

# 数字图像处理习题课

2025/6/6

# 第二次作业

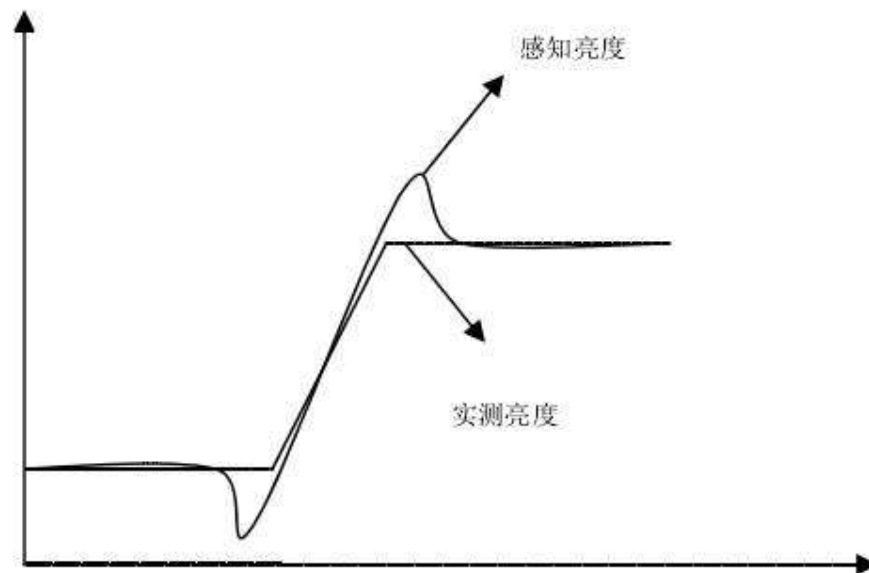
## 1.马赫带和同时对比度反映了什么共同问题?

答:

马赫带效应: 在亮度边界处觉得亮的更亮, 暗的更暗

同时对比度: 人眼对某个区域的亮度感受不仅与区域本身的亮度有关, 也与其和周围区域的相对亮度有关

这两种现象都说明了人眼看到的画面与客观世界并不一定相符, 主观亮度并不是物体表面照度的简单函数

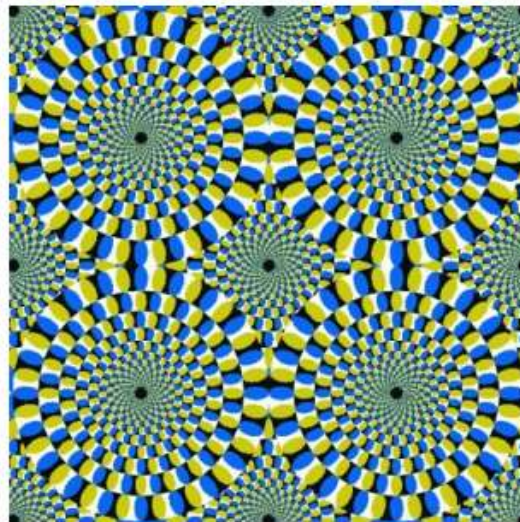


先解释马赫带和同时对比度, 再总结反映的问题



# 第二次作业

## 2.列举几个视觉错觉的例子



这张图片其实是静态的



两根红线一样长

# 第二次作业

## 3.计算5×5邻域各像素到中心像素的欧式距离,街区距离和棋盘距离

欧式距离:

$$\begin{bmatrix} 2\sqrt{2} & \sqrt{5} & 2 & \sqrt{5} & 2\sqrt{2} \\ \sqrt{5} & \sqrt{2} & 1 & \sqrt{2} & \sqrt{5} \\ 2 & 1 & 0 & 1 & 2 \\ \sqrt{5} & \sqrt{2} & 1 & \sqrt{2} & \sqrt{5} \\ 2\sqrt{2} & \sqrt{5} & 2 & \sqrt{5} & 2\sqrt{2} \end{bmatrix}$$

街区距离:

$$\begin{bmatrix} 4 & 3 & 2 & 3 & 4 \\ 3 & 2 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 2 & 3 \\ 4 & 3 & 2 & 3 & 4 \end{bmatrix}$$

棋盘距离:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

# 第二次作业

## 4. 编写一个程序，打开灰度的lena.bmp

- 读出以 (200, 200) 为左上角的10\*10区域的像素值。
- 将前256 (后修改为128) 行像素设为255，打印处理后的图像
- 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)

# 第二次作业

Offset	Length	Name	Description
0	2	bfType	"BM"
2	4	bfSize	Size of file
6	2	bfReserved1	0
8	2	bfReserved2	0
10	4	bfOffBits	Offset after file head

Offset	Length	Name	Description
14	4	biSize	Size of bitmap head, 40
18	4	biWidth	Width of Image
22	4	biHeight	Height of Image
26	2	biPlanes	Always "1"
28	2	biBitCount	Bits/pixel, 1,4,8 or 24
30	4	biCompression	Size of compressed file
34	4	bfSizeImag	Offset after file head

## 读取相关信息

# 获取图像宽度和高度（在文件头的 18 至 22 字节和 22 至 26 字节中）

```
width = int.from_bytes(bmp_data[18:22], 'little')
```

```
height = int.from_bytes(bmp_data[22:26], 'little')
```

# 获取颜色深度（每个像素的位数）

```
bpp = int.from_bytes(bmp_data[28:30], 'little') # 颜色深度 (bits per pixel)
```

# 获取图像数据的起始偏移位置

```
offset = int.from_bytes(bmp_data[10:14], 'little')
```

# BMP 标头之后，调色板一般从第 54 字节开始  
color\_map\_start = 54

# 第二次作业

在 **BMP 文件格式** 中，图像的每一行是 **从下到上** 存储的，即：

- **最后一行（底部的行）** 是存储在数据的开始部分。
- **第一行（顶部的行）** 存储在数据的末尾部分。

图像左上角第一个像素，记为  $(0, 0)$

在 **BMP 文件** 的调色板中，颜色值是按 **BGR（蓝、绿、红）**

错误：直接构造并合并三个彩色通道

# 第三次作业

(1) 计算下图的 DFT, DCT, Hadamard 变换和 Haar 变换

0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0



# 第三次作业

- 两种方法
- 按点计算（在方阵的情况下需要除以阶数）
- 按照两次变换计算  $T = FAF^T$

## 4.5.5 二维离散傅里叶变换及其反变换

类似于 4.3 节和 4.4 节的材料中的推导过程，可得到下面的二维傅里叶变换 (DFT)：

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)} \quad (4.5-15)$$

其中， $f(x, y)$  是大小为  $M \times N$  的数字图像。与一维中的情况一样，式 (4.5-15) 必须对离散变量  $u$  和  $v$  在  $u = 0, 1, 2, \dots, M-1$  和  $v = 0, 1, 2, \dots, N-1$  范围内求值<sup>①</sup>。

给出变换  $F(u, v)$ ，我们可以使用傅里叶反变换 (IDFT) 得到  $f(x, y)$ ：

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)} \quad (4.5-16)$$

其中， $x = 0, 1, 2, \dots, M-1$ ， $y = 0, 1, 2, \dots, N-1$ 。式 (4.5-15) 和式 (4.5-16) 构成了二维离散傅里叶变换对。本章的其余部分是以这两个公式的特性和它们在频率域图像滤波的应用为基础的。

有文献中，我们有时会在 DFT 而不是 IDFT 的前面看到常数  $1/MN$ 。这时，该常数表示为  $1/\sqrt{MN}$  并包含在正变换和反变换的前面，从而创建一个更为对称的变换对。这些公式都是正确的，但前提条件是要保持一致。

# 第三次作业

$$3.1 \quad f = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

DFT:

$$F(u, v) = \frac{1}{4} \sum_{x=0}^{4-1} \sum_{y=0}^{4-1} f(x, y) e^{-j2\pi(\frac{ux}{4} + \frac{vy}{4})}$$

$$w = e^{-\frac{\pi}{2}j}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^3 \\ 1 & w^2 & w^4 & w^6 \\ 1 & w^3 & w^6 & w^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-\frac{\pi}{2}j} & e^{-\pi j} & e^{-\frac{3\pi}{2}j} \\ 1 & e^{-\pi j} & e^{-2\pi j} & e^{-3\pi j} \\ 1 & e^{-\frac{3\pi}{2}j} & e^{-3\pi j} & e^{-\frac{9\pi}{2}j} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$F(u, v) = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} * \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -j-1 & 0 & j-1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)} \quad \begin{matrix} u = 0, 1, \dots, M-1 \\ v = 0, 1, \dots, N-1 \end{matrix}$$

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)} \quad \begin{matrix} x = 0, 1, \dots, M-1 \\ y = 0, 1, \dots, N-1 \end{matrix}$$

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \exp\left[\frac{-j2\pi ux}{N}\right] \sum_{y=0}^{N-1} f(x, y) \exp\left[\frac{-j2\pi vy}{N}\right]$$

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \exp\left[\frac{j2\pi ux}{N}\right] \sum_{v=0}^{N-1} F(u, v) \exp\left[\frac{j2\pi vy}{N}\right]$$

# 第三次作业

DCT:

$$F(u, v) = \frac{2}{\sqrt{4+4}} c(u) c(v) \sum_{x=0}^{4-1} \sum_{y=0}^{4-1} f(x, y) \cos\left[\frac{\pi}{2+4} (2x+1)u\right] \cos\left[\frac{\pi}{2+4} (2y+1)v\right]$$

$$c = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \cos\frac{1}{8}\pi & \cos\frac{3}{8}\pi & \cos\frac{5}{8}\pi & \cos\frac{7}{8}\pi \\ \cos\frac{2}{8}\pi & \cos\frac{6}{8}\pi & \cos\frac{10}{8}\pi & \cos\frac{14}{8}\pi \\ \cos\frac{3}{8}\pi & \cos\frac{9}{8}\pi & \cos\frac{15}{8}\pi & \cos\frac{21}{8}\pi \end{bmatrix}$$

$$F(u, v) = c * \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} * c^T = \begin{bmatrix} 2 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Hadamard:

$$H = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$H(u, v) = H * f(x, y) * H^T = \begin{bmatrix} 2 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C(u, v) = a(u)a(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$$u, v = 0, 1, \dots, N-1$$

$$a(u) = \begin{cases} \sqrt{1/N} & \text{when } u = 0 \\ \sqrt{2/N} & \text{when } u = 1, 2, \dots, N-1 \end{cases}$$

Basis function matrix

$$C_N = \sqrt{\frac{2}{N}} \begin{bmatrix} \overset{x}{\underset{u}{\downarrow}} & 1/\sqrt{2} & 1/\sqrt{2} & \dots & 1/\sqrt{2} \\ \cos\pi/2N & \cos 3\pi/2N & \dots & \cos(2N-1)\pi/2N \\ \cos 2\pi/2N & \cos 6\pi/2N & \dots & \cos 2(2N-1)\pi/2N \\ \cos 3\pi/2N & \cos 9\pi/2N & \dots & \cos 3(2N-1)\pi/2N \\ \cos 4\pi/2N & \cos 12\pi/2N & \dots & \cos 4(2N-1)\pi/2N \\ \vdots & \vdots & \ddots & \vdots \\ \cos(N-1)\pi/2N & \cos 3(N-1)\pi/2N & \dots & \cos(2N-1)(N-1)\pi/2N \end{bmatrix}$$

$$H(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) (-1)^{\sum_{i=0}^{n-1} [b_i(x) p_i(u) + b_i(y) p_i(v)]}$$

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} = H_2 \otimes H_N$$

# 第三次作业

Haar:

$$H = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

$$H(u, v) = H * f(x, y) * H^T = \begin{bmatrix} 2 & 0 & -\sqrt{2} & \sqrt{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

# 第三次作业

2. 设有一组 $64*64$ 的图像,它们的协方差矩阵是单位矩阵.如果只使用一半的原始特征值计算重建图像,那么原始图像和重建图像间的均方误差是多少?

# 第三次作业

$$C_x = E\{(x - m_x)(x - m_x)^T\}$$

MSE 是丢弃的特征值的总和。  $m_x = E\{x\}$

将  $64 \times 64$  的一幅图像用一个  $(64 \times 64) \times 1$  的向量表示  
则其协方差矩阵的大小为  $(64 \times 64) \times (64 \times 64)$   
依题意知它的协方差矩阵为单位矩阵  
 $\therefore \lambda_i = 1 \quad i = 64 \times 64, 64 \times 64 - 1, \dots, 1$   
 $\therefore \text{rms} = \sum_{i=1}^{64 \times 64} \lambda_i = \sum_{i=1}^{64 \times 64} 1 = 2048$

# 第三次作业

- DFT
- DCT

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \exp\left[-\frac{j2\pi ux}{N}\right] \sum_{y=0}^{N-1} f(x, y) \exp\left[-\frac{j2\pi vy}{N}\right]$$

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \exp\left[\frac{j2\pi ux}{N}\right] \sum_{v=0}^{N-1} F(u, v) \exp\left[\frac{j2\pi vy}{N}\right]$$

$$Y(u, v) = \sum_{x=0}^{N-1} \left[ \sum_{y=0}^{N-1} f(x, y) \cdot \cos\left[\frac{\pi}{N} \cdot \left(y + \frac{1}{2}\right) \cdot v\right] \right] \cdot \cos\left[\frac{\pi}{N} \cdot \left(x + \frac{1}{2}\right) \cdot u\right]$$

## 3. 对lena.bmp进行DCT和DFT变换

- 直接进行计算
- 根据可分离性原理：先对行变换，后列变换
- 也可以分块变换
- 对于原图进行变换可能会过暗，可以使用 $\lg(1+\text{像素值})$ 的方法

# 第三次作业

```
# 实现 DFT 变换
def dft_1d(signal):
    N = len(signal)
    result = np.zeros(N, dtype=complex)
    for k in range(N):
        sum_val = 0
        for n in range(N):
            sum_val += signal[n] * np.exp(-2j * np.pi * k * n / N)
        result[k] = sum_val
    return result

def dft_2d(image):
    # 对每一行应用 DFT
    dft_img = np.zeros_like(image, dtype=complex)
    for i in range(image.shape[0]):
        dft_img[i, :] = dft_1d(image[i, :])

    # 对每一列应用 DFT
    dft_img_final = np.zeros_like(dft_img, dtype=complex)
    for j in range(dft_img.shape[1]):
        dft_img_final[:, j] = dft_1d(dft_img[:, j])
    return dft_img_final
```



# 第四次作业

1. 为什么一般情况下对离散图像的直方图均衡并不能产生完全平坦的直方图？

离散图像灰度级是离散的，同一灰度级的像素不能映射到不同的灰度级上

在离散图像的直方图均衡化过程中，由于灰度级的离散性、映射函数的量化近似以及像素分布的天然不均匀性，通常无法得到完全平坦的直方图。

# 第四次作业

2. 设已用直方图均衡化技术对一幅图象进行了增强，试证明再用这个方法对所得结果增强并不会改变其结果。

首次直方图均衡化

设原始图像的灰度值为  $r$ ，经过第一次直方图均衡化后，得到新的灰度值  $s$ ，

根据上述变换函数有：
$$s = T(r) = (L - 1) \int_0^r p_r(\omega) d\omega$$

其中  $p_r(\omega)$  是原始图像灰度值的概率密度函数。经过均衡化后，新图像的灰度值  $s$  在  $[0, L-1]$  上近似均匀分布，其概率密度函数  $p_s(s) \approx \frac{1}{L-1}$

第二次直方图均衡化

设第二次均衡化的变换函数为  $v = G(s)$ ，则有：

$$v = G(s) = (L - 1) \int_0^s p_s(\tau) d\tau$$

于  $p_s(s) \approx \frac{1}{L-1}$ ，将其代入上式可得：


$$v = G(s) = (L - 1) \int_0^s \frac{1}{L-1} d\tau$$

这表明，第二次直方图均衡化后的灰度值  $v$  与第一次均衡化后的灰度值  $s$  相等。也就是说，再次使用直方图均衡化技术对已经经过均衡化处理的图像进行增强，不会改变图像的结果。

## 第四次作业

2. 设已用直方图均衡化技术对一幅图象进行了增强，试证明再用这个方法对所得结果增强并不会改变其结果。

第一次过后，概率直方图  $S_k$



累积直方图  $t_k = \sum_{i=1}^K S_i$

映射函数  $p_k = F(k) = \text{int}[(L-1) \sum_{i=1}^K S_i + 0.5]$

$k \rightarrow p_k$

第二次过后，概率直方图  $n_k = \sum_{i=a_k}^{b_k} S_i$  ( $a_k \leq i \leq b_k, p_i = k$ )

累积直方图  $m_k = \sum_{i=0}^K n_k = \sum_{i=0}^K \sum_{j=a_i}^{b_i} S_j = \sum_{i=0}^{b_k} S_i = t_{b_k}$

映射函数  $q_k = G(k) = \text{int}[(L-1) \sum_{i=1}^{b_k} S_i + 0.5] = p_{b_k} = k$

$k \rightarrow q_k \rightarrow k$

关键点：

- 映射
- 直方图累计和不变
- 经过第一次均衡后 直方图中概率不为零的项， 再进行直方图均衡得到的 是到自身的映射

## 第四次作业

3. 讨论用于空间滤波的平滑滤波器和锐化滤波器的相同点、不同点以及联系。

相同点：滤波

不同点：滤除高频分量/低频分量

联系：在一定表述下可以相互转化。从原始图像中减去平滑滤波器的结果得到锐化滤波器的效果

联系：对应参数的平滑滤波器与锐化滤波器的平均约为原图像

4. 有一种常用的图象增强技术是将高频增强和直方图均衡化结合起来以达到使边缘锐化的反差增强效果，  
以上两个操作的先后次序对增强结果有影响吗？为什么？

高频增强是线性变换，直方图均衡化是非线性变换，调换操作顺序会影响图像结果

答：有影响。如果首先应用直方图均衡化，这将会使图像的灰度级分布更加均匀，提高对比度。然后再进行高频增强，可以使边缘更加清晰，从而产生更强烈的边缘锐化和图像细节增强；如果首先进行高频增强，这将会突出图像的高频细节，增强图像的边缘和细节。然后再应用直方图均衡化，这将使各个像素值的分布更加均匀，有时可能会减弱通过高频增强所突出的部分细节，导致结果不如直接先进行直方图均衡化再进行高频增强的情况。

# 第四次作业

5. 编程实现对lena.bmp分别加入高斯噪声和椒盐噪声， 再进行局域平均和中值滤波。

- 加入**高斯噪声**后需要裁剪至有效像素范围[0,255]
- **边界处理**：边缘的像素无法覆盖整个窗口
  - 对图像边界填充或者边界特殊处理
- **数据处理**：确保操作后的数据仍是unit8类型
- **窗口选择**：一般选择奇数尺寸的窗口避免窗口中央位置的偏移。

# 第五次作业

- 列出逆滤波和维纳滤波图象恢复的具体步骤。

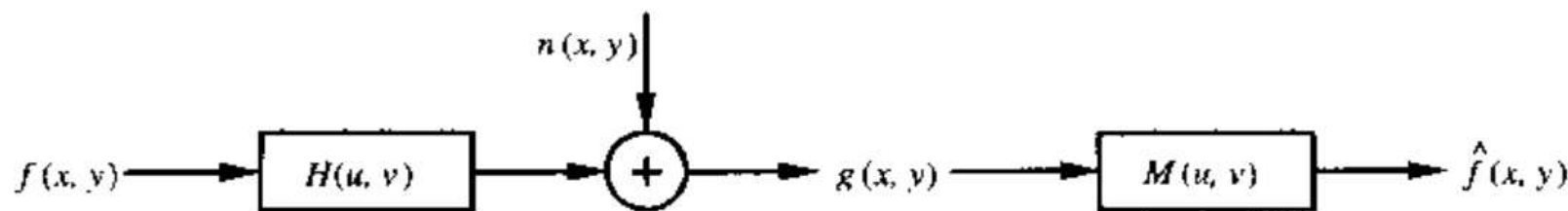


图 5.3.1 图象退化和恢复模型

退化模型:  $G(u, v) = H(u, v) F(u, v) + N(u, v)$

恢复模型:  $\hat{F}(u, v) = G(u, v) M(u, v)$

$H(u, v)$  为点扩展函数  $h(x, y)$  的傅里叶变换

$n(x, y)$  为加性噪声  $g(x, y)$  为退化后图像

$M(u, v)$  为恢复模型  $\hat{F}(u, v)$  为恢复的图像

# 第五次作业

- 列出逆滤波和维纳滤波图象恢复的具体步骤。

- 写出逆滤波和维纳滤波图像恢复的具体步骤

逆滤波  
模型

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

$$\hat{F}(u, v) = \frac{G(u, v) - N(u, v)}{H(u, v)} = \frac{G(u, v)}{H(u, v)} - \frac{N(u, v)}{H(u, v)} = G(u, v)M(u, v)$$

$$M(u, v) = \frac{1}{H(u, v)} - \frac{N(u, v)}{G(u, v)H(u, v)} \approx \frac{1}{H(u, v)}$$

考虑到噪声普遍是高频信号，可以采用如下的逆滤波函数 $M(u, v) = \begin{cases} 1/H(u, v) & \text{if } u^2 + v^2 < w_0^2 \\ 1 & \text{else} \end{cases}$

或者,我们可以限制H过小, 得到 $M(u, v) = \begin{cases} k & \text{if } H(u, v) < d \\ 1/H(u, v) & \text{else} \end{cases}$

# 第五次作业

- 列出逆滤波和维纳滤波图象恢复的具体步骤。

$$\hat{F}(u, v) = G(u, v)H_W(u, v)$$
$$H_W(u, v) = \frac{1}{H(u, v)} \times \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_n(u, v)/S_f(u, v)} = \frac{H(u, v)^*}{|H(u, v)|^2 + S_n(u, v)/S_f(u, v)}$$
$$S_n(u, v) = |N(u, v)|^2 \quad \text{噪声谱能量}$$
$$S_f(u, v) = |F(u, v)|^2 \quad \text{功率谱能量}$$

$$H_W(u, v) = \frac{H(u, v)^*}{|H(u, v)|^2 + K}$$



# 第五次作业

- 推导水平匀速直线运动模糊的点扩展函数的数学公式并画出曲线

参考答案:

实际采集的运动模糊图像为:

$$g(x, y) = \int_0^T f[x - x_0(t), y - y_0(t)] dt$$

傅立叶变换可表示为:

$$\begin{aligned} G(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \exp[-j2\pi(ux + vy)] dx dy \\ &= \int_0^T \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)] \exp[-j2\pi(ux + vy)] dx dy \right] dt \\ &= F(u, v) \int_0^T \exp\{-j2\pi[ux_0(t) + vy_0(t)]\} dt \end{aligned}$$

点扩展函数为:

$$H(u, v) = \int_0^T \exp\{-j2\pi[ux_0(t) + vy_0(t)]\} dt$$

# 第五次作业

- 推导水平匀速直线运动模糊的点扩展函数的数学公式并画出曲线
- 物体经过时间  $T$ ，会移动  $c$  的距离。

水平匀速直线运动  $x_0(t) = ct / T, y_0(t) = 0$

则：

$$H(u, v) = \int_0^T \exp[-j2\pi uct / T] dt = \frac{T}{\pi uc} \sin(\pi uc) \exp[-j\pi uc]$$

已知傅立叶变换：

$$f(t) = \begin{cases} T / c, & |t| < c / 2 \\ 0, & |t| \geq c / 2 \end{cases} \Leftrightarrow T \frac{\sin(\pi uc)}{\pi uc}$$

由平移性质：

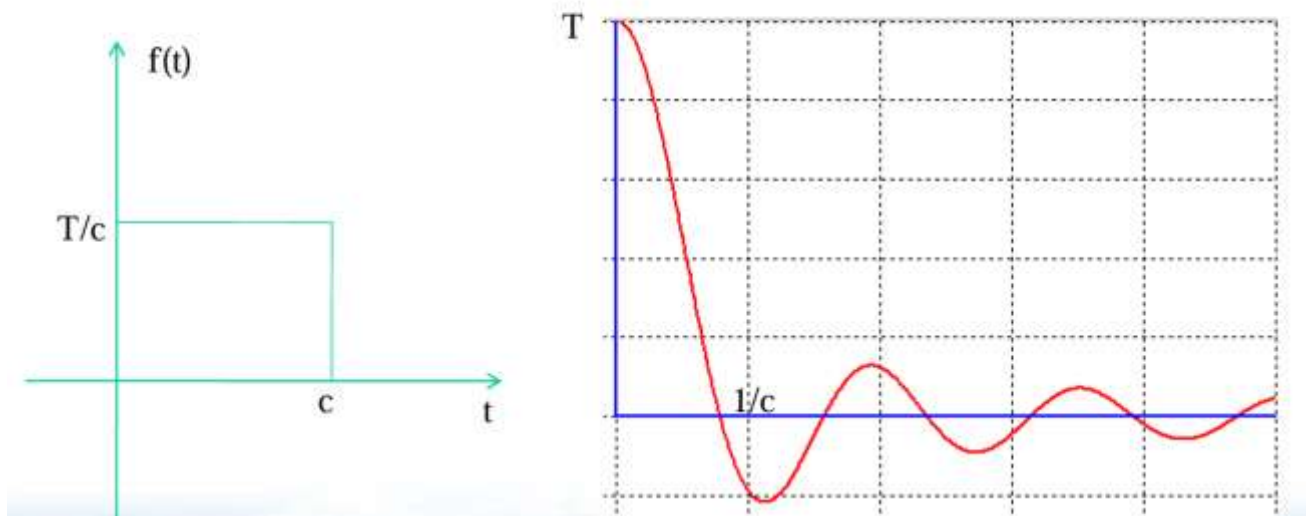
$$T \frac{\sin(\pi uc)}{\pi uc} \exp(-j\pi uc) \Leftrightarrow f(t - c / 2)$$

$$T \frac{\sin(\pi uc)}{\pi uc} \exp(-j\pi uc) \Leftrightarrow f(t) = \begin{cases} T / c, & 0 < t < c \\ 0, & \text{else} \end{cases}$$

# 第五次作业

- 推导水平匀速直线运动模糊的点扩展函数的数学公式并画出曲线

$$T \frac{\sin(\pi uc)}{\pi uc} \exp(-j\pi uc) \Leftrightarrow f(t) = \begin{cases} T/c, 0 < t < c \\ 0, \text{ else} \end{cases}$$



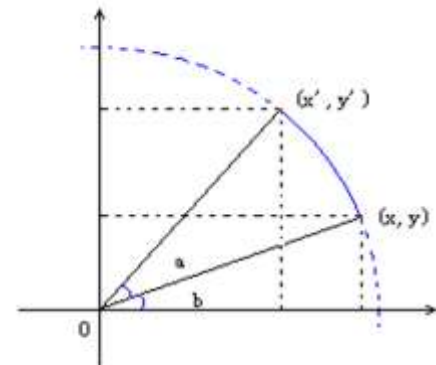
# 第五次作业

正向映射

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

反向映射

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$



- 编程实现lena.bmp的任意角旋转

- 正映射：原先像素映射到结果图中的位置，适当作平均化处理
- 逆映射：在结果图中寻找原先图的位置取该位置的像素值
- 进行旋转之前要先将坐标中心移到图像中心点，扩大和缩小
- 旋转步骤：
  - 根据旋转角度计算新图的空间，并初始化为0。
  - 对新图的每一个点利用逆映射公式得到其在原图中的对应点位置(浮点数)。
  - 使用插值方法(1.最近邻2.双线性3.其他)计算当前点的灰度值。

# 第六次作业

- 简述CT的发明过程
- 略

# 第六次作业

- 试证明投影定理
- 傅里叶切片定理：图像的投影的傅里叶变换与图像的二维傅里叶变换的切片相等

$$\begin{aligned} G(\omega, \theta) &= \int_{-\infty}^{\infty} g(\rho, \theta) e^{-j2\pi\omega\rho} d\rho \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x\cos\theta_k + y\sin\theta_k - \rho_j) e^{-j2\pi\omega\rho} dx dy d\rho \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \left[ \int_{-\infty}^{\infty} \delta(x\cos\theta_k + y\sin\theta_k - \rho_j) e^{-j2\pi\omega\rho} d\rho \right] dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi\omega(x\cos\theta + y\sin\theta)} dx dy \end{aligned}$$

$G$ 是某个角度 $\theta$ 上的密度函数， $f$ 是整体的密度函数， $\delta$ 是阶跃函数，令 $u = w\cos\theta, v = w\sin\theta$

$$\begin{aligned} G(\omega, \theta) &= \left[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \right]_{u=\omega\cos\theta; v=\omega\sin\theta} \\ &= [F(u, v)]_{u=\omega\cos\theta; v=\omega\sin\theta} \\ &= F(\omega\cos\theta, \omega\sin\theta) \end{aligned}$$

## 第七次作业

- 请说明是否能用变长编码法压缩1幅已直方图均衡化的具有 $2^n$ 级灰度的图？
- 这样的图像中包含像素间冗余吗

一幅灰度图作直方图均衡化后，离散灰度等级下，  
 $2^n$ 个灰度等级的频率（概率）不一定相等。

故应用变长编码法仍能对其进行一定程度的压缩。

但由于这些灰度等级的频率（概率）较为接近，  
故并不能得到很好的压缩效果。

直方图均衡化做的事情是

将同一灰度等级的像素变换为另一个灰度等级（有可能不变），  
图像整体的几何结构（原先表达的内容）没有产生实质性变化  
故图像仍然包含像素间冗余

第七次作业

- 对一个具有3个符号的信源，有多少唯一的Huffman码？
- 构造这些码

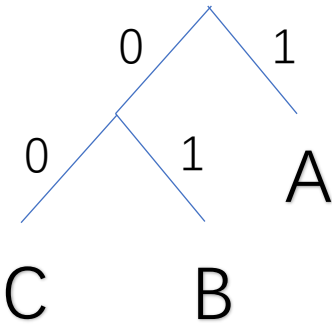
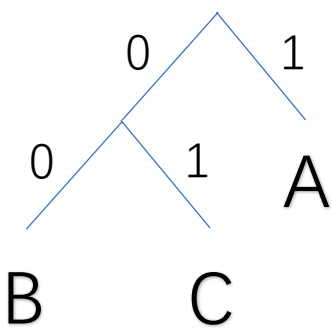
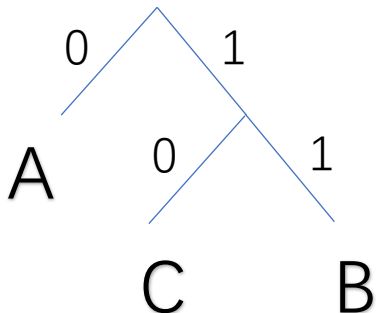
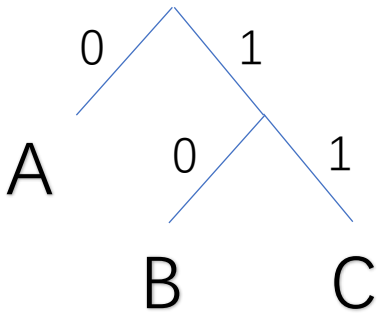
概率从左至右递减

symbol1	symbol2	symbol3
0	10	11
0	11	10
1	00	01
1	01	00

A

B

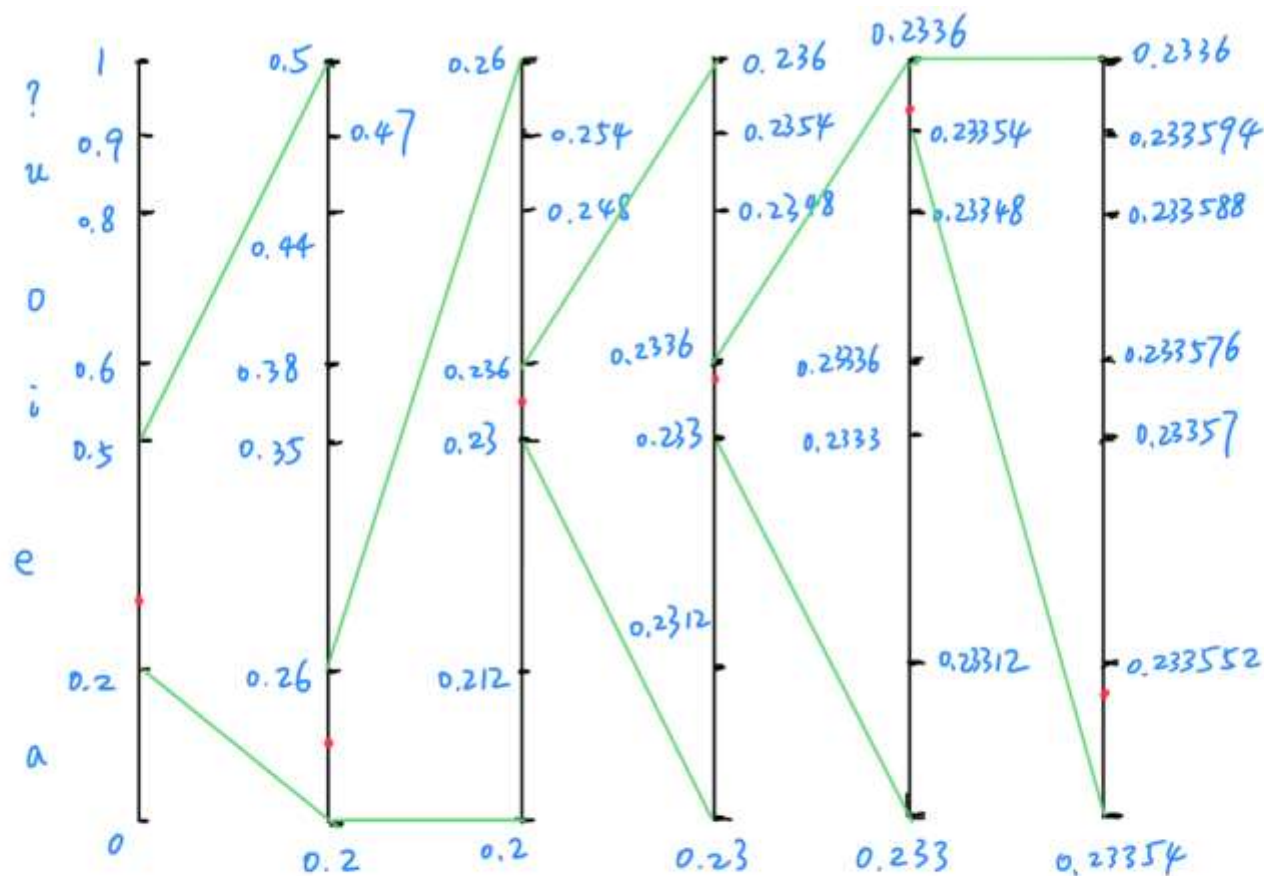
C





## 第七次作业

- 已知符号a,e,i,o,u,?的出现概率分别是0.2, 0.3, 0.1, 0.2, 0.1, 0.1
- 对0.23355解码, 解码长度为6



ea ii ? u

若从1开始, 结果为oueeoe

第一个符号:  $0.23355 \in [0.2, 0.5) \rightarrow e$

调整数值:  $(0.23355 - 0.2) / 0.3 \approx 0.111833$

第二个符号:  $0.111833 \in [0.0, 0.2) \rightarrow a$

调整数值:  $0.111833 / 0.2 \approx 0.559167$

第三个符号:  $0.559167 \in [0.5, 0.6) \rightarrow i$

调整数值:  $(0.559167 - 0.5) / 0.1 \approx 0.591667$

第四个符号:  $0.591667 \in [0.5, 0.6) \rightarrow i$

调整数值:  $(0.591667 - 0.5) / 0.1 \approx 0.916667$

第五个符号:  $0.916667 \in [0.9, 1.0) \rightarrow ?$

调整数值:  $(0.916667 - 0.9) / 0.1 \approx 0.166667$

第六个符号:  $0.166667 \in [0.0, 0.2) \rightarrow a$

最终解码结果为: ea ii ? a