 255 - i

blue = 255 – i

结果如下：

from PIL import Image,ImageDraw

# 打开图像

image\_path = "Lena.bmp"

image = Image.open(image\_path)

image\_1 = Image.open(image\_path)

# 确保图像是灰度模式

if image.mode != 'L':

print('translate successfully')

image = image.convert('L')

# 定义裁剪区域 (左上角坐标和右下角坐标)

# format: (left, upper, right, lower)

crop\_region = (199, 199, 209, 209) # 示例：xyxy

# 裁剪图像

cropped\_image = image.crop(crop\_region)

# 保存裁剪后的图像

cropped\_image.save("homework2/cropped\_image.bmp")

#填充白色

white\_image=ImageDraw.Draw(image)

crop\_region= (0,0,511,255)

white\_image.rectangle(crop\_region, fill=255)

image.save("homework2/white\_image.bmp")

#换成彩色

palette = []

for i in range(256):

# 将灰度值映射到彩色值

# 可以自由修改这里的逻辑来创建不同的彩色映像表

red = i

green = 255-i

blue = 255-i

palette.extend((red, green, blue))

# 将灰度图像转换为彩色图像

color\_image = image\_1.convert('P') # 转换为调色板模式

color\_image.putpalette(palette) # 应用调色板

# 保存彩色图像

color\_image.save("homework2/colorful\_lena.bmp")